RECONSTRUCTING THE CLIMATE OF NORTH AMERICA DURING THE PAST 2,000 YEARS USING POLLEN DATA

Matthew Jared Ladd

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
for the Doctorate in Philosophy degree in Geography

Department of Geography
Faculty of Arts
University of Ottawa

©Matthew Jared Ladd, Ottawa, Canada, 2014

Abstract

July temperature (TJUL) and total annual precipitation (ANNP) are reconstructed to better understand the spatial and temporal patterns of change in North America over the last 2,000 years using pollen databases. Using a customized application in R, the reconstructions use a composite averaging of multiple site reconstructions that show a distinct warmer Medieval Warm Period (MWP) compared to the colder Little Ice Age (LIA). Results show that, both multi-centennial scale periods are reconstructed as cooler than the last 50 years. Regional time series from several forested ecoregions show positive anomalies up to 0.6°C during the MWP and anomalies up to -0.3°C during the LIA.

In order to test whether the TJUL reconstructions are biased to the modern calibration climate data, we show a distinct difference between the reconstructions when using station versus reanalysis-based modern TJUL fields. Reconstructions using station-based modern calibration data sets better reflect the centennial to multi-centennial scale climate variability as compared to the reanalysis-based modern calibration data sets that reveal a warm-bias. We justify the choice of the Whitmore *et al.* (2005) modern data set for large-scale pollen-based paleoclimate reconstructions.

Finally we use Local Indicators of Spatial Association (LISA) to spatially filter the ANNP reconstructions in order to distinguish regional hydroclimate patterns from local site-specific conditions. Results show that a La Nina, positive North Atlantic Oscillation (+NAO) and positive Atlantic Multidecadal Oscillation (+AMO) state-like dominated both the MWP and Roman Warm Period (RWP), although the MWP was generally drier. In contrast, the Dark Ages Cold (DAC) period was likely dominated by El Nino, negative NAO and negative AMO state-like circulation. Minimum solar and high volcanic activity is likelyto have contributed to more complex hydroclimate regional patterns during the LIA.

The results presented in this dissertation can be used as benchmark data sets for future climate data-model comparisons in order to improve our understanding of natural climate variability during the past 2,000 years in the context of modern human-induced climate change.

Acknowledgements

Since an early age, I have been interested in the weather and climate; it was my experience as an undergraduate student at Carleton University that encouraged research into paleoclimate and climate change. I considered an advanced degree in meteorology, but the emphasis on pure math and physics was a discouragement. I was due to take a course in permafrost, but my course work lacked the prerequisite soils course, which did not fit into my schedule that term. The irony was that my brother had taken permafrost with the same professor many years earlier without the soils course. I decided not to argue about it and accepted the fact that I wasn't going to be able to take the course in permafrost. The course that did fit my schedule to replace the permafrost course was Quaternary Geography, a 4th year course that piqued my interest in paleoclimate. After completion of a B.Sc. in Geomatics at Carleton, I started a Masters Program with Konrad Gajewski. I am very grateful for the opportunity which the M.Sc. Geography program provided to pursue the development of research skills. I persevered and completed the M.Sc., then started a Ph.D. in the same lab as my Masters Program.

I would like to thank my supervisor Andre Viau for his dedication to helping me as a graduate student and researcher. With his guidance and support I was able to complete this doctoral degree in a timely fashion. To the committee members: Konrad Gajewski, Michael Sawada and Christian Zdanowicz for their help with every aspect of the Ph.D. I'm very grateful for your suggestions and encouragement throughout this process. I would also like to thank my external examiner, Elisabeth Levac from Bishops University for her effort and comments in the process. In addition, thanks to Luke Copland for filling in for Christian Zdanowicz for the defense. A special mention goes to my family and friends who supported me throughout my entire academic career, I could not have done this without you. Although I have completed four ironman triathlons, the very long road to completing the doctorate has been the ultimate ironman. I am grateful for the funding received from the Ontario Graduate Scholarship (OGS), the Faculty of Graduate and Postdoctoral Studies (admission scholarship and travel grants), the Graduate Students Association (GSAED)

and CUPE2626 (travel grants) and the Faculty of Arts at the University of Ottawa, Department of Geography (support and teaching assistantships). The research assistantships with my committee members in areas different from my thesis research were extremely valuable.

To my colleagues throughout my career as a graduate student and everyone at the Laboratory for Paleoclimatology and Climatology (LPC), the list is too long to name, I wish you all the best in your future endeavours. A special mention is made to Robert Way, who provided valuable input for the papers in Chapter 3 and 4. Also to Phil Bonnaventure who always provided advice about aspects of the doctoral program and general career progression and Kate Ploeger, who started the program at the same time as me (we both took GEG5105 in the winter of 2009), I will remember the trips to Godiva for chocolates. Finally to my colleague Paul Beckwith with whom I have had some great discussions about abrupt climate change and the politics of climate change, you have made an outstanding effort to raise awareness of climate change among the general public. There is no question that our planet is in grave danger from the man-made changes to our climate. The window of time for action on climate change is rapidly closing as the concentration of carbon dioxide in the atmosphere is now at 400 ppm, well above the safe threshold of 350 ppm.

List of Abbreviations

- **14C** Carbon-14 or radiocarbon dating/date
- **210Pb** Lead-210 dating/date
- **AGCM2** Atmospheric Global Climate Model 2
- **AMO** Atlantic Multidecadal Oscillation
- **AMOC** Atlantic Meridional Overturning Circulation
- **ANNP** Total annual precipitation (in mm)
- **anomaly** Reconstructed value minus modern value
- **AO** Arctic Oscillation
- **AOGCM** Atmosphere-Ocean General Circulation Model
- **ArcGIS** GIS software developed by ESRI
- **BCE** Before Common Era
- **BEST** Berkeley Earth Surface Temperature data set
- **BerkeleyEarth** R package to download and process BEST climate data
- **BP** Before present or years before 1950 CE
- **C2** Software for analysing and visualising palaeoenvironmental data developed by S. Juggins
- **CALIB** Software for calibrating radiocarbon dates developed by M. Stuiver and P.J. Reimer
- **CDO** Climate Data Operators software developed by Max Planck Institute (MPI)
- **CE** Common Era

- **CFSR** Climate Forecast Systems Reanalysis
- **CLIMAP** Climate: Long range Investigation, Mapping, and Prediction Project
- **COHMAP** The Cooperative Holocene Mapping Project
- **CRU** Climate Research Unit at the University of East Anglia
- **DEGRIB** National Digital Forecast Database (NDFD) GRIB2 Decoder software developed by the National Weather Service (NWS)
- **DAC** Dark Ages Cold Period from 350 to 650 CE
- **EBK** Emprical Bayesian Kriging
- **ECMWF** European Center For Medium Range Weather Forecasting
- **ENSO** El Nino Southern Oscillation
- **ERA** ECMWF Reanalysis
- **ERA20c** ECMWF atmospheric reanalysis of the 20th century from 1900 to 2012
- **ERA-40** ECMWF reanalysis of the global atmosphere and surface conditions from September 1957 to August 2002
- **ERA-INT** ECMWF atmospheric reanalysis from 1979 to present
- **ESRI** Environmental Systems Research Institute
- **GDD0** Growing Degree Days (base of 0°C)
- **GDD5** Growing Degree Days (base of 5°C)
- **GHCN** Global Historical Climate Network
- **GIS** Geographic Information Systems
- **GRIB** Gridded binary data format
- **HADCRUT** A climate data set combining the sea surface temperature records from the Hadley Centre of the UK Met Office with the land surface air temperature records compiled by the CRU of the University of East Anglia
- **HTM** Holocene Thermal Maximum

- **IntCal'09** A radiocarbon dating calibration curve developed by Reimer *et al.* (2009)
- **IntCal'04** A radiocarbon dating calibration curve developed by Reimer *et al.* (2004)
- **IntCal'98** A radiocarbon dating calibration curve developed by Stuiver *et al.* (1998)
- **IPCC** Intergovernmental Panel on Climate Change
- **ITCZ** Inter-tropical convergence zone
- **ka** Kilo annum, equivalent to one thousand years
- **LAT** Latitude in decimal degrees
- **LGM** Last Glacial Maximum
- **LIA** Little Ice Age from 1550 CE to 1850 CE
- **LISA** Local Indicators of Spatial Association
- **LON** Longitude in decimal degrees
- **MAD** Mean absolute difference
- **MAT** Modern Analog Technique
- **MATTOOLS** An R package developed by M. Sawada to perform Modern Calibration Functions for the Modern Analog Technique (MAT)
- **Mb** Milibar, a measure of atmospheric pressure comparable to HPa (hectopascal)
- **MCA** Medieval Climate Anomaly
- **MERRA** NASA reanalysis for the satellite era using a major new version of the Goddard Earth Observing System Data Assimilation System Version 5 (GEOS-5)
- **MI** Local Moran's I statistic
- **MIPT** Priestley-Taylor Moisture Index (actual evapotransipration divided by potential evapotransipration)

- **Mod** Modern
- **MPI** Max Planck Institute
- **MTCO** Mean temperature of the coldest month of the year (same as TJAN)
- **MTWA** Mean temperature of the warmest month of the year (same as TJUL)
- **MWP** Medieval Warm Period from 950 to 1250 CE
- **NA** Missing value in a data set
- **NAO** North Atlantic Oscillation
- **NAPD** North American Pollen Database
- **NARR** North American Regional Reanalysis
- **NASA** National Aeronautics and Space Administration
- **NCEP R1** The original (first generation) NCEP/NCAR reanalysis product
- **NCEP R2** The NCEP-DOE (Department of Energy) Reanalysis 2 is an improved version of the NCEP R1 reanalysis
- **NCEP/NCAR** National Center For Environmental Prediction / National Center for Atmospheric Research
- **Neotoma** A paleoecological database developed by E. Grimm
- **NetCDF** Network Common Data Format
- **NEW** New *et al.* (2002) climate data produced by the CRU
- **NNR1** NCEP/NCAR reanalysis for the period of 1961–1990
- **NNR2** NCEP/NCAR reanalysis for the period of 1981–2010
- **NRC** National Research Council
- **NSERC** National Science and Engineering Research Council
- **PAGES** Past Global Changes
- **PAGES NAM2k** Past Global Changes North America 2k Netowrk
- **PCA** Prinicipal Component Analysis

- **PDJF** Total Precipitation for December, January, February (in mm)
- **PDO** Pacific Decadal Oscillation
- **PDSI** Palmer Drought Severity Index
- **PETM** Paleocene-Eocene Thermal Maximum
- **PJJA** Total Precipitation for June, July, August (in mm)
- **PMAM** Total Precipitation for March, April, May (in mm)
- **PMIP3** Paleoclimate Model Intercomparison Project (Phase 3)
- **PNA** Pacific North American teleconnection pattern
- **PSON** Total Precipitation for September, October, November (in mm)
- **R** A free open-source software programming language and software environment for statistical computing and graphics
- **RGHCNv3** An R package to download and process GHCN climate data
- **ROOKCASE** An Excel 97/2000 Visual Basic (VB) Add-in for Exploring Global and Local Spatial Autocorrelation developed by M. Sawada
- **RWP** Roman Warm Period from 150 BCE to 150 BCE
- **SCD** Squared Chord Distance metric
- **SE_Res** Standard Error of residuals
- **SE_Slope** Standard Error of the slope
- **SPEI** Standardized Precipitation and Evapotranspiration Index
- **SST** Sea surface temperatures
- **St_Dev** Standard Deviation
- **TAVE** Mean Annual Temperature (in °C)
- **TJAN** Mean January Temperature (same as MTCO)
- **TJUL** Mean July Temperature (same as MTWA)
- **VBA** Visual Basic For Applications

- **XLS** MS Excel spreadsheet file format
- **X-Y** Spatial coordinates where X is easting and Y is northing
- **WCLIM** WorldClim modern climate data
- **WHIT** Whitmore modern climate data
- **Whitmore** A modern pollen database with associated bioclimatic data at each site developed by Whitmore *et al.* (2005)
- **WorldClim** A set of global climate grids with a spatial resolution of $\sim$1 km^2
- **zMI** Z-score for Local Moran's I statistic

Table of Contents

Abstract ii

Acknowledgements iii

List of Abbreviations v

List of Tables xv

List of Figures xvii

1 Introduction 1

1.1 Background . . . 1
1.2 Literature review . . . 2
1.2.1 Early large-scale paleoclimate syntheses . . . 2
1.2.2 More recent large-scale paleoclimate syntheses . . . 3
1.2.3 LGM to Mid-Holocene climate change . . . 4
1.2.4 Climate reconstructions of the last 2,000 years . . . 5
1.2.5 Hydroclimate reconstructions & spatial filtering . . . 5
1.3 Research objectives . . . 7
1.4 Study area, data & methods . . . 8
1.4.1 Study area . . . 8
1.4.2 Pollen data . . . 8
1.4.3 Climate data . . . 9
1.4.4 Methods . . . 10
1.5 Thesis organization . . . 12
References . . . 14

2 The climate of North America during the past 2,000 years reconstructed from pollen data 23

2.1 Introduction . . . 24
2.2 Methods . . . 26

2.2.1 Data . . . 26
2.2.2 Climate reconstruction . . . 27
2.2.3 Mapping . . . 28
2.2.4 Time series . . . 29
2.3 Results . . . 29
2.4 Discussion and conclusions . . . 30
References . . . 38

3 The impact of using different modern climate data sets in pollen-based paleoclimate reconstructions of North America 44
3.1 Introduction . . . 45
3.2 Data and Methods . . . 46
3.2.1 Fossil and modern pollen data . . . 46
3.2.2 New *et al.*, 2002 (NEW) . . . 46
3.2.3 Global Historical Climate Network (GHCN) . . . 47
3.2.4 Berkeley Earth Surface Temperature (BESTstn/gr1/2) . . . 47
3.2.5 World Climate (WCLIM) . . . 47
3.2.6 North American regional reanalysis (NARR) . . . 48
3.2.7 NCEP/NCAR reanalysis (NNR1, NNR2) . . . 48
3.2.8 ECMWF ERA-40 (ERA40) & ERA-Interim (ERAI) reanalyses 48
3.3 Results . . . 49
3.3.1 Comparison of reconstructions . . . 49
3.3.2 Comparison of reconstructed modern versus observed TJUL . 50
3.3.3 Comparison of TJUL by ecoregion . . . 50
3.4 Discussion . . . 50
3.4.1 Differences and similarities between the reconstructions . . . 50
3.5 Summary and Conclusions . . . 52
References . . . 61

4 Variations in precipitation over North America over the past 2,000 years 65
4.1 Introduction . . . 66
4.2 Data and Methods . . . 67
4.2.1 Pollen data . . . 67
4.2.2 Modern Analogue Technique (MAT) . . . 67
4.2.3 Spatial filtering and mapping . . . 68
4.3 Results . . . 69
4.3.1 Reconstructed patterns of ANNP for last 2,000 years . . . 69

4.3.2 Time-series by ecoregions . . . 70
4.4 Discussion . . . 70
4.4.1 Reconstructed hydroclimatic conditions in North America . . 71
4.4.2 Ocean-Atmospheric linkages . . . 72
4.5 Conclusion . . . 73
References . . . 84

5 Summary **90**
5.1 Conclusions . . . 90
5.2 Summary of Objective 1 . . . 91
5.3 Summary of Objective 2 . . . 91
5.4 Summary of Objective 3 . . . 92
5.5 Summary of Objective 4 . . . 93
5.6 Future Research Directions . . . 93
References . . . 95

All Thesis References **96**

A Overview of Appendices **113**

B User Guide for R Program **115**
B.1 MODULE 1: Pollen database editor and reconstruction builder . . . 115
B.2 MODULE 2: Pollen paleoclimate reconstruction analysis program . . 134
B.3 Acknowledgements . . . 161
B.4 Known and documented bugs in the program . . . 162

C Source Code for R Program **163**

D Additional Methods & Scripts **275**
D.1 Overview of reconstruction methodology . . . 275
D.2 Calculation of site resolution in a reconstruction . . . 301
D.3 Calibration of radiocarbon dates using IntCal '09 . . . 305
D.4 Excel Macros for linear interpolation and averaging anomalies . . . 305
D.5 Spatial filtering steps & Excel macro to spatially filter reconstructions 320
D.6 Steps to acquire and use GHCN and WorldClim data for modern climate data in a reconstruction . . . 325
D.7 Steps for revising figures in Chapter 2 (Viau *et al.*, 2012) . . . 327
D.8 Javascript tool to generate MATTOOLS code . . . 328

D.9 Example of Climate Data Operator (CDO) commands to process a NetCDF data file 332
D.10 Overview of data-model comparison methods 333

E Supplementary Plots, Maps & Tables **334**

F All Appendix References **380**

List of Tables

1 Pollen taxa used in reconstructions [Williams and Shuman (2008)]. . . 33

2 Summary of all modern TJUL data sets tested for TJUL reconstruction of the last 2,000 years. Data references (from top row to bottom row) are: Whitmore *et al.* (2005), Lawrimore *et al.* (2011), Rohde *et al.* (2012), Hijmans *et al.* (2005), New *et al.* (2002), Rohde *et al.* (2013), Messinger *et al.* (2006), Kistler *et al.* (2001), Uppala *et al.* (2005) and Dee *et al.* (2011). 54

3 Statistics for observed TJUL versus reconstructed modern (1950 CE) TJUL. SE_Slope is standard error of the slope, SE_Res is standard error of residuals and MAD is mean absolute difference. 54

4 Difference in standard deviation by ecoregion of selected modern TJUL compared to average standard deviation by ecoregion of all modern TJUL (difference from last column in table). 55

5 Pollen taxa used in reconstructions [Williams and Shuman (2008)]. . . 82

6 Positive, zero, and negative values and combinations for Moran's I and Moran's I (z-values) and the number of neighbors & percentages for sites at 100 year time intervals for the reconstruction. 83

7 TAVE reconstruction for the top analog 138

8 TAVE reconstruction after linear interpolation through time 140

9 Average anomaly and standard error for TAVE reconstruction with tuned dates 142

10 Average anomaly and standard error for TAVE reconstruction with BP dates 143

11 Reconstructed TAVE anomalies through time (with standard errors) for the forest-tundra ecoregion. 148

12 Reconstructed TAVE anomalies for the time slice of 200-800 BP (1550-1850 CE). 149

13 Reconstructed site density through time in North America 155

14 Number of sites through time for reconstruction of MTWA (using 35 pollen sum) . . . 301
15 Resolution in years for selected pollen sites . . . 304
16 Linear interpolation test results (part 1) . . . 312
17 Linear interpolation test results (part 2) . . . 313
18 Positive, zero, and negative values and combinations for Moran's I and Moran's I (z-values) and the number of neighbors & percentages for sites at different lag distances . . . 321
19 Cumulative frequencies from the table of results above . . . 321
20 Same as above except for Geary's C (z-value) . . . 322
21 Code assigned to combinations of values (see key below) . . . 322
22 Coded sites at different lag distances based on key below . . . 322
23 Sample output from Rookcase. 4th column is local Moran's I with a 50 km lag distance. . . . 325
24 Calendar year BP dates tuned to BCE/CE dates. . . . 334
25 Number and percentage of sites retained after the spatial filtering procedure is applied. The average, median, min and max are for 0-2500 BP at 100 year time steps. . . . 334

List of Figures

1 Spatial distribution of modern (A) and fossil (B) pollen sites in North America (Viau, Ladd, and Gajewski, 2012) 34

2 Maps of the anomalies of mean temperature of the warmest month (MTWA). (A) Medieval Warm Period (MWP; 650–1050 AD) in relation to the modern (1961–1990 base period). (B) Little Ice Age (LIA; 1550–1850 AD) in relation to the modern. (C) MWP minus LIA. Hatched regions indicate areas of low confidence reconstructions and are adapted from Sawada *et al.* (2004). 35

3 Time series of regional averages of the mean temperature of the warmest month (MTWA) anomalies for the period 350–1950 AD. All time series within the biomes were interpolated to 100-year intervals and averaged. Uncertainty envelopes were computed as in Viau *et al.* (2006). Note that the instrumental record is not included in the time series. 36

4 Squared chord distance (SCD) values for the top 5 analogs (A1 to A5) and MTWA anomaly, site density, dates and samples for (A) coastal, (B) mountain, (C) prairies, (D) desert, (E) forest-tundra, (F) boreal, (G) conifer-hardwood and (H) deciduous regions. These ecoregions were aggregated from Fedorova *et al.* (1994) to correspond to the major biomes of North America. The Arctic and Mediterranean regions were not included in this study due to lack of data. On the right are shown the mean temperature of the warmest month anomalies (MTWA), the number of sites averaged for each interpolated value, the radiocarbon dates used for the regional chronologies and the number of pollen samples available for the region. Note that these plots are not chronologically tuned as in Figure 3. 37

5 Distribution of fossil (A) and modern (B) pollen sites and modern TJUL from WorldClim (C). 56

6 Reconstructions of MTWA/TJUL anomalies (in °C) for the last 2,000 years using all modern data sets (A), 1961–1990 base period data sets (B) and 1979–2000 base period data sets (C). 57
7 Comparison of the range of station-based (black) versus reanalysis-based (red) MTWA/TJUL reconstructions for the last 2,000 years. The approximate age ranges for the RWP, DAC, MWP and LIA periods are indicated. 58
8 Reconstructed anomaly map (gridded at 100 km) for the average of 1050–1550 CE using different modern TJUL data sets as follows: Whitmore 1961–1990 (A), New *et al.* 1961–1990 (B), GHCN 1961–1990 (C), WorldClim 1950–2000 (D), BEST gridded 1961–1990 (E), BEST gridded 1979–2000 (F), BEST station 1961–1990 (G), NCEP 1961–1990 (H), BEST gridded 1979–2000 (I), ERA-Interim 1979–2000 (J), NARR 1979–2000 (K) and NCEP 1981–2010 (L). 59
9 Reconstruction of MTWA/TJUL anomalies (in °C) for the last 2,000 years using the WorldClim TJUL modern data (1 km) gridded at resolutions of 0.5°, 1° and 2.5°. 60
10 Distribution of modern (A) and fossil (B) pollen sites along with modern ANNP (C). 75
11 Ecoregions of North America adapted from Fedorova *et al.* (1994). . . 76
12 Reconstructed 0 BP (1950 CE) versus observed for ANNP using all original data points (A) and retained data points after spatial filtering is applied (B). 77
13 ANNP anomalies at 1300 BP [1050 CE] using all original data points (A), original data points with outliers excluded (B) and retained data points after spatial filtering with outlier data points excluded (C). Anomalies range from dark brown (-100 mm/yr) to dark aqua (+100 mm/yr). 78
14 Spatially filtered maps of ANNP anomalies at 100 year intervals for 150 BCE to 1850 CE interpolated using empirical bayesian kriging (gridded at 250 km). Anomalies range from dark brown (-100 mm/yr) to dark aqua (+100 mm/yr). 79
15 ANNP anomalies (in mm) for the RWP [150 BCE–150 CE] minus DAC [350–650 CE] (A), DAC minus MWP [950–1250 CE] (B), MWP minus LIA [1550–1850 CE] (C), RWP minus MWP (D), DAC minus LIA (E) and LIA minus modern (F). Anomalies range from dark brown (-100 mm/yr) to dark aqua (+100 mm/yr). 80

16 Average ANNP anomalies (in mm) through time by ecoregion in North America. The boreal and forest-tundra regions were divided into west and east using a threshold of 90°W (Figure 11). Panel A includes boreal west, forest-tundra west and forest-tundra east regions. Panel B includes conifer-hardwood, boreal east, deciduous forest, prairies and desert regions. The approximate age ranges for the RWP, DAC, MWP and LIA periods are indicated. 81
17 Long term mean July temperature map (1950 to 2000 CE) from WorldClim data (TMEAN7 raster layer). 126
18 Map of fossil and modern pollen sites in North America 133
19 Reconstruction site distribution at 0 BP or 1950 CE 139
20 The output file SpatialJoinReconModWithAnCalc.csv viewed in an MS Excel™worksheet. 141
21 The plot of reconstructed TAVE anomalies for the last 2,000 years. . 144
22 Plot of reconstructed TAVE anomalies through time by ecoregion. NOTE: Only 4 ecoregions are shown. 147
23 Map of TAVE anomalies for selected time slice, classified into negative, zero and positive categories. 150
24 Gridded map of TAVE anomalies for LIA relative to modern. 151
25 Contour map of TAVE anomalies for LIA relative to modern. 152
26 Squared-Chord Distance (SCD) plots by ecoregion. 154
27 Squared-Chord Distance (SCD) plot for all of North America. 154
28 Plot of reconstructed site density through time in North America. . . 156
29 Map of filtered TAVE anomalies for selected time slice, classified into negative, zero and positive categories. 157
30 Comparison of IntCal '09 (navy blue) versus IntCal '04 (magenta) time-series. Note that the Viau *et al.* (2006) reconstruction used IntCal '98. 306
31 Reconstruction of Whitmore climate variables (Panel 1) 336
32 Reconstruction of Whitmore climate variables (Panel 2) 337
33 Reconstruction of Whitmore climate variables (Panel 3) 338
34 Map of four major ecoregions of North America (adapted from Fedorova *et al.*, 1994). 339
35 Reconstruction of MTWA (anomalies in °C) using 35 (red), 64 (blue), 134 (green), 135 (gold) pollen sums and average of all 4 pollen sums (black). 339

36 MTWA reconstructed anomaly map at 100 BP (range of anomalies is +/- 2°C). 340
37 MTWA reconstructed anomaly map at 200 BP (range of anomalies is +/- 2°C). 340
38 MTWA reconstructed anomaly map at 300 BP (range of anomalies is +/- 2°C). 341
39 MTWA reconstructed anomaly map at 400 BP (range of anomalies is +/- 2°C). 341
40 MTWA reconstructed anomaly map at 500 BP (range of anomalies is +/- 2°C). 342
41 MTWA reconstructed anomaly map at 600 BP (range of anomalies is +/- 2°C). 342
42 MTWA reconstructed anomaly map at 700 BP (range of anomalies is +/- 2°C). 343
43 MTWA reconstructed anomaly map at 800 BP (range of anomalies is +/- 2°C). 343
44 MTWA reconstructed anomaly map at 900 BP (range of anomalies is +/- 2°C). 344
45 MTWA reconstructed anomaly map at 1000 BP (range of anomalies is +/- 2°C). 345
46 MTWA reconstructed anomaly map at 1100 BP (range of anomalies is +/- 2°C). 345
47 MTWA reconstructed anomaly map at 1200 BP (range of anomalies is +/- 2°C). 346
48 MTWA reconstructed anomaly map at 1300 BP (range of anomalies is +/- 2°C). 346
49 MTWA reconstructed anomaly map at 1400 BP (range of anomalies is +/- 2°C). 347
50 MTWA reconstructed anomaly map at 1500 BP (range of anomalies is +/- 2°C). 347
51 MTWA reconstructed anomaly map at 1600 BP (range of anomalies is +/- 2°C). 348
52 MTWA reconstructed anomaly map at 1700 BP (range of anomalies is +/- 2°C). 348
53 MTWA reconstructed anomaly map at 1800 BP (range of anomalies is +/- 2°C). 349

54 MTWA reconstructed anomaly map at 1900 BP (range of anomalies is +/- 2°C). 349
55 MTWA reconstructed anomaly map at 2000 BP (range of anomalies is +/- 2°C). 350
56 MTWA reconstructed anomaly map at 2100 BP (range of anomalies is +/- 2°C). 350
57 MTWA reconstructed anomaly map at 2200 BP (range of anomalies is +/- 2°C). 351
58 MTWA reconstructed anomaly map at 2300 BP (range of anomalies is +/- 2°C). 351
59 MTWA reconstructed anomaly map at 2400 BP (range of anomalies is +/- 2°C). 352
60 MTWA reconstructed anomaly map at 2500 BP (range of anomalies is +/- 2°C). 352
61 Location of BEST climate stations in North America (blue) and location of modern pollen sites (red). 353
62 Minimum, mean and maximum of grid point average (black) versus area-weighted average (red) MTWA/TJUL anomalies through time. . 353
63 Reconstructed anomaly map of MTWA/TJUL at 0 BP (1950 CE) using modern TJUL data sets with highest (NARR) (A) versus lowest (NCEP1) (C) standard deviations. Whitmore (B) is close to the median standard deviation of all data sets. 354
64 Frequency of reconstructed MTWA/TJUL anomalies at 950 and 1650 CE (courtesy of R.G. Way). 354
65 Ratio of unique fossil pollen sites divided by fossil pollen samples through time (by ecoregion). 355
66 Reconstructions of MTWA/TJUL anomalies (in °C) for the last 2,000 years using all modern data sets (composite of sites in boreal forest region only). 355
67 Long term mean July temperature map (1950 to 2000 CE) from WorldClim data (TMEAN7 raster layer) for North America. 356
68 Standard deviation of all modern TJUL data sets (1961–1990) at each modern pollen site. Lower (higher) standard deviations are indicated with colder (wamer) colours. 356
69 Scatterplot of Whitmore MTWA (TJUL) versus NCEP TJUL modern (1961-1990). 357

70 Difference in modern TJUL for the nearest WorldClim grid point to each modern pollen site. Red (blue) modern pollen sites are warmer (colder) than nearest WorldClim grid point. 358
71 Difference in modern TJUL for the nearest NARR grid point to each modern pollen site. Red (blue) modern pollen sites are warmer (colder) than nearest NARR grid point. 359
72 Difference in modern TJUL for the nearest GHCN station to each modern pollen site. Red (blue) modern pollen sites are warmer (colder) than nearest GHCN station. 360
73 MTWA (TJUL) anomalies through time (composite average of all sites in North America) using different modern TJUL data sets. 360
74 Range of modern TJUL values in °C assigned to each modern pollen site. 361
75 Percentage of sites retained through time for the ANNP reconstruction after the application of spatial filtering. 362
76 Sites retained for ANNP reconstruction after spatial filtering at different lag distances from 50 km to 500 km. 363
77 Squared Chord Distance (SCD) plot of the top five analogs for the reconstruction of ANNP. 364
78 Frequency distribution of ANNP reconstruction data points (outliers not removed) versus ANNP reconstruction data points (outliers removed) [courtesy of R.G. Way]. 365
79 ANNP anomalies (in mm) for the MWP (950–1250 CE) minus LIA (1550–1850 CE) (A), LIA minus MWP (B), RWP (150 BCE–150 CE) minus DAC (350–650 CE) (C), DAC minus RWP (D), MWP minus RWP (E), RWP minus MWP (F), LIA minus DAC (G) and DAC minus LIA (H). This was the original version of Figure 15. 366
80 Long term total annual precipitation map (1950 to 2000 CE) from WorldClim data (BIOCLIM12 raster layer) for North America. Aqua (brown) areas are wetter (drier). 367
81 Reconstruction of ANNP (anomalies are in mm/yr) using 35 (red), 64 (blue), 134 (green), 135 (gold) pollen sums and average of all 4 pollen sums (black). 367
82 Reconstruction of ANNP example with site distribution and hatched areas indicating low confidence reconstructions and sample land mask (red border). Note that purple sites are not retained after filtering. . 368

83 Example of the large distance between a fossil pollen site and the closest modern analog (modern pollen site) for the reconstruction of ANNP in Eastern Canada. 369
84 Flowchart of the spatial filtering methodology that was developed by the author. 369
85 Table of results for spatial filtering on a reconstruction of MTWA for the last 2,000 years. Counts/percentages are based on sites retained when spatial filtering at different lag distances in km. 370
86 Reconstruction of MTWA (filtered) at 1300 BP and gridded at 2°. Anomalies in red (blue) are warmer (colder) than modern. 370
87 Standard error of prediction map for unfiltered reconstruction of PDSI at 2500 BP. Black (white) areas have a higher (lower) standard error of prediction. 371
88 Example of filtered reconstruction of PDSI at 2500 BP. Black (white) areas have a higher (lower) standard error of prediction. 372
89 Unfiltered reconstruction of MIPT at 500 BP with site distribution. Brown (aqua) areas are drier (wetter). 373
90 Filtered reconstruction of MIPT at 500 BP with site distribution. Brown (aqua) areas are drier (wetter). 374
91 Standard error of prediction map with site distribution for filtered reconstruction of MIPT at 500 BP. Red (white) areas have a higher (lower) standard error of prediction. 375
92 Filtered reconstruction of MIPT at 500 BP with areas in grey having anomalies exceeding +/-1 standard deviation from the average standard error of prediction for the reconstruction at 500 BP. 376
93 Filtered reconstruction of MIPT at 500 BP with areas in grey having anomalies exceeding +/-2 standard deviations from the average standard error of prediction for the reconstruction at 500 BP. 377
94 Filtered reconstruction of MIPT at 500 BP with areas in grey having anomalies exceeding +/-3 standard deviations from the average standard error of prediction for the reconstruction at 500 BP. 378
95 Map of ecoregions of North America (adapted from Fedorova *et al.*, 1994). 379

Chapter 1

Introduction

1.1 Background

The climate of the Earth is an open, complex and nonlinear system that responds to both external forcings and internal variability (Peixoto and Oort, 1992). Climate varies on all time and space scales ranging from billions of years to the seasonal cycle and from local to global scales (Peixoto and Oort, 1992). While some climate changes operate gradually and are well understood, such as orbital scales, others occur abruptly and are generally more poorly understood (Clark *et al.*, 2008; IPCC, 2013).

Much recent scientific effort has been on the global warming trend of the past century which represents a challenge due to human contributions of greenhouse gases (GHGs) in our atmosphere (Hansen *et al.*, 2008; IPCC, 2013). Unlike past climate changes, this unique combination of forcings (both natural and human caused) represents a non-analogue period in the history of the Earth's climate (Jackson and Williams, 2004). A central research question to this recent warming is to determine how much of this warming trend is attributable to natural causes (e.g. orbital-scale changes, solar variability, volcanic activity and internal free variability) from that of human-induced changes in atmospheric composition and land use (Ammann *et al.*, 2007; IPCC, 2013). The main difficulty in attributing the human fraction of this recent warming mainly results from an instrumental record that is simply too short to distinguish the natural causes from that of human-induced changes (IPCC, 2013). Therefore, it is imperative to extend our knowledge of past natural climate variability on all time and space scales to put the current warming trend into perspective.

In order to answer how the climate system will respond to recent human activities is complex and requires regional, continental, hemispheric and global syntheses of past climates using proxy data in order to extend the instrumental record back in

time (IPCC, 2013). One of the major challenges facing climate change research today is the lack of large-scale paleoclimate data syntheses that can be used as benchmark data sets in validating climate models to improve our predictive capabilities under a global warming scenario (Bartlein *et al.*, 2011). Although there have been efforts in large-scale climate reconstruction syntheses that have proven useful in data-model comparisons (e.g. COHMAP Members, 1988; Wright Jr. *et al.*, 1993; Sawada *et al.*, 1999; Viau and Gajewski, 2001; Esper *et al.*, 2002; Sawada *et al.*, 2004; Mann *et al.*, 2009a; Bartlein *et al.*, 2011; Braconnot *et al.*, 2012), there remains unresolved issues such as the resolution of the different proxies, the lack of data in certain regions, and more importantly, the spatial heterogeneity that exists between available proxy climate reconstructions on regional scales (National Research Council, 2006; IPCC, 2013).

1.2 Literature review

1.2.1 Early large-scale paleoclimate syntheses

An early attempt to predict and map climate conditions including vegetation of the last glacial maximum (LGM) was Climate: Long Range Investigation, Mapping and Prediction (CLIMAP) (CLIMAP Project Members, 1981). CLIMAP proposed that the sea surface temperatures in the tropics at the time of the LGM were warmer than today (Pinot *et al.*, 1999). Today, we know that this is not the case and furthermore, the tropics have a subdued response to climate forcings (Braconnot *et al.*, 2012). Following CLIMAP, the first large-scale data-model comparison was the Cooperative Holocene Mapping Project (COHMAP Members, 1988). COHMAP explored the mechanisms of climate variability over the past 18,000 years at 18ka, 12ka, 9ka, 6ka and present (COHMAP Members, 1988; Wright Jr. *et al.*, 1993). This was accomplished by comparing model simulated climates with the paleoclimate data reconstructions in order to determine how well models can simulate past climates under different conditions from today (Harrison *et al.*, 1998; Edwards *et al.*, 2007). For example, the COHMAP study showed stronger monsoons in Asia and Africa from 12ka to 6ka as a result of orbitally induced increases in insolation. One shortcoming resulting from the COHMAP study was that models could not simulate certain large-scale patterns observed in the paleoclimate record, for example the northward shift in the African-Asian monsoon during the mid-Holocene ($\sim$6 ka) or anomalies in sea surface temperatures (SSTs) during the LGM ($\sim$21 ka) (Wright Jr. *et al.*, 1993).

1.2.2 More recent large-scale paleoclimate syntheses

Confidence in climate model simulations of future climate requires global models to be able to simulate past climates under vastly different conditions than from today (IPCC, 2007; IPCC, 2013). Paleoclimate reconstructions which are based on empirical observations of past climate conditions are used to improve model behaviour and help in improving model predictions for the future (Braconnot *et al.*, 2012). Currently, fully coupled Atmosphere-Ocean General Circulation Models (AOGCMs) used for future simulations of the evolution of our climate require significant computing resources and still have a very coarse resolution with many processes that have to be parameterized (hardwired) (Neelin, 2011; IPCC, 2013). Testing these models requires large-scale regional benchmark paleoclimate data sets at a resolution that compares to models. This dissertation contributes towards this objective by developing a North American paleoclimate pollen-based reconstruction database for North America.

Currently, the largest ongoing data-model comparison is the Paleoclimate Model Intercomparison project (PMIP3), which is an effort to better constrain the paleoclimate models using paleoclimate data as validation or benchmark data sets (Joussaume and Taylor, 1999; Braconnot *et al.*, 2007; Braconnot *et al.*, 2012). PMIP3 is now focusing on data-model comparisons based on transient climate model simulations (Bartlein *et al.*, 2011; Braconnot *et al.*, 2012). Transient climate model simulations are different from traditional climate model simulations but more realistic in that the forcings such as CO_2 concentration are continuously varying rather than using a fixed or prescribed forcing where the model will equilibrate with the forcing (Neelin, 2011). PMIP3 also focuses on abrupt climate changes and model uncertainties in an effort to improve our understanding of Earth's past warm periods such as the Paleocene-Eocene thermal maximum (PETM) and Stage 5e interglacial (Paleoclimate Modelling Intercomparison Project Phase III, 2012).

Past Global Changes (PAGES) 2k is another recent large scale synthesis effort, which is comprised of a network of interdisciplinary groups from around the world that synthesize regional multiproxy paleoclimate data sets that span the past 2,000 years (PAGES 2k Consortium, 2013). The aim is to provide more comprehensive regional benchmark data sets to compare to the most recent global climate model simulations. This dissertation contributes to the overall objectives of the PAGES North American (NAM2K) group.

1.2.3 LGM to Mid-Holocene climate change

The LGM is a target period in the Quaternary that has been studied extensively. Paleoclimate simulations from Bartlein *et al.* (1998) are in agreement with Wright Jr. *et al.* (1993), both of which show the split jetstream with the stronger branch displaced south of the Laurentide ice sheet at 21 ka due to the height of the ice sheet, which blocked the atmospheric flow. Bartlein *et al.* (1998) provides evidence for a strong glacial anticyclone and weakening of the Aluetian low in the winter. As summer insolation increased, the Northeast Pacific and Bermuda high pressure systems strengthened, leading to warmer summer temperatures to the south of the rapidly melting ice sheet (Bartlein *et al.*, 1998).

On orbital timescales and following deglaciation, the early Holocene is generally accepted to have been warmer than today (COHMAP Members, 1988; Wright Jr. *et al.*, 1993). However, maximum warmth, often referred to as the Holocene thermal maximum or optimum, varied regionally and occurred at different times (Wanner *et al.*, 2008). For example, maximum warmth occurred in the early Holocene in western Canada, while much later, around 6,000 years ago in Northeastern Quebec due to the effects of the small remnants of the Laurentide ice sheet (Viau and Gajewski, 2009).

Following maximum insolation around 12,000 years BP (Berger and Loutre, 1991), insolation has been declining on millennial timescales, which caused the southward shift of the Inter-tropical convergence zone (ITCZ), weakening of the African-Asian monsoon and increased amplitude and frequency of El Nino Southern Oscillation (ENSO) up to the end of the last millennium (Wanner *et al.*, 2008). During the mid-Holocene, the amplitude of ENSO decreased due to orbital changes enhancing the seasonal cycle in the Northern Hemisphere, hence strengthening the summer monsoon and the Walker circulation, effectively dampening the development of El Nino events (Zheng *et al.*, 2008).

Continental July temperature reconstructions in North America for the time period of 6 ka were estimated from pollen using the modern analogue technique (MAT)(Sawada *et al.*, 2004). The reconstruction was generally in agreement with output of the Canadian AGCM2 climate model showing warmer and drier conditions across much of North America, except in the Southwestern USA and the Canadian Arctic Archipelago (Sawada *et al.*, 2004). During the summer Williams *et al.* (1999) demonstrate that the models simulated warmer and drier conditions over much of North America at 6 ka as compared to the data. The warm and dry summer conditions are likely due to the lack of surface moisture availability and suppressed vertical motions in the atmosphere as explored in Shinker *et al.* (2006).

1.2.4 Climate reconstructions of the last 2,000 years

The last 2,000 years is important for understanding climate on many different scales. On decadal to multi-century timescales the variability in climate is complex with much recent focus on the Medieval Warm Period (MWP) or Medieval Climate Anomaly (MCA) and the Little Ice Age (LIA) using high-resolution proxy records sometimes derived from mostly tree-ring records (Shi *et al.*, 2013; Christiansen and Ljungqvist, 2012; Christiansen and Ljungqvist, 2011; Wilson *et al.*, 2010; Ljungqvist, 2009; Jones *et al.*, 2009; Mann *et al.*, 2009a; Mann *et al.*, 2008; Mann *et al.*, 2007; Hegerl *et al.*, 2007; D'Arrigo *et al.*, 2006; Moberg *et al.*, 2005; Bradley *et al.*, 2005; Jones and Mann, 2004; Mann and Jones, 2003; Esper *et al.*, 2002; Briffa and Osborn, 2002; Briffa *et al.*, 2001; Crowley, 2000; Bradley *et al.*, 1999; Jones *et al.*, 1998; Mann *et al.*, 1998), those derived from pollen records (Gajewski, 1987; Gajewski, 1988; Gajewski, 1993; Gajewski *et al.*, 2007) and those derived from ice core records (Koerner and Fisher, 1990; Fisher *et al.*, 1995; Fisher *et al.*, 2012) and also pollen records from ice cores (Bourgeois *et al.*, 2000). In addition to Northern Hemisphere temperature, pan-Arctic Ocean sea ice cover has also been reconstructed for the last 1,450 years in Kinnard *et al.* (2011), showing that the current loss in Arctic sea ice cover is unprecedented compared to the reconstructed sea ice extent. These large scale synthesis studies find that medieval warmth is unmatched both in magnitude and spatial extent by the warmth of today (Hughes and Diaz, 1994; Crowley and Lowery, 2000; Diaz *et al.*, 2011). For the Little Ice Age (LIA), studies have suggested that reduced solar activity, strong tropical volcanic activity, and orbital cooling led to the LIA (Wanner *et al.*, 2008; IPCC, 2013). Another issue is that some hemispheric reconstructions, particularly those using mostly tree-ring records, tend to underestimate low frequency variability of climate (Christiansen, 2011; Shi *et al.*, 2012; Wang *et al.*, 2014). Furthermore, there is strong evidence for multi-century to millennial scale variability extending back in time with the Dark Ages cold period, Roman warm period and Bronze/Iron Age cooling, which is usually referred to as Neoglacial cooling (Viau *et al.*, 2002; Viau *et al.*, 2006). However, there still remains a gap in our understanding of these higher frequencies due to a lack of adequate regional coverage and syntheses (National Research Council, 2006; IPCC, 2013).

1.2.5 Hydroclimate reconstructions & spatial filtering

With temperature reconstructions in good agreement, hydro climate is now the focus of an international effort, especially with the Paleoclimate Model Intercomparison Project, phase 3 (PMIP3), an ongoing effort to improve our predictions of climate

change by comparing paleoclimate reconstructions to paleoclimate model simulations (Braconnot *et al.*, 2012). A warming climate is likely to produce much longer lasting and more severe droughts yet paradoxically more frequent and extreme precipitation events (Dai, 2011; IPCC, 2013). One cause of more frequent droughts is the reduced temperature gradient from the equator to poles causing a weakened jet stream that is wavier, resulting in more blocked weather patterns (Francis and Vavrus, 2012). Global temperatures have already warmed 0.8°C since 1850 and the projected increase in temperature by the end of this century will have drastic consequences on water supplies (IPCC, 2013). We may soon cross a tipping point with respect to large scale droughts due to climate warming (Lenton *et al.*, 2008; Trenberth *et al.*, 2014). There are often certain combinations of teleconnections such as a positive phase of the Arctic Oscillation (AO), La Nina and warm mode of Atlantic Multidecadal Oscillation (AMO) and Pacific Decadal Oscillation (PDO) that can enhance these droughts (Dai, 2011; Wahl *et al.*, 2012; McCabe *et al.*, 2004). The warm mode of the AMO was a factor in the dustbowl of the 1930s in the Great Plains of the USA causing an eastward shift and weakening of the Bermuda high (Nigam *et al.*, 2011). Further back in time, there is evidence of mega-droughts affecting parts of North America during warm periods such as the MWP (MacDonald, 2007; Herweijer *et al.*, 2007). The MWP may have had a La Nina like state with a positive NAO (Trouet *et al.*, 2009). This has implications for future climate change as a warming climate has the potential to turn parts of the Southern USA and other areas of the world such as the Mediterranean and Sahel into a permanent dust-bowl due to the expanded subtropical highs and lack of precipitation (Seager and Vecchi, 2010; IPCC, 2013).

Chapter 4 presents a hydroclimatic reconstruction based on pollen (moderate-resolution) using a composite regional approach which has been successful in reconstructing mean temperature of the warmest month (Viau *et al.*, 2012). As hydro climate is more variable than temperature, this chapter outlines a unique method to improve a reconstruction by using a spatial filtering technique (further discussed in section 1.4.4). Tobler's first law of geography (Miller, 2004) applies precisely to paleoclimate reconstructions since closer sites are more similar climatically than distant sites, which is the essence of spatial autocorrelation. A site with a positive anomaly would be positively (negatively) correlated with a nearby site that has a positive (negative) anomaly. We can examine global or local spatial autocorrelation in this context. Large scale paleoclimate reconstructions from pollen databases are derived from a large number of sites. This large number of sites helps reduce the signal to noise ratio (Gajewski, 1993) but also produces a heterogeneous reconstruction, which is basically a checkerboard pattern. It is often the case in a reconstruction that there

may be several sites at a given time period showing opposing trends compared to the rest of the cluster of sites in that region. These sites with opposing trends are reflecting climate variability at a different scale then the large scale trends. For example, the large scale trend for a specific region at a specific time interval may be for warmer and drier conditions whereas a few sites in that region may show cooler and wetter conditions, which is the opposite of what should be expected. Of course there may even be a few sites showing cooler and drier or warmer and wetter conditions. The purpose of spatial filtering is to only include sites that show the large scale climate variability and to filter out the other sites. A similar method of spatial filtering using local indicators of spatial association (LISA) was developed by Viau and Gajewski (2001) and applied to mapping changes in lake levels during the last 12,000 years at 3,000 year intervals. The approach of Viau and Gajewski (2001) allowed for a separation between large scale patterns and fine-scale variability to better understand past climate variability. This is the rationale for using methods of local indicators of spatial association (Isaaks and Srivastava, 1989; Anselin, 1995) to improve paleoclimate reconstructions.

1.3 Research objectives

The objectives of this dissertation are as follows:

1. Develop a new North American continental-scale mean July temperature (TJUL) reconstruction for the past 2,000 years to explore multi-centennial scale climate variability relative to modern conditions for several ecoregions of North America.

2. Evaluate the impacts of the use of different modern TJUL climate data sets (both station-based and reanalysis products) on the pollen-based North American continental-scale reconstruction of the past 2,000 years.

3. Develop a new North American continental-scale total annual precipitation (ANNP) reconstruction for the past 2,000 years to explore multi-centennial scale climate variability relative to modern conditions for several ecoregions of North America.

4. Develop an easy to use R program that allows for user-defined pollen-based paleoclimate reconstructions.

1.4 Study area, data & methods

1.4.1 Study area

The study area is North America, excluding Mexico where the data is sparse. The diversity of climate zones and topographical features across North America provides a useful study area to examine the impacts of using different modern data sets when assessing climate change at the regional and continental scales over the past 2,000 years.

1.4.2 Pollen data

We use the newest version of the North American Pollen Database (NAPD) (Grimm, 2011) which is derived from NEOTOMA (Grimm, 2008). The extracted fossil pollen data includes 748 sites (Figure 2B) and 135 pollen taxa. Note that the 135 pollen sum is identical to the 134 sum described in Whitmore *et al.* (2005) except that pines in the southeast USA are differentiated from pines elsewhere (Williams and Shuman, 2008). The latest version of the NAPD (Grimm, 2011) is improved over the previous version of NAPD. There are more available fossil sites and the IntCal '04 (Reimer *et al.*, 2004) calibration curve is used instead of IntCal '98 (Stuiver *et al.*, 1998) for the calibration of radiocarbon dates of the pollen at each site. The modern pollen data is from the Whitmore database (Whitmore *et al.*, 2005). There are ∼4,800 sites in version 1.73 of the database, which is an update from version 1.72 where the seasonal precipitation totals (PDJF, PMAM, PJJA, PSON) at each site were corrected by the author.

We use pollen to reconstruct past climates in North America, due to its extensive spatial coverage and availability. Vegetation has a strong relationship with modern climate conditions at both the regional and continental scales hence pollen is well suited to reconstruct past climates (Webb III, 1986; COHMAP Members, 1988; Gajewski, 1983; Gajewski, 1987; Gajewski, 1988; Gajewski, 1993; Gajewski, 2006; Viau *et al.*, 2006; Gajewski *et al.*, 2007; Gajewski and Viau, 2011). In addition, pollen studies have shown a rapid response of vegetation to climate (Gajewski, 1983; Webb III, 1986; Gajewski, 1987; Peros *et al.*, 2008) and synchronous changes to vegetation as a result of abrupt climate changes during glacial times (Williams, 2003; Grimm *et al.*, 1993; Viau *et al.*, 2002) and during the last millenium (Gajewski, 1987). In the modern period, it is well known that the tundra-forest boundary coincides with the mean July position of the Arctic Front (Bryson, 1966) and the mean July position of the Arctic front has been shown to be related to the productivity of

the vegetation (Ladd and Gajewski, 2010).

1.4.3 Climate data

Climate station data sets represent the climate variability at specific locations as recorded from weather observations over a long period of time. Climate data can also be physically-modeled from stations and then interpolated to a regularly spaced grid (e.g. WorldClim in Figure 5C). Some of the common data sets are Global Historical Climate Network (GHCN), Berkeley Earth Surface Temperature (BEST), Climate Research Unit (CRU) and others, which are described in University Corporation for Atmospheric Research (2013). The modern climate data in Whitmore *et al.* (2005) is based on New *et al.* (2002) with corrections for elevation due to lapse rate as described in section 2.2 of Whitmore *et al.* (2005). New *et al.* (2002) is a 10 arc-minute data set (∼18 km) based on the CRU gridded climate data set. In order to calculate local lapse rates at each pollen site, locally weighted third-degree trend-surface regression models or splines (latitude, longitude and elevation) are fitted to the climate data surrounding the site within a given search radius, then the lapse rates at these surrounding data points are used to adjust the climate values to the elevation of the site and finally, the adjusted climate values are interpolated at the location of the site (Whitmore *et al.*, 2005).

Reanalyses are essentially data products derived from historical climate observations that are reprocessed by a numerical weather model that is frozen over the reanalysis period. There are three types of reanalysis climate variables: (1) those that are derived mostly from observations; (2) those which are partly derived from observations and partly derived from the model and finally (3) some variables that are derived mostly from the model. For this reason it is important to be aware of the regional distribution of assimilated inputs when performing analyses of climate trends over the modern period as the data sources change through time and this can cause spurious results (Trenberth *et al.*, 2008). This factor is particularly relevant over the satellite era (i.e. 1979 to present) when for example, satellite data is incorporated into most of the reanalysis products but only after 1979. This is why the ERA-Interim reanalysis is often used as it starts in 1979. In addition there are new reanalyses at very high resolution (e.g. less than 0.5°) where individual files are terabytes in size, therefore data management becomes a very important issue. Some of the newer reanalyses are the NCEP Climate Forecast System Reanalysis (CFSR), ERA 20th Century Reanalysis (ERA20c) and the Modern Era Reanalysis (MERRA). A complete description of these products can be found in University Corporation for Atmospheric Research (2013).

1.4.4 Methods

Reconstruction of July temperature for the past 2,000 years (Chapter 2)

The open source R code MATTOOLS (Sawada, 2006) along with a customized R program developed by the author (Appendix B and C) was used to reconstruct the mean July temperature from the fossil pollen series, derived from the North American Pollen Database (Grimm, 2011). Details of the original methodology are found in Appendix D.1. Essentially, this involves the following steps, which have been automated in a customized R program. First, there is the setup of modern and fossil pollen data files, then the correct MATTOOLS code is generated and run in R. Next is the linear interpolation through time of the reconstruction data and finally a spatial join of reconstruction to modern sites in order to calculate the anomalies (anomaly = reconstructed value - modern value). The reconstruction is then split by eco-regions, mapped out by time slice of data (e.g. MWP, LIA) compared to modern. In addition we can produce squared-chord distance (SCD) plots by ecoregion, plot the number of sites, samples and dates through time, calculate resolution of pollen sites and calculate uncertainty envelopes for the reconstructions.

The use of different modern July temperature data sets for a 2,000 year reconstruction (Chapter 3)

Chapter 3 examines the impacts of using different modern TJUL data sets on the July temperature reconstruction of Chapter 2. This required the development of a customized R program (Appendix B and C) to automate the reconstructions using multiple modern calibration data sets. The application uses the same modern pollen site distribution as Whitmore v1.7.3 (Whitmore *et al.*, 2005) database, but simply assigns different modern TJUL values based on each individual modern calibration data set to compute TJUL anomalies through time. Here we use three different groups of modern calibration data sets; (A) station-based data, (B) physically modeled from observation data and (C) reanalysis products. Two different approaches are used to assign modern TJUL values to each modern pollen site; the TJUL value from the closest climate station or the JUL value from the nearest climate data grid point.

Spatial filtering using Local Indicators of Spatial Association (LISA) for a 2,000 year reconstruction of total annual precipitation (Chapter 4)

The reconstructions followed the same metholodgy as in Appendix D.1 but using the R program (Appendix B and C). Unlike temperature, spatial variations in moisture

do not necessarily follow latitudinal gradients. Spatial variations of hydroclimate are much more complex and regionally diverse compared to spatial variations of temperature. Thus we expect the spatial filtering to produce maps that are slightly less smooth as compared with filtering a reconstruction of mean temperature of the warmest month (MTWA). Here we apply spatial filtering using LISA (Anselin, 1995) to the reconstruction in order to better define the spatial patterns of moisture variability/availability through space and time.

Local Moran's I is a test statistic for spatial autocorrelation between sites, essentially an indicator of spatial continuity. Reconstructed sites should not violate the assumption of spatial continuity as we expect closer sites to be more similar than sites further away. Local Moran's I can also determine the presence of outliers at a given location or in other words, extreme values that should be removed since they could cause edge effects or artefacts when interpolating. The three measures of spatial autocorrelation most often used are: Moran's I, Geary's C and Getis Ord (Gi*) statistics. We used Local Moran's I here rather than Local Geary's C. Sokal (1979) shows that Moran's I and Geary's C produce similar interpretations. The Getis Ord (Gi*) statistic simply measures the magnitude of the clustering of values.

The procedure is to examine the Moran's I and Moran's I z-values between anomalies of sites surrounding each site of interest in the reconstruction to exclude sites that are negatively autocorrelated. These would be sites with negative Moran's I and negative Moran's I z-value or mixed signs (very rare) as compared with the site of interest. Sites with a Moran's I and Moran's I z-value of zero are outside of the current lag distance and are retained until filtering is completed on sites in the largest (last) lag distance. At a given lag distance they are retained because they may be important at the next (larger) lag distance. At each lag, sites with positive Moran's I and Moran's I z-values are always retained because these sites are positively spatially auto-correlated with surrounding sites. The filtering is completed at 50 km lag intervals with a lag of 0-50 km, then 0-100 km, 0-150 km, 0-200 km until the last lag of 250 km is reached. Sites with zeroes are retained after filtering at the last lag distance. Whether zeroes are retained or removed after the last lag distance does not make a significant difference in the overall patterns. Also, using a single lag of 0-250 km versus the cumulative lag described above changes the number of sites retained with only a slight difference in the overall patterns. At progressively larger lag distances, a larger number of sites are removed from the reconstruction. The significance (p-value) of the Local Moran's I statistic was examined but was not found to be important in the filtering process. We tried filtering at lag distances larger than 250 km (up to 500 km); however, a larger percentage of sites were removed causing

excessive smoothing of the resulting interpolations.

1.5 Thesis organization

The three main chapters of the thesis are written as individual manuscripts for peer-reviewed journals. I am the primary author on the papers in chapters three and four. I am the secondary author on the paper in chapter two with my supervisor the primary author. For the published paper (Chapter 2), it incorporates suggestions from the reviewers. The order of authorship is based on the relative contribution to the intelllectual property of each paper. All analyses, including data analyses, GIS analysis, data modelling, development of scripts and programs, production of figures and tables and writing of the thesis were completed by me with the assistance and editing of my supervisor on a regular basis throughout the dissertation process.

Chapter Two: The climate of North America during the past 2,000 years reconstructed from pollen data

Viau, A.E., Ladd, M., Gajewski, K (2012). The climate of North America during the past 2,000 years using pollen data. *Global Planetary Change*. **84-85**, 75-83, DOI 10.1016/j.gloplacha.2011.09.010.

For this chapter, I was second author on this paper which was published in *Global and Planetary Change* (Viau, Ladd, and Gajewski, 2012). I was involved in all aspects of this paper from conceptualization to writing of the paper. My lead role was to develop the reconstruction methodology and analyses of the data along with interpretation, writing and editing the manuscript.

Chapter Three: The impact of using different modern climate data sets in pollen-based paleoclimate reconstructions of North America

For this chapter, I was first author on this paper which will be submitted to *Quaternary Science Reviews*. Additional assistance and input were provided by Robert Way and Andre Viau. All ideas, analysis, programming and manuscript writing were completed by me with input, edits and collaboration with my supervisor and co-authors.

Chapter Four: Variations in precipitation over North America over the past 2,000 years

For this chapter, I was first author on this paper which will be submitted to *Journal of Climate*. The spatial filtering methodology was developed in consultation with Michael Sawada and Andre Viau. Additional assistance and input were provided by Robert Way and Konrad Gajewski. All ideas, analysis, programming and manuscript writing were completed by me with input, edits and collaboration with my supervisor and co-authors.

References

Ammann, C. M., F. Joos, D.S. Schimel, B.L. Otto-Bliesner, and R.A. Tomas (2007). Solar influence on climate during the past millennium: Results from transient simulations with the NCAR Climate System Model. *PNAS* **104**(10), 3713–3718.

Anselin, L. (1995). Local Indicators of Spatial Association - LISA. *Geographical Analysis* **27**(2), 93–115.

Bartlein, P.J., K.H. Anderson, P.M. Anderson, M.E. Edwards, C.J. Mock, R.S. Thompson, R.S. Webb, T. Webb III, and C. Whitlock (1998). Paleoclimate simulations for North America over the past 21,000 years: features of the simulated climate and comparisons with paleoenvironmental data. *Quaternary Science Reviews* **17**(6-7), 549–586.

Bartlein, P.J., S.J. Harrison, S. Brewer, S. Connor, B.A.S. Davis, K. Gajewski, J. Guiot, T.I. Harrison-Prentice, A. Henderson, O. Peyron, I.C. Prentice, M. Scholze, H. Seppä, B. Shuman, S. Sugita, R.S. Thompson, A.E. Viau, J. Williams, and H. Wu (2011). Pollen-based continental climate reconstructions at 6 and 21 ka: a global synthesis. *Climate Dynamics* **37**(3-4), 775–802.

Berger, A. and M. F. Loutre (1991). Insolation values for the climate of the last 10 million years. *Quaternary Science Reviews* **10**(4), 297–317.

Bourgeois, J.C., R.M. Koerner, K. Gajewski, and D.A. Fisher (2000). A Holocene ice-core pollen record from Ellesmere Island, Nunavut, Canada. *Quaternary Research* **54**(2), 275–283.

Braconnot, P., B. Otto-Bliesner, S. Harrison, S. Joussaume, J.-Y. Peterchmitt, A. Abe-Ouchi, M. Crucifix, E. Driesschaert, Th. Fichefet, C.D. Hewitt, M. Kageyama, A. Kitoh, A. Lainé, M.-F. Loutre, O. Marti, U. Merkel, G. Ramstein, P. Valdes, S. L. Weber, Y. Yu, and Y. Zhao (2007). Results of PMIP2 coupled simulations of the Mid-Holocene and Last Glacial Maximum–Part 1: experiments and large-scale features. *Climate of the Past* **3**(2), 261–277.

Braconnot, P., S.P. Harrison, M. Kageyama, P.J. Bartlein, V. Masson-Delmotte, A. Abe-Ouchi, B. Otto-Bliesner, and Y. Zhao (2012). Evaluation of climate models using palaeoclimatic data. *Nature Climate Change* **2**(6), 417–424.

Bradley, R.S., M.E. Mann, and M.K. Hughes (1999). Northern hemisphere temperatures during the past millennium: inferences, uncertainties, and limitations. *Geophysical Research Letters* **26**(6), 759–762.

Bradley, R.S., M. Vuille, M. Werner, R.Y. Chan, T.J. Osborne, K.R. Briffa, M.K. Hughes, and P.D. Jones (2005). Proxy-based Northern Hemisphere surface temperature reconstructions: Sensitivity to method, predictor network, target season, and target domain. *Journal of Climate* **18**(13), 2308–2329.

Briffa, K.R. and T.J. Osborn (2002). Blowing hot and cold. *Science* **295**(5563), 2227–2228.

Briffa, K.R., T.J. Osborn, F.H. Schweingruber, I.C. Harris, P.D. Jones, S.G. Shiyatov, and E.A. Vaganov (2001). Low-frequency temperature variations from a northern tree ring density network. *Journal of Geophysical Research* **106**(D3), 2929–2941.

Bryson, R.A. (1966). Air masses, streamlines, and the boreal forest. *Geographical Review* **8**(3), 228–269.

Christiansen, B. (2011). Reconstructing the NH Mean Temperature: Can Underestimation of Trends and Variability Be Avoided? *Journal of Climate* **24**(3), 674–692.

Christiansen, B. and F. C. Ljungqvist (2011). Reconstruction of the extratropical NH mean temperature over the last millennium with a method that preserves low-frequency variability. *Journal of Climate* **24**(3), 6013–6034.

— (2012). The extra-tropical Northern Hemisphere temperature in the last two millennia: reconstructions of low-frequency variability. *Climate of the Past* **8**(2), 765–786.

Clark, P.U., A.J. Weaver (coordinating lead authors), E. Brook, E.R. Cook, T.L. Delworth, and K. Steffen (chapter lead authors) (2008). *Abrupt Climate Change. A report by the U.S. Climate Change Science Program and the Subcommittee on Global Change Research*. research report. Reston, VA, USA: U.S. Geological Survey, p. 244.

CLIMAP Project Members (1981). *Seasonal reconstructions of the Earth's surface at the last glacial maximum*. Geological Society of America Map and Chart Series MC-36.

COHMAP Members (1988). Climatic changes of the last 18,000 years: Observations and model simulations. *Science* **241**, 1043–1052.

Crowley, T. J. (2000). Causes of climate change over the past 1000 years. *Science* **289**(5477), 270–277.

Crowley, T. J. and T. Lowery (2000). How Warm Was the Medieval Warm Period? *Ambio* **29**(1), 51–54.

Dai, A. (2011). Drought under global warming: a review. *Wiley Interdisciplinary Reviews: Climate Change* **2**(1), 45–65.

D'Arrigo, R., R. Wilson, and G. Jacoby (2006). On the long-term context for late twentieth century warming. *Journal of Geophysical Reseach* **111**(D3).

Diaz, H.F., R. Trigo, M.K. Hughes, M.E. Mann, E. Xoplaki, and D. Barriopedro (2011). Spatial and temporal characteristics of climate in Medieval times revisited. *Bulletin of the American Meteorological Society* **92**(11), 1487–1500.

Edwards, T.L., M. Crucifix, and S.P. Harrison (2007). Using the past to constrain the future: How the paleorecord can improve estimates of global warming. *Progress in Physical Geography* **31**(5), 481–500.

Esper, J., E.R. Cook, and F.H. Schweingruber (2002). Low-frequency signals in long tree-ring chronologies for reconstructing past temperature variability. *Science* **295**(5563), 2250–2253.

Fisher, D.A., R.M. Koerner, and N. Reeh (1995). Holocene climatic records from Agassiz ice cap, Ellesmere Island, NWT, Canada. *The Holocene* **5**(1), 19–24.

Fisher, D.A., J. Zheng, D. Burgess, C. Zdanowicz, C. Kinnard, M. Sharp, and J. Bourgeois (2012). Recent melt rates of Canadian arctic ice caps are the highest in four millennia. *Global and Planetary Change* **84**, 3–7.

Francis, J.A. and S.J. Vavrus (2012). Evidence linking Arctic amplification to extreme weather in mid-latitudes. *Geophysical Research Letters* **39**(6).

Gajewski, K. (1983). On the interpretation of climatic change from the fossil record: Climatic change in central and eastern United States for the past 2000 years estimated from pollen data. PhD thesis. University of Wisconsin-Madison.

— (1987). Climatic impacts on the vegetation of eastern North America during the past 2000 years. *Vegetatio* **68**(3), 179–190.

— (1988). Late Holocene climate changes in eastern North America estimated from pollen data. *Quaternary Research* **29**(3), 255–262.

— (1993). The role of paleoecology in the study of global climatic change. *Review of palaeobotany and palynology* **79**(1), 141–151.

— (2006). Essai: Is Arctic Palynology a "Blunt Instrument"? *Géographie physique et Quaternaire* **60**(2), 95–102.

Gajewski, K. and A.E. Viau (2011). Abrupt climate changes during the Holocene across North America from pollen and paleolimnological records. *Abrupt Climate Change: Mechanisms, Patterns, and Impacts, Geophysical Monograph Series*. Ed. by L. Polyak H. Rashid and E. Mosley-Thompson. Washington, D.C., USA: AGU, 161–171.

Gajewski, K., A.E. Viau, and M.C. Sawada (2007). Millennial-scale climate variations in the Holocene - the terrestrial record. *Climate Variability and Change: Past, Present and Future*. Ed. by G. Kutzbach. Madison, Wisconsin, USA: University of Wisconsin, 133–154.

Grimm, E.C. (2008). *Neotoma database*. url: http://www.neotomadb.org.

— (2011). *North American Pollen database. Data archived at IGBP PAGES/World Data Center–A for Paleoclimatology, NOAA/NGDC Paleoclimatology Program, Boulder, CO, USA*. url: http://www.ncdc.noaa.gov/paleo.

Grimm, E.C., G.L. Jacobson Jr, W.A. Watts, B.C.S. Hansen, and K.A. Maasch (1993). A 50,000-Year Record of Climate Oscillations from Florida and its Temporal Correlation. *Science* **61**, 9.

Hansen, J., M. Sato, P. Kharecha, D. Beerling, R. Berner, V. Masson-Delmotte, M. Pagani, M. Raymo, D.L. Royer, and J.C. Zachos (2008). Target atmospheric CO2: Where should humanity aim? *arXiv preprint arXiv:0804.1126*.

Harrison, S.P., D. Jolly, F. Laarif, A. Abe-Ouchi, B. Dong, K. Herterich, C. Hewitt, S. Joussaume, J.E. Kutzbach, J. Mitchell, N. de Noblet, and Valdes P. (1998). Intercomparison of simulated global vegetation distributions in response to 6 kyr BP orbital forcing. *Journal of Climate* **11**(11), 2721–2742.

Hegerl, G.C., T.J. Crowley, M. Allen, W.T. Hyde, H.N. Pollack, J. Smerdon, and E. Zorita (2007). Detection of human influence on a new, validated 1500-year temperature reconstruction. *Journal of Climate* **20**(4), 650–666.

Herweijer, C., R. Seager, E.R. Cook, and J. Emile-Geay (2007). North American droughts of the last millennium from a gridded network of tree-ring data. *Journal of Climate* **20**(7), 1353–1376.

Hughes, M.K. and H.F. Diaz (1994). Was there a "Medieval Warm Period", and if so, where and when? *Climatic Change* **26**(2–3), 109–142.

IPCC (2007). *Climate Change 2007, The Physical Science Basis. Fourth Assessment Report*. Ed. by S. Solomon, D. Manning, Z. Chen, M. Marquis, K.B. Averyt, M. Tignor, and H. Miller. Cambridge, UK and New York, NY, USA: Cambridge University Press, p. 996.

— (2013). *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by T.F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P.M. Midgley. Cambridge, UK and New York, NY, USA: Cambridge University Press, p. 1552.

Isaaks, E.H. and R.H. Srivastava (1989). *An Introduction to Applied Geostatistics*. New York, USA: Oxford University Press.

Jackson, S.T. and J.W. Williams (2004). Modern analogs in Quaternary paleoecology: here today, gone yesterday, gone tomorrow? *Annual Review of Earth and Planetary Sciences* **32**, 495–537.

Jones, P.D. and M.E. Mann (2004). Climate over past millennia. *Reviews of Geophysics* **42**(2), RG2002.

Jones, P.D., K.R. Briffa, T.P. Barnett, and S.F.B. Tett (1998). High-resolution palaeoclimatic records for the last millennium: interpretation, integration and comparison with General Circulation Model control-run temperatures. *The Holocene* **8**(4), 455–471.

Jones, P.D., K.R. Briffa, T.J. Osborn, J.M. Lough, T.D. Van Ommen, B.M. Vinther, J. Luterbacher, E.R. Wahl, F.W. Zwiers, M.E. Mann, C.M. Schmidt G.A.and Ammann, B.M. Buckley, K.M. Cobb, J. Esper, H. Goosse, N. Graham, E. Jansen, T. Kiefer, C. Kull, M. Kttel, E. Mosley-Thompson, J.T Overpeck, N. Riedwyl, A.W. Schulz M. Tudhope, R. Villalba, H. Wanner, E. Wolff, and E. Xoplaki (2009). High-resolution palaeoclimatology of the last millennium: a review of current status and future prospects. *The Holocene* **19**(1), 3–49.

Joussaume, S. and K. E. Taylor (1999). The Paleoclimate Modeling Intercomparison Project, in Proceedings of the Third PMIP Workshop, Canada. WCRP-111, WMO/TD-1007. Available at http://www-lsce.cea.fr/pmip/. Geneva, Switzerland: World Meteorological Organization, 9–24.

Kinnard, C., C.M. Zdanowicz, D.A. Fisher, E. Isaksson, A. de Vernal, and L.G. Thompson (2011). Reconstructed changes in Arctic sea ice over the past 1,450 years. *Nature* **479**(7374), 509–512.

Koerner, R.M. and D.A. Fisher (1990). A record of Holocene summer climate from a Canadian high-Arctic ice core. *Nature* **343**, 630–631.

Ladd, M.J. and K. Gajewski (2010). The North American summer Arctic front during 1948-2007. *International Journal of Climatology* **30**(6), 874–883.

Lenton, T.M, H. Held, E. Kriegler, J.W. Hall, W. Lucht, S. Rahmstorf, and H.J. Schellnhuber (2008). Tipping elements in the Earth's climate system. *Proceedings of the National Academy of Sciences* **105**(6), 1786–1793.

Ljungqvist, F.C. (2009). Temperature proxy records covering the last two millennia: a tabular and visual overview. *Geografiska Annaler: Series A, Physical Geography* **91**(1), 11–29.

MacDonald, G.M. (2007). Severe and sustained drought in southern California and the West: Present conditions and insights from the past on causes and impacts. *Quaternary International* **173**, 87–100.

Mann, M.E and P.D. Jones (2003). Global surface temperatures over the past two millennia. *Geophysical Research Letters* **30**(15).

Mann, M.E., R.S. Bradley, and M.K. Hughes (1998). Global-scale temperature patterns and climate forcing over the past six centuries. *Nature* **392**(6678), 779–787.

Mann, M.E., S. Rutherford, E.R. Wahl, and C. Ammann (2007). Robustness of proxy-based climate field reconstruction methods. *Journal of Geophysical Research* **112**(D12), D12109.

Mann, M.E, M.K. Zhang Z.and Hughes, R.S Bradley, S.K. Miller, S. Rutherford, and F. Ni (2008). Proxy-based reconstructions of hemispheric and global surface temperature variations over the past two millennia. *Proceedings of the National Academy of Sciences* **105**(36), 13252–13257.

Mann, M.E, Z. Zhang, S. Rutherford, R.S. Bradley, M.K. Hughes, D. Shindell, C. Ammann, G. Faluvegi, and F. Ni (2009a). Global signatures and dynamical origins of the Little Ice Age and Medieval Climate Anomaly. *Science* **326**(5957), 1256–1260.

McCabe, G.J., M.A. Palecki, and J.L. Betancourt (2004). Pacific and Atlantic Ocean influences on multidecadal drought frequency in the United States. *Proceedings of the National Academy of Sciences* **101**(12), 4136–4141.

Miller, H.J. (2004). Tobler's First Law and Spatial Analysis. *Annals of the Association of American Geographers* **94**(2), 284–289.

Moberg, A., D.M. Sonechkin, K. Holmgren, N.M. Datsenko, and W. Karlén (2005). Highly variable Northern Hemisphere temperatures reconstructed from low-and high-resolution proxy data. *Nature* **433**(7026), 613–617.

National Research Council (2006). *Surface Temperature Reconstructions for the Last 2,000 Years*. Washington, D.C., USA: National Academies Press.

Neelin, J. D. (2011). *Climate Change and Climate Modeling*. Cambridge. New York, p. 282.

New, M., D. Lister, M. Hulme, and I. Makin (2002). A high-resolution data set of surface climate over global land areas. *Climate research* **21**(1), 1–25.

Nigam, S., B. Guan, and A. Ruiz-Barradas (2011). Key role of the Atlantic Multidecadal Oscillation in 20th century drought and wet periods over the Great Plains. *Geophysical Research Letters* **38**(16).

PAGES 2k Consortium (2013). Continental-scale temperature variability during the past two millennia. *Nature Geoscience* **6**, 339–346.

Paleoclimate Modelling Intercomparison Project Phase III (2012). *PMIP 3 Project Home Page*. url: http://pmip3.lsce.ipsl.fr.

Peixoto, J.P. and A.H. Oort (1992). *Physics of Climate*. New York, USA: American Institute of Physics.

Peros, M.C., K. Gajewski, and A.E. Viau (2008). Continental-scale tree population response to rapid climate change, competition and disturbance. *Global Ecology and Biogeography* **17**(5), 658–669.

Pinot, S., G. Ramstein, S.P. Harrison, I.C. Prentice, J. Guiot, M. Stute, and S. Joussaume (1999). Tropical paleoclimates at the Last Glacial Maximum: comparison of Paleoclimate Modeling Intercomparison Project (PMIP) simulations and paleodata. *Climate Dynamics* **15**(11), 857–874.

Reimer, P.J., M.G.L. Baillie, E. Bard, A. Bayliss, J.W. Beck, C.J.H. Bertrand, P.G. Blackwell, C.E. Buck, G.S. Burr, K.B. Cutler, P.E. Damon, R.L. Edwards, R.G. Fairbanks, M. Friedrich, T.P. Guilderson, A.G. Hogg, K.A. Hughen, B. Kromer, G. McCormac, S. Manning, C.B. Ramsey, R.W. Reimer, S. Remmele, J.R. Southon, M. Stuiver, S. Talamo, F.W. Taylor, J. van der Plicht, and C.E. Weyhenmeyer (2004). IntCal04 terrestrial radiocarbon age calibration, 26e0 ka BP. *Radiocarbon* **46**.

Sawada, M., A.E. Viau, G. Vettoretti, W.R. Peltier, and K. Gajewski (2004). Comparison of North-American pollen-based temperature and global lake-status with CCCma AGCM2 output at 6ka. *Quaternary Science Reviews* **23**(3), 225–244.

Sawada, M.C. (2006). An open source implementation of the Modern Analog Technique (MAT) within the R computing environment. *Computers and Geosciences* **32**(6), 818–833.

Sawada, M.C., K. Gajewski, A. de Vernal, and P. Richard (1999). Comparison of marine and terrestrial Holocene climatic reconstructions from northeastern North America. *The Holocene* **9**(3), 267–277.

Seager, R. and G.A. Vecchi (2010). Greenhouse warming and the 21st century hydroclimate of southwestern North America. *Proceedings of the National Academy of Sciences* **107**(50), 21277–21282.

Shi, F., B. Yang, F.C. Ljungqvist, and F. Yang (2012). Multi-proxy reconstruction of Arctic summer temperatures over the past 1400 years. *Climate Research* **54**, 113–128.

Shi, F., B. Yang, A. Mairesse, L. von Gunten, J. Li, A. Bräuning, F.i Yang, and X. Xiao (2013). Northern Hemisphere temperature reconstruction during the last millennium using multiple annual proxies. *Climate Research* **56**, 231–244.

Shinker, J.J., P.J. Bartlein, and B. Shuman (2006). Synoptic and dynamic climate controls of North American mid-continental aridity. *Quaternary Science Reviews* **25**(13), 1401–1417.

Sokal, R.R. (1979). Ecological parameters inferred from spatial correlograms. *Contemporary quantitative ecology and related ecometrics*, 167–196.

Stuiver, M., P. J. Reimer, E. Bard, J. W. Beck, G. S. Burr, K. A. Hughen, B. Kromer, G. McCormac, J. van der Plicht, and M. Spurk (1998). IntCal98 radiocarbon age calibration. *Radiocarbon* **24**(3), 1041.

Trenberth, K.E., T. Koike, and K. Onogi (2008). Progress and prospects for reanalysis for weather and climate. *Eos, Transactions American Geophysical Union* **89**(26), 234–235.

Trenberth, K.E., A. Dai, G. van der Schrier, P.D. Jones, J. Barichivich, K.R. Briffa, and J. Sheffield (2014). Global warming and changes in drought. *Nature Climate Change* **4**(1), 17–22.

Trouet, V., J. Esper, N.E. Graham, A. Baker, J.D. Scourse, and D.C. Frank (2009). Persistent positive North Atlantic Oscillation mode dominated the medieval climate anomaly. *Science* **324**(5923), 78–80.

University Corporation for Atmospheric Research (2013). *Climate Data Guide from NCAR's Climate Analysis Section — Climate Data Strengths, Limitations, and Applications — home*. url: http://climatedataguide.ucar.edu.

Viau, A., M. Ladd, and K. Gajewski (2012). The climate of North America during the past 2000 years reconstructed from pollen data. *Global and Planetary Change* **84**, 75–83.

Viau, A.E. and K. Gajewski (2001). Holocene variations in the global hydrological cycle quantified by objective gridding of lake level databases. *Journal of Geophysical Research: Atmospheres (1984–2012)* **106**(D23), 31703–31716.

— (2009). Reconstructing millennial-scale, regional paleoclimates of boreal Canada during the Holocene. *Journal of Climate* **22**(2), 316–330.

Viau, A.E., K. Gajewski, P. Fines, D.E. Atkinson, and M.C. Sawada (2002). Widespread evidence of 1500 yr climate variability in North America during the past 14 000 yr. *Geology* **30**(5), 455–458.

Viau, A.E., K. Gajewski, M.C. Sawada, and P. Fines (2006). Millennial-scale temperature variations in North America during the Holocene. *Journal of Geophysical Research: Atmospheres (1984-2012)* **111**(D9).

Wahl, E.R., H.F. Diaz, and C. Ohlwein (2012). A pollen-based reconstruction of summer temperature in central North America and implications for circulation patterns during medieval times. *Global and Planetary Change* **84**, 66–74.

Wang, J., J. Emile-Geay, J. E. Smerdon, D. Guillot, and B. Rajaratnam (2014). Evaluating climate field reconstruction techniques using improved emulations of real-world conditions. *Climate of the Past* **10**(1), 1–19.

Wanner, H., J. Beer, J. Bütikofer, T.J. Crowley, U. Cubasch, J. Flückiger, H. Goosse, M. Grosjean, F. Joos, J.O. Kaplan, M. Küttel, S.A. Müller, I.C. Prentice, O. Solomina, T.F. Sticker, P. Trasov, M. Wagner, and M. Widmann (2008). Mid-to Late Holocene climate change: an overview. *Quaternary Science Reviews* **27**(19), 1791–1828.

Webb III, T. (1986). Is vegetation in equilibrium with climate? How to interpret late-Quaternary pollen data. *Vegetatio* **67**(2), 75–91.

Whitmore, J., K. Gajewski, M.C. Sawada, J.W. Williams, B. Shuman, P.J. Bartlein, T. Minckley, A.E. Viau, T. Webb III, S. Shafer, P. Anderson, and L. Brubaker (2005). Modern pollen data from North America and Greenland for multi-scale paleoenvironmental applications. *Quaternary Science Reviews* **24**(16), 1828–1848.

Williams, J. W., P. J. Bartlein, and T. Webb III (1999). Data-model comparisons for eastern North America - inferred biomes and climate values from pollen data. Proceedings of the Third Paleoclimate Modeling Intercomparison Project Workshop, Oct. 4-8, 1999. Montreal, Canada. Ed. by P. Braconnot. WCRP-111, WMO/TD-No. 1007. 4–8.

Williams, J.W. (2003). Variations in tree cover in North America since the last glacial maximum. *Global and Planetary Change* **35**(1), 1–23.

Williams, J.W. and B. Shuman (2008). Obtaining accurate and precise environmental reconstructions from the modern analog technique and North American surface pollen dataset. *Quaternary Science Reviews* **27**(7), 669–687.

Wilson, R., E. Cook, R. D'Arrigo, N. Riedwyl, M.N. Evans, A. Tudhope, and R. Allan (2010). Reconstructing ENSO: the influence of method, proxy data, climate forcing and teleconnections. *Journal of Quaternary Science* **25**(1), 62–78.

Wright Jr., H. E, J. E. Kutzbach, T. Webb III, W.F. Ruddiman, F. A. Street-Perrott, and P. J. Bartlein (1993). *Global Climates Since the Last Glacial Maximum*. Minnesota, USA: University of Minnesota Press.

Zheng, W., P. Braconnot, E. Guilyardi, U. Merkel, and Y. Yu (2008). ENSO at 6ka and 21ka from ocean–atmosphere coupled model simulations. *Climate Dynamics* **30**(7–8), 745–762.

Chapter 2

The climate of North America during the past 2,000 years reconstructed from pollen data

Viau, A.E [1]*, Ladd, M.[1], Gajewski, K.[1] *corresponding author

[1] Laboratory for Paleoclimatology and Climatology, Department of Geography, University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5

Abstract

The mean temperature of the warmest month was reconstructed for the past 2000 years using 748 pollen sites from the North American Pollen Database. The Modern Analog Technique (MAT) was used to quantify paleoclimate conditions using a modern pollen database with calibration sites from across North America. Across North America, both the Medieval Warm Period (MWP) and Little Ice Age (LIA) were cooler than the present (AD 1961–1990). The MWP was warmer than the LIA over at least the boreal and eastern portions of the continent and perhaps across the continent. These reconstructed anomalies during the MWP and LIA are significant anomalies from the long-term neoglacial cooling. The atmospheric circulation was likely dominated by a poleward shift of the summer Subtropical High Pressure system in the North Atlantic during the MWP.

Highlights:

- We use 748 pollen records from across North America to reconstruct the climatic conditions during the past 2,000 years.

- Temperatures during both the Medieval Period (MWP; 650–1050 AD) and the subsequent Little Ice Age (LIA; 1550–1850 AD) appear cooler than the past few decades.
- The climate during Medieval times was warmer than the LIA across North America.
- Results can be interpreted as due to a poleward shift in the position of Subtropical High Pressure Cells in summer during the MWP.

Keywords:

Little Ice Age, Medieval Warm Period, Modern Analog Technique, pollen, North America, quantitative paleoclimate reconstruction, North American Pollen Database, North American Modern Pollen Database

2.1 Introduction

The last 2000 years of our climate history has been the focus of much research effort (e.g. Lamb, 1965; Stine, 1994; Mann *et al.*, 1998; Crowley, 2000; Crowley and Lowery, 2000; Bradley *et al.*, 2003; Mann *et al.*, 2008; Mann *et al.*, 2009a). Much of the recent discussion is centered on the period between ∼800 and 1200 AD that has been called the Medieval Warm Period (MWP) or Medieval Climate Anomaly (MCA) (Lamb, 1965; IPCC, 2007; Mann *et al.*, 2009b; McIntyre and McKitrick, 2009). Although the subsequent period, the Little Ice Age (LIA, 1450–1850 AD), is generally considered a global phenomenon, the geographical extent of warmer conditions prevailing during Medieval times has not been clearly established (Hughes and Diaz, 1994; Bradley *et al.*, 2003; IPCC, 2007).

Several studies have attempted to provide large-scale syntheses of Earth's climate history by computing time series or showing maps of average summer temperatures on global or hemispheric scales for the past 2000 years (e.g. Mann *et al.*, 1998; Esper *et al.*, 2002; Moberg *et al.*, 2005; Mann *et al.*, 2008; Mann *et al.*, 2009a). These reconstructions have been based primarily on tree-rings, as there are many series available from around the northern hemisphere and world, and these have been shown to be reliable paleoenvironmental indicators (Fritts, 1991). Tree-rings provide precisely-dated records with annual or seasonal resolution that have provided climate records on interannual to multi-decadal scales for the past 600 years, but there is less confidence in the reconstructions during Medieval times, as there are fewer available series extending back to this time period (IPCC, 2007; Jones *et al.*, 2009). This

limitation has created a need to look toward other sources of data for information about the climate during this time.

Annually resolved reconstructions based on ice cores, corals, and speleothems, as well as lower resolution borehole records (Bradley *et al.*, 1999) have been included in these large-scale temperature syntheses but have added limited geographical coverage to an already partial picture of changes in climate during the first millennium CE (IPCC, 2007). Although there have been significant advances in improving the methodology of combining tree-ring records with other high-resolution records, it remains a problem when attempting to reconstruct further back in time where the network of sites becomes extremely sparse and inadequate for firm conclusions about the climate during the MWP. Some attempts have been made to include lower frequency climate variability by incorporating records such as pollen-based reconstructions from lakes with varved sediments, and these have proven quite useful in reconstructing longer-term variations on the order of century to multi-century scales (e.g. Moberg *et al.*, 2005). Ljungqvist (2009) used a more elaborate averaging technique with several multi-proxy high-resolution temperature reconstructions, but this attempt included reconstructions of different seasons, potentially masking the amplitude of change. What is needed to better understand the climate of the past 2000 years, especially during Medieval times, is to increase the spatial coverage of past observations in order to understand the patterns of change at regional scales (IPCC, 2007). These regional reconstructions can provide physically-based climate models with a benchmark data set to test mechanisms responsible for the variability in the climate of the past 2000 years, as well as to confirm global scale reconstructions.

In this paper, we reconstruct summer temperatures of the past 2000 years using a data set consisting of moderate-resolution pollen records from lake sediments from across North America. This reconstruction provides a quantitative evaluation of the regional extent of the MWP and LIA across the continent. We use the North American Pollen Database (NAPD) (Grimm *et al.*, 2007; Gajewski, 2008; Grimm, 2011), which is the most spatially extensive terrestrial data archive that spans the entire last 2000 years and beyond. These data have been extensively used to reconstruct past environmental conditions of the past 21,000 years (Wright Jr. *et al.*, 1993; Bartlein *et al.*, 2011, and references therein). Many of the climate reconstructions using pollen records have focussed on target periods such as 6000 and 21,000 years BP, and have proven useful in validating climate model simulations at hemispheric to global scales (e.g. COHMAP Members, 1988; Wright Jr. *et al.*, 1993; Sawada *et al.*, 2004; Williams *et al.*, 2007; Bartlein *et al.*, 2011).

The use of these pollen records in high-resolution paleoclimate reconstructions has

been underexploited but we argue that pollen records can provide high-resolution climate series of the past 2 ka (Gajewski, 1983; Gajewski, 1987; Gajewski, 1993; Viau *et al.*, 2006; Gajewski *et al.*, 2007; Gajewski and Viau, 2011). Analysis of several years of pollen monitoring in Europe has shown that, whereas the overall pollen composition is a function of the vegetation type on the surrounding landscape, changes in pollen production (the influx of pollen to the deposition site) are affected by weather at a daily to seasonal scale (e.g., Hicks, 2001; Autio and Hicks, 2004; Barnekow *et al.*, 2007; Huusko and Hicks, 2009). Results going back to the 1980s have clearly shown that pollen records can be used to reconstruct climatic variations during the LIA and MWP (Gajewski, 1987; Gajewski, 1988; Gajewski, 1993; Gajewski *et al.*, 2007). Numerous studies have shown a rapid response of terrestrial ecosystems to climate variations, with little or no time lag, during the late and postglacial period (Webb III, 1986; Gajewski, 1987; Grimm and Jacobson Jr, 2007; Grimm *et al.*, 1993; Viau *et al.*, 2002; Williams, 2003; Viau *et al.*, 2006; Peros *et al.*, 2008) when the sediment cores were samples at high enough resolution (Gajewski, 1993). Viau *et al.* (2008) and Viau and Gajewski (2009) have presented regional time series of temperature and precipitation from the boreal zone of Canada and Alaska that quantified the temperature and precipitation anomalies of the LIA and MWP.

Pollen records have been collected from around the world, and these are accessible in public databases (Grimm *et al.*, 2007; Gajewski, 2008). Thus, although pollen-based paleoclimate reconstructions do not presently have the resolution of tree-rings, for example, they nevertheless can provide important insight into the magnitude and spatial distribution of multi-decadal to century-scale climate variations such as the MWP.

2.2 Methods

2.2.1 Data

We use the North American Pollen Database (NAPD)(Grimm, 2011) which is an archive of 748 pollen records distributed unevenly across North America (Figure 1B). At each site a pollen diagram was prepared, where the pollen was extracted from a series of levels in the core. The sediments were dated using radiocarbon; a smaller number of samples are dated per sequence due to cost and sediment needed (Grimm *et al.*, 2007; Gajewski, 2008). For calibration, we had available ∼4800 modern pollen samples broadly distributed across North America (Figure 1A). Modern calibration data are available from all of the major vegetation zones and from ∼75% of the

"climate space" of North America (Whitmore *et al.*, 2005).

The spatial distribution of the fossil sites is uneven across North America although the Boreal, Conifer/Harwood, and Deciduous biomes are particularly well represented (Figure 1B). The temporal resolution varies from record to record but in total includes ∼8200 pollen samples and ∼320 14C dates for the period in question. These data are sufficient to evaluate persistent century to multi-century scale changes in climate (Viau *et al.*, 2006). All radiocarbon dates were calibrated to calendar years BP using INTCAL04 (Reimer *et al.*, 2004).

The uppermost sediments of lakes are less compacted, and if we assume a linear interpolation from the uppermost radiocarbon date to the surface, this would lead to ages being older than they really are, but in this section of the core only. Although this could be accounted for by using 210Pb dating of the uppermost 150 years, this has simply not been done for the cores in the database. Chronological control was therefore "tuned" to the beginning of the LIA cooling at 1450 AD (Table 24 in Appendix E), which is a well-established climatic transition (Lamb, 1965) in order to account for compaction effect. Tuning of chronologies to known tie points is done by oceanographers (Broecker and Donk, 1970) and ice core researchers (Fisher *et al.*, 1995).

2.2.2 Climate reconstruction

The Modern Analog Technique (MAT) consists of successively comparing a fossil pollen spectrum to all available modern pollen samples using the squared-chord distance (SCD) dissimilarity measure (Overpeck *et al.*, 1985). The modern climate value at the most similar modern pollen assemblage is then assigned to the fossil sample.

The MAT is a well-established paleoclimate reconstruction methodology that has been used for nearly 30 years (Overpeck *et al.*, 1985). A considerable amount of research has been done on the methodology (e.g. Jackson and Williams, 2004; Sawada *et al.*, 2004; Viau *et al.*, 2006; Viau *et al.*, 2008; Williams and Shuman, 2008; Viau and Gajewski, 2009), however, there remains some debate concerning certain choices that need to be made when using the technique. For example, there is no clear indication as to how many analogs should be retained to compute the reconstructed climate variable of interest (Williams and Shuman, 2008), as studies have shown problems arise when some commonly-considered rules of thumb are applied (Viau *et al.*, 2008). In this study, we used the best analog technique for the reasons shown in Viau *et al.* (2006) and Viau *et al.* (2008). The pollen sum (how many pollen taxa should be included in the SCD computation) is another uncertainty. After a thorough exploratory data analysis using several pollen sums (Figure 35 in Appendix E), it

was determined that the most reliable results were obtained at intermediate values, and therefore we used the 63 taxa pollen sum of Williams and Shuman (2008) (Table 5). One advantage of this pollen sum resides in the distinction between pines found in the southeast US and elsewhere which offers a useful ecological and geographical constraint to the reconstructions (Williams and Shuman, 2008).

We used the MATTOOLS R code (Sawada, 2006) to reconstruct the mean temperature of the warmest month (MTWA) from the fossil pollen series. The climate was reconstructed for each fossil site using the modern climate data of Whitmore *et al.* (2005). Results were then linearly interpolated to 100 calendar year intervals for further analysis and plotting (Viau *et al.*, 2006; Viau *et al.*, 2008; Viau and Gajewski, 2009).

2.2.3 Mapping

Maps were prepared for two time periods: the MWP (650–1050 AD) and LIA (1550–1850 AD) using the average of all samples within the time periods. We chose the period for analysis of the MWP (650–1050 AD) rather than the more traditional period of 800 to 1200 AD in order to compare the warmest interval in our reconstruction with modern conditions. We acknowledge that this slightly earlier time interval (150 years) could be the result of our chronological tuning due to the sedimentation compaction effect. The slightly later start of the LIA (1550 CE instead of 1450 CE) was chosen since the later interval is much colder. The summer temperature anomalies (MTWA), from modern conditions (1961–1990) for the MWP and LIA, were mapped using ESRI ArcGIS™(v.9.3) and gridded at 2° resolution. A map of the difference in MTWA between the MWP and LIA (MWP–LIA) was also prepared. As there were a different number of sites with reconstructed MTWA values during these two time periods, anomaly values were computed using only sites that contained data from both periods.

Gridding was done using kriging (Isaaks and Srivastava, 1989). Kriging has the advantage of evaluating the uncertainty of the interpolation process. Ordinary prediction kriging (spherical) was used with no anisotropy and no smoothing or transformation of the data. We constrained our gridding radius to 2° in latitude and longitude and +/-4°C in order to avoid spurious extrapolation and "edge effects", where points are at the edge of data limits; although this creates a more spatially variable reconstruction, it is a more accurate depiction of the data. This resulted in some areas not being mapped due to unreasonable anomaly estimates. Areas with less reliable reconstructions due to low site density, high anomaly estimates, or "edge effects" were hatched (Figure 2). The hatching is based on Sawada *et al.* (2004) and

is also used for a 2,000 year reconstruction of ANNP (Figure 82 in Appendix E).

2.2.4 Time series

Regional time series were constructed for the major North American biomes aggregated from the Fedorova *et al.* (1994) classification, in order to evaluate regional differences in the temperature of the past 2000 years. There were a small number of sites which were unclassified in the Whitmore *et al.* (2005) database, and these were not used for this figure. The MTWA for each region was computed as the average of each 100 calendar year intervals for each ecoregion to construct the regional time series. Standard errors (i.e. uncertainty envelopes) on the regional reconstructions were computed as in Viau *et al.* (2006).

2.3 Results

Both the MWP and LIA appear relatively cooler than present (Figure 2A,B) over large areas of North America. The MWP (Figure 2A) was cooler than present across the central portion of the continent and in southeastern United States, areas where there is a reasonable density of fossil sites and calibration samples. Temperatures greater than today were reconstructed in Alaska, and much of eastern Canada and around Hudson Bay. However, there are few data in the Canadian Arctic Archipelago, or in the James Bay region, so it is not clear that these anomalies are reliable. Alaska, however, does have a reasonable density of sites and modern calibration samples.

The Little Ice Age was cooler than today across all of North America (Figure 2B). The small areas of positive anomalies in the north are extrapolations and therefore less reliable. Three areas in the boreal zone with positive anomalies are the southern Alaska coast, where there are no fossil data in the database, a "bulls-eye" of warm reconstruction in Alberta/Saskatchewan, and in Labrador. Sites in the Canadian Prairies show both positive and negative anomalies at nearby sites, although the vegetation (pollen) sequences look similar, so we cannot consider these reliable. Chance variations in the pollen assemblages may be causing the MAT to find analogs in widely varying areas. The Labrador area may be an edge effect, or alternatively, may actually have been warmer, due perhaps to influence from the adjacent ocean.

Although the MWP (Figure 2A), was cooler than the present, the pollen records show that the MWP was warmer than the LIA (Figure 2B) across most of North America (Figure 2C). Only four areas show negative anomalies. In Texas and the far northwest of Canada, there are few or no data, so we do not consider these reliable.

The northern Plains and Mountain area have a reasonable site density on the map, but given the wide topographic variability in the region we are not convinced this result is reliable. Again, Labrador shows a slight negative anomaly; this region should be considered more closely in synthesis studies.

Regional time series from the forest-tundra, boreal, conifer-hardwood forest (hardwood in Figure 3) and deciduous ecozones show positive anomalies up to 0.6°C during the MWP, although the anomaly is more typically less than 0.2°C (Figure 3). Anomalies up to 0.3 °C are reconstructed for the LIA, although the LIA cooling is less clear in the forest-tundra area, perhaps due to few data and the difficulty in defining this diffuse zone. Regional series from the central and western USA show more complex patterns, but these areas have fewer data and show larger uncertainties in an area of complex topography.

Analysis of the distribution of squared chord distance (SCD) between modern samples (Sawada, 2006) shows that a value of approximately 0.3 indicates a potentially non-analog modern sample for a fossil assemblage (Figure 77 in Appendix E). Analysis of the SCD between all fossil samples and the modern data indicates that modern analogs can be found for almost all samples, especially if the first analog is considered (Figure 4). If the first 5 analogs are considered (Figure 4), then more samples do not find a close analog in the modern data set, although with the exception of several occasional samples, reasonable analogs can be identified in the modern data set. Thus, non-analogs are not a problem in this analysis.

For the forest-tundra, boreal, conifer-hardwood and deciduous biomes, there are many sites available, with a reasonable number of radiocarbon dates and a high density of samples (Figure 4E-H). Fewer samples and radiocarbon dates are available for the remaining biomes, especially the coastal biome of the Pacific Northwest, where our reconstruction is poorly constrained by radiocarbon dates (Figure 4A, B, D). These reconstructions are therefore less reliable, and this is further complicated by the large altitudinal gradients in the mountain zone and difficulty of pollen analysis in desert regions. The prairie biome has a reasonable number of radiocarbon dates constraining the chronology, but here precipitation is a major control on the vegetation and the temperature reconstruction may be confounded with this (Figure 4C).

2.4 Discussion and conclusions

The pollen-based paleoclimate reconstructions show that warm conditions during the MWP and cool conditions in the LIA were all nevertheless cooler than the 1961–1990

base period, and this result emerges even without comparing the results to the instrumental record. Possible exceptions that warrant further study are Labrador, where the effect of the adjacent ocean on the land climate may have caused the opposite patterns (e.g. effects of the North Atlantic Oscillation, NAO) and perhaps the mountain region of the northern Plains and the Intermountain zone, where changes in the Rossby waves could have caused these patterns. These results are consistent with a regional Boreal reconstruction (Viau and Gajewski, 2009), as well as a series of reconstructions based on pollen data from varved sediments in the conifer-hardwood forest (Gajewski, 1988).

These pollen-based reconstructions are also broadly comparable to those based on tree-rings (Mann *et al.*, 2008). However, these reconstructions, along with other work of the past decade (e.g. Viau *et al.*, 2006) do not support the interpretation of the climate of the past 2000 years as simply a gradual cooling due to orbital-scale forcing alone (Kaufman *et al.*, 2009). Other high-frequency changes in climate were operating on century to millennial scales that are superimposed on the orbital-scale forcing (Gajewski, 1987; Bond *et al.*, 2001; Viau *et al.*, 2006; Feng *et al.*, 2009). These quasi-cyclical high-frequency changes appear to have been strong enough to bring about synchronous temperature changes in at least three major ecoregions of North America (boreal, conifer-hardwood, deciduous), as well as between Europe and North America (Gajewski, 2006).

Edge effects and low densities of sites in some regions suggest some caution in interpretation, as discussed above. However, in regions where we are confident of the reconstructions, these results show generally warmer conditions prevailing during the MWP as compared to the LIA (Figure 2C, Figure 3). Climatologically, these results can be interpreted as a northward shift in the subtropical high pressure systems of the Atlantic (Bermuda High) and perhaps the Northeast Pacific during the summer due to an expanded Hadley cell (Trouet *et al.*, 2009) during the MWP. This pattern could have led to less sea ice extent in the North Atlantic (Lamb, 1965; Graham *et al.*, 2011) and greater ice melt of glaciers in the Arctic (Fisher and Koerner, 1994).

The reconstructions presented here provide data in key regions of North America where other proxy-climate data are lacking. Conversely, our reconstructions are less reliable in the western United States, which is an area where longer tree-ring records are available, and where there is a high density of tree-ring sites. The North, where we have few data, is also a region where reliable ice core reconstructions are available (e.g. Fisher and Koerner, 1994). Our reconstruction provides important insight into the climate for large regions of North America during the MWP, which is precisely

where data and reconstructions are needed to better understand the geographical extent of the climate anomalies of this time period. These results also provide insight into ecosystem responses to climate variability during the past 2000 years (Gajewski, 1987).

This study has provided several conclusions:

- The modern climate appears warmer than both the MWP and LIA (Figure 2).
- The MWP was warmer than the LIA over at least three large forest ecoregions of North America (Figure 3).
- Reconstructions from the western coastal, mountain, desert, and Arctic ecoregions are less reliable due to fewer available data and larger uncertainties in the reconstructions. The prairie ecoregion reconstruction may be confounded with moisture availability rather than temperature.
- These large-scale changes during Medieval times are not insignificant and do not appear to result from a gradual decline in temperature due to orbital-scale forcing.
- These results can be interpreted as an expansion northward of the summer Subtropical High Pressure system in the North Atlantic during the MWP.
- An effort is needed to combine the pollen based reconstruction with other proxies to map the climate patterns across North America during the past 2 ka.

Acknowledgments

This research was supported by grants from the Natural Sciences and Engineering Council of Canada (NSERC) to AV and KG.

Table 1: Pollen taxa used in reconstructions [Williams and Shuman (2008)].

Abies	*Liquidambar*
Acer	Lycopodium [a]
Alnus rubra	Minor Forbs [b]
Alnus rugosa	Myricaceae
Alnus [c]	*Nyssa*
Aquifoliaceae	*Ostrya/Carpinus*
Arceuthobium	*Oxyria*
Artemisia	Papaveraceae
Asteraceae [d]	Picea [e]
Betula	*Pinus diploxylon*
Brassicaceae	*Pinus haploxylon*
Cactaceae	Pinus undifferentiated
Carya	Pinus Southeast
Caryophyllaceae	Plantaginaceae
Castanea	*Platanus*
Celtis	Poaceae
Cephalanthus	*Populus*
Chenopodiaceae/Amaranthaceae	*Prosopis*
Chrysolepis/Lithocarpus	*Pteridium*
Corylus	*Quercus*
Cupressaceae	Rosaceae
Cyperaceae	*Salix*
Dryas	*Sarcobatus*
Ephedra	Saxifraga [f]
Ericaceae [g]	*Sphagnum*
Fagus	*Taxodium*
Fraxinus nigra	*Tilia*
Fraxinus pennsylvanica/americana	*Tsuga heterophylla*
Fraxinus undifferentiated	*Tsuga mertensiana*
Juglans [h]	Tsuga undifferentiated
Larix/Pseudotsuga	*Ulmus*
Larrea	

[a]Sum of *Lycopodium* undiff. *L. annotinum*, *L. clavatum*, *L. complanatum* and *L. selago*

[b]Sum of Apiaceae, *Eriogonum*, Euphorbiaceae, Malvaceae.

[c]Sum of *Alnus crispa* and *Alnus* undiff

[d]Sum of Asteraceae undiff, Asteraceae subf. Cichorioideae, *Iva* and *Xanthium*. Note that this taxon does not include *Artemisia* or *Ambrosia*.

[e]Sum of *Picea* undiff, *Picea glauca* and *P. mariana*

[f]Sum of *Saxifraga cernua*, *S. hieracifolia*, *S. oppositifoli*, and *S. tricuspidata*

[g]Sum of Ericaceae undiff., *Chamaedaphne/Ledum* and *Vaccinium*

[h]Sum of *Juglans* undiff, *J. cinerea*, and *J. nigra*

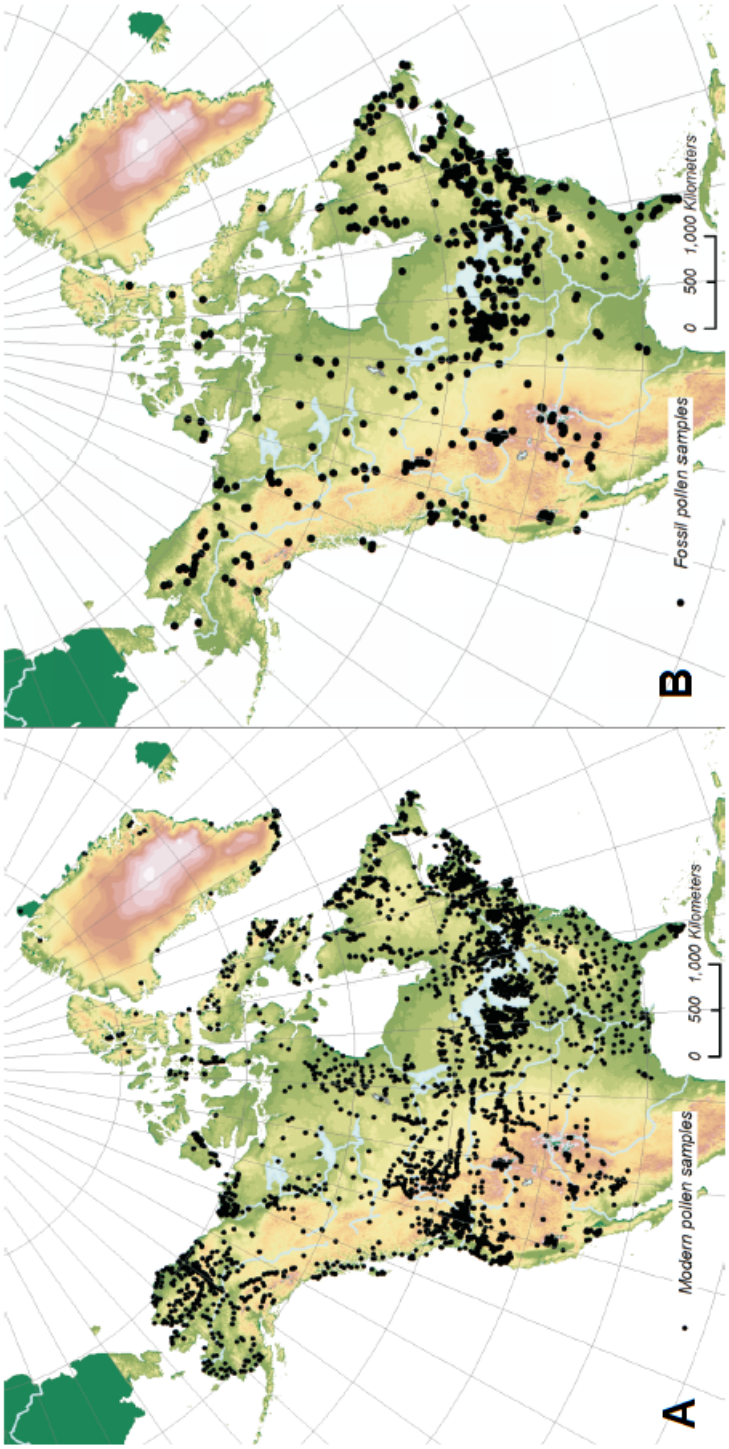


Figure 1: Spatial distribution of modern (A) and fossil (B) pollen sites in North America (Viau, Ladd, and Gajewski, 2012)

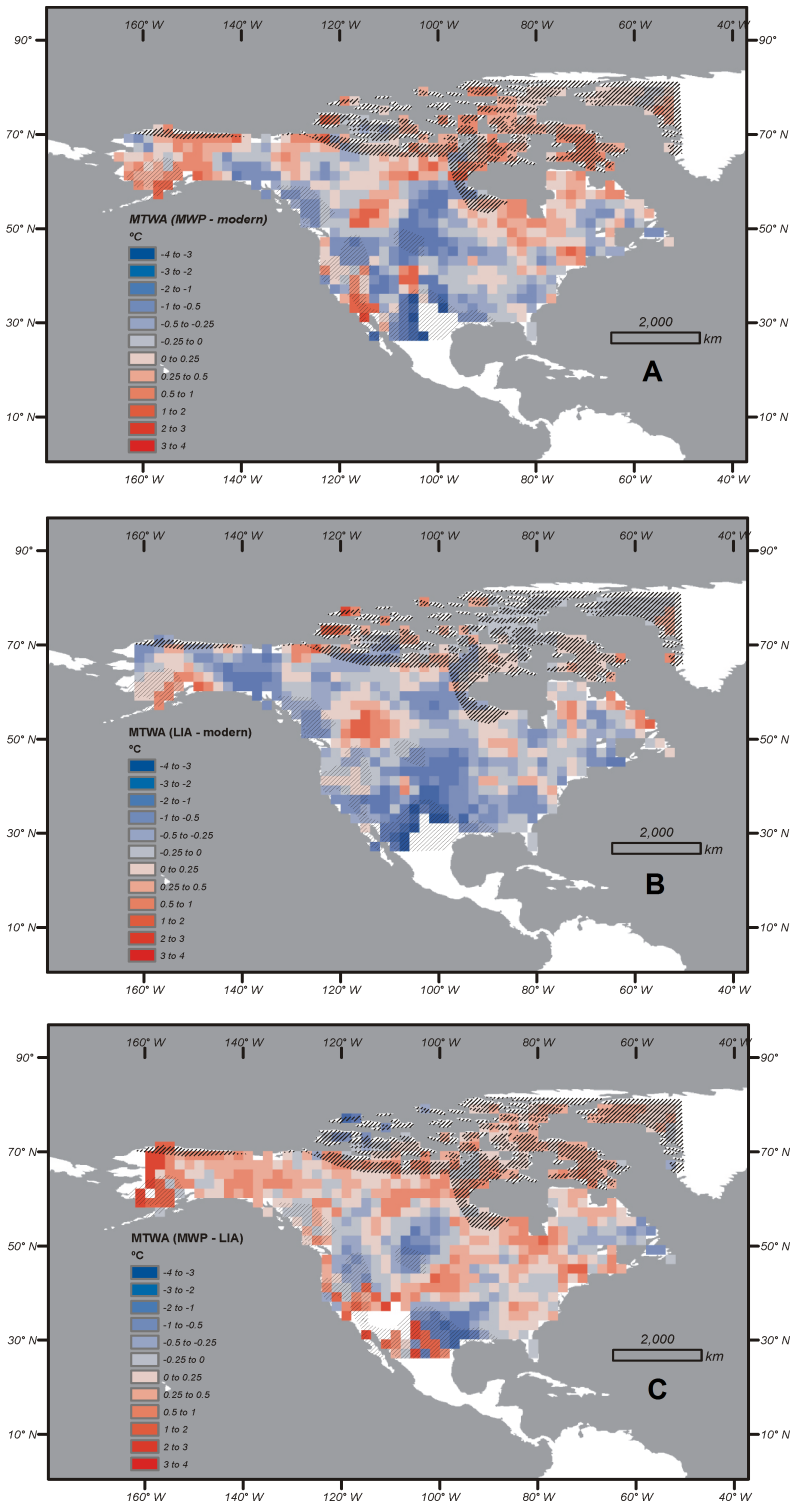


Figure 2: Maps of the anomalies of mean temperature of the warmest month (MTWA). (A) Medieval Warm Period (MWP; 650–1050 AD) in relation to the modern (1961–1990 base period). (B) Little Ice Age (LIA; 1550–1850 AD) in relation to the modern. (C) MWP minus LIA. Hatched regions indicate areas of low confidence reconstructions and are adapted from Sawada *et al.* (2004).

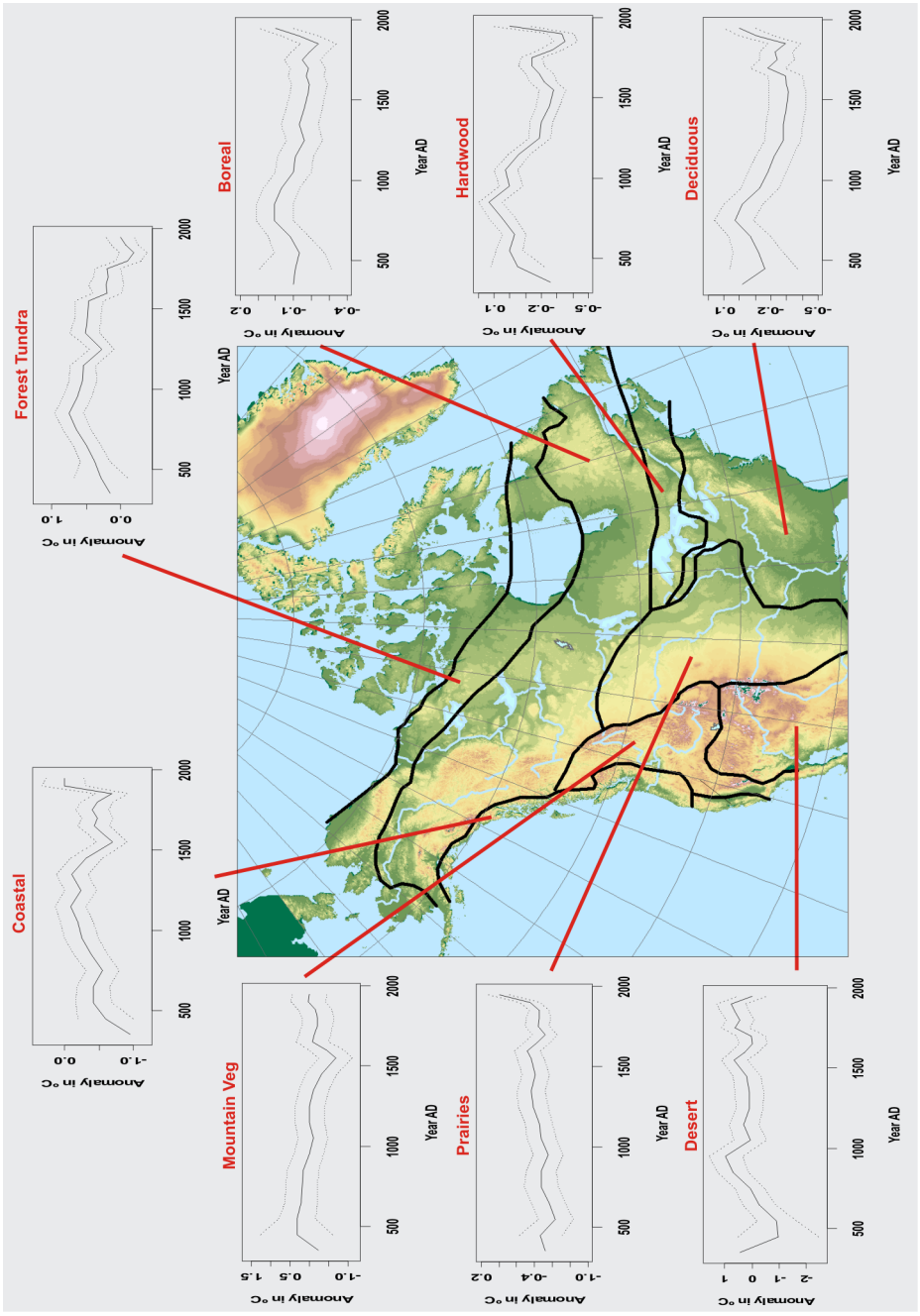


Figure 3: Time series of regional averages of the mean temperature of the warmest month (MTWA) anomalies for the period 350–1950 AD. All time series within the biomes were interpolated to 100-year intervals and averaged. Uncertainty envelopes were computed as in Viau *et al.* (2006). Note that the instrumental record is not included in the time series.

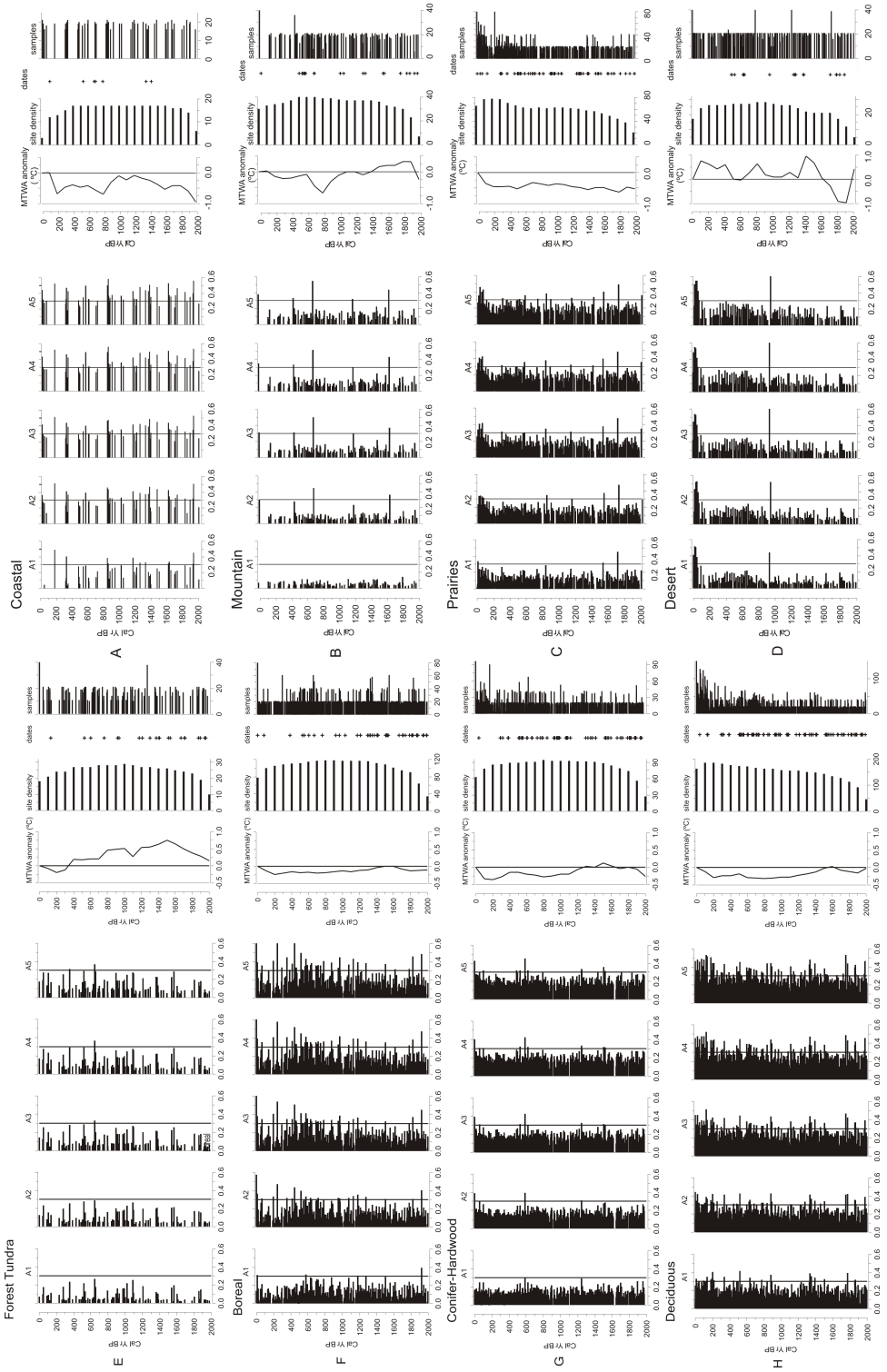


Figure 4: Squared chord distance (SCD) values for the top 5 analogs (A1 to A5) and MTWA anomaly, site density, dates and samples for (A) coastal, (B) mountain, (C) prairies, (D) desert, (E) forest-tundra, (F) boreal, (G) conifer-hardwood and (H) deciduous regions. These ecoregions were aggregated from Fedorova *et al.* (1994) to correspond to the major biomes of North America. The Arctic and Mediterranean regions were not included in this study due to lack of data. On the right are shown the mean temperature of the warmest month anomalies (MTWA), the number of sites averaged for each interpolated value, the radiocarbon dates used for the regional chronologies and the number of pollen samples available for the region. Note that these plots are not chronologically tuned as in Figure 3.

References

Autio, J and S. Hicks (2004). Annual variations in pollen deposition and meteorological conditions on the fell Aakenusunturi in northern Finland: potential for using fossil pollen as a climate proxy. *Grana* **43**(1), 31–47.

Barnekow, L., N.J. Loader, S. Hicks, C.A. Froyd, and T. Goslar (2007). Strong correlation between summer temperature and pollen accumulation rated for Pinus sylvestris, Picea abies and Betula spp in a high-resolution record from northern Sweden. *Journal of Quaternary Science* **22**(7), 653–658.

Bartlein, P.J., S.J. Harrison, S. Brewer, S. Connor, B.A.S. Davis, K. Gajewski, J. Guiot, T.I. Harrison-Prentice, A. Henderson, O. Peyron, I.C. Prentice, M. Scholze, H. Seppä, B. Shuman, S. Sugita, R.S. Thompson, A.E. Viau, J. Williams, and H. Wu (2011). Pollen-based continental climate reconstructions at 6 and 21 ka: a global synthesis. *Climate Dynamics* **37**(3-4), 775–802.

Bond, G., B. Kromer, J. Beer, R. Muscheler, N.E. Evans, W. Showers, S. Hoffman, R. Lotti-Bond, I. Hajdas, and G. Bonani (2001). Persistent solar influence on North Atlantic climate during the Holocene. *Science* **294**(5549), 2130–2136.

Bradley, R.S., M.E. Mann, and M.K. Hughes (1999). Northern hemisphere temperatures during the past millennium: inferences, uncertainties, and limitations. *Geophysical Research Letters* **26**(6), 759–762.

Bradley, R.S., M.K. Hughes, and H.F. Diaz (2003). Climate in Medieval Time. *Science* **302**(2), 404–405.

Broecker, W.S. and J. Donk (1970). Insolation changes, ice volumes, and the O18 record in deep-sea cores. *Reviews of Geophysics* **8**(1), 169–198.

COHMAP Members (1988). Climatic changes of the last 18,000 years: Observations and model simulations. *Science* **241**, 1043–1052.

Crowley, T. J. (2000). Causes of climate change over the past 1000 years. *Science* **289**(5477), 270–277.

Crowley, T. J. and T. Lowery (2000). How Warm Was the Medieval Warm Period? *Ambio* **29**(1), 51–54.

Esper, J., E.R. Cook, and F.H. Schweingruber (2002). Low-frequency signals in long tree-ring chronologies for reconstructing past temperature variability. *Science* **295**(5563), 2250–2253.

Fedorova, I.T., Y.A. Volkova, and D.L. Varlyguin (1994). World Vegetation Cover: Digital raster data on a 30-minute Cartesian orthonormal geodetic (lat./long.) 1,080× 2,160 grid. *Global Ecosystems Database Version* **2**.

Feng, S., Q. Hu, and R.J. Oglesby (2009). AMO-like variations of holocene sea surface temperatures in the North Atlantic Ocean. *Climate of the Past Discussions* **5**(6), 2465–2496.

Fisher, D.A. and R.M. Koerner (1994). Signal and noise in four ice-core records from the Agassiz Ice Cap, Ellesmere Island, Canada: details of the last millennium for stable isotopes, melt and solid conductivity. *The Holocene* **4**(2), 113–120.

Fisher, D.A., R.M. Koerner, and N. Reeh (1995). Holocene climatic records from Agassiz ice cap, Ellesmere Island, NWT, Canada. *The Holocene* **5**(1), 19–24.

Fritts, H.C. (1991). *Reconstructing large-scale climatic patterns from tree-ring data*. Tucson, AZ (United States); University of Arizona Press.

Gajewski, K. (1983). On the interpretation of climatic change from the fossil record: Climatic change in central and eastern United States for the past 2000 years estimated from pollen data. PhD thesis. University of Wisconsin-Madison.

— (1987). Climatic impacts on the vegetation of eastern North America during the past 2000 years. *Vegetatio* **68**(3), 179–190.

— (1988). Late Holocene climate changes in eastern North America estimated from pollen data. *Quaternary Research* **29**(3), 255–262.

— (1993). The role of paleoecology in the study of global climatic change. *Review of palaeobotany and palynology* **79**(1), 141–151.

— (2006). Essai: Is Arctic Palynology a "Blunt Instrument"? *Géographie physique et Quaternaire* **60**(2), 95–102.

— (2008). The Global Pollen Database in biogeographical and palaeoclimatic studies. *Progress in Physical Geography* **32**(4), 379–402.

Gajewski, K. and A.E. Viau (2011). Abrupt climate changes during the Holocene across North America from pollen and paleolimnological records. *Abrupt Climate Change: Mechanisms, Patterns, and Impacts, Geophysical Monograph Series*. Ed. by L. Polyak H. Rashid and E. Mosley-Thompson. Washington, D.C., USA: AGU, 161–171.

Gajewski, K., A.E. Viau, and M.C. Sawada (2007). Millennial-scale climate variations in the Holocene - the terrestrial record. *Climate Variability and Change: Past,*

Present and Future. Ed. by G. Kutzbach. Madison, Wisconsin, USA: University of Wisconsin, 133–154.

Graham, N.E., C.M. Ammann, D. Fleitmann, K.M. Cobb, and J. Luterbacher (2011). Support for global climate reorganization during the "Medieval Climate Anomaly". *Climate dynamics* **37**(5–6), 1217–1245.

Grimm, E.C. (2011). *North American Pollen database. Data archived at IGBP PAGES/-World Data Center–A for Paleoclimatology, NOAA/NGDC Paleoclimatology Program, Boulder, CO, USA*. url: http://www.ncdc.noaa.gov/paleo.

Grimm, E.C. and G.L. Jacobson Jr (2007). Millennial-scale climate variations in the Holocene - the terrestrial record. *Climate Variability and Change: Past, Present and Future*. Ed. by G. Kutzbach. Madison, Wisconsin, USA: University of Wisconsin, 133–154.

Grimm, E.C., G.L. Jacobson Jr, W.A. Watts, B.C.S. Hansen, and K.A. Maasch (1993). A 50,000-Year Record of Climate Oscillations from Florida and its Temporal Correlation. *Science* **61**, 9.

Grimm, E.C., J. Keltner, R. Cheddadi, S. Hicks, A.M. Lézineé, J.C. Berrio, and J.W. Wiliams (2007). Databases and their application. *Encyclopedia of Quaternary Science*. Ed. by S. A. Elias. **3**. Amsterdam: Elsevier, 2521–2528.

Hicks, S. (2001). The use of annual arboreal pollen deposition values for delimiting tree-lines in the landscape and exploring models of pollen dispersal. *Review of Palaeobotany and Palynology* **117**(1), 1–29.

Hughes, M.K. and H.F. Diaz (1994). Was there a "Medieval Warm Period", and if so, where and when? *Climatic Change* **26**(2–3), 109–142.

Huusko, A. and S. Hicks (2009). Conifer pollen abundance provides a proxy for summer temperature: evidence from the latitudinal forest limit in Finland. *Journal of Quaternary Science* **24**(5), 522–528.

IPCC (2007). *Climate Change 2007, The Physical Science Basis. Fourth Assessment Report*. Ed. by S. Solomon, D. Manning, Z. Chen, M. Marquis, K.B. Averyt, M. Tignor, and H. Miller. Cambridge, UK and New York, NY, USA: Cambridge University Press, p. 996.

Isaaks, E.H. and R.H. Srivastava (1989). *An Introduction to Applied Geostatistics*. New York, USA: Oxford University Press.

Jackson, S.T. and J.W. Williams (2004). Modern analogs in Quaternary paleoecology: here today, gone yesterday, gone tomorrow? *Annual Review of Earth and Planetary Sciences* **32**, 495–537.

Jones, P.D., K.R. Briffa, T.J. Osborn, J.M. Lough, T.D. Van Ommen, B.M. Vinther, J. Luterbacher, E.R. Wahl, F.W. Zwiers, M.E. Mann, C.M. Schmidt G.A.and Ammann, B.M. Buckley, K.M. Cobb, J. Esper, H. Goosse, N. Graham, E. Jansen, T. Kiefer, C. Kull, M. Kttel, E. Mosley-Thompson, J.T Overpeck, N. Riedwyl, A.W. Schulz M. Tudhope, R. Villalba, H. Wanner, E. Wolff, and E. Xoplaki (2009). High-resolution palaeoclimatology of the last millennium: a review of current status and future prospects. *The Holocene* **19**(1), 3–49.

Kaufman, D.S., D.P. Schneider, N.P. McKay, C.M. Ammann, R.S. Bradley, K.R. Briffa, G.H. Miller, B.L. Otto-Bliesner, J.T. Overpeck, B.M. Vinther, and Arctic Lakes 2k Project Members (2009). Recent warming reverses long-term Arctic cooling. *Science* **325**(5945), 1236–1239.

Lamb, H.H. (1965). The early medieval warm epoch and its sequel. *Palaeogeography, Palaeoclimatology, Palaeoecology* **1**, 13–37.

Ljungqvist, F.C. (2009). Temperature proxy records covering the last two millennia: a tabular and visual overview. *Geografiska Annaler: Series A, Physical Geography* **91**(1), 11–29.

Mann, M.E., R.S. Bradley, and M.K. Hughes (1998). Global-scale temperature patterns and climate forcing over the past six centuries. *Nature* **392**(6678), 779–787.

Mann, M.E, M.K. Zhang Z.and Hughes, R.S Bradley, S.K. Miller, S. Rutherford, and F. Ni (2008). Proxy-based reconstructions of hemispheric and global surface temperature variations over the past two millennia. *Proceedings of the National Academy of Sciences* **105**(36), 13252–13257.

Mann, M.E, Z. Zhang, S. Rutherford, R.S. Bradley, M.K. Hughes, D. Shindell, C. Ammann, G. Faluvegi, and F. Ni (2009a). Global signatures and dynamical origins of the Little Ice Age and Medieval Climate Anomaly. *Science* **326**(5957), 1256–1260.

Mann, M.E., R.S. Bradley, and M.K. Hughes (2009b). Reply to McIntyre and McKitrick: Proxy-based temperature reconstructions are robust. *Proceedings of the National Academy of Sciences* **106**(6), E11.

McIntyre, S. and R. McKitrick (2009). Proxy inconsistency and other problems in millennial paleoclimate reconstructions. *Proceedings of the National Academy of Sciences* **106**(6), E10.

Moberg, A., D.M. Sonechkin, K. Holmgren, N.M. Datsenko, and W. Karlén (2005). Highly variable Northern Hemisphere temperatures reconstructed from low-and high-resolution proxy data. *Nature* **433**(7026), 613–617.

Overpeck, J.T., T. Webb III, and I.C. Prentice (1985). Quantitative interpretation of fossil pollen spectra: dissimilarity coefficients and the method of modern analogs. *Quaternary Research* **23**(1), 87–108.

Peros, M.C., K. Gajewski, and A.E. Viau (2008). Continental-scale tree population response to rapid climate change, competition and disturbance. *Global Ecology and Biogeography* **17**(5), 658–669.

Reimer, P.J., M.G.L. Baillie, E. Bard, A. Bayliss, J.W. Beck, C.J.H. Bertrand, P.G. Blackwell, C.E. Buck, G.S. Burr, K.B. Cutler, P.E. Damon, R.L. Edwards, R.G. Fairbanks, M. Friedrich, T.P. Guilderson, A.G. Hogg, K.A. Hughen, B. Kromer, G. McCormac, S. Manning, C.B. Ramsey, R.W. Reimer, S. Remmele, J.R. Southon, M. Stuiver, S. Talamo, F.W. Taylor, J. van der Plicht, and C.E. Weyhenmeyer (2004). IntCal04 terrestrial radiocarbon age calibration, 26e0 ka BP. *Radiocarbon* **46**.

Sawada, M., A.E. Viau, G. Vettoretti, W.R. Peltier, and K. Gajewski (2004). Comparison of North-American pollen-based temperature and global lake-status with CCCma AGCM2 output at 6ka. *Quaternary Science Reviews* **23**(3), 225–244.

Sawada, M.C. (2006). An open source implementation of the Modern Analog Technique (MAT) within the R computing environment. *Computers and Geosciences* **32**(6), 818–833.

Stine, S. (1994). Extreme and persistent drought in California and Patagonia during mediaeval time. *Nature* **369**(6481), 546–549.

Trouet, V., J. Esper, N.E. Graham, A. Baker, J.D. Scourse, and D.C. Frank (2009). Persistent positive North Atlantic Oscillation mode dominated the medieval climate anomaly. *Science* **324**(5923), 78–80.

Viau, A., M. Ladd, and K. Gajewski (2012). The climate of North America during the past 2000 years reconstructed from pollen data. *Global and Planetary Change* **84**, 75–83.

Viau, A.E. and K. Gajewski (2009). Reconstructing millennial-scale, regional paleoclimates of boreal Canada during the Holocene. *Journal of Climate* **22**(2), 316–330.

Viau, A.E., K. Gajewski, P. Fines, D.E. Atkinson, and M.C. Sawada (2002). Widespread evidence of 1500 yr climate variability in North America during the past 14 000 yr. *Geology* **30**(5), 455–458.

Viau, A.E., K. Gajewski, M.C. Sawada, and P. Fines (2006). Millennial-scale temperature variations in North America during the Holocene. *Journal of Geophysical Research: Atmospheres (1984-2012)* **111**(D9).

Viau, A.E., K. Gajewski, M.C. Sawada, and J. Bunbury (2008). Low-and high-frequency climate variability in eastern Beringia during the past 25 000 years. *Canadian Journal of Earth Sciences* **45**(11), 1435–1453.

Webb III, T. (1986). Is vegetation in equilibrium with climate? How to interpret late-Quaternary pollen data. *Vegetatio* **67**(2), 75–91.

Whitmore, J., K. Gajewski, M.C. Sawada, J.W. Williams, B. Shuman, P.J. Bartlein, T. Minckley, A.E. Viau, T. Webb III, S. Shafer, P. Anderson, and L. Brubaker (2005). Modern pollen data from North America and Greenland for multi-scale paleoenvironmental applications. *Quaternary Science Reviews* **24**(16), 1828–1848.

Williams, J.W. (2003). Variations in tree cover in North America since the last glacial maximum. *Global and Planetary Change* **35**(1), 1–23.

Williams, J.W. and B. Shuman (2008). Obtaining accurate and precise environmental reconstructions from the modern analog technique and North American surface pollen dataset. *Quaternary Science Reviews* **27**(7), 669–687.

Williams, J.W., S.T. Jackson, and J.E. Kutzbach (2007). Projected distributions of novel and disappearing climates by 2100 AD. *Proceedings of the National Academy of Sciences* **104**(14), 5738–5742.

Wright Jr., H. E, J. E. Kutzbach, T. Webb III, W.F. Ruddiman, F. A. Street-Perrott, and P. J. Bartlein (1993). *Global Climates Since the Last Glacial Maximum*. Minnesota, USA: University of Minnesota Press.

Chapter 3

The impact of using different modern climate data sets in pollen-based paleoclimate reconstructions of North America

Ladd, M.[1]*, Way, R.G.[2], Viau, A.E [1] *corresponding author

[1] Laboratory for Paleoclimatology and Climatology, Department of Geography, University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5

[2] Department of Geography, University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5

Abstract

The use of different modern climate data sets is shown to impact a continental-scale pollen-based reconstruction of mean July temperature (TJUL) over the last 2,000 years for North America. Data from climate stations, physically modeled from climate stations and reanalysis products are used to calibrate the reconstructions. Results show that the use of different modern data sets produces warmer and/or smoother reconstructions, especially using reanalysis products. The reconstructions during the period of 1050–1550 CE are shown to be more variable because of a high latitude cold-bias in the modern TJUL data. The ultra-high resolution WorldClim gridded data may only useful if the modern pollen sites have at least the same spatial precision as the gridded data set. Hence we justify the use of the lapse-rate corrected University of East Anglia Climate Research Unit (CRU) based Whitmore modern

climate data (Whitmore *et al.*, 2005) set for North American pollen-based climate reconstructions.

3.1 Introduction

The past decade has seen a rapid increase in available computing resources, resulting in improved data assimilation methods and modern climate data sets. But more importantly, the recent development of free online climate data explorers and repositories has permitted easier access to new data resources and research tools for mainstream climate scientists. These widely available research tools and data sets are increasingly used in paleoclimatological research (e.g. Trouet and Van Oldenborgh, 2013). This poses a new question in paleoclimate research in how much the choice of a particular modern calibration data set can impact paleoclimate reconstructions (Bartlein *et al.*, 2011). For example, Fortin and Gajewski (2012) showed that an Arctic paleoclimate reconstruction was cold biased using the New *et al.* (2002) gridded climate data set (10 arc-minutes / ∼18 km / ∼0.17°) as compared to a very high resolution (30 arc-seconds / ∼1 km / ∼0.009°) regional climate data set that accounted for meteorological processes thought to operate in the Arctic. The lower resolution modern climate data was not capturing the local climate effects in the Arctic region (Atkinson and Gajewski, 2002).

The most widely used and readily available modern calibration climate data sets can broadly be described as fitting into three categories: (a) meteorological station data [e.g. BEST, Berkeley Earth Surface Temperature (2013), Rohde *et al.* (2012), and Rohde *et al.* (2013); CRU, New *et al.* (2002)]; (b) atmospheric reanalysis data sets [e.g. NCEP, Kistler *et al.* (2001); ERA-40,Uppala *et al.* (2005)] and; (c) observation-based physical models [e.g. WorldClim, Hijmans *et al.* (2005); WHIT, Whitmore *et al.* (2005)]. Interactive websites and software packages such as the Climate Explorer [http://climexp.knmi.nl; Trouet and Van Oldenborgh (2013)], Geographic Information Systems (GIS) and open-source software such as R and Python facilitate reading and processing spatio-temporal data sets (e.g. Network Common Data Format [NetCDF]).

Paleoclimate reconstructions rely on the accuracy of modern climate data sets for calibration of fossil records. Various methods are used to reconstruct past climates (Bradley *et al.*, 1999) with strengths and weaknesses discussed in Birks *et al.* (2010). For example, pollen reconstructions have used a variety of methods such as the Modern Analog Technique (MAT) (Overpeck *et al.*, 1985), response surfaces (Gonzales *et al.*, 2009) and weighted-average partial least squares (PLS) regression

methods (Braak and Juggins, 1993). However, these methods all rely on a modern calibration data set to reconstruct changes in climate through time. In this paper, we explore the sensitivity of using different modern climate data sets for a 2,000 year pollen-based paleoclimate reconstruction of summer temperature in North America for the last 2,000 years (Viau *et al.*, 2012).

3.2 Data and Methods

3.2.1 Fossil and modern pollen data

The ensemble of reconstructions presented in this study follow the same methodological approach (Modern Analog Technique [MAT]) as in the Viau *et al.* (2012) North American pollen-based summer temperature reconstruction for the past 2,000 years. The fossil data consists of ∼750 fossil pollen records distributed across North America (Figure 5A). We used the modern sample distribution from the Whitmore *et al.* (2005) database that consists of ∼4,800 modern pollen samples and associated bioclimatic variables across North America (Figure 5B). The Whitmore *et al.* (2005) modern climate variables are a modified version of the New *et al.* (2002) data set where regional lapse rates are used to adjust site-specific temperatures (e.g. Spenke *et al.*, 1994; Kerwin *et al.*, 2004). A customized R program was used to reconstruct July temperature using multiple modern calibration data sets with the Whitmore *et al.* (2005) modern climate data result being identical to the reconstruction presented in Viau *et al.* (2012) (Chapter 2). Each additional reconstruction is based on different modern climate data sources (Table 2) where the nearest grid point or station value is assigned at each modern pollen site.

3.2.2 New *et al.*, 2002 (NEW)

The New *et al.* (2002) data set has a lat/lon resolution of ∼0.17° and is based on the Climate Research Unit (CRU) gridded climate data set with spatial interpolation used for regions with no data. The mean July temperature for 1961–1990 (standard base period) was used from New *et al.* (2002). The New *et al.* (2002) data set was used rather than the newer CRU gridded product (Harris *et al.*, 2013) because of its availability.

3.2.3 Global Historical Climate Network (GHCN)

In this study, we also use individual station data derived from the nearest climate station in the GHCN database to each modern pollen site. There was no limit placed on the search radius for the nearest station but 10 modern pollen sites were outside of a 5 degree search radius. The GHCN product incorporates raw climate station data after it has been adjusted for non-meteorological factors, such as changes in station location and time of observation. This type of homogenization of climate records is described in Brohan *et al.* (2006) and Menne *et al.* (2009). In total, we use 2,717 out of 7,279 stations in the GHCN database. The R package RGhcnV3 was used to download and process the data, although some additional post-processing of the data was performed in R to format the data and calculate the mean July temperature for the 1961–1990 baseline at each station.

3.2.4 Berkeley Earth Surface Temperature (BESTstn/gr1/2)

The BEST individual station data and gridded products were also used because this data set has wider coverage of North America (∼13,700 stations) especially in many high latitude regions (Rohde *et al.*, 2012; Rohde *et al.*, 2013). Data for the nearest station to each modern pollen site was downloaded and processed using the R package BerkeleyEarth and the July average temperature for 1961–1990 was calculated for all stations in North America. For the 1° gridded product we first downloaded the anomaly series and then re-added the surface baseline climatology provided with each file. The base period of 1979-2000 was tested only for BEST gridded (BESTgr2) not for BESTstn.

3.2.5 World Climate (WCLIM)

WCLIM is a very high resolution gridded climate data with a resolution of 30 arc-seconds (∼1 km or ∼0.09°), which is approximately 20 times the spatial resolution of New *et al.* (2002). The data set and methods used to develop it are described in Hijmans *et al.* (2005). The July mean temperature grid is derived from GHCN and other station data with a period that is representative of 1950–2000 though is largely comparable to 1961–1990 (Hijmans *et al.*, 2005) (Figure 5C). The effect of spatial resolution on modern calibration data sets was tested by resampling the WCLIM grid to various lower resolutions (0.5°, 1° and 2.5°). The results of this down-sampling resolution test are discussed later in this paper. The high spatial autocorrelation of this data set means that the climate is relatively stable spatially (Hijmans *et al.*, 2005).

3.2.6 North American regional reanalysis (NARR)

The NARR (Mesinger *et al.*, 2006) is a high resolution (∼32 km or ∼0.3°) gridded atmospheric reanalysis data set at 3 hour time steps and 45 atmospheric levels. Over North America there is a significant improvement in the accuracy of 2 m temperature and 10m winds relative to the NCEP R2 reanalysis product (Mesinger *et al.*, 2006). The July long term mean temperature was used, which covers the period 1979–2000. It should be noted that the NARR data set covers 1979 to present with the long term mean for 1979–2000, however, we wanted to compare a regional reanalysis to a global one such as NNR, and the NARR has the same spatial domain as our study area. However, there are some issues with NARR such as a warm bias as noted in Royer and Poirier (2010), however nonetheless this data set was still tested.

3.2.7 NCEP/NCAR reanalysis (NNR1, NNR2)

NNR (Kalnay *et al.*, 1996; Kistler *et al.*, 2001) is a gridded climate data set at 2.5° resolution derived from historical observations, re-processed by a model with its atmospheric physics fixed or frozen over the reanalysis period (1948 to present). The model resolution is 210 km with 28 vertical levels. Two 30 year mean periods were used; 1981–2010 and 1961–1990 which are calculated using the Max Plank Institute (MPI) Climate Data Operators (CDO) command-line software. The use of a warmer base period (1981-2010) compared to a colder base period (1961-1990) provides an interesting test. We also tested the 1979–2000 period to to compare with the NARR and ERA-Interim data sets which have the same base period.

3.2.8 ECMWF ERA-40 (ERA40) & ERA-Interim (ERAI) reanalyses

The ERA40 Reanalysis product (Uppala *et al.*, 2005) has the same resolution (2.5°) as the NNR, with periods of 1961–1990 and 1979–2000. The ERAI reanalysis product (Dee *et al.*, 2011) covers the period 1979–2000 associated with a resolution of ∼0.7°. The ERA40 product is available for the period of 1958–2002 where the 30-year period (1961–1990) and the ERAI period (1979–2000) are used for consistency. The atmospheric model used in the ERAI data set has the same number of atmospheric levels (60) as the ERA40 product.

3.3 Results

3.3.1 Comparison of reconstructions

July temperature (TJUL) reconstructions over the past 2,000 years using all modern data sets are largely in agreement with one another on the major features including a Roman Warm Period (RWP), Dark Ages Cold Period (DAC), Medieval Warm Period (MWP), Little Ice Age (LIA) and rapid anthropogenic warming since 1900 CE (Figure 6). The agreement between the various reconstructions is strong throughout the record with the notable exception being during the period of 1050–1550 CE where reconstructions show considerable differences in amplitude (e.g. range of 0.25°C at 1250 CE). Comparison between all reconstructions with a common base period of 1961-1990 and 1979-2000 show similar variability (Figure 6B,C). Moreover, differences between the station (e.g. CRU) and reanalysis based reconstructions suggest the reanalysis-based reconstructions exhibit a warm bias, especially during the period of 1050-1550 CE (Figure 7).

A more detailed spatial analysis for the period of 1050–1550 CE reveals similarities and discrepancies between reconstructed anomaly maps. These can be classified into four groups: group 1 includes WHIT, NEW and WCLIM (Figure 8A,B,D); group 2 comprises BestGr1,Gr2, ERA40 and ERAI (Figure 8E,I,F,J); group 3 for NNR1/NNR2 (Figure 8H,L) and group 4 with BESTstn and GHCN (Figure 8G,C). The NARR reconstruction is the only one which is largely dissimilar from each of the four groups mentioned above (Figure 8K). These differences between groups are likely to explain the large amplitude of variability observed during the period of 1050–1550 CE.

Regional spatial patterns for group 1 show strong positive TJUL anomalies over Labrador and the Western Arctic, part of the BC/Alberta Rockies and strong negative anomalies over the Southwest USA and California (Figure 8A,B,D). Group 2 is similar to group 1 except that the strong positive anomalies cover the Maritimes and far Northwestern North America (near the BC/Alaska border) as opposed to the Western Arctic (Figure 8E,I,F,J) (except in panel F). Group 3 reveals slightly stronger positive anomalies in the Northwestern North America and along the Labrador coast and the Canadian Maritime regions using NNR2 1981–2010 (Figure 8L) as compared to NNR1 1961–1990 (Figure 8H). In group 4, larger positive anomalies are reconstructed using BESTstn along the southern BC/Alberta Rockies, the Arctic and coastal Labrador (Figure 8G) as compared to the GHCN (Figure 8C). The NARR reconstruction show overall TJUL anomalies which are inconsistent with the other groups (Figure 8K).

3.3.2 Comparison of reconstructed modern versus observed TJUL

Reconstructed modern TJUL using all modern data sets shows that reconstructed and observed values are strongly related (Table 3). Coefficients of determination range from 0.93 to 0.98, with a mean value of 0.97, indicating that reconstructed TJUL explains 93% to 98% of the variance in observed TJUL. Ranked according to their mean absolute difference (MAD), we show that the NNR1 (1961–1990) modern reconstruction has the lowest MAD with NARR having the highest and WHIT roughly near the median value (Table 3). The spatial patterns of reconstructed modern (1950 CE) TJUL anomalies for NNR1, WHIT and NARR show some regional variability with both NNR1 and Whitmore showing similar ranges of anomalies as compared to NARR (Figure 63).

3.3.3 Comparison of TJUL by ecoregion

Overall, the BESTstn and GHCN modern TJUL both have the lowest average standard deviation (Table 4). NNR1 and NNR2 both show slightly larger negative differences in standard deviations as compared to the WCLIM, WHIT and NEW data sets. As expected, NARR reveals the highest average standard deviation (Table 4). Regionally, results by ecoregion (Figure 95 in Appendix E) show that NARR has greater variability than all other modern data sets across 9 of 10 ecoregions (Table 4). While there are similarities between WHIT, NEW and WCLIM, some differences exist for different ecoregions (e.g. mountain vegetation and prairies). Similarly the GHCN and BESTstn also show some discrepancies in some ecoregions (e.g. forest-tundra). The reanalysis products (NNR1, NNR2) show systematic biases, especially in the desert region.

3.4 Discussion

3.4.1 Differences and similarities between the reconstructions

Reconstructions using different modern TJUL data sets show the highest range of spread between 1050–1550 CE (Figure 6A,B). One possible explanation for this variability is the regional cold bias in absolute air temperatures over Northern Canada in several reanalysis products such as NCEP (Cowtan and Way, 2013; Lindsay *et al.*, 2014). Recent research has suggested potential issues with the Arctic temperatures

below the 500mb level in the ERA40 after 1997 (Screen and Simmonds, 2011) and this could be part of the explanation for the smoothing and warm bias during the early onset of the LIA. For example, the reconstruction of TJUL using ERA40 for 1050–1550 CE is the largest outlier between the station and reanalysis-based reconstructions (Figure 6A,B). Moreover, the ERA40 reconstruction appears too warm and smooth during this particular period suggesting a much longer MWP and much shorter and recent LIA. Other studies have pointed to cold biases when using certain gridded climate data sets in high latitude regions of North America due to complex topography and local-scale variability (Atkinson and Gajewski, 2002; Fortin and Gajewski, 2012). We suggest that during the LIA (coldest of the last 2,000 years) a fossil pollen site could point to a modern pollen site that is cold-biased depending on the modern TJUL data used. As a result there is an artificial warm bias in the computed anomaly value (anomaly = reconstructed minus modern value). Therefore if the reconstructed TJUL value at a fossil site is 5°C and the modern TJUL value is 4°C, the resulting anomaly is +1°C. However, if the modern TJUL value is slightly colder than 4°C (e.g. 3°C), then the resulting anomaly is +2°C rather than +1°C.

Similarly to ERA40, NNR1 and NNR2 also show systematic warmer and smoother reconstructions compared to those using station data (Figure 6 and Figure 7). This may be due to the lower resolution (i.e. 2.5°) of NNR1/NNR2 and ERA40 grid point value assigned to each modern pollen site. This can become an important issue in complex terrain such as in the western USA or where stations are sparse. Therefore some type of statistical downscaling approach could be implemented in such cases (e.g. Cavazos and Hewitson, 2005). Conversely, the use of a higher resolution reanalysis product (e.g. NARR with ∼32 km resolution) may yield a more similar reconstruction to the station based time-series. The NARR reconstruction is even more warm-biased than the other reanalysis-based reconstructions (anomaly of +0.07°C at 1950 CE). We also note that for several of the reanalyses products the use of a warmer base period (e.g. NNR2, 1981–2010 relative to NNR1, 1961–1990) produces a warmer reconstruction.

The WCLIM, which is physically modeled from station data produces similar results to station-based only (e.g. WHIT, NEW) reconstructions (Figure 6). This would be expected as similar interpolation methods (e.g. splines) are used. However, this ultra-high resolution (1 km) data set does not provide any more information since the location of the modern pollen have a lower spatial resolution compared to the WCLIM grid resolution. For example, our analysis of intersecting the modern pollen site to the WCLIM grid cell caused ∼14 modern pollen sites to be assigned

NA values, because they were outside the WCLIM grid. In order to avoid this problem, we converted the 1 km raster to a point shape file where each point is located at the center of each raster grid cell. Then each modern pollen site is assigned the TJUL value of the nearest grid point with no weighting scheme used for surrounding stations. Nearly identical results were obtained at 1 km using the raster intersect operation versus the nearest grid point method. Although this was not a problem using the WHIT (Whitmore *et al.*, 2005) modern pollen site distribution, we note that this could represent a challenge with different site distributions. Since the WCLIM reconstruction is slightly warmer than that of WHIT, we tested whether a lower resolution modern climate data set produces a warm-biased reconstruction. We re-sampled WCLIM at different resolutions (1 km, 0.5°, 1° and 2.5°) (Figure 9). Results show that in the case of WCLIM, a lower resolution resulted in a warm-biased reconstruction with the exception of the 1 km reconstruction, although this may not apply to other very high resolution climate data sets. We suggest that a more in-depth analysis would be required before using any very high resolution modern climate data set in a paleoclimate reconstruction.

The station-based reconstruction of NEW (New *et al.*, 2002) is very similar to WHIT where only slight differences are likely due to the lapse-rate corrections applied in the WHIT. The GHCN reconstruction is also comparable to the CRU-based data set (WHIT, NEW) except for a slightly higher amplitude on multi-centennial scales which is probably due to choosing the nearest GHCN station value without any corrections for lapse rate (Figure 6). For example, in areas of complex topography (e.g. Western North America) or sparse station data (e.g. in the Arctic), the nearest station could be further away or at a higher (lower) elevation compared to the modern pollen site resulting in a colder (warmer) temperature value. In contrast the BESTstn reconstruction is warmer than any other station-based reconstructions, although not as warm as the reanalysis-based reconstructions or even BESTgr1/BESTgr2 (Figure 6 and Figure 7). Interestingly, the larger number of stations over North America in the BEST data set (∼13,700) does not provide an improved reconstruction as compared to GHCN (∼2,700).

3.5 Summary and Conclusions

Five recommendations are presented for large-scale North American pollen-based paleoclimate reconstructions:

1. Reanalysis-based data sets (e.g. NNR1, NNR2, ERA40, ERAI) are not recommended for continental-scale paleoclimate reconstructions as they result in smoothed and warm-biased reconstructions, especially during the period of 1050–1550 CE. In addition, the use of NARR remains outside the range of the warmest station-based reconstructions.

2. The use of a very high spatial resolution modern data set such as the WCLIM 1 km grid is only useful if the precision of the location of the modern pollen sites is equal or better than the precision of the WCLIM grid resolution. Therefore, if the precision of the location of the modern pollen sites is lower than the grid resolution of WCLIM, it will not provide any additional information and can be resampled to an appropriate grid size to reduce computational needs.

3. Conversely, an ultra-high resolution modern climate data set such as WCLIM may be useful approach for a local or small-scale reconstruction; however some type of statistical downscaling approach may be needed to ensure the reliability of surface or near-surface modern climate conditions.

4. When using the nearest modern climate station data approach, a much larger number of well distributed climate stations (e.g. BEST compared to GHCN) along with a spatially heterogeneous distribution of modern pollen sites in a reconstruction, may not provide any additional information for the reconstruction.

5. The use of the lapse rate corrected CRU-based WHIT modern climate data for pollen-based climate reconstructions is justified as compared to other station-based data sets, ultra-high resolution data sets and is superior to the use of reanalysis-based products. However, regional variation in lapse rates must be considered during the application of this type of correction to station-based data sets.

This study provides important results in light of the recent development of modern climate data sets and research tools currently available to climate scientists, especially for the paleoclimate scientific community. The choice of the modern calibration data set in paleoclimate research becomes an important decision as it can lead to conflicting results and interpretations of past climate change. The analysis presented in this study does not solely apply to pollen-based reconstructions but also should be considered when using other independent proxies and/or multi-proxies in reconstructing past climates.

Acknowledgments

This research was supported by grants from the Natural Sciences and Engineering Council of Canada (NSERC) to AEV. We thank the contributors for their available modern and fossil pollen data sets and modern climate data.

Table 2: Summary of all modern TJUL data sets tested for TJUL reconstruction of the last 2,000 years. Data references (from top row to bottom row) are: Whitmore *et al.* (2005), Lawrimore *et al.* (2011), Rohde *et al.* (2012), Hijmans *et al.* (2005), New *et al.* (2002), Rohde *et al.* (2013), Messinger *et al.* (2006), Kistler *et al.* (2001), Uppala *et al.* (2005) and Dee *et al.* (2011).

Data set	Type	Resolution	Period
Whitmore (WHIT)	Station	4,833 sites	1961–1990
Global Historical Climate Network (GHCN)	Station	∼2,700 stations	1961–1990
Berkeley Earth Surface Temperature (BESTstn)	Station	∼13,700 stations	1961–1990
World Climate (WCLIM)	Physically modeled	∼0.009°	1950–2000
New *et al.*(NEW)	Physically modeled	∼0.17°	1961–1990
BEST gridded (BESTgr1, BESTgr2)	Physically modeled	1°	1961–1990 and 1979–2000
North American Regional Reanalysis (NARR)	Reanalysis	∼0.3°	1979–2000
NCEP/NCAR reanalysis (NNR1, NNR2)	Reanalysis	2.5°	1961–1990 and 1981–2010
ECMWF European reanalysis (ERA40)	Reanalysis	2.5°	1961–1990 and 1979–2000
ECMWF European reanalysis interim (ERAI)	Reanalysis	∼0.7°	1979–2000

Table 3: Statistics for observed TJUL versus reconstructed modern (1950 CE) TJUL. SE_Slope is standard error of the slope, SE_Res is standard error of residuals and MAD is mean absolute difference.

data set	R^2	N	SE_Slope	SE_Res	St_Dev	MAD
NNR1	0.98	455	0.006	0.61	0.58	0.19
NNR2	0.98	455	0.007	0.62	0.59	0.2
BESTgr1	0.98	455	0.006	0.63	0.59	0.21
ERAI	0.98	455	0.006	0.63	0.6	0.22
NEW	0.98	455	0.007	0.7	0.66	0.24
BESTst	0.98	455	0.007	0.72	0.68	0.26
WCLIM	0.98	455	0.007	0.75	0.71	0.25
WHIT	0.97	455	0.008	0.81	0.76	0.27
GHCN	0.96	455	0.009	0.85	0.81	0.29
ERA40	0.94	455	0.115	1.19	1.17	0.32
NARR	0.93	455	0.012	1.35	1.32	0.39

Table 4: Difference in standard deviation by ecoregion of selected modern TJUL compared to average standard deviation by ecoregion of all modern TJUL (difference from last column in table).

ECOREGION	WHIT	BESTst	GHCN	NEW	WCLIM	NNR1	NNR2	NARR	ST_Dev All Mod
Arctic	-0.66	0.16	-0.8	-0.65	-0.68	0.27	-0.16	2.51	2.47
Boreal Forest	-0.24	-0.24	-0.26	-0.2	-0.27	0.49	-0.14	0.86	2.81
Coastal	-0.48	-0.04	-0.06	-0.33	-0.2	0.27	0.14	0.72	2.51
Conifer/Hardwood	-0.08	0.09	-0.19	-0.08	-0.12	0.04	-0.11	0.44	1.48
Deciduous Forest	0.01	0.08	0.02	-0.09	0.05	-0.31	-0.37	0.63	3.41
Desert	0.38	-0.09	0.84	0.43	0.64	-1.63	-1.82	1.26	4.91
Forest-Tundra	-0.6	-0.05	1.15	-0.6	-0.73	-0.51	-0.46	1.8	2.50
Mediterranian	0	0.18	0.05	-0.18	-0.16	-0.07	0.18	0.32	0.32
Mountain Veg	-0.03	-0.23	-0.61	-0.04	0.58	-0.89	-0.78	2.01	3.59
Prairies	0.22	0.06	-0.23	0.01	0.06	-0.22	-0.24	0.35	2.74
No data	-0.25	0.03	0.19	-0.18	0.11	-0.37	-0.49	0.96	2.77
AVERAGE	**-0.16**	**0**	**0.01**	**-0.17**	**-0.07**	**-0.27**	**-0.39**	**1.05**	

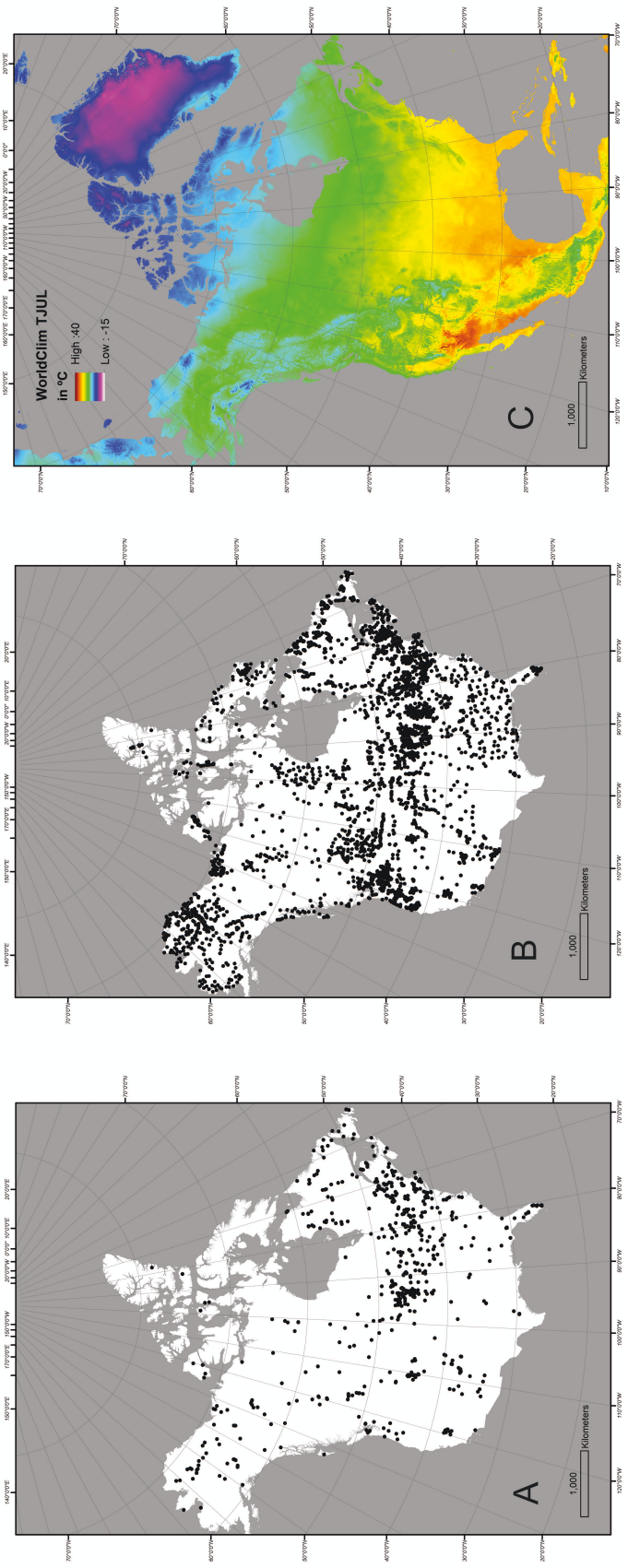


Figure 5: Distribution of fossil (A) and modern (B) pollen sites and modern TJUL from WorldClim (C).

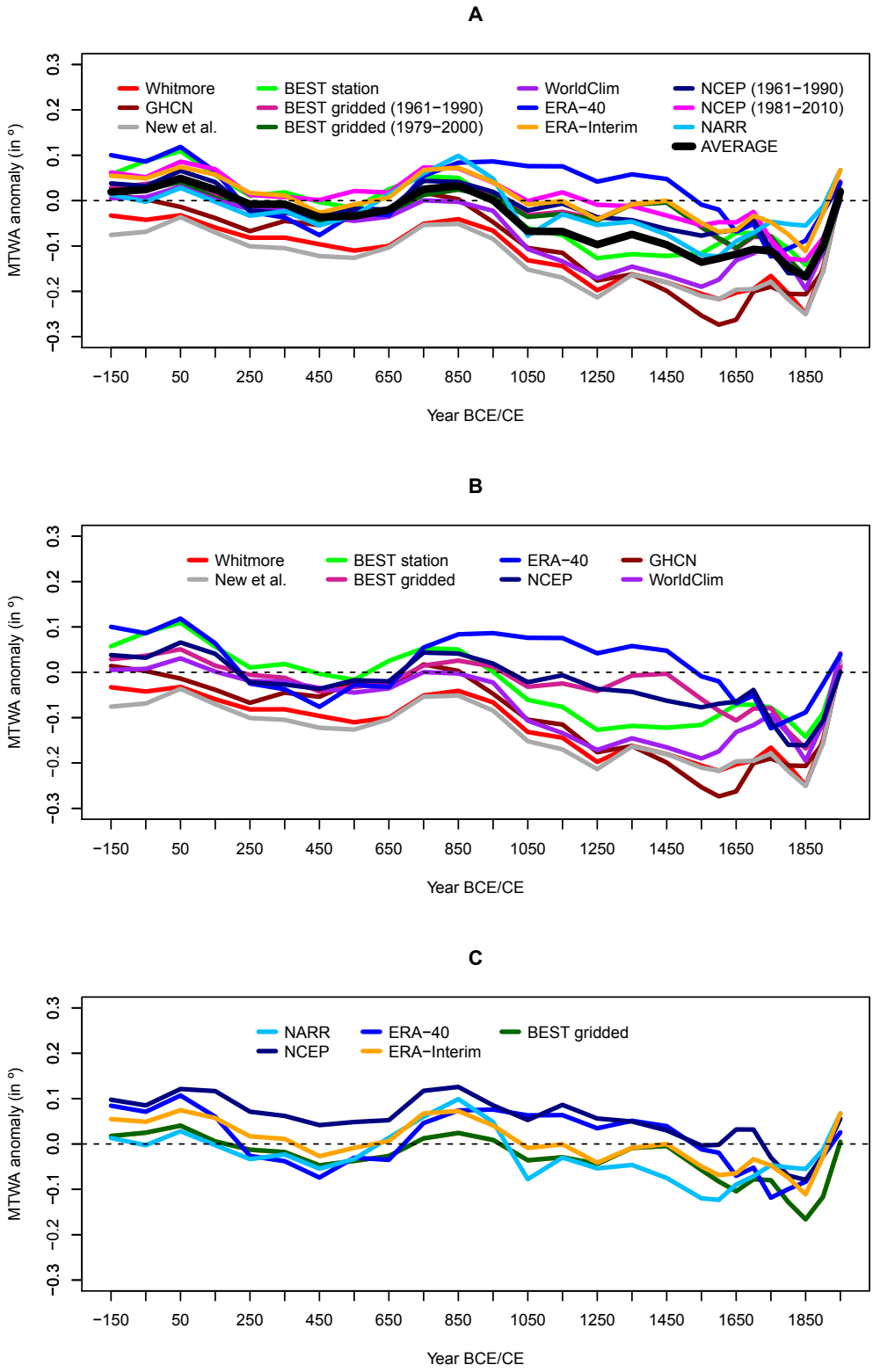


Figure 6: Reconstructions of MTWA/TJUL anomalies (in °C) for the last 2,000 years using all modern data sets (A), 1961–1990 base period data sets (B) and 1979–2000 base period data sets (C).

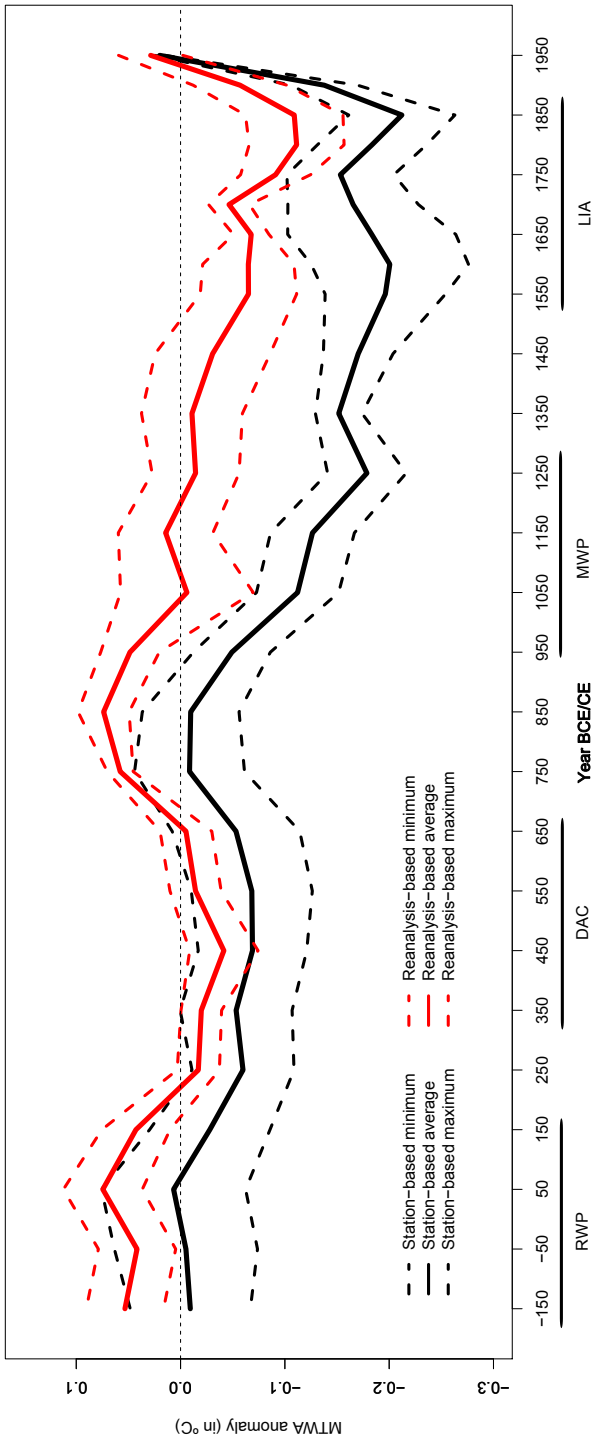


Figure 7: Comparison of the range of station-based (black) versus reanalysis-based (red) MTWA/TJUL reconstructions for the last 2,000 years. The approximate age ranges for the RWP, DAC, MWP and LIA periods are indicated.

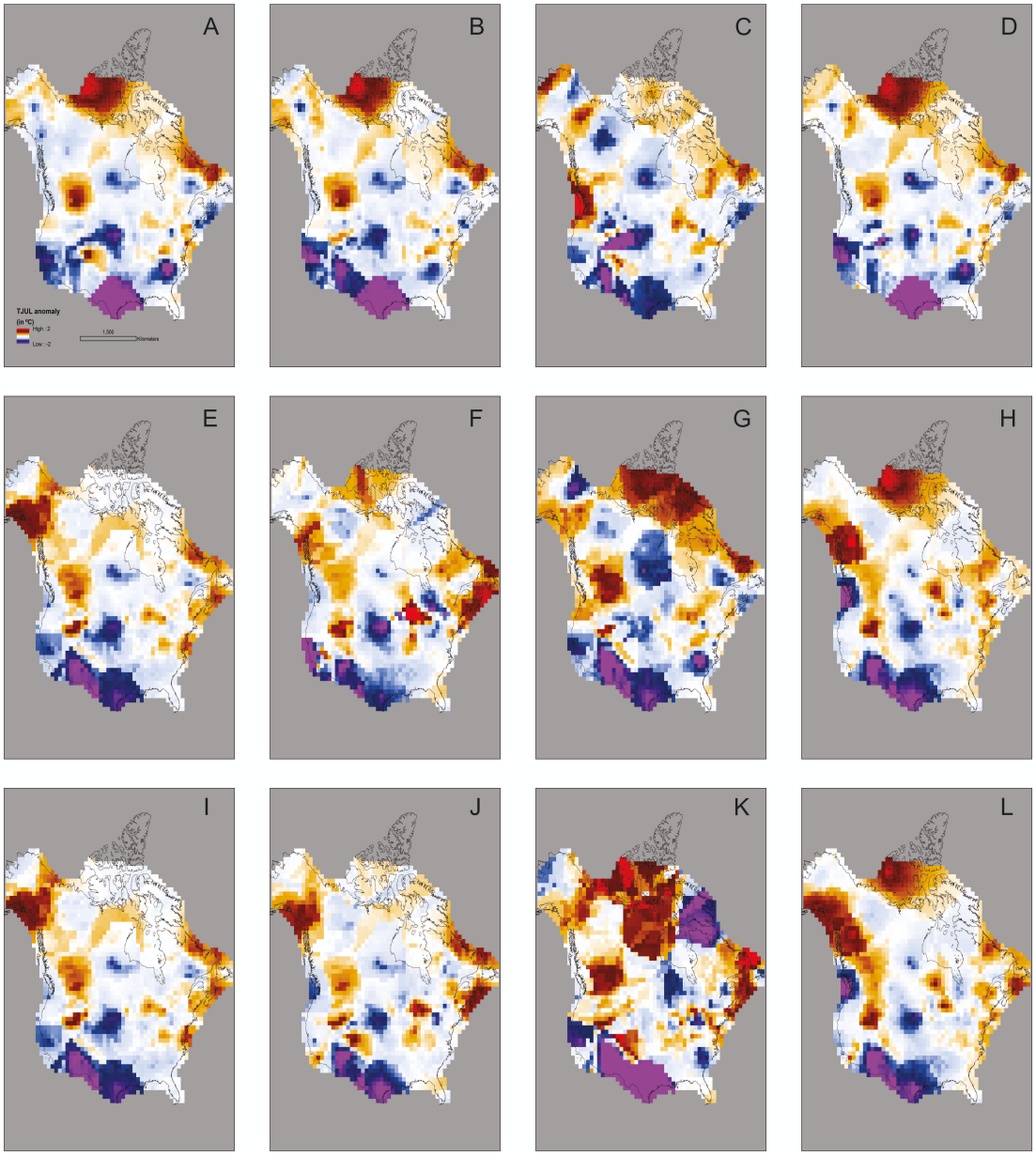


Figure 8: Reconstructed anomaly map (gridded at 100 km) for the average of 1050–1550 CE using different modern TJUL data sets as follows: Whitmore 1961–1990 (A), New *et al.* 1961–1990 (B), GHCN 1961–1990 (C), WorldClim 1950–2000 (D), BEST gridded 1961–1990 (E), BEST gridded 1979–2000 (F), BEST station 1961–1990 (G), NCEP 1961–1990 (H), BEST gridded 1979–2000 (I), ERA-Interim 1979–2000 (J), NARR 1979–2000 (K) and NCEP 1981–2010 (L).

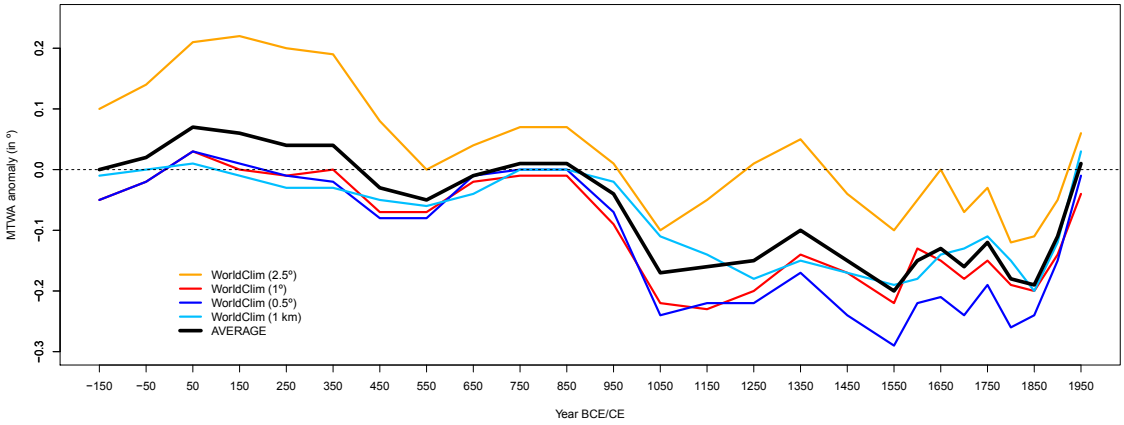


Figure 9: Reconstruction of MTWA/TJUL anomalies (in °C) for the last 2,000 years using the WorldClim TJUL modern data (1 km) gridded at resolutions of 0.5°, 1° and 2.5°.

References

Atkinson, D.E. and K. Gajewski (2002). High-resolution estimation of summer surface air temperature in the Canadian Arctic Archipelago. *Journal of Climate* **15**(24).

Bartlein, P.J., S.J. Harrison, S. Brewer, S. Connor, B.A.S. Davis, K. Gajewski, J. Guiot, T.I. Harrison-Prentice, A. Henderson, O. Peyron, I.C. Prentice, M. Scholze, H. Seppä, B. Shuman, S. Sugita, R.S. Thompson, A.E. Viau, J. Williams, and H. Wu (2011). Pollen-based continental climate reconstructions at 6 and 21 ka: a global synthesis. *Climate Dynamics* **37**(3-4), 775–802.

Berkeley Earth Surface Temperature (2013). *Berkeley Earth*. url: http://www.berkeleyearth.org.

Birks, H.J.B., O. Heiri, H. Seppå, and A.E. Bjune (2010). Strengths and weaknesses of quantitative climate reconstructions based on late-quaternary biological proxies. *The Open Ecology Journal* **3**, 68–110.

Braak, C.F.J. ter and S. Juggins (1993). Weighted averaging partial least squares regression (WA-PLS): an improved method for reconstructing environmental variables from species assemblages. *Hydrobiologia* **269**(1), 485–502.

Bradley, R.S., M.E. Mann, and M.K. Hughes (1999). Northern hemisphere temperatures during the past millennium: inferences, uncertainties, and limitations. *Geophysical Research Letters* **26**(6), 759–762.

Brohan, P., J.J. Kennedy, I. Harris, S.F.B. Tett, and P.D. Jones (2006). Uncertainty estimates in regional and global observed temperature changes: A new data set from 1850. *Journal of Geophysical Research: Atmospheres (1984–2012)* **111**(D12).

Cavazos, T. and B.C. Hewitson (2005). Performance of NCEP-NCAR reanalysis variables in statistical downscaling of daily precipitation. *Climate Research* **28**, 95–107.

Cowtan, K. and R.G. Way (2013). Coverage bias in the HadCRUT4 temperature series and its impact on recent temperature trends. *Quarterly Journal of the Royal Meteorological Society*.

Dee, D.P., S.M. Uppala, A.J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M.A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A.C.M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A.J. Geer, L. Haimberger, S.B. Healy, H. Hersbach, E. Hólm, L. Isaksen, P. Källberg, M. Köhler, M. Matricardi, A.P. McNally, B.M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and Vitart F. (2011). The ERA-Interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society* **137**(656), 553–597.

Fortin, M.-C. and K. Gajewski (2012). Potential problems with the use of gridded climate data in regional quantitative paleoenvironmental studies from data-poor regions. *Journal of Paleolimnology* **48**(3), 641–650.

Gonzales, L.M., J.W. Williams, and E.C. Grimm (2009). Expanded response-surfaces: a new method to reconstruct paleoclimates from fossil pollen assemblages that lack modern analogues. *Quaternary Science Reviews* **28**(27), 3315–3332.

Harris, I., P.D. Jones, T.J. Osborn, and D.H. Lister (2013). Updated high-resolution grids of monthly climatic observations–the CRU TS3. 10 Dataset. *International Journal of Climatology*.

Hijmans, R.J., S.E. Cameron, J.L. Parra, P. G. Jones, and A. Jarvis (2005). Very high resolution interpolated climate surfaces for global land areas. *International journal of climatology* **25**(15), 1965–1978.

Kalnay, E., M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K.C. Mo, C. Roperlewski, J. Wang, A. Leetmaa, R. Reynolds, R. Jenne, and D. Joseph (1996). The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American meteorological Society* **77**(3), 437–471.

Kerwin, M.W., J.T. Overpeck, R.S. Webb, and K.H. Anderson (2004). Pollen-based summer temperature reconstructions for the eastern Canadian boreal forest, subarctic, and Arctic. *Quaternary Science Reviews* **23**(18), 1901–1924.

Kistler, R., E. Kalnay, Collins W., S. Suranjana, G. White, J. Woollen, Muthuvel C., W. Ebisuzaki, M. Kanamitsu, V. Kousky, H. van den Dool, R. Jenne, and M. Fiorino (2001). The NCEP-NCAR 50-year reanalysis: Monthly means CD-ROM and documentation. *Bulletin of the American Meteorological society* **82**(2), 247–267.

Lindsay, R., M. Wensnahan, A. Schweiger, and J. Zhang (2014). Evaluation of seven different atmospheric reanalysis products in the Arctic. *Journal of Climate*(2014).

Menne, M.J., C.N. Williams Jr, and R.S. Vose (2009). The US Historical Climatology Network monthly temperature data, version 2. *Bulletin of the American Meteorological Society* **90**(7).

Mesinger, F., E. DiMego G.and Kalnay, K. Mitchell, P.C. Shafran, W. Ebisuzaki, D. Jovic, J. Woollen, E. Rogers, E.H. Berbery, M.B. Ek, Y. Fan, R. Grumbine, W. Higgins, Hong Li, Y. Lin, G. Manikin, D. Parrish, and W. Shi (2006). North American regional reanalysis. *Bulletin of the American Meteorological Society* **87**(3), 343–360.

New, M., D. Lister, M. Hulme, and I. Makin (2002). A high-resolution data set of surface climate over global land areas. *Climate research* **21**(1), 1–25.

Overpeck, J.T., T. Webb III, and I.C. Prentice (1985). Quantitative interpretation of fossil pollen spectra: dissimilarity coefficients and the method of modern analogs. *Quaternary Research* **23**(1), 87–108.

Rohde, R., R.A. Muller, R. Jacobsen, E. Muller, S. Perlmutter, A. Rosenfeld, J. Wurtele, D. Groom, and C. Wickham (2012). A New Estimate of the average Earth surface land temperature spanning 1753 to 2011, Geoinfor. *Geostat: An Overview* **1**(1).

Rohde, R., R.A. Muller, R. Jacobsen, S. Perlmutter, A. Rosenfeld, J. Wurtele, J. Curry, C. Wickham, and S. Mosher (2013). Berkeley Earth Temperature Averaging Process. *Geoinfor Geostat: An Overview* **1**(2).

Royer, A. and S. Poirier (2010). Surface temperature spatial and temporal variations in North America from homogenized satellite SMMR-SSM/I microwave measurements and reanalysis for 1979–2008. *Journal of Geophysical Research: Atmospheres (1984–2012)* **115**(D8).

Screen, J.A. and I. Simmonds (2011). Erroneous Arctic Temperature Trends in the ERA-40 Reanalysis: A Closer Look. *Journal of Climate* **24**(10).

Spenke, M., C. Beilken, and T. Berlage (1994). Modern climatic data for paleoenvironmental interpretations. *American Quaternary Association Program and Abstracts of the 13th Biennial Meeting, 19-22 June*, 197.

Trouet, V. and G.J. Van Oldenborgh (2013). KNMI Climate Explorer: a web-based research tool for high-resolution paleoclimatology. *Tree-Ring Research* **69**(1), 3–13.

Uppala, S.M., P.W. Källberg, A.J. Simmons, U. Andrae, V. Bechtold, M. Fiorino, J.K. Gibson, J. Haseler, A. Hernandez, G.A. Kelly, X. Li, K. Onogi, S. Saarinen, N. Sokka, R. P. Allan, E. Andersson, K. Arpe, M. A. Balmaseda, A. C. M. Beljaars, L. Van De Berg, J. Bidlot, N. Bormann, S. Caires, F. Chevallier, A. Dethof, M. Dragosavac, M. Fisher, M. Fuentes, S. Hagemann, E. Hólm, B. J. Hoskins, L.

Isaksen, P. A. E. M. Janssen, R. Jenne, A. P. Mcnally, J.-F. Mahfouf, J.-J. Morcrette, N. A. Rayner, R. W. Saunders, P. Simon, A. Sterl, K. E. Trenberth, A. Untch, D. Vasiljevic, P. Viterbo, and J. Woollen (2005). The ERA-40 re-analysis. *Quarterly Journal of the Royal Meteorological Society* **131**(612), 2961–3012.

Viau, A., M. Ladd, and K. Gajewski (2012). The climate of North America during the past 2000 years reconstructed from pollen data. *Global and Planetary Change* **84**, 75–83.

Whitmore, J., K. Gajewski, M.C. Sawada, J.W. Williams, B. Shuman, P.J. Bartlein, T. Minckley, A.E. Viau, T. Webb III, S. Shafer, P. Anderson, and L. Brubaker (2005). Modern pollen data from North America and Greenland for multi-scale paleoenvironmental applications. *Quaternary Science Reviews* **24**(16), 1828–1848.

Chapter 4

Variations in precipitation over North America over the past 2,000 years

Ladd, M.[1]*, Viau, A.E [1], Way, R.G.[2], Gajewski, K.[1], Sawada, M.C.[3] *corresponding author

[1] Laboratory for Paleoclimatology and Climatology, Department of Geography, University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5

[2] Department of Geography, University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5

[3] Laboratory for Applied Geomatics and GIS Science (LAGGIS), Department of Geography, University of Ottawa, Ottawa, Ontario, Canada, K1N 6N5

Abstract

Century-scale filtered pollen-based reconstructions of total annual precipitation (ANNP) for the last 2,000 years are regionally more spatially coherent than unfiltered reconstructions, allowing for better interpretation of atmospheric circulation patterns responsible for past hydroclimate variability across North America. La Nina, positive North Atlantic Oscillation (+NAO) and positive Atlantic Multidecadal Oscillation (+AMO) state-like dominated both the Medieval Warm Period (MWP) and Roman Warm Period (RWP), although the MWP was generally drier. In contrast, the Dark Ages Cold Period (DAC) period was likely dominated by El Nino, negative NAO and negative Atlantic Multidecadal Oscillation (AMO) state-like circulation. The hydroclimate changes during the Little Ice Age (LIA) is more complex to interpret,

although this study shows that opposing patterns of wet/dry conditions in the western and eastern boreal region suggest a positive phase of the Pacific North American (+PNA) pattern associated with a dominant El Nino state-like circulation is comparable to the DAC. Minimum solar and high volcanic activity may have contributed to alter the circulation patterns and thereby lead to more complex hydroclimate regional patterns during the LIA. The use of Local Indices of Spatial Association (LISA) distinguishes regional hydroclimate patterns from local site-specific conditions across North America during the past two millennia.

Keywords:

North America, hydro-climate, spatial filtering, late-Holocene, pollen reconstructions, Little Ice Age, Medieval Warm Period, LISA

4.1 Introduction

Temperature reconstructions, from regional to global scales, show evidence of unprecedented warming over the past century relative to the past 2,000 years (e.g. PAGES 2k Consortium, 2013). A series of high-resolution reconstructions of the past 2,000 years with annual to multi-decadal resolution and using a multi-proxy approach reveal hemispheric low-frequency temperature changes at century scales [e.g. Little Ice Age (LIA), Medieval Climate Anomaly (MCA), Dark Ages Cool Period (DCA) and Roman Warm Period (RWP)] that are superimposed on a millennial-scale orbital cooling (IPCC, 2013).

One of the major challenges in deriving these paleoclimate reconstructions is the decreasing site density through time, especially during the first millennium of the Common Era (IPCC, 2007; IPCC, 2013). Reconstructions using tree rings, ice cores, speleothems, boreholes, corals and historical/documentary records provide invaluable information on historical climate changes throughout the past 2,000 years (PAGES 2k Consortium, 2013). Although these reconstructions show similar trends during the past 2,000 years, there remain uncertainties relating to the range of variability on lower-frequency climate variability, sensitivity to the environmental changes and methodological approaches (Wanner *et al.*, 2008; IPCC, 2013). For example, it is not clear which tree-ring reconstruction methods are most successful in retaining low-frequency climate variability (Wilson *et al.*, 2007; Esper *et al.*, 2012).

In this study, we investigate centennial-scale hydro-climate conditions in North America during the past 2,000 years using pollen records. Pollen records offer several advantages over other available proxies in the region, including (1) a large network

of fossil sites across all of North America; (2) temporal representation of the entire Holocene with little loss in sample density for the past 10 ka and; (3) preservation of climate variability across a wide range of frequencies, from millennial through to at least decadal, depending on the resolution of the analysis within the sedimentary sequence.

4.2 Data and Methods

4.2.1 Pollen data

Two pollen data sets are used: a modern set for calibration and a fossil data set for reconstruction. The data of modern pollen distribution and associated climatic information used for calibration purposes is Whitmore *et al.* (2005). This modern pollen database consists of ∼4,800 sites that are well distributed over North America (Figure 10B). Fossil pollen records were extracted from the Neotoma database (Grimm, 2011). There are ∼750 fossil pollen sites available across North America spanning parts of the last 21,000 years. The fossil pollen distribution covers most ecoregions of North America (Figure 11) with the Northeastern United States being most densely sampled (Figure 10A). Modern annual precipitation for 1961–1990 (the standard base period) using the WorldClim data set (Hijmans *et al.*, 2005)(Figure 10C) was reproduced using the Whitmore *et al.* (2005) modern pollen database.

4.2.2 Modern Analogue Technique (MAT)

Pollen data has been used successfully to reconstruct past climates on various temporal and spatial scales using the Modern Analogue Technique (MAT) (e.g. Bartlein *et al.*, 2011; Cheddadi *et al.*, 1997; Gajewski, 1987; Gajewski, 1988; Gajewski, 1993; Gajewski, 2008; Gajewski and Viau, 2011; Overpeck *et al.*, 1985; Sawada *et al.*, 2004; Trouet *et al.*, 2013; Viau *et al.*, 2006; Viau *et al.*, 2012; Williams and Shuman, 2008). We use a pollen sum of 63 taxa (Table 5) which has been shown to provide accurate results at regional scales in North America (Williams and Shuman, 2008; Viau *et al.*, 2012). We use the best analogue for each fossil pollen spectrum to assign past climates to the fossil data. Currently, there is no agreement on the number of analogues to retain when applying the MAT, however previous work has shown that the use of the best analogue provides more reasonable results, especially in boreal regions (Viau *et al.*, 2006; Viau *et al.*, 2008; Viau and Gajewski, 2009; Viau *et al.*, 2012) than using an average of the top several analogues (Juggins and Birks, 2012; Williams and Shuman, 2008). The R package MATTOOLS (Sawada, 2006) was used

to find the most similar modern analogue to fossil samples down-core. The climate variable of interest, in this study, mean total annual precipitation (ANNP) was then extracted and mapped. The anomalies are calculated as reconstructed values minus modern values. Scatter plots of reconstructed modern (1950 CE) ANNP compared to observed ANNP (Whitmore *et al.*, 2005) shows a strong relationship ($R^2 = 0.96$) (Figure 12A).

4.2.3 Spatial filtering and mapping

From the original data points, we first only retained ANNP values within the percentile interval of 0.5 (i.e. ≤ 99.5) in order to remove extreme reconstructed values. Then, we applied a spatial filter at specific distance radii using local indicators of spatial association (LISA) (Isaaks and Srivastava, 1989; Anselin, 1995). Reconstructions of climate should not violate the assumption of spatial continuity (i.e. we expect closer sites to be more similar than sites further away), so the use of the LISA is reasonable in this situation. Moran's I and Moran's I z-values of site anomalies provide information on whether individual sites within a specified search radius are negatively or positively spatially autocorrelated (Anselin, 1995). Here, we exclude sites with negative Moran's I and negative Moran's I z-value or mixed signs (very uncommon). By contrast, sites with positive Moran's I and Moran's I z-values are retained. Those sites with a Moran's I and Moran's I z-value of zero (that fall outside the search radius) are retained for larger search radii.

The filtering was expanded to a 250 km search radius at 50 km increments or lag distances (e.g. 0–50, 0–100, etc.). Using one lag from 0–250 km was also tested (rather than five cumulative lags), however a much lower percentage of sites were retained, and therefore this method was not pursued. Filtering was explored at larger search radii (at 50 km increments up to 500 km); however, the result was poorer site coverage in North America (Table 25 in Appendix E). Optimal spatial filtering was obtained using a search radius of 250 km in order to maximize spatial coverage of sites across North America and to ensure sufficient sites in each ecoregion. Approximately 48% of all original sites were retained based on the local Moran's I statistic (Table 6).

After filtering, the modern reconstructed original data point values remain strongly related to modern observations in ANNP ($R^2 = 0.94$) (Figure 12B). For example, the original reconstructed ANNP at 1300 BP (1050 CE) without any extreme outliers removed nor any application of a spatial filter (i.e. LISA) (Figure 13A) shows how extreme outliers can impact different regional spatial patterns (e.g. Northern Canada

appears wetter) as compared to the same reconstruction that excludes extreme outliers (Figure 13B). Moreover, the application of a spatial filter (LISA) in addition to the removal of extreme outliers results in improved regional contrasts (Figure 13C).

Maps at 100 year intervals using only the sites retained after filtering were interpolated with a constraint of 250 km using empirical bayesian kriging (EBK) (Figure 14). EBK is used as an alternative approach to ordinary kriging because it automates the estimation of the semi-variogram and its error distribution is based on many simulations (Krivoruchko and Gribov, 2014). However, the resulting spatial patterns through time are comparable when using either EBK or ordinary kriging.

4.3 Results

4.3.1 Reconstructed patterns of ANNP for last 2,000 years

The maps of the reconstructed ANNP anomalies show an evolution of multi-centennial to centennial scale variability, which alternates between dry and wet periods relative to present (Figure 14). In the Southwest, conditions were generally wetter than present, except for the period from 1150–1750 CE. Generally wetter conditions prevail in the North over the entire 1100 years compared to present, with the exception of the periods between 1050–1250 CE and 1650–1850 CE. However, this persistence of pattern could result from edge effects as this area is data-sparse (Figure 10A).

Average composite anomaly maps of the well-known multi-centennial scale warm and cold periods in North America; the Roman Warm Period [RWP] (50 BCE–150 CE), Dark Ages Cold Period [DAC] (350-650 CE), Medieval Warm Period [MWP] (950-1250 CE) and Little Ice Age [LIA] (1550-1850 CE) (Viau *et al.*, 2006; Viau *et al.*, 2012) are presented in Figure 15.

The regional features of the RWP as compared to the DAC are a north-south dry-wet dipole pattern over the west coast and the reverse over central North America. Wetter conditions are found in boreal Canada and along the Eastern Seaboard to the Florida panhandle region (Figure 15A). The DAC compared to the MWP appears slightly wetter over Northern Canada and the Southern USA but much drier over the Great Lakes and Eastern Canada and the Pacific Northwest (Figure 15B). The MWP appears slightly drier than the LIA across central and Northwestern North America, and there is a band of wetter conditions from Texas though to Labrador (Figure 15C).

Comparing the two warm and cold periods should give insight as to which ocean-atmospheric circulation patterns dominate warm and cold periods of the past 2,000

years. The RWP was generally wetter than the MWP, except for a band across central North America just to the north of the present-day Canadian US border (Figure 15D). The DAC as compared to the LIA is more heterogeneous with wetter conditions in Northern Canada, Northern Quebec/ Labrador and the southern USA. Drier conditions are found over the Midwest, Northern Ontario, Southwest Quebec and Alaska (Figure 15E). In addition, areas near the edge of the continent have less reliable reconstructions most likely due to the edge effects of the interpolation process. Maps could be masked out based on standard error of prediction values that are one, two or even three standard deviations from the average standard error of the interpolated map (e.g. Figures 92-94 in Appendix E).

4.3.2 Time-series by ecoregions

Average ANNP anomaly time series were computed (Figure 16) for the various ecoregions of North America (Figure 11) and further illustrate regional differences in precipitation patterns (Figure 16A and Figure 16B). The eastern boreal region (east of 90°W) shows opposite precipitation anomalies to the western boreal region (west of 90°W) during both the DAC and LIA (Figure 16A) and comparable precipitation anomalies during the RWP and MWP. The forest-tundra ecoregion was generally wetter than today, especially in the west. However, this ecoregion has very few sites (N=14 for west and N=16 for east), leading to a greater uncertainty (Figure 16A). The desert region shows a wet DAC but was drier than present in most other periods over the last 2,000 years, although again, there are few sites (N=27) (Figure 16B). To some extent the time series of conifer/hardwood forest is similar to the Eastern boreal region, although the conifer/hardwood ecoregion was drier during the RWP. The deciduous forest and prairie regions show little variability in precipitation (Figure 16B). However, these ecoregions have roughly equal areas of wet and dry anomalies that would cancel each other out. Time series of anomalies for the Mediterranean (in Southern California), Arctic, coastal and mountain vegetation ecoregions are not shown due to a lack of sites in those regions after spatial filtering.

4.4 Discussion

Our 2,000 year pollen-based hydroclimate reconstruction uses the composite approach and chronology from Viau *et al.* (2012). However, in several cases large-scale reconstructions may show spatial heterogeneity due to a number of factors such as different

levels of chronological control, varying temporal sampling resolution and regional differences in the spatial distribution of sites. This study makes use of local indicators of spatial autocorrelation (LISA) to improve the interpretation of regional precipitation (ANNP) patterns at a consistent spatial resolution. The application of a spatial filter excludes sites that are primarily responding to local-scale variability rather than at the selected regional-scale, which may have data problems such as lower chronological control (Viau and Gajewski, 2001). This methodology is a useful alternative to other paleoclimate reconstruction methods (e.g. Principal Component Analysis [PCA]) and could be applied to other large-scale reconstructions (e.g. Mann *et al.*, 1998).

Our paleoclimate reconstructions document centennial to multi-centennial changes in ANNP over the past two millennia. Comparison between our reconstructions and high resolution records such as tree-rings can be complicated by both temporal and spatial variability but the PAGES NAM 2k effort has been successful in combining reconstructions in this manner (PAGES 2k Consortium, 2013). For example, a given century may be reconstructed as slightly wetter over an ecoregion even when multi-decadal scale events (e.g. mega-droughts) dominated certain decades. Other measures such as moisture/drought indices (e.g. Palmer Drought Severity Index [PDSI]) are often used to reconstruct wet/dry periods based on tree-rings which further complicate comparisons with ANNP (e.g. Woodhouse and Overpeck, 1998).

4.4.1 Reconstructed hydroclimatic conditions in North America

Precipitation anomalies and variability tends to be of smaller scale than temperature (Fritz *et al.*, 2000). Comparison between our multi-centennial-scale reconstructions for the MWP and LIA show that the MWP was characterized by regional patterns of diverse precipitation anomalies (Figure 15) (Stine, 1994; Cook *et al.*, 2004; Herweijer *et al.*, 2007; MacDonald, 2007). For example, drier conditions in the four corners region of the USA (i.e. southwestern corner of Colorado, northwestern corner of New Mexico, northeastern corner of Arizona and southeastern corner of Utah) coincide with the timing of peak Medieval warmth at 850 and 950 CE (Figure 15) (Christiansen and Ljungqvist, 2012; Viau *et al.*, 2012; Trouet *et al.*, 2013). Our reconstruction generally agrees with reconstructions based on tree-rings in Northwest New Mexico (D'Arrigo and Jacoby, 1991; Grissino-Mayer *et al.*, 1996) and centennial-scale wet/dry periods in the Northern Great Plains over the last 2,000 years (Clark *et al.*, 2002; Brown *et al.*, 2005) (Figure 14). Interpretation of hydroclimate patterns

during the LIA is complicated by century-scale regional mega-droughts in the 16th century (Stahle *et al.*, 2000; Cook *et al.*, 2004; Stahle *et al.*, 2007) (Figure 14 and 15), although it is not clear if similar anomalies had occurred during the DAC period, for example. We show evidence for a transition from MWP to LIA moisture conditions similar to that recorded from peat bogs between 700 and 1000 BP (∼1350 and 1600 CE) in the Western Great Lakes (Booth *et al.*, 2006) and the Southeastern USA from tree-rings (Stahle and Cleaveland, 1992). In addition, our reconstructions show evidence of steep moisture gradients both to the west and east of the Northern Great Plains during the LIA (1550–1850 CE) as shown by Fritz *et al.* (1994). Although the RWP and DAC periods remain understudied, our results suggest that the MWP was generally drier than the RWP. Conversely, our results suggest a drier LIA as compared to the DAC (Figure 15).

4.4.2 Ocean-Atmospheric linkages

Trouet *et al.* (2009) have suggested that the Medieval Warm Period (MWP) was characterized by a La Nina-like state, with a positive North Atlantic Oscillation (NAO) and positive (warm) Atlantic Multidecadal Oscillation (AMO). Our reconstructions between 1150 and 1450 CE also suggest a La Nina and negative (cold) Pacific Decadal Oscillation (PDO) pattern which led to drier conditions throughout the south, southwest and Florida and wetter conditions along the west coast and the south-eastern Atlantic coast(Figure 14 and 15)(e.g. McCabe *et al.*, 2004). Moreover, the positive phase of the NAO is associated with La Nina (Müller and Roeckner, 2006; Trouet *et al.*, 2009). The pattern of ANNP anomalies during the RWP is similar to the MWP, but with a lower magnitude (Figure 14). Alternatively, the DAC (350–650 CE) suggests opposite patterns to the MWP (1150–1450 CE) where more persistent El Nino, negative (cold) AMO and positive (warm) PDO conditions resulted in wetter conditions in the south and southwest, drier conditions in the Ohio Valley and wetter conditions along the south-eastern Atlantic coast (Figure 14) (Figure 3a in McCabe *et al.* (2008)). The LIA (1550–1850 CE) is more difficult to interpret possibly due to regional scale mega-droughts (e.g. Cook *et al.*, 2004; Stahle *et al.*, 2007), low solar and high volcanic activity. For example, high solar and low volcanic activity are considered to be main forcings during the MWP as compared to the LIA (Shindell *et al.*, 2003; Mann *et al.*, 2009a).

Another prominent feature in our reconstructions is a moisture dipole pattern (a north-south seesaw) along the west coast (Figure 14). We suggest this could be related to the PDO and Northeast Pacific sea level pressure patterns (Dettinger *et al.*, 1998; MacDonald and Case, 2005). The phase of the Pacific North American

(PNA) pattern is related to the PDO and El Nino Southern Oscillation (ENSO), which has an important impact on precipitation anomalies in North America (e.g. Yu and Zwiers, 2007; Trouet and Taylor, 2010). For example, the wetter western boreal forest could be associated with the positive phase of the PNA pattern and the drier boreal east could be associated with the negative phase of the PNA circulation (Trouet and Taylor, 2010)(Figure 16). Modern zonal to meridional changes in atmospheric circulation over the boreal region where increased meridional flow over the last few decades in central and eastern Canada has been linked to tropical Pacific SSTs (Girardin *et al.*, 2006). The +PNA is associated with a high amplitude upper level ridge in Western North America and a high amplitude upper level trough in Eastern North America at the 500 mb level (Leathers *et al.*, 1991). When ENSO and PDO are in phase (either El Nino-warm PDO or La Nina-cold PDO), there is a PNA-like climate response in North America (Yu and Zwiers, 2007). Multi-century GCM simulations show anti-correlation between the PNA and NAO so positive (negative) NAO would be correlated with negative (positive) PNA (Pinto *et al.*, 2011). It is possible that during the MWP (and possibly RWP), the La Nina and cold PDO resulted in a negative phase of the PNA (-PNA) pattern leading to a more zonal atmospheric flow across North America with a weak upper level trough (ridge) in Western (Eastern) North America (Trouet and Taylor, 2010). As a result, it is possible to have smaller differences in ANNP anomalies between the eastern and western boreal regions (Figure 16).

4.5 Conclusion

Mapped patterns of the total annual precipitation for the past 2,000 years based on pollen records show evidence for La Nina, cold Pacific Decadal Oscillation (PDO), warm Atlantic Multidecadal Oscillation (AMO) and positive North Atlantic Oscillation (NAO)-like states during the Medieval Warm Period (MWP). These regional patterns are consistent with high resolution reconstructions from tree-rings and other high-resolution paleoclimate proxies (Mann *et al.*, 2009a; Diaz *et al.*, 2011; PAGES 2k Consortium, 2013). The Roman Warm Period (RWP) shows similar regional precipitation anomalies to the MWP suggesting that comparable ocean-atmosphere circulation patterns dominated these two warm periods. The boreal east/west ecoregions time series' provide evidence for a positive (negative) Pacific North American Pattern (PNA) during both the DAC and LIA (RWP and MWP) suggesting El Nino (La Nina) states. The DAC period shows regional patterns associated with a dominant El Nino-like state. These teleconnections may have operated on longer-timescales

than present.

This study has presented a novel approach using spatial smoothing through the use of Local Indicators of Spatial autocorrelation (LISA) for reconstructing climate patterns from noisy data. The data presented in this study provide useful benchmark data set for data-model comparisons that can lead to a better understanding of hydroclimatic variability in North America during the past two millennia.

Acknowledgments

This research was supported by grants from the Natural Sciences and Engineering Council of Canada (NSERC). We thank the contributors for their available modern and fossil pollen data sets.

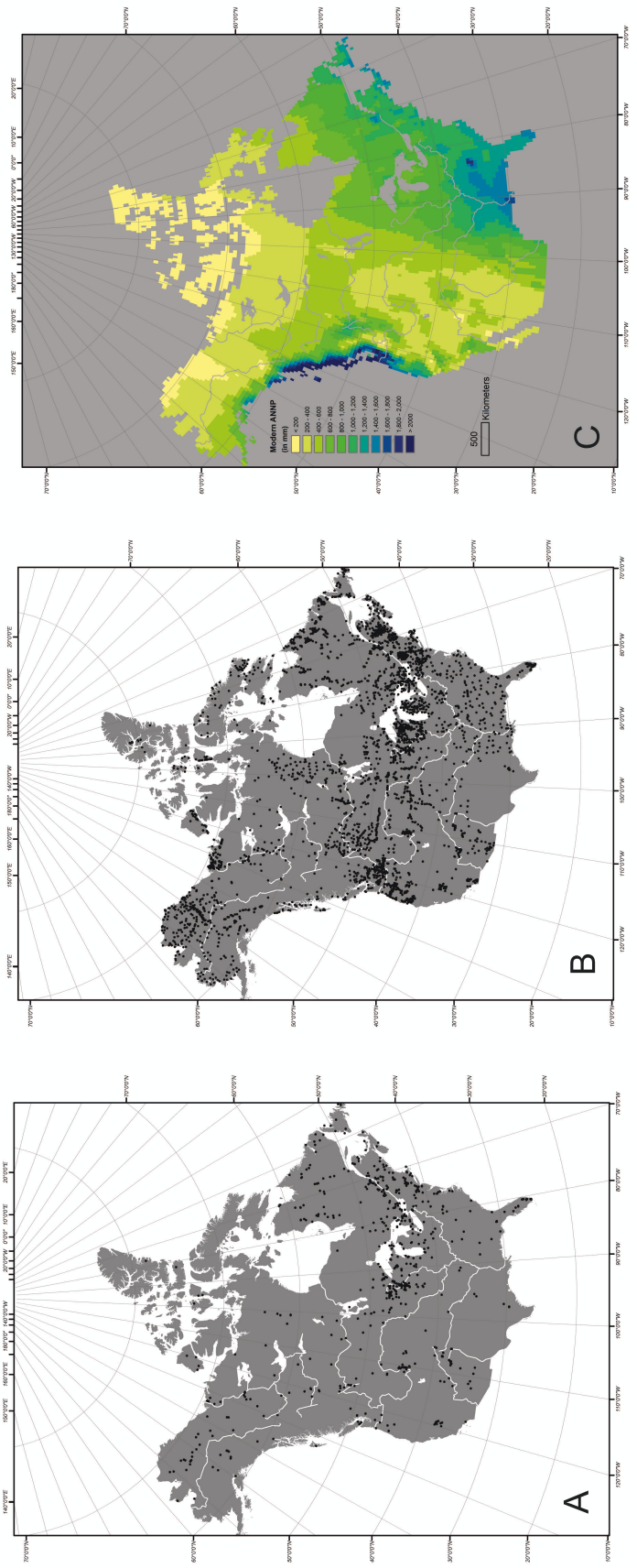


Figure 10: Distribution of modern (A) and fossil (B) pollen sites along with modern ANNP (C).

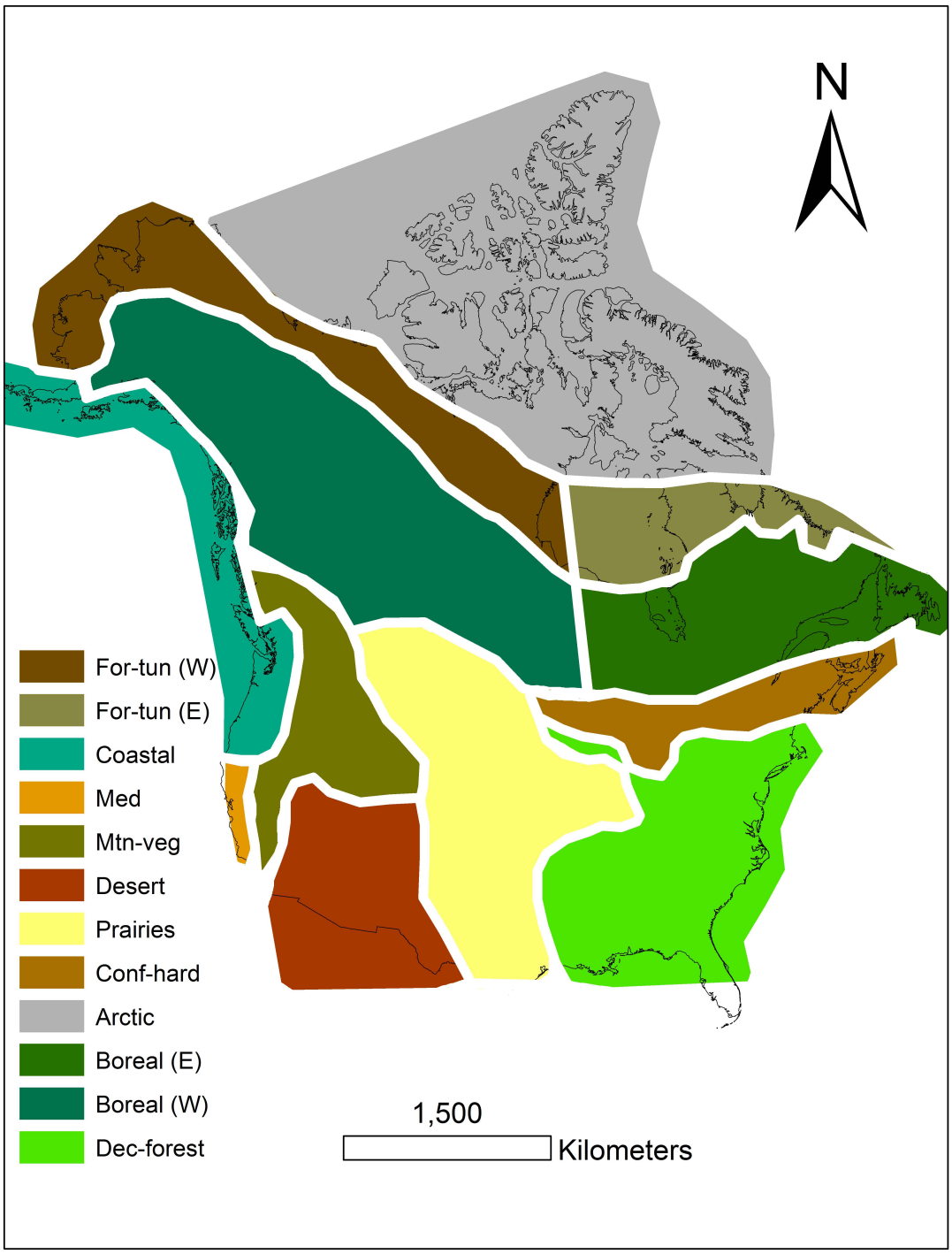


Figure 11: Ecoregions of North America adapted from Fedorova *et al.* (1994).

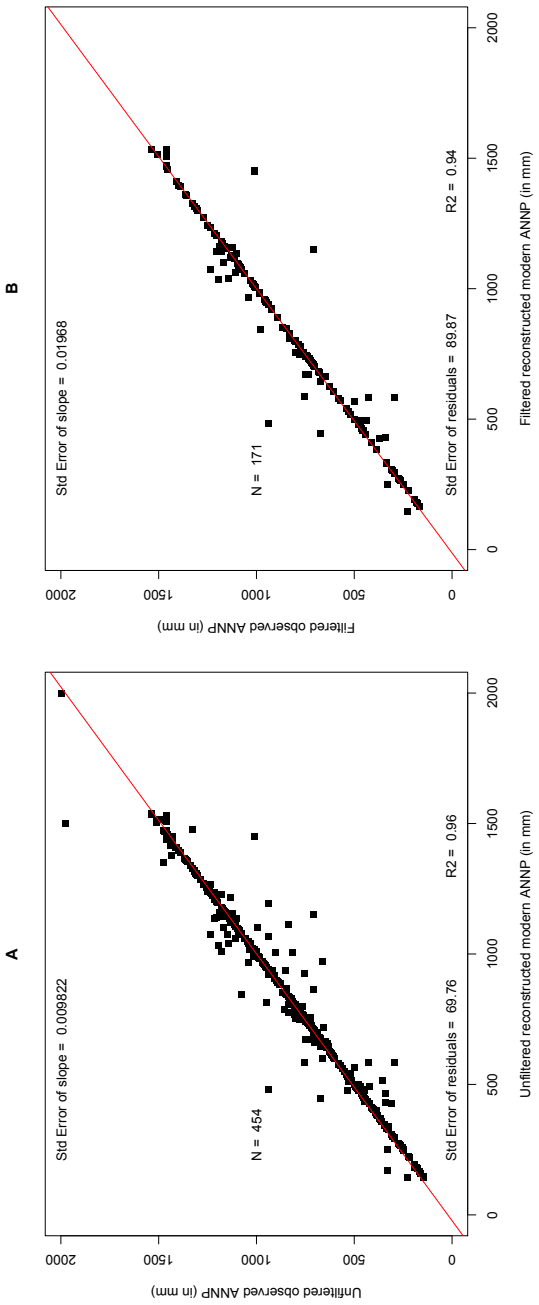


Figure 12: Reconstructed 0 BP (1950 CE) versus observed for ANNP using all original data points (A) and retained data points after spatial filtering is applied (B).

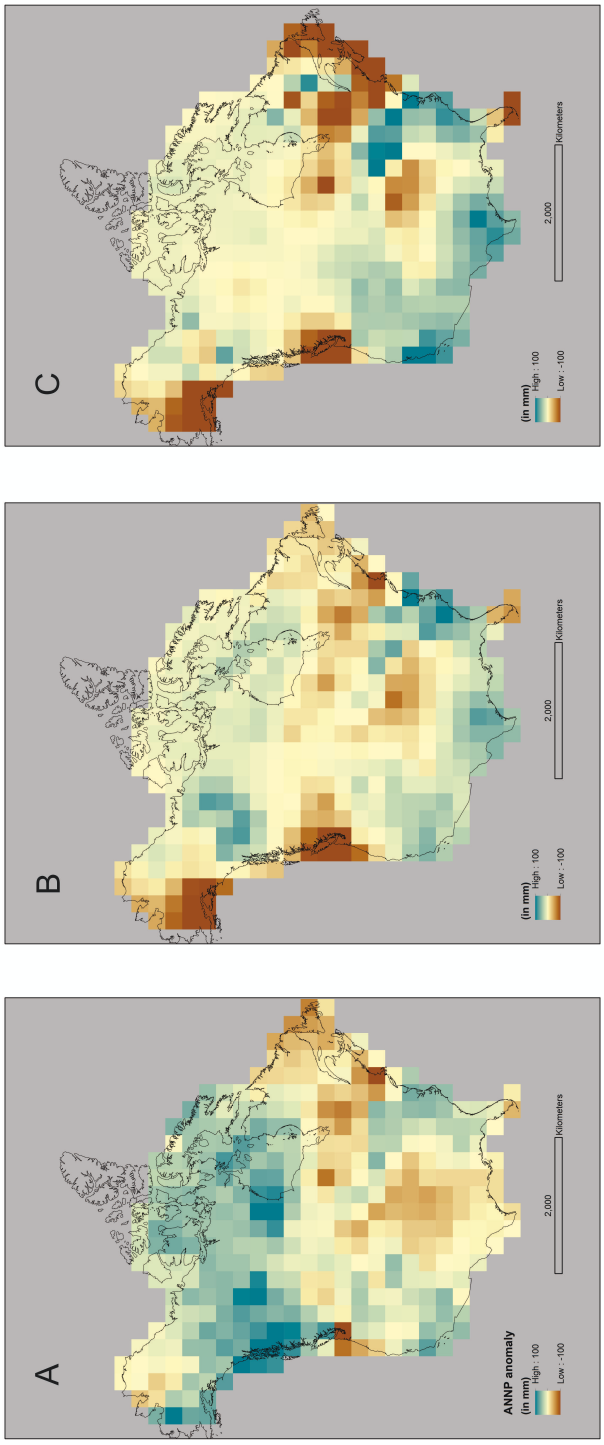


Figure 13: ANNP anomalies at 1300 BP [1050 CE] using all original data points (A), original data points with outliers excluded (B) and retained data points after spatial filtering with outlier data points excluded (C). Anomalies range from dark brown (-100 mm/yr) to dark aqua (+100 mm/yr).

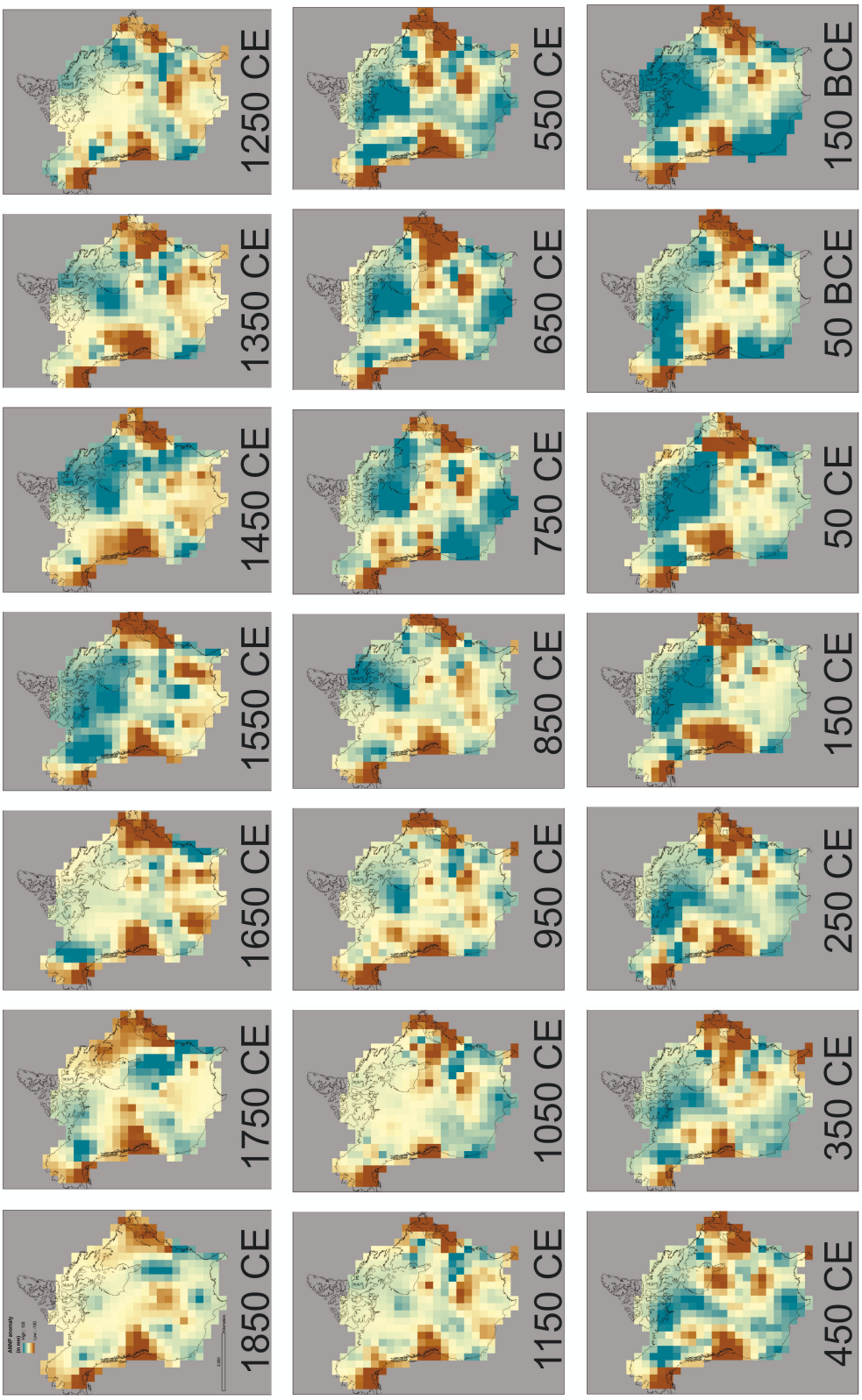


Figure 14: Spatially filtered maps of ANNP anomalies at 100 year intervals for 150 BCE to 1850 CE interpolated using empirical bayesian kriging (gridded at 250 km). Anomalies range from dark brown (-100 mm/yr) to dark aqua (+100 mm/yr).

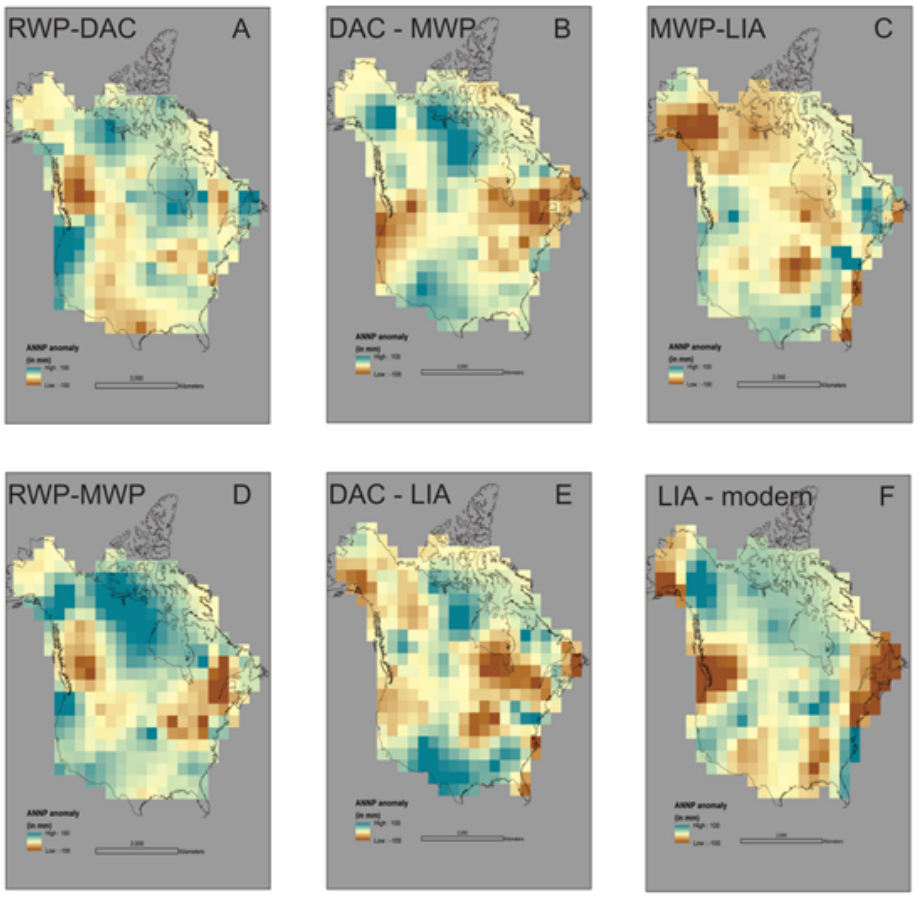


Figure 15: ANNP anomalies (in mm) for the RWP [150 BCE–150 CE] minus DAC [350–650 CE] (A), DAC minus MWP [950–1250 CE] (B), MWP minus LIA [1550–1850 CE] (C), RWP minus MWP (D), DAC minus LIA (E) and LIA minus modern (F). Anomalies range from dark brown (-100 mm/yr) to dark aqua (+100 mm/yr).

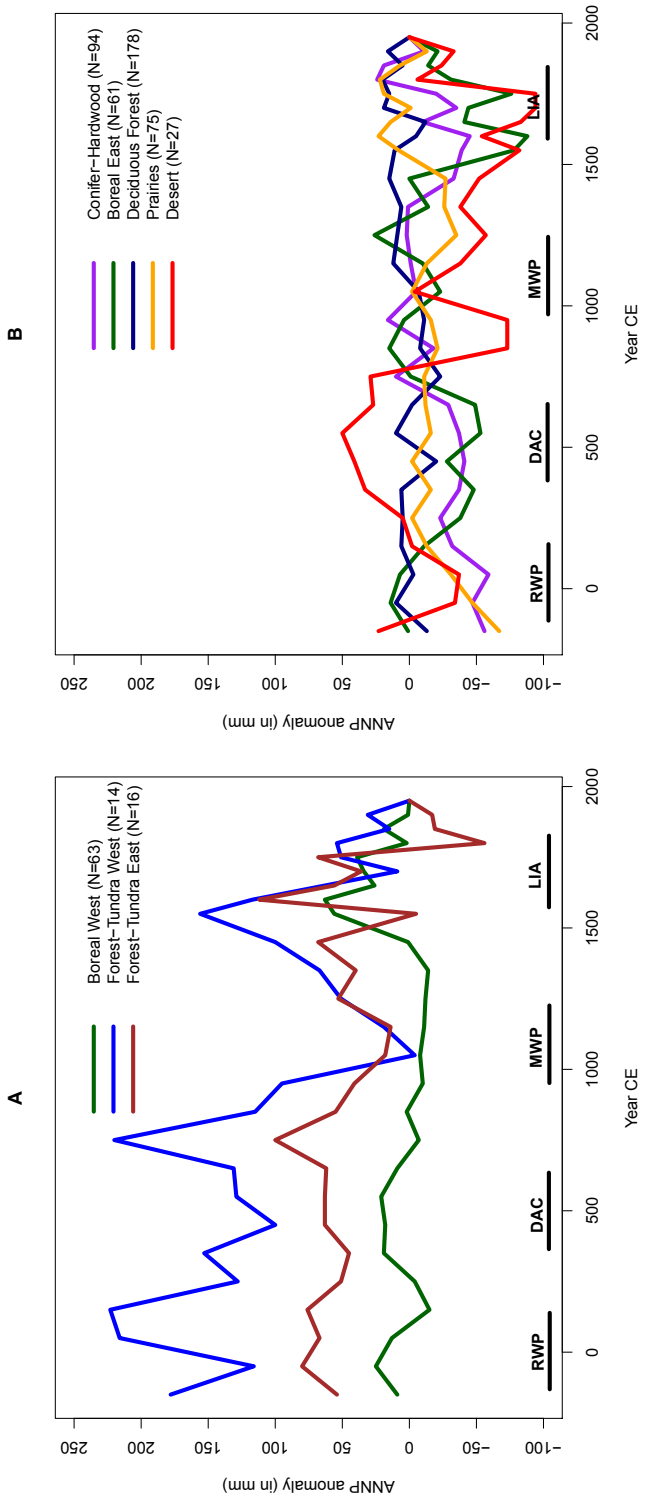


Figure 16: Average ANNP anomalies (in mm) through time by ecoregion in North America. The boreal and forest-tundra regions were divided into west and east using a threshold of 90°W (Figure 11). Panel A includes boreal west, forest-tundra west and forest-tundra east regions. Panel B includes conifer-hardwood, boreal east, deciduous forest, prairies and desert regions. The approximate age ranges for the RWP, DAC, MWP and LIA periods are indicated.

Table 5: Pollen taxa used in reconstructions [Williams and Shuman (2008)].

Abies	*Liquidambar*
Acer	Lycopodium [a]
Alnus rubra	Minor Forbs [b]
Alnus rugosa	Myricaceae
Alnus [c]	*Nyssa*
Aquifoliaceae	*Ostrya/Carpinus*
Arceuthobium	*Oxyria*
Artemisia	Papaveraceae
Asteraceae [d]	Picea [e]
Betula	*Pinus diploxylon*
Brassicaceae	*Pinus haploxylon*
Cactaceae	Pinus undifferentiated
Carya	Pinus Southeast
Caryophyllaceae	Plantaginaceae
Castanea	*Platanus*
Celtis	Poaceae
Cephalanthus	*Populus*
Chenopodiaceae/Amaranthaceae	*Prosopis*
Chrysolepis/Lithocarpus	*Pteridium*
Corylus	*Quercus*
Cupressaceae	Rosaceae
Cyperaceae	*Salix*
Dryas	*Sarcobatus*
Ephedra	Saxifraga [f]
Ericaceae [g]	*Sphagnum*
Fagus	*Taxodium*
Fraxinus nigra	*Tilia*
Fraxinus pennsylvanica/americana	*Tsuga heterophylla*
Fraxinus undifferentiated	*Tsuga mertensiana*
Juglans [h]	Tsuga undifferentiated
Larix/Pseudotsuga	*Ulmus*
Larrea	

[a]Sum of *Lycopodium* undiff. *L. annotinum*, *L. clavatum*, *L. complanatum* and *L. selago*

[b]Sum of Apiaceae, *Eriogonum*, Euphorbiaceae, Malvaceae.

[c]Sum of *Alnus crispa* and *Alnus* undiff

[d]Sum of Asteraceae undiff, Asteraceae subf. Cichorioideae, *Iva* and *Xanthium*. Note that this taxon does not include *Artemisia* or *Ambrosia*.

[e]Sum of *Picea* undiff, *Picea glauca* and *P. mariana*

[f]Sum of *Saxifraga cernua*, *S. hieracifolia*, *S. oppositifoli*, and *S. tricuspidata*

[g]Sum of Ericaceae undiff., *Chamaedaphne/Ledum* and *Vaccinium*

[h]Sum of *Juglans* undiff, *J. cinerea*, and *J. nigra*

Table 6: Positive, zero, and negative values and combinations for Moran's I and Moran's I (z-values) and the number of neighbors & percentages for sites at 100 year time intervals for the reconstruction.

period	MI (neg)	MI (zero)	MI (pos)	zMI (neg)	zMI (zero)	zMI (pos)	neg/neg	pos/pos	mixed	zero
1950 CE	0	454	0	0	454	0	0	0	0	454
0 BP	0	1	0	0	1	0	0	0	0	1
1900 CE	46	28	138	40	28	144	40	138	6	28
100 BP	0.22	0.13	0.65	0.19	0.13	0.68	0.19	0.65	0.03	0.13
1850 CE	17	24	163	14	24	166	14	163	3	24
200 BP	0.08	0.12	0.8	0.07	0.12	0.81	0.07	0.8	0.01	0.12
1800 CE	20	27	160	15	27	165	15	160	5	27
300 BP	0.1	0.13	0.77	0.07	0.13	0.79	0.07	0.77	0.02	0.13
1750 CE	28	23	169	20	23	177	20	169	8	23
400 BP	0.13	0.1	0.76	0.09	0.1	0.8	0.09	0.76	0.04	0.1
1700 CE	16	26	174	9	26	181	9	174	7	26
500 BP	0.07	0.12	0.8	0.04	0.12	0.83	0.04	0.8	0.03	0.12
1650 CE	12	26	180	7	26	185	7	180	5	26
600 BP	0.05	0.12	0.82	0.03	0.12	0.84	0.03	0.82	0.02	0.12
1600 CE	13	28	181	9	28	185	9	181	4	28
700 BP	0.06	0.13	0.81	0.04	0.13	0.83	0.04	0.81	0.02	0.13
1550 CE	21	28	191	10	28	202	10	191	11	28
800 BP	0.09	0.12	0.79	0.04	0.12	0.84	0.04	0.79	0.05	0.12
1450 CE	47	27	199	25	27	221	25	199	22	27
900 BP	0.17	0.1	0.73	0.09	0.1	0.81	0.09	0.73	0.08	0.1
1350 CE	24	25	204	17	25	211	17	204	7	25
1000 BP	0.09	0.1	0.8	0.07	0.1	0.83	0.07	0.8	0.03	0.1
1250 CE	35	23	207	18	23	224	18	207	17	23
1100 BP	0.13	0.09	0.78	0.07	0.09	0.84	0.07	0.78	0.06	0.09
1150 CE	29	25	184	11	25	202	11	184	18	25
1200 BP	0.12	0.1	0.77	0.05	0.1	0.85	0.05	0.77	0.08	0.1
1050 CE	25	29	173	22	29	176	22	173	3	29
1300 BP	0.11	0.13	0.76	0.1	0.13	0.77	0.1	0.76	0.01	0.13
950 CE	21	28	190	19	28	192	19	190	2	28
1400 BP	0.09	0.12	0.79	0.08	0.12	0.8	0.08	0.79	0.01	0.12
850 CE	16	29	212	13	29	215	13	212	3	29
1500 BP	0.06	0.11	0.82	0.05	0.11	0.83	0.05	0.82	0.01	0.11
750 CE	24	25	184	19	25	189	19	184	5	25
1600 BP	0.1	0.11	0.79	0.08	0.11	0.81	0.08	0.79	0.02	0.11
650 CE	15	25	241	7	25	249	7	241	8	25
1700 BP	0.05	0.09	0.85	0.02	0.09	0.88	0.02	0.85	0.03	0.09
550 CE	22	24	207	17	24	212	17	207	5	24
1800 BP	0.09	0.09	0.81	0.07	0.09	0.83	0.07	0.81	0.02	0.09
450 CE	13	26	257	6	26	264	6	257	7	26
1900 BP	0.04	0.09	0.87	0.02	0.09	0.89	0.02	0.87	0.02	0.09
350 CE	34	25	175	26	25	183	26	175	8	25
2000 BP	0.14	0.11	0.74	0.11	0.11	0.78	0.11	0.74	0.03	0.11
250 CE	16	27	195	12	27	199	12	195	4	27
2100 BP	0.07	0.11	0.82	0.05	0.11	0.83	0.05	0.82	0.02	0.11
150 CE	21	26	192	14	26	199	14	192	7	26
2200 BP	0.09	0.11	0.8	0.06	0.11	0.83	0.06	0.8	0.03	0.11
50 CE	20	26	171	19	26	172	19	171	1	26
2300 BP	0.09	0.12	0.78	0.09	0.12	0.79	0.09	0.78	0	0.12
50 BCE	15	25	218	11	25	222	11	218	4	25
2400 BP	0.06	0.1	0.84	0.04	0.1	0.86	0.04	0.84	0.02	0.1
150 BCE	17	23	195	14	23	198	14	195	3	23
2500 BP	0.07	0.1	0.83	0.06	0.1	0.84	0.06	0.83	0.01	0.1

References

Anselin, L. (1995). Local Indicators of Spatial Association - LISA. *Geographical Analysis* **27**(2), 93–115.

Bartlein, P.J., S.J. Harrison, S. Brewer, S. Connor, B.A.S. Davis, K. Gajewski, J. Guiot, T.I. Harrison-Prentice, A. Henderson, O. Peyron, I.C. Prentice, M. Scholze, H. Seppä, B. Shuman, S. Sugita, R.S. Thompson, A.E. Viau, J. Williams, and H. Wu (2011). Pollen-based continental climate reconstructions at 6 and 21 ka: a global synthesis. *Climate Dynamics* **37**(3-4), 775–802.

Booth, R.K, M. Notaro, S.T. Jackson, and J.E. Kutzbach (2006). Widespread drought episodes in the western Great Lakes region during the past 2000 years: geographic extent and potential mechanisms. *Earth and Planetary Science Letters* **242**(3), 415–427.

Brown, K.J., J.S. Clark, E.C. Grimm, J.J. Donovan, P.G. Mueller, B.C.S. Hansen, and I. Stefanova (2005). Fire cycles in North American interior grasslands and their relation to prairie drought. *Proceedings of the National Academy of Sciences of the United States of America* **102**(25), 8865–8870.

Cheddadi, R., G. Yu, J. Guiot, S. P. Harrison, and I.C. Prentice (1997). The climate of Europe 6000 years ago. *Climate Dynamics* **13**(1), 1–9.

Christiansen, B. and F. C. Ljungqvist (2012). The extra-tropical Northern Hemisphere temperature in the last two millennia: reconstructions of low-frequency variability. *Climate of the Past* **8**(2), 765–786.

Clark, J.S., E.C. Grimm, J.J. Donovan, S.C. Fritz, D.R. Engstrom, and J.E. Almendinger (2002). Drought cycles and landscape responses to past aridity on prairies of the northern Great Plains, USA. *Ecology* **83**(3), 595–601.

Cook, E.R., C.A. Woodhouse, C.M. Eakin, D.M. Meko, and D.W. Stahle (2004). Long-term aridity changes in the western United States. *Science* **306**(5698), 1015–1018.

D'Arrigo, Rosanne D and Gordon C Jacoby (1991). A 1000-year record of winter precipitation from northwestern New Mexico, USA: a reconstruction from tree-rings and its relation to El Niño and the Southern Oscillation. *The Holocene* **1**(2), 95–101.

Dettinger, M.D., D.R. Cayan, H.F. Diaz, and D.M. Meko (1998). North–South Precipitation Patterns in Western North America on Interannual-to-Decadal Timescales. *Journal of Climate* **11**(12).

Diaz, H.F., R. Trigo, M.K. Hughes, M.E. Mann, E. Xoplaki, and D. Barriopedro (2011). Spatial and temporal characteristics of climate in Medieval times revisited. *Bulletin of the American Meteorological Society* **92**(11), 1487–1500.

Esper, J., D.C. Frank, M. Timonen, E. Zorita, R.J.S Wilson, J. Luterbacher, S. Holzkämper, N. Fischer, S. Wagner, D. Nievergelt, A. Verstege, and U. Büntgen (2012). Orbital forcing of tree-ring data. *Nature Climate Change* **2**(12), 862–866.

Fritz, S.C., D.R. Engstrom, and B.J. Haskell (1994). 'Little Ice Age'aridity in the North American Great Plains: a high-resolution reconstruction of salinity fluctuations from Devils Lake, North Dakota, USA. *The Holocene* **4**(1), 69–73.

Fritz, S.C., E. Ito, Z. Yu, K.R. Laird, and D.R. Engstrom (2000). Hydrologic variation in the northern Great Plains during the last two millennia. *Quaternary Research* **53**(2), 175–184.

Gajewski, K. (1987). Climatic impacts on the vegetation of eastern North America during the past 2000 years. *Vegetatio* **68**(3), 179–190.

— (1988). Late Holocene climate changes in eastern North America estimated from pollen data. *Quaternary Research* **29**(3), 255–262.

— (1993). The role of paleoecology in the study of global climatic change. *Review of palaeobotany and palynology* **79**(1), 141–151.

— (2008). The Global Pollen Database in biogeographical and palaeoclimatic studies. *Progress in Physical Geography* **32**(4), 379–402.

Gajewski, K. and A.E. Viau (2011). Abrupt climate changes during the Holocene across North America from pollen and paleolimnological records. *Abrupt Climate Change: Mechanisms, Patterns, and Impacts, Geophysical Monograph Series*. Ed. by L. Polyak H. Rashid and E. Mosley-Thompson. Washington, D.C., USA: AGU, 161–171.

Girardin, M-P., J.C. Tardif, M.D. Flannigan, and Y. Bergeron (2006). Synoptic-scale atmospheric circulation and boreal Canada summer drought variability of the past three centuries. *Journal of Climate* **19**(10).

Grimm, E.C. (2011). *North American Pollen database. Data archived at IGBP PAGES/-World Data Center–A for Paleoclimatology, NOAA/NGDC Paleoclimatology Program, Boulder, CO, USA*. url: http://www.ncdc.noaa.gov/paleo.

Grissino-Mayer, H.D. *et al.* (1996). A 2129-year reconstruction of precipitation for northwestern New Mexico, USA. *Tree rings, environment, and humanity*, 191–204.

Herweijer, C., R. Seager, E.R. Cook, and J. Emile-Geay (2007). North American droughts of the last millennium from a gridded network of tree-ring data. *Journal of Climate* **20**(7), 1353–1376.

Hijmans, R.J., S.E. Cameron, J.L. Parra, P. G. Jones, and A. Jarvis (2005). Very high resolution interpolated climate surfaces for global land areas. *International journal of climatology* **25**(15), 1965–1978.

IPCC (2007). *Climate Change 2007, The Physical Science Basis. Fourth Assessment Report*. Ed. by S. Solomon, D. Manning, Z. Chen, M. Marquis, K.B. Averyt, M. Tignor, and H. Miller. Cambridge, UK and New York, NY, USA: Cambridge University Press, p. 996.

— (2013). *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by T.F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P.M. Midgley. Cambridge, UK and New York, NY, USA: Cambridge University Press, p. 1552.

Isaaks, E.H. and R.H. Srivastava (1989). *An Introduction to Applied Geostatistics*. New York, USA: Oxford University Press.

Juggins, S. and H.J.B. Birks (2012). Quantitative environmental reconstructions from biological data. *Tracking environmental change using lake sediments*, 431–494.

Krivoruchko, K. and A. Gribov (2014). Pragmatic Bayesian Kriging for Non-Stationary and Moderately Non-Gaussian Data. *Mathematics of Planet Earth*. Ed. by E. Pardo-Igúzquiza, C. Guardiola-Albert, J. Heredia, L. Moreno-Merino, J.J. Durán, and J.A. Vargas-Guzmán. Lecture Notes in Earth System Sciences, 61–64.

Leathers, D.J., B. Yarnal, and M.A. Palecki (1991). The Pacific/North American teleconnection pattern and United States climate. Part I: Regional temperature and precipitation associations. *Journal of Climate* **4**(5), 517–528.

MacDonald, G.M. (2007). Severe and sustained drought in southern California and the West: Present conditions and insights from the past on causes and impacts. *Quaternary International* **173**, 87–100.

MacDonald, G.M. and R.A. Case (2005). Variations in the Pacific Decadal Oscillation over the past millennium. *Geophysical Research Letters* **32**(8).

Mann, M.E., R.S. Bradley, and M.K. Hughes (1998). Global-scale temperature patterns and climate forcing over the past six centuries. *Nature* **392**(6678), 779–787.

Mann, M.E, Z. Zhang, S. Rutherford, R.S. Bradley, M.K. Hughes, D. Shindell, C. Ammann, G. Faluvegi, and F. Ni (2009a). Global signatures and dynamical origins of the Little Ice Age and Medieval Climate Anomaly. *Science* **326**(5957), 1256–1260.

McCabe, G.J., M.A. Palecki, and J.L. Betancourt (2004). Pacific and Atlantic Ocean influences on multidecadal drought frequency in the United States. *Proceedings of the National Academy of Sciences* **101**(12), 4136–4141.

McCabe, G.J., J.L. Betancourt, S.T. Gray, M.A. Palecki, and H.G. Hidalgo (2008). Associations of multi-decadal sea-surface temperature variability with US drought. *Quaternary International* **188**(1), 31–40.

Müller, W.A. and E. Roeckner (2006). ENSO impact on midlatitude circulation patterns in future climate change projections. *Geophysical research letters* **33**(5).

Overpeck, J.T., T. Webb III, and I.C. Prentice (1985). Quantitative interpretation of fossil pollen spectra: dissimilarity coefficients and the method of modern analogs. *Quaternary Research* **23**(1), 87–108.

PAGES 2k Consortium (2013). Continental-scale temperature variability during the past two millennia. *Nature Geoscience* **6**, 339–346.

Pinto, J.G., M. Reyers, and U. Ulbrich (2011). The variable link between PNA and NAO in observations and in multi-century CGCM simulations. *Climate dynamics* **36**(1-2), 337–354.

Sawada, M., A.E. Viau, G. Vettoretti, W.R. Peltier, and K. Gajewski (2004). Comparison of North-American pollen-based temperature and global lake-status with CCCma AGCM2 output at 6ka. *Quaternary Science Reviews* **23**(3), 225–244.

Sawada, M.C. (2006). An open source implementation of the Modern Analog Technique (MAT) within the R computing environment. *Computers and Geosciences* **32**(6), 818–833.

Shindell, D.T., G.A. Schmidt, R.L. Miller, and M.E. Mann (2003). Volcanic and solar forcing of climate change during the preindustrial era. *Journal of Climate* **16**(24).

Stahle, D.W. and M.K. Cleaveland (1992). Reconstruction and analysis of spring rainfall over the southeastern US for the past 1000 years. *Bulletin of the American Meteorological Society* **73**(12), 1947–1961.

Stahle, D.W., E.R. Cook, M.K. Cleaveland, M.D. Therrell, D.M. Meko, H.D. Grissino-Mayer, E. Watson, and B.H. Luckman (2000). Tree-ring data document 16th century megadrought over North America. *EOS, Transactions American Geophysical Union* **81**(12), 121–125.

Stahle, D.W., F. K. Fye, E.R. Cook, and R.D. Griffin (2007). Tree-ring reconstructed megadroughts over North America since AD 1300. *Climatic change* **83**(1–2), 133–149.

Stine, S. (1994). Extreme and persistent drought in California and Patagonia during mediaeval time. *Nature* **369**(6481), 546–549.

Trouet, V. and A.H. Taylor (2010). Multi-century variability in the Pacific North American circulation pattern reconstructed from tree rings. *Climate dynamics* **35**(6), 953–963.

Trouet, V., J. Esper, N.E. Graham, A. Baker, J.D. Scourse, and D.C. Frank (2009). Persistent positive North Atlantic Oscillation mode dominated the medieval climate anomaly. *Science* **324**(5923), 78–80.

Trouet, V., H.F. Diaz, E.R. Wahl, A.E. Viau, R. Graham, N. Graham, and E.R. Cook (2013). A 1500-year reconstruction of annual mean temperature for temperate North America on decadal-to-multidecadal time scales. *Environmental Research Letters* **8**(2), 024008.

Viau, A., M. Ladd, and K. Gajewski (2012). The climate of North America during the past 2000 years reconstructed from pollen data. *Global and Planetary Change* **84**, 75–83.

Viau, A.E. and K. Gajewski (2001). Holocene variations in the global hydrological cycle quantified by objective gridding of lake level databases. *Journal of Geophysical Research: Atmospheres (1984–2012)* **106**(D23), 31703–31716.

— (2009). Reconstructing millennial-scale, regional paleoclimates of boreal Canada during the Holocene. *Journal of Climate* **22**(2), 316–330.

Viau, A.E., K. Gajewski, M.C. Sawada, and P. Fines (2006). Millennial-scale temperature variations in North America during the Holocene. *Journal of Geophysical Research: Atmospheres (1984-2012)* **111**(D9).

Viau, A.E., K. Gajewski, M.C. Sawada, and J. Bunbury (2008). Low-and high-frequency climate variability in eastern Beringia during the past 25 000 years. *Canadian Journal of Earth Sciences* **45**(11), 1435–1453.

Wanner, H., J. Beer, J. Bütikofer, T.J. Crowley, U. Cubasch, J. Flückiger, H. Goosse, M. Grosjean, F. Joos, J.O. Kaplan, M. Küttel, S.A. Müller, I.C. Prentice, O. Solomina, T.F. Sticker, P. Trasov, M. Wagner, and M. Widmann (2008). Mid-to Late Holocene climate change: an overview. *Quaternary Science Reviews* **27**(19), 1791–1828.

Whitmore, J., K. Gajewski, M.C. Sawada, J.W. Williams, B. Shuman, P.J. Bartlein, T. Minckley, A.E. Viau, T. Webb III, S. Shafer, P. Anderson, and L. Brubaker

(2005). Modern pollen data from North America and Greenland for multi-scale paleoenvironmental applications. *Quaternary Science Reviews* **24**(16), 1828–1848.

Williams, J.W. and B. Shuman (2008). Obtaining accurate and precise environmental reconstructions from the modern analog technique and North American surface pollen dataset. *Quaternary Science Reviews* **27**(7), 669–687.

Wilson, R., .R D'Arrigo, .B Buckley, U. Büntgen, J. Esper, D. Frank, B. Luckman, S. Payette, R. Vose, and D. Youngblut (2007). A matter of divergence: tracking recent warming at hemispheric scales using tree ring data. *Journal of Geophysical Research: Atmospheres (1984–2012)* **112**(D17).

Woodhouse, C.A. and J.T. Overpeck (1998). 2000 years of drought variability in the central United States. *Bulletin of the American Meteorological Society* **79**(12), 2693–2714.

Yu, B. and F.W. Zwiers (2007). The impact of combined ENSO and PDO on the PNA climate: a 1,000-year climate modeling study. *Climate dynamics* **29**(7-8), 837–851.

Chapter 5

Summary

5.1 Conclusions

The goal of this dissertation was to gain a better understanding of the climate of North America over the last two millenia using pollen as proxy to past climate variability. In order to achieve this goal, four objectives were established:

1. Develop a new North American continental-scale mean July temperature (TJUL) reconstruction for the past 2,000 years to explore multi-centennial scale climate variability relative to modern conditions for several ecoregions of North America.

2. Evaluate the impacts of the use of different modern TJUL climate data sets (both station and reanalysis products) on the pollen-based North American continental-scale reconstruction of the past 2,000 years.

3. Develop a new North American continental-scale total annual precipitation (ANNP) reconstruction for the past 2,000 years to explore multi-centennial scale climate variability relative to modern conditions for several ecoregions of North America and develop a method of spatial filtering to improve the reconstruction.

4. Develop a user-friendly R program that allows for user-defined pollen-based paleoclimate reconstructions.

This section will outline the main contributions of each chapter and future research directions.

5.2 Summary of Objective 1

A reconstruction of the MTWA (summer temperature or TJUL) in North America (by ecoregion) for the last 2,000 years was completed. The key contributions of chapter 2 are as follows:

- The modern climate appears warmer than both the Medieval Warm Period (MWP; 650–1050 AD) the subsequent Little Ice Age (LIA; 1550–1850 AD).
- The MWP was warmer than the LIA over at least three large ecoregions of North America.
- Reconstructions from the western coastal, mountain, desert, and Arctic ecoregions are less reliable due to fewer available data and larger uncertainties in the reconstructions. The prairie ecoregion reconstruction potentially reflects moisture availability rather than July temperature.
- These large-scale changes during Medieval times are not insignificant and do not appear to result from a gradual decline in temperature due to orbital-scale forcing.
- These results can be interpreted as an expansion northward of the summer Subtropical High Pressure system (i.e. the Bermuda High) in the North Atlantic during the MWP.

5.3 Summary of Objective 2

Several modern July temperature (TJUL) data sets are tested against the mean TJUL reconstructions of the past 2,000 years from objective 1 in order to examine the impacts on the reconstruction. The key contributions of chapter 3 are as follows:

- Reanalysis-based data sets (e.g. National Center for Environmental Prediction [NCEP] and European Reanalysis [ERA] products) are not recommended for continental-scale paleoclimate reconstructions as they result in smoothed and warm-biased reconstructions, especially during the period of 1050–1550 CE. In addition, the use of North American Regional Reanalysis [NARR] remains outside the range of the warmest station-based reconstructions.
- The use of a very high spatial resolution modern data set such as the WorldClim [WCLIM] 1 km grid is only useful if the precision of the location of the modern pollen sites is equal or better than the precision of the WorldClim grid

resolution. Therefore, if the precision of the location of the modern pollen sites is lower than the grid resolution of WCLIM, then WCLIM will not provide any additional information and should be resampled to an appropriate size.

- Conversely, an ultra-high resolution modern climate data set such as WCLIM may be useful approach for a local or small-scale reconstruction; however some type of statistical downscaling approach may be needed to ensure its reliability for surface or near-surface modern climate conditions.

- When using the nearest modern climate station data approach (e.g. Berkeley Earth Surface Temperature [BEST], Climate Research Unit [CRU], Global Historical Climate Network [GHCN]), a spatially heterogeneous distribution of modern pollen sites in a reconstruction, may not provide any additional information.

- The use of the lapse rate corrected CRU-based (Whitmore *et al.*, 2005) modern climate data for pollen-based climate reconstructions provides a reasonable compromise as compared to other station-based data sets, ultra-high resolution data sets and is superior to the use of reanalysis-based products. However, regional variation in lapse rates must be considered during the application of this type of correction.

5.4 Summary of Objective 3

A unique automated spatial filtering technique using Local Indices of Spatial Association (LISA) was developed and applied to a 2,000 year hydroclimatic reconstruction (ANNP) of North America. The key contributions of this spatially filtered reconstruction are as follows:

- We show evidence for La Nina, cold Pacific Decadal Oscillation (PDO), warm Atlantic Multidecadal Oscillation (AMO) and positive North Atlantic Oscillation (NAO)-like states during the MWP. These regional patterns are consistent with high resolution reconstructions from tree-rings.

- The RWP shows similar regional precipitation anomalies to the MWP although slightly drier suggesting that comparable ocean-atmosphere circulation patterns dominated these periods.

- In contrast, the Dark Ages Cold Period (DAC) period was likely dominated by El Nino, negative NAO and negative (AMO) state-like circulation.

- The boreal east/west ecoregions time series provide evidence for a positive (negative) Pacific North American pattern (PNA) during both DAC and LIA (MWP and RWP) suggesting El Nino (La Nina) states.

- It is likely that minimum solar and high volcanic activity may have contributed to more complex hydroclimate regional patterns during the LIA.

5.5 Summary of Objective 4

In order to achieve objectives 1 to 3, a user-friendly menu-based R application was developed in order to automate pollen-based paleoclimate reconstructions and allow for user-defined research questions. This objective is presented in Appendix B (user manual) and Appendix C (source code). This is slated to be made available to the scientific community through publication in the journal *Computers and Geosciences*.

5.6 Future Research Directions

- In the near future, pollen-based reconstructions could attempt a reconstruction of atmospheric pressure changes (either surface or 850 mb) through time under the PAGES 2k framework.

- The use of these newly-developed reconstructions can be used in paleoclimate data-model comparisons (e.g. PMIP3).

- One main issue with pollen-based reconstruction remains the resolution of most fossil pollen sites; most have century-scale resolution. One way around the issue of resolution is to explore the incorporation of lower-frequency pollen records with high resolution proxies (e.g. PAGES 2k Consortium, 2013; Trouet *et al.*, 2013).

- A potential challenge would be to explore different hydroclimate variables such as the Priestley-Taylor Moisture Index (MIPT), Palmer Drought Severity Index (PDSI) and Standardized Precipitation and Evapotranspiration Index (SPEI) to see how they compare with the reconstructions of ANNP. There are many useful ways to quantify the moisture availability in a region and also at specific sites. These moisture/drought indexes can be helpful to better understand the spatial and temporal changes in hydro climate. A review of these drought indices is given in Heim Jr (2002). These indices are important given that measurements of soil moisture are often sparse (Robock *et al.*, 2000). The most

widely used drought index is the PDSI (Palmer, 1965; Heim Jr, 2002) derived from both temperature and precipitation and is based on the evapotranspiration work of Thornthwaite (1948). However, it is known that the PDSI does have some limitations (Alley, 1984). Guttman (1991) found that the PDSI is more suitable for arid/semi-arid regions. Evapotranspiration in the Palmer model is based on a rather simple method (Penman, 1948). With adequate soil moisture, actual evapotranspiration should equal potential evapotranspiration (Heim Jr, 2002). The relationship between measured/observed climate variables such as total annual precipitation and these moisture/drought indexes is likely much more complex and depends on many different factors such as temperature, solar radiation, wind speed, evaporation rate, vegetation and soil conditions. This has implications for better understanding hydroclimate changes through time and future hydroclimate changes under a warming climate.

- Using the newly developed open-source R program (Appendix B and C) will enable the paleoclimate research community to be able to easily produce North American pollen-based paleoclimate reconstructions based on the specific user-defined parameters. This will facilitate the use of pollen databases as a proxy for paleoclimate research.

References

Alley, W.M. (1984). The Palmer drought severity index: limitations and assumptions. *Journal of Climate and Applied Meteorology* **23**(7), 1100–1109.

Guttman, N.B. (1991). A SENSITIVITY ANALYSIS OF THE PALMER HYDROLOGIC DROUGHT INDEX1. *JAWRA Journal of the American Water Resources Association* **27**(5), 797–807.

Heim Jr, R.R. (2002). A review of twentieth-century drought indices used in the United States. *Bulletin of the American Meteorological Society* **83**(8), 1149–1165.

PAGES 2k Consortium (2013). Continental-scale temperature variability during the past two millennia. *Nature Geoscience* **6**, 339–346.

Palmer, W.C. (1965). *Meteorological drought*. Washington, DC, USA: US Department of Commerce, Weather Bureau.

Penman, H.L. (1948). Natural evaporation from open water, bare soil and grass. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **193**(1032), 120–145.

Robock, A., K.Y. Vinnikov, G. Srinivasan, J.K. Entin, S.E. Hollinger, N.A. Speranskaya, S. Liu, and A. Namkhai (2000). The global soil moisture data bank. *Bulletin of the American Meteorological Society* **81**(6), 1281–1299.

Thornthwaite, C.W. (1948). An approach toward a rational classification of climate. *Geographical Review* **38**(1), 55–94.

Trouet, V., H.F. Diaz, E.R. Wahl, A.E. Viau, R. Graham, N. Graham, and E.R. Cook (2013). A 1500-year reconstruction of annual mean temperature for temperate North America on decadal-to-multidecadal time scales. *Environmental Research Letters* **8**(2), 024008.

Whitmore, J., K. Gajewski, M.C. Sawada, J.W. Williams, B. Shuman, P.J. Bartlein, T. Minckley, A.E. Viau, T. Webb III, S. Shafer, P. Anderson, and L. Brubaker (2005). Modern pollen data from North America and Greenland for multi-scale paleoenvironmental applications. *Quaternary Science Reviews* **24**(16), 1828–1848.

All Thesis References

Alley, W.M. (1984). The Palmer drought severity index: limitations and assumptions. *Journal of Climate and Applied Meteorology* **23**(7), 1100–1109.

Ammann, C. M., F. Joos, D.S. Schimel, B.L. Otto-Bliesner, and R.A. Tomas (2007). Solar influence on climate during the past millennium: Results from transient simulations with the NCAR Climate System Model. *PNAS* **104**(10), 3713–3718.

Anselin, L. (1995). Local Indicators of Spatial Association - LISA. *Geographical Analysis* **27**(2), 93–115.

Atkinson, D.E. and K. Gajewski (2002). High-resolution estimation of summer surface air temperature in the Canadian Arctic Archipelago. *Journal of Climate* **15**(24).

Autio, J and S. Hicks (2004). Annual variations in pollen deposition and meteorological conditions on the fell Aakenusunturi in northern Finland: potential for using fossil pollen as a climate proxy. *Grana* **43**(1), 31–47.

Barnekow, L., N.J. Loader, S. Hicks, C.A. Froyd, and T. Goslar (2007). Strong correlation between summer temperature and pollen accumulation rated for Pinus sylvestris, Picea abies and Betula spp in a high-resolution record from northern Sweden. *Journal of Quaternary Science* **22**(7), 653–658.

Bartlein, P.J., K.H. Anderson, P.M. Anderson, M.E. Edwards, C.J. Mock, R.S. Thompson, R.S. Webb, T. Webb III, and C. Whitlock (1998). Paleoclimate simulations for North America over the past 21,000 years: features of the simulated climate and comparisons with paleoenvironmental data. *Quaternary Science Reviews* **17**(6-7), 549–586.

Bartlein, P.J., S.J. Harrison, S. Brewer, S. Connor, B.A.S. Davis, K. Gajewski, J. Guiot, T.I. Harrison-Prentice, A. Henderson, O. Peyron, I.C. Prentice, M. Scholze, H. Seppä, B. Shuman, S. Sugita, R.S. Thompson, A.E. Viau, J. Williams, and H. Wu (2011). Pollen-based continental climate reconstructions at 6 and 21 ka: a global synthesis. *Climate Dynamics* **37**(3-4), 775–802.

Berger, A. and M. F. Loutre (1991). Insolation values for the climate of the last 10 million years. *Quaternary Science Reviews* **10**(4), 297–317.

Berkeley Earth Surface Temperature (2013). *Berkeley Earth*. url: http://www.berkeleyearth.org.

Birks, H.J.B., O. Heiri, H. Seppå, and A.E. Bjune (2010). Strengths and weaknesses of quantitative climate reconstructions based on late-quaternary biological proxies. *The Open Ecology Journal* **3**, 68–110.

Bond, G., B. Kromer, J. Beer, R. Muscheler, N.E. Evans, W. Showers, S. Hoffman, R. Lotti-Bond, I. Hajdas, and G. Bonani (2001). Persistent solar influence on North Atlantic climate during the Holocene. *Science* **294**(5549), 2130–2136.

Booth, R.K, M. Notaro, S.T. Jackson, and J.E. Kutzbach (2006). Widespread drought episodes in the western Great Lakes region during the past 2000 years: geographic extent and potential mechanisms. *Earth and Planetary Science Letters* **242**(3), 415–427.

Bourgeois, J.C., R.M. Koerner, K. Gajewski, and D.A. Fisher (2000). A Holocene ice-core pollen record from Ellesmere Island, Nunavut, Canada. *Quaternary Research* **54**(2), 275–283.

Braak, C.F.J. ter and S. Juggins (1993). Weighted averaging partial least squares regression (WA-PLS): an improved method for reconstructing environmental variables from species assemblages. *Hydrobiologia* **269**(1), 485–502.

Braconnot, P., B. Otto-Bliesner, S. Harrison, S. Joussaume, J.-Y. Peterchmitt, A. Abe-Ouchi, M. Crucifix, E. Driesschaert, Th. Fichefet, C.D. Hewitt, M. Kageyama, A. Kitoh, A. Lainé, M.-F. Loutre, O. Marti, U. Merkel, G. Ramstein, P. Valdes, S. L. Weber, Y. Yu, and Y. Zhao (2007). Results of PMIP2 coupled simulations of the Mid-Holocene and Last Glacial Maximum–Part 1: experiments and large-scale features. *Climate of the Past* **3**(2), 261–277.

Braconnot, P., S.P. Harrison, M. Kageyama, P.J. Bartlein, V. Masson-Delmotte, A. Abe-Ouchi, B. Otto-Bliesner, and Y. Zhao (2012). Evaluation of climate models using palaeoclimatic data. *Nature Climate Change* **2**(6), 417–424.

Bradley, R.S., M.E. Mann, and M.K. Hughes (1999). Northern hemisphere temperatures during the past millennium: inferences, uncertainties, and limitations. *Geophysical Research Letters* **26**(6), 759–762.

Bradley, R.S., M.K. Hughes, and H.F. Diaz (2003). Climate in Medieval Time. *Science* **302**(2), 404–405.

Bradley, R.S., M. Vuille, M. Werner, R.Y. Chan, T.J. Osborne, K.R. Briffa, M.K. Hughes, and P.D. Jones (2005). Proxy-based Northern Hemisphere surface temperature reconstructions: Sensitivity to method, predictor network, target season, and target domain. *Journal of Climate* **18**(13), 2308–2329.

Briffa, K.R. and T.J. Osborn (2002). Blowing hot and cold. *Science* **295**(5563), 2227–2228.

Briffa, K.R., T.J. Osborn, F.H. Schweingruber, I.C. Harris, P.D. Jones, S.G. Shiyatov, and E.A. Vaganov (2001). Low-frequency temperature variations from a northern tree ring density network. *Journal of Geophysical Research* **106**(D3), 2929–2941.

Broecker, W.S. and J. Donk (1970). Insolation changes, ice volumes, and the O18 record in deep-sea cores. *Reviews of Geophysics* **8**(1), 169–198.

Brohan, P., J.J. Kennedy, I. Harris, S.F.B. Tett, and P.D. Jones (2006). Uncertainty estimates in regional and global observed temperature changes: A new data set from 1850. *Journal of Geophysical Research: Atmospheres (1984–2012)* **111**(D12).

Brown, K.J., J.S. Clark, E.C. Grimm, J.J. Donovan, P.G. Mueller, B.C.S. Hansen, and I. Stefanova (2005). Fire cycles in North American interior grasslands and their relation to prairie drought. *Proceedings of the National Academy of Sciences of the United States of America* **102**(25), 8865–8870.

Bryson, R.A. (1966). Air masses, streamlines, and the boreal forest. *Geographical Review* **8**(3), 228–269.

Cavazos, T. and B.C. Hewitson (2005). Performance of NCEP-NCAR reanalysis variables in statistical downscaling of daily precipitation. *Climate Research* **28**, 95–107.

Cheddadi, R., G. Yu, J. Guiot, S. P. Harrison, and I.C. Prentice (1997). The climate of Europe 6000 years ago. *Climate Dynamics* **13**(1), 1–9.

Christiansen, B. (2011). Reconstructing the NH Mean Temperature: Can Underestimation of Trends and Variability Be Avoided? *Journal of Climate* **24**(3), 674–692.

Christiansen, B. and F. C. Ljungqvist (2011). Reconstruction of the extratropical NH mean temperature over the last millennium with a method that preserves low-frequency variability. *Journal of Climate* **24**(3), 6013–6034.

— (2012). The extra-tropical Northern Hemisphere temperature in the last two millennia: reconstructions of low-frequency variability. *Climate of the Past* **8**(2), 765–786.

Clark, J.S., E.C. Grimm, J.J. Donovan, S.C. Fritz, D.R. Engstrom, and J.E. Almendinger (2002). Drought cycles and landscape responses to past aridity on prairies of the northern Great Plains, USA. *Ecology* **83**(3), 595–601.

Clark, P.U., A.J. Weaver (coordinating lead authors), E. Brook, E.R. Cook, T.L. Delworth, and K. Steffen (chapter lead authors) (2008). *Abrupt Climate Change. A report by the U.S. Climate Change Science Program and the Subcommittee*

on Global Change Research. research report. Reston, VA, USA: U.S. Geological Survey, p. 244.

CLIMAP Project Members (1981). *Seasonal reconstructions of the Earth's surface at the last glacial maximum*. Geological Society of America Map and Chart Series MC-36.

COHMAP Members (1988). Climatic changes of the last 18,000 years: Observations and model simulations. *Science* **241**, 1043–1052.

Cook, E.R., C.A. Woodhouse, C.M. Eakin, D.M. Meko, and D.W. Stahle (2004). Long-term aridity changes in the western United States. *Science* **306**(5698), 1015–1018.

Cowtan, K. and R.G. Way (2013). Coverage bias in the HadCRUT4 temperature series and its impact on recent temperature trends. *Quarterly Journal of the Royal Meteorological Society*.

Crowley, T. J. (2000). Causes of climate change over the past 1000 years. *Science* **289**(5477), 270–277.

Crowley, T. J. and T. Lowery (2000). How Warm Was the Medieval Warm Period? *Ambio* **29**(1), 51–54.

Dai, A. (2011). Drought under global warming: a review. *Wiley Interdisciplinary Reviews: Climate Change* **2**(1), 45–65.

D'Arrigo, R., R. Wilson, and G. Jacoby (2006). On the long-term context for late twentieth century warming. *Journal of Geophysical Reseach* **111**(D3).

D'Arrigo, Rosanne D and Gordon C Jacoby (1991). A 1000-year record of winter precipitation from northwestern New Mexico, USA: a reconstruction from tree-rings and its relation to El Niño and the Southern Oscillation. *The Holocene* **1**(2), 95–101.

Dee, D.P., S.M. Uppala, A.J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M.A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A.C.M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A.J. Geer, L. Haimberger, S.B. Healy, H. Hersbach, E. Hólm, L. Isaksen, P. Källberg, M. Köhler, M. Matricardi, A.P. McNally, B.M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and Vitart F. (2011). The ERA-Interim reanalysis: Configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society* **137**(656), 553–597.

Dettinger, M.D., D.R. Cayan, H.F. Diaz, and D.M. Meko (1998). North–South Precipitation Patterns in Western North America on Interannual-to-Decadal Timescales. *Journal of Climate* **11**(12).

Diaz, H.F., R. Trigo, M.K. Hughes, M.E. Mann, E. Xoplaki, and D. Barriopedro (2011). Spatial and temporal characteristics of climate in Medieval times revisited. *Bulletin of the American Meteorological Society* **92**(11), 1487–1500.

Edwards, T.L., M. Crucifix, and S.P. Harrison (2007). Using the past to constrain the future: How the paleorecord can improve estimates of global warming. *Progress in Physical Geography* **31**(5), 481–500.

Esper, J., E.R. Cook, and F.H. Schweingruber (2002). Low-frequency signals in long tree-ring chronologies for reconstructing past temperature variability. *Science* **295**(5563), 2250–2253.

Esper, J., D.C. Frank, M. Timonen, E. Zorita, R.J.S Wilson, J. Luterbacher, S. Holzkämper, N. Fischer, S. Wagner, D. Nievergelt, A. Verstege, and U. Büntgen (2012). Orbital forcing of tree-ring data. *Nature Climate Change* **2**(12), 862–866.

Fedorova, I.T., Y.A. Volkova, and D.L. Varlyguin (1994). World Vegetation Cover: Digital raster data on a 30-minute Cartesian orthonormal geodetic (lat./long.) 1,080× 2,160 grid. *Global Ecosystems Database Version* **2**.

Feng, S., Q. Hu, and R.J. Oglesby (2009). AMO-like variations of holocene sea surface temperatures in the North Atlantic Ocean. *Climate of the Past Discussions* **5**(6), 2465–2496.

Fisher, D.A. and R.M. Koerner (1994). Signal and noise in four ice-core records from the Agassiz Ice Cap, Ellesmere Island, Canada: details of the last millennium for stable isotopes, melt and solid conductivity. *The Holocene* **4**(2), 113–120.

Fisher, D.A., R.M. Koerner, and N. Reeh (1995). Holocene climatic records from Agassiz ice cap, Ellesmere Island, NWT, Canada. *The Holocene* **5**(1), 19–24.

Fisher, D.A., J. Zheng, D. Burgess, C. Zdanowicz, C. Kinnard, M. Sharp, and J. Bourgeois (2012). Recent melt rates of Canadian arctic ice caps are the highest in four millennia. *Global and Planetary Change* **84**, 3–7.

Fortin, M.-C. and K. Gajewski (2012). Potential problems with the use of gridded climate data in regional quantitative paleoenvironmental studies from data-poor regions. *Journal of Paleolimnology* **48**(3), 641–650.

Francis, J.A. and S.J. Vavrus (2012). Evidence linking Arctic amplification to extreme weather in mid-latitudes. *Geophysical Research Letters* **39**(6).

Fritts, H.C. (1991). *Reconstructing large-scale climatic patterns from tree-ring data*. Tucson, AZ (United States); University of Arizona Press.

Fritz, S.C., D.R. Engstrom, and B.J. Haskell (1994). 'Little Ice Age'aridity in the North American Great Plains: a high-resolution reconstruction of salinity fluctuations from Devils Lake, North Dakota, USA. *The Holocene* **4**(1), 69–73.

Fritz, S.C., E. Ito, Z. Yu, K.R. Laird, and D.R. Engstrom (2000). Hydrologic variation in the northern Great Plains during the last two millennia. *Quaternary Research* **53**(2), 175–184.

Gajewski, K. (1983). On the interpretation of climatic change from the fossil record: Climatic change in central and eastern United States for the past 2000 years estimated from pollen data. PhD thesis. University of Wisconsin-Madison.

— (1987). Climatic impacts on the vegetation of eastern North America during the past 2000 years. *Vegetatio* **68**(3), 179–190.

— (1988). Late Holocene climate changes in eastern North America estimated from pollen data. *Quaternary Research* **29**(3), 255–262.

— (1993). The role of paleoecology in the study of global climatic change. *Review of palaeobotany and palynology* **79**(1), 141–151.

— (2006). Essai: Is Arctic Palynology a "Blunt Instrument"? *Géographie physique et Quaternaire* **60**(2), 95–102.

— (2008). The Global Pollen Database in biogeographical and palaeoclimatic studies. *Progress in Physical Geography* **32**(4), 379–402.

Gajewski, K. and A.E. Viau (2011). Abrupt climate changes during the Holocene across North America from pollen and paleolimnological records. *Abrupt Climate Change: Mechanisms, Patterns, and Impacts, Geophysical Monograph Series*. Ed. by L. Polyak H. Rashid and E. Mosley-Thompson. Washington, D.C., USA: AGU, 161–171.

Gajewski, K., A.E. Viau, and M.C. Sawada (2007). Millennial-scale climate variations in the Holocene - the terrestrial record. *Climate Variability and Change: Past, Present and Future*. Ed. by G. Kutzbach. Madison, Wisconsin, USA: University of Wisconsin, 133–154.

Girardin, M-P., J.C. Tardif, M.D. Flannigan, and Y. Bergeron (2006). Synoptic-scale atmospheric circulation and boreal Canada summer drought variability of the past three centuries. *Journal of Climate* **19**(10).

Gonzales, L.M., J.W. Williams, and E.C. Grimm (2009). Expanded response-surfaces: a new method to reconstruct paleoclimates from fossil pollen assemblages that lack modern analogues. *Quaternary Science Reviews* **28**(27), 3315–3332.

Graham, N.E., C.M. Ammann, D. Fleitmann, K.M. Cobb, and J. Luterbacher (2011). Support for global climate reorganization during the "Medieval Climate Anomaly". *Climate dynamics* **37**(5–6), 1217–1245.

Grimm, E.C. (2008). *Neotoma database*. url: http://www.neotomadb.org.

Grimm, E.C. (2011). *North American Pollen database. Data archived at IGBP PAGES/-World Data Center–A for Paleoclimatology, NOAA/NGDC Paleoclimatology Program, Boulder, CO, USA.* url: http://www.ncdc.noaa.gov/paleo.

Grimm, E.C. and G.L. Jacobson Jr (2007). Millennial-scale climate variations in the Holocene - the terrestrial record. *Climate Variability and Change: Past, Present and Future*. Ed. by G. Kutzbach. Madison, Wisconsin, USA: University of Wisconsin, 133–154.

Grimm, E.C., G.L. Jacobson Jr, W.A. Watts, B.C.S. Hansen, and K.A. Maasch (1993). A 50,000-Year Record of Climate Oscillations from Florida and its Temporal Correlation. *Science* **61**, 9.

Grimm, E.C., J. Keltner, R. Cheddadi, S. Hicks, A.M. Lézineé, J.C. Berrio, and J.W. Wiliams (2007). Databases and their application. *Encyclopedia of Quaternary Science*. Ed. by S. A. Elias. **3**. Amsterdam: Elsevier, 2521–2528.

Grissino-Mayer, H.D. *et al.* (1996). A 2129-year reconstruction of precipitation for northwestern New Mexico, USA. *Tree rings, environment, and humanity*, 191–204.

Guttman, N.B. (1991). A SENSITIVITY ANALYSIS OF THE PALMER HYDROLOGIC DROUGHT INDEX1. *JAWRA Journal of the American Water Resources Association* **27**(5), 797–807.

Hansen, J., M. Sato, P. Kharecha, D. Beerling, R. Berner, V. Masson-Delmotte, M. Pagani, M. Raymo, D.L. Royer, and J.C. Zachos (2008). Target atmospheric CO2: Where should humanity aim? *arXiv preprint arXiv:0804.1126*.

Harris, I., P.D. Jones, T.J. Osborn, and D.H. Lister (2013). Updated high-resolution grids of monthly climatic observations–the CRU TS3. 10 Dataset. *International Journal of Climatology*.

Harrison, S.P., D. Jolly, F. Laarif, A. Abe-Ouchi, B. Dong, K. Herterich, C. Hewitt, S. Joussaume, J.E. Kutzbach, J. Mitchell, N. de Noblet, and Valdes P. (1998). Intercomparison of simulated global vegetation distributions in response to 6 kyr BP orbital forcing. *Journal of Climate* **11**(11), 2721–2742.

Hegerl, G.C., T.J. Crowley, M. Allen, W.T. Hyde, H.N. Pollack, J. Smerdon, and E. Zorita (2007). Detection of human influence on a new, validated 1500-year temperature reconstruction. *Journal of Climate* **20**(4), 650–666.

Heim Jr, R.R. (2002). A review of twentieth-century drought indices used in the United States. *Bulletin of the American Meteorological Society* **83**(8), 1149–1165.

Herweijer, C., R. Seager, E.R. Cook, and J. Emile-Geay (2007). North American droughts of the last millennium from a gridded network of tree-ring data. *Journal of Climate* **20**(7), 1353–1376.

Hicks, S. (2001). The use of annual arboreal pollen deposition values for delimiting tree-lines in the landscape and exploring models of pollen dispersal. *Review of Palaeobotany and Palynology* **117**(1), 1–29.

Hijmans, R.J., S.E. Cameron, J.L. Parra, P. G. Jones, and A. Jarvis (2005). Very high resolution interpolated climate surfaces for global land areas. *International journal of climatology* **25**(15), 1965–1978.

Hughes, M.K. and H.F. Diaz (1994). Was there a "Medieval Warm Period", and if so, where and when? *Climatic Change* **26**(2–3), 109–142.

Huusko, A. and S. Hicks (2009). Conifer pollen abundance provides a proxy for summer temperature: evidence from the latitudinal forest limit in Finland. *Journal of Quaternary Science* **24**(5), 522–528.

IPCC (2007). *Climate Change 2007, The Physical Science Basis. Fourth Assessment Report*. Ed. by S. Solomon, D. Manning, Z. Chen, M. Marquis, K.B. Averyt, M. Tignor, and H. Miller. Cambridge, UK and New York, NY, USA: Cambridge University Press, p. 996.

— (2013). *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by T.F. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P.M. Midgley. Cambridge, UK and New York, NY, USA: Cambridge University Press, p. 1552.

Isaaks, E.H. and R.H. Srivastava (1989). *An Introduction to Applied Geostatistics*. New York, USA: Oxford University Press.

Jackson, S.T. and J.W. Williams (2004). Modern analogs in Quaternary paleoecology: here today, gone yesterday, gone tomorrow? *Annual Review of Earth and Planetary Sciences* **32**, 495–537.

Jones, P.D. and M.E. Mann (2004). Climate over past millennia. *Reviews of Geophysics* **42**(2), RG2002.

Jones, P.D., K.R. Briffa, T.P. Barnett, and S.F.B. Tett (1998). High-resolution palaeoclimatic records for the last millennium: interpretation, integration and comparison with General Circulation Model control-run temperatures. *The Holocene* **8**(4), 455–471.

Jones, P.D., K.R. Briffa, T.J. Osborn, J.M. Lough, T.D. Van Ommen, B.M. Vinther, J. Luterbacher, E.R. Wahl, F.W. Zwiers, M.E. Mann, C.M. Schmidt G.A.and Ammann, B.M. Buckley, K.M. Cobb, J. Esper, H. Goosse, N. Graham, E. Jansen, T. Kiefer, C. Kull, M. Kttel, E. Mosley-Thompson, J.T Overpeck, N. Riedwyl, A.W. Schulz M. Tudhope, R. Villalba, H. Wanner, E. Wolff, and E. Xoplaki (2009).

High-resolution palaeoclimatology of the last millennium: a review of current status and future prospects. *The Holocene* **19**(1), 3–49.

Joussaume, S. and K. E. Taylor (1999). The Paleoclimate Modeling Intercomparison Project, in Proceedings of the Third PMIP Workshop, Canada. WCRP-111, WMO/TD-1007. Available at http://www-lsce.cea.fr/pmip/. Geneva, Switzerland: World Meteorological Organization, 9–24.

Juggins, S. and H.J.B. Birks (2012). Quantitative environmental reconstructions from biological data. *Tracking environmental change using lake sediments*, 431–494.

Kalnay, E., M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K.C. Mo, C. Roperlewski, J. Wang, A. Leetmaa, R. Reynolds, R. Jenne, and D. Joseph (1996). The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American meteorological Society* **77**(3), 437–471.

Kaufman, D.S., D.P. Schneider, N.P. McKay, C.M. Ammann, R.S. Bradley, K.R. Briffa, G.H. Miller, B.L. Otto-Bliesner, J.T. Overpeck, B.M. Vinther, and Arctic Lakes 2k Project Members (2009). Recent warming reverses long-term Arctic cooling. *Science* **325**(5945), 1236–1239.

Kerwin, M.W., J.T. Overpeck, R.S. Webb, and K.H. Anderson (2004). Pollen-based summer temperature reconstructions for the eastern Canadian boreal forest, subarctic, and Arctic. *Quaternary Science Reviews* **23**(18), 1901–1924.

Kinnard, C., C.M. Zdanowicz, D.A. Fisher, E. Isaksson, A. de Vernal, and L.G. Thompson (2011). Reconstructed changes in Arctic sea ice over the past 1,450 years. *Nature* **479**(7374), 509–512.

Kistler, R., E. Kalnay, Collins W., S. Suranjana, G. White, J. Woollen, Muthuvel C., W. Ebisuzaki, M. Kanamitsu, V. Kousky, H. van den Dool, R. Jenne, and M. Fiorino (2001). The NCEP-NCAR 50-year reanalysis: Monthly means CD-ROM and documentation. *Bulletin of the American Meteorological society* **82**(2), 247–267.

Koerner, R.M. and D.A. Fisher (1990). A record of Holocene summer climate from a Canadian high-Arctic ice core. *Nature* **343**, 630–631.

Krivoruchko, K. and A. Gribov (2014). Pragmatic Bayesian Kriging for Non-Stationary and Moderately Non-Gaussian Data. *Mathematics of Planet Earth*. Ed. by E. Pardo-Igúzquiza, C. Guardiola-Albert, J. Heredia, L. Moreno-Merino, J.J. Durán, and J.A. Vargas-Guzmán. Lecture Notes in Earth System Sciences, 61–64.

Ladd, M.J. and K. Gajewski (2010). The North American summer Arctic front during 1948-2007. *International Journal of Climatology* **30**(6), 874–883.

Lamb, H.H. (1965). The early medieval warm epoch and its sequel. *Palaeogeography, Palaeoclimatology, Palaeoecology* **1**, 13–37.

Leathers, D.J., B. Yarnal, and M.A. Palecki (1991). The Pacific/North American teleconnection pattern and United States climate. Part I: Regional temperature and precipitation associations. *Journal of Climate* **4**(5), 517–528.

Lenton, T.M, H. Held, E. Kriegler, J.W. Hall, W. Lucht, S. Rahmstorf, and H.J. Schellnhuber (2008). Tipping elements in the Earth's climate system. *Proceedings of the National Academy of Sciences* **105**(6), 1786–1793.

Lindsay, R., M. Wensnahan, A. Schweiger, and J. Zhang (2014). Evaluation of seven different atmospheric reanalysis products in the Arctic. *Journal of Climate*(2014).

Ljungqvist, F.C. (2009). Temperature proxy records covering the last two millennia: a tabular and visual overview. *Geografiska Annaler: Series A, Physical Geography* **91**(1), 11–29.

MacDonald, G.M. (2007). Severe and sustained drought in southern California and the West: Present conditions and insights from the past on causes and impacts. *Quaternary International* **173**, 87–100.

MacDonald, G.M. and R.A. Case (2005). Variations in the Pacific Decadal Oscillation over the past millennium. *Geophysical Research Letters* **32**(8).

Mann, M.E and P.D. Jones (2003). Global surface temperatures over the past two millennia. *Geophysical Research Letters* **30**(15).

Mann, M.E., R.S. Bradley, and M.K. Hughes (1998). Global-scale temperature patterns and climate forcing over the past six centuries. *Nature* **392**(6678), 779–787.

Mann, M.E., S. Rutherford, E.R. Wahl, and C. Ammann (2007). Robustness of proxy-based climate field reconstruction methods. *Journal of Geophysical Research* **112**(D12), D12109.

Mann, M.E, M.K. Zhang Z.and Hughes, R.S Bradley, S.K. Miller, S. Rutherford, and F. Ni (2008). Proxy-based reconstructions of hemispheric and global surface temperature variations over the past two millennia. *Proceedings of the National Academy of Sciences* **105**(36), 13252–13257.

Mann, M.E, Z. Zhang, S. Rutherford, R.S. Bradley, M.K. Hughes, D. Shindell, C. Ammann, G. Faluvegi, and F. Ni (2009a). Global signatures and dynamical origins of the Little Ice Age and Medieval Climate Anomaly. *Science* **326**(5957), 1256–1260.

Mann, M.E., R.S. Bradley, and M.K. Hughes (2009b). Reply to McIntyre and McKitrick: Proxy-based temperature reconstructions are robust. *Proceedings of the National Academy of Sciences* **106**(6), E11.

McCabe, G.J., M.A. Palecki, and J.L. Betancourt (2004). Pacific and Atlantic Ocean influences on multidecadal drought frequency in the United States. *Proceedings of the National Academy of Sciences* **101**(12), 4136–4141.

McCabe, G.J., J.L. Betancourt, S.T. Gray, M.A. Palecki, and H.G. Hidalgo (2008). Associations of multi-decadal sea-surface temperature variability with US drought. *Quaternary International* **188**(1), 31–40.

McIntyre, S. and R. McKitrick (2009). Proxy inconsistency and other problems in millennial paleoclimate reconstructions. *Proceedings of the National Academy of Sciences* **106**(6), E10.

Menne, M.J., C.N. Williams Jr, and R.S. Vose (2009). The US Historical Climatology Network monthly temperature data, version 2. *Bulletin of the American Meteorological Society* **90**(7).

Mesinger, F., E. DiMego G.and Kalnay, K. Mitchell, P.C. Shafran, W. Ebisuzaki, D. Jovic, J. Woollen, E. Rogers, E.H. Berbery, M.B. Ek, Y. Fan, R. Grumbine, W. Higgins, Hong Li, Y. Lin, G. Manikin, D. Parrish, and W. Shi (2006). North American regional reanalysis. *Bulletin of the American Meteorological Society* **87**(3), 343–360.

Miller, H.J. (2004). Tobler's First Law and Spatial Analysis. *Annals of the Association of American Geographers* **94**(2), 284–289.

Moberg, A., D.M. Sonechkin, K. Holmgren, N.M. Datsenko, and W. Karlén (2005). Highly variable Northern Hemisphere temperatures reconstructed from low-and high-resolution proxy data. *Nature* **433**(7026), 613–617.

Müller, W.A. and E. Roeckner (2006). ENSO impact on midlatitude circulation patterns in future climate change projections. *Geophysical research letters* **33**(5).

National Research Council (2006). *Surface Temperature Reconstructions for the Last 2,000 Years*. Washington, D.C., USA: National Academies Press.

Neelin, J. D. (2011). *Climate Change and Climate Modeling*. Cambridge. New York, p. 282.

New, M., D. Lister, M. Hulme, and I. Makin (2002). A high-resolution data set of surface climate over global land areas. *Climate research* **21**(1), 1–25.

Nigam, S., B. Guan, and A. Ruiz-Barradas (2011). Key role of the Atlantic Multidecadal Oscillation in 20th century drought and wet periods over the Great Plains. *Geophysical Research Letters* **38**(16).

Overpeck, J.T., T. Webb III, and I.C. Prentice (1985). Quantitative interpretation of fossil pollen spectra: dissimilarity coefficients and the method of modern analogs. *Quaternary Research* **23**(1), 87–108.

PAGES 2k Consortium (2013). Continental-scale temperature variability during the past two millennia. *Nature Geoscience* **6**, 339–346.

Paleoclimate Modelling Intercomparison Project Phase III (2012). *PMIP 3 Project Home Page*. url: http://pmip3.lsce.ipsl.fr.

Palmer, W.C. (1965). *Meteorological drought*. Washington, DC, USA: US Department of Commerce, Weather Bureau.

Peixoto, J.P. and A.H. Oort (1992). *Physics of Climate*. New York, USA: American Institute of Physics.

Penman, H.L. (1948). Natural evaporation from open water, bare soil and grass. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **193**(1032), 120–145.

Peros, M.C., K. Gajewski, and A.E. Viau (2008). Continental-scale tree population response to rapid climate change, competition and disturbance. *Global Ecology and Biogeography* **17**(5), 658–669.

Pinot, S., G. Ramstein, S.P. Harrison, I.C. Prentice, J. Guiot, M. Stute, and S. Joussaume (1999). Tropical paleoclimates at the Last Glacial Maximum: comparison of Paleoclimate Modeling Intercomparison Project (PMIP) simulations and paleodata. *Climate Dynamics* **15**(11), 857–874.

Pinto, J.G., M. Reyers, and U. Ulbrich (2011). The variable link between PNA and NAO in observations and in multi-century CGCM simulations. *Climate dynamics* **36**(1-2), 337–354.

Reimer, P.J., M.G.L. Baillie, E. Bard, A. Bayliss, J.W. Beck, C.J.H. Bertrand, P.G. Blackwell, C.E. Buck, G.S. Burr, K.B. Cutler, P.E. Damon, R.L. Edwards, R.G. Fairbanks, M. Friedrich, T.P. Guilderson, A.G. Hogg, K.A. Hughen, B. Kromer, G. McCormac, S. Manning, C.B. Ramsey, R.W. Reimer, S. Remmele, J.R. Southon, M. Stuiver, S. Talamo, F.W. Taylor, J. van der Plicht, and C.E. Weyhenmeyer (2004). IntCal04 terrestrial radiocarbon age calibration, 26e0 ka BP. *Radiocarbon* **46**.

Robock, A., K.Y. Vinnikov, G. Srinivasan, J.K. Entin, S.E. Hollinger, N.A. Speranskaya, S. Liu, and A. Namkhai (2000). The global soil moisture data bank. *Bulletin of the American Meteorological Society* **81**(6), 1281–1299.

Rohde, R., R.A. Muller, R. Jacobsen, E. Muller, S. Perlmutter, A. Rosenfeld, J. Wurtele, D. Groom, and C. Wickham (2012). A New Estimate of the average Earth surface land temperature spanning 1753 to 2011, Geoinfor. *Geostat: An Overview* **1**(1).

Rohde, R., R.A. Muller, R. Jacobsen, S. Perlmutter, A. Rosenfeld, J. Wurtele, J. Curry, C. Wickham, and S. Mosher (2013). Berkeley Earth Temperature Averaging Process. *Geoinfor Geostat: An Overview* **1**(2).

Royer, A. and S. Poirier (2010). Surface temperature spatial and temporal variations in North America from homogenized satellite SMMR-SSM/I microwave measurements and reanalysis for 1979–2008. *Journal of Geophysical Research: Atmospheres (1984–2012)* **115**(D8).

Sawada, M., A.E. Viau, G. Vettoretti, W.R. Peltier, and K. Gajewski (2004). Comparison of North-American pollen-based temperature and global lake-status with CCCma AGCM2 output at 6ka. *Quaternary Science Reviews* **23**(3), 225–244.

Sawada, M.C. (2006). An open source implementation of the Modern Analog Technique (MAT) within the R computing environment. *Computers and Geosciences* **32**(6), 818–833.

Sawada, M.C., K. Gajewski, A. de Vernal, and P. Richard (1999). Comparison of marine and terrestrial Holocene climatic reconstructions from northeastern North America. *The Holocene* **9**(3), 267–277.

Screen, J.A. and I. Simmonds (2011). Erroneous Arctic Temperature Trends in the ERA-40 Reanalysis: A Closer Look. *Journal of Climate* **24**(10).

Seager, R. and G.A. Vecchi (2010). Greenhouse warming and the 21st century hydroclimate of southwestern North America. *Proceedings of the National Academy of Sciences* **107**(50), 21277–21282.

Shi, F., B. Yang, F.C. Ljungqvist, and F. Yang (2012). Multi-proxy reconstruction of Arctic summer temperatures over the past 1400 years. *Climate Research* **54**, 113–128.

Shi, F., B. Yang, A. Mairesse, L. von Gunten, J. Li, A. Bräuning, F.i Yang, and X. Xiao (2013). Northern Hemisphere temperature reconstruction during the last millennium using multiple annual proxies. *Climate Research* **56**, 231–244.

Shindell, D.T., G.A. Schmidt, R.L. Miller, and M.E. Mann (2003). Volcanic and solar forcing of climate change during the preindustrial era. *Journal of Climate* **16**(24).

Shinker, J.J., P.J. Bartlein, and B. Shuman (2006). Synoptic and dynamic climate controls of North American mid-continental aridity. *Quaternary Science Reviews* **25**(13), 1401–1417.

Sokal, R.R. (1979). Ecological parameters inferred from spatial correlograms. *Contemporary quantitative ecology and related ecometrics*, 167–196.

Spenke, M., C. Beilken, and T. Berlage (1994). Modern climatic data for paleoenvironmental interpretations. *American Quaternary Association Program and Abstracts of the 13th Biennial Meeting, 19-22 June*, 197.

Stahle, D.W. and M.K. Cleaveland (1992). Reconstruction and analysis of spring rainfall over the southeastern US for the past 1000 years. *Bulletin of the American Meteorological Society* **73**(12), 1947–1961.

Stahle, D.W., E.R. Cook, M.K. Cleaveland, M.D. Therrell, D.M. Meko, H.D. Grissino-Mayer, E. Watson, and B.H. Luckman (2000). Tree-ring data document 16th century megadrought over North America. *EOS, Transactions American Geophysical Union* **81**(12), 121–125.

Stahle, D.W., F. K. Fye, E.R. Cook, and R.D. Griffin (2007). Tree-ring reconstructed megadroughts over North America since AD 1300. *Climatic change* **83**(1–2), 133–149.

Stine, S. (1994). Extreme and persistent drought in California and Patagonia during mediaeval time. *Nature* **369**(6481), 546–549.

Stuiver, M., P. J. Reimer, E. Bard, J. W. Beck, G. S. Burr, K. A. Hughen, B. Kromer, G. McCormac, J. van der Plicht, and M. Spurk (1998). IntCal98 radiocarbon age calibration. *Radiocarbon* **24**(3), 1041.

Thornthwaite, C.W. (1948). An approach toward a rational classification of climate. *Geographical Review* **38**(1), 55–94.

Trenberth, K.E., T. Koike, and K. Onogi (2008). Progress and prospects for reanalysis for weather and climate. *Eos, Transactions American Geophysical Union* **89**(26), 234–235.

Trenberth, K.E., A. Dai, G. van der Schrier, P.D. Jones, J. Barichivich, K.R. Briffa, and J. Sheffield (2014). Global warming and changes in drought. *Nature Climate Change* **4**(1), 17–22.

Trouet, V. and A.H. Taylor (2010). Multi-century variability in the Pacific North American circulation pattern reconstructed from tree rings. *Climate dynamics* **35**(6), 953–963.

Trouet, V. and G.J. Van Oldenborgh (2013). KNMI Climate Explorer: a web-based research tool for high-resolution paleoclimatology. *Tree-Ring Research* **69**(1), 3–13.

Trouet, V., J. Esper, N.E. Graham, A. Baker, J.D. Scourse, and D.C. Frank (2009). Persistent positive North Atlantic Oscillation mode dominated the medieval climate anomaly. *Science* **324**(5923), 78–80.

Trouet, V., H.F. Diaz, E.R. Wahl, A.E. Viau, R. Graham, N. Graham, and E.R. Cook (2013). A 1500-year reconstruction of annual mean temperature for temperate North America on decadal-to-multidecadal time scales. *Environmental Research Letters* **8**(2), 024008.

University Corporation for Atmospheric Research (2013). *Climate Data Guide from NCAR's Climate Analysis Section — Climate Data Strengths, Limitations, and Applications — home*. url: http://climatedataguide.ucar.edu.

Uppala, S.M., P.W. Källberg, A.J. Simmons, U. Andrae, V. Bechtold, M. Fiorino, J.K. Gibson, J. Haseler, A. Hernandez, G.A. Kelly, X. Li, K. Onogi, S. Saarinen, N. Sokka, R. P. Allan, E. Andersson, K. Arpe, M. A. Balmaseda, A. C. M. Beljaars, L. Van De Berg, J. Bidlot, N. Bormann, S. Caires, F. Chevallier, A. Dethof, M. Dragosavac, M. Fisher, M. Fuentes, S. Hagemann, E. Hólm, B. J. Hoskins, L. Isaksen, P. A. E. M. Janssen, R. Jenne, A. P. Mcnally, J.-F. Mahfouf, J.-J. Morcrette, N. A. Rayner, R. W. Saunders, P. Simon, A. Sterl, K. E. Trenberth, A. Untch, D. Vasiljevic, P. Viterbo, and J. Woollen (2005). The ERA-40 re-analysis. *Quarterly Journal of the Royal Meteorological Society* **131**(612), 2961–3012.

Viau, A., M. Ladd, and K. Gajewski (2012). The climate of North America during the past 2000 years reconstructed from pollen data. *Global and Planetary Change* **84**, 75–83.

Viau, A.E. and K. Gajewski (2001). Holocene variations in the global hydrological cycle quantified by objective gridding of lake level databases. *Journal of Geophysical Research: Atmospheres (1984–2012)* **106**(D23), 31703–31716.

— (2009). Reconstructing millennial-scale, regional paleoclimates of boreal Canada during the Holocene. *Journal of Climate* **22**(2), 316–330.

Viau, A.E., K. Gajewski, P. Fines, D.E. Atkinson, and M.C. Sawada (2002). Widespread evidence of 1500 yr climate variability in North America during the past 14 000 yr. *Geology* **30**(5), 455–458.

Viau, A.E., K. Gajewski, M.C. Sawada, and P. Fines (2006). Millennial-scale temperature variations in North America during the Holocene. *Journal of Geophysical Research: Atmospheres (1984-2012)* **111**(D9).

Viau, A.E., K. Gajewski, M.C. Sawada, and J. Bunbury (2008). Low-and high-frequency climate variability in eastern Beringia during the past 25 000 years. *Canadian Journal of Earth Sciences* **45**(11), 1435–1453.

Wahl, E.R., H.F. Diaz, and C. Ohlwein (2012). A pollen-based reconstruction of summer temperature in central North America and implications for circulation patterns during medieval times. *Global and Planetary Change* **84**, 66–74.

Wang, J., J. Emile-Geay, J. E. Smerdon, D. Guillot, and B. Rajaratnam (2014). Evaluating climate field reconstruction techniques using improved emulations of real-world conditions. *Climate of the Past* **10**(1), 1–19.

Wanner, H., J. Beer, J. Bütikofer, T.J. Crowley, U. Cubasch, J. Flückiger, H. Goosse, M. Grosjean, F. Joos, J.O. Kaplan, M. Küttel, S.A. Müller, I.C. Prentice, O.

Solomina, T.F. Sticker, P. Trasov, M. Wagner, and M. Widmann (2008). Mid-to Late Holocene climate change: an overview. *Quaternary Science Reviews* **27**(19), 1791–1828.

Webb III, T. (1986). Is vegetation in equilibrium with climate? How to interpret late-Quaternary pollen data. *Vegetatio* **67**(2), 75–91.

Whitmore, J., K. Gajewski, M.C. Sawada, J.W. Williams, B. Shuman, P.J. Bartlein, T. Minckley, A.E. Viau, T. Webb III, S. Shafer, P. Anderson, and L. Brubaker (2005). Modern pollen data from North America and Greenland for multi-scale paleoenvironmental applications. *Quaternary Science Reviews* **24**(16), 1828–1848.

Williams, J. W., P. J. Bartlein, and T. Webb III (1999). Data-model comparisons for eastern North America - inferred biomes and climate values from pollen data. Proceedings of the Third Paleoclimate Modeling Intercomparison Project Workshop, Oct. 4-8, 1999. Montreal, Canada. Ed. by P. Braconnot. WCRP-111, WMO/TD-No. 1007. 4–8.

Williams, J.W. (2003). Variations in tree cover in North America since the last glacial maximum. *Global and Planetary Change* **35**(1), 1–23.

Williams, J.W. and B. Shuman (2008). Obtaining accurate and precise environmental reconstructions from the modern analog technique and North American surface pollen dataset. *Quaternary Science Reviews* **27**(7), 669–687.

Williams, J.W., S.T. Jackson, and J.E. Kutzbach (2007). Projected distributions of novel and disappearing climates by 2100 AD. *Proceedings of the National Academy of Sciences* **104**(14), 5738–5742.

Wilson, R., .R D'Arrigo, .B Buckley, U. Büntgen, J. Esper, D. Frank, B. Luckman, S. Payette, R. Vose, and D. Youngblut (2007). A matter of divergence: tracking recent warming at hemispheric scales using tree ring data. *Journal of Geophysical Research: Atmospheres (1984–2012)* **112**(D17).

Wilson, R., E. Cook, R. D'Arrigo, N. Riedwyl, M.N. Evans, A. Tudhope, and R. Allan (2010). Reconstructing ENSO: the influence of method, proxy data, climate forcing and teleconnections. *Journal of Quaternary Science* **25**(1), 62–78.

Woodhouse, C.A. and J.T. Overpeck (1998). 2000 years of drought variability in the central United States. *Bulletin of the American Meteorological Society* **79**(12), 2693–2714.

Wright Jr., H. E, J. E. Kutzbach, T. Webb III, W.F. Ruddiman, F. A. Street-Perrott, and P. J. Bartlein (1993). *Global Climates Since the Last Glacial Maximum*. Minnesota, USA: University of Minnesota Press.

Yu, B. and F.W. Zwiers (2007). The impact of combined ENSO and PDO on the PNA climate: a 1,000-year climate modeling study. *Climate dynamics* **29**(7-8), 837–851.

Zheng, W., P. Braconnot, E. Guilyardi, U. Merkel, and Y. Yu (2008). ENSO at 6ka and 21ka from ocean–atmosphere coupled model simulations. *Climate Dynamics* **30**(7–8), 745–762.

Appendix A

Overview of Appendices

- Appendix B and C: This program in R allows users to load fossil and modern pollen databases along with modern climate data to reconstruct the climate in North America and analyze the reconstruction through time, spatially, and by ecoregion. The user has a choice of 4 pollen sums or a custom pollen sum, a choice of 69 climate variables or importing custom modern climate data. The user can subset temporally, spatially or by site resolution, create plots of anomalies (time slice & ecoregion), plot site density through time and plot SCD and perform spatial filtering on a reconstruction. Details are provided in the user guide (Appendix B). The full source code (Appendix C) will be made available in a digital archive on the internet where it will be accessible to researchers.

- Appendix D: This contains supplementary methods including code for the original reconstruction methodology that was developed using a series of MS Excel™Macros along with ESRI ArcGIS Modelbuilder™. In addition there are specific macros for calculating the fossil pollen site resolution, linear interpolation of reconstruction values through time and averaging site anomalies along with spatial filtering. There is also documentation for calibrating radiocarbon dates, using GHCN and WorldClim data, generating code for MATTOOLS, and processing NetCDF files using CDO command line software. Finally, there are some notes about modifications to figures in Chapter 2 (published paper) and a short literature review on data-model comparisons.

- Appendix E: This contains supplementary figures for Chapter 2 to 4. The first 3 figures 31-33 are climate reconstructions for all climate variables in the Whitmore modern database and do not fit into a specific chapter. Figures 34-60 are for Chapter 2, figures 61-74 are for Chapter 3 and figures 75-85 are for

Chapter 4. Additional figures 86-94 are based on supplementary results.

- Appendix F: This provides the list of references that were cited in only Appendices A through F. Some of these references may have been cited in the main thesis and therefore appear in the appropriate reference list(s) within the main thesis.

Appendix B

User Guide for R Program

User Guide for Pollen Editor, Reconstruction Builder and Pollen Paleoclimate Database Analysis Program in R

Prepared by: Matthew Ladd

Note that this R program will eventually be made available to the research community as part of a publication submitted to a journal. The code for this program has been tested using a minimum of R v2.15.0.

DISCLAIMER: This R program and documentation are provided "as is" without warranty of any kind, either express or implied, including, without limitation, any and all implied suitability for any given purpose or task. The entire risk arising out of the use or performance of this program remains with the end-user. In no event will the author of this software be liable for any damages whatsoever arising out of the use of or inability to use the program.

B.1 MODULE 1: Pollen database editor and reconstruction builder

STARTUP

1. Before starting the program, ensure that you have at least R v2.15.0. The program has not been tested with any version of R older than 2.15.0.

2. Unzip the file lpcRecon.zip to a new directory (e.g. C:/recon). Note that all output is written to the working directory so it would be wise to make a subfolder for output files.

3. Start R either from the menu or by double clicking a .RData file.

4. Set the working directory to the location where all files were unzipped in step two.

   ```
   setwd("C:\\working") OR setwd("C:/working")
   ```

   NOTE: the working directory can be checked by typing getwd()

5. Type: *source("globalsource.r")* and make note if there are any warnings.

6. To start the program simply type: *MainProgram1()* OR *MainProgram2()*

7. When starting module 1 (MainProgram1), the default fossil and modern pollen database is loaded which contains 135 taxa. The databases with 35 (36), 64 (63) and 134 taxa will be built at this time (if they do not already exist). The custom pollen sum database is initially built with all 135 pollen taxa, however, it can be user modified by selecting menu item 1.

   DO NOT rename the psCustom135.modern OR psCustom135.modern files otherwise the program will not run.

   ```
   Modern pollen sum 135 database is loaded!
   Fossil pollen sum 135 database is loaded!
   Custom modern pollen sum database file created with 135 (default) taxa!
   Custom fossil pollen sum database file created with 135 (default) taxa!
   Building modern pollen sum 35 database.....
   Building fossil pollen sum 35 database.....
   Building modern pollen sum 64 database.....
   Building fossil pollen sum 64 database.....
   Building modern pollen sum 134 database.....
   Building fossil pollen sum 134 database.....
   ```

   The next time you run module 1, the following messages will be displayed:

   ```
   Modern pollen sum 135 database is loaded!
   Fossil pollen sum 135 database is loaded!
   Modern pollen sum 35 database already created!
   Fossil pollen sum 35 database already created!
   Modern pollen sum 64 database already created!
   Fossil pollen sum 64 database already created!
   Modern pollen sum 134 database already created!
   Fossil pollen sum 134 database already created!
   ```

8. The following menu will appear (will be displayed after each menu item is run):

```
[1] Create a custom pollen database file (optional)
[2] Load a pollen db file (default pollen sums available of 35,64,134,135 (Williams and Shuman, 2008) or 0 for custom pollen sum as
      created in [1] - default pollen sum is 135)
[3] Show which fossil and modern pollen databases are currently loaded (optional)
[4] Show list of pollen taxa in currently loaded modern & fossil pollen database (optional)
[5] Join custom climate data to modern pollen sites (default Whitmore et al., 2005 v 1.7.2 - optional: CRU, GHCN, NCEP, ERA-40, NARR,
      BEST, etc. can be imported by user)
[6] Select a climate variable to reconstruct (required - from modern climate database)
[7] Display which climate variable from modern database is selected (optional - does not depend on which modern climate data is
      selected)
[8] Do reconstruction of selected climate variable (require MATTOOLS v1.1 from CRAN - only necessary if no .recon file is available to
      be loaded)
[9] Create a map of pollen sites from database (optional: fossil, modern or both)
[10] List all program files in working directory (optional)
[11] Set or change the current working directory (optional)
[12] Start climate reconstruction analysis program (optional)
[13] Quit program and return to R command prompt
```

9. To do a reconstruction with the default pollen sum database simply select a climate variable (menu item 6), then do the reconstruction (menu item 8). You need enough memory to run menu item 8 otherwise the function will generate an error. The program includes some .recon files for several climate variables (for pollen sum 64) which can be loaded in module 2.

10. After doing the reconstruction a .recon file is output which can be loaded in module 2 (menu item 1) for further analysis.

Overview of each menu item:

1. Create a custom pollen database file (optional)

   The program will ask:

```
Do you wish to include taxa ABIES in your custom pollen sum? Use <y> or <n> or <a> to aggregate taxa: n
Do you wish to include taxa ACERNEGUNDO in your custom pollen sum? Use <y> or <n> or <a> to aggregate taxa: a
```

   With the aggregate option, we are combining the pollen sums of two or more taxa. The list of available taxa to be aggregated is shown. Note that ACERNEGUNDO is removed from the list after selecting it as it cannot be aggregated with itself. If a taxa is aggregated, it cannot then be aggregated with other taxa.

```
3 AcerPensylvanicum
4 AcerRubrum
5 AcerSaccharinum
6 AcerSaccharum
7 AcerUndiff
8 AlnusCrispa
9 AlnusRubra
10 AlnusRugosa
11 AlnusUndiff
12 AmbrosiaType
13 Amorpha
14 Anacardiaceae
15 Apiaceae
16 Aquifoliaceae
17 Arceuthobium
18 Arecaceae
19 Armeria
20 Artemisia
```

21 Asteraceae
22 AsteraceaeSubfCichorioideae
23 BetulaUndiff
24 Boraginaceae
25 Botrychium
26 Brassicaceae
27 Cactaceae
28 Campanulaceae
29 Caprifoliaceae
30 Carya
31 Caryophyllaceae
32 Castanea
33 Ceanothus
34 Celtis
35 Cephalanthus
36 Cercocarpus
37 ChenopodiaceaeAmaranthaceae
38 Chrysolepis
39 Cornus
40 Corylus
41 Cupressaceae
42 Cyperaceae
43 Dodecatheon
44 Dryas
45 Elaeagnaceae
46 Ephedra
47 Equisetum
48 EricaceaeChamadaphneLedum
49 EricaceaeVaccinium
50 EricaceaeUndiff
51 Ericales
52 Eriogonum
53 Euphorbiaceae
54 Fabaceae
55 Fagus
56 FraxinusNigraType
57 FraxinusPennsylvanicaAmericanaType
58 FraxinusUndiff
59 Iva
60 JuglansCinerea
61 JuglansNigra
62 JuglansUndiff
63 KoenigiaIslandica
64 Lamiaceae
65 LarixPseudotsuga
66 Larrea
67 Liliaceae
68 Liquidambar
69 Liriodendron
70 LycopodiumAnnotinum
71 LycopodiumClavatum
72 LycopodiumComplanatum
73 LycopodiumSelago
74 LycopodiumUndiff
75 Magnoliaceae
76 Malvaceae
77 Moraceae
78 Myricaceae
79 Nyssa
80 Onagraceae
81 Osmundaceae
82 OstryaCarpinus
83 Oxyria
84 Papaveraceae
85 Pedicularis
86 PiceaGlauca
87 PiceaMariana
88 PiceaUndiff
89 PinusDiploxylon
90 PinusHaploxylon
91 PinusUndiff
92 PinusSE
93 Plantaginaceae
94 Platanus
95 Poaceae
96 Polemoniaceae

```
97 PolygonaceaeUndiff
98 PolygonumBistortoides
99 PolygonumUndiff
100 PolygonumViviparum
101 Polypodiaceae
102 Populus
103 Potentilla
104 Prosopis
105 Pteridium
106 Quercus
107 RanunculaceaeUndiff
108 RhamnaceaeVitaceae
109 RosaceaeUndiff
110 Rubiaceae
111 Rubus
112 Rumex
113 RumexOxyria
114 Salix
115 Sanguisorba
116 Sarcobatus
117 SaxifragaCernua
118 SaxifragaHieracifolia
119 SaxifragaOppositifolia
120 SaxifragaTricuspidata
121 SaxifragaceaeUndiff
122 Scrophulariaceae
123 Selaginella
124 ShepherdiaCanadensis
125 Sphagnum
126 Taxodium
127 Taxus
128 Thalictrum
129 Tilia
130 TsugaHeterophylla
131 TsugaMertensiana
132 TsugaUndiff
133 Ulmus
134 Urticaceae
135 Xanthium

Enter the name of the new aggregate column (will be converted to uppercase): ACER_AG
Enter taxa ID to aggregate with selected taxa 2 ACERNEGUNDO <use 0 to finish aggregation & show which taxa are aggregated> 3
You have aggregated 1 taxa
Enter taxa ID to aggregate with selected taxa 2 ACERNEGUNDO <use 0 to finish aggregation & show which taxa are aggregated> 4
You have aggregated 2 taxa
Enter taxa ID to aggregate with selected taxa 2 ACERNEGUNDO <use 0 to finish aggregation & show which taxa are aggregated> 5
You have aggregated 3 taxa
Enter taxa ID to aggregate with selected taxa 2 ACERNEGUNDO <use 0 to finish aggregation & show which taxa are aggregated> 6
You have aggregated 4 taxa
Enter taxa ID to aggregate with selected taxa 2 ACERNEGUNDO <use 0 to finish aggregation & show which taxa are aggregated> 7
You have aggregated 5 taxa
Enter taxa ID to aggregate with selected taxa 2 ACERNEGUNDO <use 0 to finish aggregation & show which taxa are aggregated> 0
AcerNegundo AcerPensylvanicum AcerRubrum AcerSaccharinum AcerSaccharum AcerUndiff
Do you wish to include taxa ALNUSCRISPA in your custom pollen sum? Use <y> or <n> or <a> to aggregate taxa:
```

And the user would continue until the end of the taxa list is reached. Then the user is asked:

```
Existing custom pollen sum files in working directory will be overwritten, do you wish to continue? <1 for yes, 2 for no>
```

If 2 is entered then the following is displayed:

```
Please move or rename the existing custom pollen sum database file before running menu item 1 again!

Writing files psCustomPerc.modern & psCustomPerc.fossil to working directory...
```

None of the .fossil or .modern files should be renamed if they will be used to build a reconstruction (.recon) file. If you create a new custom pollen sum database, these files will be overwritten if option 1 is chosen at the end. If you would like to create a new custom pollen sum database but keep these files

please move or rename the .modern and .fossil files. To open the files in MS Excel™simply rename them with a .csv extension.

NOTE: At least five pollen taxa must be selected to create a custom pollen sum database but obviously better results will be obtained with a higher number of taxa and using combinations of taxa which are appropriate for the reconstruction. More importantly, some knowledge of pollen taxa is needed in order to use this feature! In addition, having too few pollen taxa in the custom pollen sum database is likely to generate an error when attempting to build the reconstruction file with MATTOOLS in menu item 8.

2. Load a pollen db file (default pollen sums available of 35,64,134,135 (Williams and Shuman, 2008) or 0 for custom pollen sum as created in [1] - default pollen sum is 135)

   This allows the user to load another pollen database other than the default (135 taxa):

```
Select either 35, 64, 134, 135 or 0 for custom: 35
Modern pollen DB ( pollen sum [ 35 ]) is loaded
Fossil pollen DB ( pollen sum [ 35 ]) is loaded
```

   If no custom pollen sum database has been created the zero option will re-load the 135 pollen sum database:

```
Modern pollen DB ( custom pollen sum [ 135 ]) is loaded
Fossil pollen DB ( pollen pollen sum [ 135 ]) is loaded
```

   If a custom pollen sum database has been created, the number of taxa will be indicated in square brackets:

```
Modern pollen DB ( custom pollen sum [ 89 ]) is loaded
Fossil pollen DB ( pollen pollen sum [ 89 ]) is loaded
```

3. Show which fossil and modern pollen databases are currently loaded (optional)

   In this case the 35 pollen sum database is loaded:

```
Modern pollen DB ( pollen sum [ 35 ]) is loaded
Fossil pollen DB ( pollen sum [ 35 ]) is loaded
```

4. Show list of pollen taxa in currently loaded modern & fossil pollen database (optional)

```
---Custom Pollen Taxa List---
ABIES
ACER
ALNUS
ARTEMISIA
ASTERACEAE
BETULAUNDIFF
CACTACEAE
CARYA
```

```
CASTANEA
CHENOPODIACEAEAMARANTHACEAE
CORYLUS
CUPRESSACEAE
CYPERACEAE
EPHEDRA
FAGUS
FRAXINUS
JUGLANS
LARIXPSEUDOTSUGA
LIQUIDAMBAR
LIRIODENDRON
PICEA
PINUS
PINUSSE
PLATANUS
POACEAE
POPULUS
PROSOPIS
QUERCUS
SALIX
SARCOBATUS
SAXIFRAGA
THALICTRUM
TILIA
TSUGAHETEROPHYLLA
TSUGAUNDIFF
ULMUS
------------------------------
Total number of taxa: 36
------------------------------
```

5. Join custom climate data to modern pollen sites (default Whitmore *et al.*, 2005 v1.7.2 - optional : STATION, RE-ANALYSIS, RASTER or OTHER data can be imported by user)

   There are many different modern climate data sets that can be used in paleoclimate reconstructions. This feature allows the user to import another modern climate data set and assign each modern pollen site the nearest station or grid point value for the climate variable in the file. There are 4 options: (1) Station data [either Berkeley Earth Surface Temperature data set (BEST) or Global Historical Climate Network (GHCN)] using the packages BerkeleyEarth and RGHCNv3 respectively, (2) re-analysis products [either NCEP or North American Regional Re-analysis (NARR)] in NetCDF format, (3) raster data [e.g. WorldClim in .bil format] and (4) other data in .csv format.

   Re-analysis data (NCEP, NARR) can be imported directly as a NetCDF(.nc) file but should be a long term mean file (e.g. air.2m.ltm.mean.nc) format. A future option if using an NCEP monthly file is to select any ranges (either monthly or seasonal) from 1948-present (e.g. JJA 1971-2000 or July 1961-1990). It is recommended to use the program Climate Data Operators (CDO,

available from https://code.zmaw.de/projects/cdo) to average time steps of a netcdf file before either creating a .bil raster or new netcdf to use in this program. Some reanalysis products are only available in gridded binary (.grib) format. A separate program called Degrib would be needed (available from http://www.nws.noaa.gov/mdl/degrib/), to convert specific variables/time-slices in the grib file to csv format. However, the use of programs such as CDO and Degrib are beyond the scope of this user guide.

Raster data from WorldClim can be imported as long as there is a .bil and .hdr file present for each raster to be loaded. Other raster data sets that are not in .bil format have not been tested and therefore may or may not work. However, exporting a point shapefile as a raster in .bil format using ESRI ArcGIS™software has been tested and is recommended in certain cases to speed up processing. The modern pollen site at each raster cell is assigned the value of the raster cell that the modern pollen site intersects.

The csv imported in option 4 needs to be in csv format with a latitude, longitude, and climate value field. The program will not check to see if the values in these fields are correct. However, the function will exit if there are missing values in any of the three fields.

The user needs to be aware that if they are reconstructing a variable comparing a custom modern climate variable to the existing modern climate variable, there needs to be the equivalent variable in the custom modern climate data set joined to each modern pollen site. For example:

- If MTWA (mean temperature of the warmest month) is being reconstructed, there should be TJUL in the custom modern climate file.
- If MTCO (mean temperature of the coldest month) is being reconstructed, there should be TJAN in the custom modern climate file.

For some variables, there might not be an equivalent, for example GDD5, which is growing degree days (with a base of 5°C), is only present in the default modern data set. However, 500 hgt (atmospheric pressure at the 500 mb level) is

only present in the NARR or NCEP re-analysis file. And for some variables, particularly at the upper level of the atmosphere (e.g. 300 mb zonal winds), the reconstructions may not be reliable.

Example data sets (e.g. New *et al.*, 2002 derived from CRU and ERA-40 July mean temperatures) are included in the zip file for this R program. Long-term monthly mean files cover the period 1961-1990 for both New *et al.* and ERA-40.

More information about NCEP can be found at:
http://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis.html

More information about NARR can be found at:
http://www.esrl.noaa.gov/psd/data/gridded/data.narr.html

Here we select option 4 (csv data file):

- Examples of climate station data are BEST and GHCN
  [using the R package BerkeleyEarth or RGHCNv3]
  Processing of the data can take some time depending on memory available and the subset of data selected
  BEST files must be downloaded to disk before running option 1
- Examples of climate re-analysis data are NCEP and NARR
  [must be in NetCDF (.nc) format]
  Re-analysis data must be a long term or monthly mean file (e.g. varname.mon.ltm.nc)
  Regular monthly mean files (e.g. varname.mon.mean.nc) have not been tested yet!
- Example of climate raster data is WorldClim
  [other raster formats such as .asc have not been tested yet!]
  Directory with Worlclim data must have .bil and .hdr file for each climate variable (e.g. tmean7 for July temp)

- Examples of other custom climate data in .csv format are New *et al.*(2002) [based on CRU] or ERA-40 re-analysis
  [converted from grib format]
  The data must be converted by the user and have the following fields: latitude, longitude & value of climate variable (specified by user)

```
Do you want to use station, re-analysis, raster or other data for custom climate data? <1 for STATION, 2 for RE-ANALYSIS, 3 for RASTER,
      4 for OTHER>
```

Here the fields from the custom climate data file are listed:

```
List of column names in loaded climate data file:

lat
lon
jan
feb
mar
apr
may
jun
jul
aug
sep
oct
nov
dec
```

The user is informed if the climate variable selected is present or not in the modern pollen database. The modern pollen database has a field TJUL, however, jul is not recognized as TJUL.

```
Enter the name of the column with the climate data needed from this file (not case sensitive): jul
Climate variable jul does not exist in modern pollen database file!
```

Here is where each modern pollen site is joined to the nearest climate station or grid point.

```
Joining the nearest climate station or gridpoint (for selected climate variable) to each modern pollen site, this could take a long
      time depending on the data set used......
```

Optionally we can correct station temperature values based on elevation difference between modern pollen site and nearest station. The correction is (pollen site elevation - station elevation) / 100 * lapse rate where the user can enter an average lapse rate (for the continent) of 0.6°C to 1°C / 100m.

```
Do you want to correct the temperature values of station data based on elevation difference (using lapse rate) between modern pollen
      site and nearest station? <1 for YES, 2 for NO> 2
```

The variable name entered needs to be identical to the one in the modern pollen database (not case sensitive). You can also change the name of the field.

```
TJAN
TFEB
TMAR
TAPR
TMAY
```

```
TJUN
TJUL
TAUG
TSEP
TOCT
TNOV
TDEC
PJAN
PFEB
PMAR
PAPR
PMAY
PJUN
PJUL
PAUG
PSEP
POCT
PNOV
PDEC
SJAN
SFEB
SMAR
SAPR
SMAY
SJUN
SJUL
SAUG
SSEP
SOCT
SNOV
SDEC
TAVE
TMAX
TMIN
GDD0
GDD5
CHILL
MTCO
MTWA
MIPT
AAETPT
APETPT
MIPTEV
AAETPTEV
APETPTEV
MIPTDC
AAETPTDC
APETPTDC
TOTSNOPT
ANNP
PJANPANN
PJULPANN
TMAM
TJJA
TSON
TDJF
PMAM
PJJA
PSON
PDJF
SMAM
SJJA
SSON
SDJF

Enter the name of the column from the modern pollen database to overwrite:
tjul
Enter a new column name for tjul :
Tjul
```

If the process is successful, the modern pollen database with updated modern climate is written to a csv file. This file should be renamed so it is not overwritten.

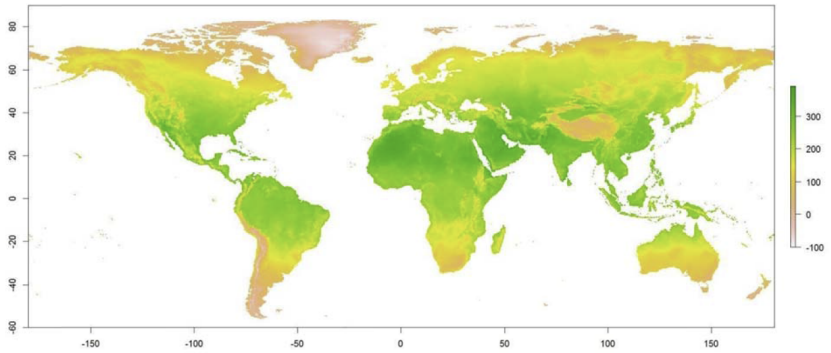


Figure 17: Long term mean July temperature map (1950 to 2000 CE) from WorldClim data (TMEAN7 raster layer).

```
Updating of modern climate data in modern pollen database file is complete!
Wrote new modern pollen file ps135_modern_updated.csv with updated climate information to disk!
```

If there is a problem with the custom modern climate data file (e.g. missing values), the user is returned to the main menu and the modern pollen database file is not changed.

```
Problem with spatial join, check custom modern climate file for missing values!
```

Option 3 is similar to above except that the user loads a WorldClim raster file:

```
Do you want to use station, re-analysis, raster or other data for custom climate data? <1 for STATION, 2 for RE-ANALYSIS, 3 for RASTER,
       4 for OTHER>
3
NOTE: Only WorldClim data has been tested with this function!
```

The following feature is useful to check the loaded raster to make sure it is correct. This example shows mean July temperature (°C values * 10):

```
Do you want to plot the loaded raster file? <use 1 for YES or 2 for NO> 1
```

The user is given the option to correct temperature values of the WorldClim temperature raster:

```
Do you want to correct the temperature values of the raster? (WorldClim values will be divided by 10) <1 for YES, 2 for NO>
```

The user is warned about missing (NA) values when assigning the raster cell that intersects a modern pollen site. Most likely the modern pollen site lat/lon is not precise enough for very high resolution rasters where each raster cell is 1 km^2:

```
NA values found, these values will be substituted with the appropriate values from the modern pollen file!
```

The user is given the option to substitute existing modern climate data at each modern pollen site. So for example, if a total precipitation data set is loaded (e.g. WorldClim bioclim12) and NA values are found, the user can substitute the ANNP value from the default modern climate data set instead.

```
Enter the name of the column with the climate data needed from this file (not case sensitive):
```

Then the columns from the modern climate data file are listed:

```
List of column names in loaded climate data file:

LONDD
LATDD
TMEAN7

Enter the name of the column with the climate data needed from this file (exactly as it appears above):
TMEAN7
Climate variable TMEAN7 does not exist in modern pollen database file!
```

Option 2: Here we load a NARR long term monthly mean netcdf file (surface air temperature). The program will identify whether there is a level field (with hgt file) and ask the user for pressure levels (e.g. 500 mb, 700 mb, 850 mb, etc.) if necessary.

```
Do you want to use station, re-analysis, raster or other data for custom climate data? <1 for STATION, 2 for RE-ANALYSIS, 3 for RASTER,
       4 for OTHER>
2

[1] "file C:\Documents and Settings\g1779631\Desktop\bigReconV2-9\extra_files\air.sfc.mon.ltm.nc has 4 dimensions:"
[1] "time Size: 12"
[1] "y Size: 277"
[1] "x Size: 349"
[1] "nbnds Size: 2"
[1] "------------------------"
[1] "file C:\\Documents and Settings\g1779631\Desktop\bigReconV2-9\extra_files\air.sfc.mon.ltm.nc has 5 variables:"
[1] "int Lambert_Conformal[] Longname:Lambert_Conformal Missval:NA"
[1] "short air[x,y,time] Longname:Long Term Monthly Mean air temperature at Surface Missval:32766"
[1] "float lat[x,y] Longname:latitude coordinate Missval:1e+30"
[1] "float lon[x,y] Longname:longitude coordinate Missval:1e+30"
[1] "double climatological_bnds[nbnds,time] Longname:Time Boundaries Missval:1e+30"
```

Note than an error will be generated if this is not entered correctly:

```
What is the short name of the variable of interest? (usually just above lat or below lon in output above, e.g. air for surface temp)
      air
```

Select the appropriate re-analysis data type (note this when you download the file). An error will be generated if the wrong re-analysis file type is selected:

```
What type of re-analysis file is this? <use 1 for NARR or 2 for NCEP> 1

For long term mean file, starting and ending month must be the same:
Enter the starting month of data to be selected: (use 1 for Jan, 2 for Feb, ... 7
Enter the ending month of data to be selected: (use 1 for Jan, 2 for Feb, ... 7
```

Check to make sure first data value is correct. In this example for mean July temperature, the first value looks reasonable. The selected data is written to csv file and the data can be used in the same manner as shown in option 4:

```
First data value for air is shown below:
26.8300083041191
Writing file NARR_1979-2000_MEAN_AIR_MONTH_7.csv to disk...
List of column names in loaded climate data file:

lat
lon
AIR

Enter the name of the column with the climate data needed from this file (exactly as it appears above):
```

Here is an example using NARR hgt long term mean file, this file has one more dimension (pressure level) than the example above:

```
Do you want to use station, re-analysis, raster or other data for custom climate data? <1 for STATION, 2 for RE-ANALYSIS, 3 for RASTER,
       4 for OTHER>
 2

[1] "file C:\\Documents and Settings\\g1779631\\Desktop\\bigReconV2-9\\extra\_files\\hgt.mon.ltm.nc has 5 dimensions:"
[1] "y Size: 277"
[1] "x Size: 349"
 [1] "level Size: 29"
[1] "time Size: 12"
[1] "nbnds Size: 2"
[1] "------------------------"
[1] "file C:\\Documents and Settings\\g1779631\\Desktop\\bigReconV2-9\\extra_files\\hgt.mon.ltm.nc has 4 variables:"
 [1] "float lat[x,y] Longname:latitude Missval:1e+30"
[1] "float lon[x,y] Longname:longitude Missval:1e+30"
[1] "double climatological_bnds[nbnds,time] Longname:climatological time bounds Missval:1e+30"
 [1] "short hgt[x,y,level,time] Longname:Monthly Long Term Mean Geopotential Heights on Pressure Levels Missval:32766"


What is the short name of the variable of interest? (usually just above lat or below lon in output above, e.g. air for surface temp)
      hgt
What type of re-analysis file is this? <use 1 for NARR or 2 for NCEP> 1
NARR has 29 pressure levels (hPa):
1000, 975, 950, 925, 900, 875, 850, 825, 800, 775, 750, 725, 700, 650, 600, 550, 500, 450, 400, 350, 300, 275, 250, 225, 200, 175, 150,
       125, 100

Enter the index of the pressure level needed (must be between 1 for 1000mb to 29 for 100mb) 17
Enter the starting month of data to be selected: (use 1 for Jan, 2 for Feb, ... 7
Enter the ending month of data to be selected: (use 1 for Jan, 2 for Feb, ... 7
 First data value for hgt is shown below:
584.8
Writing file NARR_1979-2000_MEAN_HGT17_MONTH_7.csv to disk...
List of column names in loaded climate data file:

lat
lon
HGT

Enter the name of the column with the climate data needed from this file (not case sensitive):
 hgt
Climate variable hgt does not exist in modern pollen database file!
Joining the nearest climate station or gridpoint (for selected climate variable) to each modern pollen site, this could take a long
      time depending on the data set used......
 TJAN
.
SDJF

Enter the name of the column from the modern pollen database (listed above) to overwrite:
 tjul
Enter a new column name for tjul :
tjul500hgt
Updating of modern climate data in modern pollen database file is complete!
Wrote new modern pollen file ps135_modern_updated.csv with updated climate information to disk!
```

Option 1: BEST and GHCN data can also be used. Global Historical Climate Network (GHCN) is for North America only and produced by the National Climatic Data Center (NCDC). BEST is a relatively new data set for the entire earth surface, the most extensive instrumental climate data set thus far, produced by the Berkeley Earth Surface Temperature group and available to all researchers. BEST files must be downloaded by the user rather than using the download function in this program.

```
Do you want to use station, re-analysis, raster or other data for custom climate data? <1 for STATION, 2 for RE-ANALYSIS, 3 for RASTER,
       4 for OTHER>
1
Which station data set to select? <use 1 for BEST, 2 for GHCN> 2
```

```
Downloading latest GHCN files (TAVG), this could take some time....

trying URL 'ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/v3/ghcnm.tavg.latest.qca.tar.gz'
ftp data connection made, file length 12117510 bytes
opened URL
downloaded 11.6 Mb

trying URL 'ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/v3/ghcnm.tavg.latest.qca.tar.gz'
ftp data connection made, file length 12117510 bytes
opened URL
downloaded 11.6 Mb

Stations read in 7280
Stations with more than 1 year of data 7280
Standard baseline for modern climate data is 1961-1990

Enter starting year, must be between 1900 and 1980: 1961
Enter ending year, must be between 1930 and 2010: 1990
Enter the month of data to be selected: (use 1 for Jan, 7 for Jul, etc...)

Writing file NamGhcnList.csv to disk...
File NamGhcnList.csv has:
376 data columns
2827 stations
Averaging the data for selected month for selected range of years....
List of column names in loaded climate data file:

Id
Lat
Lon
Elevation
Name
GridEl
Rural
Population
Topography
Vegetation
Coastal
DistanceToCoast
Airport
DistanceToTown
NDVI
Light_Code
TAVG_7

Enter the name of the column with the climate data needed from this file (exactly as it appears above):
```

With BEST, the files need to be present on the user's hard disk, so it is better to download the files manually from http://www.berkeleyearth.org. Otherwise the following error is generated:

```
trying URL 'http://download.berkeleyearth.org/downloads/TAVG/LATEST%20-%20Quality%20Controlled.zip'
Error in download.file(urls$Url[thisUrl], destfile = file.path(destDir, :
  cannot open URL 'http://download.berkeleyearth.org/downloads/TAVG/LATEST%20-%20Quality%20Controlled.zip'
In addition: Warning message:
In download.file(urls$Url[thisUrl], destfile = file.path(destDir, :
  unable to connect to 'download.berkeleyearth.org' on port 80.
```

Here is the full working example:

```
Which station data set to select? <use 1 for BEST, 2 for GHCN> 1

Berkeley earth surface temperature TAVG files will be loaded from disk....

Enter the month of data to be selected: (use 1 for Jan, 2 for Feb, etc... 7

Standard baseline for modern climate data is 1961-1990
Enter starting year, must be between 1900 and 1980: 1961
Enter ending year, must be between 1930 and 2010 : 1990
Writing file NAM_BEST_MONTH_7_TAVG_1961-1990.csv to disk...
with 6 data columns
and 13164 stations
```

```
List of column names in loaded climate data file:

Id
Lat
Lon
Name
Altitude
TAVG_7

Enter the name of the column with the climate data needed from this file (not case sensitive):
tavg_7
Climate variable tavg_7 does not exist in modern pollen database file!
Joining the nearest climate station or gridpoint (for selected climate variable) to each modern pollen site, this could take a long
      time depending on the data set used......
Do you want to correct the temperature values of station data based on elevation difference (using lapse rate) between modern pollen
      site and nearest station? <1 for YES, 2 for NO> 1
Enter the lapse rate in degrees Celsius per 100m (must be between 0.6 and 1): 0.8
TJAN
SDJF

Enter the name of the column from the modern pollen database (listed above) to overwrite:
tjul
Enter a new column name for tjul :
tjulBESTcorr
Updating of modern climate data in modern pollen database file is complete!
Wrote new modern pollen file ps135_modern_updated.csv with updated climate information to disk!
```

Note that menu item 2 needs to be run in order to restore the default modern climate data (i.e. to reload the modern climate data from the modern pollen database file). So if you want to restore the original modern climate data for MTWA from pollen sum 64, you will need to enter 64 in menu item 2. Note that the original .modern files are never modified, which allows you to compare the .csv with the imported modern climate data to the default modern climate data.

6. Select a climate variable to reconstruct (required - from modern climate database)

```
1 : Mean monthly temperature for January in C
2 : Mean monthly temperature for February in C
3 : Mean monthly temperature for March in C
4 : Mean monthly temperature for April in C
5 : Mean monthly temperature for May in C
6 : Mean monthly temperature for June in C
7 : Mean monthly temperature for July in C
8 : Mean monthly temperature for August in C
9 : Mean monthly temperature for September in C
10 : Mean monthly temperature for October in C
11 : Mean monthly temperature for November in C
12 : Mean monthly temperature for December in C
13 : Mean monthly precipitation for January in mm
14 : Mean monthly precipitation for February in mm
15 : Mean monthly precipitation for March in mm
16 : Mean monthly precipitation for April in mm
17 : Mean monthly precipitation for May in mm
18 : Mean monthly precipitation for June in mm
19 : Mean monthly precipitation for July in mm
20 : Mean monthly precipitation for August in mm
21 : Mean monthly precipitation for September in mm
22 : Mean monthly precipitation for October in mm
23 : Mean monthly precipitation for November in mm
24 : Mean monthly precipitation for December in mm
25 : Mean monthly percent possible sunshine for January
26 : Mean monthly percent possible sunshine for February
27 : Mean monthly percent possible sunshine for March
28 : Mean monthly percent possible sunshine for April
29 : Mean monthly percent possible sunshine for May
30 : Mean monthly percent possible sunshine for June
```

```
31 : Mean monthly percent possible sunshine for July
32 : Mean monthly percent possible sunshine for August
33 : Mean monthly percent possible sunshine for September
34 : Mean monthly percent possible sunshine for October
35 : Mean monthly percent possible sunshine for November
36 : Mean monthly percent possible sunshine for December
37 : Average annual temperature in C
38 : Absolute maximum temperature in C
39 : Absolute minimum temperature in C
40 : Growing-degree days (0 C base)
41 : Growing-degree days (5 C base)
42 : Chilling requirement (number of days when pseudo-daily temperature was below 5 C)
43 : Mean temperature (in C) of the coldest month of the year
44 : Mean temperature (in C) of the warmest month of the year
45 : Total annual precipitation (in mm)
46 : January/Annual precipitation ratio
47 : July/Annual precipitation ratio
48 : Priestley-Taylor (alpha) parameter (AE/PE)
49 : Actual evapotranspiration (AE)
50 : Potential evapotranspiration (PE)
51 : Alpha calculated for the evergreen pft assimilation period (days when the pseudo-daily temperature > -4C
52 : AE calculated for the deciduous pft assimilation period
53 : PE calculated for the deciduous pft assimilation period
54 : Annual Priestly-Taylor moisture index (alpha) for deciduous assimilation period (when pseudo-daily temperatures are greater than 5
      C) [calculated using Cramer & Prentice approach]
55 : Annual actual evapotraspiration for deciduous assimilation period (when pseudo-daily temperatures are greater than 5C) [calculated
       using Cramer & Prentice approach]
56 : Annual potential evapotraspiration for deciduous assimilation period (when pseudo-daily temperatures are greater than 5C) [
      calculated using Cramer & Prentice approach]
57 : Total snow-water equivalent from bucket model (computed)
58 : Mean temperature (in C) for March, April, May
59 : Mean temperature (in C) for June, July, August
60 : Mean temperature (in C) for September, October, November
61 : Mean temperature (in C) for December, January, February
62 : Mean precipitation (in mm) for March, April, May
63 : Mean precipitation (in mm) for June, July, August
64 : Mean precipitation (in mm) for September, October, November
65 : Mean precipitation (in mm) for December, January, February
66 : Mean sunshine (%) for March, April, May
67 : Mean sunshine (%) for June, July, August
68 : Mean sunshine (%) for September, October, November
69 : Mean sunshine (%) for December, January, February
70 : [NEW CLIMATE VARIABLE]
Enter the number of the climate variables to be reconstructed from above: 37
```

If 70 is entered above:

```
You need to run the custom modern climate feature (menu item 5) before doing the reconstruction!
NEW CLIMATE VARIABLE selected is CUSTOM
```

If pollen database is changed, but menu item 6 is not run after, the following message is displayed:

```
NOTE: pollen sum database was changed, you need to re-select the climate variable (menu item 6) before doing reconstruction!
```

7. Display which climate variable from modern database is selected (from modern climate database or new variable from custom climate data - optional)

```
The default climate variable selected is TJAN. If CUSTOM is selected in menu item 6 but menu 5 is not run, the default TJAN will be
      reconstructed.
```

Menu item 7:

```
Climate variable selected is [NEW CLIMATE VARIABLE] ( CUSTOM )
```

Menu item 8:

```
TJAN will be reconstructed ( column index is 161 )

How many analogs from 2 to 20 ?
```

Otherwise, the selected variable will be reconstructed:

```
Climate variable selected is Average annual temperature in C
( TAVE )
```

If 70 is entered, the user will need to import a custom modern climate data set using menu item 5:

```
Climate variable selected is [NEW CLIMATE VARIABLE] ( CUSTOM )
```

If no climate variable is selected the following message is displayed:

```
No climate variable selected! (Run menu item 6)
```

8. Do reconstruction of selected climate variable (require MATTOOLS v1.1 from CRAN; output .recon file can be loaded in module 2)

This is where the reconstruction file is built. If you receive an error about the reconstruction, your custom pollen sum may be invalid. After selecting a variable (TAVE) and a pollen sum database (PS64) the user is asked the following:

```
TAVE will be reconstructed (column index is 198)
Loaded pollen sum database for fossil and modern is: 64
Keep in mind that a higher number of analogs takes longer to process
How many analogs from 2 to 20 ?
Building reconstruction file using MATTOOLS v1.1, this could take some time....
Finished running mat.dissim(), now building reconstruction object....
Now attempting to write reconstruction object to disk as a .recon file....
MATTOOLS code generated and run with output file PS64_RECONS_TAVE.recon written to working directory!
```

Here is an example of custom modern climate data being reconstructed:

```
Custom modern climate data loaded. To re-load the original modern climate data simply run menu item 2

jul500hgt will be reconstructed ( column index is 68 )
Loaded pollen sum database for fossil and modern is: 35

Keep in mind that a higher number of analogs takes longer to process

How many analogs from 2 to 20 ?
2
Building reconstruction file using MATTOOLS v1.1, this could take some time......
Finished running mat.dissim(), now building reconstruction object....
 Now attempting to write reconstruction object to disk as a .recon file....
Reconstruction file PS35_JUL500HGT.recon written to working directory!
```

NOTE: If a memory error is generated please ensure that you have enough RAM to build the reconstruction file (at least 2 GB of RAM is recommended). A memory error will look like this:

```
Building reconstruction file using MATTOOLS v1.1, this could take some time....
Error: cannot allocate vector of size 241 Kb
In addition: Warning messages:
1: In data.frame(..., check.names = FALSE) :
  Reached total allocation of 1022Mb: see help(memory.size)
2: In data.frame(..., check.names = FALSE) :
  Reached total allocation of 1022Mb: see help(memory.size)
```

If you get the following error, check your custom pollen sum database, you may not have enough taxa:

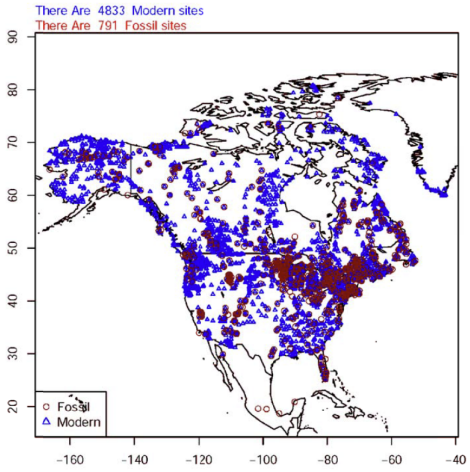


Figure 18: Map of fossil and modern pollen sites in North America

```
Error doing reconstruction with MATTOOLS! Possibly your custom pollen sum database loaded may be invalid.
```

If no climate variable is selected the following message is displayed:

```
No climate variable selected! (Run menu item 6)
```

9. Create a map of pollen sites from database (optional: fossil, modern or both)

   For this map you can either have only fossil pollen sites displayed (option 1), only modern pollen sites displayed (option 2), or both fossil and modern sites displayed (option 3).

```
Do you wish to create a map of modern or fossil sites? (enter <1> for fossil, <2> for modern, <3> for both) 3
File pollenSiteMap.pdf will be written to working directory!
```

10. List all program files in working directory (optional)

```
Listing all files in working directory with extension .modern, .fossil, and .recon
[1] "ps134perc.modern" "ps135perc.modern" "ps35perc.modern" "ps64perc.modern" "psCustomPerc.modern"
[1] "ps134perc.fossil" "ps135perc.fossil" "ps35perc.fossil" "ps64perc.fossil" "psCustomPerc.fossil"
[1] "PS64_RECONS_ANNP.recon" "PS64_RECONS_GDD5.recon" "PS64_RECONS_MIPT.recon" "PS64_RECONS_MTCO.recon"
[5] "PS64_RECONS_MTWA.recon" "PS64_RECONS_TAVE.recon"
```

   If a certain file type is not found in the working directory this will be indicated:

```
No .modern files found in working directory!
No .fossil files found in working directory!
No .recon files found in working directory!
```

11. Set or change the current working directory (optional)

    If the working directory specified exists, it will be changed, otherwise it will not be changed. In this case C:/pollen exists on the hard drive.

```
Enter the path to working directory (e.g. C:/pollen)
Working directory set to C:/pollen
```

12. Start climate reconstruction analysis program (optional)

    NOTE: This module will be discussed in part 2 of this program guide.

```
Starting reconstruction analysis program....

[1] Select a reconstruction file (.recon) to load for analysis (required)
[2] Display which reconstruction file is loaded for analysis & subset details if applicable (optional)
[3] Select a spatial subset from reconstruction (default continental-scale; optional based on lat/lon corners of rectangular box)
[4] Select/change calendar year BP range from reconstruction (required: valid range is -1000 to 12000)
[5] Select a subset from reconstruction based on calculated pollen site resolution (default all data, optional ranges from 10 to 1000
      years)
[6] Select analog or average of analogs from the reconstruction (default is first analogue, optional valid range is 1-20, or avg of top
       n analogues)
[7] Do linear interpolation of reconstruction (default is 100 year interval, optional valid range is 10-1000 year intervals)
[8] Calculate average anomalies by cal year BP for climate variable of interest (csv output)
[9] Plot average anomalies by cal year BP for climate variable of interest (optional)
[10] Plot average anomalies by cal year BP by ecoregion (default is Fedorova et al., 1994, no options currently available)
[11] Get average anomalies for all sites by time slice (optional: output csv file)
[12] Create Squared-Chored Distance (SCD) plots by ecoregion (optional)
[13] Plot SCD values from reconstruction file through time for all regions (optional)
[14] Calculate and plot site density through time (optional)
[15] Spatial filtering of reconstruction based on calculated local Moran's I values (optional)
[16] Perform a data-data or data-model comparison (beta-version, functionality in development)
[17] Set or change the current working directory (optional)
[18] List all program files (.pdf, .csv, .recon) in working directory (optional)
[19] Start pollen database editor program (optional)
[20] Quit program and return to R command prompt

\item Quit program and return to R command prompt

Exiting program, returning to the R command prompt....
```

END OF DOCUMENTATION FOR MODULE 1

B.2 MODULE 2: Pollen paleoclimate reconstruction analysis program

NOTE: Module 2 can be run without running module 1 first if .recon files have already been created. Also, if any output files are open, a warning will be generated if a function is trying to write to one of the open files. For example:

```
filename.csv cannot be written to disk, please close the file!
```

Note that the program allows .recon files with no ecoregion column however, certain menu items will not be available (such as SCD by eco-region).

1. Select a reconstruction file (.recon) to load for analysis (required)

   If the select file dialog box is canceled or an invalid reconstruction file is selected:

```
Invalid or no reconstruction file selected! Required file format is .recon.
```

   If this is a new .recon file being loaded in this session, the following message will be displayed:

```
Resetting all variables so that a new reconstruction can be loaded....
```

The following information is displayed based on the selected .recon file:

```
The following reconstruction file is loaded: PS64_RECONS_TAVE.recon
Pollen sum for selected reconstruction file is 64
Climate variable in reconstruction file is TAVE
Modern climate variable is in column 125
Total number of analogs in selected reconstruction file: 20
```

If a .recon file created with a custom climate variable is loaded, the following six lines of information should be displayed (notice the extra row):

```
The following reconstruction file is loaded: PS64_PDSI.recon
Pollen sum for selected reconstruction file is 64
Climate variable in reconstruction file is PDSI
Loading custom modern climate data file: ps64_modern_updated.csv
Modern climate variable is in column 130
Total number of analogs in selected reconstruction file: 5
```

If only the following 4 lines are displayed, then the custom modern climate file did not load properly:

```
The following reconstruction file is loaded: PS64_PDSI.recon
Pollen sum for selected reconstruction file is 64
Climate variable in reconstruction file is PDSI
Loading custom modern climate data file: ps64_modern_updated.csv
```

To correct the problem, we need to ensure that the correct ps64_modern_updated.csv is copied to the working directory. This file should have a column with the data for the custom climate variable each modern pollen site, in this case PDSI.

The number of analogs in the file is important for menu item #6 where an analog or average of several analogs is selected.

2. Display which reconstruction file is loaded for analysis & subset details if applicable (optional)

```
The following reconstruction file is loaded: PS64_RECONS_TAVE.recon
There are 30861 records in the reconstruction file!
There are 748 sites in the reconstruction file!
```

If no reconstruction file was selected the following is displayed:

```
No reconstruction file loaded! (Run menu item 1)
```

If a subset of the reconstruction is loaded then the following message will be displayed before the number of records and sites:

```
You have a subset of the reconstruction file!
```

Creating a subset by any combination of menu items 3 and 4 is permitted, as long as each of the two menu items is only run once. For example, you can get a spatial subset then subset that subset by pollen site resolution. You cannot subset spatially then subset by pollen site resolution twice.

3. Select a spatial subset from reconstruction (default continental-scale; optional based on lat/lon corners of rectangular box)

   By selecting a smaller spatial subset the processing time will be reduced however the smaller number of sites/records may affect the quality of further analyses. This subset can be further refined by running menu item 4 after creating a spatial subset.

```
The default study area for this program is North America. Changing the study area will affect the analysis of the reconstruction
and furthermore the ecoregion output may not be valid or available depending on the subset of North America selected.
For the reconstruction, would you like to use a subset of North America? <1 for yes, 2 for no> 1
```

   If 1 is entered, the user will be asked for the coordinates of the study area box. The box needs to span at least 5° of latitude and at least 5° of longitude to be valid. The subset of records will be created with the number of sites and records indicated. The site density calculation provides a measure of the density of the sites in the specified study area:

```
Enter maximum (west) longitude: -90
Enter minimum (east) longitude: -70
Enter minimum (south) latitude: 40
Enter maximum (north) latitude: 50
Subset of reconstruction by lat/lon created!
Subset has 10327 records
Subset has 229 sites
Area of subset is 2328175 km2
Site density is 9.83603006660495e-05 sites per km2
```

   If 2 is entered, the user will return to the main menu and the default study area will be used (all of North America).

   If the user attempts to create a spatial subset of an existing spatial subset, the following is displayed:

```
You already have a spatial subset of the reconstruction. Do you wish to keep this subset <1> or reload the whole reconstruction <2>?
```

   After either choice the user is returned to the main menu. The user can select either menu item 4 or select menu item 5 if no further sub-setting needs to be performed. Menu items 3, 4 and 7 create row subsets of the reconstruction file. Menu item 5 creates a column subset.

   The file subsetLatLon.csv has the coordinates (in X-Y values) for the study area subset.

   If no reconstruction file was selected the following message will be displayed:

```
No reconstruction file loaded! (Run menu item 1)
```

4. Select a subset from reconstruction based on calculated pollen site resolution (default all data, optional ranges from 10 to 1000 years)

   This allows the user to select pollen sites based on a certain resolution, calculated by the formula: [(AgeMax AgeMin) / number of samples]. Selecting higher or lower resolutions may reduce the quality of subsequent analyses. If there is a spatial subset already, this subset by pollen site resolution will be a subset of that subset. Processing time for this subset option can be significantly longer depending on the processing power of the user's computer.

   The fossil pollen site report will contain the following fields: siteID, ageMax, ageMin, ageDiff, siteRes, totSamp. The final output file is named fossilPollenSiteResolution.csv and will be identical in format to the original .recon file but will contain only the sites that meet the specified pollen site resolution criteria. There will also be an extra column (siteResCol) indicating the site resolution in years. For example, if 100 is specified only sites that have a resolution of less than or equal to 100 years will be included.

```
What is the lowest pollen site resolution (largest value in years) you wish to include from the reconstruction (valid range is 10-1000
      years): 50
Creating the subset by pollen site resolution, this could take a long time depending on the subset used....
Loading required package: plyr
Writing fossil pollen site resolution report to csv file....
Would you like to plot samples through time? <1 for YES, 2 for NO> 2
Subset of reconstruction by fossil pollen site resolution created!
Subset has 2042 records
Subset has 87 sites
```

   In this case our pollen site resolution specified was very fine, hence only 2,042 out of 30,861 records (all records) were retained.

   The samples through time plot can be used to see if there are any gaps in the data.

   If the user attempts to create a pollen site resolution subset of an existing pollen site resolution subset, the following is displayed:

```
You already have a subset of the reconstruction based on pollen resolution. Do you wish to keep this subset <1> or reload the whole
      reconstruction <2> ? 2
```

If a reconstruction file is missing, this will be indicated:

```
No reconstruction file loaded! (Run menu item 1)
```

5. Select analog or average of analogs from the reconstruction (default is first analogue, optional valid range is 1-20, or average of top N analogues)

   Here there are two options, take a single analog, so enter in 1 for the top analog and 1 for the bottom analog or [2 and 2] or [5 and 5]. Or enter 1 for the top analog and 3 for the bottom analog, which will average the top three analogs. To average the top 5 analogs enter 1 for the top analog and 5 for the bottom analog.

   ```
   If only one analog is used enter the same analog at the first and second prompts.
   To average the top 3 analogs enter 1 then 3. You cannot average analogs that are
   not adjacent to each other in the reconstruction file [e.g. (1 and 3) or (1,2 and 4) or (3 and 5)]
   What is the top analog you wish to use? (valid range is 1 to 20) 1
   What is the bottom analog you wish to use? For one analog, use the same value as top (valid range is 1 to 20) 1
   ```

   The subset of data by analog or average of analogs will be written to the file: reconSelAnalogData.csv (in the format below). As the top analog was selected, the last column is simply TAVE1. If the top three analogs were selected, then the last column has the average of the columns TAVE1, TAVE2 and TAVE3.

   Table 7: TAVE reconstruction for the top analog

SITEID	LATITUDENORTH	LONGITUDEEAST	CALYRBP	TAVE1
261	71.8	-124.3	-50	-9.93
1726	71.8	-122.7	-50	-14.22
1987	38.9	-76.4	-47	17.44
2522	61	-138	-45	-5.95
2541	46.1	-123.9	-45	9.53
2805	44.3	-74.4	-44	4.48
965	25.8	-80.5	-43	23.86

   If a reconstruction file is missing, this will be indicated:

   ```
   No reconstruction file loaded! (Run menu item 1)
   ```

6. Do linear interpolation of reconstruction (default is 100 year interval, optional valid range is 10-1000 year intervals)

   Higher values will take less time to process but may smooth the reconstruction excessively. Lower values will take more time to process and may not provide any more information than a higher value. The interval entered is rounded to the nearest 10 year interval. Linear interpolation is performed on all dates no older than 15,000 BP. The calendar year BP range must be selected in option 7 after the linear interpolation is run.

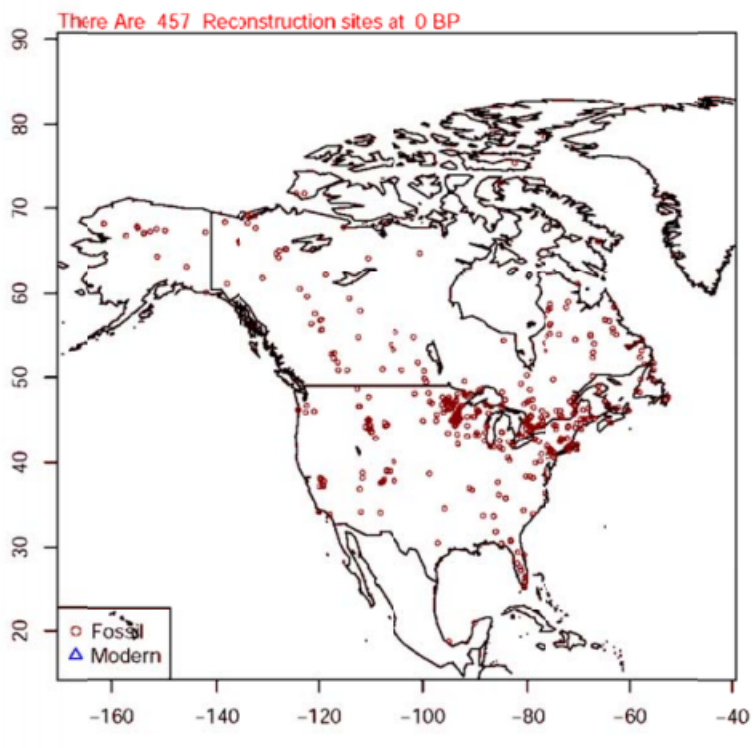


Figure 19: Reconstruction site distribution at 0 BP or 1950 CE

```
Enter the interval for linear interpolation? (valid range is 10-1000 years) 100
Performing the linear interpolation for all dates no older than 15000 BP, this could take some time......
Do you want to plot reconstruction site distribution by interpolated calendar year BP? <1 for YES, 2 for NO> 1
```

Optionally, the reconstruction site distribution at a given calendar year BP interval can be mapped to determine the quality of the site distribution:

```
Enter the calendar year BP interval to map (e.g. 1500): 0
```

7. Select/change calendar year bp range from interpolated reconstruction (required: valid range is 0 to 15000 BP)

This option creates a subset of the linearly interpolated reconstruction. For example, for a 2,000 year reconstruction, enter in 0 for youngest and 2000 for oldest. Note that each date entered (oldest and youngest) will be rounded to the nearest 10 year interval.

```
What is oldest calYrBP range needed from the interpolation for the analysis? (max age is 15000, min age is 0) 2000
Oldest calendar year BP is set to 2000 and note that oldest calendar year cannot be the same value as youngest calendar year BP!
What is youngest calYrBP range needed from the interpolation for the analysis? (min age is 0, max age is 15000) 0
Youngest calendar year BP is set to 0 and note that youngest calendar year cannot be the same value as oldest calendar year BP!
Subset of reconstruction by calendar year BP created!
Subset has 12169 records
Subset has 670 sites
```

The linearly interpolated data for TAVE (0-2ka) will be written to the file: reconInterpData.csv (format is below, interpolation is done for each site)

Table 8: TAVE reconstruction after linear interpolation through time

SiteID	Lat	Lon	CalMean	InterpVal
7	47	-80.1167	0	2.571111
7	47	-80.1167	100	2.46199
7	47	-80.1167	200	2.681885
7	47	-80.1167	300	2.998429
7	47	-80.1167	400	3.727
7	47	-80.1167	500	4.297584
7	47	-80.1167	600	3.270738
7	47	-80.1167	700	2.86
7	47	-80.1167	800	2.86
7	47	-80.1167	900	3.267753
7	47	-80.1167	1000	3.725903
7	47	-80.1167	1100	3.623262
7	47	-80.1167	1200	3.17691
7	47	-80.1167	1300	2.86
7	47	-80.1167	1400	2.86
7	47	-80.1167	1500	3.0197
7	47	-80.1167	1600	3.409213
7	47	-80.1167	1700	3.798727
7	47	-80.1167	1800	3.59216
7	47	-80.1167	1900	3.17616
7	47	-80.1167	2000	2.86
7	47	-80.1167	2100	2.86
7	47	-80.1167	2200	2.86
7	47	-80.1167	2300	2.86
7	47	-80.1167	2400	2.86
7	47	-80.1167	2500	2.86

If the user attempts to create a temporal subset of an existing temporal subset, the following is displayed:

```
You already have a subset of the reconstruction based on cal yr BP range. Do you wish to keep this subset <1> or reload the whole
      reconstruction <2> ?
After either choice the user is returned to the main menu.
```

8. Calculate average anomalies by cal year BP for climate variable of interest (csv output)

   In order to calculate the anomalies, we need to join the reconstruction to the nearest modern site. Use option 1 for more accurate distance calculation at high latitudes. For GC, enter distance in km, for Euclidean, enter distance in degrees.

SiteID	Lon	Lat	CalMean	InterpVal	CloseSiteIndex	Distance	ModClimVal	Ecoregion	reconSPjoinAnCalc
7	-89.1	47	0	2.571111111	455	0.017	2.86	Conifer/Hardwood	-0.288888889
7	-89.1	47	100	2.461989529	455	0.017	2.86	Conifer/Hardwood	-0.398010471
7	-89.1	47	200	2.681884817	455	0.017	2.86	Conifer/Hardwood	-0.178115183
7	-89.1	47	300	2.998428571	455	0.017	2.86	Conifer/Hardwood	0.138428571
7	-89.1	47	400	3.727	455	0.017	2.86	Conifer/Hardwood	0.867
7	-89.1	47	500	4.297583893	455	0.017	2.86	Conifer/Hardwood	1.437583893
7	-89.1	47	600	3.270738255	455	0.017	2.86	Conifer/Hardwood	0.410738255
7	-89.1	47	700	2.86	455	0.017	2.86	Conifer/Hardwood	0
7	-89.1	47	800	2.86	455	0.017	2.86	Conifer/Hardwood	0
7	-89.1	47	900	3.267753304	455	0.017	2.86	Conifer/Hardwood	0.407753304
7	-89.1	47	1000	3.725903084	455	0.017	2.86	Conifer/Hardwood	0.865903084
7	-89.1	47	1100	3.623261803	455	0.017	2.86	Conifer/Hardwood	0.763261803
7	-89.1	47	1200	3.176909871	455	0.017	2.86	Conifer/Hardwood	0.316909871
7	-89.1	47	1300	2.86	455	0.017	2.86	Conifer/Hardwood	0
7	-89.1	47	1400	2.86	455	0.017	2.86	Conifer/Hardwood	0
7	-89.1	47	1500	3.019700375	455	0.017	2.86	Conifer/Hardwood	0.159700375
7	-89.1	47	1600	3.409213483	455	0.017	2.86	Conifer/Hardwood	0.549213483
7	-89.1	47	1700	3.798726592	455	0.017	2.86	Conifer/Hardwood	0.938726592
7	-89.1	47	1800	3.59216	455	0.017	2.86	Conifer/Hardwood	0.73216
7	-89.1	47	1900	3.17616	455	0.017	2.86	Conifer/Hardwood	0.31616
18	-97.5	29.6	200	20.09	3864	0.017	20.09	Deciduous Forest	0
18	-97.5	29.6	300	20.09	3864	0.017	20.09	Deciduous Forest	0
18	-97.5	29.6	400	20.09	3864	0.017	20.09	Deciduous Forest	0
18	-97.5	29.6	500	20.09	3864	0.017	20.09	Deciduous Forest	0
18	-97.5	29.6	600	19.926875	3864	0.017	20.09	Deciduous Forest	-0.163125
18	-97.5	29.6	700	17.47347826	3864	0.017	20.09	Deciduous Forest	-2.616521739
18	-97.5	29.6	800	9.95173913	3864	0.017	20.09	Deciduous Forest	-10.13826087
18	-97.5	29.6	900	11.90074627	3864	0.017	20.09	Deciduous Forest	-8.189253731
18	-97.5	29.6	1000	18.10472637	3864	0.017	20.09	Deciduous Forest	-1.985273632

Figure 20: The output file SpatialJoinReconModWithAnCalc.csv viewed in an MS Excel™worksheet.

```
Which method of spatial join do you wish to use? <1 for Great Circle distance, 2 for Euclidean> 2
Performing the analysis, this could take some time......
```

If 1 is entered:

```
Enter search radius for joining the reconstruction to modern sites? (valid range is 1-1000 km, 150 km is recommended)
```

All sites can be used or only sites within a certain search radius specified. The search radius entered will affect the analysis of the reconstruction.

```
Do you wish to use all sites in the average anomaly by cal yr BP calculation? <1 for YES, 2 for NO> 2
Enter search radius for joining the reconstruction to modern sites? (valid range is 0.1-10 deg, 2 deg is recommended) 2
```

Optionally you can spatially filter the reconstruction using spatial statistics. This is covered later in this user manual (same functionality as menu item #15). The feature allows for a comparison between a spatially filtered and unfiltered reconstruction.

```
Would you like to spatially filter this reconstruction output? <1 for YES, 2 for NO> 2
```

The output file is SpatialJoinReconModWithAnCalc.csv (for TAVE). The value for reconSPjoinAnCalc is InterpVal (recon) ModClim Val (modern). The closest siteID to lat/lon is shown in CloseSiteIndex with the distance measured (Distance) in either degrees (Euclidean) or km (Great Circle distance). The ecoregion for each site is also given.

If a reconstruction file is missing, this will be indicated:

```
No interpolated reconstruction file loaded! (Run menu item 7)
```

9. Plot average anomalies by cal year BP for climate variable of interest (optional)

Menu item 8 needs to be run beforehand. If not, the function will not be run however, no warning message is displayed.

For the version of the program in this thesis, there is the option to tune dates either using a default tuning for a 2,000 year reconstruction or to manually enter tuned dates.

```
Would you like to tune the cal year BP dates for the analysis? <1 for yes, 2 for no> 1
Reconstruction of the last 2k at 100 yr intervals detected. Use default <1> or manual <2> tuning? 1
Please enter the units for the y-axis (e.g. for ANNP enter 'mm', for MTWA enter 'C') C
Writing plot to file avgAnomalyPlotTuned.pdf in working directory...
Writing file avgAnomalyByTunedYearAD.csv to working directory...
```

The output file avgAnomalyByTunedYearAD.csv is shown below:

Table 9: Average anomaly and standard error for TAVE reconstruction with tuned dates

CalMean	AvgAnom	StdError
1950	-0.01148	0.153788
1900	-0.19542	0.136916
1850	-0.30952	0.13561
1800	-0.27884	0.135376
1750	-0.25478	0.137321
1700	-0.27545	0.13768
1650	-0.28623	0.138526
1600	-0.29997	0.138246
1550	-0.29674	0.13704
1450	-0.2579	0.137383
1350	-0.25867	0.13768
1250	-0.28888	0.138318
1150	-0.19556	0.139192
1050	-0.18718	0.138849
950	-0.14093	0.139024
850	-0.10838	0.139262
750	-0.09751	0.13932
650	-0.14872	0.139351
550	-0.15018	0.138577
450	-0.15906	0.138745
350	-0.1477	0.138795

```
Would you like to tune the cal year BP dates for the analysis? <1 for yes, 2 for no> 2

Please enter the units for the y-axis (e.g. for ANNP enter 'mm', for MTWA enter 'C') C
Writing plot to file avgAnomalyPlot.pdf in working directory...
Writing file avgAnomalyByCalYrBP.csv to working directory...
```

The output file avgAnomalyByCalYrBP.csv is shown below:

Table 10: Average anomaly and standard error for TAVE reconstruction with BP dates

CalMean	AvgAnom	StdError
0	-0.01148	0.153788
100	-0.19542	0.136916
200	-0.30952	0.13561
300	-0.27884	0.135376
400	-0.25478	0.137321
500	-0.27545	0.13768
600	-0.28623	0.138526
700	-0.29997	0.138246
800	-0.29674	0.13704
900	-0.2579	0.137383
1000	-0.25867	0.13768
1100	-0.28888	0.138318
1200	-0.19556	0.139192
1300	-0.18718	0.138849
1400	-0.14093	0.139024
1500	-0.10838	0.139262
1600	-0.09751	0.13932
1700	-0.14872	0.139351
1800	-0.15018	0.138577
1900	-0.15906	0.138745
2000	-0.1477	0.138795

The grey lines in the plot are bounds of uncertainty (standard error) which are calculated according to the formula in Viau *et al.* (2006):

SQRT[STDEV(Fos) squared + STDEV(Mod) squared - 2 * p * STDEV(Fos) * STDEV(Mod)] / SQRT[n]

Where:

p (corr coefficient) is 0.9

n (sample size) is number of samples at each interval (nSamp)

10. Plot average anomalies by cal year BP by ecoregion (default is Fedorova *et al.*, 1994, no options currently available)

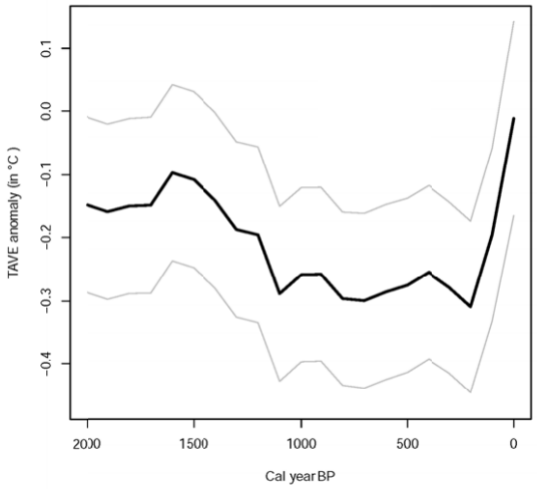


Figure 21: The plot of reconstructed TAVE anomalies for the last 2,000 years.

These ecoregions are adapted from Fedorova *et al.*, 1994. If an error is generated this means that one of more ecoregions have zero records at one or more time intervals. Most likely the subset of data selected is too small. There is an option to exclude certain ecoregions before plotting. In the example below, the Arctic ecoregion should be excluded due a lack of sites.

```
There are 18 sites in ecoregion Forest-Tundra at 0 bp
There are 21 sites in ecoregion Forest-Tundra at 100 bp
There are 24 sites in ecoregion Forest-Tundra at 200 bp
There are 24 sites in ecoregion Forest-Tundra at 300 bp
There are 27 sites in ecoregion Forest-Tundra at 400 bp
There are 27 sites in ecoregion Forest-Tundra at 500 bp
There are 27 sites in ecoregion Forest-Tundra at 600 bp
There are 28 sites in ecoregion Forest-Tundra at 700 bp
There are 28 sites in ecoregion Forest-Tundra at 800 bp
There are 28 sites in ecoregion Forest-Tundra at 900 bp
There are 29 sites in ecoregion Forest-Tundra at 1000 bp
There are 28 sites in ecoregion Forest-Tundra at 1100 bp
There are 28 sites in ecoregion Forest-Tundra at 1200 bp
There are 28 sites in ecoregion Forest-Tundra at 1300 bp
There are 28 sites in ecoregion Forest-Tundra at 1400 bp
There are 28 sites in ecoregion Forest-Tundra at 1500 bp
There are 28 sites in ecoregion Forest-Tundra at 1600 bp
There are 28 sites in ecoregion Forest-Tundra at 1700 bp
There are 28 sites in ecoregion Forest-Tundra at 1800 bp
There are 28 sites in ecoregion Forest-Tundra at 1900 bp
There are 28 sites in ecoregion Forest-Tundra at 2000 bp
There are 64 sites in ecoregion Conifer/Hardwood at 0 bp
There are 79 sites in ecoregion Conifer/Hardwood at 100 bp
There are 87 sites in ecoregion Conifer/Hardwood at 200 bp
There are 88 sites in ecoregion Conifer/Hardwood at 300 bp
There are 90 sites in ecoregion Conifer/Hardwood at 400 bp
There are 90 sites in ecoregion Conifer/Hardwood at 500 bp
There are 92 sites in ecoregion Conifer/Hardwood at 600 bp
There are 92 sites in ecoregion Conifer/Hardwood at 700 bp
There are 95 sites in ecoregion Conifer/Hardwood at 800 bp
There are 94 sites in ecoregion Conifer/Hardwood at 900 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1000 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1100 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1200 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1300 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1400 bp
```

```
There are 94 sites in ecoregion Conifer/Hardwood at 1500 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1600 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1700 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1800 bp
There are 94 sites in ecoregion Conifer/Hardwood at 1900 bp
There are 92 sites in ecoregion Conifer/Hardwood at 2000 bp
There are 78 sites in ecoregion Boreal Forest at 0 bp
There are 101 sites in ecoregion Boreal Forest at 100 bp
There are 106 sites in ecoregion Boreal Forest at 200 bp
There are 109 sites in ecoregion Boreal Forest at 300 bp
There are 111 sites in ecoregion Boreal Forest at 400 bp
There are 114 sites in ecoregion Boreal Forest at 500 bp
There are 118 sites in ecoregion Boreal Forest at 600 bp
There are 120 sites in ecoregion Boreal Forest at 700 bp
There are 121 sites in ecoregion Boreal Forest at 800 bp
There are 122 sites in ecoregion Boreal Forest at 900 bp
There are 122 sites in ecoregion Boreal Forest at 1000 bp
There are 123 sites in ecoregion Boreal Forest at 1100 bp
There are 123 sites in ecoregion Boreal Forest at 1200 bp
There are 123 sites in ecoregion Boreal Forest at 1300 bp
There are 123 sites in ecoregion Boreal Forest at 1400 bp
There are 123 sites in ecoregion Boreal Forest at 1500 bp
There are 124 sites in ecoregion Boreal Forest at 1600 bp
There are 124 sites in ecoregion Boreal Forest at 1700 bp
There are 124 sites in ecoregion Boreal Forest at 1800 bp
There are 125 sites in ecoregion Boreal Forest at 1900 bp
There are 125 sites in ecoregion Boreal Forest at 2000 bp
There are 163 sites in ecoregion Deciduous Forest at 0 bp
There are 186 sites in ecoregion Deciduous Forest at 100 bp
There are 187 sites in ecoregion Deciduous Forest at 200 bp
There are 185 sites in ecoregion Deciduous Forest at 300 bp
There are 180 sites in ecoregion Deciduous Forest at 400 bp
There are 177 sites in ecoregion Deciduous Forest at 500 bp
There are 177 sites in ecoregion Deciduous Forest at 600 bp
There are 173 sites in ecoregion Deciduous Forest at 700 bp
There are 170 sites in ecoregion Deciduous Forest at 800 bp
There are 169 sites in ecoregion Deciduous Forest at 900 bp
There are 166 sites in ecoregion Deciduous Forest at 1000 bp
There are 164 sites in ecoregion Deciduous Forest at 1100 bp
There are 162 sites in ecoregion Deciduous Forest at 1200 bp
There are 162 sites in ecoregion Deciduous Forest at 1300 bp
There are 161 sites in ecoregion Deciduous Forest at 1400 bp
There are 158 sites in ecoregion Deciduous Forest at 1500 bp
There are 156 sites in ecoregion Deciduous Forest at 1600 bp
There are 156 sites in ecoregion Deciduous Forest at 1700 bp
There are 155 sites in ecoregion Deciduous Forest at 1800 bp
There are 154 sites in ecoregion Deciduous Forest at 1900 bp
There are 153 sites in ecoregion Deciduous Forest at 2000 bp
There are 66 sites in ecoregion Prairies at 0 bp
There are 77 sites in ecoregion Prairies at 100 bp
There are 77 sites in ecoregion Prairies at 200 bp
There are 76 sites in ecoregion Prairies at 300 bp
There are 70 sites in ecoregion Prairies at 400 bp
There are 66 sites in ecoregion Prairies at 500 bp
There are 62 sites in ecoregion Prairies at 600 bp
There are 61 sites in ecoregion Prairies at 700 bp
There are 62 sites in ecoregion Prairies at 800 bp
There are 61 sites in ecoregion Prairies at 900 bp
There are 63 sites in ecoregion Prairies at 1000 bp
There are 63 sites in ecoregion Prairies at 1100 bp
There are 62 sites in ecoregion Prairies at 1200 bp
There are 62 sites in ecoregion Prairies at 1300 bp
There are 61 sites in ecoregion Prairies at 1400 bp
There are 61 sites in ecoregion Prairies at 1500 bp
There are 61 sites in ecoregion Prairies at 1600 bp
There are 61 sites in ecoregion Prairies at 1700 bp
There are 61 sites in ecoregion Prairies at 1800 bp
There are 61 sites in ecoregion Prairies at 1900 bp
There are 61 sites in ecoregion Prairies at 2000 bp
There are 5 sites in ecoregion Arctic at 0 bp
There are 7 sites in ecoregion Arctic at 100 bp
There are 8 sites in ecoregion Arctic at 200 bp
There are 8 sites in ecoregion Arctic at 300 bp
There are 9 sites in ecoregion Arctic at 400 bp
There are 9 sites in ecoregion Arctic at 500 bp
There are 10 sites in ecoregion Arctic at 600 bp
```

```
There are 10 sites in ecoregion Arctic at 700 bp
There are 10 sites in ecoregion Arctic at 800 bp
There are 10 sites in ecoregion Arctic at 900 bp
There are 10 sites in ecoregion Arctic at 1000 bp
There are 10 sites in ecoregion Arctic at 1100 bp
There are 10 sites in ecoregion Arctic at 1200 bp
There are 10 sites in ecoregion Arctic at 1300 bp
There are 10 sites in ecoregion Arctic at 1400 bp
There are 10 sites in ecoregion Arctic at 1500 bp
There are 10 sites in ecoregion Arctic at 1600 bp
There are 10 sites in ecoregion Arctic at 1700 bp
There are 10 sites in ecoregion Arctic at 1800 bp
There are 10 sites in ecoregion Arctic at 1900 bp
There are 10 sites in ecoregion Arctic at 2000 bp
There are 3 sites in ecoregion Coastal at 0 bp
There are 12 sites in ecoregion Coastal at 100 bp
There are 13 sites in ecoregion Coastal at 200 bp
There are 15 sites in ecoregion Coastal at 300 bp
There are 17 sites in ecoregion Coastal at 400 bp
There are 17 sites in ecoregion Coastal at 500 bp
There are 17 sites in ecoregion Coastal at 600 bp
There are 17 sites in ecoregion Coastal at 700 bp
There are 17 sites in ecoregion Coastal at 800 bp
There are 17 sites in ecoregion Coastal at 900 bp
There are 17 sites in ecoregion Coastal at 1000 bp
There are 17 sites in ecoregion Coastal at 1100 bp
There are 17 sites in ecoregion Coastal at 1200 bp
There are 17 sites in ecoregion Coastal at 1300 bp
There are 17 sites in ecoregion Coastal at 1400 bp
There are 17 sites in ecoregion Coastal at 1500 bp
There are 17 sites in ecoregion Coastal at 1600 bp
There are 17 sites in ecoregion Coastal at 1700 bp
There are 17 sites in ecoregion Coastal at 1800 bp
There are 17 sites in ecoregion Coastal at 1900 bp
There are 17 sites in ecoregion Coastal at 2000 bp
There are 31 sites in ecoregion Mountain vegetation at 0 bp
There are 34 sites in ecoregion Mountain vegetation at 100 bp
There are 35 sites in ecoregion Mountain vegetation at 200 bp
There are 36 sites in ecoregion Mountain vegetation at 300 bp
There are 38 sites in ecoregion Mountain vegetation at 400 bp
There are 41 sites in ecoregion Mountain vegetation at 500 bp
There are 41 sites in ecoregion Mountain vegetation at 600 bp
There are 41 sites in ecoregion Mountain vegetation at 700 bp
There are 41 sites in ecoregion Mountain vegetation at 800 bp
There are 41 sites in ecoregion Mountain vegetation at 900 bp
There are 41 sites in ecoregion Mountain vegetation at 1000 bp
There are 41 sites in ecoregion Mountain vegetation at 1100 bp
There are 41 sites in ecoregion Mountain vegetation at 1200 bp
There are 41 sites in ecoregion Mountain vegetation at 1300 bp
There are 41 sites in ecoregion Mountain vegetation at 1400 bp
There are 41 sites in ecoregion Mountain vegetation at 1500 bp
There are 41 sites in ecoregion Mountain vegetation at 1600 bp
There are 41 sites in ecoregion Mountain vegetation at 1700 bp
There are 41 sites in ecoregion Mountain vegetation at 1800 bp
There are 41 sites in ecoregion Mountain vegetation at 1900 bp
There are 41 sites in ecoregion Mountain vegetation at 2000 bp
There are 17 sites in ecoregion Desert at 0 bp
There are 24 sites in ecoregion Desert at 100 bp
There are 26 sites in ecoregion Desert at 200 bp
There are 27 sites in ecoregion Desert at 300 bp
There are 27 sites in ecoregion Desert at 400 bp
There are 28 sites in ecoregion Desert at 500 bp
There are 28 sites in ecoregion Desert at 600 bp
There are 28 sites in ecoregion Desert at 700 bp
There are 28 sites in ecoregion Desert at 800 bp
There are 28 sites in ecoregion Desert at 900 bp
There are 28 sites in ecoregion Desert at 1000 bp
There are 28 sites in ecoregion Desert at 1100 bp
There are 28 sites in ecoregion Desert at 1200 bp
There are 28 sites in ecoregion Desert at 1300 bp
There are 28 sites in ecoregion Desert at 1400 bp
There are 28 sites in ecoregion Desert at 1500 bp
There are 28 sites in ecoregion Desert at 1600 bp
There are 28 sites in ecoregion Desert at 1700 bp
There are 28 sites in ecoregion Desert at 1800 bp
There are 28 sites in ecoregion Desert at 1900 bp
```

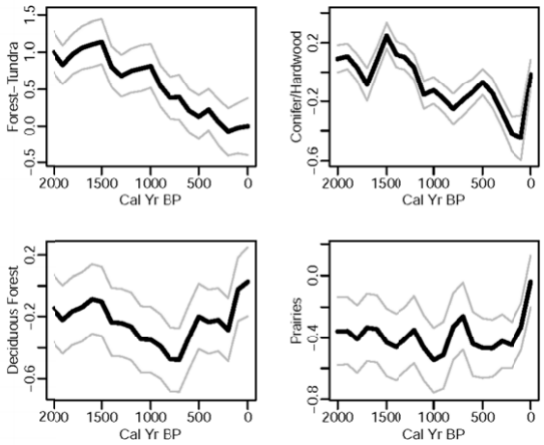


Figure 22: Plot of reconstructed TAVE anomalies through time by ecoregion. NOTE: Only 4 ecoregions are shown.

```
There are 28 sites in ecoregion Desert at 2000 bp
There are 17 sites in ecoregion Mediterranian at 0 bp
There are 24 sites in ecoregion Mediterranian at 100 bp
There are 26 sites in ecoregion Mediterranian at 200 bp
There are 27 sites in ecoregion Mediterranian at 300 bp
There are 27 sites in ecoregion Mediterranian at 400 bp
There are 28 sites in ecoregion Mediterranian at 500 bp
There are 28 sites in ecoregion Mediterranian at 600 bp
There are 28 sites in ecoregion Mediterranian at 700 bp
There are 28 sites in ecoregion Mediterranian at 800 bp
There are 28 sites in ecoregion Mediterranian at 900 bp
There are 28 sites in ecoregion Mediterranian at 1000 bp
There are 28 sites in ecoregion Mediterranian at 1100 bp
There are 28 sites in ecoregion Mediterranian at 1200 bp
There are 28 sites in ecoregion Mediterranian at 1300 bp
There are 28 sites in ecoregion Mediterranian at 1400 bp
There are 28 sites in ecoregion Mediterranian at 1500 bp
There are 28 sites in ecoregion Mediterranian at 1600 bp
There are 28 sites in ecoregion Mediterranian at 1700 bp
There are 28 sites in ecoregion Mediterranian at 1800 bp
There are 28 sites in ecoregion Mediterranian at 1900 bp
There are 28 sites in ecoregion Mediterranian at 2000 bp

NOTE: Mediterranian and no data ecoregions are excluded.

Please enter the units for the y-axis (e.g. for ANNP enter 'mm', for MTWA enter 'C') C
Do you wish to include ecoregion Forest-Tundra <1 for yes, 2 for no> 1
Do you wish to include ecoregion Conifer/Hardwood <1 for yes, 2 for no> 1
Do you wish to include ecoregion Boreal Forest <1 for yes, 2 for no> 1
Do you wish to include ecoregion Deciduous Forest <1 for yes, 2 for no> 1
Do you wish to include ecoregion Prairies <1 for yes, 2 for no> 1
Do you wish to include ecoregion Arctic <1 for yes, 2 for no> 1
Do you wish to include ecoregion Coastal <1 for yes, 2 for no> 1
Do you wish to include ecoregion Mountain vegetation <1 for yes, 2 for no> 1
Do you wish to include ecoregion Desert <1 for yes, 2 for no> 1
Plot output to file avgAnomalyPlotByEcoregions.pdf in working directory!
```

The file anomByCalYrERfinal.csv will contain the average anomaly (AvgAn) and standard error (StdErr) for each ecoregion (forest-tundra is shown below):

If a reconstruction file is missing, this will be indicated:

```
Reconstruction file is not interpolated! (Run menu item 7)
```

11. Get average anomalies for all sites by time slice (optional: output csv file)

Table 11: Reconstructed TAVE anomalies through time (with standard errors) for the forest-tundra ecoregion.

CalMean	AvgAn_FT	StdErr_FT
0	-0.01611	0.382764
100	-0.03763	0.338016
200	-0.08846	0.315908
300	0.041912	0.311797
400	0.208761	0.283228
500	0.112514	0.295076
600	0.195785	0.295612
700	0.380479	0.31057
800	0.372658	0.287384
900	0.534064	0.28391
1000	0.810128	0.300559
1100	0.777534	0.316278
1200	0.743653	0.303031
1300	0.674033	0.287454
1400	0.800008	0.278293
1500	1.140638	0.304641
1600	1.103443	0.308142
1700	1.059147	0.292639
1800	0.970912	0.274276
1900	0.824626	0.264806
2000	1.002075	0.277893

This is a useful feature that allows you get the anomalies at each site averaged over a time period (e.g. 1300 to 1700 BP for the MWP). Then output a shapefile, then either map out these anomalies relative to modern or relative to another time slice file. For example you could output and then map MWP-modern, LIA-modern and then MWP-LIA.

The range of oldest to youngest must fall within the range of the data.

```
Your data has a range of 0 to 2000 calendar years BP
Example of time slice anomaly (cal yr bp interval): 800 (youngest) to 1200 (oldest) or 1000 (youngest) to 1000 (oldest)
What is the start interval (youngest) of the time slice in Cal Yr BP? 200
What is the stop interval (oldest) of the time slice in Cal Yr BP? 800
How do you wish to calculate the average anomalies by time slice?
A site must be in each time interval of the entire range <enter 1> or a site must be in at least one time interval of the range <enter
      2> to be part of the calculated average: 2
Writing file AvgAnomByTimeSlice.csv with SiteID, Lon, Lat and avg anomaly for time slice to working directory...


Do you wish to output a shapefile of the avg anomaly data for selected time slice? <1 for YES, 2 for NO> 1
```

If 1 is selected, a .dbf, .shx and .shp file will be written to disk. This shape file can be loaded into a GIS program for interpolation and mapping.

```
Writing file AvgAnomByTimeSlice shapefile with SiteID, Lon, Lat and avg anomaly for time slice to working directory...
```

Table 12: Reconstructed TAVE anomalies for the time slice of 200-800 BP (1550-1850 CE).

Site	Lon	Lat	200to800bp_AN
7	-80.1	47	0.382234
18	-97.5	29.6	-1.84542
29	-94.5	44.8	-2.09917
196	-89.7	46.2	0.632198
197	-89.9	46.3	0
203	-85.5	36	-2.3717
206	-128.1	64.7	1.020749
207	-153.9	67.1	-0.02342
208	-70.7	47.5	0.231063

Here is where you can compare the entered range (e.g. 200-800 BP in this case) with another anomaly time slice file (e.g. 1300-1700 BP) by entering 1 for this option. If you are only comparing the average of 200 to 800 BP to modern, enter 2 for this option.

```
Map will be created for North America
Do you want to interpolate the output with kriging and then create a map? <1 for YES, 2 for NO> 1
```

Presently, the kriging uses the autokrige function with no search radius (by default). The purpose of these maps is to get a sense of the patterns. For publication quality maps, the output should be interpolated and mapped in a GIS program.

A map of anomalies for the selected time slice is displayed with the details below:

```
There are 316 sites with negative anomalies
There are 103 sites with zero anomalies
There are 218 sites with positive anomalies
```

Note that maps will reflect the study area selected. The default is all of North America unless a spatial subset is selected. Here we can see the heterogeneity of the reconstruction site anomalies.

This provides some information about the extreme anomaly values:

```
Max average anomaly difference value is 5.24359497645212
Min average anomaly difference value is -14.8809278350515
What is the max/min anomaly range for map? (e.g. if 1 is entered, max will be 1, min will be -1) Should be based on min/max values
      above or plot could fail!) 1
```

The may be difficult to interpret if the max/min anomaly range is not correctly specified.

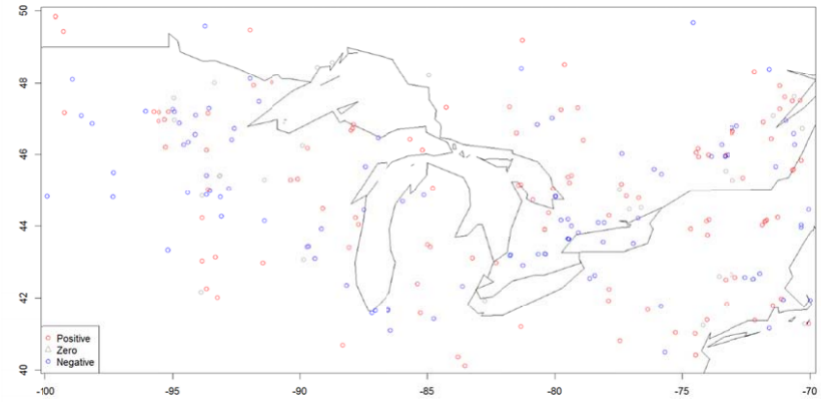


Figure 23: Map of TAVE anomalies for selected time slice, classified into negative, zero and positive categories.

Duplicates are removed here (NOTE: lat/lon values for each site have been converted to X-Y coordinates before performing the kriging).

```
Warning in autoKrige(avgAN ~ 1, project_df) :
  Removed 66 duplicate observation(s) in input_data:
            coordinates avgAN
 [using ordinary kriging]
```

Here, use a descriptive title for the output map:

```
Enter the title/file name for this plot: TAVE (LIA-mod)
```

Here you need to be careful with the number of colour levels chosen. For example, 20 colour levels will often work well. If the number of colour levels is not appropriate, the map may be difficult to interpret. If option 1 is chosen, a grid cell size must be appropriate for the analysis.

```
Do you want a gridded or filled contour map? <1 for GRIDDED, 2 for FILLED CONTOUR> 1
How many colour levels should the output map have, must be between 10 and 100 (use higher values for a less blocky appearance): 20
How many rows/columns should the gridded map have, must be between 10 and 100 (use higher values for smaller grid boxes): 50
```

Same plot but using filled contour instead of grid boxes (anomalies greater than +/- 1°C appear as transparent:

```
Do you want to do mapping of avg anomalies for sequential time intervals (oldest to youngest)? <1 for YES, 2 for NO> 2
```

This option allows for maps to be created in sequence (e.g. 2000-1900 BP, 1900-1800 BP, 1800-1700 BP,, 100-0 BP). A series of pdf maps will be output here depending on the starting and ending calendar year bp interval and the linear interpolation interval. For a 2,000 year reconstruction at 100 year intervals, a total of 20 maps will be output.

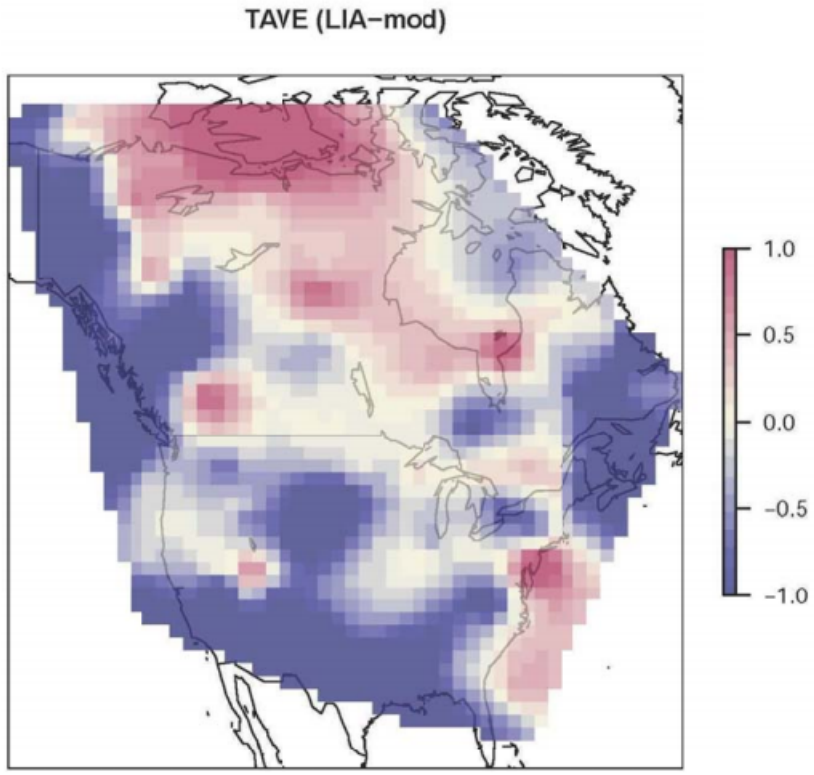


Figure 24: Gridded map of TAVE anomalies for LIA relative to modern.

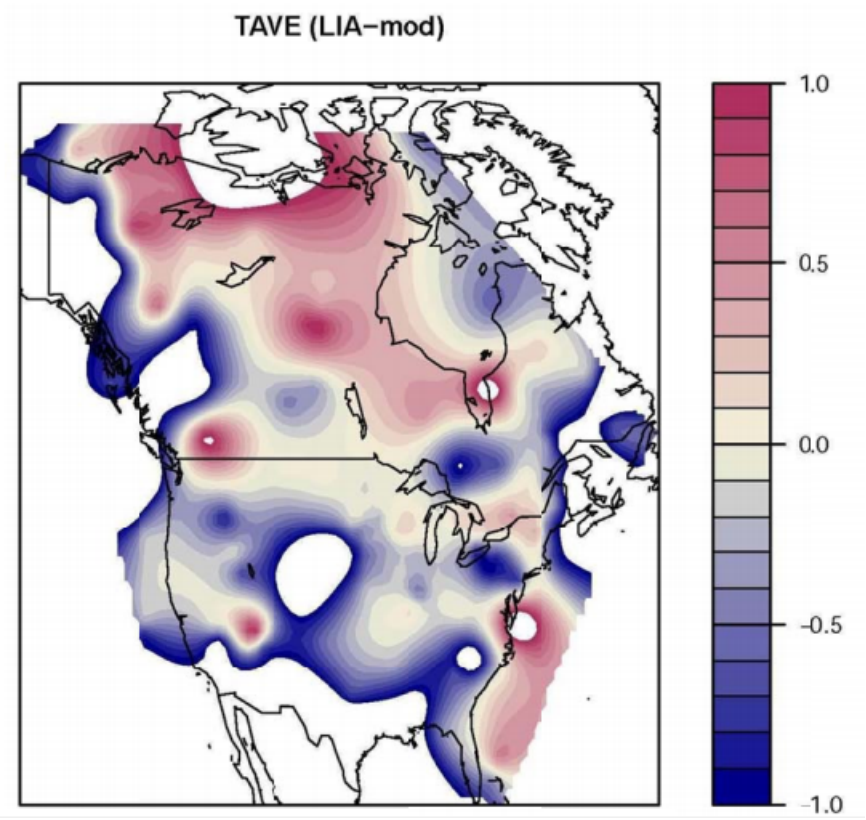


Figure 25: Contour map of TAVE anomalies for LIA relative to modern.

```
Processing average anomaly difference between 2000 and 1900 ....
Max average anomaly difference value is 3.80435443037975
Min average anomaly difference value is -2.63959390862944

What is the max/min range for map anomalies? (e.g. if 2 is entered, max will be 2, min will be -2). Should be based on min/max values
      above or plot could fail!) 2
[using ordinary kriging]

Do you want a gridded or filled contour map? <1 for GRIDDED, 2 for FILLED CONTOUR> 2

How many colour levels should the output map have, must be between 10 and 100 (use higher values for a less blocky appearance): 10

Processing average anomaly difference between 1900 and 1800 ....
Max average anomaly difference value is 5.13931034482759
Min average anomaly difference value is -3.9
[using ordinary kriging]

Processing average anomaly difference between 1800 and 1700 ....
Max average anomaly difference value is 6.0690100430416
Min average anomaly difference value is -4.34336713995943
[using ordinary kriging]
```

Finally if an oldest calendar year bp value is not specified (default youngest calendar BP value is 0), the function will not run.

```
Oldest calendar year BP is missing! (Run menu item 4)
```

12. Create Squared-Chord Distance (SCD) plots by ecoregion (optional)

    There is the option for a threshold line and to exclude any ecoregion.

```
Writing file SCDbyEcoregionFinal.csv to working directory...
Writing plot to file SCD_ecoregions_plot.pdf in working directory...
At what SCD value do you want the horizontal line placed? (valid range is 0 to 1, use -1 for no line) -1
Do you wish to include ecoregion Arctic <1 for yes, 2 for no> 1
Do you wish to include ecoregion Boreal Forest <1 for yes, 2 for no> 1
Do you wish to include ecoregion Deciduous Forest <1 for yes, 2 for no> 1
Do you wish to include ecoregion Coastal <1 for yes, 2 for no> 1
Do you wish to include ecoregion Desert <1 for yes, 2 for no> 1
Do you wish to include ecoregion Mountain vegetation <1 for yes, 2 for no> 1
Do you wish to include ecoregion Prairies <1 for yes, 2 for no> 1
Do you wish to include ecoregion Conifer/Hardwood <1 for yes, 2 for no> 1
Do you wish to include ecoregion Forest-Tundra <1 for yes, 2 for no> 1
```

    You need to have a reconstruction file loaded (menu item #1) and an analog selected (menu item #6).

13. Plot SCD values from reconstruction file through time for all regions (optional)

    There is the option for a threshold line.

```
At what SCD value do you want the vertical line placed? (valid range is 0 to 1, use -1 for no line) 0.3
Writing plot to file SCD_allregions.pdf in working directory...
```

    You need to have run SCD by ecoregion before running this option.

```
You need to run menu item 12 first!
```

14. Calculate and plot site density through time (optional)

```
Writing file siteDensity.csv to working directory...
Writing plot to file siteDensity.pdf in working directory...
```

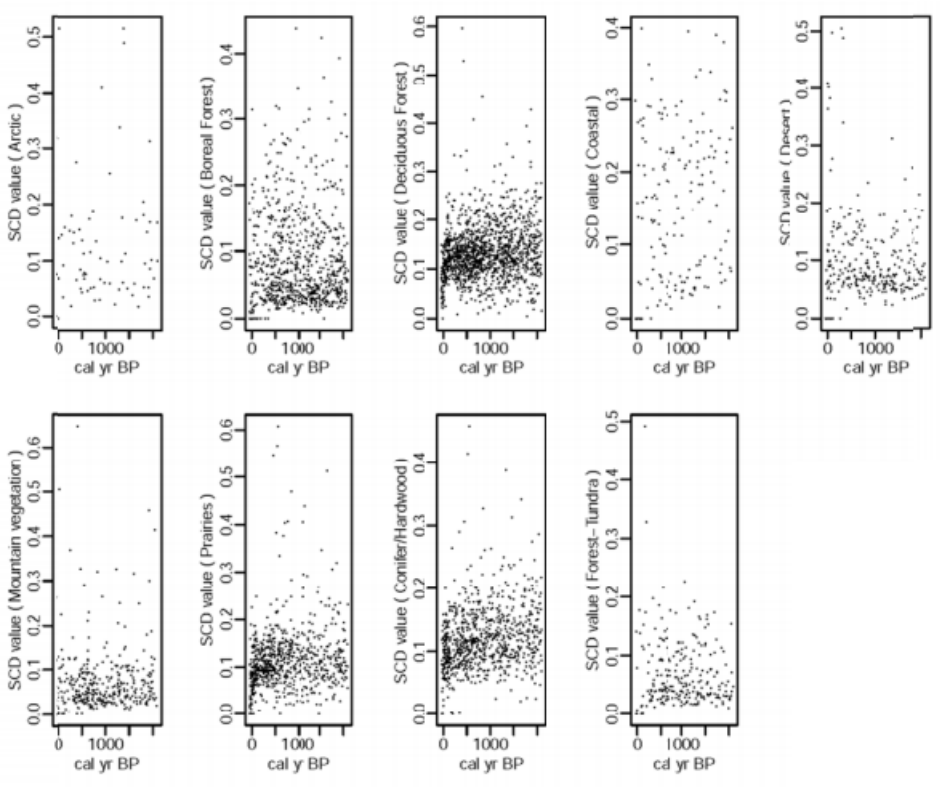


Figure 26: Squared-Chord Distance (SCD) plots by ecoregion.

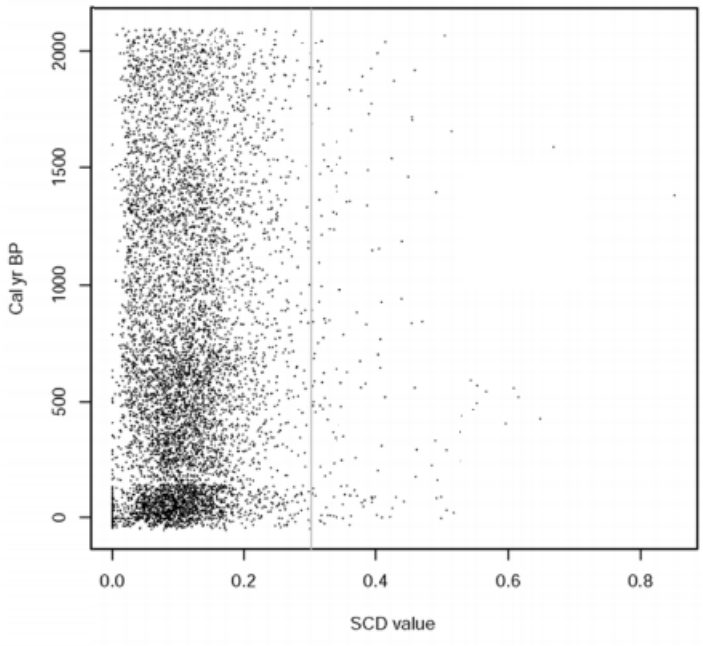


Figure 27: Squared-Chord Distance (SCD) plot for all of North America.

Table 13: Reconstructed site density through time in North America

calYrBP	siteCount
0	457
100	557
200	579
300	583
400	584
500	583
600	584
700	581
800	582
900	578
1000	573
1100	568
1200	563
1300	556
1400	541
1500	525
1600	497
1700	471
1800	431
1900	333
2000	158

The output file siteDensity.csv is shown below:

You need to have an interpolated reconstruction file (by calendar year BP interval) otherwise the following message is displayed:

```
No interpolated reconstruction file loaded! (Run menu item 7)
```

15. Spatial filtering of reconstruction based on calculated local Moran's I values (optional)

Spatial filtering allows reconstruction sites to be filtered out based on Moran's I and Moran's I z-values calculated from the anomaly file created in menu item 8. Spatial filtering ensures that sites retained are showing climate variability at the appropriate scale. When running menu item 8 you can filter the reconstruction and then compare it to an unfiltered reconstruction. An existing anomaly file can also be filtered in menu item 15 but no plots/maps can be created here. Menu item 15 is useful if the user simply wants to map the csv output in a GIS program. This function requires an anomaly file created in menu item 8 (i.e. SpatialJoinReconModWithAnCalc.csv)

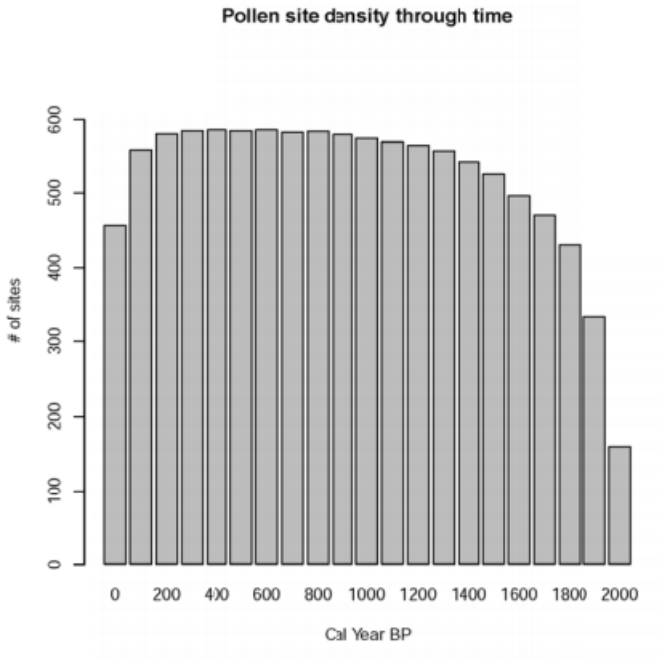


Figure 28: Plot of reconstructed site density through time in North America.

Make sure you are not spatial filtering an anomaly file that is already spatially filtered in menu item 8!

Range of reconstruction is 0 to 2500 bp at 100 year intervals

Enter appropriate lag distances (first, last) and lag interval for the filtering.

```
Enter the lag interval in km (must be either 10, 25 or 50):
50
Enter the first lag distance (in km) of points for which to calculate Moran's I values (must be between 50 and 500 km, rounded down to
      nearest 50k interval):
50
Enter the last lag distance (in km) of points for which to calculate Moran's I values (must be between 50 and 500 km, rounded down to
      nearest 50k interval):
250
```

Select the options for filtering. It is recommended to filter by anomalies (option 3), keeping MI zero values (option 1) and retain only MI pos and zMI pos values (option 1):

```
Do you want to filter by reconstructed values <enter 1>, modern values <enter 2> or anomalies (reconstructed minus modern) <enter 3> ?
3
Do you want to keep MI zero values <enter 1> or remove them <enter 2> after filtering at the last lag ?
1
Do you want to retain only MI pos and zMi pos values <enter 1> or both pos/pos and pos/neg <enter 2> ?
1
```

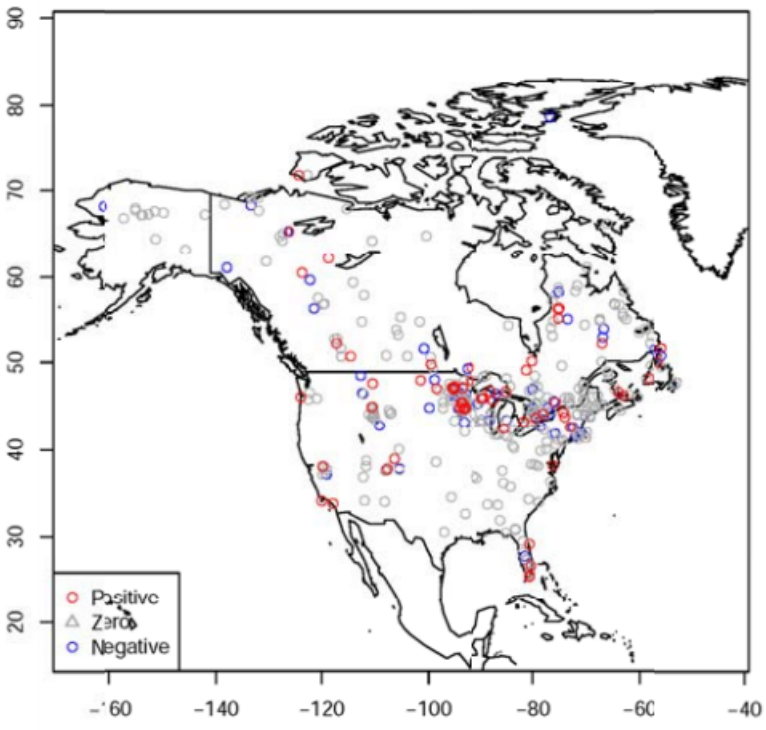


Figure 29: Map of filtered TAVE anomalies for selected time slice, classified into negative, zero and positive categories.

The p-value cutoff is not used to filter sites, it is simply used in the counts file, to show you how many of the sites have Moran's I values that are significant/not-significant.

```
What is the p-value cutoff? (p-value cutoff must be between 0.01 [99% sig level] and 0.4 [60% sig level])
0.2
```

If you want a map of anomalies at each time interval, select option 1.

```
Do you want to map the anomalies (negative/zero/positive)? <1 for YES, 2 for NO>
1
```

The next option is for aggregating counts (with percentages) of Moran's I statistics over specific calendar year bp range (e.g. 1300 to 1700 BP). The format of the output table is the same as single calendar year bp intervals.

```
Do you want to aggregate counts for a range of cal year intervals for a specified lag? <1 for YES, 2 for NO>
2
```

An example of nearest neighbor summary information is shown below.

```
Do you want to display nearest neighbor summary information at each lag for each cal yr interval? <1 for YES, 2 for NO>
2

Neighbour list object:
Number of regions: 455
Number of nonzero links: 1074
Percentage nonzero weights: 0.5187779
Average number of links: 2.36044
```

```
159 regions with no links:
2 5 7 13 19 20 21 26 27 28 29 31 33 36 42 45 53 54 58 61 64 65 66 67 69 71 72 74 75 77 78 80 82 83 86 87 88 90 91 92 94 98 100 103 105
      112 113 117 119 122 123 126 128 130 135 136 138 139 141 147 151 153 156 160 163 166 167 169 173 175 181 182 184 185 188 198 203
      206 207 209 213 214 215 221 228 230 231 232 234 243 251 253 256 262 263 272 275 277 279 282 286 287 294 295 300 301 306 310 312
      313 315 316 324 325 329 330 333 334 338 339 340 341 343 346 349 351 352 353 361 362 363 364 365 368 371 374 381 382 385 389 393
      397 399 403 408 412 419 424 425 429 431 434 436 438 439 441 442 446 447
Link number distribution:

  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 16 17
159 101 66 37 19 20 11 3 1 6 4 5 6 10 3 2 2
101 least connected regions:
6 8 9 16 40 49 60 70 73 81 84 93 101 106 107 110 120 124 127 144 152 155 157 161 164 165 168 180 187 190 192 193 196 197 208 210 217
      218 224 225 226 227 233 235 236 237 239 242 248 252 254 260 266 270 271 273 276 278 280 296 304 307 318 321 326 327 328 332 335
      336 337 342 347 348 354 358 369 375 377 379 387 388 398 401 406 410 411 414 416 418 422 426 428 430 444 449 450 451 452 453 454
      with 1 link
2 most connected regions:
39 317 with 17 links

There are 4 possibilities for combinations of Moran's I and Moran's I z-values:

positive/positive (criteria 1)
negative/negative (criteria 2)
mixed signs either pos/neg or neg/pos (criteria 3)
all zeroes (criteria 4)

All rows (sites) that do not meet either criteria 1 or criteria 4 will be removed at each lag until the last lag where only sites
      meeting criteria 1 will be retained.

Summary information is displayed at each step of the filtering process:


Total sites for 0 bp: 455
Total sites for 0 bp input into lag distance of 50 km: 455

Sum of Moran's I is 119.08

Writing table of statistics for filtered reconstruction to file filtered_recon_stats_50km_0bp.csv ....
Total sites for 0 bp filtered out at lag distance of 50 km: 21
Total sites for 0 bp retained at lag distance of 50 km: 434
At 0 bp for a lag distance of 50 km:
There are 63 sites with negative anomalies
There are 304 sites with zero anomalies
There are 67 sites with positive anomalies
Total sites for 0 bp input into lag distance of 100 km: 434

Sum of Moran's I is 24.47

Writing table of statistics for filtered reconstruction to file filtered_recon_stats_100km_0bp.csv ....
Total sites for 0 bp filtered out at lag distance of 100 km: 2
Total sites for 0 bp retained at lag distance of 100 km: 432
At 0 bp for a lag distance of 100 km:
There are 61 sites with negative anomalies
There are 304 sites with zero anomalies
There are 67 sites with positive anomalies
Total sites for 0 bp input into lag distance of 150 km: 432

Sum of Moran's I is -123.63

Writing table of statistics for filtered reconstruction to file filtered_recon_stats_150km_0bp.csv ....
Total sites for 0 bp filtered out at lag distance of 150 km: 9
Total sites for 0 bp retained at lag distance of 150 km: 423
At 0 bp for a lag distance of 150 km:
There are 58 sites with negative anomalies
There are 304 sites with zero anomalies
There are 61 sites with positive anomalies
Total sites for 0 bp input into lag distance of 200 km: 423

Sum of Moran's I is -167.73

Writing table of statistics for filtered reconstruction to file filtered_recon_stats_200km_0bp.csv ....
Total sites for 0 bp filtered out at lag distance of 200 km: 4
Total sites for 0 bp retained at lag distance of 200 km: 419
At 0 bp for a lag distance of 200 km:
There are 58 sites with negative anomalies
There are 304 sites with zero anomalies
There are 57 sites with positive anomalies
```

```
Total sites for 0 bp input into lag distance of 250 km: 419

Sum of Moran's I is -228.57

Writing table of statistics for filtered reconstruction to file filtered_recon_stats_250km_0bp.csv ....
Total sites for 0 bp filtered out at lag distance of 250 km: 24
Total sites for 0 bp retained at lag distance of 250 km: 395
At 0 bp for a lag distance of 250 km:
There are 55 sites with negative anomalies
There are 286 sites with zero anomalies
There are 54 sites with positive anomalies

Total sites for 100 bp: 553
Total sites for 100 bp input into lag distance of 50 km: 553

This is repeated for each lag of each calendar year bp interval in the reconstruction. You will be asked to create a table of spatial
      filtering results at the last lag distance for either all calendar year intervals or selected calendar year intervals in the
      reconstruction.

The output files created are as follows:
1 filteredRecon_LAGDISTkm_CALYRbp.csv
2 recon_unfiltered_LAGDISTkm_CALYRbp.csv
3 filtered_recon_stats_LAGDISTkm_CALYRbp.csv
4 filteredReconstructionAllCalYrBP.csv
```

For file output types 1-3, a separate file is output for each LAGDIST and each CALYR.

The last file contains all sites for the entire reconstruction that are retained after filtering at the last lag distance and the associated MI, zMI, and pMI values for each site. The filtered stats contains counts and percentages of positive/negative/zero values for MI and zMI, counts of combinations of these two fields (so positive/positive, negative/negative, mixed and zeroes) and counts of pMI (significant), not significant, and zero. There are two total sites fields, the top row is sites input into this lag distance and the bottom row is total sites for the selected cal yr BP interval input into the first lag distance (before filtering).

In order to perform spatial filtering of a reconstruction, you need an average anomaly file created from the reconstruction otherwise the following will be displayed:

```
No reconstruction anomaly file loaded! (Run menu item 8)
```

16. Perform a data-data or data-model comparison (beta-version, the full functionality is not yet available)

    Basically we load the netCDF file, get the lat,lon and val from the reconstruction file, find nearest model gridpoint to each recon point, do spatial join

between each site and the nearest model grid point, then simply subtract modelVal - reconVal and output the results (with lat/lon) to a csv file. The actual method of comparison between the reconstruction points and the model output will be further developed in a later version of this program.

```
Data-model comparison feature will enable the user to do a comparison between a reconstruction created by this program
and paleoclimate model output imported in NetCDF (.nc) format. NOTE: This functionality is not available yet!

Please select a paleoclimate model output file in netCDF format (.nc).....
```

First, the user selects a paleoclimate model output file in NetCDF (.nc) format. The list of variables in the netCDF file is displayed on screen.

It is possible that the fields could be latitude/longitude or lat/lon.

```
What is the name of the latitude field in the model output from above (e.g. lat) lat
What is the name of the longitude field in the model output from above (e.g. lon) lon

What is the name of the variable of interest in the model output from above (e.g. TS) TS
```

In this case surface temperature (TS) is selected from the list of variables above.

```
NOTE: 3 dimensions is [longitude,latitude,time] \& 4 dimensions is [longitude,latitude,time,level]
```

Make sure this value entered is correct.

```
How many dimensions in the variable of interest from above (must be either be 3 or 4) 3
```

Same options with spatial join as in menu item #8.

```
Which method of spatial join do you wish to use? <1 for Great Circle distance, 2 for Euclidean> 2

Writing data-model comparison table to working directory in .csv format...
```

The output file will have lat, lon, diff where diff is the difference between the value for that site (for selected time slice) and the value of the nearest model grid point.

If the output file is open the following message will be generated.

```
dataModelCompOutput.csv cannot be written to disk, please close the file!
```

If an invalid .nc file is chosen, the following message will be displayed:

```
Invalid netCDF file selected!
```

You need a time slice of average anomalies by calendar year BP range otherwise the following message is displayed:

```
No reconstruction time slice file loaded! (Run menu item 11)
```

17. Set or change the current working directory (optional)

    If the working directory specified exists, it will be changed, otherwise it will not be changed. In this case C:/pollen exists on the hard drive.

```
Enter the path to working directory (e.g. C:/pollen)
Working directory set to C:/pollen
```

18. List all program files (.pdf, .csv, .recon) in working directory (optional)

```
Listing all files in working directory with extension .recon, .pdf and .csv....
[1] "PS64_RECONS_ANNP.recon" "PS64_RECONS_GDD5.recon" "PS64_RECONS_MIPT.recon" "PS64_RECONS_MTCO.recon"
[5] "PS64_RECONS_MTWA.recon" "PS64_RECONS_TAVE.recon"
 [1] "avgAnomalyPlot.pdf" "avgAnomalyPlot_filtered.pdf" "avgAnomalyPlotByEcoregions.pdf"
 [4] "fossilPollenSiteMap.pdf" "modernPollenSiteMap.pdf" "MTCO(MWP-mod).pdf"
 [7] "MTWA(lia-mod)filtered.pdf" "MTWA(lia-mod)unfiltered.pdf" "MTWA_MWP-mod.pdf"
[10] "MTWAforMWP(filtered).pdf" "PollenSiteMap.pdf" "samplesThroughTimePlot.pdf"
[13] "SCD_allregions_plot.pdf" "SCD_ecoregions_plot.pdf" "siteDensityPlot.pdf"
 [1] "anByCalYrAg.csv" "anByCalYrCalc.csv" "anomByCalYrERfinal.csv"
 [4] "avgAnomalyByCalYrBP.csv" "avgAnomalyByCalYrBP_filtered.csv" "AvgAnomByTimeSlice_MTWA.csv
```

    If a certain file type is not found in the working directory this will be indicated:

```
No .pdf files found in working directory!
No .csv files found in working directory!
No .recon files found in working directory!
```

19. Start pollen database editor program (optional) This will load the menu from module 1.

```
[1] Create a custom pollen database file (optional)
[2] Load a pollen db file (default pollen sums available of 35,64,134,135 (Williams and Shuman, 2008) or 0 for custom pollen sum as
      created in [1] - default pollen sum is 135)
[3] Show which fossil and modern pollen databases are currently loaded (optional)
[4] Show list of pollen taxa in currently loaded modern & fossil pollen database (optional)
[5] Join custom climate data to modern pollen sites (default Whitmore et al., 2005 v 1.7.2 - optional: CRU, GHCN, NCEP, ERA-40, NARR,
      BEST, etc. can be imported by user)
[6] Select a climate variable to reconstruct (required - from modern climate database)
[7] Display which climate variable from modern database is selected (optional - does not depend on which modern climate data is
      selected)
[8] Do reconstruction of selected climate variable (require MATTOOLS v1.1 from CRAN - only necessary if no .recon file is available to
      be loaded)
[9] Create a map of pollen sites from database (optional: fossil, modern or both)
[10] List all program files in working directory (optional)
[11] Set or change the current working directory (optional)
[12] Start climate reconstruction analysis program (optional)
[13] Quit program and return to R command prompt
```

20. Quit program and return to R command prompt

```
Exiting program, returning to the R command prompt....
```

B.3 Acknowledgements

Thanks to Steve Mosher at the Berkeley Earth Surface Temperature Group for assistance and discussions to improve various aspects of several functions.

B.4 Known and documented bugs in the program

There may be additional bugs that are not listed here. Please contact the author of this program if you find one that is not listed here.

1. Sometimes the user will receive an out of memory error when running menu item 8 in module 1. Typically this will occur with less than 2 GB of RAM.

```
Building reconstruction file using MATTOOLS v1.1, this could take some time....
Error: cannot allocate vector of size 241 Kb
In addition: Warning messages:
1: In data.frame(..., check.names = FALSE) :
  Reached total allocation of 1022Mb: see help(memory.size)
2: In data.frame(..., check.names = FALSE) :
  Reached total allocation of 1022Mb: see help(memory.size)
```

2. There is a problem with building of pollen sum 35 and 64 database when starting the program (if these files are not already present). The 35 pollen sum percentages are slightly off and 4 columns in the 64 sum database have NAs. As a result the correct 35 and 64 files are included. Building of the 134 pollen sum database is correct.

3. The filtered reconstruction time slice anomaly map may sometimes appear blank with no data points. However, the csv file output is correct and can be mapped in a GIS program.

4. Warning about plot.axes() not being a graphical parameter.

```
Warning messages:
1: In plot.window(...) : "plot.axes" is not a graphical parameter
```

5. This error occurs when using R v 3.0.0 or higher.

   Running menu item 9 in module 1:

```
File pollenSiteMap.pdf will be written to working directory!
Loading required package: maps
Error in mapPollenSites(NULL, modernData, fossilData, box <- c(lonMax, :
        could not find function "extent"
```

END OF DOCUMENTATION FOR MODULE 2

Appendix C

Source Code for R Program

NOTE: the following functions have been improved with the help of Steve Mosher (affiliation: Berkeley Earth Surface Temperature group): Constants.R, createReconObj.R, getAnalogCount.R getTaxaColumns.R, mapPollenSites.R, pickClimateOption.R, selectClimateVariable.R, selectReconFile.R

To run the code first set the working directory to where all the .r files are located. Ensure that there are ps135perc.fossil and ps135perc.modern files in that directory.

Then type in the following commands:

source("globalsource.r")
MainProgram1() or MainProgram2()

```
build_35_64_134_pollen_db <- function(pollen135, PSUM, modern)
{
  # this function builds the 35, 64 and 134 pollen sum database files
  # first we need to make sure all column names (taxa) are in upper case
  colnames(pollen135) <- toupper(colnames(pollen135))

  if (modern)
  {
     ps134mod = pollen135 # 134 is almost identical to 135 so load ps135 file

     ps64modA = pollen135[,1:13] # gets the basic site info
     ps64modB = pollen135[,149:230] # gets the modern climate data & ecoregion

     ps35modA = pollen135[,1:13] # gets the basic site info
     ps35modB = pollen135[,149:230] # gets the modern climate data & ecoregion

  }else
  {
     ps134fos = pollen135

     ps64fos = pollen135[,1:13] # gets the basic site info
     ps35fos = pollen135[,1:13] # gets the basic site info

  }
   # pollen 35 sum db file will have 36 taxa, pollen 64 sum will have 63 taxa
   pollen35taxa <- as.data.frame(matrix(NA,nrow = nrow(pollen135), ncol = 36))
   pollen64taxa <- as.data.frame(matrix(NA,nrow = nrow(pollen135), ncol = 63))
```

```
 # set column names to default non-NULL value
 for (t1 in 1:length(pollen35taxa))
    colnames(pollen35taxa) [t1] <- c("taxa")

 for (t2 in 1:length(pollen64taxa))
    colnames(pollen64taxa) [t2] <- c("taxa")


if (PSUM == 134)
{
  # to create 134 ps database simply remove the PinusSE column (col 105)
  colToRemove <- colnames(pollen135)[105]

  if (modern)
  {
    ps134mod <- ps134mod[, setdiff(colnames(ps134mod),c(colToRemove)) ]
    psModData <- ps134mod
  }else
  {
    ps134fos <- ps134fos[, setdiff(colnames(ps134fos),c(colToRemove)) ]
    psFosData <- ps134fos
  }

}else if (PSUM == 64) # now build taxa columns based on pollen sum 64
{

   # (1) ABIES is col 14, keep this
   pollen64taxa[,1] <- pollen135[,14]
   colnames(pollen64taxa)[1]<- colnames(pollen135)[14]

   # (2) new column ACER is the sum of col 15-20
   pollen64taxa[,2] <- rowSums(pollen135[,15:20])
   colnames(pollen64taxa)[2] <- c("ACER")

   # (3) new column ALNUSXC is the sum of col 21,24
   pollen64taxa[,3] <- rowSums(pollen135[,c(21,24)])
   colnames(pollen64taxa)[3] <- c("ALNUSXC")

   # (4) keep col 22,23
   pollen64taxa[,4] <- pollen135[,22]
   pollen64taxa[,5] <- pollen135[,23]
   colnames(pollen64taxa)[4]<- colnames(pollen135)[22]
   colnames(pollen64taxa)[5]<- colnames(pollen135)[23]

   # exclude col 25-28

   # (5) keep col 29,30
   pollen64taxa[,6] <- pollen135[,29]
   pollen64taxa[,7] <- pollen135[,30]
   colnames(pollen64taxa)[6]<- colnames(pollen135)[29]
   colnames(pollen64taxa)[7]<- colnames(pollen135)[30]

   # (6) keep col 31,32

   # (7) ARTEMISIA is col 33, keep this
   pollen64taxa[,8] <- pollen135[,33]
   colnames(pollen64taxa)[8]<- colnames(pollen135)[33]

   # (8) new col ASTERACEAE is the sum of col 34, 35, 72, 148
   pollen64taxa[,9] <- rowSums(pollen135[,c(34,35,72,148)])
   colnames(pollen64taxa)[9] <- c("ASTERACEAE")

   # (9) BETULA is col 36 - keep this
   pollen35taxa[,10] <- pollen135[,36]
   colnames(pollen64taxa)[10]<- colnames(pollen135)[36]

   # exclude col 37,38

   # (10) keep col 39,40
   pollen64taxa[,11] <- pollen135[,39]
   pollen64taxa[,12] <- pollen135[,40]
   colnames(pollen64taxa)[11]<- colnames(pollen135)[39]
   colnames(pollen64taxa)[12]<- colnames(pollen135)[40]

   # exclude col 41,42
```

```
# (11) keep col 43-45
pollen35taxa[,13] <- pollen135[,43]
pollen35taxa[,14] <- pollen135[,44]
pollen35taxa[,15] <- pollen135[,45]
### STEVE FIXED
colnames(pollen64taxa)[13]<- colnames(pollen135)[43]
colnames(pollen64taxa)[14]<- colnames(pollen135)[44]
colnames(pollen64taxa)[15]<- colnames(pollen135)[45]

# exclude col 46

# (10) keep col 47,48
pollen64taxa[,16] <- pollen135[,47]
pollen64taxa[,17] <- pollen135[,48]
colnames(pollen64taxa)[16]<- colnames(pollen135)[47]
colnames(pollen64taxa)[17]<- colnames(pollen135)[48]

# exclude col 49

# (11) keep col 50,51
pollen64taxa[,18] <- pollen135[,50]
pollen64taxa[,19] <- pollen135[,51]
colnames(pollen64taxa)[18]<- colnames(pollen135)[50]
colnames(pollen64taxa)[19]<- colnames(pollen135)[51]

# exclude col 52

# (12) keep col 53-55
pollen64taxa[,20] <- pollen135[,53]
pollen64taxa[,21] <- pollen135[,54]
pollen64taxa[,22] <- pollen135[,55]
colnames(pollen64taxa)[20]<- colnames(pollen135)[53]
colnames(pollen64taxa)[21]<- colnames(pollen135)[54]
colnames(pollen64taxa)[22]<- colnames(pollen135)[55]

# exclude col 56

# (13) keep col 57
pollen64taxa[,23] <- pollen135[,57]
colnames(pollen64taxa)[23]<- colnames(pollen135)[57]

# exclude col 58

# (14) keep col 59
pollen64taxa[,24] <- pollen135[,59]
colnames(pollen64taxa)[24]<- colnames(pollen135)[59]

# exclude col 60

# (15) new col ERICACEAE is the sum of col 61:63
pollen64taxa[,25] <- rowSums(pollen135[,61:63])
colnames(pollen64taxa)[25] <- c("ERICACEAE")

# exclude col 64

# exclude col 67

# (16) keep col 68-71
pollen64taxa[,26] <- pollen135[,68]
pollen64taxa[,27] <- pollen135[,69]
pollen64taxa[,28] <- pollen135[,70]
pollen64taxa[,29] <- pollen135[,71]
colnames(pollen64taxa)[26]<- colnames(pollen135)[68]
colnames(pollen64taxa)[27]<- colnames(pollen135)[69]
colnames(pollen64taxa)[28]<- colnames(pollen135)[70]
colnames(pollen64taxa)[29]<- colnames(pollen135)[71]

# (18)new col JUGLANS is the sum of col 73-75
pollen64taxa[,30] <- rowSums(pollen135[,73:75])
colnames(pollen64taxa)[30] <- c("JUGLANS")

# exclude col 76,77

# (19) keep col 78,79
pollen64taxa[,31] <- pollen135[,78]
pollen64taxa[,32] <- pollen135[,79]
```

```
colnames(pollen64taxa)[31]<- colnames(pollen135)[78]
colnames(pollen64taxa)[32]<- colnames(pollen135)[79]

# exclude col 80

# (20) keep col 81
pollen64taxa[,33] <- pollen135[,81]
colnames(pollen64taxa)[33]<- colnames(pollen135)[81]

# exclude col 82

# (21)new col LYCOPODIUM is the sum of col 83-87
pollen64taxa[,34] <- rowSums(pollen135[,83:87])
colnames(pollen64taxa)[34] <- c("LYCOPODIUM")

# exclude col 88

# (22)new col MINOR FORBS is the sum of col 28, 65, 66, 89
pollen64taxa[,35] <- rowSums(pollen135[,c(28,65,66,89)])
colnames(pollen64taxa)[35] <- c("MINOR FORBS")

# exclude col 90

# (23) keep col 91,92
pollen64taxa[,36] <- pollen135[,91]
pollen64taxa[,37] <- pollen135[,92]
colnames(pollen64taxa)[36]<- colnames(pollen135)[91]
colnames(pollen64taxa)[37]<- colnames(pollen135)[92]

# exclude col 93,94

# (24) keep col 95-97
pollen64taxa[,38] <- pollen135[,95]
pollen64taxa[,39] <- pollen135[,96]
pollen64taxa[,40] <- pollen135[,97]
colnames(pollen64taxa)[38]<- colnames(pollen135)[95]
colnames(pollen64taxa)[39]<- colnames(pollen135)[96]
colnames(pollen64taxa)[40]<- colnames(pollen135)[97]

# exclude col 98

# (25)new col PICEA is the sum of col 99-101
pollen64taxa[,41] <- rowSums(pollen135[,99:101])
colnames(pollen64taxa)[41] <- c("PICEA")

# (26) keep col 102-108
pollen64taxa[,42] <- pollen135[,102]
pollen64taxa[,43] <- pollen135[,103]
pollen64taxa[,44] <- pollen135[,104]
pollen64taxa[,45] <- pollen135[,105]
pollen64taxa[,46] <- pollen135[,106]
pollen64taxa[,47] <- pollen135[,107]
pollen64taxa[,48] <- pollen135[,108]
colnames(pollen64taxa)[42]<- colnames(pollen135)[102]
colnames(pollen64taxa)[43]<- colnames(pollen135)[103]
colnames(pollen64taxa)[44]<- colnames(pollen135)[104]
colnames(pollen64taxa)[45]<- colnames(pollen135)[105]
colnames(pollen64taxa)[46]<- colnames(pollen135)[106]
colnames(pollen64taxa)[47]<- colnames(pollen135)[107]
colnames(pollen64taxa)[48]<- colnames(pollen135)[108]

# exclude col 109-114

# (27) keep col 115
pollen64taxa[,49] <- pollen135[,115]
colnames(pollen64taxa)[49]<- colnames(pollen135)[115]

# exclude col 116

# (28) keep col 117-119
pollen64taxa[,50] <- pollen135[,117]
pollen64taxa[,51] <- pollen135[,118]
pollen64taxa[,52] <- pollen135[,119]
colnames(pollen64taxa)[50]<- colnames(pollen135)[117]
colnames(pollen64taxa)[51]<- colnames(pollen135)[118]
colnames(pollen64taxa)[52]<- colnames(pollen135)[119]
```

```
  # exclude col 120,121

  # (29) keep col 122
  pollen64taxa[,53] <- pollen135[,122]
  colnames(pollen64taxa)[53]<- colnames(pollen135)[122]

  # exclude col 123-126

  # (30) keep col 127
  pollen64taxa[,54] <- pollen135[,127]
  colnames(pollen64taxa)[54]<- colnames(pollen135)[127]

  # exclude col 128

  # (31) keep col 129
  pollen64taxa[,55] <- pollen135[,129]
  colnames(pollen64taxa)[55]<- colnames(pollen135)[129]

  # (32)new col SAXIFRAGA is the sum of col 130-133
  pollen64taxa[,56] <- rowSums(pollen135[,130:133])
  colnames(pollen64taxa)[56] <- c("SAXIFRAGA")

  # exclude col 134-137

  # (33) keep col 138,139
  pollen64taxa[,57] <- pollen135[,138]
  pollen64taxa[,58] <- pollen135[,139]
  colnames(pollen64taxa)[57]<- colnames(pollen135)[138]
  colnames(pollen64taxa)[58]<- colnames(pollen135)[139]

  # exclude col 140,141

  # (34) keep col 142-146
  pollen64taxa[,59] <- pollen135[,142]
  pollen64taxa[,60] <- pollen135[,143]
  pollen64taxa[,61] <- pollen135[,144]
  pollen64taxa[,62] <- pollen135[,145]
  pollen64taxa[,63] <- pollen135[,146]
  colnames(pollen64taxa)[59]<- colnames(pollen135)[142]
  colnames(pollen64taxa)[60]<- colnames(pollen135)[143]
  colnames(pollen64taxa)[61]<- colnames(pollen135)[144]
  colnames(pollen64taxa)[62]<- colnames(pollen135)[145]
  colnames(pollen64taxa)[63]<- colnames(pollen135)[146]

  # exclude col 147

  if (modern)
    # for modern 64, cbind the following (ps64modA, cols from steps (1)-(34), ps64modB)
    psModData <- cbind(ps64modA,pollen64taxa,ps64modB)
  else
    # for fossil 64, cbind the following (ps64fosA, cols from steps (1)-(34))
    psFosData <- cbind(ps64fos,pollen64taxa)

}else # PSUM == 35 now build taxa columns based on pollen sum 35
{

  # (1) ABIES is col 14, keep this
  pollen35taxa[,1] <- pollen135[,14]
  colnames(pollen35taxa)[1]<- colnames(pollen135)[14]

  # (2) new column ACER is the sum of col 15-20
  pollen35taxa[,2] <- rowSums(pollen135[,15:20])
  colnames(pollen35taxa)[2] <- c("ACER")

  # (3) new column ALNUS is the sum of col 21-24
  pollen35taxa[,3] <- rowSums(pollen135[,21:24])
  colnames(pollen35taxa)[3] <- c("ALNUS")

  # exclude col 25-32

  # (4) ARTEMISIA is col 33, keep this
  pollen35taxa[,4] <- pollen135[,33]
  colnames(pollen35taxa)[4]<- colnames(pollen135)[33]

  # (5) new col ASTERACEAE is the sum of col 34, 35, 72, 148
```

```
pollen35taxa[,5] <- rowSums(pollen135[,c(34,35,72,148)])
colnames(pollen35taxa)[5] <- c("ASTERACEAE")

# (6) BETULA is col 36 - keep this
pollen35taxa[,6] <- pollen135[,36]
colnames(pollen35taxa)[6]<- colnames(pollen135)[36]
# exclude col 37-39

# (7) keep col 40
pollen35taxa[,7] <- pollen135[,40]
colnames(pollen35taxa)[7]<- colnames(pollen135)[40]
# exclude col 41,42

# (8) keep col 43
pollen35taxa[,8] <- pollen135[,43]
colnames(pollen35taxa)[8]<- colnames(pollen135)[43]
# exclude col 44

# (9)keep col 45
pollen35taxa[,9] <- pollen135[,45]
colnames(pollen35taxa)[9]<- colnames(pollen135)[45]
# exclude col 46-49

# (10)keep col 50
pollen35taxa[,10] <- pollen135[,50]
colnames(pollen35taxa)[10]<- colnames(pollen135)[50]
# exclude col 51,52

# (11)keep col 53,54,55
pollen35taxa[,11] <- pollen135[,53]
pollen35taxa[,12] <- pollen135[,54]
pollen35taxa[,13] <- pollen135[,55]
colnames(pollen35taxa)[11]<- colnames(pollen135)[53]
colnames(pollen35taxa)[12]<- colnames(pollen135)[54]
colnames(pollen35taxa)[13]<- colnames(pollen135)[55]
# exclude col 56-58

# (12)keep col 59
pollen35taxa[,14] <- pollen135[,59]
colnames(pollen35taxa)[14]<- colnames(pollen135)[59]
# exclude col 60-67

# (13)keep col 68
pollen35taxa[,15] <- pollen135[,68]
colnames(pollen35taxa)[15]<- colnames(pollen135)[68]
# (14)new col FRAXINUS is the sum of col 69-71
pollen35taxa[,16] <- rowSums(pollen135[,69:71])
colnames(pollen35taxa)[16] <- c("FRAXINUS")

# (15)new col JUGLANS is the sum of col 73-75
pollen35taxa[,17] <- rowSums(pollen135[,73:75])
colnames(pollen35taxa)[17] <- c("JUGLANS")

# exclude col 76,77

# (16)keep col 78
pollen35taxa[,18] <- pollen135[,78]
colnames(pollen35taxa)[18]<- colnames(pollen135)[78]
# exclude col 79,80

# (17)keep col 81
pollen35taxa[,19] <- pollen135[,81]
colnames(pollen35taxa)[19]<- colnames(pollen135)[81]

# exclude col 83-94

# (18)keep col 95
pollen35taxa[,20] <- pollen135[,82]
colnames(pollen35taxa)[20]<- colnames(pollen135)[82]

# exclude col 96-98

# (19)new col PICEA is the sum of col 99-101
pollen35taxa[,21] <- rowSums(pollen135[,99:101])
colnames(pollen35taxa)[21] <- c("PICEA")
```

```
# (20)new col PINUS is the sum of col 102-104
pollen35taxa[,22] <- rowSums(pollen135[,102:104])
colnames(pollen35taxa)[22] <- c("PINUS")

# (21)keep col 105
pollen35taxa[,23] <- pollen135[,105]
colnames(pollen35taxa)[23]<- colnames(pollen135)[105]

# exclude col 106

# (22)keep col 107,108
pollen35taxa[,24] <- pollen135[,107]
pollen35taxa[,25] <- pollen135[,108]
colnames(pollen35taxa)[24]<- colnames(pollen135)[107]
colnames(pollen35taxa)[25]<- colnames(pollen135)[108]

# exclude col 109-114

# (23)keep col 115
pollen35taxa[,26] <- pollen135[,115]
colnames(pollen35taxa)[26]<- colnames(pollen135)[115]

# exclude col 116

# (24)keep col 117
pollen35taxa[,27] <- pollen135[,117]
colnames(pollen35taxa)[27]<- colnames(pollen135)[117]

# exclude col 118

# (25)keep col 119
pollen35taxa[,28] <- pollen135[,119]
colnames(pollen35taxa)[28]<- colnames(pollen135)[119]

# exclude col 120-126

# (26)keep col 127
pollen35taxa[,29] <- pollen135[,127]
colnames(pollen35taxa)[29]<- colnames(pollen135)[127]

# exclude col 128

# (27)keep col 129
pollen35taxa[,30] <- pollen135[,129]
colnames(pollen35taxa)[30]<- colnames(pollen135)[129]

# (28)new col SAXIFRAGA is the sum of col 130-133
pollen35taxa[,31] <- rowSums(pollen135[,130:133])
colnames(pollen35taxa)[31] <- c("SAXIFRAGA")

# exclude col 134-140

# (29)keep col 141-143
pollen35taxa[,32] <- pollen135[,141]
pollen35taxa[,33] <- pollen135[,142]
pollen35taxa[,34] <- pollen135[,143]
colnames(pollen35taxa)[32]<- colnames(pollen135)[141]
colnames(pollen35taxa)[33]<- colnames(pollen135)[142]
colnames(pollen35taxa)[34]<- colnames(pollen135)[143]

# exclude col 144

# (30)keep col 145,146
pollen35taxa[,35] <- pollen135[,145]
pollen35taxa[,36] <- pollen135[,146]

colnames(pollen35taxa)[35]<- colnames(pollen135)[145]
colnames(pollen35taxa)[36]<- colnames(pollen135)[146]

# exclude col 147

if (modern)
   # for modern 35, cbind the following (ps35modA, cols from steps (1)-(30), ps35modB)
   psModData <- cbind(ps35modA,pollen35taxa,ps35modB)
else
   # for fossil 35, cbind the following (ps35fosA, cols from steps (1)-(30))
```

```
      psFosData <- cbind(ps35fos,pollen35taxa)

  }

  if (modern) # now write the files to disk
  {
     modDBstr <- paste("ps",PSUM,"perc.modern")
     modDBstr2 = gsub(" ","", modDBstr ,fixed=TRUE)
     tryCatch({write.csv(psModData,modDBstr2,row.names=FALSE)}, error = function(e) {message(paste("File",modDBstr2,"cannot be written
          to disk, please close the file!",sep=" "))}, warning = function(w){})

  }else
  {
     fosDBstr <- paste("ps",PSUM,"perc.fossil")
     fosDBstr2 = gsub(" ","", fosDBstr ,fixed=TRUE)
     tryCatch({write.csv(psFosData,fosDBstr2,row.names=FALSE)}, error = function(e) {message(paste("File",fosDBstr2,"cannot be written
          to disk, please close the file!",sep=" "))}, warning = function(w){})

  }

  #if (modern)
  # return (psModData)
  #else
  # return (psFosData)

}calcAnomalies <- function(reconSPjoin, spFiltFlag)
{
  spFilterYN = 0
  nInd = 1 # start at first cal mean
  nSites = 0 # number of sites count
  calcValue = 0
  stdDevFos = 0
  stdDevMod = 0

  reconSPjoinFinal = NULL
  reconSPjoinFinal2 = NULL

  if (!spFiltFlag) # not a filtered anomamly file
  {
    # do anomaly calculation (reconstruction minus modern for reconSPjoin)
    reconSPjoinAnCalc <- reconSPjoin[,5] - reconSPjoin[,8]

    # add this column to matrix passed into this function
    reconSPjoinFinal <- cbind(reconSPjoin,reconSPjoinAnCalc)

    # write the output to a csv file
    tryCatch({write.csv(reconSPjoinFinal,file="SpatialJoinReconModWithAnCalc.csv",row.names=FALSE)}, error = function(e){message("
          SpatialJoinReconModWithAnCalc.csv cannot be written to working directory, please close the file!")}, warning = function(w){})

    while (is.na(spFilterYN) | spFilterYN < 1 | spFilterYN > 2)
       spFilterYN <- as.numeric(readline("Would you like to spatially filter this reconstruction output? <1 for YES, 2 for NO> "))

    if (spFilterYN == 1)
    {
       reconSPjoinFinalList <- spatialFiltering(reconSPjoinFinal)
       reconSPjoinFinal2 <- reconSPjoinFinalList[[1]]
       reconSPjoinFinal <- reconSPjoinFinal2
    }

  }else # if already filtered no need to do the above steps
  {

    reconSPjoinFinal <- reconSPjoin

  }

  # sort by CalYrBP (column 4 in reconSPjoin)
  reconSPjoinFinal <- reconSPjoinFinal[order(reconSPjoinFinal[,4]),]

  # get a list of unique cal yr values
  calYear <- unique(reconSPjoinFinal[,4])

  # aggregate avg anomaly values by cal mean
  anByCalYrCalc <- aggregate(reconSPjoinFinal[,10], list(reconSPjoinFinal[,4]), mean)

  # aggregate so that we have 4 cols: cal mean, recon value, mod value, and anomaly calc
```

```
  anByCalYrAg <- cbind(reconSPjoinFinal[,4],reconSPjoinFinal[,5],reconSPjoinFinal[,8],reconSPjoinFinal[,10])

  lastRow = nrow(anByCalYrCalc) # number of cal yr intervals
  reconAnomStdErr <- numeric(lastRow) # create new vector for std err values
  reconAnomFinal <- cbind(anByCalYrCalc,reconAnomStdErr)

  while (nInd <= length(calYear))
  {
    # reset these variables
    subsetCalYr <- NULL
    nSites = 0
    stdDevFos = 0
    stdDevMod = 0
    calcValue = 0

    # get subset by cal year interval
    subsetCalYr <- subset(anByCalYrAg, anByCalYrAg[,1] == calYear[nInd])

    nSites = nrow(subsetCalYr) # number of sites is number of rows

    # now need to get the standard dev of recon and modern values
    stdDevFos <- sd(subsetCalYr[,2])
    stdDevMod <- sd(subsetCalYr[,3])

    # p (corr coefficient) is 0.9, n (sample size) is number of sites at each interval (nSites)
    # SQRT[ STDEV(Fos) squared + STDEV(Mod) squared - 2 * p * STDEV(Fos) * STDEV(Mod) ] / SQRT[n]
    calcValue <- sqrt(stdDevFos * stdDevFos + stdDevMod * stdDevMod - 2 * 0.9 * stdDevFos * stdDevMod) / sqrt(nSites)
    reconAnomFinal[nInd,3] <- calcValue
    nInd = nInd + 1
  }

  # fix up the col names
  colnames(reconAnomFinal)[1] <- "CalMean"
  colnames(reconAnomFinal)[2] <- "AvgAnom"
  colnames(reconAnomFinal)[3] <- "StdError"

  message("Writing file avgAnomalyByCalYrBP.csv to working directory...")
  tryCatch({write.csv(reconAnomFinal,file="avgAnomalyByCalYrBP.csv",row.names=FALSE)}, error = function(e) {message("Cannot write
        avgAnomalyByCalYrBP.csv to disk, please close the file!")}, warning = function(w){})

  anomList <- list(reconAnomFinal, reconSPjoinFinal)
  return (anomList)
}calcAvgAnomaliesByEcoregion <- function(params=ECOREGIONS, reconSPjoin, tuned, selClimVar, oldYrBP, youngYrBP, intYear)
{
  tunedAnomByCalYrERList <- NULL
  tunedAnomByCalYrERfinal <- NULL
  anomAgSubsetAllEcoregions <- NULL
  avgAnomAllEcoregions <- NULL
  reconAnomFinal <- NULL
  reconAnomFinalER <- NULL
  anByCalYrCalcSel <- NULL
  anByCalYrCalcSub <- NULL
  reconSubsetSel <- NULL
  countsSubsetSel <- NULL
  anByCalYrAg2 <- NULL

  cInd = 1 # start at first cal mean
  eInd = 1 # start at first ecoregion

  reg = 0 # ecoregion numeric value
  nSites = 0 # number of sites count
  calcValue = 0
  stdDevFos = 0
  stdDevMod = 0

  # only proceed if each calendar year bp interval has an avg anomaly value by ecoregion for all ecoregions
  tryCatch({

    # Make sure -99 and blank (no-data) ecoregions are NOT included (exclude these rows)
    reconSPjoinFinal <- reconSPjoin[reconSPjoin[,9] != -99 & reconSPjoin[,9] != "" ,]

    # aggregate avg anomaly values by cal mean
    anByCalYrCalc <- aggregate(reconSPjoinFinal[,10], list(reconSPjoinFinal[,4],reconSPjoinFinal[,9]), mean, na.rm=TRUE)

    # get a list of unique cal yr values
    calYear <- unique(reconSPjoinFinal[,4])
```

```
# get a list of unique ecoregions
ecoRegions <- unique(reconSPjoinFinal[,9])

# we have 5 cols: cal mean, recon value, mod value, anomaly calc and ecoregion
anByCalYrAg <- cbind(reconSPjoinFinal[,4],reconSPjoinFinal[,5],reconSPjoinFinal[,8],reconSPjoinFinal[,10],reconSPjoinFinal[,9])

# need to loop through ecoregions
while (eInd <= length(ecoRegions))
{
  # reset these variables
  anByCalYrCalcSel <- NULL
  anByCalYrCalcSub <- NULL
  reconSubsetSel <- NULL
  countsSubsetSel <- NULL
  reconAnomFinal <- NULL
  nSites = 0
  calcValue = 0
  stdDevFos = 0
  stdDevMod = 0
  cInd = 1

  # ecoregions were assigned numeric values above so correct this
  # e.g. Forest-tundra is index 1 but was assigned a value of 9

  if (eInd == 1) # Forest-Tundra
     reg = 9
  else if (eInd == 2) # Conifer/Hardwood
     reg = 6
  else if (eInd == 3) # Boreal Forest
     reg = 4
  else if (eInd == 4) # Deciduous Forest
     reg = 7
  else if (eInd == 5) # Prairies
     reg = 12
  else if (eInd == 6) # Arctic
     reg = 3
  else if (eInd == 7) # Coastal
     reg = 5
  else if (eInd == 8) # Mountain vegetation
     reg = 11
  else if (eInd == 9) # Desert
     reg = 8
  # eInd == 10 for Mediterranian but this region is excluded

  # get a subset for selected ecoregion
  anByCalYrCalcSel <- subset(anByCalYrCalc, anByCalYrCalc[,2] == ECOREGIONS[eInd,2])
  anByCalYrCalcSub <- anByCalYrCalcSel[c(1,3)]

  lastRow = nrow(anByCalYrCalcSel) # number of cal yr intervals
  reconAnomStdErr <- numeric(lastRow) # create new vector for std err values
  reconAnomFinal <- cbind(anByCalYrCalcSub,reconAnomStdErr)

  # get a subset for selected ecoregion
  reconSubsetSel <- subset(anByCalYrAg, anByCalYrAg[,5] == reg)

  while (cInd <= length(calYear))
  {
    # reset these variables
    subsetCalYr <- NULL
    nSites = 0
    stdDevFos = 0
    stdDevMod = 0
    calcValue = 0

    # get subset by calendar year interval
    subsetCalYr <- subset(reconSubsetSel, reconSubsetSel[,1] == calYear[cInd])
    nSites = nrow(subsetCalYr) # number of sites is number of rows

    message(paste("There are",nSites,"sites in ecoregion",ECOREGIONS[eInd,2],"at",calYear[cInd],"bp",sep=" "))

    # need to get the standard dev of recon and modern values
    stdDevFos <- sd(subsetCalYr[,2])
    stdDevMod <- sd(subsetCalYr[,3])

    # p (corr coefficient) is 0.9, n (sample size) is number of sites at each interval (nSites)
```

```
      # SQRT[ STDEV(Fos) squared + STDEV(Mod) squared - 2 * p * STDEV(Fos) * STDEV(Mod) ] / SQRT[n]

      calcValue <- sqrt(stdDevFos * stdDevFos + stdDevMod * stdDevMod - 2 * 0.9 * stdDevFos * stdDevMod) / sqrt(nSites)
      reconAnomFinal[cInd,3] <- calcValue
      cInd = cInd + 1
    }
    # make sure to not cbind a null matrix to a null matrix
    if (eInd == 1)
       reconAnomFinalER <- reconAnomFinal
    else # eInd > 1
       reconAnomFinalER <- cbind(reconAnomFinalER,reconAnomFinal)
    eInd = eInd + 1
  }

  anomAgSubsetAllEcoregions <- reconAnomFinalER

  message("")
  message("NOTE: Mediterranian and no data ecoregions are excluded.")
  message("")

  # fix up all the col names (BF = boreal forest, etc)
  # "Forest-Tundra","Conifer/Hardwood", "Boreal Forest", "Deciduous Forest",
  # "Prairies","Arctic","Coastal","Mountain vegetation","Desert","Mediterranian"

  colnames(anomAgSubsetAllEcoregions)[1] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[3] <- "StdErr_FT"
  colnames(anomAgSubsetAllEcoregions)[2] <- "AvgAn_FT"
  colnames(anomAgSubsetAllEcoregions)[4] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[6] <- "StdErr_CH"
  colnames(anomAgSubsetAllEcoregions)[5] <- "AvgAn_CH"
  colnames(anomAgSubsetAllEcoregions)[7] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[9] <- "StdErr_BF"
  colnames(anomAgSubsetAllEcoregions)[8] <- "AvgAn_BF"
  colnames(anomAgSubsetAllEcoregions)[10] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[12] <- "StdErr_DF"
  colnames(anomAgSubsetAllEcoregions)[11] <- "AvgAn_DF"
  colnames(anomAgSubsetAllEcoregions)[13] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[15] <- "StdErr_P"
  colnames(anomAgSubsetAllEcoregions)[14] <- "AvgAn_P"
  colnames(anomAgSubsetAllEcoregions)[16] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[18] <- "StdErr_A"
  colnames(anomAgSubsetAllEcoregions)[17] <- "AvgAn_A"
  colnames(anomAgSubsetAllEcoregions)[19] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[21] <- "StdErr_C"
  colnames(anomAgSubsetAllEcoregions)[20] <- "AvgAn_C"
  colnames(anomAgSubsetAllEcoregions)[22] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[24] <- "StdErr_MV"
  colnames(anomAgSubsetAllEcoregions)[23] <- "AvgAn_MV"
  colnames(anomAgSubsetAllEcoregions)[25] <- "CalMean"
  colnames(anomAgSubsetAllEcoregions)[27] <- "StdErr_D"
  colnames(anomAgSubsetAllEcoregions)[26] <- "AvgAn_D"

  # remove duplicate CalMean columns
  avgAnomAllEcoregions <- anomAgSubsetAllEcoregions[,c(1,2,3,5,6,8,9,11,12,14,15,17,18,20,21,23,24,26,27)]

  tryCatch({write.csv(avgAnomAllEcoregions,file="anomByCalYrERfinal.csv",row.names=FALSE)}, error = function(e){message("
        anomByCalYrERfinal.csv cannot be written to working directory, please close the file!")}, warning = function(w){})

  if (tuned)
  {
     tunedAnomByCalYrERList <- tuneYearAD(avgAnomAllEcoregions, intYear, oldYrBP, youngYrBP)
     tunedAnomByCalYrERfinal <- tunedAnomByCalYrERList[[1]] # this will have the tuned dates
     startYr <- tunedAnomByCalYrERList[[2]] # e.g 1950 AD
     endYr <- tunedAnomByCalYrERList[[3]] # e.g. 350 AD
     tryCatch({write.csv(tunedAnomByCalYrERfinal,file="tunedAnomByAdYrERfinal.csv",row.names=FALSE)}, error = function(e){message("
           tunedAnomByAdYrERfinal.csv cannot be written to working directory, please close the file!")}, warning = function(w){})
     plotAvgAnomaliesByEcoregion(params=ECOREGIONS,tunedAnomByCalYrERfinal,selClimVar,endYr,startYr,TRUE)
  }else # not tuned
     plotAvgAnomaliesByEcoregion(params=ECOREGIONS,avgAnomAllEcoregions,selClimVar,oldYrBP,youngYrBP,FALSE)

 }, error = function(e){message("Average anomaly by ecoregion is not valid for this reconstruction subset due to one or more
       ecoregions having no sites for one or more cal year bp intervals!")}, warning = function(w){})

}calcSCDbyEcoregion <- function(reconFile, analogsNum, analogTop, analogBottom)
{
   # these would be done by eco-region, SCD data is in recon file
```

```
# also take into account if analogs were averaged (e.g. top 5 analogs)
# then average the top 5 SCD columns, also output to a csv file

# figure out which analog was selected for the reconstruction
# if (analogTop == analogBottom) get the appropriate SCD column from reconFile

# else average the SCD columns (from analogTop to analogBottom)
# to create a new column if top n analogs were averaged (e.g. top 5 analogs)
# aggregate this column by eco-region
# output a csv file with multiple columns

# Make sure -99 and blank (no-data) ecoregions are not included (exclude these rows)

scdCutoffVal = -99 # default value

strMatch = paste("ECOREGION",analogTop)
strMatch = gsub(" ","", strMatch ,fixed=TRUE)
matchCol = which(colnames(reconFile) == strMatch)

if (length(matchCol) > 0)
{
  reconFile <- reconFile[reconFile[,matchCol] != -99 & reconFile[,matchCol] != "" ,]
  reconSelAnalogCalYRcol <- reconFile[,7] # get subset of data
  reconSelAnalogSitecol <- reconFile[,1]
  selAnalogEcoregionCol <- reconFile[,matchCol]

  selSCDcol = 7 + analogTop # get the SCD col of the selected analog
  reconSelAnalogSCDcol <- reconFile[,selSCDcol]

  # get the col index of the Ecoregion col for chosen analog
  scdMatch = paste("X",analogTop)
  scdMatch2 = gsub(" ","", scdMatch ,fixed=TRUE)
  matchCol = which(colnames(reconFile) == scdMatch2)
  selAnalogEcoregionSCDcol <- reconFile[,matchCol]

 if (analogTop != analogBottom)
 {

    # follow same format as above except SCD column will need to be the average of the range of analogs

    lastSCDcol = selSCDcol + analogsNum - 1
    selSCDecoregionCol <- reconFile[,selSCDcol:lastSCDcol]

    analogsNum = analogBottom - analogTop + 1

    rowSumsSCDAvgSum = rowSums(selSCDecoregionCol)
    reconSCDFinal = rowSumsSCDAvgSum / analogsNum

    #fix col header for avg analog SCD col
    anAvgColStr <- paste("SCD",analogBottom,"to",analogTop)
    anAvgColStr2 = gsub(" ","", anAvgColStr ,fixed=TRUE)
    selAnalogEcoregionSCDcol = reconSCDFinal
 }

 # aggregate by ecoregion & create SCD plot for each eco-region - also need Cal Yr BP
 SCDByEcoregion <- aggregate(selAnalogEcoregionSCDcol, list(reconSelAnalogCalYRcol,selAnalogEcoregionCol), mean, na.rm=TRUE)


 # order by ecoregion then reshape data
 SCDByEcoregionSorted <- SCDByEcoregion[order(SCDByEcoregion[,1]),]


 # fix up the column headers here
 colnames(SCDByEcoregionSorted)[1] <- "CalYrBP"
 colnames(SCDByEcoregionSorted)[2] <- "Ecoregion"
 colnames(SCDByEcoregionSorted)[3] <- "SCD_val"


 # need to reshape the data to wide format to make plotting easier
 SCDByEcoregionFinal <- reshape(SCDByEcoregionSorted,v.names = "SCD_val", idvar = "CalYrBP", timevar = "Ecoregion", direction = "
       wide")

 colnames(SCDByEcoregionFinal)[1] <- "CalYrBP"

 # fix SCD values
 SCDByEcoregionFinal = SCDByEcoregionFinal / 100
```

```
    # except cal yr BP values
    SCDByEcoregionFinal[,1] = SCDByEcoregionFinal[,1] * 100

    # finally write csv and plot it
    message("Writing file SCDbyEcoregionFinal.csv to working directory...")
    tryCatch({write.csv(SCDByEcoregionFinal,"SCDbyEcoregionFinal.csv",row.names=FALSE)}, error = function(e){message("
          SCDbyEcoregionFinal.csv cannot be written to working directory, please close the file!")}, warning = function(w){})
    newList <- list(SCDByEcoregionFinal, reconSelAnalogSCDcol, reconSelAnalogCalYRcol)

    return(newList)
  }
  else
  {
    message("Reconstruction file does not have an ecoregion column therefore cannot calculate SCD by ecoregion!")
    return(NULL)
  }
}calcSiteResolution <- function( reconFile,oldestYr )
{
   require(plyr)
   plotSamp = 0

   # calculate the resolution of pollen sites
   # only get a subset based on oldest cal Yr BP of reconstruction
   # reconFile <- reconFile[reconFile[,7] <= oldestCalYr, ]

   # otherwise resolution of sites is calculated over all dates in the database

    minAges <- aggregate(reconFile[,7], list(reconFile[,1]),min) # get max age
    maxAges <- aggregate(reconFile[,7], list(reconFile[,1]),max) # get min age

    siteIDs <- length(unique(reconFile[,1])) # count unique pollen sites
    #message(paste("There are ", siteIDs, " fossil pollen sites in the database."))

    siteIDlist <- unique(reconFile[,1]) # get unique list of SiteIDs

    # here is where we get the count of samples for each site
    counts <- ddply(reconFile, .(reconFile$SITEID), nrow)
    names(counts) <- c("y", "Freq")

    resOutput<- cbind(maxAges,minAges,counts)

    # remove extra columns
    resOutput[,3] <- NULL
    resOutput[,5] <- NULL

    # fix up the column headers
    colnames(resOutput)[1] <- "SiteID"
    colnames(resOutput)[2] <- "AgeMax"
    colnames(resOutput)[3] <- "AgeMin"
    colnames(resOutput)[4] <- "SiteID" # duplicate

    # for each site in fossil DB get (ageMax - ageMin)
    resOutput2 <- cbind(resOutput, (resOutput$AgeMax - resOutput$AgeMin))

    resOutput2[,4] <- NULL # remove extra column
    colnames(resOutput2)[4] <- "AgeDiff"

    # for each site in fossil DB get (ageMax - ageMin / totSamp)
    resOutput3 <- cbind(resOutput2, (resOutput2$AgeDiff / counts), counts)

    resOutput3[,5] <- NULL # remove extra column
    resOutput3[,6] <- NULL # remove extra column

    # fix up a couple of column headers

    colnames(resOutput3)[5] <- "SiteRes"
    colnames(resOutput3)[6] <- "TotSamp"

    # write the final output to disk
    message("Writing fossil pollen site resolution report to csv file....")
    tryCatch({write.csv(resOutput3,"fossilPollenSiteResolution.csv",row.names=FALSE)
    }, error = function(e){message("fossilPollenSiteResolution.csv cannot be written to disk, please close the file!")}, warning =
          function(w){})
```

```
  siteResCol <- numeric(nrow(reconFile))
  reconFile <- cbind(reconFile,siteResCol)

  rowMatchID = 0
  rowIndex = 1
  lastRow = nrow(reconFile)
  lastCol = ncol(reconFile)

  while(rowIndex <= lastRow) # we need to write the site res values for every site in recon file
  {
     rowMatchID = which(resOutput3[,1] == reconFile[rowIndex,1])
     reconFile[rowIndex,lastCol] = resOutput3[rowMatchID,5]
     rowIndex = rowIndex + 1
  }

  # ask user if they they want to plot samples through time?
  while (is.na(plotSamp) | plotSamp < 1 | plotSamp > 2)
  {
     plotSamp=as.numeric(readline("Would you like to plot samples through time? <1 for YES, 2 for NO> "))
  }
  if (plotSamp == 1)
  {
     reconDataFile <- reconFile

     # first need to sort cal Yr field
     reconDataFile <- reconDataFile[order(reconDataFile[,7]),]

     # each sample ID gets a count of 1
     reconDataFile[,5] <- 1

     # only get a subset based on oldest cal Yr BP of reconstruction
     reconSubset <- reconDataFile[reconDataFile[,7] <= oldestYr, ]

     # should be able to plot samples vs CalYrBP

     tryCatch(
     {
        pdf("samplesThroughTimePlot.pdf")
        plot(reconSubset[,5],reconSubset[,7],pch=21,cex=0.1, bg="blue", ylab="Cal yr BP", xlab="", xaxt = "n")
        dev.off()
     }, error = function(e){message("samplesThroughTimePlot.pdf cannot be written to disk, please close the file!")}, warning =
           function(w){})

  }

  return (reconFile)
}checkForPS35_54_135files <- function(modernPollDB,fossilPollDB)
{
  # build 35, 64 and 134 pollen sums - check if these files are already in directory
  fileList <- list.files()

  # search for the ps35 files in the vector that gets returned

  found35mod <- "ps35perc.modern" %in% fileList
  if(!found35mod)
  {
     message("Building modern pollen sum 35 database.....")
     build_35_64_134_pollen_db(modernPollDB, 35, modern=TRUE)
  }else
  {
     message("Modern pollen sum 35 database already created!")
  }

  found35fos <- "ps35perc.fossil" %in% fileList
  if(!found35fos)
  {
     message("Building fossil pollen sum 35 database.....")
     build_35_64_134_pollen_db(fossilPollDB, 35, modern=FALSE)
  }else
  {
     message("Fossil pollen sum 35 database already created!")
  }

  # search for the ps64 files in the vector that gets returned
  found64mod <- "ps64perc.modern" %in% fileList
```

```
  if(!found64mod)
  {
     message("Building modern pollen sum 64 database.....")
     build_35_64_134_pollen_db(modernPollDB, 64, modern=TRUE)
  }else
  {
     message("Modern pollen sum 64 database already created!")
  }

  found64fos <- "ps64perc.fossil" %in% fileList
  if(!found64fos)
  {
     message("Building fossil pollen sum 35 database.....")
     build_35_64_134_pollen_db(fossilPollDB, 64, modern=FALSE)
  }else
  {
     message("Fossil pollen sum 64 database already created!")
  }


  # search for the ps134 files in the vector that gets returned
  found134mod <- "ps134perc.modern" %in% fileList
  if(!found134mod)
  {
     message("Building modern pollen sum 134 database.....")
     build_35_64_134_pollen_db(modernPollDB, 134, modern=TRUE)
  }else
  {
     message("Modern pollen sum 134 database already created!")
  }

  found134fos <- "ps134perc.fossil" %in% fileList
  if(!found134fos)
  {
     message("Building fossil pollen sum 134 database.....")
     build_35_64_134_pollen_db(fossilPollDB, 134, modern=FALSE)


  }else
  {
     message("Fossil pollen sum 134 database already created!")
  }
}DISTANCE_METHOD <- "spherical"
COUNTS <- FALSE

### Needed for pollen editor
MODERN_DIR <- "ModernPollen"
FOSSIL_DIR <- "FossilPollen"
CLIMATE_DIR <- "ClimateData"
RECONSTRUCTIONS <- "Reconstructions"
POLLEN_COLUMN_START <- 13
END_COLS <- 82
MAX_TAXA <- 135


ECOREGIONS <- data.frame(Id=1:10, ecoregionList = c("Forest-Tundra","Conifer/Hardwood", "Boreal Forest", "Deciduous Forest",
"Prairies","Arctic","Coastal","Mountain vegetation","Desert","Mediterranian"),stringsAsFactors = FALSE)


CLIMATE_VARIABLES <- data.frame(Id=1:70, climateVarList =c("tjan","tfeb","tmar","tapr","tmay","tjun","tjul","taug","tsep","toct", "tnov
      ",
                  "tdec","pjan","pfeb","pmar","papr","pmay","pjun","pjul","paug","psep","poct","pnov","pdec","sjan",
                  "sfeb","smar","sapr","smay","sjun","sjul","saug","ssep","soct","snov","sdec","tave","tmax","tmin",
                  "gdd0","gdd5","chill","mtco","mtwa","annp","pjanpann","pjulpann","mipt","aaetpt","apetpt","miptev","aaetptev","
                        apetptev","miptdc",
                  "aaetptdc","apetptdc","totsnopt","tmam","tjja","tson","tdjf","pmam",
                  "pjja","pson","pdjf","smam","sjja","sson","sdjf","CUSTOM"),

                  climateVarDesc = c("Mean monthly temperature for January in C",
                  "Mean monthly temperature for February in C",
                  "Mean monthly temperature for March in C",
                  "Mean monthly temperature for April in C",
                  "Mean monthly temperature for May in C",
                  "Mean monthly temperature for June in C",
                  "Mean monthly temperature for July in C",
                  "Mean monthly temperature for August in C",
```

```
                "Mean monthly temperature for September in C",
                "Mean monthly temperature for October in C",
                "Mean monthly temperature for November in C",
                "Mean monthly temperature for December in C",
                "Mean monthly precipitation for January in mm",
                "Mean monthly precipitation for February in mm",
                "Mean monthly precipitation for March in mm",
                "Mean monthly precipitation for April in mm",
                "Mean monthly precipitation for May in mm",
                "Mean monthly precipitation for June in mm",
                "Mean monthly precipitation for July in mm",
                "Mean monthly precipitation for August in mm",
                "Mean monthly precipitation for September in mm",
                "Mean monthly precipitation for October in mm",
                "Mean monthly precipitation for November in mm",
                "Mean monthly precipitation for December in mm",
                "Mean monthly percent possible sunshine for January",
                "Mean monthly percent possible sunshine for February",
                "Mean monthly percent possible sunshine for March",
                "Mean monthly percent possible sunshine for April",
                "Mean monthly percent possible sunshine for May",
                "Mean monthly percent possible sunshine for June",
                "Mean monthly percent possible sunshine for July",
                "Mean monthly percent possible sunshine for August",
                "Mean monthly percent possible sunshine for September",
                "Mean monthly percent possible sunshine for October",
                "Mean monthly percent possible sunshine for November",
                "Mean monthly percent possible sunshine for December",
                "Average annual temperature in C",
                "Absolute maximum temperature in C",
                "Absolute minimum temperature in C",
                "Growing-degree days (0 C base)",
                "Growing-degree days (5 C base)",
                "Chilling requirement (number of days when pseudo-daily temperature was below 5 C)",
                "Mean temperature (in C) of the coldest month of the year",
                "Mean temperature (in C) of the warmest month of the year",
                "Total annual precipitation (in mm)",
                "January/Annual precipitation ratio",
                "July/Annual precipitation ratio",
                "Priestley-Taylor (alpha) parameter (AE/PE)",
                "Actual evapotranspiration (AE)",
                "Potential evapotranspiration (PE)",
                "Alpha calculated for the evergreen pft assimilation period (days when the pseudo-daily temperature > -4C",
                "AE calculated for the deciduous pft assimilation period",
                "PE calculated for the deciduous pft assimilation period",
                "Annual Priestly-Taylor moisture index (alpha) for deciduous assimilation period (when pseudo-daily temperatures are
                      greater than 5C) [calculated using Cramer & Prentice approach]",
                "Annual actual evapotraspiration for deciduous assimilation period (when pseudo-daily temperatures are greater than 5
                      C) [calculated using Cramer & Prentice approach]",
                "Annual potential evapotraspiration for deciduous assimilation period (when pseudo-daily temperatures are greater
                      than 5C) [calculated using Cramer & Prentice approach]",
                "Total snow-water equivalent from bucket model (computed)",
                "Mean temperature (in C) for March, April, May",
                "Mean temperature (in C) for June, July, August",
                "Mean temperature (in C) for September, October, November",
                "Mean temperature (in C) for December, January, February",
                "Mean precipitation (in mm) for March, April, May",
                "Mean precipitation (in mm) for June, July, August",
                "Mean precipitation (in mm) for September, October, November",
                "Mean precipitation (in mm) for December, January, February",
                "Mean sunshine (%) for March, April, May",
                "Mean sunshine (%) for June, July, August",
                "Mean sunshine (%) for September, October, November",
                "Mean sunshine (%) for December, January, February","[NEW CLIMATE VARIABLE]"),stringsAsFactors = FALSE)

PS135_TAXA_LIST <- data.frame(Id=1:135, taxaList135 = c("Abies", "AcerNegundo",
  "AcerPensylvanicum",
  "AcerRubrum",
  "AcerSaccharinum",
  "AcerSaccharum",
  "AcerUndiff",
  "AlnusCrispa",
  "AlnusRubra",
  "AlnusRugosa",
  "AlnusUndiff",
  "AmbrosiaType",
```

```
"Amorpha",
"Anacardiaceae",
"Apiaceae",
"Aquifoliaceae",
"Arceuthobium",
"Arecaceae",
"Armeria",
"Artemisia",
"Asteraceae",
"AsteraceaeSubfCichorioideae",
"BetulaUndiff",
"Boraginaceae",
"Botrychium",
"Brassicaceae",
"Cactaceae",
"Campanulaceae",
"Caprifoliaceae",
"Carya",
"Caryophyllaceae",
"Castanea",
"Ceanothus",
"Celtis",
"Cephalanthus",
"Cercocarpus",
"ChenopodiaceaeAmaranthaceae",
"Chrysolepis",
"Cornus",
"Corylus",
"Cupressaceae",
"Cyperaceae",
"Dodecatheon",
"Dryas",
"Elaeagnaceae",
"Ephedra",
"Equisetum",
"EricaceaeChamadaphneLedum",
"EricaceaeVaccinium",
"EricaceaeUndiff",
"Ericales",
"Eriogonum",
"Euphorbiaceae",
"Fabaceae",
"Fagus",
"FraxinusNigraType",
"FraxinusPennsylvanicaAmericanaType",
"FraxinusUndiff",
"Iva",
"JuglansCinerea",
"JuglansNigra",
"JuglansUndiff",
"KoenigiaIslandica",
"Lamiaceae",
"LarixPseudotsuga",
"Larrea",
"Liliaceae",
"Liquidambar",
"Liriodendron",
"LycopodiumAnnotinum",
"LycopodiumClavatum",
"LycopodiumComplanatum",
"LycopodiumSelago",
"LycopodiumUndiff",
"Magnoliaceae",
"Malvaceae",
"Moraceae",
"Myricaceae",
"Nyssa",
"Onagraceae",
"Osmundaceae",
"OstryaCarpinus",
"Oxyria",
"Papaveraceae",
"Pedicularis",
"PiceaGlauca",
"PiceaMariana",
"PiceaUndiff",
```

```
  "PinusDiploxylon",
  "PinusHaploxylon",
  "PinusUndiff",
  "PinusSE",
  "Plantaginaceae",
  "Platanus",
  "Poaceae",
  "Polemoniaceae",
  "PolygonaceaeUndiff",
  "PolygonumBistortoides",
  "PolygonumUndiff",
  "PolygonumViviparum",
  "Polypodiaceae",
  "Populus",
  "Potentilla",
  "Prosopis",
  "Pteridium",
  "Quercus",
  "RanunculaceaeUndiff",
  "RhamnaceaeVitaceae",
  "RosaceaeUndiff",
  "Rubiaceae",
  "Rubus",
  "Rumex",
  "RumexOxyria",
  "Salix",
  "Sanguisorba",
  "Sarcobatus",
  "SaxifragaCernua",
  "SaxifragaHieracifolia",
  "SaxifragaOppositifolia",
  "SaxifragaTricuspidata",
  "SaxifragaceaeUndiff",
  "Scrophulariaceae",
  "Selaginella",
  "ShepherdiaCanadensis",
  "Sphagnum",
  "Taxodium",
  "Taxus",
  "Thalictrum",
  "Tilia",
  "TsugaHeterophylla",
  "TsugaMertensiana",
  "TsugaUndiff",
  "Ulmus",
  "Urticaceae",
  "Xanthium"),stringsAsFactors = FALSE)

POLLEN_SUMS <-c(0, 35,64,134,135)

FOSSIL_CSV <- data.frame(pSum = POLLEN_SUMS, filenames = c("psCustomPerc.fossil","ps35perc.fossil","ps64perc.fossil","ps134perc.fossil
      ","ps135perc.fossil"),stringsAsFactors = FALSE)

MODERN_CSV<-data.frame(pSum = POLLEN_SUMS, filenames = c("psCustomPerc.modern","ps35perc.modern","ps64perc.modern","ps134perc.modern","
      ps135perc.modern"),stringsAsFactors = FALSE)
createReconObj <- function(disObj,inModern,inFossil,clidex,clivar,modCust)
{

  # the number of analogs can be determined from the matrices

  x=disObj
  NumAnalogs <- nrow(x$sqdist)
  ecoCol=0

  # now we build add columns onto the dissimilarity object
  # we get site ID, lat lon, alt, sample id, age and cybpr
  # then we get distance matix

  #baseobj <-cbind(inFossil[,c(2,4:6,9:11)],t(x$sqdist))

  baseobj <-cbind(inFossil[,c(2,4:6,9:11)],t(x$sqdist[1:NumAnalogs,]))

  # Next we add a column for every analog some tricking indexing
  startCol <- ncol(baseobj)
  cnames <- vector()
```

```
  for( i in 1:NumAnalogs)
  {
    cnames <- c( cnames,paste("ID",i,sep=""))
    baseobj <- cbind(baseobj,inModern[x$position[i,],1])
  }
  for( i in 1:NumAnalogs)
  {
    cnames <- c( cnames,paste( toupper(clivar) ,i,sep=""))
    baseobj <- cbind(baseobj,inModern[x$position[i,],clidex])
  }
  if (modCust) # custom modern clim var is always after ecoregion col
     ecoCol <- ncol(inModern) - 1
  else
     ecoCol <- ncol(inModern)

  for( i in 1:NumAnalogs)
  {
    cnames <- c( cnames,paste( "ECOREGION" ,i,sep=""))
    baseobj <- cbind(baseobj,inModern[x$position[i,],ecoCol])
  }

  colnames(baseobj)[(startCol+1):ncol(baseobj)]<-cnames
  return(baseobj)

}createSCDplotsByEcoregion <- function(SCDByEcoregionFinal)
{

  # finally do a plot for each ecoregion, cal yr BP vs SCD values
  # 10 plots for each panel & format the plots nicely

  message("Writing plot to file SCD_ecoregions_plot.pdf in working directory...")
  tryCatch(
  {
    # option to save plot as pdf
    pdf("SCD_ecoregions_plot.pdf")

    par(mfrow=c(3,3))
    # these values can be adjusted
    par(mar = c(2.5, 3.5, 2, 1))
    par(mgp = c(1.5, 0.5, 0))
    par(oma = c(10, 1, 1, 1))

    plotIndex = 2 # first column of data
    plotIndexMax = ncol(SCDByEcoregionFinal) - 1 # exclude Mediterranian ecoregion due to lack of data
    useEcoregion = 0
    scdCutoffVal = -99 # default value
    # ask user SCD value (e.g. 0.3) where vertical line will be placed (optional)
    while (is.na(scdCutoffVal) | (scdCutoffVal < 0 & scdCutoffVal != -1) | scdCutoffVal > 1)
       scdCutoffVal=as.numeric(readline("At what SCD value do you want the horizontal line placed? (valid range is 0 to 1, use -1 for no
              line) "))

    while (plotIndex <= plotIndexMax)
    {
       useEcoregion = 0 # reset this
       selColName <- colnames(SCDByEcoregionFinal)[plotIndex]
       selColName2 = gsub("SCD values","", selColName ,fixed=TRUE)
       selColName3 = gsub("SCD_val.","", selColName2 ,fixed=TRUE)

       # here is where the user can optionally exclude eco-regions
       while (is.na(useEcoregion) | useEcoregion < 1 | useEcoregion > 2)
         useEcoregion <- as.numeric(readline(paste("Do you wish to include ecoregion",selColName3,"<1 for yes, 2 for no> ")))

       if (useEcoregion == 1)
       {
         # SCD values need to be divided by 100
         properSCDval <- SCDByEcoregionFinal[,plotIndex]
         yLabel = paste("SCD value (",selColName3,")")
         plot(SCDByEcoregionFinal$CalYrBP,properSCDval,pch=21,cex=0.1,bg="red",xlab="cal yr BP",ylab=yLabel)
         if (scdCutoffVal != -1)
            abline(h=scdCutoffVal,col="gray")
       }
       plotIndex = plotIndex + 1
    }

    dev.off()
```

```
  }, error = function(e){message("SCD_ecoregions_plot.pdf cannot be written to disk, please close the file!")}, warning = function(w)
        {})

} ### end of functioncustomClimateOption <- function(modernData,climVar,modType,lonMin,lonMax,latMin,latMax)
{
  linkClimData <- NA
  linkRastData <- NA

  foundClimCol = NULL
  matchClimVar = NULL
  nameClimCol = NULL
  foundSelCol = NULL
  modernDataJoin = NULL
  climateDataJoinCol = NULL
  siteElevData = NULL
  stnElevData = NULL
  nc = NULL

  oWriteAppend = 2 # for testing
  fileTypeMatch = 0
  flagFound = 0
  selMonth = 0
  mCount = 0
  monDiff = 0
  totMonths = 0
  calcData = 0
  corrTemp = 0
  foundElev = 0
  siteElevJoin = 0
  stnElevJoin = 0
  lapseRate = 0
  selCol = 1
  colInd = 1
  climColNew = ""

  if (modType == 1) # station data
  {
    # ask user to select BEST or GHCN data set

    stnType = 0
    while (is.na(stnType) | stnType < 1 | stnType > 2)
      stnType=as.numeric(readline("Which station data set to select? <use 1 for BEST, 2 for GHCN> "))

    if (stnType == 1)
    {
      ###############################################
      # BEST code courtest of Steven Mosher
      ###############################################

      is.installed <- function(mypkg) is.element(mypkg, installed.packages()[,1])

      if(!is.installed("BerkeleyEarth"))install.packages("BerkeleyEarth")
      #if(!is.installed("RghcnV3"))install.packages("RghcnV3")

      if (packageDescription('BerkeleyEarth')$Version < "1.8") { install.packages('BerkeleyEarth') }
      #if (packageDescription('RghcnV3')$Version < "2.8") { install.packages('RghcnV3') }

      library(BerkeleyEarth)
      #library(RghcnV3)

      source("writeCustomBin.R")
      source("selectiveRead.R")

      fileList <- NULL
      message("")

      # ask user for MEAN (TAVG), TMAX or TMIN
      #valid <- FALSE
      #while(!valid)
      #{
      # cat("Do you want to use TAVG, TMAX or TMIN ? ")
      # Answer <- toupper(readLines(n=1))

      # if(Answer %in% c("TAVG","TMAX","TMIN"))
      # valid = TRUE
      #}
```

```
# check for BEST files on disk
tryCatch({fileList <- list.files("TAVG")
}, error = function(e) {return(NULL)}, warning = function(w){})

# check for these GHCN data files before downloading them
if ("LATEST-QualityControlled.zip" %in% fileList)
{
    message("Berkeley earth surface temperature TAVG files will be loaded from disk....")
    message("")

}else
{
  tryCatch({

    message("Attempting to download Berkeley earth surface temperature TAVG files....")
    message("")

    ################ List the berkeley Urls
    #BerkeleyUrls
    #urls=BerkeleyUrls[2,]

    # for now we will just download quality controlled data
    # if you wantto download all data just feed in all the urls

    downloadBerkeley()

  }, error = function(e) {message("Unable to download Berkeley earth surface temperature TAVG files....");return(NULL)}, warning
         = function(w){})
}

tryCatch({

  #set the BerkeleyData directory
  BerkDir <- file.path("TAVG","LATEST-QualityControlled", fsep = .Platform$file.sep)

  ## get the station inventory, detailed information
  Stations <- readSiteDetail(Directory= BerkDir)

  # ask user for month needed
  selMonth = 0
  while (is.na(selMonth) | selMonth < 1 | selMonth > 12)
     selMonth=as.numeric(readline("Enter the month of data to be selected: (use 1 for Jan, 2 for Feb, etc... "))

  ## read ALL the sites for the month of August
  ## I could also read my custom bin NH
  dataSelMonth <- selectiveRead(Directory = BerkDir, Month = selMonth, file="data.bin")

  message("")
  message("Standard baseline for modern climate data is 1961-1990")

  # ask user for years to average (e.g. 1961-1990)
  currYr = 2013
  startYr = 0
  endYr = currYr - 1

  # min 30 yr interval required
  while (is.na(startYr) | startYr < 1900 | startYr > 1980 | startYr >= endYr)
       startYr=as.numeric(readline("Enter starting year, must be between 1900 and 1980: "))
  while (is.na(endYr) | endYr > currYr-3 | endYr < 1930 | endYr <= startYr)
       endYr=as.numeric(readline(paste("Enter ending year, must be between 1930 and",currYr-3,": ",sep=" ")))

  ## Suppose I just want 1960-1991
  ## I use indexing to create logical vectors to subset the matrix
  ## "take all the rows of data where the Date is > 1960 and < 1992
  dataSubset <- dataSelMonth[which(dataSelMonth[,"Date"] >= startYr & dataSelMonth[,"Date"] < endYr),]

  ## Now Suppose I want only those stations within a certain Lat Lon
  NamStations <- cropInv(Stations, bBox = extent(lonMax,lonMin,latMin,latMax))

  # plot(Nam$Lon,Nam$Lat)
  # map("world",add= T)

  ## and I use the Ids to further select my temperatures

  dataSubset <- dataSubset[which(dataSubset[,"Id"] %in% NamStations$Id),]
```

```
    matchStns <- NamStations[which(NamStations[,"Id"] %in% dataSubset[,"Id"]),]

    stnInfo <- cbind(matchStns, matchStns[,"Lat"],matchStns[,"Lon"],matchStns[,"Name"],matchStns[,"Altitude"])

    # Now aggregate all years and average

    dataAvg <- aggregate(as.data.frame(dataSubset), list(dataSubset[,"Id"]), FUN = mean, simplify=T)

    BESTdata <- cbind (stnInfo,dataAvg)

    BESTdataFINAL <- cbind(BESTdata[,1:5],BESTdata[,ncol(BESTdata)])

    colnames(BESTdataFINAL)[ncol(BESTdataFINAL)] <- paste("TAVG_",selMonth,sep="")

    fileStr <- paste("NAM_BEST_MONTH_",selMonth,"_TAVG_",startYr,"-",endYr,".csv",sep="")

    message(paste("Writing file",fileStr,"to disk...",sep=" "))
    tryCatch({write.csv(BESTdataFINAL,fileStr,row.names=FALSE)}, error = function(e) {message(paste("Cannot write",fileStr,"to
          disk, file may be open!",sep=" "))}, warning = function(w){})

    message(paste("with",ncol(BESTdataFINAL),"data columns",sep=" "))
    message(paste("and",nrow(BESTdataFINAL),"stations",sep=" "))

    climateData <- BESTdataFINAL
    message("")
  }, error = function(e) {message("Unable to load Berkeley earth surface temperature TAVG files from disk....");return(NULL)},
       warning = function(w){})

}
else # stnType == 2
{

  # Some code from http://noconsensus.wordpress.com/2011/07/18/introduction-to-rghcnv3-steve-mosher/
  # Note that TMAX and TMIN had problems after downloading

  require(RghcnV3)

  # ask user for MEAN (TAVG), TMAX or TMIN
  #valid <- FALSE
  #while(!valid)
  #{
  # cat("Do you want to use TAVG, TMAX or TMIN ? ")
  # Answer <- toupper(readLines(n=1))

  # if(Answer %in% c("TAVG","TMAX","TMIN"))
  # valid = TRUE
  #}


  #if (Answer == "TAVG")
  # downloadV3(url = V3.MEAN.ADJ.URL, directory = getwd(), overwrite = TRUE, remove = FALSE)
  #else if (Answer == "TMAX")
  # downloadV3(url = V3.MAX.ADJ.URL, directory = getwd(), overwrite = TRUE, remove = FALSE)
  #else if (Answer == "TMIN")
  # downloadV3(url = V3.MIN.ADJ.URL, directory = getwd(), overwrite = TRUE, remove = FALSE)


  fileList <- list.files() # vector with list of files in working directory

  # check for these GHCN data files before downloading them
  if (!("ghcnm.tavg.latest.qca.tar.gz" %in% fileList) | !("ghcnm.tavg.latest.qca.tar" %in% fileList))
  {
    message("Downloading latest GHCN files (TAVG), this could take some time....")
    message("")
    files <- downloadV3()
    downloadV3(url = V3.MEAN.ADJ.URL, directory = getwd(), overwrite = TRUE, remove = FALSE)
    v3Mean <- readV3Data(filename = files$DataFilename, output ="Array")
    v3Inv <- readInventory(filename = files$InventoryFile)

  }else
  {
   message("Loading TAVG files from disk, this could take some time....")
   v3Mean <- readV3Data("ghcnm.v3.2.0.20130120/ghcnm.tavg.v3.2.0.20130120.qca.dat", output ="Array")
   v3Inv <- readInventory("ghcnm.v3.2.0.20130120/ghcnm.tavg.v3.2.0.20130120.qca.inv")
  }
```

```
  # ask user for years to average (e.g. 1961-1990)
  currYr = 2013
  startYr = 0
  endYr = currYr - 1

  message("Standard baseline for modern climate data is 1961-1990")
  message("")

  # min 30 yr interval required
  while (is.na(startYr) | startYr < 1900 | startYr > 1980 | startYr >= endYr)
    startYr=as.numeric(readline("Enter starting year, must be between 1900 and 1980: "))
  while (is.na(endYr) | endYr > currYr-3 | endYr < 1930 | endYr <= startYr)
    endYr=as.numeric(readline(paste("Enter ending year, must be between 1930 and ",currYr-3,": ",sep="")))

  selMonth = 0
  while (is.na(selMonth) | selMonth < 1 | selMonth > 12)
    selMonth=as.numeric(readline("Enter the month of data to be selected: (use 1 for Jan, 2 for Feb, etc... "))

  message("")

  NamData <- windowArray(v3Mean, start = startYr, end = endYr)
  # Anomaly <- createAnomaly(v3Mean)

  # North America limits (unless user has a spatial subset of reconstruction)
  lonLatLimits <- c(lonMax,lonMin,latMin,latMax)
  NamStat <- cropInv(v3Inv, bBox = extent(lonLatLimits))

  NamGhcnList <- intersectInvData(NamStat,NamData)

  message("Writing file NamGhcnList.csv to disk...")
  tryCatch({write.csv(NamGhcnList,"NamGhcnList.csv",row.names=FALSE)}, error = function(e) {message("Cannot write NamGhcnList.csv
        to disk, file may be open!")}, warning = function(w){})

  NamGhcnMat <- read.csv("NamGhcnList.csv")

  message("File NamGhcnList.csv has:")
  message(paste(ncol(NamGhcnMat),"data columns",sep=" "))
  message(paste(nrow(NamGhcnMat),"stations",sep=" "))

  NamSubsetInfo <- NamGhcnMat[,1:16]
  NamSubsetTemp <- NULL
  NamTempAvg <- NULL
  NamGHCNfinal <- NULL

  message("Averaging the data for selected month for selected range of years....")

  # first temp data column (Jan of oldest year selected) is 17
  # if selMonth is 7 (July), starting column for oldest year selected is 23
  # NA values are ignored when doing rowSums

  colInd <- selMonth + 16
  while (colInd <= ncol(NamGhcnMat))
  {
    if (colInd == selMonth)
      NamSubsetTemp <- NamGhcnMat[,colInd]
    else
      NamSubsetTemp <- cbind(NamSubsetTemp,NamGhcnMat[,colInd])
    colInd <- colInd + 12
  }

  # calculate the averages, fix up the column names & write out the file
  NamTempAvg <- rowSums(NamSubsetTemp,na.rm=TRUE) / (endYr - startYr + 1)
  NamGHCNfinal <- cbind(NamSubsetInfo,NamTempAvg)

  fileStr <- paste("NAM_GHCN_MONTH_",selMonth,"_TAVG_",startYr,"-",endYr,".csv",sep="")

  colnames(NamGHCNfinal) = gsub("Inventory.","",colnames(NamGHCNfinal),fixed=TRUE)
  colnames(NamGHCNfinal)[ncol(NamGHCNfinal)] <- paste("TAVG_",selMonth,sep="")

  tryCatch({write.csv(NamGHCNfinal,fileStr,row.names=FALSE)}, error = function(e) {message(paste("Cannot write",fileStr,"to disk,
        file may be open!",sep=" "))}, warning = function(w){})

  climateData <- NamGHCNfinal

}
```

```
}
else if (modType == 2) # ask user for netcdf file
{

  # long term monthly mean files are 1981-2010 for NCEP and 1979-2000 for NARR
  # if using a monthly file can select any ranges from 1979-present (NARR) or 1948-present (NCEP)


  # NCEP doc at http://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis.html
  # NARR doc at http://www.esrl.noaa.gov/psd/data/gridded/data.narr.html

  # tested with both NARR and NCEP re-analyses, should work with other re-analyses in netcdf format

  # ERA-40 is in grib format, which would need to be converted into netcdf first
  # info at http://www.r-bloggers.com/ncep-global-forecast-system/

  fileTypeMatch = -1

  tryCatch({linkClimData <- file.choose()
  fileTypeMatch <- regexpr("mon.ltm.nc",basename(linkClimData))
  }, error = function(e) {}, warning = function(w){})

  if (!is.na(linkClimData) & fileTypeMatch != -1)
     nc <- open.ncdf(linkClimData)
  else
     message("Invalid netcdf file selected! Must be a valid re-analysis data file in .nc format.")

  dataName = NULL
  ncdata <- nc
  #summary(nc)

  if(!is.null(ncdata))
     print(ncdata) # show list of variables in the netcdf file

  ltMeanMatch = -1
  monMeanMatch = 1

  # determin if file is long term mean or monthly mean
  ltMeanMatch <- regexpr("mon.ltm.nc",basename(linkClimData))
  monMeanMatch <- regexpr("mon.mean.nc",basename(linkClimData))

  # if (ltMeanMatch == -1 & monMeanMatch == -1)
  if (ltMeanMatch == -1)
  {
     message("")
     message("Invalid re-analysis file selected! Must be a long term monthly mean file (i.e. varname.ltm.mon.nc)")
     return(NULL)
  }

  #fileStr1a = NULL
  #fileStr1b = NULL
  #fileStr1c = NULL

  #fileStr1a = gsub("ncep.","", basename(linkClimData) ,fixed=TRUE)
  #fileStr1a = gsub("narr.","", basename(linkClimData) ,fixed=TRUE)

  # now try to get variable name from file (these should not be renamed)
  #if (ltMeanMatch != -1) # e.g. air.2m.mon.ltm.nc (var is air)
  # fileStr1b = gsub(".mon.ltm.nc","", fileStr1a ,fixed=TRUE)
  #else
  # fileStr1b = gsub(".mon.mean.nc","", fileStr1a ,fixed=TRUE)

  #fileStr1c = unlist(strsplit(fileStr1b, "\\."))

  #vName = fileStr1c[1]
  #message(paste("Variable of interest is:",vName,sep=" "))

  vName=readline("What is the short name of the variable of interest? (usually just above lat or below lon in output above, e.g. air
         for surface temp) ")

  tryCatch({
  v3 <- nc$var[[vName]]
  varsize <- v3$varsize
  ndims <- v3$ndims
```

```
#print (v3)
#print (varsize)
#print (ndims)

latName="lat"
lonName="lon"

# output will show that 5 (4) dims is actually 4 (3) dims
# 4 data types (NCEP 4 dims, NCEP 4 dims, NARR 5 dims, NARR 5 dims)
# NARR 4 dims is time/y/x/nbnds, 5 dims is y/x/level/time/nbnds
# NCEP 4 dims is time/lat/lon/nbnds, 5 dims is lat/lon/level/time/nbnds
# no need to worry about nbnds (climatology bounds) for now
# not tested with any other re-analyses yet (future project)

levelSel = 0
dataType = 0

while (is.na(dataType) | dataType < 1 | dataType > 2)
   dataType=as.numeric(readline("What type of re-analysis file is this? <use 1 for NARR or 2 for NCEP> "))

if (dataType == 1 & ndims == 4)
{
   message("NARR has 29 pressure levels (hPa):")
   message("1000, 975, 950, 925, 900, 875, 850, 825, 800, 775, 750, 725, 700, 650, 600, 550, 500, 450, 400, 350, 300, 275, 250, 225,
          200, 175, 150, 125, 100")
   message("")

   while (is.na(levelSel) | levelSel < 1 | levelSel > 29)
      levelSel=as.numeric(readline("Enter the index of the pressure level needed (must be between 1 for 1000mb to 29 for 100mb) "))
}


if (dataType == 2 & ndims == 4)
{
   message("NCEP has 17 pressure levels (hPa):")
   message("1000, 925, 850, 700, 600, 500, 400, 300, 250, 200, 150, 100, 70, 50, 30, 20, 10")
   message("")

   while (is.na(levelSel) | levelSel < 1 | levelSel > 17)
      levelSel=as.numeric(readline("Enter the index of the pressure level needed (must be between 1 for 1000mb to 17 for 10mb) "))
}

getLon = dim(get.var.ncdf(nc,lonName))
getLat = dim(get.var.ncdf(nc,latName))

lon = get.var.ncdf(nc,lonName)
lat = get.var.ncdf(nc,latName)

t1 = NULL

endMonth <- 0
selMonth <- 0

while (is.na(selMonth) | selMonth < 1 | selMonth > 12)
   selMonth=as.numeric(readline("Enter the starting month of data to be selected (use 1 for Jan, 2 for Feb, ...): "))

while (is.na(endMonth) | endMonth < 1 | endMonth > 12 | endMonth < selMonth)
   endMonth=as.numeric(readline("Enter the ending month of data to be selected (use 1 for Jan, 2 for Feb, ...): "))


if (dataType == 2 & ltMeanMatch == -1)
{
  # ask user for years to average (e.g. 1961-1990) for NCEP monthly files only
  # NARR long term mean is 1979-2000, no need to get any other subset for NARR

  currYr = 2013
  startYr = 0
  endYr = currYr - 1

  message("Standard baseline for modern climate data is 1961-1990")
  message("")

  # min 30 yr interval required
  while (is.na(startYr) | startYr < 1950 | startYr > 1990 | startYr >= endYr)
     startYr=as.numeric(readline("Enter starting year, must be between 1950 and 1990: "))
  while (is.na(endYr) | endYr > currYr-3 | endYr < 1980 | endYr <= startYr)
```

```
    endYr=as.numeric(readline(paste("Enter ending year, must be between 1980 and ",currYr-3,": ",sep="")))

}

if (monMeanMatch == -1) # handle long term mean file here
{

  if (dataType == 2 & ndims == 3) # NCEP 4 dims
     t1 = get.var.ncdf(nc,vName,start=c(1,1,selMonth),count=c(getLon,getLat,1))
  else if (dataType == 1 & ndims == 3) # NARR 4 dims getLon[1] identical to getLat[1]
     t1 = get.var.ncdf(nc,vName,start=c(1,1,selMonth),count=c(getLon[1],getLon[2],1))
  else if (dataType == 2 & ndims == 4) # NCEP 5 dims
     t1 = get.var.ncdf(nc,vName,start=c(1,1,levelSel,selMonth),count=c(getLon,getLat,1,1))
  else if (dataType == 1 & ndims == 4) # NARR 5 dims getLon[1] identical to getLat[1]
     t1 = get.var.ncdf(nc,vName,start=c(1,1,levelSel,selMonth),count=c(getLon[1],getLon[2],1,1))

  # now get lat, lon, value
  t1data = data.frame(cbind(as.vector(lat),as.vector(lon),as.vector(t1)))

  # get rid of any NA values in column 3 (this is necessary for projected grids such as NARR)
  dataTable <- subset(t1data,!is.na(t1data[,3]))

  # NARR prmsl (mslp) values are in hPa but need to be divided by 100
  if (vName == "prmsl")
     dataTable[,ncol(dataTable)] <- dataTable[,ncol(dataTable)] / 100

  # NARR air (2m temp) values are in K but need to be converted to deg C
  if (vName == "air" & dataType == 1)
     dataTable[,ncol(dataTable)] <- dataTable[,ncol(dataTable)] - 273.15

  #if (vName == "hgt" & dataType == 1 & levelSel == 17) # NARR 500 hgt values convert from m to dam
  # dataTable[,ncol(dataTable)] <- dataTable[,ncol(dataTable)] / 10

  message(paste("First data value for",vName,"is shown below:",sep=" "))
  message(dataTable[1,ncol(dataTable)])

  names(dataTable)=c(latName,lonName,toupper(vName))

  if (dataType == 1)
     dataName = "NARR_1979-2000_MEAN"
  else # dataType == 2
     dataName = "NCEP_1981-2010_MEAN"

  # write a file with a descriptive name here
  if (levelSel == 0) # no level selected
     fileStr = paste(dataName,"_",toupper(vName),"_MONTH_",selMonth,".csv",sep="")
  else
     fileStr = paste(dataName,"_",toupper(vName),levelSel,"_MONTH_",selMonth,".csv",sep="")

  message(paste("Writing file",fileStr,"to disk...",sep=" "))

  tryCatch({write.csv(dataTable,fileStr,row.names=FALSE)}, error = function(e) {message(paste("Cannot write",fileStr,"to disk, file
         may be open!",sep=" "))}, warning = function(w){})

  climateData <- dataTable

}else # monthly mean file - some code from https://stat.ethz.ch/pipermail/r-sig-geo/2009-April/005460.html
{
    nt <- varsize[ndims] # Remember timelike dim is always the LAST dimension!

    # need starting and ending timesteps for monthly means to average
    startInd = ((startYr - 1948) * 12) + selMonth
    endInd = ((endYr - 1948) * 12) + selMonth

    message(paste("Averaging variable",vName,"over",(endInd - startInd) / 12,"timesteps...",sep=" "))

    i = startInd
    calcData = 0
    dfCalcData <- NULL
    selData <- NULL

    while (i <= endInd) # to average months
    {
      mCount = 1
      if (endMonth == selMonth)
      {
```

```
      # for( i in 1:nt )
      # Initialize start and count to read one timestep of the variable.
      # start <- rep(1,ndims) # begin with start=(1,1,1,...,1)
      # start[ndims] <- i # change to start=(1,1,1,...,i) to read timestep i
      # count <- varsize # begin w/count=(nx,ny,nz,...,nt), reads entire var
      # count[ndims] <- 1 # change to count=(nx,ny,nz,...,1) to read 1 tstep

      if (dataType == 2)
      {
         if (ndims == 4) # ncep 4 dims
            selData <- get.var.ncdf( ncdata, v3, start=c(1,1,1,i), count=c(getLon,getLat,1,1) )
         else if (ndims == 5) # ncep 5 dims
            selData <- get.var.ncdf( ncdata, v3, start=c(1,1,1,1,i), count=c(getLon,getLat,1,1,1) )
      }

      calcData <- calcData + selData
      dfCalcData = data.frame(cbind(as.vector(lat),as.vector(lon),as.vector(calcData)))

      # Now read in the value of the timelike dimension
      # timeval <- get.var.ncdf( ncdata, v3$dim[[ndims]]$name,start=i, count=count )

      # print(paste("Data for variable",v3$name,"at timestep",i," (time value=",timeval,v3$dim[[ndims]]$units,"):"))

    }else # to average seasons
    {
       monDiff <- endMonth - selMonth
       totMonths = totMonths + mCount
       mCount = 1 # count number of months in range of months (for each year, e.g. JJA) specified by user
       while (mCount <= monDiff)
       {
          if (dataType == 2)
          {
             if (ndims == 4) # ncep 4 dims
                selData <- get.var.ncdf( ncdata, v3, start=c(1,1,1,i), count=c(getLon,getLat,1,1) )
             else if (ndims == 5) # ncep 5 dims
                selData <- get.var.ncdf( ncdata, v3, start=c(1,1,1,1,i), count=c(getLon,getLat,1,1,1) )
          }
          calcData <- calcData + selData
          dfCalcData = data.frame(cbind(as.vector(lat),as.vector(lon),as.vector(calcData)))
          mCount = mCount + 1
          i = i + 1
       }
    }
    i = i + 12 # only selected month
  }

  #write.csv(dfCalcData,"dfCalcData.csv",row.names=FALSE)

  if (endMonth == selMonth) # different timesteps to calc averages (monthly vs seasonal)
     dfCalcData[,3] <- dfCalcData[,3] / (endYr - startYr)
  else
     dfCalcData[,3] <- dfCalcData[,3] / totMonths

  colnames(dfCalcData)[1] <- "lat"
  colnames(dfCalcData)[2] <- "lon"
  colnames(dfCalcData)[3] <- vName

  if (dataType == 1)
     dataName = "NARR"
  else # dataType == 2
     dataName = "NCEP"

  # write a file with a descriptive name here
  if (levelSel == 0) # no level selected
      fileStr = paste(dataName,"_",toupper(vName),"_MEAN_MONTHS_",selMonth,"-",endMonth,"_",startYr,"-",endYr,".csv",sep="")
  else
      fileStr = paste(dataName,"_",toupper(vName),"_",levelSel,"_MEAN_MONTHS_",selMonth,"-",endMonth,"_",startYr,"-",endYr,".csv",
            sep="")

  message(paste("Writing file",fileStr,"to disk...",sep=" "))

  tryCatch({write.csv(dfCalcData,fileStr,row.names=FALSE)}, error = function(e) {message(paste("Cannot write",fileStr,"to disk,
        file may be open!",sep=" "))}, warning = function(w){})

  climateData <- dfCalcData
}
```

```
  }, error = function(e) { message("Error processing netcdf file, possibly the variable you entered was invalid!"); return(NULL) },
       warning = function(w){})
}
else if (modType == 3) # raster data (this is not working yet)
{
   require(raster)

   message("NOTE: Only .bil raster data sets have been tested with this function!")

   tryCatch({linkRastData <- file.choose()
       fileTypeMatch <- regexpr(c(".bil"),basename(linkRastData))
       }, error = function(e) {}, warning = function(w){})


   if (!is.na(linkRastData) & fileTypeMatch != -1)
   {
      tryCatch({
       r = raster(linkRastData)

       wcVar = gsub(".bil","", basename(linkRastData) ,fixed=TRUE) #var name is tmean1, tmean2, etc.

       plotRast = 0
       while (is.na(plotRast) | plotRast < 1 | plotRast > 2)
          plotRast=as.numeric(readline("Do you want to plot a map of the raster file loaded? <use 1 for YES or 2 for NO> "))

       if (plotRast == 1)
          plot(r)

       lonlat <- cbind(modernData["LONDD"],modernData["LATDD"])
       rastVar <- extract(r, lonlat, method='simple') # extract raster value at each mod pollen site


       wcOut <- cbind(lonlat,rastVar) # lon, lat & value from raster
       colnames(wcOut)[3] <- toupper(wcVar)

       corrTemp = 0
       while (is.na(corrTemp) | corrTemp < 1 | corrTemp > 2)
       {
          corrTemp <- as.numeric(readline("Do you want to correct the temperature values of the raster? (WorldClim values will be
                divided by 10) <1 for YES, 2 for NO> "))
           if (corrTemp == 1)
             wcOut[,3] <- wcOut[,3] / 10
       }


       wcRows <- nrow(wcOut)
       rowInd = 1
       while (rowInd <= wcRows)
       {
          if (is.na(wcOut[rowInd,3]))
          {
             message("NA values found, these values will be substituted with the appropriate values from the modern pollen file!")
             while(length(foundClimCol) == 0)
             {
                 cat("Enter the name of the column with the climate data needed from this file (not case sensitive):","\n")
                 climColName <- readLines(n=1)
                 foundClimCol <- which(colnames(modernData) == climColName | colnames(modernData) == toupper(climColName)) # index of
                       column with climate data
                 print(foundClimCol)
             }
             rowInd = 4835
          }
          rowInd = rowInd + 1
        }
        rowInd = 1
        while (rowInd <= wcRows)
        {
          if (is.na(wcOut[rowInd,3]))
             wcOut[rowInd,3] = modernData[rowInd,foundClimCol]
          rowInd = rowInd + 1
        }


       fileStr <- paste("worldclim_",wcVar,".csv",sep="")
       tryCatch({write.csv(wcOut,fileStr,row.names=FALSE)}, error = function(e) {message(paste("Cannot write",fileStr,"to disk, file
             may be open!",sep=" "))}, warning = function(w){})
```

```
       climateData <- wcOut
     }, error = function(e) {message("Invalid raster file format!");return(NULL)}, warning = function(w){})

   }
   else
     message("Invalid raster data file selected, required format is .bil!")

}
else if (modType == 4) # custom climate data in csv format
{

   tryCatch({linkClimData <- file.choose()
       fileTypeMatch <- regexpr(".csv",basename(linkClimData))
      }, error = function(e) {}, warning = function(w){})

   if (!is.na(linkClimData) & fileTypeMatch != -1)
     climateData <- read.csv(linkClimData, TRUE, ",")
   else
     message("Invalid custom climate data file selected! Must be a .csv file.")
}

# now that we have csv from any of the options above, we can do stuff!

tryCatch({

  if (modType != 3)
  {

    # lat/lon field names could be any of these
    latList <- c("lat","Lat","LAT","latitude","Latitude","LATITUDE","LATDD")
    lonList <- c("lon","Lon","LON","longitude","Longitude","LONGITUDE","LONDD")
    elevList <- c("ELEVATION","Elevation","ALTITUDE","Altitude")

    # get column index of lat/lon fields
    foundLat <- which(colnames(climateData) %in% latList)
    foundLon <- which(colnames(climateData) %in% lonList)
    foundElev <- which(colnames(climateData) %in% elevList)
  }

  message("List of column names in loaded climate data file: ")
  message("")
  lastCol = ncol(climateData)
  while (selCol <= lastCol)
  {
     message(colnames(climateData)[selCol])
     selCol = selCol + 1
  }
  message("")


  while(length(foundClimCol) == 0)
  {
     cat("Enter the name of the column with the climate data needed from this file (not case sensitive):","\n")
     climColName <- readLines(n=1)
     foundClimCol <- which(colnames(climateData) == climColName | colnames(climateData) == toupper(climColName)) # index of column
           with climate data
  }

  if (modType != 3)
  {
    if (length(foundLat) == 0 | length(foundLon) == 0)
    {
       message("Custom climate data file missing either lat or lon fields!")
    }else
    {
      # try to see if climate variable name selected from file already exists in mod climate file
      matchClimVar <- which(colnames(modernData) == toupper(climColName))

      if (length(matchClimVar) > 0)
      {
        flagFound = 1
        oWriteAppend = 2 # cannot append col to modern pollen file as that would create a duplicate
        message(paste("Climate variable",climColName,"already exists in modern pollen database file!",sep=" "))
      }else
      {
```

```
      flagFound = 0
      message(paste("Climate variable",climColName,"does not exist in modern pollen database file!",sep=" "))
    }

    # does user want to overwrite existing modern climate variable or append to file?
    while (is.na(oWriteAppend) | oWriteAppend < 1 | oWriteAppend > 2)
      oWriteAppend <- as.numeric(readline("Do you want to append this column or overwrite an existing column (climate variable) in
            the modern pollen file? <1 for APPEND, 2 for OVERWRITE> "))

    if (length(foundElev))
    {
      climDataSubset <- cbind(climateData[foundLon],climateData[foundLat],climateData[foundClimCol],climateData[foundElev])
      colnames(climDataSubset)[4] = "elev"
    }else
    {
      climDataSubset <- cbind(climateData[foundLon],climateData[foundLat],climateData[foundClimCol])
    }

    nameClimCol <- colnames(climDataSubset)[3]
    colnames(climDataSubset)[3] = "val"

    message("Joining the nearest climate station or gridpoint (for selected climate variable) to each modern pollen site, this could
           take a long time depending on the data set used......")

    if (ncol(climDataSubset) == 4)
      climateDataJoin <- joinClimateData(modernData,climDataSubset,colnames(climDataSubset)[2],colnames(climDataSubset)[1],colnames
            (climDataSubset)[4])
    else
      climateDataJoin <- joinClimateData(modernData,climDataSubset,colnames(climDataSubset)[2],colnames(climDataSubset)[1],NULL)

    if (length(foundElev) > 0)
    {
      while (is.na(corrTemp) | corrTemp < 1 | corrTemp > 2)
        corrTemp <- as.numeric(readline("Do you want to correct the temperature values of station data based on elevation
              difference (using lapse rate) between modern pollen site and nearest station? <1 for YES, 2 for NO> "))
        if (corrTemp == 1)
        {
          while (is.na(lapseRate) | lapseRate < 0.6 | lapseRate > 1)
            lapseRate <- as.numeric(readline("Enter the lapse rate in degrees Celsius per 100m (must be between 0.6 and 1): "))
        }
    }

    if (ncol(climDataSubset) == 4)
      climateDataJoinCol <- climateDataJoin[,7]
    else
      climateDataJoinCol <- climateDataJoin[,6]

    if (length(foundElev) > 0 & corrTemp == 1)
    {
      # get pollen site elevation column
      siteElevJoin = which(colnames(climateDataJoin) == "SiteElevation")
      siteElevData <- climateDataJoin[,siteElevJoin]

      # get station elevation column
      stnElevJoin = which(colnames(climateDataJoin) == "StnElevation")
      stnElevData <- climateDataJoin[,stnElevJoin]

      # replace any NAs in stnElevJoin col with pollen site elevation
      j=1
      lenVec = length(stnElevData)
      while (j <= lenVec)
      {
        if (is.na(stnElevData[j]))
          stnElevData[j] = siteElevData[j]
        j = j+1
      }
      # do correction (pollen site elev - stn elev) / 100 * 0.8
      # assume lapse rate of 0.8c / 100m
      climateDataJoinCol <- climateDataJoinCol + (((stnElevData - siteElevData) / 100) * lapseRate)
    }
    # make sure data is ordered by siteID before updating the col of data for selected variable
    modernData <- modernData[order(modernData[,1]),]
    climateDataJoin <- climateDataJoin[order(climateDataJoin[,1]),]
    colnames(climateDataJoin)[6] <- nameClimCol
  }
}else
```

```
    {
      climateDataJoinCol = wcOut[,ncol(wcOut)]
    }

    if (oWriteAppend == 1) # simply tack on the extra col here
    {
       modernDataJoin <- cbind(modernData,climateDataJoinCol)
       message("New climate variable was appended to modern pollen file!")
       colnames(modernDataJoin) [ncol(modernDataJoin)] <- climColName
       modernData <- modernDataJoin
    }else if (oWriteAppend == 2)
    {

      if (flagFound == 1)
      {
        message(paste("Climate variable",climColName,"in the modern pollen database will be updated with the new climate data from
              column with same name in loaded file!",sep=" "))
        # figure out which climate variable col in modern pollen file needs to be updated
        climCol = matchClimVar
        # do the updating of the data in the column
        modernData[,climCol] = climateDataJoinCol
      }else if (flagFound == 0)
      {
        selColMod <- which(colnames(modernData) == "TJAN")
        endColMod <- which(colnames(modernData) == "SDJF")

        while (selColMod <= endColMod)
        {
           message(colnames(modernData)[selColMod])
           selColMod = selColMod + 1
        }
        message("")
        while(length(foundSelCol) == 0) # make sure climate variable name entered is in modern pollen database
        {
           cat("Enter the name of the column from the modern pollen database (listed above) to overwrite:","\n")
           climColName <- readLines(n=1)
           foundSelCol <- which(colnames(modernData) == toupper(climColName)) # index of column with climate data
        }
        # do the updating of the data in the column
        modernData[,foundSelCol] = climateDataJoinCol
        message(paste("Enter a new column name for ",climColName,":",sep=" "))
        climColNew <- readLines(n=1)
        climColNew <- toupper(climColNew)
        colnames(modernData)[foundSelCol] <- climColNew
      }
      message("Updating of modern climate data in modern pollen database file is complete!")
    }
    modList <- list(modernData, climColName, climColNew)
    return (modList)
  }, error = function(e) {message("Problem with processing csv data file, possibly one of the options 1-4 failed!"); return(NULL)},
       warning = function(w){})
}customPollenSum <- function(modernDB, fossilDB, params=PS135_TAXA_LIST)
{
  # keep running count of taxa selected with customPollenSumCount
  # customModernPollenSum will match customFossilPollenSum in terms of number of taxa!
  # if modern pollen sum has 90 taxa, fossil pollen sum will have 90 taxa!!
  # minimum number of taxa for a custom pollen sum is one taxa
  # when a taxa is not included or aggregated (n or a), it does not appear on the taxa list
  # any taxa aggregated to that selected taxa, will be removed from the list as well
  # it is not possible to aggregate a taxa with itself
  # aggregate taxa column name is always in uppercase to make it easy to identify
  # Also the user should have some knowledge of pollen taxa before choosing the custom
  # pollen sum option, otherwise the reconstruction file produced may have problems

  colToRemove = NULL
  colToRemove2 = NULL
  customList = NULL
  continueAg = NULL
  taxa = NULL
  selTaxa = NULL
  modCol = NULL
  fosCol = NULL
  newAgColMod = NULL
  newAgColFos = NULL
  newAgColName = NULL
  newAgColModSum = NULL
```

```
newAgColFosSum = NULL
newAgColModTemp = NULL
newAgColFosTemp = NULL
taxaColAgWithFos = NULL
taxaColAgWithMod = NULL
taxaToAgWithFos = NULL
taxaToAgWithMod = NULL
taxaColAgWith = NULL
colToRemoveID = NULL
colToRemoveID2 = NULL

removedTaxaList <- numeric() # for taxa that we remove or aggregate
agTaxaList <- character() # taxa for selected aggregation
taxaYesNoAg = ""
taxaToAgWith = -1
customPollenSumCount = 0
agCount = 0

#colnames(modernDB) <- tolower(colnames(modernDB))
#colnames(fossilDB) <- tolower(colnames(fossilDB))

customPollenSumModDB <- modernDB[,1:148] # make a copy of modern db
customPollenSumModDB_part2 <- modernDB[,149:230] # split since last taxa col is 148
customPollenSumFosDB <- fossilDB # make a copy of fossil db

for (selTaxa in 1:135)
{

   if(!(selTaxa %in% removedTaxaList))
   {

     # ask user whether to include each taxa (y/n/a) NOTE: a is to aggregate
     while (taxaYesNoAg != "y" & taxaYesNoAg != "n" & taxaYesNoAg != "a")
     {
        agCount = 0 # reset this for next aggregation
        taxaYesNoAg = readline(paste("Do you wish to include taxa ", toupper(params[selTaxa,2]), " in your custom pollen sum? Use <y>
              or <n> or <a> to aggregate taxa: "))
     }
     if(taxaYesNoAg == "y") # add taxa column only if user enters y
     {
       # update custom pollen sum count
       customPollenSumCount = customPollenSumCount + 1
       taxaYesNoAg = "" #to make sure we get back into the while loop

     }else if(taxaYesNoAg == "n") #add taxa column only if user enters y
     {
      removedTaxaList <- append(removedTaxaList, selTaxa)

      #remove the column that does not appear on the taxa list
      colToRemoveID <- which(colnames(customPollenSumFosDB)== toupper(params[selTaxa,2]))
      colToRemove <- colnames(customPollenSumFosDB)[colToRemoveID]
      customPollenSumFosDB=customPollenSumFosDB[, setdiff(colnames(customPollenSumFosDB),c(colToRemove)) ]

      colToRemoveID2 <- which(colnames(customPollenSumModDB)== toupper(params[selTaxa,2]))
      colToRemove2 <- colnames(customPollenSumModDB)[colToRemoveID2]
      customPollenSumModDB=customPollenSumModDB[, setdiff(colnames(customPollenSumModDB),c(colToRemove2)) ]

      taxaYesNoAg = "" # to make sure we get back into the while loop
     }else if(taxaYesNoAg == "a") # aggregate taxa option picked
     {
      customPollenSumCount = customPollenSumCount + 1
      removedTaxaList <- append(removedTaxaList, selTaxa)
      # list all taxa (pick one to aggregate selected taxa with)
      for (taxa in 1:135)
      {
        if(!(taxa %in% removedTaxaList)) # do not show removed or taxa that have already been aggregated
           message(paste(params[taxa,1],params[taxa,2] ))
      }
      continueAg = TRUE

      newAgColName=readline("Enter the name of the new aggregate column (will be converted to uppercase): ")
      newAgColName=toupper(newAgColName) # make aggregated taxa column name easy to identify

      modCol <- which(colnames(customPollenSumModDB)== toupper(params[selTaxa,2]))
      fosCol <- which(colnames(customPollenSumFosDB)== toupper(params[selTaxa,2]))
```

```
newAgColMod <- customPollenSumModDB[,modCol] # selected taxa to aggregate
newAgColFos <- customPollenSumFosDB[,fosCol]

while (continueAg)
{
  # cannot aggegate a taxa with itself or pick 0 (which is a flag to end aggregation)
  while (is.na(taxaToAgWith) | (taxaToAgWith < 1 & taxaToAgWith != 0) | taxaToAgWith == selTaxa | taxaToAgWith > 135)
  {
    taxaToAgWith <- as.numeric(readline(paste("Enter taxa ID to aggregate with selected taxa", params[selTaxa,1],toupper(
          params[selTaxa,2]),"<use 0 to finish aggregation & show which taxa are aggregated> ")))
  }

  if (taxaToAgWith == 0) # this allows us to stop aggregating taxa and go on to next taxa in 135 list
  {
    message(agTaxaList) # show list of aggregated taxa
    continueAg = FALSE # to make sure we exit the 2nd while loop
    taxaToAgWith = -1 # to make sure we get back into 3rd while loop
    taxaYesNoAg = "" # to make sure we get back into the first while loop
    agTaxaList = NULL # clear this vector for next aggregation

     # remove the original column that was part of the aggregation
     colToRemoveID <- which(colnames(customPollenSumFosDB)== toupper(params[selTaxa,2]))
     colToRemove <- colnames(customPollenSumFosDB)[colToRemoveID]
     customPollenSumFosDB=customPollenSumFosDB[, setdiff(colnames(customPollenSumFosDB),c(colToRemove)) ]

     colToRemoveID2 <- which(colnames(customPollenSumModDB)== toupper(params[selTaxa,2]))
     colToRemove2 <- colnames(customPollenSumModDB)[colToRemoveID2]
     customPollenSumModDB=customPollenSumModDB[, setdiff(colnames(customPollenSumModDB),c(colToRemove2)) ]

  }else
  {
    agCount = agCount + 1
    # keep track of removed taxa
    removedTaxaList <- append(removedTaxaList, taxaToAgWith)

    # figure out which column have the aggregated taxa
    taxaColAgWithFos <- which(colnames(customPollenSumFosDB)== toupper(params[taxaToAgWith,2]))
    taxaColAgWithMod <- which(colnames(customPollenSumModDB)== toupper(params[taxaToAgWith,2]))

    if (agCount == 1)
    {
      message(paste("You have aggregated ",agCount, " taxa"))
      # add these taxa together so we can do row sums on these cols
      newAgColModTemp <- cbind(newAgColMod,customPollenSumModDB[,taxaColAgWithMod])
      newAgColFosTemp <- cbind(newAgColFos,customPollenSumFosDB[,taxaColAgWithFos])

      # keep track of current taxa aggregation
      agTaxaList <- append(agTaxaList,params[selTaxa,2])
      agTaxaList <- append(agTaxaList," ")
      agTaxaList <- append(agTaxaList, params[taxaToAgWith,2])
      agTaxaList <- append(agTaxaList," ")

      # remove the column that was part of the aggregation
      colToRemoveID <- which(colnames(customPollenSumFosDB)== toupper(params[taxaToAgWith,2]))
      colToRemove <- colnames(customPollenSumFosDB)[colToRemoveID]
      customPollenSumFosDB=customPollenSumFosDB[, setdiff(colnames(customPollenSumFosDB),c(colToRemove)) ]

      colToRemoveID2 <- which(colnames(customPollenSumModDB)== toupper(params[taxaToAgWith,2]))
      colToRemove2 <- colnames(customPollenSumModDB)[colToRemoveID2]
      customPollenSumModDB=customPollenSumModDB[, setdiff(colnames(customPollenSumModDB),c(colToRemove2)) ]

      taxaToAgWith = -1 # need to reset this
      }else
      {
        message(paste("You have aggregated ",agCount, " taxa"))
        newAgColModTemp <- cbind(newAgColModTemp,customPollenSumModDB[,taxaColAgWithMod])
        newAgColFosTemp <- cbind(newAgColFosTemp,customPollenSumFosDB[,taxaColAgWithFos])
        agTaxaList <- append(agTaxaList, params[taxaToAgWith,2])
        agTaxaList <- append(agTaxaList," ")

        # remove the column that was part of the aggregation
        colToRemoveID <- which(colnames(customPollenSumFosDB)== toupper(params[taxaToAgWith,2]))
        colToRemove <- colnames(customPollenSumFosDB)[colToRemoveID]
        customPollenSumFosDB=customPollenSumFosDB[, setdiff(colnames(customPollenSumFosDB),c(colToRemove)) ]

        colToRemoveID2 <- which(colnames(customPollenSumModDB)== toupper(params[taxaToAgWith,2]))
```

```
            colToRemove2 <- colnames(customPollenSumModDB)[colToRemoveID2]
            customPollenSumModDB=customPollenSumModDB[, setdiff(colnames(customPollenSumModDB),c(colToRemove2)) ]
            taxaToAgWith = -1 # need to reset this
           }
          }
        }
        tryCatch({

        # sum the rows of the appropriate taxa cols
        newAgColModSum <- rowSums(newAgColModTemp)
        newAgColFosSum <- rowSums(newAgColFosTemp)

        # add aggregated col to custom fossil db
        customPollenSumFosDB <- cbind(customPollenSumFosDB, newAgColFosSum)
        lastCol <- ncol(customPollenSumFosDB)
        colnames(customPollenSumFosDB)[lastCol] <- newAgColName #set column name

        # add aggregated col to custom modern db
        customPollenSumModDB <- cbind(customPollenSumModDB, newAgColModSum)
        lastCol <- ncol(customPollenSumModDB)
        colnames(customPollenSumModDB)[lastCol] <- newAgColName #set column name
        taxaToAgWith = -1 # need to reset this

        }, error = function(e){message(paste("Error with aggregation (minimum of 2 taxa required)",toupper(params[selTaxa,2]),"will
              not be included!",sep=" "))}, warning = function(w){})

      }
     }
   }
  # return list with appropriate info
  customPollenSumModDBnew <- cbind(customPollenSumModDB,customPollenSumModDB_part2)
  customList <- list(customPollenSumModDBnew,customPollenSumFosDB,customPollenSumCount)
  return (customList)
}dataModelComparison <- function(tsReconData)
{
  # can get model output from http://www.ncdc.noaa.gov/paleo/ftp-gcmmodel.html
  # or http://www.ncdc.noaa.gov/paleo/modelvis.html
  # function adapted from http://r.789695.n4.nabble.com/HIRHAM-netcdf-files-td3704309.html

  # now get lat,lon and val from reconFile
  # find nearest model gridpoint to each reconFile point
  # after join simply subtract modelVal - reconVal (or vice versa)
  # output to a csv & maybe create an anomaly map - could use spplot

  require(ncdf)
  require(sp)

  spJoinMethod = 0
  numDims = 0

  ncfile <- NA
  tryCatch({
     ncfile <- file.choose()
     }, error = function(ex) {
  })

  if (!is.na(ncfile))
  {

    fileTypeMatch <- regexpr(".nc",basename(linkReconInput))

    if (fileTypeMatch != -1)
    {

      nc <- open.ncdf(ncfile) # check for valid nc file here?

      print(nc) # show list of variables in the file

      latName=readline("What is the name of the latitude field in the model output from above (e.g. lat) ")
      lonName=readline("What is the name of the longitude field in the model output from above (e.g. lon) ")

      vName=readline("What is the name of the variable of interest in the model output from above (e.g. TS) ")

      message("NOTE: 3 dimensions is [longitude,latitude,time] & 4 dimensions is [longitude,latitude,time,level]")

      while (is.na(numDims) | numDims < 3 | numDims > 4)
```

```
   numDims=as.numeric(readline("How many dimensions in the variable of interest from above (must be either be 3 or 4) "))

#dim(get.var.ncdf(nc,lonName))
getLon = dim(get.var.ncdf(nc,lonName))

#dim(get.var.ncdf(nc,latName))
getLat = dim(get.var.ncdf(nc,latName))

lon = get.var.ncdf(nc,lonName)
lat = get.var.ncdf(nc,latName)

# e.g. c(1,1,1,1) is starting value for lon, lat, level & time, same idea for c(getLon,getLat,1,1)
# but need to check whether variable has 3 or 4 dims (lon,lat,time,level is the 4th)
# we are getting timestep #1 here as the model output is an avg at a given interval (e.g. 6ka)

if (numDims == 3)
   t1 = get.var.ncdf(nc,vName,start=c(1,1,1),count=c(getLon,getLat,1))
else if (numDims == 4)
   t1 = get.var.ncdf(nc,vName,start=c(1,1,1,1),count=c(getLon,getLat,1,1))

t1data = data.frame(cbind(as.vector(lat),as.vector(lon),as.vector(t1)))
names(t1data)=c("lat","lon","val")

#coordinates(t1data)=~lon+lat
climModelNC = t1data

# fix col headers
colnames(tsReconData)[1] = "SiteID"
colnames(tsReconData)[2] = "Lon"
colnames(tsReconData)[3] = "Lat"
colnames(tsReconData)[4] = "avgVal"

while (is.na(spJoinMethod) | spJoinMethod < 1 | spJoinMethod > 2)
   spJoinMethod <- as.numeric(readline("Which method of spatial join do you wish to use? <1 for Great Circle distance, 2 for
         Euclidean> "))

# SPATIAL JOIN CODE

# we already have the modern DB file loaded so get a subset of the data
# lat, lon, elevation, mod climate columns, and fedorova column
# use this file in the spatial join of modern to fossil sites

# parameters are A, B, C, D, E

#do spatial join between between fossil and modern (closest within specified distance)

# adapted from http://www.nceas.ucsb.edu/scicomp/usecases/AssignClosestPointsToPoints

# Latest version: Assign closest points from a second point set


#temperaturePoints <- read.csv("temperaturePoints.csv")
#densityPoints <- read.csv("densityPoints.csv")

# promote the input lists to SpatialPointsDataFrames


coordinates(climModelNC) <- c("lon", "lat")
coordinates(tsReconData) <- c("Lon", "Lat")

# Remove temparature points with 'invalid' value of -199

#validTempPoints <- temperaturePoints[!temperaturePoints$temperature %in% c(-199),]

# Define these vectors, used in the loop.

closestSiteVec <- vector(mode = "numeric",length = nrow(tsReconData))
minDistVec <- vector(mode = "numeric",length = nrow(tsReconData))


# Get the vector index of the temperature station closest to each field station.
# Use the spDistsN1 function to compute the distance vector between each
# field station site and all of the temperature stations. Then, find and
# retain the actual temperature, and the index of the closest temperature
# to each transect station.
#
```

```
    # spDistsN1 usage: spDistsN1(pointList, pointToMatch, longlat)
    #
    # where:
    # pointList : List of candidate points.
    # pointToMatch: Single point for which we seek the closest point in pointList.
    # longlat : TRUE computes Great Circle distance in km, (BETTER)
    # FALSE computes Euclidean distance in units of input geographic coordinates
    #
    # We use Great Circle distance to increase distance calculation accuracy at high latitudes
    # See the discussion of distance units in the header portion of this file
    #
    # minDistVec stores distance from the closest temperature station to each density measurement point.
    # closestSiteVec stores the index of the closest temperature station to each density measurement point.
    #

    for (i in 1 : nrow(tsReconData))
    {
      if (spJoinMethod == 1)
         distVec <- spDistsN1(climModelNC,tsReconData[i,],longlat = TRUE)
      else # spJoinMethod == 2
         distVec <- spDistsN1(climModelNC,tsReconData[i,],longlat = FALSE)

      minDistVec[i] <- min(distVec)
      closestSiteVec[i] <- which.min(distVec)
    }

    #
    # Create the Temperature Assignment table: merge the temperature point list with the transect point list
    # into a five-column table by merging the temperature point and transect point lists.
    #

    PointAssignTemps1 <- as(climModelNC[closestSiteVec,]$val,"numeric")
    FinalTable1 = data.frame(tsReconData$SiteID, coordinates(tsReconData),tsReconData$avgVal,closestSiteVec,minDistVec,
          PointAssignTemps1)


    # Update the Temperature Assignment table column names
    #
    names(FinalTable1) <- c("siteID","Lon","Lat","avgValTS","CloseSite","DistDD","modelVal")
    #
    # And, at the end, write the point assignment file.
    #

    #cal the data-model diff and add this column to matrix
    dataModelDiff <- FinalTable1[,4] - FinalTable1[,7]
    FinalTable2 <- cbind(FinalTable1,dataModelDiff)

    # fix this col header
    colnames(FinalTable2)[8] = "dataModelDiff"

    message("Writing data-model comparison table to working directory in .csv format...")
    tryCatch({write.csv(FinalTable2,file="dataModelCompOutput.csv",row.names=FALSE)}, error = function(e){message("
          dataModelCompOutput.csv cannot be written to disk, please close the file!")}, warning = function(w){})

    return(FinalTable2)
   }else
    message("Invalid netCDF file selected!")
  }
}doLinearInterp <- function(fossilDataFrame,colFirstTaxa=C1,colLat=2,colLon=3,colCalmean=4,colE.=1,target.dates=targets)
# modified code from Dr. Mike Sawada (code is further down in this script)
{
  # this function written by Dr. Mike Sawada and modified by Matthew Ladd (March 16, 2011)
  # modifed to read and write lat/lon values, neccessary for mapping in GIS software
  # not neccessary to change anything in this function!!

  holdnames<-as.vector(unlist(names(fossilDataFrame)))

  # get enumbers and the date column plus the columns with the variables to interpolate.
  datainterp1 <- as.matrix(fossilDataFrame[,c(colE.,colLat,colLon,colCalmean,colFirstTaxa)])

  #write.csv(datainterp1,"datainterp1.csv")

  # make no dimnames to speed up calculations
  dimnames(datainterp1)<-NULL

  # get list of enums for each site
```

```
  enumbers<-as.numeric(unique(datainterp1[,1]))

  outtable<-vector("numeric")

  for(i in 1:length(enumbers))
  {
    datainterp<-datainterp1[which(datainterp1[,1] == enumbers[i]), ]

    #write.csv(datainterp,"datainterp.csv")

    if(length(dim(datainterp)) == 2)
    {

      tryCatch({interpolatedPrcts<-data.frame(approx(datainterp[,4],datainterp[,5],target.dates))}, error = function(e){message("Cannot
             do linear interpolation! Please change the cal yr BP range (menu item 4) and/or change the interpolation interval for the
            reconstruction.")}, warning = function(w){message(w)})

      #write.csv(interpolatedPrcts,"interpolatedPrcts.csv")
      inter<-cbind(E. = rep(fossilDataFrame[which(fossilDataFrame[,colE.] == enumbers[i])[1],colE.],length(target.dates)),Lat = rep(
            fossilDataFrame[which(fossilDataFrame[,colE.] == enumbers[i])[1],colLat],length(target.dates)),Lon = rep(fossilDataFrame[
            which(fossilDataFrame[,colE.] == enumbers[i])[1],colLon],length(target.dates)),interpolatedPrcts)
      #write.csv(inter,"inter.csv")
      outtable <- rbind(outtable,inter[which(is.na(inter[,5])==F), ])
      #write.csv(outtable,"outtable.csv")
    }
  }

  colnames(outtable)[1] <- "SiteID"
  colnames(outtable)[2] <- "Lat"
  colnames(outtable)[3] <- "Lon"
  colnames(outtable)[4] <- "CalMean"
  colnames(outtable)[5] <- "InterpVal"

  return (outtable)
}
# END OF FUNCTION #
doSpatialJoin <- function( reconInterpData, modClimate, spJoinOption, filename )
{

  # we already have the modern DB file loaded so get a subset of the data
  # lat, lon, elevation, mod climate columns, and fedorova column
  # use this file in the spatial join of modern to fossil sites

  # parameters are A, B, C, D, E

  #do spatial join between between fossil and modern (closest within specified distance)

  # code adapted from http://www.nceas.ucsb.edu/scicomp/usecases/AssignClosestPointsToPoints

  # Latest version: Assign closest points from a second point set

     require(sp)

     #temperaturePoints <- read.csv("temperaturePoints.csv")
     #densityPoints <- read.csv("densityPoints.csv")

  # promote the input lists to SpatialPointsDataFrames

     coordinates(modClimate) <- c("LONDD", "LATDD")
     coordinates(reconInterpData) <- c("Lon", "Lat")

  # Remove temparature points with 'invalid' value of -199

     #validTempPoints <- temperaturePoints[!temperaturePoints$temperature %in% c(-199),]

  # Define these vectors, used in the loop.

     closestSiteVec <- vector(mode = "numeric",length = nrow(reconInterpData))
     minDistVec <- vector(mode = "numeric",length = nrow(reconInterpData))

  # Get the vector index of the temperature station closest to each field station.
  # Use the spDistsN1 function to compute the distance vector between each
  # field station site and all of the temperature stations. Then, find and
  # retain the actual temperature, and the index of the closest temperature
  # to each transect station.
  #
```

```
  # spDistsN1 usage: spDistsN1(pointList, pointToMatch, longlat)
  #
  # where:
  # pointList : List of candidate points.
  # pointToMatch: Single point for which we seek the closest point in pointList.
  # longlat : TRUE computes Great Circle distance in km, (BETTER)
  # FALSE computes Euclidean distance in units of input geographic coordinates
  #
  # We use Great Circle distance to increase distance calculation accuracy at high latitudes
  # See the discussion of distance units in the header portion of this file
  #
  # minDistVec stores distance from the closest temperature station to each density measurement point.
  # closestSiteVec stores the index of the closest temperature station to each density measurement point.
  #
     for (i in 1 : nrow(reconInterpData))
     {
        if (spJoinOption == 1)
           distVec <- spDistsN1(modClimate,reconInterpData[i,],longlat = TRUE)
        else # spJoinOption == 2
           distVec <- spDistsN1(modClimate,reconInterpData[i,],longlat = FALSE)

        minDistVec[i] <- min(distVec)
        closestSiteVec[i] <- which.min(distVec)
     }

  #
  # Create the Temperature Assignment table: merge the temperature point list with the transect point list
  # into a five-column table by merging the temperature point and transect point lists.
  #
     PointAssignTemps <- as(modClimate[closestSiteVec,]$modClimVal,"numeric")

     #need modClimate$ECOREGION where modClimate$ID2 matches siteID in closestSiteVec
     ecoregionVec <- modClimate[closestSiteVec,]$ECOREGION

     FinalTable = data.frame(reconInterpData$SiteID,coordinates(reconInterpData),reconInterpData$CalMean,reconInterpData$InterpVal,
           closestSiteVec,minDistVec,PointAssignTemps,ecoregionVec)

  #
  # Update the Temperature Assignment table column names
  #
     names(FinalTable) <- c("SiteID","Lon","Lat","CalMean","InterpVal","CloseSiteIndex","Distance","ModClimVal","Ecoregion")
  #
  # And, at the end, write the point assignment file.
  #
  #if(!is.null(filename)){
  # message("Writing modern/fossil assignment table to working directory in .csv format")
  # tryCatch({write.csv(FinalTable,file=filename)}, error = function(e){message("SpatialJoinReconModEcoregions.csv cannot be written to
         disk, please close the file!")}, warning = function(w){})
  #}

  return(FinalTable)
}
getAnalogCount <- function(range = 2:20)
{
  valid = FALSE
  while(!valid){
    cat("How many analogs from",range[1], "to", range[length(range)],"?\n")
    val <- as.numeric(readline())
    if((val %in% range & !is.na(val)))valid <- TRUE
  }
  return(val)
}getAnalogTotal <- function(reconData)
{
  EcoregionYN = 0

  # determine number of analogs from number of columns in reconData matrix
  numberOfCols <- ncol(reconData)

  EcoregionYN <- regexpr("ECOREGION",colnames(reconData)[numberOfCols])

  if (EcoregionYN == -1) # file has no ecoregion columns
     Analogs <- c(10,13,16,19,22,25,28,31,34,37,40,43,46,49,52,55,58,61,64,67)
  else # file has ecoregion columns
     Analogs <- c(11,15,19,23,27,31,35,39,43,47,51,55,59,63,67,71,75,79,83,87)

  analogCount <- which(Analogs == numberOfCols)
```

```
  return(analogCount)
}getModClimDataFromPollenDB <- function(climVar, filePS, modClim, custClim)
{
  # modernData is matrix, figure out which column of climate data is needed
  # get siteID, lat, lon, selected climate variable and ecoregion columns (5 cols)
  # this is the file that will be joined to the interpolated reconstruction file

  modernData <- NULL
  modClimData <- NULL
  climVarMatch <- NULL
  k=1

  # handle the case of modern pollen file not existing for pollens sum of selected reconstruction file
  tryCatch({
     if (modClim & filePS != 0) # must be new climate variable column, use updated modern file
     {
         modern <- paste("ps",filePS,"_modern_updated.csv",sep="")
         message(paste("Loading custom modern climate data file:",modern,sep=" "))
         modernData <- read.csv(modern)
     }else
     {
        modern <- readPollenDbFile(filePS,csv = MODERN_CSV)
        modernData <- read.csv(modern)
     }

     modClimCol = which(colnames(modernData) == tolower(climVar))
     if (length(modClimCol) == 0)
        modClimCol = which(colnames(modernData) == toupper(climVar))
     ecoRegionCol = which(colnames(modernData) == "ECOREGION")
     # in case different file is used but still has TJUL_NAME
     if (length(modClimCol) == 0)
     {
        while (k <= ncol(modernData))
        {
           if (modClim)
              climVarMatch <- regexpr(custClim,colnames(modernData)[k])
           else
              climVarMatch <- regexpr(custClim,colnames(modernData)[k])

           if (climVarMatch > 0)
           {
              modClimCol <- k
              k = 999
           }
           k = k + 1
        }
      }
     message(paste("Modern climate variable is in column",modClimCol,sep=" "))
     #message("Loading modern climate data from associated modern pollen database file based on reconstruction file....")
     modClimData <- modernData[c(1,10:11,modClimCol,ecoRegionCol)]
     colnames(modClimData)[4] <- "modClimVal"
     write.csv(modClimData,"modClimData.csv",row.names=FALSE)
     return(modClimData)

  }, error = function(e){message("Error reading or writing modern pollen data file for pollen sum of selected reconstruction file!");
        message("Please check that the file exists, is not open and has has the appropriate custom climate variable column!"); return(
        NULL)}, warning = function(w){})

}getTaxaColumns<- function(pollenDB,modern=TRUE)
{
   # determine the ending columns for the modern taxa, for mattools code
   # this depends on which pollen database file is used (35,64,134,135 or custom)
   # NOTE: starting column is always col 14 for both modern & fossil databases
   # this function will work whether pollen sum is 35,64,134,135 or custom

   #taxaRange <- mat.or.vec(2,1)
   lastCol <- ncol(pollenDB)
   if(modern)lastCol <- lastCol-END_COLS
   return(c(14,lastCol))
}### globalsource.r ###
# Last modified: November 12, 2012 by Matthew Ladd

is.installed <- function(mypkg) is.element(mypkg, installed.packages()[,1])

############### Packages to install ###############
```

```
if(!is.installed("MATTOOLS"))install.packages("MATTOOLS")
#if(!is.installed("BerkeleyEarth"))install.packages("BerkeleyEarth")
#if(!is.installed("RghcnV3"))install.packages("RGhcnV3")
if(!is.installed("maps"))install.packages("maps")
if(!is.installed("plyr"))install.packages("plyr")
if(!is.installed("maptools"))install.packages("maptools")
if(!is.installed("RColorBrewer"))install.packages("RColorBrewer")
if(!is.installed("classInt"))install.packages("classInt")
#if(!is.installed("gpclib"))install.packages("gpclib")
if(!is.installed("mapdata"))install.packages("mapdata")
if(!is.installed("sp"))install.packages("sp")
if(!is.installed("spdep"))install.packages("spdep")
if(!is.installed("shapefiles"))install.packages("shapefiles")
if(!is.installed("ncdf"))install.packages("ncdf")
if(!is.installed("proj4"))install.packages("proj4")
if(!is.installed("rgdal"))install.packages("rgdal")
if(!is.installed("ggplot2"))install.packages("ggplot2")
if(!is.installed("automap"))install.packages("automap")

################# Library load ###############
library(MATTOOLS)
#library(BerkeleyEarth)
#library(RghcnV3)
library(maps)
library(plyr)
library(maptools)
library(RColorBrewer)
library(classInt)
#library(gpclib)
library(mapdata)
library(sp)
library(spdep)
library(shapefiles)
library(ncdf)
library(proj4)
library(rgdal)
library(ggplot2)
library(automap)

#### other libraries that might be needed in the future?? ###############
#library(ape)
#library(raster)
#library(rasterVis)
#library(spatial)
#library(GEOmap)
#library(geosphere)
#library(maptools)
#library(mapproj)
#library(RNetCDF)
#library(zoo)

### FILES TO SOURCE ###

source("MainProgram1.R")
source("MainProgram2.R")
source("Constants.R")
source("checkForPS35_54_135files.R")
source("build_35_64_134_pollen_db.R")
source("calcAnomalies.R")
source("calcAvgAnomaliesByEcoregion.R")
source("calcSCDbyEcoregion.R")
source("calcSiteResolution.R")
source("createReconObj.R")
source("createSCDplotsByEcoregion.R")
source("customClimateOption.R")
source("customPollenSum.R")
source("dataModelComparison.R")
source("doLinearInterp.R")
source("doSpatialJoin.R")
source("getAnalogCount.R")
source("getAnalogTotal.R")
source("getModClimDataFromPollenDB.R")
source("getTaxaColumns.R")
source("image.maker.R")
source("joinClimateData.R")
source("listAllFiles.R")
```

```
source("listPollenTaxaInLoadedDB.R")
source("mapPollenSites.R")
source("outputAnomaliesByTimeSlice.R")
source("pickClimateOption.R")
source("plotAvgAnomalies.R")
source("plotAvgAnomaliesByEcoregion.R")
source("plotSCDthroughTime.R")
source("plotSiteDensityThroughTime.R")
source("readPollenDBFile.R")
source("refreshMainMenu1.R")
source("refreshMainMenu2.R")
source("selectClimateVariable.R")
source("selectPollenSum.R")
source("selectReconFile.R")
source("setLinearInterpInterval.R")
source("setNumberOfAnalogs.R")
source("setCalYrRange.R")
source("setWorkingDirectory.R")
source("sortRecon.R")
source("subsetReconBySpatialExtent.R")
source("subsetReconByCalYrRange.R")
source("subsetReconBySiteRes.R")
source("subsetSPjoinReconModBySearchRad.R")
source("spatialFiltering.R")
source("tuneYearAD.R")
source("writeReconObj.R")

################################################# from http://www.mail-archive.com/r-help@r-project.org/msg23971.html
image.maker <- function(coords, value)
{
  N <- length(unique(coords[,1]))
  image.out <- matrix(NA, nrow = N, ncol = N)
  coords[,1] <- as.numeric(factor(coords[,1]))
  coords[,2] <- as.numeric(factor(coords[,2]))
  for (i in 1:nrow(coords))
    image.out[coords[i,1], coords[i,2]] <- value[i]
  return(image.out)
}joinClimateData <- function( modPollenSites, climateDataFile, latitude, longitude, elevation )
{
  require(sp)

  # essentially join modern climate station file to modern pollen sites from DB


  # MODERN CLIMATE OPTION

  # update: if option 3, data is already in csv format
  # if option 2, data will be converted from netcdf to csv
  # if option 1, packages will output the appropriate csv file
  #---------------------------------------------------------------

  # which modern climate data set to use (default is Whitmore)
  # options will be (1) STATION, (2) RE-ANALYSIS
  # at this point both (1) and (2) are in csv format
  # user will need some info about how to use a custom modern climate data set

  # but allow user to use other modern climate data files e.g. BEST, GHCN, reanalysis (with custom option)
  # as long as they are formatted properly and cover all of North America
  # ask user if they wish to use the default modern climate data for each site in the modern pollen db file
  # if yes exit function, else if no then need to read in mod clim file & join to modern pollen db file

  ### user needs to be aware that if they are reconstructing a variable, there needs to be the equivalent
  # in the modern climate data set joined to whitmore modern pollen sites, e.g.
  # if TAVE is being reconstructed, then there should be TAVE in modern climate file
  # if MTWA is being reconstructed, then there should be TJUL in modern climate file
  # if MTCO is being reconstructed, then there should be TJAN in modern climate file
  # etc...

  # for some variables, there might not be an equivalent (for example MIPT, which is
  # an alpha parameter of actual evapotranspiration / potential evapotranspiration
  # there is no issue if user keeps the default of using the Whitmore modern climate data
  # at each modern pollen site

  # also when joining there may be a few options:
  # nearest station within X km (with or without lapse rate correction due to elevation difference)
  # weighted average of nearest stations within X km
```

```
# other methods for the spatial join ???

spJoinMethod = 1 # user GC distance by default
while (is.na(spJoinMethod) | spJoinMethod < 1 | spJoinMethod > 2)
   spJoinMethod <- as.numeric(readline("Which method of spatial join do you wish to use? <1 for Great Circle distance, 2 for
         Euclidean> "))

# we already have the modern DB file loaded so get a subset of the data
# lat, lon, elevation, mod climate columns, and fedorova column
# use this file in the spatial join of modern to fossil sites

# parameters are A, B, C, D, E

#do spatial join between between fossil and modern (closest within specified distance)

# adapted from http://www.nceas.ucsb.edu/scicomp/usecases/AssignClosestPointsToPoints

# Latest version: Assign closest points from a second point set

# temperaturePoints <- read.csv("temperaturePoints.csv")
# densityPoints <- read.csv("densityPoints.csv")

# promote the input lists to SpatialPointsDataFrames

coordinates(climateDataFile) <- c(longitude, latitude)
coordinates(modPollenSites) <- c("LONDD", "LATDD")

# Remove temparature points with 'invalid' value of -199

#validTempPoints <- temperaturePoints[!temperaturePoints$temperature %in% c(-199),]

# Define these vectors, used in the loop.

closestSiteVec <- vector(mode = "numeric",length = nrow(modPollenSites))
minDistVec <- vector(mode = "numeric",length = nrow(modPollenSites))

# Get the vector index of the temperature station closest to each field station.
# Use the spDistsN1 function to compute the distance vector between each
# field station site and all of the temperature stations. Then, find and
# retain the actual temperature, and the index of the closest temperature
# to each transect station.
#
# spDistsN1 usage: spDistsN1(pointList, pointToMatch, longlat)
#
# where:
# pointList : List of candidate points.
# pointToMatch: Single point for which we seek the closest point in pointList.
# longlat : TRUE computes Great Circle distance in km, (BETTER)
# FALSE computes Euclidean distance in units of input geographic coordinates
#
# We use Great Circle distance to increase distance calculation accuracy at high latitudes
# See the discussion of distance units in the header portion of this file
#
# minDistVec stores distance from the closest temperature station to each density measurement point.
# closestSiteVec stores the index of the closest temperature station to each density measurement point.
#

 for (i in 1 : nrow(modPollenSites))
 {
    if (spJoinMethod == 1)
       distVec <- spDistsN1(climateDataFile,modPollenSites[i,],longlat = TRUE)
    else # spJoinMethod == 2
       distVec <- spDistsN1(climateDataFile,modPollenSites[i,],longlat = FALSE)

    minDistVec[i] <- min(distVec)
    closestSiteVec[i] <- which.min(distVec)
 }

#
# Create the Temperature Assignment table: merge the temperature point list with the transect point list
# into a five-column table by merging the temperature point and transect point lists.
#
PointAssignTemps <- as(climateDataFile[closestSiteVec,]$val,"numeric")
if (!is.null(elevation))
{
   PointAssignElev <- as(climateDataFile[closestSiteVec,]$elev,"numeric")
```

```
    FinalDataTable = data.frame(modPollenSites$ID2,coordinates(modPollenSites),modPollenSites$ELEVATION,closestSiteVec,minDistVec,
          PointAssignTemps,PointAssignElev)
  }else
    FinalDataTable = data.frame(modPollenSites$ID2,coordinates(modPollenSites),closestSiteVec,minDistVec,PointAssignTemps)

  #
  # Update the Temperature Assignment table column names
  #
  if (!is.null(elevation))
    names(FinalDataTable) <- c("SiteID","Lon","Lat","SiteElevation","CloseSiteIndex","Distance","ModClimVal","StnElevation")
  else
    names(FinalDataTable) <- c("SiteID","Lon","Lat","CloseSiteIndex","Distance","ModClimVal")

  #
  # And, at the end, write the point assignment file.
  #

  # while (is.na(searchRad) | searchRad < 1 | searchRad > 10)
  # searchRad=as.numeric(readline("What search radius do you wish to use for joining the climate station data to modern pollen sites? (
        must be between 1 and 10 degrees) "))

  # FinalDataTable <- FinalDataTable[FinalDataTable$Distance <= searchRadius,]

  # message("Wrote modern/fossil assignment table to disk in .csv format")

  # print("Please update the modern pollen database with the custom climate values for the variable of interest")
  # print("keeping the column header (e.g. mtwa) as is. Ensure you have a backup of this file with the orginal values!")

  #tryCatch({write.csv(FinalDataTable,file="modern_pollen_new_climate.csv",row.names=FALSE)}, error = function(e) {message("Cannot
        write file modern_pollen_new_climate.csv to disk!")}, warning = function(w){})

  return(FinalDataTable)

  # return(TRUE)
}
listAllFiles <- function(fileOption)
{
   if (fileOption == 1)
   {
     message("Listing all files in working directory with extension .modern, .fossil, and .recon")

     list1 <- list.files(pattern = ".modern")
     if (length(list1) > 0)
        print(list1)
     else
        message("No .modern files found in working directory!")

     list2 <- list.files(pattern = ".fossil")
     if (length(list2) > 0)
        print(list2)
     else
        message("No .fossil files found in working directory!")

     list3 <- list.files(pattern = ".recon")
     if (length(list3) > 0)
        print(list3)
     else
        message("No .recon files found in working directory!")

   }else
   {
     message("Listing all files in working directory with extension .recon, .pdf and .csv....")

     list1 <- list.files(pattern = ".recon")
     if (length(list1) > 0)
        print(list1)
     else
        message("No .recon files found in working directory!")

     list2 <- list.files(pattern = ".pdf")
     if (length(list2) > 0)
        print(list2)
     else
        message("No .pdf files found in working directory!")

     list3 <- list.files(pattern = ".csv")
```

```
    if (length(list3) > 0)
       print(list3)
    else
       message("No .csv files found in working directory!")
   }
}listPollenTaxaInLoadedDB <- function(fosCustDB)
{
  # NOTE that 64 pollen sum is actually 63 as one taxa (Iva) was double counted
  # in the paper by Jack Williams# NOTE that 64 pollen sum is actually 63 as one
  # taxa (Iva) was double counted in the Williams paper
  # fossil DB has same number of taxa as modern DB

  # Also, 35 pollen sum has 36 taxa because of PinusSE
  # 135 and 134 pollen sums have 135 and 134 taxa respectively
  # Default custom pollen sum has 135 taxa
  # User should run this function after creating a custom pollen sum database
  # to make sure the file has the correct taxa and the correct number of taxa

  customPollenSumCount = 0
  lastCol <- ncol(fosCustDB)
  message("---Custom Pollen Taxa List---")
  taxaID=14 # first taxa col is 14
  while (taxaID <= lastCol)
  {
      message(colnames(fosCustDB)[taxaID])
      customPollenSumCount = customPollenSumCount + 1
      taxaID = taxaID + 1
  }
  message("------------------------------")
  message(paste("Total number of taxa: ", customPollenSumCount))
  message("------------------------------")
}MainProgram1 <- function()
{
  ######################################################################
  #
  # Main program 1 with menu for pollen-based climate reconstruction
  # This version replaces TestingFrameworkMenu.r & MainProgram.r
  # Last modified: February 12, 2013 by Matthew Ladd
  #
  ######################################################################


  ### INITIALIZE AND DESCRIBE ALL VARIABLES USED ###

  # lat/lon coordinates of default study area (North America)
  lonMax = -170
  lonMin = -40
  latMin = 15
  latMax = 90

  pSum = 135 # default pollen sum set to 135 taxa
  newPS = -99 # gets the value of the new pollen sum, then pSum is assigned this value, then newPS is reset

  fileList = NULL # list of files from list.files()

  modern = NULL # file name including path to loaded modern pollen database file
  modernPollDB = NULL # this is the 135 taxa modern pollen database file (will always available to be loaded)
  modernData = NULL # selected modern pollen database file that is currently loaded
  modTaxaRange = NULL # the column ranges of the taxa in the modern pollen database file loaded (e.g. 14:149)

  fossil = NULL # file name including path to loaded fossil pollen database file
  fossilPollDB = NULL # this is the 135 taxa fossil pollen database file (will always available to be loaded)
  fossilData = NULL # selected fossil pollen database file that is currently loaded
  fosTaxaRange = NULL # the column ranges of the taxa in the fossil pollen database file loaded (e.g. 14:76)

  foundCustMod = FALSE # indicates whether custom fossil database file found in fileList vector
  foundCustFos = FALSE # indicates whether custom modern database file found in fileList vector

  menuItemInput = "NA" # the menu item input by user before converting it to a numeric value
  menuItem = 0 # determines which menu item 1-25 will be run (is the as.numeric version of menuItemInput

  psCustomDBlist = NULL # the list of objects returned by the function customPollenSum
  psCustomModernDB = NULL # the custom pollen sum (modern) database created in menu item #2
  psCustomFossilDB = NULL # the custom pollen sum (fossil) database created in menu item #2
  psCustomTaxaCount = 0 # count of custom pollen sum taxa in database
  psCount = 0 # number of taxa in custom pollen sum database file
```

```
lastCol = 0 # last column of pollen database file (either fossil or modern)
modColCount = 0
confirmWrite = 0 # ask user whether old custom pollen sum db file will be overwritten (1) yes (2) no

selClimVar = 1 # selected climate variable from modern pollen DB (this is set in menu item #6)

climNameSel = NULL
climName = ""
modOutFile = NULL

modClimOption = FALSE # determines whether we are using a custom climate data set for the modern climate data
climColIndex = 0 # get column index of selected climate variable from modern DB (this is needed to create recon file using mat.dissim
      )
modernDataNew = NULL # modern climate data with selected column containing custom rather than default climate data

analogRange = 0 # the number of analogs to reconstruct, must be between 1 and 20

modCustom = FALSE
climNameNew = NULL
modernDataNewList = NULL

NA_recons_SCD = NULL # the output reconstruction dissimilarity object by mat.dissim in menu item #8
NA_recons_Obj = NULL # the formatted recon object that will be written to csv in menu item #8

### END OF VARIABLE LIST ###

refreshMainMenu1() # call the function to display the main menu #1
fileList <- list.files() # vector with list of files in working directory

# check for pollen sum 135 fossil and modern database files
if (!("ps135perc.modern" %in% fileList) | !("ps135perc.fossil" %in% fileList))
{
  message("No pollen sum 135 files found in working directory....exiting!")

}else
{

  # load ps135 modern database file by default
  modern <- readPollenDbFile(pSum,csv = MODERN_CSV)
  modernPollDB <- read.csv(modern)
  colnames(modernPollDB) <- toupper(colnames(modernPollDB))
  modernData <- modernPollDB # make a copy to be available to custom pollen sum function
  message("Modern pollen sum ",pSum," database is loaded!")
  modTaxaRange <- getTaxaColumns(modernPollDB,modern=TRUE)
  climColIndex <- which(colnames(modernData) == "TJAN")

  # load ps135 fossil database file by default
  fossil <- readPollenDbFile(pSum,csv= FOSSIL_CSV)
  fossilPollDB <- read.csv(fossil)
  colnames(fossilPollDB) <- toupper(colnames(fossilPollDB))
  fossilData <- fossilPollDB # make a copy to be available to custom pollen sum function
  message("Fossil pollen sum ",pSum," database is loaded!")
  fosTaxaRange <- getTaxaColumns(fossilPollDB,modern=FALSE)

  # check for custom pollen sum modern db file
  foundCustMod <- "psCustomPerc.modern" %in% fileList
  if(!foundCustMod) # write default custom modern pollen DB file
  {
     tryCatch({write.csv(modernPollDB,"psCustomPerc.modern",row.names=FALSE)}, error = function(e) {message("Cannot write psCustomPerc
           .modern to disk, file may be open!")}, warning = function(w){})
     message("Custom modern pollen sum database file created with 135 (default) taxa!")
  }

  # check for custom pollen sum fossil db file
  foundCustFos <- "psCustomPerc.fossil" %in% fileList
  if(!foundCustFos) # write default custom fossil pollen DB file
  {
     tryCatch({write.csv(fossilPollDB,"psCustomPerc.fossil",row.names=FALSE)}, error = function(e) {message("Cannot write psCustomPerc
           .fossil to disk, file may be open!")}, warning = function(w){})
     message("Custom fossil pollen sum database file created with 135 (default) taxa!")
  }

  # check for 35, 64 and 134 pollen database files before creating them
  checkForPS35_54_135files(modernPollDB,fossilPollDB)
```

```
while (!is.numeric(menuItemInput) | menuItem < 1 | menuItem > 13) # option 13 quits program
{
   menuItemInput <- readline("PLEASE SELECT A MENU ITEM FROM ABOVE: ")

   tryCatch({menuItem <- as.numeric(menuItemInput)}, error = function(e){}, warning = function(w){})

   # get input from user (select menu item from list)
   # check for NA values!!

   if (!is.na(menuItem))
   {
      if (menuItem == 1) # need to allow for aggregation of taxa (at least 10 taxa needed) !!
      {
          psCustomDBlist <- customPollenSum(modernPollDB,fossilPollDB,params=PS135_TAXA_LIST)

          psCustomModernDB <- psCustomDBlist[[1]]
          psCustomFossilDB <- psCustomDBlist[[2]]
          psCustomTaxaCount <- psCustomDBlist[[3]]

          if (!is.null(psCustomModernDB) & !is.null(psCustomFossilDB) & pSum == 135)
          {
             if (psCustomTaxaCount >= 5)
             {
               while (is.na(confirmWrite) | confirmWrite < 1 | confirmWrite > 2)
               {
                  confirmWrite <- as.numeric(readline("Existing custom pollen sum files in working directory will be overwritten, do
                        you wish to continue? <1 for yes, 2 for no> "))
               }
               if(confirmWrite == 1)
               {

                   modern <- "psCustomPerc.modern"
                   fossil <- "psCustomPerc.fossil"

                   message(paste("Writing files ",modern," & ",fossil," to working directory..."))

                   tryCatch({write.csv(psCustomModernDB,modern,row.names=FALSE)}, error = function(e){message("Custom modern file
                         cannot be written to disk, please close the file!")}, warning = function(w){})
                   tryCatch({write.csv(psCustomFossilDB,fossil,row.names=FALSE)}, error = function(e){message("Custom fossil file
                         cannot be written to disk, please close the file!")}, warning = function(w){})


                }else
                {
                  message("Please move or rename the existing custom pollen sum database file before running menu item 1 again!")
                }
              }else
              {
                 message("You need to have at least 5 taxa in your custom pollen sum database! Custom modern & fossil files not
                       written to disk.")
              }
           }
           refreshMainMenu1()
       }else if (menuItem == 2)
       {
         while (is.na(newPS) | newPS != 35 & newPS != 64 & newPS != 134 & newPS != 135 & newPS != 0)
            newPS=as.numeric(readline("Select either 35, 64, 134, 135 or 0 for custom: "))
         pSum = newPS
         newPS = -1 # reset this

         tryCatch({
            modern <- readPollenDbFile(pSum,csv = MODERN_CSV)
            modernData <- read.csv(modern)
            modTaxaRange <- getTaxaColumns(modernData,modern=TRUE)
         }, error = function(e){message("Cannot open custom modern pollen sum database file!")}, warning = function(w){})

         tryCatch({
           fossil <- readPollenDbFile(pSum,csv= FOSSIL_CSV)
           fossilData <- read.csv(fossil)
           fosTaxaRange <- getTaxaColumns(pSum,modern=FALSE)
         }, error = function(e){message("Cannot open custom fossil pollen sum database file!")}, warning = function(w){})

         if (pSum == 0)
         {
           lastCol <- ncol(fossilData)
           psCount = (lastCol - 14) + 1
```

```
    message(paste("Modern pollen DB ( custom pollen sum [",psCount,"]) is loaded"))
    message(paste("Fossil pollen DB ( pollen pollen sum [",psCount,"]) is loaded"))
  }else
  {
    message(paste("Modern pollen DB ( pollen sum [",pSum,"]) is loaded"))
    message(paste("Fossil pollen DB ( pollen sum [",pSum,"]) is loaded"))
  }
  refreshMainMenu1()
}else if (menuItem == 3)
{
  message(paste("Modern pollen database selected is ", modern))
  message(paste("Fossil pollen database selected is ", fossil))

  refreshMainMenu1()
}else if (menuItem == 4)
{
  # using fossil here because last taxa col is last col of the db
  listPollenTaxaInLoadedDB(fossilData)

  refreshMainMenu1()
}else if (menuItem == 5) # this menu item is in development
{
  # user can append
  if (is.null(selClimVar))
    message("No climate variable selected! (Run menu item 6)")
  else
  {

    modClimOption <- pickClimateOption()

    climNameSel <- toupper(CLIMATE_VARIABLES[selClimVar,2])

    modernDataNewList <- customClimateOption(modernData,climNameSel,modClimOption,lonMin,lonMax,latMin,latMax)
    modernDataNew <- modernDataNewList[[1]]
    climName <- modernDataNewList[[2]]
    climNameNew <- modernDataNewList[[3]]

    # will allow the user to join a subset of other custom climate data to the
    # modern pollen sites (e.g. station, re-analysis, raster, or other data set)

    if (!is.null(modernDataNew))
    {
      # modernData <- modernDataNew
      modOutFile <- paste("ps",pSum,"_modern_updated.csv",sep="")
      tryCatch({write.csv(modernDataNew,modOutFile,row.names=FALSE)}, error = function(e) {message(paste("Cannot write file",
            modOutFile,"to disk, file may be open!",sep=" "))}, warning = function(w){})
      message(paste("Wrote new modern pollen file",modOutFile,"with updated climate information to disk!",sep=" "))
    }

  }
  refreshMainMenu1()
}else if (menuItem == 6)
{

  # set climate variable to be reconstructed
  selClimVar <- selectClimateVariable(params=CLIMATE_VARIABLES)

  # if OTHER is selected inform user to add new climate variable by running menu item #5
  if (selClimVar == 70)
  {
    climName = "CUSTOM"

    message("You need to run the custom modern climate feature (menu item 5) before doing the reconstruction!")
    message(paste("NEW CLIMATE VARIABLE selected is",climName,sep=" "))
  }
  else
  {
    # get column index of selected climate data from modern DB selected
    climName = toupper(CLIMATE_VARIABLES[selClimVar,2])
    climColIndex = which(colnames(modernData) == climName)
  }
  refreshMainMenu1()
}else if (menuItem == 7)
{
  # default climate variable selected is TJAN - check for new mod climate var here
  if(!is.null(modernDataNew))
```

```
    message(paste("Climate variable selected is: ",climNameNew,sep=" "))
  else
    message(paste("Climate variable selected is ",CLIMATE_VARIABLES[selClimVar,3]," (",toupper(CLIMATE_VARIABLES[selClimVar
        ,2]),")"))
  refreshMainMenu1()
}else if (menuItem == 8) # this is where we run mat.dissim from MATTOOLS v1.1
{

  if (!is.null(modOutFile))
  {
    require(MATTOOLS)

    loadCustMod = 0

    if(!is.null(modernDataNew))
    {
     modernData <- modernDataNew
     climColIndex <- which(colnames(modernData) == climNameNew)

     if (colnames(modernDataNew)[ncol(modernDataNew)] != "ECOREGION") # appended col
       modCustom = TRUE
    }

    message("Custom modern climate data loaded. To re-load the original modern climate data simply run menu item 2")

  }
  if (climName == "CUSTOM") # if user is not loading custom climate data, reset this
  {
    climName = "TJAN" # otherwise it will be set to the selected climate variable
    climColIndex <- which(colnames(modernData) == climName)
  }

  if (climColIndex == 161)
    climName = "TJAN"

  if (!is.null(climNameNew))

    climName <- climNameNew

    pSumVec <- mat.or.vec(1,4)
    pSumVec[1] <- 35
    pSumVec[2] <- 64
    pSumVec[3] <- 134
    pSumVec[4] <- 135


  if (!pSum %in% pSumVec) # get modern & fossil taxa ranges for custom pollen sum file
  {
    modRangeCust2 <- which(colnames(modernData) == "KGDISCR") - 1
    modRangeCust1 <- which(colnames(modernData) == "POLLSUM") + 1

    fosRangeCust2 <- lastCol
    fosRangeCust1 <- which(colnames(fossilData) == "TOTAL.POLLEN.SUM") + 1
  }

  if (climColIndex < ncol(modernData)) # quick check to see if pollen db changed before doing recon
  {
    message("")
    message(paste(climName,"will be reconstructed ( column index is",climColIndex,")",sep=" "))
    message(paste("Loaded pollen sum database for fossil and modern is:",pSum,sep=" "))
    message("")
    message("Keep in mind that a higher number of analogs takes longer to process")
    analogRange <- getAnalogCount(range= 2:20)

    tryCatch({

      message("Building reconstruction file using MATTOOLS v1.1, this could take some time......")

      if (!pSum %in% pSumVec)
      {
        paste("NA_recons_SCD=mat.dissim(inFossil=fossilData,inModern=modernData,llMod=10:11,modTaxa=",modRangeCust1,":",
              modRangeCust2,",llFoss=4:5,fosTaxa=",fosRangeCust1,":",fosRangeCust2,",numAnalogs=analogRange,counts=COUNTS,
              dist.method=DISTANCE_METHOD)",sep="")
      }else # pSum is 35, 64 134 or 135
      {
       # run mat.dissim to return dissim object, which is then passed to createReconObj function
```

```
            if (modTaxaRange[2] == 49)
              NA_recons_SCD=mat.dissim(inFossil=fossilData,inModern=modernData,llMod=10:11,modTaxa=14:49,llFoss=4:5,fosTaxa
                    =14:49,numAnalogs=analogRange,counts=COUNTS,dist.method=DISTANCE_METHOD)
            else if (modTaxaRange[2] == 76)
              NA_recons_SCD=mat.dissim(inFossil=fossilData,inModern=modernData,llMod=10:11,modTaxa=14:76,llFoss=4:5,fosTaxa
                    =14:76,numAnalogs=analogRange,counts=COUNTS,dist.method=DISTANCE_METHOD)
            else if (modTaxaRange[2] == 147)
              NA_recons_SCD=mat.dissim(inFossil=fossilData,inModern=modernData,llMod=10:11,modTaxa=14:147,llFoss=4:5,fosTaxa
                    =14:147,numAnalogs=analogRange,counts=COUNTS,dist.method=DISTANCE_METHOD)
            else # modTaxaRange[2] == 148
              NA_recons_SCD=mat.dissim(inFossil=fossilData,inModern=modernData,llMod=10:11,modTaxa=14:148,llFoss=4:5,fosTaxa
                    =14:148,numAnalogs=analogRange,counts=COUNTS,dist.method=DISTANCE_METHOD)
          }

          message("Finished running mat.dissim(), now building reconstruction object....")

          # create ReconObj function here
          NA_recons_Obj <- createReconObj(NA_recons_SCD,modernData,fossilData,climColIndex,climName,modCustom)

          message("Now attempting to write reconstruction object to disk as a .recon file....")

          # write ReconObj function here
          writeReconObj(NA_recons_Obj,climName,pSum,psCount)

        }, error = function(e){message("Error doing reconstruction with MATTOOLS! Possibly your custom pollen sum database
              loaded may be invalid.")}, warning = function(w){})
      }else
        message("NOTE: pollen sum database was changed, you need to re-select the climate variable (menu item 6) before doing
             reconstruction!")

      refreshMainMenu1()

    }else if (menuItem == 9)
    {
      option = 0 # reset this option
      while (is.na(option) | option < 1 | option > 3)
        option=as.numeric(readline("Do you wish to create a map of modern or fossil sites? (enter <1> for fossil, <2> for modern
              , <3> for both) "))
      if (option == 1)
      {
        message("File fossilPollenSiteMap.pdf will be written to working directory!")
        mapPollenSites(NULL,NULL,fossilData,box <- c(lonMax,lonMin,latMin,latMax),NULL)
      }else if (option == 2)
      {
        message("File modernPollenSiteMap.pdf will be written to working directory!")
        mapPollenSites(NULL,modernData,NULL,box <- c(lonMax,lonMin,latMin,latMax),NULL)
      }else
      {
        message("File pollenSiteMap.pdf will be written to working directory!")
        mapPollenSites(NULL,modernData,fossilData,box <- c(lonMax,lonMin,latMin,latMax),NULL)
      }

      refreshMainMenu1()
    }else if (menuItem == 10)
    {
      # list all program files in working directory (.modern, .fossil, .recon)
      listAllFiles(1)
      refreshMainMenu1()
    }else if (menuItem == 11)
    {
      # check here to make sure that specified working dir is valid!
      setWorkingDirectory()
      refreshMainMenu1()

    }else if (menuItem == 12)
    {
      message("Starting reconstruction analysis program....")
      MainProgram2()
      break
    }else if (menuItem == 13)
    {
      message("Exiting program, returning to the R command prompt....")
      break # get out of while loop to exit the program
    }
  }
}
```

```
  }
}
#################### END OF MAIN PROGRAM 1 ###########################################################
MainProgram2 <- function()
{
  ######################################################################
  #
  # Main program 2 with menu for pollen-based climate reconstruction
  # This version replaces TestingFrameworkMenu.r & MainProgram.r
  # Last modified: February 22, 2013 by Matthew Ladd
  #
  ######################################################################

  ### INITIALIZE AND DESCRIBE ALL VARIABLES USED ###
  # lat/lon coordinates of default study area (North America)
  lonMax = -170
  lonMin = -40
  latMin = 15
  latMax = 90

  strMessage = "" # file name of reconstruction file loaded in menu item 1
  menuItemInput = "NA" # the menu item input by user before converting it to a numeric value
  linkReconInput = "NA" # path & filename of reconstruction (.recon) file loaded in menu item #1
  selClimVar = "NA" # selected climate variable from modern pollen DB (this is set in menu item #1)

  menuItem = 0 # determines which menu item 1-25 will be run (is the as.numeric version of menuItemInput
  countSites = 0 # number of unique sites in the reconstruction file
  siteRes = 0 # resolution in years for fossil pollen sites
  startTime = 0 # starting system time for a process

  tuned = FALSE # determines whether cal yr BP values for x-axis of avg anomaly by cal yr BP plot will be tuned
  namSubset = FALSE # determines whether we are using a subset of North America for analysis of recon file

  VarTS = NULL # this is the avg anomaly (for all sites) by time slice data that will be plotted in menu item #12
  modClimData = NULL # the modern climate data that will be spatially joined to the reconstruction (siteID, lat, lon, value, ecoregion

  option = 0 # determines whether we plot modern (1), fossil (2) or both (3) sites from the cb
  spJoinMethod = 0 # determines the spatial join method used (1) for great circle, (2) for euclidean
  useAllSites = 0 # allows user to select all sites (1) or use a subset based on search radius (2)
  keepCalYrBPSubset = 0 # keep temporal subset of reconstruction <1 for yes> <2 for no>
  keepPollenResSubset = 0 # keep subset by pollen site resolution of reconstruction <1 for yes> <2 for no>
  keepSpatialSubset = 0 # keep spatial subset of reconstruction <1 for yes> <2 for no>
  changeAnalogSubset = 0 # keep analog subset of reconstruction <1 for yes> <2 for no>
  studyAreaSubset = 0 # determines whether to use a subset of North America for the analysis of the reconstruction file (1 for yes, 2
        for no)
  tuneYesNo = 0 # determines whether tuned dates are used for the avg anomaly by cal yr BP plot (1 for yes, 2 for no)
  searchRad = 0 # this is the search radius for joining the reconstruction to modern pollen sites to calculate the anomalies
  # the valid range is either 1 to 10 degrees for Euclidean or 100-1000km for Great Circle distance

  reconSelAnalogSCDcol = NULL # for plotting SCD in menu item 14, this would be col X1, X2, etc.
  reconSelAnalogCalYRcol = NULL # for plotting SCD in menu item 14, this would be col with cal yr BP
  SCDbyEcoregion = NULL # the output from calcSCDbyEcoregion

  analogTop = 0 # this is the top analog for the analysis of the reconstruction file (e.g. 1)
  analogBottom = 0 #this is the top analog for the analysis of the reconstruction file (e.g. 5)
  # if analogBottom > analogTop then these analogs will be averaged
  numAnalogs = 0 # the difference between analogBottom and analogTop + 1,
  # if > 1 it will determines the number of analog columns to average
  analogCount = 0 # number of analogs determined from number of columns in reconDataFile
  analogColUsed1 = 0 # this is the top analog for the analysis of the reconstruction file (e.g. 1)
  analogColUsed2 = 0 # this is the bottom analog for the analysis of the reconstruction file (e.g. 5)
  modClimCol = 0 # when joining the avg anomalies by cal yr BP to modern sites, need to know which column has the data for the climate
  # variable of interest in the modern pollen database file
  ecoRegionCol = 0 # when joining the avg anomalies by cal yr BP to modern sites, need to know which column has the data for the
        ecoregion
  # in the modern pollen database file
  siteIDcol = 1 # column number of the siteID field in the reconstruction file (always col 1)

  startYr = 0 # youngest cal yr BP for plot (e.g. 0 BP or 1950 AD)
  endYr = 0 # oldest cal yr BP for plot (this is specified by the user)

  oldCalYr = NULL # this is the oldest cal yr BP for the analysis of the recon file (valid range is -99 to 12000 bp)
  youngCalYr = NULL # this is the youngest cal yr BP for the analysis of the recon file (valid range is -100 to 11099 bp)
  yearInt = NULL # interpolation interval in years specified by user (must be 10 year intervals from 10 to 100)
  anomByCalYr = NULL # anomalies by cal yr BP returned by the function calcAnomalies in menu item #8
  tunedAnomByCalYr = NULL # the result of tuning the list of mean dates for avg anomaly by Cal Yr BP plot in menu item #9
```

```
reconDataFile = NULL # this is the matrix of the reconstruction data file (.recon) loaded in menu item #1
reconSorted = NULL # this is the reconstruction data for selected analog (or avg of analogs) that will be used in the linear interp
reconInterp = NULL # this is the linearly interpolated subset of reconstruction data created in menu item #7
reconSPjoin = NULL # the result of the spatial join for menu item #8 joining the reconstruction to modern climate data to get anomaly
reconSPjoinSubset = NULL # the subset of reconSpJoinFinal determined by search radius entered (assuming useAllSites is no)
reconSpJoinFinal = NULL # the joined reconstruction with calculated anomalies the result of running the function calcAnomalies
reconTS = NULL # this is the avg anomaly (for all sites) by time slice data (set in menu item #11, needed in #18)

anomByCalYrList = NULL # list of objects returned by the function calcAnomalies in menu item #8
calYrList = NULL # list of objects returned by setCalYrRange()
myList = NULL # in menu item #1, selectReconFile() returns a list of objects, this is the list
reconTSList = NULL # the list of objects returned by the outputAnomaliesByTimeSlice function
SCDlist = NULL # the list of objects returned by the createSCDplotsByEcoregion function
tunedAnomByCalYrList = NULL # the list of objects returned by the function tuneYearAD

# new variables added
reconSpFilterOut = NULL
linkSpFilterData = NULL
reconInterpRange = NULL
reconInterpFinal = NULL
reconInterp2 = NULL
reconDataFile2 = NULL
spFilteredRecon = NULL
custMod = FALSE
plotReconSites = 0
calYrInt = -1


### END OF VARIABLE LIST ###

refreshMainMenu2() # call the function to display the main menu

while (!is.numeric(menuItemInput) | menuItem < 1 | menuItem > 20) # option 20 quits the program
{
  menuItemInput <- readline("PLEASE SELECT A MENU ITEM FROM ABOVE: ")

    tryCatch({menuItem <- as.numeric(menuItemInput)}, error = function(e){}, warning = function(w){})

    # get input from user (select menu item from list)
    # check for NA values!!

    if (!is.na(menuItem))
    {
      if (menuItem == 1)
      {

        ### RESET ALL RELEVANT VARIABLES HERE ###
        if (!is.null(myList))
        {
           message("Resetting all variables so that a new reconstruction can be loaded....")

           lonMax = -170
           lonMin = -40
           latMin = 15
           latMax = 90

           menuItem = 0
           menuItemInput = "NA"

           linkReconInput = ""
           strMessage = ""
           selClimVar = "NA"

           countSites = 0
           siteRes = 0
           startTime = 0

           tuned = FALSE
           namSubset = FALSE

           VarTS = NULL
           modClimData = NULL

           option = 0
           spJoinMethod = 0
```

```
   useAllSites = 0
   keepCalYrBPSubset = 0
   keepPollenResSubset = 0
   keepSpatialSubset = 0
   changeAnalogSubset = 0
   studyAreaSubset = 0
   tuneYesNo = 0
   searchRad = 0

   reconSelAnalogSCDcol = NULL
   reconSelAnalogCalYRcol = NULL

   analogTop = 0
   analogBottom = 0
   numAnalogs = 0
   analogCount = 0
   analogColUsed1 = 0
   analogColUsed2 = 0
   modClimCol = 0
   ecoRegionCol = 0
   siteIDcol = 1

   startYr = 0
   endYr = 0

   oldCalYr = NULL
   youngCalYr = NULL
   yearInt = NULL
   anomByCalYr = NULL
   tunedAnomByCalYr = NULL

   reconDataFile = NULL
   reconSorted = NULL
   reconInterp = NULL
   reconSPjoin = NULL
   reconSPjoinSubset = NULL
   reconSpJoinFinal = NULL
   reconTS = NULL
   spfilterLIST = NULL

   anomByCalYrList = NULL
   calYrList = NULL
   myList = NULL
   reconTSList = NULL
   SCDlist = NULL
   tunedAnomByCalYrList = NULL
   SCDbyEcoregion = NULL
   reconDataFile2 = NULL
   reconInterpRange = NULL

   reconSpFilterOut = NULL
   linkSpFilterData = NULL
   reconInterpRange = NULL
   reconInterpFinal = NULL
   reconInterp2 = NULL
   reconDataFile2 = NULL
   spFilteredRecon = NULL
   custMod = FALSE
   plotReconSites = 0
   calYrInt = -1
}

myList <- selectReconFile()

if (!is.null(myList[[3]]))
{
  selClimVar <- myList[[1]]
  reconDataFile <- myList[[3]]
  reconDataFile2 <- reconDataFile # keep a copy of the data
  psFromFile <- myList[[5]]

  if (!identical(selClimVar,myList[[6]]))
  {
     selClimVar <- myList[[6]]
     custMod = TRUE
  }
```

```
    modClimData <- getModClimDataFromPollenDB(selClimVar,psFromFile,custMod,myList[[1]])
    #write.csv(modClimData,"modClimData.csv",row.names=FALSE)
  }

  if (!is.null(reconDataFile))
    {

    analogCount <- getAnalogTotal(reconDataFile)

    if (!is.null(myList[[4]]) & !is.null(modClimData))
      message(paste("Total number of analogs in selected reconstruction file: ",analogCount))
  }else
  {
    message("Invalid or no reconstruction file selected! Required file format is .recon.")
  }
  refreshMainMenu2()

}else if (menuItem == 2)
{
  strMessage = myList[[2]] # get file name of reconstruction file loaded in menu item 1
  if (is.null(reconDataFile))
    message("No reconstruction file is loaded! (run menu item 1)")
  else
  {
    message(strMessage)
    if (nrow(reconDataFile) < 30861)
      message("You have a subset of the reconstruction file!")
    message(paste("There are",nrow(reconDataFile),"records in the reconstruction file!"))
    countSites <- unique(reconDataFile[,1])
    message(paste("There are",length(countSites),"sites in the reconstruction file!"))
    if (nrow(reconDataFile) == 0)
    {
      message("Cannot have subset with zero records! Reloading the whole reconstruction....")
      reconDataFile <- reconDataFile2
      namSubset = FALSE
      reconTS = NULL
      oldCalYr = NULL
      youngCalYr = NULL
      reconSorted = NULL
      reconInterp = NULL
      anomByCalYr = NULL
      reconSpJoinFinal = NULL
      reconSelAnalogSCDcol = NULL
      analogColUsed1 = 0
      analogColUsed2 = 0
      numAnalogs = 0
      lonMax = -170
      lonMin = -40
      latMin = 15
      latMax = 90
    }
  }
  refreshMainMenu2()
}else if (menuItem == 3) # spatial subset of reconstruction
{
  # this functionality could be modifed to allow the user to select a study area from the world
  # if for example, the global pollen database is loaded

  if (is.null(reconDataFile))
    message("No reconstruction file is loaded! (run menu item 1)")
  else
  {

  message("The default study area for this program is North America. Changing the study area will affect the analysis of the
        reconstruction")
  message("and furthermore the ecoregion output may not be valid or available depending on the subset of North America selected
        .")

  studyAreaSubset = 0 # reset this

  while (is.na(studyAreaSubset) | studyAreaSubset < 1 | studyAreaSubset > 2)
    studyAreaSubset <- as.numeric(readline("For the reconstruction, would you like to use a subset of North America? <1 for
          yes, 2 for no> "))

  if (studyAreaSubset == 1)
  {
```

```
    # calculate & output spatial site density here (convert lat/lon boundaries to x/y then can figure out area in km2)
    # after each subset, number of unique sites should be listed
    if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90)
    {
      reconDataFileSubsetList <- subsetReconBySpatialExtent(reconDataFile,lonMax,lonMin,latMax,latMin)
      reconDataFile <- reconDataFileSubsetList[[1]]
      latMax <- reconDataFileSubsetList[[2]]
      latMin <- reconDataFileSubsetList[[3]]
      lonMax <- reconDataFileSubsetList[[4]]
      lonMin <- reconDataFileSubsetList[[5]]
      namSubset = TRUE

      if (nrow(reconDataFile) == 0)
      {
        message("Cannot have subset with zero records! Reloading the whole reconstruction....")
        reconDataFile <- reconDataFile2
        reconInterpRange <- NULL
        namSubset = FALSE
        reconTS = NULL
        oldCalYr = NULL
        youngCalYr = NULL
        reconSorted = NULL
        reconInterp = NULL
        anomByCalYr = NULL
        reconSpJoinFinal = NULL
        reconSelAnalogSCDcol = NULL
        analogColUsed1 = 0
        analogColUsed2 = 0
        numAnalogs = 0
        lonMax = -170
        lonMin = -40
        latMin = 15
        latMax = 90
      }
      # write.csv(reconDataFile,"reconDataFileSubset.csv",row.names=FALSE)

    }else # we already have a spatial subset of North America
    {
      keepSpatialSubset = 0 # reset this
      while (is.na(keepSpatialSubset) | (keepSpatialSubset < 1 & namSubset) | (keepSpatialSubset > 2 & namSubset))
        keepSpatialSubset <- as.numeric(readline("You already have a spatial subset of the reconstruction. Do you wish to keep
               this subset <1> or reload the whole reconstruction <2>? "))
      if (keepSpatialSubset == 2)
      {
        reconDataFile <- reconDataFile2
        namSubset = FALSE
        reconTS = NULL
        oldCalYr = NULL
        youngCalYr = NULL
        reconSorted = NULL
        reconInterp = NULL
        anomByCalYr = NULL
        reconSpJoinFinal = NULL
        reconSelAnalogSCDcol = NULL
        analogColUsed1 = 0
        analogColUsed2 = 0
        numAnalogs = 0
        lonMax = -170
        lonMin = -40
        latMin = 15
        latMax = 90
      }
    }
   }
  }
  refreshMainMenu2()
}else if (menuItem == 7)
# get a subset of the reconstruction by cal yr BP range
{
   if (is.null(reconInterp))
     message("No interpolated reconstruction file is loaded! (run menu item 6)")
   else if (oldCalYr < 15000 )
   {
       keepCalYrBPSubset = 0
       while (is.na(keepCalYrBPSubset) | keepCalYrBPSubset < 1 | keepCalYrBPSubset > 2)
```

```
        keepCalYrBPSubset <- as.numeric(readline("You already have a subset of the interpolated reconstruction based on cal yr
              BP range. Do you wish to keep this subset <1> or reload the whole interpolated reconstruction <2> ? "))

       if (keepCalYrBPSubset == 2)
       {
          reconInterpRange = NULL
          reconInterp <- reconInterp2
          oldCalYr = 15000 + (yearInt-1)
          youngCalYr = 0-yearInt
       }
     }else
     {
       calYrList <- setCalYrRange()
       oldCalYr <- calYrList[[1]]
       youngCalYr <- calYrList[[2]]
       reconInterpRange <- subsetReconByCalYrRange(reconInterp,oldCalYr,youngCalYr)
       tryCatch({write.csv(reconInterpRange,"reconInterpSubsetCalYrRange.csv",row.names=FALSE)}, error = function(e) {message("
             Cannot write reconInterpSubsetCalYrRange.csv to disk, file may be open!")}, warning = function(w){})

   }
   refreshMainMenu2()
}else if (menuItem == 4)
# should be able to get a subset of recon file (based on pollen site resolution) here to be used in linear interp
{

   if (is.null(reconDataFile))
     message("No reconstruction file loaded! (Run menu item 1)")
   else
   {
     if (ncol(reconDataFile) != ncol(reconDataFile2))
     {
       while (is.na(keepPollenResSubset) | keepPollenResSubset < 1 | keepPollenResSubset > 2)
         keepPollenResSubset <- as.numeric(readline("You already have a subset of the reconstruction based on pollen resolution
               . Do you wish to keep this subset <1> or reload the whole reconstruction <2> ? "))

       if (keepPollenResSubset == 2)
         reconDataFile <- reconDataFile2 # reload original recon file
     }else # do the subset by fossil pollen site resolution
     {
       siteRes = 0 # reset this
       while (is.na(siteRes) | siteRes < 10 | siteRes > 1000)
         # ask user for max pollen site resolution to subset by
         siteRes=as.numeric(readline("What is the lowest pollen site resolution (largest value in years) you wish to include
               from the reconstruction (valid range is 10-1000 years): "))

       message("Creating the subset by pollen site resolution, this could take a long time depending on the subset used....")

       # calculate site resolution, last column has pollen site resoution
       reconFileWithSiteResCol <- calcSiteResolution(reconDataFile,oldCalYr)
       siteResCol <- ncol(reconFileWithSiteResCol)

       # now do the subsetting by pollen site resoution
       reconDataFile <- subsetReconBySiteRes(reconFileWithSiteResCol,siteRes,siteResCol)

       if (nrow(reconDataFile) == 0)
       {
         message("Cannot have subset with zero records! Reloading the whole reconstruction....")
         reconDataFile <- reconDataFile2
         reconInterpRange <- NULL
         namSubset = FALSE
         reconTS = NULL
         oldCalYr = NULL
         youngCalYr = NULL
         reconSorted = NULL
         reconInterp = NULL
         anomByCalYr = NULL
         reconSpJoinFinal = NULL
         reconSelAnalogSCDcol = NULL
         analogColUsed1 = 0
         analogColUsed2 = 0
         numAnalogs = 0
         lonMax = -170
         lonMin = -40
         latMin = 15
         latMax = 90
       }
```

```
        tryCatch({write.csv(reconDataFile,"reconSubsetSiteResFinal.csv",row.names=FALSE)}, error = function(e) {message("Cannot
              write reconSubsetSiteResFinal.csv to disk, file may be open!")}, warning = function(w){})

      }
   }
   refreshMainMenu2()
}else if (menuItem == 5)
{
  if (is.null(reconDataFile))
     message("No reconstruction file loaded! (Run menu item 1)")
  else if (analogColUsed1 > 0)
  {
     changeAnalogSubset = 0
     while (is.na(changeAnalogSubset) | changeAnalogSubset < 1 | changeAnalogSubset > 2)
     {
        message("You already have a subset of the reconstruction based on analog columns! By changing the selected analog you
              will lose your current subset.")
        changeAnalogSubset <- as.numeric(readline("Would you like to change the analog selected? <1 for yes, 2 for no> "))
     }
     if (changeAnalogSubset == 1)
     {
          reconDataFile <- reconDataFile2
          reconInterpRange <- NULL
          namSubset = FALSE
          reconTS = NULL
          oldCalYr = NULL
          youngCalYr = NULL
          reconSorted = NULL
          reconInterp = NULL
          anomByCalYr = NULL
          reconSpJoinFinal = NULL
          reconSelAnalogSCDcol = NULL
          analogColUsed1 = 0
          analogColUsed2 = 0
          numAnalogs = 0
          lonMax = -170
          lonMin = -40
          latMin = 15
          latMax = 90
     }
  }
  if (length(selClimVar) > 0)
  {

    #write.csv(reconDataFile,"reconDataFile.csv")

    if (analogColUsed1 == 0)
    {
      message("If only one analog is used enter the same analog at the first and second prompts.")
      message("To average the top 3 analogs enter 1 then 3. You cannot average analogs that are")
      message("not adjacent to each other in the reconstruction file [e.g. (1 and 3) or (1,2 and 4) or (3 and 5)]")
    }

    while (is.na(analogColUsed1) | analogColUsed1 < 1 | analogColUsed1 > analogCount)
       analogColUsed1=as.numeric(readline("What is the top analog you wish to use? (valid range is 1 to 20) "))

    while (is.na(analogColUsed2) | analogColUsed2 < 1 | analogColUsed2 > analogCount)
       analogColUsed2=as.numeric(readline("What is the bottom analog you wish to use? For one analog, use the same value as top
             analog (valid range is 1 to 20) "))

    # can get either one analog or an average of multiple adjacent analogs
    reconAnalogSel <- setNumberOfAnalogs(reconDataFile,analogColUsed1,analogColUsed2,selClimVar)

    # order the data by siteID (first col)
    sortCol = 1
    reconSorted <- sortRecon(reconAnalogSel, sortCol)
    tryCatch({write.csv(reconAnalogSel,"reconSelAnalogData.csv",row.names=FALSE)}, error = function(e) {message("Cannot write
          reconSelAnalogData.csv to disk, file may be open!")}, warning = function(w){})

  }
  refreshMainMenu2()
}else if (menuItem == 6)
{

   if (is.null(reconSorted))
      message("Number of analogs not set! (run menu item 5)")
```

```
  else
  {

    yearInt <- setLinearInterpInterval() # ensure that this is set

    # oldCalYr and youngCalYr depend on yearInt
    # e.g. if yearInt is 100, oldCalYr will be 15099, youngCalYr will be -100
    # this is neccessary so that linear interp goes from 0 to 15000 at 100 year intervals

    oldCalYr <- 15000 + (yearInt-1)
    youngCalYr <- 0-yearInt


      message("Performing the linear interpolation for all dates no older than 15000 bp, this could take some time......")
      reconInterp <- doLinearInterp( fossilDataFrame = reconSorted ,
                                          colFirstTaxa = 5,
                                          colLat = 2,
                                          colLon =3,
                                          colCalmean = 4,
                                          colE. = 1,
                                          target.dates = seq(youngCalYr,oldCalYr,yearInt))
      tryCatch({write.csv(reconInterp,"reconInterpData.csv",row.names=FALSE)}, error = function(e){message("reconInterp.csv
            cannot be written to working directory, please close the file!")}, warning = function(w){})
      reconInterp2 <- reconInterp
      plotReconSites = 0
      while (is.na(plotReconSites) | plotReconSites < 1 | plotReconSites > 2)
        plotReconSites <- as.numeric(readline("Do you want to plot reconstruction site distribution by interpolated calendar
              year BP? <1 for YES, 2 for NO> "))

      if (plotReconSites == 1)
      {

        colnames(reconInterp2)[2] <- "LATITUDENORTH"
        colnames(reconInterp2)[3] <- "LONGITUDEEAST"
        calYrInt = -1
        while (is.na(calYrInt) | calYrInt < 0 | calYrInt > 12000)
          calYrInt <- as.numeric(readline("Enter the calendar year BP interval to map (e.g. 1500): "))
        reconInterpFinal <- subset(reconInterp2, reconInterp2[,4] == calYrInt)
        if (nrow(reconInterpFinal) > 0)
          mapPollenSites(calYrInt,NULL,reconInterpFinal,box <- c(lonMax,lonMin,latMin,latMax))
        else
          message("Entered calendar year BP interval does not exist in data file!")
        plotReconSites = 0
       }
  }
  refreshMainMenu2()
}else if (menuItem == 8)
{
  if (is.null(reconInterpRange))
  {
    message("No interpolated by Cal Yr BP range reconstruction file loaded! (Run menu item 7)")
  }else if (is.null(modClimData))
  {
    message("No modern climate data loaded! (Run menu item 1)")
  }else
  {
    spJoinMethod = 0 # reset these
    useAllSites = 0
    searchRad = 0

    while (is.na(spJoinMethod) | spJoinMethod < 1 | spJoinMethod > 2)
      spJoinMethod <- as.numeric(readline("Which method of spatial join do you wish to use? <1 for Great Circle distance, 2
            for Euclidean> "))

    message("Performing the analysis, this could take some time......")
    reconSPjoin <- doSpatialJoin(reconInterpRange,modClimData,spJoinMethod,filename = "SpatialJoinReconModEcoregions.csv")

    if (is.null(reconSPjoin))
       message("Reconstruction file is not interpolated! (Run menu item 6)")
     else
     {
       while (is.na(useAllSites) | useAllSites < 1 | useAllSites > 2)
        useAllSites <- as.numeric(readline("Do you wish to use all sites in the average anomaly by cal yr BP calculation? <1
              for YES, 2 for NO> "))

       if (useAllSites == 2)
```

```
        {
          ### SEARCH RADIUS (latlon=FALSE is Euclidean distance OR latlon=TRUE is Great Circle distance)
          # for joining modern pollen sites to fossil pollen site reconstruction
          # < min dist and too many sites are excluded, > max dist does not make any difference

           if (spJoinMethod == 1) # GC distance so units are in km
           {
              while (is.na(searchRad) | searchRad < 1 | searchRad > 1000)
                 searchRad=as.numeric(readline("Enter search radius for joining the reconstruction to modern sites? (valid range
                        is 1-1000 km, 150 km is recommended) "))
           }else # Euclidean distance so units are decimal degrees
           {
              while (is.na(searchRad) | searchRad < 0.1 | searchRad > 10)
                 searchRad=as.numeric(readline("Enter search radius for joining the reconstruction to modern sites? (valid range
                        is 0.1-10 deg, 2 deg is recommended) "))
           }
           reconSPjoinSubset <- subsetSPjoinReconModBySearchRad(reconSPjoin, searchRad)
        }else
        {
           reconSPjoinSubset <- reconSPjoin
        }

        # get required info from list object
        anomByCalYrList <- calcAnomalies(reconSPjoinSubset,FALSE)
        anomByCalYr <- anomByCalYrList[[1]]
        reconSpJoinFinal <- anomByCalYrList[[2]]
        #tryCatch({write.csv(reconSpJoinFinal,"reconSpJoinFinal.csv",row.names=FALSE)}, error = function(e) {message("Cannot
              write reconSpJoinFinal.csv to disk, file may be open!")}, warning = function(w){})

     }
   }
   spJoinMethod = 0 # reset this
   refreshMainMenu2()
}else if (menuItem == 9)
{
   # ask user if they would like to tune the cal year Bp dates
   tuneYesNo = 0

   if (!is.null(anomByCalYr))
   {
     while (is.na(tuneYesNo) | tuneYesNo < 1 | tuneYesNo > 2)
        tuneYesNo <- as.numeric(readline("Would you like to tune the cal year BP dates for the analysis? <1 for yes, 2 for no>
              "))

     if (tuneYesNo == 1)
     {

        tuned = TRUE
        tunedAnomByCalYrList <- tuneYearAD(anomByCalYr, yearInt, oldCalYr, youngCalYr)
        tunedAnomByCalYr <- tunedAnomByCalYrList[[1]]
        startYr <- tunedAnomByCalYrList[[2]] # e.g 1950 AD
        endYr <- tunedAnomByCalYrList[[3]] # e.g. 350 AD
        plotAvgAnomalies(anomByCalYr, tunedAnomByCalYr, tuned, selClimVar, startYr, endYr)

     }else
     {
        tuned = FALSE
        startYr = youngCalYr
        endYr = oldCalYr # e.g. oldCalYr = 2099, 3099, etc.
        plotAvgAnomalies(anomByCalYr, tunedAnomByCalYr, tuned, selClimVar, startYr, endYr)
     }
   }
   refreshMainMenu2()
}else if (menuItem == 10)
{
   if (is.null(reconSpJoinFinal))
      message("Reconstruction file is not interpolated! (Run menu item 6)")
   else
   {
     tuneYesNo = 0
     while (is.na(tuneYesNo) | tuneYesNo < 1 | tuneYesNo > 2)
        tuneYesNo <- as.numeric(readline("Would you like to tune the cal year BP dates for the analysis? <1 for yes, 2 for no>
              "))

     if (tuneYesNo == 1)
       calcAvgAnomaliesByEcoregion(params=ECOREGIONS, reconSpJoinFinal, TRUE, selClimVar, oldCalYr, youngCalYr, yearInt)
```

```
     else
       calcAvgAnomaliesByEcoregion(params=ECOREGIONS, reconSpJoinFinal, FALSE, selClimVar, oldCalYr, youngCalYr, yearInt)
   }
   refreshMainMenu2()
}else if (menuItem == 11)
{
   if (is.null(oldCalYr))
     message("Oldest calendar year BP is missing! (Run menu item 7)")
   else if (is.null(reconSpJoinFinal))
     message("Reconstruction file with average anomalies is missing! (Run menu item 8)")
   else
   {
     reconTSList <- outputAnomaliesByTimeSlice(reconSpJoinFinal,oldCalYr,youngCalYr,reconSPjoin,lonMax,lonMin,latMax,latMin)
     reconTS = reconTSList[[1]]
     VarTS <- reconTSList[[2]]
   }
   refreshMainMenu2()
}else if (menuItem == 12) # subset by calYrRange
{

   analogTop = analogColUsed1
   analogBottom = analogColUsed2

   numAnalogs = analogColUsed2 - analogColUsed1 + 1

   if (is.null(reconDataFile))
     message("No reconstruction file loaded! (Run menu item 1)")
   else if (numAnalogs == 0)
     message("Total number of analogs is missing! (Run menu item 5)")
   else if (analogTop == 0)
     message("Top analog not specified! (Run menu item 5)")
   else if (analogBottom == 0)
     message("Bottom analog not specified! (Run menu item 5)")
   else
   {
     SCDlist <- calcSCDbyEcoregion(reconDataFile, numAnalogs, analogTop, analogBottom)
     if (!is.null(SCDlist))
     {
       SCDbyEcoregion <- SCDlist[[1]]
       reconSelAnalogSCDcol = SCDlist[[2]]
       reconSelAnalogCalYRcol = SCDlist[[3]]
       createSCDplotsByEcoregion(SCDbyEcoregion)
     }
   }
   refreshMainMenu2()
}else if (menuItem == 13) # plot SCD through time (all regions)
{
   if (is.null(reconSelAnalogSCDcol) | is.null(reconSelAnalogCalYRcol))
   {
     message("You need to run menu item 12 first!")
   }else
   {
     plotSCDthroughTime(reconSelAnalogSCDcol,reconSelAnalogCalYRcol)
   }
   refreshMainMenu2()
}else if (menuItem == 14)
{
   if (is.null(reconInterpRange))
     message("No interpolated reconstruction file loaded! (Run menu item 6)")
   else
     plotSiteDensityThroughTime(reconInterpRange)
   refreshMainMenu2()
}else if (menuItem == 15) ## spatial filtering using Moran's I statistics
{
    fileTypeMatch = -1
    tryCatch({linkSpFilterData <- file.choose()
        fileTypeMatch <- regexpr("SpatialJoinReconModWithAnCalc",basename(linkSpFilterData))
        }, error = function(e) {}, warning = function(w){})

    if (!is.null(linkSpFilterData) & fileTypeMatch != -1)
    {
      reconSpFilter <- read.csv(linkSpFilterData)
      message("Make sure you are not spatial filtering an anomaly file that is already spatially filtered in menu item 8!")
      spfilterLIST <- spatialFiltering(reconSpFilter)
    }
    refreshMainMenu2()
```

```
      }else if (menuItem == 16) ### functionality in development (beta-version)
      {
        message("Data-model comparison feature will enable the user to do a comparison between a reconstruction created by this
              program")
        message("and paleoclimate model output imported in NetCDF (.nc) format. NOTE: This functionality is not available yet!")

        #message("Please select a paleoclimate model output file in netCDF format (.nc).....")
        #if (is.null(reconTS))
        # message("No reconstruction time slice file loaded! (Run menu item 11)")
        #else
        # dataModelComparison(reconTS)

        refreshMainMenu2()
      }else if (menuItem == 17)
      {
        # check here to make sure that specified working directory is valid!
        setWorkingDirectory()
        refreshMainMenu2()

      }else if (menuItem == 18)
      {
        # list all program files in working directory (.recon, .pdf, .csv)
        listAllFiles(2)
        refreshMainMenu2()
      }else if (menuItem == 19)
      {
        message("Starting pollen database editor program....")
        MainProgram1()
        break
      }else if (menuItem == 20)
      {
        message("Exiting program, returning to the R command prompt....")
        break # get out of while loop to exit the program
      }
    }
  }
}
#################### END OF MAIN PROGRAM ################################################################
mapPollenSites <- function(calYrVal,modernDB,fossilDB,bBox,border=5)
{
  require(maps)

  fileStr = ""
  fileStr2 = ""

  # if no file is passed we sense that and determine what to plot
  # if both are passed we sense that
  # if no extent is passed we use the data to limit the plot
  # we add a "bump" or border if we are looking at data


  stopifnot(!all(is.null(modernDB),is.null(fossilDB)))
  # option to plot either fossil(1) or modern(2) sites
  if(!is.null(fossilDB))option = 1
  if(!is.null(modernDB))option = 2
  if(!is.null(fossilDB) & !is.null(modernDB))option = 3

  # check to make sure file is not already open before writing it to disk
  if (!is.null(modernDB))
  {
    tryCatch({pdf("modernPollenSiteMap.pdf")},
    error = function(e)
    {
      message("File modernPollenSiteMap.pdf cannot be written to disk, please close the file!")
    }, warning = function(w){})
  }

  if (!is.null(fossilDB))
  {
    if (is.null(calYrVal))
    {
      tryCatch({pdf("fossilPollenSiteMap.pdf")},
      error = function(e)
      {
        message("File fossilPollenSiteMap.pdf cannot be written to disk, please close the file!")
      }, warning = function(w){})
```

```
   }else
   {
     fileStr = paste("ReconstructionSiteMap",calYrVal,".pdf")
     fileStr2 <- gsub(" ","", fileStr ,fixed=TRUE)
     tryCatch({pdf(fileStr2)},
     error = function(e)
     {
        message(paste("File",fileStr2,"cannot be written to disk, please close the file!"))
     }, warning = function(w){})
   }
}

if(!is.null(fossilDB) & !is.null(modernDB))
{
   tryCatch({pdf("PollenSiteMap.pdf")},
   error = function(e)
   {
      message("File pollenSiteMap.pdf cannot be written to disk, please close the file!")
   }, warning = function(w){})
}

# option 1 - plot fossil pollen sites
# option 2 - plot modern pollen sites
# option 3 - plot both

lonLimits <- c(-180,180)
latLimits <- c(-90,90)
if(is.null(bBox)){
  if(option == 1 ){
    lonLimits <- c(min(fossilDB$LONGITUDEEAST,na.rm=T)-border,max(fossilDB$LONGITUDEEAST,na.rm=T)+border)
    latLimits <- c(min(fossilDB$LATITUDENORTH,na.rm=T)-border,max(fossilDB$LATITUDENORTH,na.rm=T)+border)
    }
  if(option == 2 ){
    lonLimits <- c(min(modernDB$LONDD,na.rm=T)-border,max(modernDB$LONDD,na.rm=T)+border)
    latLimits <- c(min(modernDB$LATDD,na.rm=T)-border,max(modernDB$LATDD,na.rm=T)+border)
  }
  if(option == 3 ){
    lonLimits <- c(min(fossilDB$LONGITUDEEAST,modernDB$LONDD,na.rm=T)-border,max(fossilDB$LONGITUDEEAST,modernDB$LONDD,na.rm=T)+
          border)
    latLimits <- c(min(fossilDB$LATITUDENORTH,modernDB$LATDD,na.rm=T)-border,max(fossilDB$LATITUDENORTH,modernDB$LATDD,na.rm=T)+
          border)
  }
}else {
  if(class(bBox)=="numeric")bBox <- extent(bBox)
  lonLimits <-c(xmin(bBox),xmax(bBox))
  latLimits <-c(ymin(bBox),ymax(bBox))
}

# set boundary
if(lonLimits[1] < -180)lonLimits[1]<- -180
if(lonLimits[2] > 180)lonLimits[2]<- 180
if(latLimits[1] < -90)lonLimits[1]<- -90
if(latLimits[2] > 90)lonLimits[2]<- 90
map(database = "world",xlim = lonLimits,ylim = latLimits)
map.axes()

if (option == 2 | option == 3)
  points(modernDB$LONDD,modernDB$LATDD,pch=2,col="blue",cex=0.5)
if (option == 1 | option == 3){
  if(option ==1)myCex=.5
  if(option ==3)myCex= 1
  points(fossilDB$LONGITUDEEAST,fossilDB$LATITUDENORTH,pch=21,col="darkred",cex=myCex)

}

### drawing other stuff

legend("bottomleft",legend=c("Fossil","Modern"),pch =c(21,2),col=c("darkred","blue"))

if(option == 1 | option == 3){
  if (is.null(calYrVal))
     mtext(paste("There Are ",length(unique(fossilDB[,1]))," Fossil sites",sep=" "), side = 3, adj = 0, line =0,col = "red")
  else
     mtext(paste("There Are ",length(unique(fossilDB[,1]))," Reconstruction sites at ", calYrVal, "BP",sep=" "), side = 3, adj = 0,
           line =0,col = "red")
}
```

```
  if(option == 2 | option == 3){
    mtext(paste("There Are ",length(unique(modernDB[,1])),"  Modern sites",sep=" "),
          side = 3, adj = 0, line =option-2,col = "blue")
  }

  dev.off()

}
outputAnomaliesByTimeSlice <- function( spatialJoinFile,oldCalYr,youngCalYR,reconFileJoin,lonMax,lonMin,latMax,latMin )
{
   # output a time slice of anomalies (useful for mapping in GIS software)
   # also creates a basic map of the anomalies & outputs a csv file
   # user needs to specify the range of calYrBP dates (e.g. 800 and 1200)
   # get anomalies for timeslice (e.g. cal YR BP between 800 & 1200)
   # can now easily get anomalies (timeslice - modern)

   # ask user for subset (time slice - can either be a single interval such as 1000 bp
   # or the avg of several intervals such as 800 to 1200bp) of avg anomaly data
   # next, do they want to map it (grid or contour) OR
   # load a second avg anomaly file (another time slice)
   # in csv format to compare with (option to map it)

   # finally, need to ask user about dynamic mapping for all time intervals
   # here we would output a series of maps in pdf format, procedure would be:
   # get subset of 2000bp and 1900bp, calc diff between them & then map
   # repeat for 1900bp and 1800bp, etc...to 100 and 0bp

   ### so the 3 options are with anomaly by time slice:
   # 1. map a single time slice relative to modern (e.g. 1000bp or 800-1200 bp or 100-600bp)
   # 2. map the difference between 2 time slices (e.g. 800-1200 from 200-600 bp or 1000bp from 100-600bp)
   # 3. map the difference between sequential intervals for the cal year bp range of the reconstruction
   # e.g. 2000-1900bp, 1900-1800bp, 1800-1700bp etc...to 100-0 bp


   require(shapefiles)
   require(maps)
   require(rgdal)
   require(automap)
   require(ggplot2)

   intervalStart = -99
   intervalStop = -99

   rSel = 1
   interpMapYN = 0
   dynMapYN = 0
   outputSHP = 0
   timeSliceDiffYN = 0
   fileTypeMatch = 0
   mapType = 0
   zLimVal = 0
   gCellRC = 0
   levCol = 0
   calTSavg = 0

   plotTitle = NULL
   fileName = NULL
   fileNameOut = NULL
   diffMat1 = NULL
   diffMat2 = NULL
   siteMatch = NULL
   selAvgAnTS = NULL
   my.image = NULL

   mapOK = TRUE

   #write.csv(spatialJoinFile,"spatialJoinFile.csv",row.names=FALSE)
   #write.csv(reconFileJoin,"reconFileJoin.csv",row.names=FALSE)

   message(paste("Your data has a range of",youngCalYR,"to",oldCalYr,"calendar years BP"))
   message("Example of time slice anomaly (cal yr bp interval): 800 (youngest) to 1200 (oldest) or 1000 (youngest) to 1000 (oldest) ")

   while (is.na(intervalStart) | intervalStart < youngCalYR | intervalStart >= oldCalYr | intervalStart < 0)
      intervalStart=as.numeric(readline("What is the start interval (youngest) of the time slice in Cal Yr BP? "))

   while (is.na(intervalStop) | intervalStop > oldCalYr | intervalStop < 0 | intervalStop < intervalStart)
```

```
   intervalStop=as.numeric(readline("What is the stop interval (oldest) of the time slice in Cal Yr BP? "))

# only get rows where calMean value falls within the range of the time slice specified above
anByTimeSlice1 <- spatialJoinFile[spatialJoinFile$CalMean >= intervalStart & spatialJoinFile$CalMean <= intervalStop, ]

calYrList <- unique(anByTimeSlice1[,4])
siteList <- unique(anByTimeSlice1[,1])

#print(length(calYrList))
#print(siteList[1])
#print(calYrList[1])

anByTimeSlice2 <- aggregate(anByTimeSlice1[,10], list(anByTimeSlice1[,4],anByTimeSlice1[,1],anByTimeSlice1[,2],anByTimeSlice1[,3]),
      mean)
anByTimeSlice2 <- anByTimeSlice2[order(anByTimeSlice2[,2]),]

#print(anByTimeSlice2[1,2])

#write.csv(anByTimeSlice1,"AvgAnomByTimeSlice1.csv",row.names=FALSE)
#write.csv(anByTimeSlice2,"AvgAnomByTimeSlice2.csv",row.names=FALSE)

#anByTimeSliceA <- reconFileJoin[reconFileJoin$CalMean >= intervalStart & reconFileJoin$CalMean <= intervalStop, ]

# now need to avg the values at each site (e.g. if 800-1200 bp is specified, for site 7 will avg the values at 800,900
# 1000, 1100 and 1200 to have 1 value for the period 800-1200 at that site, this is repeated for each site in the file

# avg the anomalies (for cal Yr BP interval range) for each site
# make sure to get lat and lon values for each site

# anByTimeSliceC <- aggregate(anByTimeSliceA[,10], list(anByTimeSliceA[,1],anByTimeSliceA[,2],anByTimeSliceA[,3],anByTimeSliceA[,4])
      , mean)
# anByTimeSliceC <- aggregate(anByTimeSliceB[,5], list(anByTimeSliceB[,2],anByTimeSliceB[,3],anByTimeSliceB[,4]), mean)

while (is.na(calTSavg) | calTSavg < 1 | calTSavg > 2)
{
   message("How do you wish to calculate the average anomalies by time slice?")
   calTSavg=as.numeric(readline("A site must be in each time interval of the entire range <enter 1> or a site must be in at least
         one time interval of the range <enter 2> to be part of the calculated average: "))
}
if (calTSavg == 1)
{
  k=1 #row index for output matrix
  j=1 #row index for input matrix

  # setup the output matrix
  calcMat <- mat.or.vec(length(siteList),4)
  colnames(calcMat) = c("Site","Lon","Lat","AvgAn")
  #nRows = nrow(anByTimeSlice2)

  while (j <= length(siteList)) # go through list of unique sites
  {
     anByTimeSlice2bySite <- subset(anByTimeSlice2, anByTimeSlice2$Group.2 == siteList[j]) # get subset for that site
     if (nrow(anByTimeSlice2bySite) == length(calYrList)) #only want to average anomaly by site for sites that are common to all
           intervals in cal yr range
     {
        calcMat[k,1] <- anByTimeSlice2bySite[1,2] # siteID
        calcMat[k,2] <- anByTimeSlice2bySite[1,3] # Lon
        calcMat[k,3] <- anByTimeSlice2bySite[1,4] # Lat
        # now calc avg anom for all intervals in cal yr bp range for current site
        calcMat[k,4] <- sum(anByTimeSlice2bySite[,5]) / nrow(anByTimeSlice2bySite)
        #print(anByTimeSlice2bySite)
        #print(k)
        k = k+1
     }
     j = j+1
  }

  # get rid of any rows with zeroes
  calcMat <- subset(calcMat,calcMat[,1] > 0)

  anByTimeSlice3 <- calcMat
}else
{
   # write.csv(calcMat,"calcMatOutput1.csv",row.names=FALSE)
   anByTimeSlice3 <- aggregate(anByTimeSlice2[,5], list(anByTimeSlice2[,2],anByTimeSlice2[,3],anByTimeSlice2[,4]), mean)
}
```

```
anByTimeSliceC <- anByTimeSlice3

# fix column headers here
colnames(anByTimeSlice3)[1] <- "Site"
colnames(anByTimeSlice3)[2] <- "Lon"
colnames(anByTimeSlice3)[3] <- "Lat"
timeInterval = paste(intervalStart,"to",intervalStop, "bp_Anom")
timeInterval = gsub(" ","", timeInterval ,fixed=TRUE)
colnames(anByTimeSlice3)[4] <- timeInterval

colnames(anByTimeSliceC)[1] <- "Site"
colnames(anByTimeSliceC)[2] <- "Lon"
colnames(anByTimeSliceC)[3] <- "Lat"
colnames(anByTimeSliceC)[4] <- "avgAnTS"

# now output to a csv file
message("Writing file AvgAnomByTimeSlice.csv with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
tryCatch({write.csv(anByTimeSlice3,"AvgAnomByTimeSlice.csv",row.names=FALSE)}, error = function(e){message("AvgAnomByTimeSliceFinal.
      csv cannot be written to disk, please close the file!")}, warning = function(w){})

# ask user if they want to output a shapefile
while (is.na(outputSHP) | outputSHP < 1 | outputSHP > 2)
   outputSHP=as.numeric(readline("Do you wish to output a shapefile of the avg anomaly data for selected time slice? <1 for YES, 2
         for NO> "))

if (outputSHP == 1)
{
   tryCatch({
   # now output shapefile (can be loaded in GIS software)
   ddTable <- data.frame(Id=anByTimeSlice3[,1],lon=anByTimeSlice3[,2],lat=anByTimeSlice3[,3],avgAN=anByTimeSlice3[,4])
   ddShapefile <- convert.to.shapefile(anByTimeSlice3, ddTable, "Id", 1)
   message("Writing file AvgAnomByTimeSlice shapefile with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
   write.shapefile(ddShapefile, "AvgAnomByTimeSlice", arcgis=T)}, error = function(e){message("Error creating AvgAnomByTimeSlice
         shapefile!")}, warning = function(w){})
}
colnames(anByTimeSlice3)[4] <- "avgAnom"

# Ask user if they would like to compare selected avg anomaly by time slice to another avg anomaly by time slice (from file) before
      doing interpolation & mapping
# If YES, need to read in file then create a new matrix with differences beween 2 time periods (need to match sites here, exclude
      any non-matching sites)
# If NO, simply proceed to the interpolation & mapping below

timeSliceDiffYN = 2
# ask user if they want to compare selected avg anomaly by time slice data to another avg anom by time slice (from file)
while (is.na(timeSliceDiffYN) | timeSliceDiffYN < 1 | timeSliceDiffYN > 2)
   timeSliceDiffYN=as.numeric(readline("Do you wish to compare selected average anomaly by time slice data set to another avg
         anomaly by time slice data set (from file) before doing interpolation/mapping ? <1 for YES, 2 for NO> "))

if (timeSliceDiffYN == 1)
{

  tryCatch({
     linkTSdataFile <- file.choose()
     }, error = function(ex) {
  })

  if(!is.na(linkTSdataFile))
  {

    fileTypeMatch <- regexpr("AvgAnom",basename(linkTSdataFile))

    # ts1 = current subset
    # ts2 = subset read in from csv file

    ts1 <- anByTimeSlice3

    if (fileTypeMatch != -1)
    {
       ts2 <- read.csv(linkTSdataFile, TRUE, ",")

       # anomalies are ts2 relative to ts1
       # when comparing matrices, always use the one with the lesser # of rows

       ts1Rows <- nrow(ts1)
```

```
      ts2Rows <- nrow(ts2)

    if (ts1Rows >= ts2Rows)
    {
      diffMat2 <- ts2
    }else # ts1Rows < ts2Rows
    {
      diffMat1 <- ts1
    }
    tryCatch({
    if (is.null(diffMat1)) # use diffMat2
    {
      while (rSel <= ts2Rows)
      {
        siteMatch <- which(ts2[rSel,1] == ts1[,1]) # match siteIDs
        if(!length(siteMatch) == 0) # skip sites that do not match
          diffMat2[rSel,4] = ts2[rSel,4] - ts1[siteMatch,4] # calc diff
        rSel = rSel + 1
      }
      ind <- apply(diffMat2, 1, function(diffMat2) all(is.na(diffMat2)))
      diffMat2 <- diffMat2[ !ind, ] # remove NAs
      colnames(diffMat2)[4] = "AnDiff"
      tryCatch({write.csv(diffMat2,"avgAnTSdiff.csv",row.names=FALSE)}, error = function(e) {message("Cannot write avgAnTSdiff.csv
            to disk, file may be open!")}, warning = function(w){})
      if (outputSHP == 1)
      {
        tryCatch({
        # now output shapefile (can be loaded in GIS software)
        ddTable <- data.frame(Id=diffMat2[,1],lon=diffMat2[,2],lat=diffMat2[,3],avgAN=diffMat2[,4])
        ddShapefile <- convert.to.shapefile(diffMat2, ddTable, "Id", 1)
        message("Writing file diffMat2 shapefile with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
        write.shapefile(ddShapefile, "diffMat2", arcgis=T)}, error = function(e){message("diffMat2 shapefile cannot be created!")
              }, warning = function(w){})
      }
    }else
    {
      while (rSel <= ts1Rows) # use diffMat1
      {
        siteMatch <- which(ts1[rSel,1] == ts2[,1]) # match siteIDs
        if(!length(siteMatch) == 0) # skip sites that do not match
          diffMat1[rSel,4] = ts1[rSel,4] - ts2[siteMatch,4] # calc diff
        rSel = rSel + 1
      }
      ind <- apply(diffMat1, 1, function(diffMat1) all(is.na(diffMat1)))
      diffMat1 <- diffMat1[ !ind, ] # remove NAs
      colnames(diffMat1)[4] = "AnDiff"
      tryCatch({write.csv(diffMat1,"avgAnTSdiff.csv",row.names=FALSE)}, error = function(e) {message("Cannot write avgAnTSdiff.csv
            to disk, file may be open!")}, warning = function(w){})
      if (outputSHP == 1)
      {
        tryCatch({
        # now output shapefile (can be loaded in GIS software)
        ddTable <- data.frame(Id=diffMat1[,1],lon=diffMat1[,2],lat=diffMat1[,3],avgAN=diffMat1[,4])
        ddShapefile <- convert.to.shapefile(diffMat1, ddTable, "Id", 1)
        message("Writing file diffMat1 shapefile with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
        write.shapefile(ddShapefile, "diffMat1", arcgis=T)}, error = function(e){message("diffMat1 shapefile cannot be created!")
              }, warning = function(w){})
      }
    }
    }, error = function(e){}, warning = function(w){})
   }else
     message("Invalid average anomaly time slice data file selected!")
  }else
    message("No average anomaly time slice data file selected!")
}

if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90)
{
  lonLimits <- c(-170,-40)
  latLimits <- c(15,90)
  mapOK <- TRUE
  message("Map will be created for North America")
}else
{
  lonLimits <- c(lonMax,lonMin)
  latLimits <- c(latMin,latMax)
```

```
   mapOK <- FALSE
   message("Subset of North America detected, map will not be created!")
}

while (is.na(interpMapYN) | interpMapYN < 1 | interpMapYN > 2)
   interpMapYN=as.numeric(readline("Do you want to interpolate the output with kriging and then create a map? <1 for YES, 2 for NO>
         "))

if (interpMapYN == 1)
{

   # check whether we need to map anTS, diffMat1 or diffMat2 (can check is.null for diffMat1 and 2)

   if (is.null(diffMat1) & is.null(diffMat2))
      selAvgAnTS = anByTimeSlice3

   if (!is.null(diffMat1))
      selAvgAnTS = diffMat1

   if (!is.null(diffMat2))
      selAvgAnTS = diffMat2

   df1=data.frame(lat=selAvgAnTS[,3],long=selAvgAnTS[,2],avgAN=selAvgAnTS[,4])
   initial.df=df1

   # ask user if they want to plot a map of point anomalies?
   tryCatch({
     selAvgAnTSneg <- subset(selAvgAnTS, selAvgAnTS[,4] < -0.000001)
     selAvgAnTSzero <- subset(selAvgAnTS, selAvgAnTS[,4] >= -0.000001 & selAvgAnTS[,4] <= 0.000001 )
     selAvgAnTSpos <- subset(selAvgAnTS, selAvgAnTS[,4] > 0.000001)

     map(database = "world",xlim = lonLimits,ylim = latLimits)
     map.axes()

     points(selAvgAnTSneg$Lon,selAvgAnTSneg$Lat,pch=21,col="blue",cex=1)
     points(selAvgAnTSzero$Lon,selAvgAnTSzero$Lat,pch=21,col="darkgrey",cex=1)
     points(selAvgAnTSpos$Lon,selAvgAnTSpos$Lat,pch=21,col="red",cex=1)
     legend("bottomleft",legend=c("Positive","Zero","Negative"),pch =c(21,2),col=c("red","darkgrey","blue"))

     message(paste("There are ",length(unique(selAvgAnTSneg[,1]))," sites with negative anomalies",sep=" "))
     message(paste("There are ",length(unique(selAvgAnTSzero[,1]))," sites with zero anomalies",sep=" "))
     message(paste("There are ",length(unique(selAvgAnTSpos[,1]))," sites with positive anomalies",sep=" "))

     message(paste("Max average anomaly difference value is",max(selAvgAnTS[,4])))
     message(paste("Min average anomaly difference value is",min(selAvgAnTS[,4])))

   }, error = function(e){message("Problem creating map of average anomalies by time slice!");mapOK = FALSE }, warning = function(w)
         {})


   if (mapOK)
   {
     # ask user for zlim based on the following
     while (is.na(zLimVal) | zLimVal == 0)
        zLimVal=as.numeric(readline("What is the max/min anomaly range for map? (e.g. if 1 is entered, max will be 1, min will be -1)
              Should be based on min/max values above or plot could fail!) "))

     # now convert to spatialPoints object to reproject data, then do kriging, the plot the interpolated map

     coordinates(df1)=~long+lat
     proj4string(df1)="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
     # project_df=spTransform(df1, CRS("+proj=eqdc +lat_0=45 +lon_0=-100 +lat_1=43.33 +lat_2=66.66 +x_0=0 +y_0=0 +ellps=GRS80 +datum=
           NAD83 +units=m +no_defs +towgs84=0,0,0"))
     project_df=spTransform(df1, CRS("+proj=aea +lat_0=40 +lon_0=-96 +lat_1=20 +lat_2=60 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +
           units=m +no_defs +towgs84=0,0,0"))

     # would be nice to specify search radius & possibly other parameters for kriging
     kr=autoKrige(avgAN~1,project_df)
     kr.output=as.data.frame(kr$krige_output)

     while (is.null(plotTitle))
     {
        plotTitle=readline("Enter the title/file name for this plot: ")
     }

     fileName = plotTitle
```

```
  fileNameOut = paste(fileName,".pdf")
  fileNameOut = gsub(" ","", fileNameOut ,fixed=TRUE)

  # ask user if they want a gridded map or filled contour map?
  while (is.na(mapType) | mapType < 1 | mapType > 2)
  {
     mapType=as.numeric(readline("Do you want a gridded or filled contour map? <1 for GRIDDED, 2 for FILLED CONTOUR> "))
  }
  while (is.na(levCol) | levCol < 10 | levCol > 100)
  {
     levCol = as.numeric(readline("How many colour levels should the output map have, must be between 10 and 100 (use higher values
            for a less blocky appearance): "))
     levCol = levCol %/% 10 * 10
  }
  if (mapType == 2)
  {
     tryCatch({
        pdf(fileNameOut)
        my.image <- image.maker(kr.output[,c("x1","x2")],kr.output$var1.pred)
        if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
           filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
                 beige", "maroon"))(levCol),plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
        else
           filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
                 beige", "maroon"))(levCol),plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
        dev.off()
     }, error = function(e){message(paste(fileNameOut,"Error creating contour map, possibly file is already open!"))}, warning =
          function(w){})
  }else
  {
     # ask user for grid cell resolution (i.e. number of rows & cols - use higher values for smaller grid boxes)
     while (is.na(gCellRC) | gCellRC < 10 | gCellRC > 100)
     {
        gCellRC = as.numeric(readline("How many rows/columns should the gridded map have, must be between 10 and 100 (use higher
              values for smaller grid boxes): "))
        gCellRC = gCellRC %/% 10 * 10
     }
     coordinates(kr.output) <- ~x1+x2
     rast <- raster(ncol=gCellRC,nrow=gCellRC)
     extent(rast) <- extent(kr.output)
     rmap <- rasterize(kr.output, rast, kr.output$var1.pred, fun = mean)
     # tryCatch({
        pdf(fileNameOut)
        if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
           plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
                 levCol),alpha = 0.8,plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
        else
           plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
                 levCol),alpha = 0.8,plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
        dev.off()
     # }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning = function(
          w){})
  }
}

if (mapOK)
{
# now call another function (optional) for sequential time slice difference (dynamic) mapping
while (is.na(dynMapYN) | dynMapYN < 1 | dynMapYN > 2)
   dynMapYN=as.numeric(readline("Do you want to do mapping of avg anomalies for sequential time intervals (oldest to youngest)? <1
         for YES, 2 for NO> "))

if (dynMapYN == 1)
{

  source("image.maker.R")

  fileName = NULL
  diffMat1 = NULL
  diffMat2 = NULL
  selAvgAnTS = NULL
  my.image = NULL
  plotTitle = NULL
  ts1 = NULL
  ts2 = NULL
```

```
siteMatch = 0
zLimVal = 0
mapType = 0
gCellRC = 0
levCol = 0
rSel = 1

calYrList = unique(spatialJoinFile[,4])
calYrLast = length(calYrList)
calYrLast2 = calYrLast - 1
calYrInt = calYrList[calYrLast] - calYrList[calYrLast2] # determine interval

intervalStart = min(calYrList) # finish with youngest year
selMapYear2 = max(calYrList) # start with oldest year
selMapYear1 = selMapYear2 - calYrInt # one interval less than selMapYear2

while (selMapYear2 >= intervalStart & selMapYear1 >= intervalStart)
{
   message(paste("Processing average anomaly difference between",selMapYear2,"and",selMapYear1,"...."))

   # only get rows where calMean value falls within the range of the time slice specified above
   anByTimeSlice1 <- spatialJoinFile[spatialJoinFile$CalMean == selMapYear2, ]
   anByTimeSliceA <- spatialJoinFile[spatialJoinFile$CalMean == selMapYear1, ]

   # avg the anomalies (for cal Yr BP interval range) for each site
   # make sure to get lat and lon values for each site

   anByTimeSlice2 <- aggregate(anByTimeSlice1[,10], list(anByTimeSlice1[,4],anByTimeSlice1[,1],anByTimeSlice1[,2],anByTimeSlice1
         [,3]), mean)
   anByTimeSlice3 <- aggregate(anByTimeSlice2[,5], list(anByTimeSlice2[,2],anByTimeSlice2[,3],anByTimeSlice2[,4]), mean)

   anByTimeSliceB <- aggregate(anByTimeSliceA[,10], list(anByTimeSliceA[,4],anByTimeSliceA[,1],anByTimeSliceA[,2],anByTimeSliceA
         [,3]), mean)
   anByTimeSliceC <- aggregate(anByTimeSliceB[,5], list(anByTimeSliceB[,2],anByTimeSliceB[,3],anByTimeSliceB[,4]), mean)

   # fix column headers here
   colnames(anByTimeSlice3)[1] <- "Site"
   colnames(anByTimeSlice3)[2] <- "Lon"
   colnames(anByTimeSlice3)[3] <- "Lat"
   colnames(anByTimeSlice3)[4] <- "avgAnTS"

   colnames(anByTimeSliceC)[1] <- "Site"
   colnames(anByTimeSliceC)[2] <- "Lon"
   colnames(anByTimeSliceC)[3] <- "Lat"
   colnames(anByTimeSliceC)[4] <- "avgAnTS"

   ts2 = anByTimeSlice3
   ts1 = anByTimeSliceC

   tryCatch({write.csv(ts2,"ts2.csv",row.names=FALSE)}, error = function(e) {message("Cannot write ts2.csv to disk, file may be
         open!")}, warning = function(w){})
   tryCatch({write.csv(ts1,"ts1.csv",row.names=FALSE)}, error = function(e) {message("Cannot write ts1.csv to disk, file may be
         open!")}, warning = function(w){})

   # anomalies are ts2 relative to ts1
   # when comparing matrices, always use the one with the lesser # of rows

   ts1Rows <- nrow(ts1)
   ts2Rows <- nrow(ts2)

   # print(ts1Rows)
   # print(ts2Rows)

   if (ts1Rows >= ts2Rows)
   {
      diffMat2 <- ts2
   }else # ts1Rows < ts2Rows
   {
      diffMat1 <- ts1
   }

   if (is.null(diffMat1)) # use diffMat2
   {
     while (rSel <= ts2Rows)
     {
        siteMatch <- which(ts2[rSel,1] == ts1[,1]) # match siteIDs
```

```
    if(!length(siteMatch) == 0) # skip sites that do not match
      diffMat2[rSel,4] = ts2[rSel,4] - ts1[siteMatch,4] # calc diff
    rSel = rSel + 1
  }
  ind <- apply(diffMat2, 1, function(diffMat2) all(is.na(diffMat2)))
  diffMat2 <- diffMat2[ !ind, ] # remove NAs
  colnames(diffMat2)[4] = "AnDiff"
}else
{
  while (rSel <= ts1Rows) # use diffMat1
  {
    siteMatch <- which(ts1[rSel,1] == ts2[,1]) # match siteIDs
    if(!length(siteMatch) == 0) # skip sites that do not match
      diffMat1[rSel,4] = ts1[rSel,4] - ts2[siteMatch,4] # calc diff
    rSel = rSel + 1
  }
  ind <- apply(diffMat1, 1, function(diffMat1) all(is.na(diffMat1)))
  diffMat1 <- diffMat1[ !ind, ] # remove NAs
  colnames(diffMat1)[4] = "AnDiff"
}

# check whether we need to select diffMat1 or diffMat2 (can check is.null for diffMat1 and 2)
if (!is.null(diffMat1))
  selAvgAnTS = diffMat1
else
  selAvgAnTS = diffMat2

message(paste("Max average anomaly difference value is",max(selAvgAnTS[,4])))
message(paste("Min average anomaly difference value is",min(selAvgAnTS[,4])))

# ask user for zlim based on the following
while (is.na(zLimVal) | zLimVal == 0)
  zLimVal=as.numeric(readline("What is the max/min range for map anomalies? (e.g. if 1 is entered, max will be 1, min will be
      -1). Should be based on min/max values above or plot could fail!) "))

# now convert to spatialPoints object to reproject data, then do kriging, the plot the interpolated map
df1=data.frame(lat=selAvgAnTS[,3],long=selAvgAnTS[,2],avgAN=selAvgAnTS[,4])
initial.df=df1
coordinates(df1)=~long+lat
proj4string(df1)="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
project_df=spTransform(df1, CRS("+proj=aea +lat_0=40 +lon_0=-96 +lat_1=20 +lat_2=60 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +
      units=m +no_defs +towgs84=0,0,0"))

# would be nice to specify search radius & possibly other parameters for kriging
kr=autoKrige(avgAN~1,project_df)
kr.output=as.data.frame(kr$krige_output)

plotTitle=paste("Average anomaly for",selMapYear2,"-",selMapYear1,"bp")
fileName <- paste("AvgAnom_",selMapYear2,"-",selMapYear1,"_bp.pdf")
fileNameOut = gsub(" ","", fileName ,fixed=TRUE)

 # ask user if they want a gridded map or filled contour map?
 while (is.na(mapType) | mapType < 1 | mapType > 2)
 {
   mapType=as.numeric(readline("Do you want a gridded or filled contour map? <1 for GRIDDED, 2 for FILLED CONTOUR> "))
 }
 while (is.na(levCol) | levCol < 10 | levCol > 100)
 {
   levCol = as.numeric(readline("How many colour levels should the output map have, must be between 10 and 100 (use higher
        values for a less blocky appearance): "))
   levCol = levCol %/% 10 * 10
 }
 if (mapType == 2)
 {
   my.image <- image.maker(kr.output[,c("x1", "x2")], kr.output$var1.pred)
   tryCatch({
   pdf(fileNameOut)
   if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
     filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
          beige", "maroon"))(levCol),plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
   else
     filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
          beige", "maroon"))(levCol),plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
   dev.off()
   }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning =
        function(w){})
```

```
        }else
        {
           # ask user for grid cell resolution (i.e. number of rows & cols - use higher values for smaller grid boxes)
           while (is.na(gCellRC) | gCellRC < 10 | gCellRC > 100)
           {
              gCellRC = as.numeric(readline("How many rows/columns should the gridded map have, must be between 10 and 100 (use higher
                     values for smaller grid boxes): "))
              gCellRC = gCellRC %/% 10 * 10
           }

           coordinates(kr.output) <- ~x1+x2
           rast <- raster(ncol=gCellRC,nrow=gCellRC)
           extent(rast) <- extent(kr.output)
           rmap <- rasterize(kr.output, rast, kr.output$var1.pred, fun = mean)
           # tryCatch({
           pdf(fileNameOut)
           if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
              plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
                     levCol),alpha = 0.8,plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
           else
              plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
                     levCol),alpha = 0.8,plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
           dev.off()
           # }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning =
                 function(w){})
        }

        rSel = 1

        diffMat1 = NULL
        diffMat2 = NULL

        selMapYear2 = selMapYear2 - calYrInt
        selMapYear1 = selMapYear1 - calYrInt
      }
     }
    }
  }
  aList <- list(anByTimeSliceC, anByTimeSlice3)
  return(aList)
}
#########################################################################################################################
      outputAnomaliesByTimeSlice2 <- function( )
{
   # output a time slice of anomalies (useful for mapping in GIS software)
   # also creates a basic map of the anomalies & outputs a csv file
   # user needs to specify the range of calYrBP dates (e.g. 800 and 1200)
   # get anomalies for timeslice (e.g. cal YR BP between 800 & 1200)
   # can now easily get anomalies (timeslice - modern)

   # ask user for subset (time slice - can either be a single interval such as 1000 bp
   # or the avg of several intervals such as 800 to 1200bp) of avg anomaly data
   # next, do they want to map it (grid or contour) OR
   # load a second avg anomaly file (another time slice)
   # in csv format to compare with (option to map it)

   # finally, need to ask user about dynamic mapping for all time intervals
   # here we would output a series of maps in pdf format, procedure would be:
   # get subset of 2000bp and 1900bp, calc diff between them & then map
   # repeat for 1900bp and 1800bp, etc...to 100 and 0bp

   ### so the 3 options are with anomaly by time slice:
   # 1. map a single time slice relative to modern (e.g. 1000bp or 800-1200 bp or 100-600bp)
   # 2. map the difference between 2 time slices (e.g. 800-1200 from 200-600 bp or 1000bp from 100-600bp)
   # 3. map the difference between sequential intervals for the cal year bp range of the reconstruction
   # e.g. 2000-1900bp, 1900-1800bp, 1800-1700bp etc...to 100-0 bp


   require(shapefiles)
   require(maps)
   require(rgdal)
   require(automap)
   require(ggplot2)

   intervalStart = -99
   intervalStop = -99
```

```
rSel = 1
interpMapYN = 0
dynMapYN = 0
outputSHP = 0
timeSliceDiffYN = 0
fileTypeMatch = 0
mapType = 0
zLimVal = 0
gCellRC = 0
levCol = 0

plotTitle = NULL
fileName = NULL
fileNameOut = NULL
diffMat1 = NULL
diffMat2 = NULL
siteMatch = NULL
selAvgAnTS = NULL
my.image = NULL

mapOK = TRUE

intervalStart=1300
intervalStop=1700

spatialJoinFile <- read.csv("sp_filtering/_filteredReconstructionAllCalYrBP.csv")

#write.csv(spatialJoinFile,"spatialJoinFile.csv",row.names=FALSE)
#write.csv(reconFileJoin,"reconFileJoin.csv",row.names=FALSE)

#message(paste("Your data has a range of",youngCalYR,"to",oldCalYr,"calendar years BP"))
#message("Example of time slice anomaly (cal yr bp interval): 800 (youngest) to 1200 (oldest) or 1000 (youngest) to 1000 (oldest) ")

#while (is.na(intervalStart) | intervalStart < youngCalYR | intervalStart >= oldCalYr | intervalStart < 0)
# intervalStart=as.numeric(readline("What is the start interval (youngest) of the time slice in Cal Yr BP? "))

#while (is.na(intervalStop) | intervalStop > oldCalYr | intervalStop < 0 | intervalStop < intervalStart)
# intervalStop=as.numeric(readline("What is the stop interval (oldest) of the time slice in Cal Yr BP? "))


# only get rows where calMean value falls within the range of the time slice specified above
anByTimeSlice1 <- spatialJoinFile[spatialJoinFile$CalMean >= intervalStart & spatialJoinFile$CalMean <= intervalStop, ]

calYrList <- unique(anByTimeSlice1[,4])
siteList <- unique(anByTimeSlice1[,1])

#print(length(calYrList))
#print(siteList[1])
#print(calYrList[1])

anByTimeSlice2 <- aggregate(anByTimeSlice1[,10], list(anByTimeSlice1[,4],anByTimeSlice1[,1],anByTimeSlice1[,2],anByTimeSlice1[,3]),
      mean)
anByTimeSlice2 <- anByTimeSlice2[order(anByTimeSlice2[,2]),]

#print(anByTimeSlice2[1,2])

#write.csv(anByTimeSlice1,"AvgAnomByTimeSlice1.csv",row.names=FALSE)
write.csv(anByTimeSlice2,"AvgAnomByTimeSlice5.csv",row.names=FALSE)

#anByTimeSliceA <- reconFileJoin[reconFileJoin$CalMean >= intervalStart & reconFileJoin$CalMean <= intervalStop, ]

# now need to avg the values at each site (e.g. if 800-1200 bp is specified, for site 7 will avg the values at 800,900
# 1000, 1100 and 1200 to have 1 value for the period 800-1200 at that site, this is repeated for each site in the file

# avg the anomalies (for cal Yr BP interval range) for each site
# make sure to get lat and lon values for each site

# anByTimeSliceC <- aggregate(anByTimeSliceA[,10], list(anByTimeSliceA[,1],anByTimeSliceA[,2],anByTimeSliceA[,3],anByTimeSliceA[,4])
      , mean)
# anByTimeSliceC <- aggregate(anByTimeSliceB[,5], list(anByTimeSliceB[,2],anByTimeSliceB[,3],anByTimeSliceB[,4]), mean)


k=1 #row index for output matrix
j=1 #row index for input matrix
```

```
# setup the output matrix
calcMat <- mat.or.vec(length(siteList),4)
colnames(calcMat) = c("Site","Lon","Lat","AvgAn")
#nRows = nrow(anByTimeSlice2)

while (j <= length(siteList)) # go through list of unique sites
{
   anByTimeSlice2bySite <- subset(anByTimeSlice2, anByTimeSlice2$Group.2 == siteList[j]) # get subset for that site
   if (nrow(anByTimeSlice2bySite) == length(calYrList)) #only want to average anomaly by site for sites that are common to all
         intervals in cal yr range
   {
      calcMat[k,1] <- anByTimeSlice2bySite[1,2] # siteID
      calcMat[k,2] <- anByTimeSlice2bySite[1,3] # Lon
      calcMat[k,3] <- anByTimeSlice2bySite[1,4] # Lat
      # now calc avg anom for all intervals in cal yr bp range for current site
      calcMat[k,4] <- sum(anByTimeSlice2bySite[,5]) / nrow(anByTimeSlice2bySite)
      #print(anByTimeSlice2bySite)
      #print(k)
      k = k+1
   }
   j = j+1
}

# get rid of any rows with zeroes
calcMat <- subset(calcMat,calcMat[,1] > 0)

anByTimeSlice3 <- calcMat

# write.csv(calcMat,"calcMatOutput1.csv",row.names=FALSE)

# anByTimeSlice3 <- aggregate(anByTimeSlice2[,5], list(anByTimeSlice2[,1],anByTimeSlice2[,2],anByTimeSlice2[,3]), mean)

anByTimeSliceC <- anByTimeSlice3

# fix column headers here
colnames(anByTimeSlice3)[1] <- "Site"
colnames(anByTimeSlice3)[2] <- "Lon"
colnames(anByTimeSlice3)[3] <- "Lat"
timeInterval = paste(intervalStart,"to",intervalStop, "avgAN")
timeInterval = gsub(" ","", timeInterval ,fixed=TRUE)
colnames(anByTimeSlice3)[4] <- timeInterval

colnames(anByTimeSliceC)[1] <- "Site"
colnames(anByTimeSliceC)[2] <- "Lon"
colnames(anByTimeSliceC)[3] <- "Lat"
colnames(anByTimeSliceC)[4] <- "avgAnTS"

# now output to a csv file
message("Writing file AvgAnomByTimeSlice.csv with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
tryCatch({write.csv(anByTimeSlice3,"AvgAnomByTimeSlice3.csv",row.names=FALSE)}, error = function(e){message("AvgAnomByTimeSliceFinal
      .csv cannot be written to disk, please close the file!")}, warning = function(w){})

return("DONE!")

# ask user if they want to output a shapefile
while (is.na(outputSHP) | outputSHP < 1 | outputSHP > 2)
   outputSHP=as.numeric(readline("Do you wish to output a shapefile of the avg anomaly data for selected time slice? <1 for YES, 2
         for NO> "))

if (outputSHP == 1)
{
   # now output shapefile (can be loaded in GIS software)
   ddTable <- data.frame(Id=anByTimeSlice3[,1],lon=anByTimeSlice3[,2],lat=anByTimeSlice3[,3],avgAN=anByTimeSlice3[,4])
   ddShapefile <- convert.to.shapefile(anByTimeSlice3, ddTable, "Id", 1)
   message("Writing file AvgAnomByTimeSlice shapefile with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
   tryCatch({ write.shapefile(ddShapefile, "AvgAnomByTimeSlice", arcgis=T)}, error = function(e){message("AvgAnomByTimeSlice
         shapefile cannot be written to disk, please close the file!")}, warning = function(w){})
}
colnames(anByTimeSlice3)[4] <- "avgAnom"

# Ask user if they would like to compare selected avg anomaly by time slice to another avg anomaly by time slice (from file) before
      doing interpolation & mapping
# If YES, need to read in file then create a new matrix with differences beween 2 time periods (need to match sites here, exclude
      any non-matching sites)
# If NO, simply proceed to the interpolation & mapping below
```

```
 # ask user if they want to compare selected avg anomaly by time slice data to another avg anom by time slice (from file)
 while (is.na(timeSliceDiffYN) | timeSliceDiffYN < 1 | timeSliceDiffYN > 2)
    timeSliceDiffYN=as.numeric(readline("Do you wish to compare selected average anomaly by time slice data set to another avg
          anomaly by time slice data set (from file) before doing interpolation/mapping ? <1 for YES, 2 for NO> "))

 if (timeSliceDiffYN == 1)
 {

   tryCatch({
      linkTSdataFile <- file.choose()
      }, error = function(ex) {
   })

   if(!is.na(linkTSdataFile))
   {

     fileTypeMatch <- regexpr("AvgAnomTS",basename(linkTSdataFile))

     # ts1 = current subset
     # ts2 = subset read in from csv file

     ts1 <- anByTimeSlice3

     if (fileTypeMatch != -1)
     {
        ts2 <- read.csv(linkTSdataFile, TRUE, ",")

        # anomalies are ts2 relative to ts1
        # when comparing matrices, always use the one with the lesser # of rows

        ts1Rows <- nrow(ts1)
        ts2Rows <- nrow(ts2)

       if (ts1Rows >= ts2Rows)
       {
          diffMat2 <- ts2
       }else # ts1Rows < ts2Rows
       {
          diffMat1 <- ts1
       }
       tryCatch({
       if (is.null(diffMat1)) # use diffMat2
       {
          while (rSel <= ts2Rows)
          {
             siteMatch <- which(ts2[rSel,1] == ts1[,1]) # match siteIDs
             if(!length(siteMatch) == 0) # skip sites that do not match
                diffMat2[rSel,4] = ts2[rSel,4] - ts1[siteMatch,4] # calc diff
             rSel = rSel + 1
          }
          ind <- apply(diffMat2, 1, function(diffMat2) all(is.na(diffMat2)))
          diffMat2 <- diffMat2[ !ind, ] # remove NAs
          colnames(diffMat2)[4] = "AnDiff"
          write.csv(diffMat2,"avgAnTSdiff.csv",row.names=FALSE)
       }else
       {
          while (rSel <= ts1Rows) # use diffMat1
          {
             siteMatch <- which(ts1[rSel,1] == ts2[,1]) # match siteIDs
             if(!length(siteMatch) == 0) # skip sites that do not match
                diffMat1[rSel,4] = ts1[rSel,4] - ts2[siteMatch,4] # calc diff
             rSel = rSel + 1
          }
          ind <- apply(diffMat1, 1, function(diffMat1) all(is.na(diffMat1)))
          diffMat1 <- diffMat1[ !ind, ] # remove NAs
          colnames(diffMat1)[4] = "AnDiff"
          write.csv(diffMat1,"avgAnomTSdiff.csv",row.names=FALSE)
       }
       }, error = function(e){}, warning = function(w){})
     }else
        message("Invalid average anomaly time slice data file selected!")
   }else
      message("No average anomaly time slice data file selected!")
}

if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90)
```

```
{
   lonLimits <- c(-170,-40)
   latLimits <- c(15,90)
   mapOK <- TRUE
   message("Map will be created for North America")
}else
{
   lonLimits <- c(lonMax,lonMin)
   latLimits <- c(latMin,latMax)
   mapOK <- FALSE
   message("Subset of North America detected, map will not be created!")
}

while (is.na(interpMapYN) | interpMapYN < 1 | interpMapYN > 2)
   interpMapYN=as.numeric(readline("Do you want to interpolate the output with kriging and then create a map? <1 for YES, 2 for NO>
         "))

if (interpMapYN == 1)
{

   # check whether we need to map anTS, diffMat1 or diffMat2 (can check is.null for diffMat1 and 2)

   if (is.null(diffMat1) & is.null(diffMat2))
      selAvgAnTS = anByTimeSlice3

   if (!is.null(diffMat1))
      selAvgAnTS = diffMat1

   if (!is.null(diffMat2))
      selAvgAnTS = diffMat2

   df1=data.frame(lat=selAvgAnTS[,3],long=selAvgAnTS[,2],avgAN=selAvgAnTS[,4])
   initial.df=df1

   # ask user if they want to plot a map of point anomalies?
   selAvgAnTSneg <- subset(selAvgAnTS, selAvgAnTS[,4] < -0.000001)
   selAvgAnTSzero <- subset(selAvgAnTS, selAvgAnTS[,4] >= -0.000001 & selAvgAnTS[,4] <= 0.000001 )
   selAvgAnTSpos <- subset(selAvgAnTS, selAvgAnTS[,4] > 0.000001)

   map(database = "world",xlim = lonLimits,ylim = latLimits)
   map.axes()

   points(selAvgAnTSneg$Lon,selAvgAnTSneg$Lat,pch=21,col="blue",cex=1)
   points(selAvgAnTSzero$Lon,selAvgAnTSzero$Lat,pch=21,col="darkgrey",cex=1)
   points(selAvgAnTSpos$Lon,selAvgAnTSpos$Lat,pch=21,col="red",cex=1)
   legend("bottomleft",legend=c("Positive","Zero","Negative"),pch =c(21,2),col=c("red","darkgrey","blue"))

   message(paste("There are ",length(unique(selAvgAnTSneg[,1])),"  sites with negative anomalies",sep=" "))
   message(paste("There are ",length(unique(selAvgAnTSzero[,1])),"  sites with zero anomalies",sep=" "))
   message(paste("There are ",length(unique(selAvgAnTSpos[,1])),"  sites with positive anomalies",sep=" "))

   message(paste("Max average anomaly difference value is",max(selAvgAnTS[,4])))
   message(paste("Min average anomaly difference value is",min(selAvgAnTS[,4])))

   if (mapOK)
   {
     # ask user for zlim based on the following
     while (is.na(zLimVal) | zLimVal == 0)
        zLimVal=as.numeric(readline("What is the max/min anomaly range for map? (e.g. if 1 is entered, max will be 1, min will be -1)
              Should be based on min/max values above or plot could fail!) "))

     # now convert to spatialPoints object to reproject data, then do kriging, the plot the interpolated map

     coordinates(df1)=~long+lat
     proj4string(df1)="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
     # project_df=spTransform(df1, CRS("+proj=eqdc +lat_0=45 +lon_0=-100 +lat_1=43.33 +lat_2=66.66 +x_0=0 +y_0=0 +ellps=GRS80 +datum=
           NAD83 +units=m +no_defs +towgs84=0,0,0"))
     project_df=spTransform(df1, CRS("+proj=aea +lat_0=40 +lon_0=-96 +lat_1=20 +lat_2=60 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +
           units=m +no_defs +towgs84=0,0,0"))

     # would be nice to specify search radius & possibly other parameters for kriging
     kr=autoKrige(avgAN~1,project_df)
     kr.output=as.data.frame(kr$krige_output)

     while (is.null(plotTitle))
     {
```

```
   plotTitle=readline("Enter the title/file name for this plot: ")
 }

 fileName = plotTitle
 fileNameOut = paste(fileName,".pdf")
 fileNameOut = gsub(" ","", fileNameOut ,fixed=TRUE)

 # ask user if they want a gridded map or filled contour map?
 while (is.na(mapType) | mapType < 1 | mapType > 2)
 {
   mapType=as.numeric(readline("Do you want a gridded or filled contour map? <1 for GRIDDED, 2 for FILLED CONTOUR> "))
 }
 while (is.na(levCol) | levCol < 10 | levCol > 100)
 {
   levCol = as.numeric(readline("How many colour levels should the output map have, must be between 10 and 100 (use higher values
          for a less blocky appearance): "))
   levCol = levCol %/% 10 * 10
 }
 if (mapType == 2)
 {
   my.image <- image.maker(kr.output[,c("x1","x2")],kr.output$var1.pred)
   tryCatch({
     pdf(fileNameOut)
     if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
       filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
            beige", "maroon"))(levCol),plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
     else
       filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
            beige", "maroon"))(levCol),plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
     dev.off()
   }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning = function(w)
        {})
 }else
 {
   # ask user for grid cell resolution (i.e. number of rows & cols - use higher values for smaller grid boxes)
   while (is.na(gCellRC) | gCellRC < 10 | gCellRC > 100)
   {
     gCellRC = as.numeric(readline("How many rows/columns should the gridded map have, must be between 10 and 100 (use higher
          values for smaller grid boxes): "))
     gCellRC = gCellRC %/% 10 * 10
   }
   coordinates(kr.output) <- ~x1+x2
   rast <- raster(ncol=gCellRC,nrow=gCellRC)
   extent(rast) <- extent(kr.output)
   rmap <- rasterize(kr.output, rast, kr.output$var1.pred, fun = mean)
   # tryCatch({
     pdf(fileNameOut)
     if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
       plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
            levCol),alpha = 0.8,plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
     else
       plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
            levCol),alpha = 0.8,plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
     dev.off()
   # }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning = function(
        w){})
 }
}

if (mapOK)
{
# now call another function (optional) for sequential time slice difference (dynamic) mapping
while (is.na(dynMapYN) | dynMapYN < 1 | dynMapYN > 2)
  dynMapYN=as.numeric(readline("Do you want to do mapping of avg anomalies for sequential time intervals (oldest to youngest)? <1
       for YES, 2 for NO> "))

if (dynMapYN == 1)
{

 source("image.maker.R")

 fileName = NULL
 diffMat1 = NULL
 diffMat2 = NULL
 selAvgAnTS = NULL
 my.image = NULL
```

```
plotTitle = NULL
ts1 = NULL
ts2 = NULL

siteMatch = 0
zLimVal = 0
mapType = 0
gCellRC = 0
levCol = 0
rSel = 1

calYrList = unique(spatialJoinFile[,4])
calYrLast = length(calYrList)
calYrLast2 = calYrLast - 1
calYrInt = calYrList[calYrLast] - calYrList[calYrLast2] # determine interval

intervalStart = min(calYrList) # finish with youngest year
selMapYear2 = max(calYrList) # start with oldest year
selMapYear1 = selMapYear2 - calYrInt # one interval less than selMapYear2

while (selMapYear2 >= intervalStart & selMapYear1 >= intervalStart)
{
   message(paste("Processing average anomaly difference between",selMapYear2,"and",selMapYear1,"...."))

   # only get rows where calMean value falls within the range of the time slice specified above
   anByTimeSlice1 <- spatialJoinFile[spatialJoinFile$CalMean == selMapYear2, ]
   anByTimeSliceA <- spatialJoinFile[spatialJoinFile$CalMean == selMapYear1, ]

   # avg the anomalies (for cal Yr BP interval range) for each site
   # make sure to get lat and lon values for each site

   anByTimeSlice2 <- aggregate(anByTimeSlice1[,10], list(anByTimeSlice1[,4],anByTimeSlice1[,1],anByTimeSlice1[,2],anByTimeSlice1
         [,3]), mean)
   anByTimeSlice3 <- aggregate(anByTimeSlice2[,5], list(anByTimeSlice2[,2],anByTimeSlice2[,3],anByTimeSlice2[,4]), mean)

   anByTimeSliceB <- aggregate(anByTimeSliceA[,10], list(anByTimeSliceA[,4],anByTimeSliceA[,1],anByTimeSliceA[,2],anByTimeSliceA
         [,3]), mean)
   anByTimeSliceC <- aggregate(anByTimeSliceB[,5], list(anByTimeSliceB[,2],anByTimeSliceB[,3],anByTimeSliceB[,4]), mean)

   # fix column headers here
   colnames(anByTimeSlice3)[1] <- "Site"
   colnames(anByTimeSlice3)[2] <- "Lon"
   colnames(anByTimeSlice3)[3] <- "Lat"
   colnames(anByTimeSlice3)[4] <- "avgAnTS"

   colnames(anByTimeSliceC)[1] <- "Site"
   colnames(anByTimeSliceC)[2] <- "Lon"
   colnames(anByTimeSliceC)[3] <- "Lat"
   colnames(anByTimeSliceC)[4] <- "avgAnTS"

   ts2 = anByTimeSlice3
   ts1 = anByTimeSliceC

   # write.csv(ts2,"ts2.csv",row.names=FALSE)
   # write.csv(ts1,"ts1.csv",row.names=FALSE)

   # anomalies are ts2 relative to ts1
   # when comparing matrices, always use the one with the lesser # of rows

   ts1Rows <- nrow(ts1)
   ts2Rows <- nrow(ts2)

   # print(ts1Rows)
   # print(ts2Rows)

   if (ts1Rows >= ts2Rows)
   {
      diffMat2 <- ts2
   }else # ts1Rows < ts2Rows
   {
      diffMat1 <- ts1
   }

   if (is.null(diffMat1)) # use diffMat2
   {
     while (rSel <= ts2Rows)
```

```
  {
    siteMatch <- which(ts2[rSel,1] == ts1[,1]) # match siteIDs
    if(!length(siteMatch) == 0) # skip sites that do not match
      diffMat2[rSel,4] = ts2[rSel,4] - ts1[siteMatch,4] # calc diff
    rSel = rSel + 1
  }
  ind <- apply(diffMat2, 1, function(diffMat2) all(is.na(diffMat2)))
  diffMat2 <- diffMat2[ !ind, ] # remove NAs
  colnames(diffMat2)[4] = "AnDiff"
}else
{
  while (rSel <= ts1Rows) # use diffMat1
  {
    siteMatch <- which(ts1[rSel,1] == ts2[,1]) # match siteIDs
    if(!length(siteMatch) == 0) # skip sites that do not match
      diffMat1[rSel,4] = ts1[rSel,4] - ts2[siteMatch,4] # calc diff
    rSel = rSel + 1
  }
  ind <- apply(diffMat1, 1, function(diffMat1) all(is.na(diffMat1)))
  diffMat1 <- diffMat1[ !ind, ] # remove NAs
  colnames(diffMat1)[4] = "AnDiff"
}

# check whether we need to select diffMat1 or diffMat2 (can check is.null for diffMat1 and 2)
if (!is.null(diffMat1))
  selAvgAnTS = diffMat1
else
  selAvgAnTS = diffMat2

message(paste("Max average anomaly difference value is",max(selAvgAnTS[,4])))
message(paste("Min average anomaly difference value is",min(selAvgAnTS[,4])))

# ask user for zlim based on the following
while (is.na(zLimVal) | zLimVal == 0)
  zLimVal=as.numeric(readline("What is the max/min range for map anomalies? (e.g. if 1 is entered, max will be 1, min will be
       -1). Should be based on min/max values above or plot could fail!) "))

 # now convert to spatialPoints object to reproject data, then do kriging, the plot the interpolated map
 df1=data.frame(lat=selAvgAnTS[,3],long=selAvgAnTS[,2],avgAN=selAvgAnTS[,4])
 initial.df=df1
 coordinates(df1)=~long+lat
 proj4string(df1)="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
 project_df=spTransform(df1, CRS("+proj=aea +lat_0=40 +lon_0=-96 +lat_1=20 +lat_2=60 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +
      units=m +no_defs +towgs84=0,0,0"))

 # would be nice to specify search radius & possibly other parameters for kriging
 kr=autoKrige(avgAN~1,project_df)
 kr.output=as.data.frame(kr$krige_output)

 plotTitle=paste("Average anomaly for",selMapYear2,"-",selMapYear1,"bp")
 fileName <- paste("AvgAnom_",selMapYear2,"-",selMapYear1,"_bp.pdf")
 fileNameOut = gsub(" ","", fileName ,fixed=TRUE)

  # ask user if they want a gridded map or filled contour map?
  while (is.na(mapType) | mapType < 1 | mapType > 2)
  {
    mapType=as.numeric(readline("Do you want a gridded or filled contour map? <1 for GRIDDED, 2 for FILLED CONTOUR> "))
  }
  while (is.na(levCol) | levCol < 10 | levCol > 100)
  {
    levCol = as.numeric(readline("How many colour levels should the output map have, must be between 10 and 100 (use higher
         values for a less blocky appearance): "))
    levCol = levCol %/% 10 * 10
  }
  if (mapType == 2)
  {
    my.image <- image.maker(kr.output[,c("x1", "x2")], kr.output$var1.pred)
    tryCatch({
    pdf(fileNameOut)
    if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
      filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
            beige", "maroon"))(levCol),plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
    else
      filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
            beige", "maroon"))(levCol),plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
    dev.off()
```

```
      }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning =
            function(w){})
    }else
    {
      # ask user for grid cell resolution (i.e. number of rows & cols - use higher values for smaller grid boxes)
      while (is.na(gCellRC) | gCellRC < 10 | gCellRC > 100)
      {
        gCellRC = as.numeric(readline("How many rows/columns should the gridded map have, must be between 10 and 100 (use higher
              values for smaller grid boxes): "))
        gCellRC = gCellRC %/% 10 * 10
      }

      coordinates(kr.output) <- ~x1+x2
      rast <- raster(ncol=gCellRC,nrow=gCellRC)
      extent(rast) <- extent(kr.output)
      rmap <- rasterize(kr.output, rast, kr.output$var1.pred, fun = mean)
      # tryCatch({
      pdf(fileNameOut)
      if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
        plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
            levCol),alpha = 0.8,plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
      else
        plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
            levCol),alpha = 0.8,plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
      dev.off()
      # }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning =
            function(w){})
    }

    rSel = 1

    diffMat1 = NULL
    diffMat2 = NULL

    selMapYear2 = selMapYear2 - calYrInt
    selMapYear1 = selMapYear1 - calYrInt
   }
  }
 }
 aList <- list(anByTimeSliceC, anByTimeSlice3)
 return(aList)
}
#########################################################################################################################
    outputAnomaliesByTimeSlice3 <- function()
{
  # output a time slice of anomalies (useful for mapping in GIS software)
  # also creates a basic map of the anomalies & outputs a csv file
  # user needs to specify the range of calYrBP dates (e.g. 800 and 1200)
  # get anomalies for timeslice (e.g. cal YR BP between 800 & 1200)
  # can now easily get anomalies (timeslice - modern)

  # ask user for subset (time slice - can either be a single interval such as 1000 bp
  # or the avg of several intervals such as 800 to 1200bp) of avg anomaly data
  # next, do they want to map it (grid or contour) OR
  # load a second avg anomaly file (another time slice)
  # in csv format to compare with (option to map it)

  # finally, need to ask user about dynamic mapping for all time intervals
  # here we would output a series of maps in pdf format, procedure would be:
  # get subset of 2000bp and 1900bp, calc diff between them & then map
  # repeat for 1900bp and 1800bp, etc...to 100 and 0bp

  ### so the 3 options are with anomaly by time slice:
  # 1. map a single time slice relative to modern (e.g. 1000bp or 800-1200 bp or 100-600bp)
  # 2. map the difference between 2 time slices (e.g. 800-1200 from 200-600 bp or 1000bp from 100-600bp)
  # 3. map the difference between sequential intervals for the cal year bp range of the reconstruction
  # e.g. 2000-1900bp, 1900-1800bp, 1800-1700bp etc...to 100-0 bp


  require(shapefiles)
  require(maps)
  require(rgdal)
  require(automap)
  require(ggplot2)
```

```
intervalStart = -99
intervalStop = -99

rSel = 1
interpMapYN = 0
dynMapYN = 0
outputSHP = 0
timeSliceDiffYN = 0
fileTypeMatch = 0
mapType = 0
zLimVal = 0
gCellRC = 0
levCol = 0
calTSavg = 0

plotTitle = NULL
fileName = NULL
fileNameOut = NULL
diffMat1 = NULL
diffMat2 = NULL
siteMatch = NULL
selAvgAnTS = NULL
my.image = NULL
linkSpFilterData = NULL
anByTimeSlice1 = NULL
anByTimeSlice2 = NULL
anByTimeSlice3 = NULL
anByTimeSliceC = NULL

mapOK = TRUE

 fileTypeMatch = -1
 tryCatch({linkSpFilterData <- file.choose()
     fileTypeMatch <- regexpr(".csv",basename(linkSpFilterData))
     }, error = function(e) {}, warning = function(w){})

if (!is.null(linkSpFilterData) & fileTypeMatch != -1)
{
    spatialJoinFile <- read.csv(linkSpFilterData)

tryCatch({
#write.csv(spatialJoinFile,"spatialJoinFile.csv",row.names=FALSE)
#write.csv(reconFileJoin,"reconFileJoin.csv",row.names=FALSE)

calYrList <- unique(spatialJoinFile[,4])
youngCalYR <- min(calYrList)
oldCalYr <- max(calYrList)

message(paste("Your data has a range of",youngCalYR,"to",oldCalYr,"calendar years BP"))
message("Example of time slice anomaly (cal yr bp interval): 800 (youngest) to 1200 (oldest) or 1000 (youngest) to 1000 (oldest) ")

while (is.na(intervalStart) | intervalStart < youngCalYR | intervalStart >= oldCalYr | intervalStart < 0)
   intervalStart=as.numeric(readline("What is the start interval (youngest) of the time slice in Cal Yr BP? "))

while (is.na(intervalStop) | intervalStop > oldCalYr | intervalStop < 0 | intervalStop < intervalStart)
   intervalStop=as.numeric(readline("What is the stop interval (oldest) of the time slice in Cal Yr BP? "))

# only get rows where calMean value falls within the range of the time slice specified above
anByTimeSlice1 <- spatialJoinFile[spatialJoinFile$CalMean >= intervalStart & spatialJoinFile$CalMean <= intervalStop, ]

calYrList <- unique(anByTimeSlice1[,4])
siteList <- unique(anByTimeSlice1[,1])

#print(length(calYrList))
#print(siteList[1])
#print(calYrList[1])

anByTimeSlice2 <- aggregate(anByTimeSlice1[,5], list(anByTimeSlice1[,4],anByTimeSlice1[,1],anByTimeSlice1[,2],anByTimeSlice1[,3],
      anByTimeSlice1[,8]), mean)
anByTimeSlice2 <- anByTimeSlice2[order(anByTimeSlice2[,2]),]

#print(anByTimeSlice2[1,2])

#write.csv(anByTimeSlice1,"AvgAnomByTimeSlice1.csv",row.names=FALSE)
#write.csv(anByTimeSlice2,"AvgAnomByTimeSlice2.csv",row.names=FALSE)
```

```
#anByTimeSliceA <- reconFileJoin[reconFileJoin$CalMean >= intervalStart & reconFileJoin$CalMean <= intervalStop, ]

# now need to avg the values at each site (e.g. if 800-1200 bp is specified, for site 7 will avg the values at 800,900
# 1000, 1100 and 1200 to have 1 value for the period 800-1200 at that site, this is repeated for each site in the file

# avg the anomalies (for cal Yr BP interval range) for each site
# make sure to get lat and lon values for each site

# anByTimeSliceC <- aggregate(anByTimeSliceA[,10], list(anByTimeSliceA[,1],anByTimeSliceA[,2],anByTimeSliceA[,3],anByTimeSliceA[,4])
      , mean)
# anByTimeSliceC <- aggregate(anByTimeSliceB[,5], list(anByTimeSliceB[,2],anByTimeSliceB[,3],anByTimeSliceB[,4]), mean)

while (is.na(calTSavg) | calTSavg < 1 | calTSavg > 2)
{
   message("How do you wish to calculate the average anomalies by time slice?")
   calTSavg=as.numeric(readline("A site must be in each time interval of the entire range <enter 1> or a site must be in at least
         one time interval of the range <enter 2> to be part of the calculated average: "))
}
if (calTSavg == 1)
{
  k=1 #row index for output matrix
  j=1 #row index for input matrix

  # setup the output matrix
  calcMat <- mat.or.vec(length(siteList),4)
  colnames(calcMat) = c("Site","Lon","Lat","AvgAn")
  #nRows = nrow(anByTimeSlice2)

  while (j <= length(siteList)) # go through list of unique sites
  {
     anByTimeSlice2bySite <- subset(anByTimeSlice2, anByTimeSlice2$Group.2 == siteList[j]) # get subset for that site
     if (nrow(anByTimeSlice2bySite) == length(calYrList)) #only want to average anomaly by site for sites that are common to all
           intervals in cal yr range
     {
        calcMat[k,1] <- anByTimeSlice2bySite[1,2] # siteID
        calcMat[k,2] <- anByTimeSlice2bySite[1,3] # Lon
        calcMat[k,3] <- anByTimeSlice2bySite[1,4] # Lat
        # now calc avg anom for all intervals in cal yr bp range for current site
        calcMat[k,4] <- sum(anByTimeSlice2bySite[,5]) / nrow(anByTimeSlice2bySite)
        #print(anByTimeSlice2bySite)
        #print(k)
        k = k+1
     }
     j = j+1
  }

  # get rid of any rows with zeroes
  calcMat <- subset(calcMat,calcMat[,1] > 0)

  anByTimeSlice3 <- calcMat
}else
{
   # write.csv(calcMat,"calcMatOutput1.csv",row.names=FALSE)
   anByTimeSlice3 <- aggregate(anByTimeSlice2[,6], list(anByTimeSlice2[,2],anByTimeSlice2[,3],anByTimeSlice2[,4],anByTimeSlice2[,5])
         , mean)
}

anByTimeSliceC <- anByTimeSlice3

# fix column headers here
colnames(anByTimeSlice3)[1] <- "Site"
colnames(anByTimeSlice3)[2] <- "Lon"
colnames(anByTimeSlice3)[3] <- "Lat"
colnames(anByTimeSlice3)[4] <- "ModVal"
#timeInterval = paste(intervalStart,"to",intervalStop, "bp_AN")
#timeInterval = gsub(" ","", timeInterval ,fixed=TRUE)
colnames(anByTimeSlice3)[5] <- "Recon"

colnames(anByTimeSliceC)[1] <- "Site"
colnames(anByTimeSliceC)[2] <- "Lon"
colnames(anByTimeSliceC)[3] <- "Lat"
colnames(anByTimeSliceC)[4] <- "avgAnTS"

# now output to a csv file
message("Writing file AvgAnomByTimeSlice.csv with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
```

```
tryCatch({write.csv(anByTimeSlice3,"AvgAnomByTimeSlice.csv",row.names=FALSE)}, error = function(e){message("AvgAnomByTimeSliceFinal.
      csv cannot be written to disk, please close the file!")}, warning = function(w){})
}, error = function(e){message("Error!!")}, warning = function(w){})
}



# ask user if they want to output a shapefile
while (is.na(outputSHP) | outputSHP < 1 | outputSHP > 2)
   outputSHP=as.numeric(readline("Do you wish to output a shapefile of the avg anomaly data for selected time slice? <1 for YES, 2
         for NO> "))

if (outputSHP == 1)
{
   tryCatch({
   # now output shapefile (can be loaded in GIS software)
   ddTable <- data.frame(Id=anByTimeSlice3[,1],lon=anByTimeSlice3[,2],lat=anByTimeSlice3[,3],avgAN=anByTimeSlice3[,4])
   ddShapefile <- convert.to.shapefile(anByTimeSlice3, ddTable, "Id", 1)
   message("Writing file AvgAnomByTimeSlice shapefile with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
   write.shapefile(ddShapefile, "AvgAnomByTimeSlice", arcgis=T)}, error = function(e){message("Error creating AvgAnomByTimeSlice
         shapefile!")}, warning = function(w){})
}
colnames(anByTimeSlice3)[4] <- "avgAnom"

# Ask user if they would like to compare selected avg anomaly by time slice to another avg anomaly by time slice (from file) before
      doing interpolation & mapping
# If YES, need to read in file then create a new matrix with differences beween 2 time periods (need to match sites here, exclude
      any non-matching sites)
# If NO, simply proceed to the interpolation & mapping below

#timeSliceDiffYN = 2
# ask user if they want to compare selected avg anomaly by time slice data to another avg anom by time slice (from file)
while (is.na(timeSliceDiffYN) | timeSliceDiffYN < 1 | timeSliceDiffYN > 2)
   timeSliceDiffYN=as.numeric(readline("Do you wish to compare selected average anomaly by time slice data set to another avg
         anomaly by time slice data set (from file) before doing interpolation/mapping ? <1 for YES, 2 for NO> "))

if (timeSliceDiffYN == 1)
{

  tryCatch({
     linkTSdataFile <- file.choose()
     }, error = function(ex) {
  })

  if(!is.na(linkTSdataFile))
  {

    fileTypeMatch <- regexpr("AvgAnom",basename(linkTSdataFile))

    # ts1 = current subset
    # ts2 = subset read in from csv file

    ts1 <- anByTimeSlice3

    if (fileTypeMatch != -1)
    {
       ts2 <- read.csv(linkTSdataFile, TRUE, ",")

       # anomalies are ts2 relative to ts1
       # when comparing matrices, always use the one with the lesser # of rows

       ts1Rows <- nrow(ts1)
       ts2Rows <- nrow(ts2)

      if (ts1Rows >= ts2Rows)
      {
         diffMat2 <- ts2
      }else # ts1Rows < ts2Rows
      {
         diffMat1 <- ts1
      }
      tryCatch({
      if (is.null(diffMat1)) # use diffMat2
      {
         while (rSel <= ts2Rows)
         {
```

```
          siteMatch <- which(ts2[rSel,1] == ts1[,1]) # match siteIDs
          if(!length(siteMatch) == 0) # skip sites that do not match
            diffMat2[rSel,5] = ts2[rSel,5] - ts1[siteMatch,5] # calc diff
          rSel = rSel + 1
        }
        ind <- apply(diffMat2, 1, function(diffMat2) all(is.na(diffMat2)))
        diffMat2 <- diffMat2[ !ind, ] # remove NAs
        colnames(diffMat2)[5] = "AnDiff"
        tryCatch({write.csv(diffMat2,"avgAnTSdiff.csv",row.names=FALSE)}, error = function(e) {message("Cannot write avgAnTSdiff.csv
              to disk, file may be open!")}, warning = function(w){})
        if (outputSHP == 1)
        {
          tryCatch({
          # now output shapefile (can be loaded in GIS software)
          ddTable <- data.frame(Id=diffMat2[,1],lon=diffMat2[,2],lat=diffMat2[,3],avgAN=diffMat2[,5])
          ddShapefile <- convert.to.shapefile(diffMat2, ddTable, "Id", 1)
          message("Writing file diffMat2 shapefile with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
          write.shapefile(ddShapefile, "diffMat2", arcgis=T)}, error = function(e){message("diffMat2 shapefile cannot be created!")
                }, warning = function(w){})
        }
      }else
      {
        while (rSel <= ts1Rows) # use diffMat1
        {
          siteMatch <- which(ts1[rSel,1] == ts2[,1]) # match siteIDs
          if(!length(siteMatch) == 0) # skip sites that do not match
            diffMat1[rSel,5] = ts1[rSel,5] - ts2[siteMatch,5] # calc diff
          rSel = rSel + 1
        }
        ind <- apply(diffMat1, 1, function(diffMat1) all(is.na(diffMat1)))
        diffMat1 <- diffMat1[ !ind, ] # remove NAs
        colnames(diffMat1)[5] = "AnDiff"
        tryCatch({write.csv(diffMat1,"avgAnTSdiff.csv",row.names=FALSE)}, error = function(e) {message("Cannot write avgAnTSdiff.csv
              to disk, file may be open!")}, warning = function(w){})
        if (outputSHP == 1)
        {
          tryCatch({
          # now output shapefile (can be loaded in GIS software)
          ddTable <- data.frame(Id=diffMat1[,1],lon=diffMat1[,2],lat=diffMat1[,3],avgAN=diffMat1[,5])
          ddShapefile <- convert.to.shapefile(diffMat1, ddTable, "Id", 1)
          message("Writing file diffMat1 shapefile with SiteID, Lon, Lat and avg anomaly for time slice to working directory...")
          write.shapefile(ddShapefile, "diffMat1", arcgis=T)}, error = function(e){message("diffMat1 shapefile cannot be created!")
                }, warning = function(w){})
        }
      }
      }, error = function(e){}, warning = function(w){})
    }else
      message("Invalid average anomaly time slice data file selected!")
  }else
    message("No average anomaly time slice data file selected!")
}

return(message("Done!"))

if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90)
{
  lonLimits <- c(-170,-40)
  latLimits <- c(15,90)
  mapOK <- TRUE
  message("Map will be created for North America")
}else
{
  lonLimits <- c(lonMax,lonMin)
  latLimits <- c(latMin,latMax)
  mapOK <- FALSE
  message("Subset of North America detected, map will not be created!")
}

while (is.na(interpMapYN) | interpMapYN < 1 | interpMapYN > 2)
  interpMapYN=as.numeric(readline("Do you want to interpolate the output with kriging and then create a map? <1 for YES, 2 for NO>
        "))

if (interpMapYN == 1)
{

  # check whether we need to map anTS, diffMat1 or diffMat2 (can check is.null for diffMat1 and 2)
```

```
if (is.null(diffMat1) & is.null(diffMat2))
   selAvgAnTS = anByTimeSlice3

if (!is.null(diffMat1))
   selAvgAnTS = diffMat1

if (!is.null(diffMat2))
   selAvgAnTS = diffMat2

df1=data.frame(lat=selAvgAnTS[,3],long=selAvgAnTS[,2],avgAN=selAvgAnTS[,4])
initial.df=df1

# ask user if they want to plot a map of point anomalies?
tryCatch({
  selAvgAnTSneg <- subset(selAvgAnTS, selAvgAnTS[,4] < -0.000001)
  selAvgAnTSzero <- subset(selAvgAnTS, selAvgAnTS[,4] >= -0.000001 & selAvgAnTS[,4] <= 0.000001 )
  selAvgAnTSpos <- subset(selAvgAnTS, selAvgAnTS[,4] > 0.000001)

  map(database = "world",xlim = lonLimits,ylim = latLimits)
  map.axes()

  points(selAvgAnTSneg$Lon,selAvgAnTSneg$Lat,pch=21,col="blue",cex=1)
  points(selAvgAnTSzero$Lon,selAvgAnTSzero$Lat,pch=21,col="darkgrey",cex=1)
  points(selAvgAnTSpos$Lon,selAvgAnTSpos$Lat,pch=21,col="red",cex=1)
  legend("bottomleft",legend=c("Positive","Zero","Negative"),pch =c(21,2),col=c("red","darkgrey","blue"))

  message(paste("There are ",length(unique(selAvgAnTSneg[,1])),"  sites with negative anomalies",sep=" "))
  message(paste("There are ",length(unique(selAvgAnTSzero[,1])),"  sites with zero anomalies",sep=" "))
  message(paste("There are ",length(unique(selAvgAnTSpos[,1])),"  sites with positive anomalies",sep=" "))

  message(paste("Max average anomaly difference value is",max(selAvgAnTS[,4])))
  message(paste("Min average anomaly difference value is",min(selAvgAnTS[,4])))

}, error = function(e){message("Problem creating map of average anomalies by time slice!");mapOK = FALSE }, warning = function(w)
      {})


if (mapOK)
{
  # ask user for zlim based on the following
  while (is.na(zLimVal) | zLimVal == 0)
     zLimVal=as.numeric(readline("What is the max/min anomaly range for map? (e.g. if 1 is entered, max will be 1, min will be -1)
           Should be based on min/max values above or plot could fail!) "))

  # now convert to spatialPoints object to reproject data, then do kriging, the plot the interpolated map

  coordinates(df1)=~long+lat
  proj4string(df1)="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
  # project_df=spTransform(df1, CRS("+proj=eqdc +lat_0=45 +lon_0=-100 +lat_1=43.33 +lat_2=66.66 +x_0=0 +y_0=0 +ellps=GRS80 +datum=
       NAD83 +units=m +no_defs +towgs84=0,0,0"))
  project_df=spTransform(df1, CRS("+proj=aea +lat_0=40 +lon_0=-96 +lat_1=20 +lat_2=60 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +
       units=m +no_defs +towgs84=0,0,0"))

  # would be nice to specify search radius & possibly other parameters for kriging
  kr=autoKrige(avgAN~1,project_df)
  kr.output=as.data.frame(kr$krige_output)

  while (is.null(plotTitle))
  {
     plotTitle=readline("Enter the title/file name for this plot: ")
  }

  fileName = plotTitle
  fileNameOut = paste(fileName,".pdf")
  fileNameOut = gsub(" ","", fileNameOut ,fixed=TRUE)

  # ask user if they want a gridded map or filled contour map?
  while (is.na(mapType) | mapType < 1 | mapType > 2)
  {
     mapType=as.numeric(readline("Do you want a gridded or filled contour map? <1 for GRIDDED, 2 for FILLED CONTOUR> "))
  }
  while (is.na(levCol) | levCol < 10 | levCol > 100)
  {
     levCol = as.numeric(readline("How many colour levels should the output map have, must be between 10 and 100 (use higher values
            for a less blocky appearance): "))
```

```
    levCol = levCol %/% 10 * 10
  }
  if (mapType == 2)
  {
    tryCatch({
      pdf(fileNameOut)
      my.image <- image.maker(kr.output[,c("x1","x2")],kr.output$var1.pred)
      if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
        filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
            beige", "maroon"))(levCol),plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
      else
        filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
            beige", "maroon"))(levCol),plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
      dev.off()
    }, error = function(e){message(paste(fileNameOut,"Error creating contour map, possibly file is already open!"))}, warning =
         function(w){})
  }else
  {
    # ask user for grid cell resolution (i.e. number of rows & cols - use higher values for smaller grid boxes)
    while (is.na(gCellRC) | gCellRC < 10 | gCellRC > 100)
    {
      gCellRC = as.numeric(readline("How many rows/columns should the gridded map have, must be between 10 and 100 (use higher
           values for smaller grid boxes): "))
      gCellRC = gCellRC %/% 10 * 10
    }
    coordinates(kr.output) <- ~x1+x2
    rast <- raster(ncol=gCellRC,nrow=gCellRC)
    extent(rast) <- extent(kr.output)
    rmap <- rasterize(kr.output, rast, kr.output$var1.pred, fun = mean)
    # tryCatch({
      pdf(fileNameOut)
      if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
        plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
            levCol),alpha = 0.8,plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
      else
        plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
            levCol),alpha = 0.8,plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
      dev.off()
    # }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning = function(
        w){})
  }
}

if (mapOK)
{
# now call another function (optional) for sequential time slice difference (dynamic) mapping
while (is.na(dynMapYN) | dynMapYN < 1 | dynMapYN > 2)
  dynMapYN=as.numeric(readline("Do you want to do mapping of avg anomalies for sequential time intervals (oldest to youngest)? <1
       for YES, 2 for NO> "))

if (dynMapYN == 1)
{

  source("image.maker.R")

  fileName = NULL
  diffMat1 = NULL
  diffMat2 = NULL
  selAvgAnTS = NULL
  my.image = NULL
  plotTitle = NULL
  ts1 = NULL
  ts2 = NULL

  siteMatch = 0
  zLimVal = 0
  mapType = 0
  gCellRC = 0
  levCol = 0
  rSel = 1

  calYrList = unique(spatialJoinFile[,4])
  calYrLast = length(calYrList)
  calYrLast2 = calYrLast - 1
  calYrInt = calYrList[calYrLast] - calYrList[calYrLast2] # determine interval
```

```
intervalStart = min(calYrList) # finish with youngest year
selMapYear2 = max(calYrList) # start with oldest year
selMapYear1 = selMapYear2 - calYrInt # one interval less than selMapYear2

while (selMapYear2 >= intervalStart & selMapYear1 >= intervalStart)
{
   message(paste("Processing average anomaly difference between",selMapYear2,"and",selMapYear1,"...."))

   # only get rows where calMean value falls within the range of the time slice specified above
   anByTimeSlice1 <- spatialJoinFile[spatialJoinFile$CalMean == selMapYear2, ]
   anByTimeSliceA <- spatialJoinFile[spatialJoinFile$CalMean == selMapYear1, ]

   # avg the anomalies (for cal Yr BP interval range) for each site
   # make sure to get lat and lon values for each site

   anByTimeSlice2 <- aggregate(anByTimeSlice1[,10], list(anByTimeSlice1[,4],anByTimeSlice1[,1],anByTimeSlice1[,2],anByTimeSlice1
         [,3]), mean)
   anByTimeSlice3 <- aggregate(anByTimeSlice2[,5], list(anByTimeSlice2[,2],anByTimeSlice2[,3],anByTimeSlice2[,4]), mean)

   anByTimeSliceB <- aggregate(anByTimeSliceA[,10], list(anByTimeSliceA[,4],anByTimeSliceA[,1],anByTimeSliceA[,2],anByTimeSliceA
         [,3]), mean)
   anByTimeSliceC <- aggregate(anByTimeSliceB[,5], list(anByTimeSliceB[,2],anByTimeSliceB[,3],anByTimeSliceB[,4]), mean)

   # fix column headers here
   colnames(anByTimeSlice3)[1] <- "Site"
   colnames(anByTimeSlice3)[2] <- "Lon"
   colnames(anByTimeSlice3)[3] <- "Lat"
   colnames(anByTimeSlice3)[4] <- "avgAnTS"

   colnames(anByTimeSliceC)[1] <- "Site"
   colnames(anByTimeSliceC)[2] <- "Lon"
   colnames(anByTimeSliceC)[3] <- "Lat"
   colnames(anByTimeSliceC)[4] <- "avgAnTS"

   ts2 = anByTimeSlice3
   ts1 = anByTimeSliceC

   tryCatch({write.csv(ts2,"ts2.csv",row.names=FALSE)}, error = function(e) {message("Cannot write ts2.csv to disk, file may be
         open!")}, warning = function(w){})
   tryCatch({write.csv(ts1,"ts1.csv",row.names=FALSE)}, error = function(e) {message("Cannot write ts1.csv to disk, file may be
         open!")}, warning = function(w){})

   # anomalies are ts2 relative to ts1
   # when comparing matrices, always use the one with the lesser # of rows

   ts1Rows <- nrow(ts1)
   ts2Rows <- nrow(ts2)

   # print(ts1Rows)
   # print(ts2Rows)

   if (ts1Rows >= ts2Rows)
   {
      diffMat2 <- ts2
   }else # ts1Rows < ts2Rows
   {
      diffMat1 <- ts1
   }

   if (is.null(diffMat1)) # use diffMat2
   {
     while (rSel <= ts2Rows)
     {
        siteMatch <- which(ts2[rSel,1] == ts1[,1]) # match siteIDs
        if(!length(siteMatch) == 0) # skip sites that do not match
           diffMat2[rSel,4] = ts2[rSel,4] - ts1[siteMatch,4] # calc diff
        rSel = rSel + 1
     }
     ind <- apply(diffMat2, 1, function(diffMat2) all(is.na(diffMat2)))
     diffMat2 <- diffMat2[ !ind, ] # remove NAs
     colnames(diffMat2)[4] = "AnDiff"
   }else
   {
     while (rSel <= ts1Rows) # use diffMat1
     {
        siteMatch <- which(ts1[rSel,1] == ts2[,1]) # match siteIDs
```

```
     if(!length(siteMatch) == 0) # skip sites that do not match
        diffMat1[rSel,4] = ts1[rSel,4] - ts2[siteMatch,4] # calc diff
     rSel = rSel + 1
   }
   ind <- apply(diffMat1, 1, function(diffMat1) all(is.na(diffMat1)))
   diffMat1 <- diffMat1[ !ind, ] # remove NAs
   colnames(diffMat1)[4] = "AnDiff"
 }

# check whether we need to select diffMat1 or diffMat2 (can check is.null for diffMat1 and 2)
if (!is.null(diffMat1))
   selAvgAnTS = diffMat1
else
   selAvgAnTS = diffMat2

message(paste("Max average anomaly difference value is",max(selAvgAnTS[,4])))
message(paste("Min average anomaly difference value is",min(selAvgAnTS[,4])))

# ask user for zlim based on the following
while (is.na(zLimVal) | zLimVal == 0)
   zLimVal=as.numeric(readline("What is the max/min range for map anomalies? (e.g. if 1 is entered, max will be 1, min will be
         -1). Should be based on min/max values above or plot could fail!) "))

 # now convert to spatialPoints object to reproject data, then do kriging, the plot the interpolated map
 df1=data.frame(lat=selAvgAnTS[,3],long=selAvgAnTS[,2],avgAN=selAvgAnTS[,4])
 initial.df=df1
 coordinates(df1)=~long+lat
 proj4string(df1)="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
 project_df=spTransform(df1, CRS("+proj=aea +lat_0=40 +lon_0=-96 +lat_1=20 +lat_2=60 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +
       units=m +no_defs +towgs84=0,0,0"))

 # would be nice to specify search radius & possibly other parameters for kriging
 kr=autoKrige(avgAN~1,project_df)
 kr.output=as.data.frame(kr$krige_output)

 plotTitle=paste("Average anomaly for",selMapYear2,"-",selMapYear1,"bp")
 fileName <- paste("AvgAnom_",selMapYear2,"-",selMapYear1,"_bp.pdf")
 fileNameOut = gsub(" ","", fileName ,fixed=TRUE)

  # ask user if they want a gridded map or filled contour map?
  while (is.na(mapType) | mapType < 1 | mapType > 2)
  {
     mapType=as.numeric(readline("Do you want a gridded or filled contour map? <1 for GRIDDED, 2 for FILLED CONTOUR> "))
  }
  while (is.na(levCol) | levCol < 10 | levCol > 100)
  {
     levCol = as.numeric(readline("How many colour levels should the output map have, must be between 10 and 100 (use higher
           values for a less blocky appearance): "))
     levCol = levCol %/% 10 * 10
  }
  if (mapType == 2)
  {
     my.image <- image.maker(kr.output[,c("x1", "x2")], kr.output$var1.pred)
     tryCatch({
     pdf(fileNameOut)
     if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
        filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
              beige", "maroon"))(levCol),plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
     else
        filled.contour(my.image,zlim=c(-1*zLimVal,zLimVal),nlevels=levCol,main=plotTitle,col=colorRampPalette(c("darkblue", "
              beige", "maroon"))(levCol),plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
     dev.off()
     }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning =
           function(w){})
  }else
  {
     # ask user for grid cell resolution (i.e. number of rows & cols - use higher values for smaller grid boxes)
     while (is.na(gCellRC) | gCellRC < 10 | gCellRC > 100)
     {
        gCellRC = as.numeric(readline("How many rows/columns should the gridded map have, must be between 10 and 100 (use higher
              values for smaller grid boxes): "))
        gCellRC = gCellRC %/% 10 * 10
     }

     coordinates(kr.output) <- ~x1+x2
     rast <- raster(ncol=gCellRC,nrow=gCellRC)
```

```
            extent(rast) <- extent(kr.output)
            rmap <- rasterize(kr.output, rast, kr.output$var1.pred, fun = mean)
            # tryCatch({
            pdf(fileNameOut)
            if (lonMax == -170 & lonMin == -40 & latMin == 15 & latMax == 90) # adjust if North America
               plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
                     levCol),alpha = 0.8,plot.axes={par(usr=c(-145,-55,25,75));map(add=T)})
            else
               plot(rmap,axes=F,asp=1,zlim=c(-1*zLimVal,zLimVal),main=plotTitle,col=colorRampPalette(c("darkblue", "beige", "maroon"))(
                     levCol),alpha = 0.8,plot.axes={par(usr=c(lonMax,lonMin,latMin,latMax));map(add=T)})
            dev.off()
            # }, error = function(e){message(paste(fileNameOut,"cannot be written to disk, please close the file!"))}, warning =
                function(w){})
          }

          rSel = 1

          diffMat1 = NULL
          diffMat2 = NULL

          selMapYear2 = selMapYear2 - calYrInt
          selMapYear1 = selMapYear1 - calYrInt
       }
      }
    }
  }
  aList <- list(anByTimeSliceC, anByTimeSlice3)
  return(aList)
}
#########################################################################################################################
       pickClimateOption <- function()
{

  message("")
  message("Examples of climate station data are BEST and GHCN [using the R package BerkeleyEarth or RGHCNv3]")
  message("Processing of the data can take some time depending on memory and the subset needed")
  message("BEST files need to be already downloaded to disk before running option 1")
  message("")
  message("Examples of climate re-analysis data are NCEP and NARR [must be in NetCDF (.nc) format]")
  message("Re-analysis data must be a long term or monthly mean file (e.g. varname.mon.ltm.nc)")
  message("Regular monthly mean files (e.g. varname.mon.mean.nc) have not been tested yet!")
  message("")
  message("Example of climate raster data is WorldClim [other raster formats such as .asc have not been tested yet!]")
  message("Directory with Worlclim data must have .bil and .hdr file for each climate variable (e.g. tmean7 for July temp)")
  message("")
  message("Examples of other custom climate data in .csv format are New et al.(2002) [based on CRU] or ERA-40 re-analysis [converted
        from grib format]")
  message("The data must be converted by the user and have the following fields: latitude, longitude & value of climate variable (
        specified by user)")
  message("")

  valid <- FALSE
  while(!valid)
  {
    cat("Do you want to use station, re-analysis, raster or other data for custom climate data? <1 for STATION, 2 for RE-ANALYSIS, 3
         for RASTER, 4 for OTHER>","\n")
    Answer <- readLines(n=1)
    if(Answer %in% c(1,2,3,4))valid = TRUE
  }

  return(Answer)
}

plotAvgAnomalies <- function(anByCalYr2,anByCalYrTuned,tunedYN,selClimVar,firstYear,lastYear)
{
  # create a t-s plot of anomalies at interpolated intervals through time & save to file

  anUnits = ""
  strYlab = ""
  strXlab = ""

  if (!tunedYN) # not tuned - uses cal yr BP dates
  {
    anUnits <- readline("Please enter the units for the y-axis (e.g. for ANNP enter 'mm', for MTWA enter 'C') ")
    strYlab = paste(selClimVar, "anomaly (in",anUnits,")")
```

```
    # determine max and min anomaly & std error values (y-axis) for plot
    minYlist = min(anByCalYr2[,2])
    maxYlist = max(anByCalYr2[,2])
    maxError = max(anByCalYr2[,3])
    minY = unique(minYlist)
    maxY = unique(maxYlist)
    maxErr = unique(maxError)
    maxY = maxY + maxErr # ensure a buffer for maxY
    minY = minY - maxErr # ensure a buffer for minY

    message("Writing plot to file avgAnomalyPlot.pdf in working directory...")
    tryCatch(
    {
      pdf("avgAnomalyPlot.pdf")
      if (firstYear < 0) # plots always start at 0 bp or older
         firstYear = 0
      plot(anByCalYr2$CalMean, anByCalYr2$AvgAnom, type="l", xlim=c(lastYear,firstYear),ylim=c(minY,maxY),xlab="Cal year BP", ylab=
            strYlab, col="black", lwd=3)

      lines(anByCalYr2$CalMean, anByCalYr2$AvgAnom + anByCalYr2$StdErr, col="gray")
      lines(anByCalYr2$CalMean, anByCalYr2$AvgAnom - anByCalYr2$StdErr, col="gray")
      # abline(h=0,col="lightgray")
      dev.off()
    }, error = function(e){message("avgAnomalyPlot.pdf cannot be written to disk, please close the file!")}, warning = function(w){})

    # message("Writing file avgAnomalyByCalYrBP.csv to working directory...")
    # tryCatch({write.csv(anByCalYr2,"avgAnomalyByCalYrBP.csv",row.names=FALSE)}, error = function(e){message("avgAnomalyByCalYrBP.csv
          cannot be written to disk, please close the file!")}, warning = function(w){})

  }else # tuned - use year AD dates (or year BC) specified by user or default 2k tuned dates
  {
    anUnits <- readline("Please enter the units for the y-axis (e.g. for ANNP enter 'mm', for MTWA enter 'C') ")
    strYlab = paste(selClimVar, "anomaly (in",anUnits,")")

    # determine max and min anomaly & std error values (y-axis) for plot
    minYlist = min(anByCalYrTuned[,2])
    maxYlist = max(anByCalYrTuned[,2])
    maxError = max(anByCalYrTuned[,3])
    minY = unique(minYlist)
    maxY = unique(maxYlist)
    maxErr = unique(maxError)
    maxY = maxY + maxErr # ensure a buffer for maxY
    minY = minY - maxErr # ensure a buffer for minY

    message("Writing plot to file avgAnomalyPlotTuned.pdf in working directory...")
    tryCatch(
    {
      pdf("avgAnomalyPlotTuned.pdf")
      if (lastYear < 0) # using both AD and BC dates
         strXlab="Year BC/AD"
      else
         strXlab="Year AD"

      plot(anByCalYrTuned$CalMean, anByCalYrTuned$AvgAnom, type="l", xlim=c(lastYear,firstYear),ylim=c(minY,maxY),xlab=strXlab, ylab=
            strYlab, col="black", lwd=3)
      lines(anByCalYrTuned$CalMean, anByCalYrTuned$AvgAnom + anByCalYrTuned$StdErr, col="gray")
      lines(anByCalYrTuned$CalMean, anByCalYrTuned$AvgAnom - anByCalYrTuned$StdErr, col="gray")
      # abline(h=0,col="lightgray")
      dev.off()
    }, error = function(e){message("avgAnomalyPlot.pdf cannot be written to disk, please close the file!")}, warning = function(w){})

    message("Writing file avgAnomalyByTunedYearAD.csv to working directory...")
    tryCatch({write.csv(anByCalYrTuned,"avgAnomalyByTunedYearAD.csv",row.names=FALSE)}, error = function(e){message("
          avgAnomalyByTunedYearAD.csv cannot be written to disk, please close the file!")}, warning = function(w){})
  }
}

plotAvgAnomaliesByEcoregion <- function(params=ECOREGIONS,anByEcoregionFinal,selClimVar,oldYrBP,youngYrBP,tuned)
{
  # NOTE: should have map of eco-regions in help/doc file
  # 9 plots for each panel (if all ecoregions selected) & format the plots nicely

  tryCatch({
    if (!tuned)
       pdf("avgAnomalyPlotBySelEcoregions.pdf")
    else
```

```
   pdf("avgAnomalyPlotByEcoregionsTuned.pdf")

par(mfrow=c(3,3))
# these values can be adjusted
par(mar = c(2.5, 3.5, 2, 1))
par(mgp = c(1.5, 0.5, 0))
par(oma = c(10, 1, 1, 1))

plotIndex = 2 # first anomaly col
plotStdErrInd = 3 # first std error col
# Mediterranian ecoregion is excluded because of NA std error values
plotIndexMax = ncol(anByEcoregionFinal)
ERcount = 1 # keep track of ecoregions

strXlabER = "Cal Yr BP"
strYlab = ""
strYlabER = ""

useEcoregion = 0

anUnits <- readline("Please enter the units for the y-axis (e.g. for ANNP enter 'mm', for MTWA enter 'C') ")
strYlab = paste(selClimVar, "anomaly(", anUnits,")")

while (plotIndex <= plotIndexMax & ERcount <= 9)
{
   useEcoregion = 0 # reset this
   selColName <- ECOREGIONS[ERcount,2] # get selected eco-region name from list

   # here is where the user can optionally exclude eco-regions
   while (is.na(useEcoregion) | useEcoregion < 1 | useEcoregion > 2)
     useEcoregion <- as.numeric(readline(paste("Do you wish to include ecoregion",selColName,"<1 for yes, 2 for no> ")))

   if (useEcoregion == 1)
   {
     # determine max and min anomaly values (y-axis) for plot
      minYlist = min(anByEcoregionFinal[,plotIndex])
      maxYlist = max(anByEcoregionFinal[,plotIndex])
      maxError = max(anByEcoregionFinal[,plotStdErrInd])
      minY = unique(minYlist)
      maxY = unique(maxYlist)
      maxErr = unique(maxError)
      maxY = maxY + maxErr # ensure a buffer for maxY
      minY = minY - maxErr # ensure a buffer for minY

     # include eco-region name in Y axis label
     # strYlabER = paste("anomaly for",selColName,sep=" ")

     if (tuned)
     {
        if (oldYrBP < 0) # using both AD and BC dates
           strXlabER="Year BC/AD"
        else
           strXlabER="Year AD"
     }else
     {
        if (youngYrBP < 0) # plots always start at 0 cal yr BP for non-tuned plots
          youngYrBP = 0
     }
     # x-axis of plot should be from oldest cal yr BP (left) to youngest cal yr BP (right)
     # plot(anByEcoregionFinal$CalMean,anByEcoregionFinal[,plotIndex],xlim=c(oldYrBP,youngYrBP),type="l",xlab=strXlabER,ylab=
           strYlabER,col="black",lwd=3)
     plot(anByEcoregionFinal$CalMean,anByEcoregionFinal[,plotIndex],xlim=c(oldYrBP,youngYrBP),ylim=c(minY,maxY),type="l",xlab=
           strXlabER,ylab=selColName,col="black",lwd=3)
     lines(anByEcoregionFinal$CalMean,anByEcoregionFinal[,plotIndex] + anByEcoregionFinal[,plotStdErrInd], col="gray")
     lines(anByEcoregionFinal$CalMean,anByEcoregionFinal[,plotIndex] - anByEcoregionFinal[,plotStdErrInd], col="gray")
     # abline(h=0,col="lightgray")
   }
   plotIndex = plotIndex + 2
   plotStdErrInd = plotStdErrInd + 2
   ERcount = ERcount + 1
}
title(main = strYlab)
if (!tuned)
   message("Plot output to file avgAnomalyPlotBySelEcoregions.pdf in working directory!")
else
   message("Plot output to file avgAnomalyPlotByEcoregionsTuned.pdf in working directory!")
```

```
    dev.off()
  }, error = function(e){message("Error in creating plot by ecoregion, possibly you selected ecoregions that do not have sites at
        certain time intervals!")}, warning = function(w){})
}plotSCDthroughTime <- function(reconSelAnalogSCDcol,reconSelAnalogCalYRcol)
{
   scdCutoffVal = -99 # default value

   # ask user SCD value (e.g. 0.3) where vertical line will be placed (optional)
   while (is.na(scdCutoffVal) | (scdCutoffVal < 0 & scdCutoffVal != -1) | scdCutoffVal > 1)
     scdCutoffVal=as.numeric(readline("At what SCD value do you want the vertical line placed? (valid range is 0 to 1, use -1 for no
           line) "))

   message("Writing plot to file SCD_allregions.pdf in working directory...")

   tryCatch(
   {
      pdf("SCD_allregions_plot.pdf")
      plot(reconSelAnalogSCDcol / 100,reconSelAnalogCalYRcol,pch=21,cex=0.1, bg="red", ylab="Cal yr BP", xlab="SCD value")
      if (scdCutoffVal != -1)
         abline(v=scdCutoffVal,col="gray")
      dev.off()
   }, error = function(e){message("SCD_allregions.pdf cannot be written to disk, please close the file!")}, warning = function(w){})

}plotSiteDensityThroughTime <- function(spJoinSubset)
{
   require(plyr)

   # get number of sites for each cal Yr BP interval
   counts <- ddply(spJoinSubset, .(spJoinSubset$CalMean), nrow)

   names(counts) <- c("calYrBP", "siteCount")

   message("Writing file siteDensity.csv to working directory...")
   tryCatch({write.csv(counts,"siteDensity.csv",row.names=FALSE)}, error = function(e){message("siteDensity.csv cannot be written to
         disk, please close the file!")}, warning = function(w){})

   maxY=unique(max(counts$siteCount)) + 100 # need a buffer for # of sites in plot

   message("Writing plot to file siteDensity.pdf in working directory...")

   tryCatch(
   {
      pdf("siteDensityPlot.pdf")
      barplot(counts$siteCount,main="Pollen site density through time",horiz=FALSE,xlab="Cal Year BP",ylim=c(0,maxY),ylab="# of sites",
            names.arg=counts$calYrBP)
      dev.off()
   }, error = function(e){message("siteDensityPlot.pdf cannot be written to disk, please close the file!")}, warning = function(w){})
}

readPollenDbFile <- function(pollenSum,csv)
{
  filename <- csv$filenames[which(csv$pSum == pollenSum)]

  # read the csv
  # is there a directory? add it to the name
  # data <- read.csv(filename)

  return(filename)

}refreshMainMenu1 <- function()
{
  message("")
  message
        ("-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

  message("-------------------- POLLEN DATABASE EDITOR & RECONSTRUCTION BUILDER
        ----------------------------------------------------------------------------------------------------")
  message
        ("-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------


  message("[1] Create a custom pollen database file (optional)")
  # this menu item can be run at any time, any menu item can be run after this
  # menu item #6 should be run after this
  # variables that are set in this menu item:
```

```
# psCustomModernDB, psCustomFossilDB, psCustomTaxaCount, confirmWrite, modern, fossil
# functions that are called: customPollenSum.r

message("[2] Load a pollen db file (default pollen sums available of 35,64,134,135 (Williams and Shuman, 2008) or 0 for custom pollen
       sum as created in [1] - default pollen sum is 135)")
# this menu item can be run at any time, any menu item can be run after this
# menu item #6 should be run after this
# variables that are set in this menu item:
# pSum, newPS, modern, modernData, modTaxaRange, fossil, fossilData, fosTaxaRange,lastCol, psCount
# functions that are called: readPollenDbFile.r, getTaxaColumns.r

message("[3] Show which fossil and modern pollen databases are currently loaded (optional)")
# this menu item can be run at any time, any menu item can be run after this
# menu item #6 should be run after this
# variables that are set in this menu item: N/A
# functions that are called: N/A

message("[4] Show list of pollen taxa in currently loaded modern & fossil pollen database (optional)")
# this menu item can be run at any time, any menu item can be run after this
# menu item #6 should be run after this
# variables that are set in this menu item: N/A
# functions that are called: listPollenTaxaInLoadedDB.r

message("[5] Join custom climate data to modern pollen sites (default - Whitmore et al., 2005 v1.7.2 - optional: STATION, RE-ANALYSIS
      , RASTER or OTHER data can be imported by user)")
# this menu item can be run at any time, any menu item can be run after this
# menu item #8 should be run after this
# variables that are set in this menu item:
# modClimOption, modernData, modernDataNew
# functions that are called: pickClimateOption.r, customClimateOption.r

message("[6] Select a climate variable to reconstruct (required - from modern climate database or new variable from custom climate
      data - optional )")
# this menu item can be run at any time, any menu item can be run after this
# menu item #8 should be run after this
# variables that are set in this menu item: selClimVar, climColIndex
# functions that are called: selectClimateVariable.r

message("[7] Display which climate variable from modern database is selected (if pollen db is changed, this needs to be run again)")
# this menu item can be run at any time, any menu item can be run after this
# menu item #8 should be run after this
# variables that are set in this menu item: N/A
# functions that are called: N/A

message("[8] Do reconstruction of selected climate variable (using MATTOOLS v1.1 from CRAN - .recon file created can be loaded in
      module 2)")
# this menu item can be run at any time, any menu item can be run after this
# variables that are set in this menu item: analogRange, NA_recons_SCD, NA_recons_Obj
# functions that are called: getAnalogCount.r, createReconObj.r, writeReconObj.r, mat.dissim.r (from MATTOOLS v.1)

message("[9] Create a map of pollen sites from database (optional: fossil, modern or both)")
# this menu item can be run at any time after running menu item #2, any menu item can be run after this
# variables that are set in this menu item: option
# functions that are called: mapPollenSites.r

message("[10] List all program files in working directory (optional)")
# this menu item can be run at any time, no variables are set here
# functions that are called: listAllFiles.r

message("[11] Set or change the current working directory (optional)")
# this menu item can be run at any time, the workingDirectory variable is set here
# functions that are called: setWorkingDirectory.r

message("[12] Start climate reconstruction analysis program (optional)")
# this menu item can be run at any time, no variables are set here
# functions that are called: MainProgram2.r

message("[13] Quit program and return to R command prompt")
# this menu item can be run at any time, no variables are set here
# functions that are called: N/A

message
      ("-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------


### END OF REFRESH MAIN MENU 1 ###
```

```
}refreshMainMenu2 <- function()
{
  message("")
  message
        ("-------------------
```

```
  # tunedAnomByCalYr, startYr, endYr
  # functions that are called: plotAvgAnomalies.r, tuneYearAD.r

  message("[10] Plot average anomalies by calendar year BP by ecoregion (default is Fedorova et al., 1994, no options currently
        available)")
  # this menu item should be run after running menu item #9, any menu item can be run after this
  # variables that are set in this menu item: N/A
  # functions that are called: plotAvgAnomaliesByEcoregion.r

  message("[11] Get average anomalies for all sites by time slice (optional: output csv file, shapefile & create maps)")
  # this menu item can be run at any time after running menu item #8, any menu item can be run after this
  # but menu item #12 should be run after this
  # variables that are set in this menu item: reconTSList, reconTS, Var
  # functions that are called: outputAnomaliesByTimeSlice.r

  message("[12] Create Squared-Chored Distance (SCD) plots by ecoregion (optional: same ecoregions as menu item 10)")
  # this menu item can be run at any time after running menu item #1
  # any menu item can be run after this, menu item #14 should be run after this
  # variables that are set in this menu item: analogTop, analogBottom, numAnalogs
  # SCDlist, reconSelAnalogSCDcol, reconSelAnalogCalYRcol
  # functions that are called: createSCDplotsByEcoregion.r

  message("[13] Plot SCD values from reconstruction file through time for all regions (optional)")
  # this menu item can be run at any time after running menu item #6, any menu item can be run after this
  # variables that are set in this menu item: N/A
  # functions that are called: N/A

  message("[14] Calculate and plot site density of reconstruction through time (optional)")
  # this menu item can be run at any time after running menu item #7, any menu item can be run after this
  # variables that are set in this menu item: N/A
  # functions that are called: plotSiteDensitySamplesThroughTime.r

  message("[15] Spatial filtering of reconstruction based on calculated local Moran's I and z-values (optional)")
  # this menu item can be run at any time after running menu item #1, any menu item can be run after this
  # variables that are set in this menu item: N/A
  # functions that are called: spatialFiltering.r

  message("[16] Perform a data-data or data-model comparison (beta-version, functionality in development)")
  # this menu item can be run at any time after running menu item #11, any menu item can be run after this
  # variables that are set in this menu item: N/A
  # functions that are called: dataModelComparison.r

  message("[17] Set or change the current working directory (optional)")
  # this menu item can be run at any time, no variables are set here

  message("[18] List all program files (.pdf, .csv, .recon) in working directory (optional)")
  # this menu item can be run at any time, no variables are set here

  message("[19] Start pollen database editor program (optional)")
  # this menu item can be run at any time, no variables are set here

  message("[20] Quit program and return to R command prompt")
  # this menu item can be run at any time, no variables are set here

  message
        ("----------------------------------------------------------------------------------------------------------------------------------------------------------------


  ### END OF REFRESH MAIN MENU ###
}selectClimateVariable <- function(params=CLIMATE_VARIABLES)
{
  valid <- FALSE
  for(i in 1:nrow(params)){
    cat(i,":",params[i,3],"\n")
  }
  while (!valid)
  {
    climVar<- readline("Enter the number of the climate variables to be reconstructed from above: ")

    if (climVar %in% params$Id) valid<-TRUE
  }
  return (as.numeric(climVar ))
}selectiveRead <- function(Directory, Month, file ="data.bin"){


  D <- attach.big.matrix(dget(file.path(Directory, "data.desc", fsep=.Platform$file.sep)),
```

```
                        path = Directory)


  lowerBound <- (Month-1)/12
  upperBound <- Month/12
  dex <- which((D[,3] %% 1) > lowerBound & (D[,3] %% 1)< upperBound)
  return(D[dex,c(1,3,4)])

}selectPollenSum <- function(POLLEN_SUMS)
{
  valid <- FALSE
  while(!valid){
     cat("Which pollen sum do you wish to use? (35, 64, 134, 135 or 0 for custom, if choosing a new sum need to reload modern & fossil
           databases) ","\n")
     val <- as.numeric(readline())
     if(!(val %in% POLLEN_SUMS) | is.na(val)) cat("Choose a pollen sum from the list:","\n")
     else valid <- TRUE
   }
  return(val)
}selectReconFile <- function()
{
  # Allow only 1 variable at a time to be reconstructed
  # Input data set would be for example, PS64_RECONS_MIPT.csv
  # or PS_CUSTOM_5_MTWA.csv for a custom file

  strMsg = ""
  newPS = ""
  strRep = NULL
  climStrRep1 = NULL
  climStrRep2 = NULL
  psStr = NULL
  myList <- NULL
  climVarSel <- NULL
  climVarTemp <- NULL
  selClimVar <- NULL
  custClimVar <- NULL
  linkReconInput <- NA
  fileTypeMatch = -1

  tryCatch({
     linkReconInput <- file.choose()
     }, error = function(ex) {
  })
  if(!is.na(linkReconInput))
  {
    # check for valid reconstruction file here
    fileTypeMatch <- regexpr(".recon",basename(linkReconInput))

    ps35Match <- regexpr("35",basename(linkReconInput))
    ps64Match <- regexpr("64",basename(linkReconInput))
    ps134Match <- regexpr("134",basename(linkReconInput))
    ps135Match <- regexpr("135",basename(linkReconInput))
    psCustomMatch <- regexpr("CUSTOM",basename(linkReconInput))

    # determine pollen sum from file name
    if (ps35Match != -1)
      psStr = "35"

    if (ps64Match != -1)
      psStr = "64"

    if (ps134Match != -1)
      psStr = "134"

    if (ps135Match != -1)
      psStr = "135"

    if (psCustomMatch != -1)
      psStr = "CUSTOM"

    if (fileTypeMatch != -1)
    {
    if (is.null(psStr))
    {
      message("Could not determine the pollen sum from reconstruction file name!")
      while (newPS != "35" & newPS != "64" & newPS != "134" & newPS != "135" & newPS != "0")
```

```
    newPS=readline("Please enter either 35, 64, 134, 135 or 0 for custom pollen sum: ")
  psStr = newPS
}else
{
  strMsg = paste("The following reconstruction file is loaded: ",basename(linkReconInput))
  message(strMsg)
  message(paste("Pollen sum for selected reconstruction file is", psStr,sep=" "))
}

  reconInput <- read.csv(linkReconInput, TRUE, ",")
  colnames(reconInput) <- toupper(colnames(reconInput))

  # get rid of any rows with NA cal yr bp dates
  reconInput <- reconInput[!is.na(reconInput[,7]),]

  # cal yr bp is col 7
  # check if cal yr bp is < 0
  # if yes, set cal yr bp to 0

  #lastRow <- nrow(reconInput)
  #rowInd = 1

  #while (rowInd <= lastRow)
  #{
  # if (reconInput[rowInd,7] < 0)
  # reconInput[rowInd,7] = 0
  # rowInd = rowInd + 1
  #}

  # determine selected climate variable from recon file loaded
  Flags <- mapply(FUN = grepl,CLIMATE_VARIABLES[,2],
                      MoreArgs = list(x=linkReconInput,ignore.case = TRUE))

  selClimVar <- toupper(names(Flags[Flags==TRUE]))

  # if custom variable, will not be in the constants file, so get varname from filename
  if (!length(selClimVar))
  {
      strRep = paste("PS",psStr,"_",sep="")
      climStrRep1 = gsub(strRep,"", basename(linkReconInput),fixed=TRUE)
      climStrRep2 = gsub(".recon","", climStrRep1 ,fixed=TRUE)
      climVarSel <- climStrRep2
  }else
     climVarSel <- selClimVar

  # check if this varname from filename is actually in the recon file
  k=1
  climVarMatch <- NULL
  while (k <= ncol(reconInput))
  {
     climVarMatch <- regexpr(climVarSel,colnames(reconInput)[k])
     if (climVarMatch > 0)
     {
        climVarTemp = colnames(reconInput)[k]
        custClimVar = gsub(1,"", climVarTemp ,fixed=TRUE)
        k = 999
     }
     k = k + 1
  }

  if(!identical(custClimVar,climVarSel)) # get custom climate var name from file
  {
     message(paste("Climate variable in reconstruction file is",custClimVar,sep=" "))
     climVarSel <- NULL
  }else
  {
     message(paste("Climate variable in reconstruction file is",climVarSel,sep=" "))
  }

  if (psStr == "CUSTOM")
     psStr = "0"

  # sort by CalYrBP (always column 7 in reconInput)
  reconInput <- reconInput[order(reconInput[,7]),]
  if (length(selClimVar) > 1)
     myList <- list(selClimVar[2], strMsg, reconInput, linkReconInput, as.numeric(psStr),custClimVar)
```

```
      else
         myList <- list(selClimVar, strMsg, reconInput, linkReconInput, as.numeric(psStr),custClimVar)
    }
  }else
      myList <- list(NULL, NULL, NULL, NULL, NULL, NULL)
  return (myList)
}setCalYrRange <- function()
{
   # ask user for oldest & youngest calYrBP needed for recon

   oldCalYr = -9999
   youngCalYr = -9999

   #message("For a linear interpolation at a 50 year interval for 0 to 2000 bp, enter -50 for youngest, 2099 for oldest cal year BP")
   #message("For a linear interpolation at a 200 year interval for 0 to 3000 bp, enter -200 for youngest, 3099 for oldest cal year BP")

   while (is.na(oldCalYr) | oldCalYr < 0 | oldCalYr > 15000)
   {
      oldCalYr=as.numeric(readline("What is oldest calYrBP range needed from the interpolation for the analysis? (max age is 15000, min
             age is 0) "))
      # oldCalYr2=oldCalYr %/% 100 * 100 + 99
      message(paste("Oldest calendar year BP is set to",oldCalYr,"and note that oldest calendar year cannot be the same value as
            youngest calendar year BP!"))

   }

   while (is.na(youngCalYr) | youngCalYr < 0 | youngCalYr > 15000 | youngCalYr >= oldCalYr)
   {
      youngCalYr=as.numeric(readline("What is youngest calYrBP range needed from the interpolation for the analysis? (min age is 0, max
             age is 15000) "))
      # youngCalYr2=youngCalYr %/% 100 * 100
      message(paste("Youngest calendar year BP is set to",youngCalYr,"and note that youngest calendar year cannot be the same value as
            oldest calendar year BP!"))
   }

   calYrRange <- list(oldCalYr,youngCalYr)
   return(calYrRange)
}setLinearInterpInterval <- function()
{
   # interpolate the cal yr BP dates in this subset
   yearInt = 0
   while (is.na(yearInt) | yearInt < 10 | yearInt > 1000)
   {
      yearInt=yearInt %/% 10 * 10 # must be 10 to 1000 at 10 year intervals
      yearInt=as.numeric(readline("Enter the interval for linear interpolation? (valid range is 10-1000 years) "))
   }
   return (yearInt)
}setNumberOfAnalogs <- function(reconInput,analogColUsed1,analogColUsed2,selClimVar)
{
  # NOTE: this is not the same as numAnalogs, numAnalogs is the total number
  # of analogs in the reconstruction file, this will selected one of those
  # analogs, with the option to average analogs (e.g. avg of top 5 analogs)

  # if the user would like an average of the top analogs then
  # user should enter 1 at the first prompt and then 5 at the second prompt

  # 1st analog is column 48, 2nd analog is col 49, etc.

  matchCol = 0
  analogColLast = 0

  # only 1 analog, simply get the reconstructed climate data in that col
  if (analogColUsed1 == analogColUsed2)
  {
     # figure out which is the top analog column with climate data
     # we know the top analog will be climVar1, second analog will be climVar2
     # therefore, we can set up a search string in order to get the index
     # of the column header with climVarN

     strMatch = paste(selClimVar,analogColUsed1)
     strMatch = gsub(" ","", strMatch ,fixed=TRUE)
     matchCol = which(colnames(reconInput) == strMatch)

     analogColIndex1 = matchCol

     # select subset of data (eg - siteID, lat, lon, calYrBP, reconstructed climate variable)
```

```
    reconSubsetFinal <- reconInput[c(1:3,7,analogColIndex1)]
    # these col indexes (1 to 3 and 7) do not change as mattools (mat.dissim) output is
    # always formatted the same way no matter which climate variable was selected

  } else # multiple analogs are used, will need to average the correct columns
  {
    strMatch = paste(selClimVar,analogColUsed1)
    strMatch = gsub(" ","", strMatch ,fixed=TRUE)
    matchCol = which(colnames(reconInput) == strMatch)

    analogColIndex1 = matchCol
    analogColIndex2 = matchCol + (analogColUsed2 - analogColUsed1)

    #select subset of data (eg - siteID, lat, lon, calYrBP, reconstructed climate variable)
    if (analogColIndex2 - analogColIndex1 == 1)
        reconSubset2 <- reconInput[c(analogColIndex1,analogColIndex2)]
    else
        reconSubset2 <- reconInput[c(analogColIndex1:analogColIndex2)]

    reconSubset3 <- reconInput[c(1:3,7)] # separate subset for final output

    analogsNumTot = analogColUsed2 - analogColUsed1 + 1

    # if 2 columns sum the 2 columns and divide the values by 2 to get the new avg analog col
    reconSubsetFinal <- cbind(reconSubset3,(rowSums(reconSubset2) / analogsNumTot))
    analogColLast = ncol(reconSubsetFinal)

    # fix col header for avg analog col
    anAvgColStr <- paste(selClimVar,analogColUsed1,"to",analogColUsed2)
    anAvgColStr2 = gsub(" ","", anAvgColStr ,fixed=TRUE)
    colnames(reconSubsetFinal)[analogColLast] <- anAvgColStr2

  }
  return(reconSubsetFinal)
}setWorkingDirectory <- function()
{
   workingDir=readline("Enter the path to working directory (e.g. C:/pollen) ")
   if (!is.null(workingDir))
   {
      tryCatch({
      setwd(workingDir)
      }, error = function(ex) {
      })
      # if invalid working dir entered here, working dir will not change
      message(paste("Working directory set to ",getwd()))
   }
}showPollenTaxaList <- function()
{
  # has ID in first col and taxa name in 2nd col
  message("Reading from file customPollenSumTaxaList2.csv....")
  tryCatch({customPollenSumTaxaList2 <- read.csv("customPollenSumTaxaList2.csv")
  customTaxaList <- customPollenSumTaxaList2[,1]

  message("---Custom Pollen Taxa List---")
  tc=1
  while (tc <= 135)
  {
    if (tc %in% customTaxaList)
    {
      message(colnames(modernPollDB)[tc+13]) # 13 col offset between taxa list and pollen db
      customPollenSumCount = customPollenSumCount + 1
    }
    tc = tc+1
  }
  message("------------------------------")
  message(paste("Total number of taxa: ", customPollenSumCount))
  message("------------------------------")
  if (customPollenSumCount < 10)
  {
     message("Warning: You cannot build a custom pollen sum database with less than 10 taxa!")
     message("Please edit the custom pollen sum taxa file to include at least 10 taxa!")
  }
  }, error = function(e) {message("Cannot read from custom pollen taxa list, please check to make sure file exists!")}, warning =
        function(w){})
}sortRecon <- function(reconSubset,colID)
{
```

```
   reconSorted <- reconSubset[order(reconSubset[,colID]),]
}
spatialFiltering <- function(reconAnomFile)
{
    #-------------------------------------------------------------------------------------------
    # LAST MODIFIED: June 27, 2013
    #-------------------------------------------------------------------------------------------
    # LIST OF FEATURES TO ADD:
    # Could output some basic stats such as median, mode, avg distance, any other statistics?
    #
    # Num. Neighbours at last lag - use summary(recon_dnn)
    # Lag Number (not needed)
    # Sum of LocalI (implemented)
    # Sum of Num. Neighbours (?)
    # Constant of Proportionality (?)
    # Global Moran's I (?)
    #-------------------------------------------------------------------------------------------

    require(sp)
    require(spdep)
    require(proj4)
    require(rgdal)

    lagSel = 0
    lagLast = 0
    lagFirst = 0
    lagInt = 0
    selYr = 1
    lastCol = 10
    reconModAnom = 0
    addCounts = 0
    agCounts = 0
    showInfo = 0
    totSites = 0
    mapAnom = 0
    cyCount = 0
    sigValue = 0
    alag = 0
    keepZeroes = 0
    posPosOnly = 0

    n1CountAg = 0
    z1CountAg = 0
    p1CountAg = 0
    n2CountAg = 0
    z2CountAg = 0
    p2CountAg = 0
    nnCountAg = 0
    ppCountAg = 0
    pnCountAg = 0
    zzCountAg = 0
    pSigCountAg = 0
    pNotSigCountAg = 0
    pZeroCountAg = 0
    nSitesAg = 0
    calYrStr = ""

    oldCalYr = -999
    youngCalYr = -999
    youngCalYrMin = 0
    oldCalYrMax = 0
    nSitesLag = 0

    reconFiltered = NULL
    reconUnfiltered = NULL
    finalMatrixAllCalYrBP = NULL
    reconFile = NULL
    calYear = NULL
    spFilterList = NULL

    # read in the reconstruction anomaly file
    # reconFile = "mtwa_2k_new/SpatialJoinReconModWithAnCalc.csv"
    # reconAnomFile <- read.csv(reconFile)

    # make sure file is sorted by siteID
    reconAnomFile<- reconAnomFile[order(reconAnomFile[,1]),]
```

```
# change some col names here
colnames(reconAnomFile)[5] = "RECON"
colnames(reconAnomFile)[8] = "MOD"
colnames(reconAnomFile)[10] = "ANOM"

# need to loop through all cal yr BP intervals in file
calYear <- unique(reconAnomFile[,4])
youngCalYrMin = calYear[1]
oldCalYrMax = calYear[length(calYear)]

message(paste("Range of reconstruction is",youngCalYrMin,"to",oldCalYrMax,"bp at",oldCalYrMax /(length(calYear)-1),"year intervals
      ",sep=" "))

lagInt = 50
# ask user for lag interval in km
while (is.na(lagInt) | (lagInt != 10 & lagInt != 25 & lagInt != 50))
{
   message("Enter the lag interval in km (must be either 10, 25 or 50): ")
   lagInt <- as.numeric(readline())
}
message("Lag interval set to 50 km (default)")

# ask user for starting lag (must be in the range of 50k to 500k)
while (is.na(lagFirst) | lagFirst < 50 | lagFirst > 500)
{
   message("Enter the first lag distance (in km) of points for which to calculate Moran's I values (must be between 50 and 500 km,
         rounded down to nearest 50k interval): ")
   lagFirst <- as.numeric(readline())
}
lagFirst = lagFirst %/% 50 * 50 # rounded to nearest 50k interval

# ask user for ending lag (must be in the range of 50k to 500k)
while (is.na(lagLast) | lagLast < 50 | lagLast > 500)
{
   message("Enter the last lag distance (in km) of points for which to calculate Moran's I values (must be between 50 and 500 km,
         rounded down to nearest 50k interval): ")
   lagLast <- as.numeric(readline())
}
lagLast = lagLast %/% 50 * 50 # rounded to nearest 50k interval

#reconModAnom = 3
while (is.na(reconModAnom) | reconModAnom < 1 | reconModAnom > 3)
{
   message("Do you want to filter by reconstructed values <enter 1>, modern values <enter 2> or anomalies (reconstructed minus
         modern) <enter 3> ? ")
   reconModAnom <- as.numeric(readline())
}

#keepZeroes = 2
while (is.na(keepZeroes) | keepZeroes < 1 | keepZeroes > 2)
{
   message("Do you want to keep MI zero values <enter 1> or remove them <enter 2> after filtering at the last lag ? ")
   keepZeroes <- as.numeric(readline())
}

#posPosOnly = 1
while (is.na(posPosOnly) | posPosOnly < 1 | posPosOnly > 2)
{
   message("Do you want to retain only MI pos and zMi pos values <enter 1> or both pos/pos and pos/neg <enter 2> ? ")
   posPosOnly <- as.numeric(readline())
}

while (is.na(sigValue) | sigValue > 0.4 | sigValue < 0.01)
{
   message("What is the p-value cutoff? (p-value cutoff must be between 0.01 [99% sig level] and 0.4 [60% sig level], is only used
         for counts not filtering)")
   sigValue <- as.numeric(readline())
}

while (is.na(mapAnom) | mapAnom < 1 | mapAnom > 2)
{
   message("Do you want to map the anomalies (negative/zero/positive)? <1 for YES, 2 for NO> ")
   mapAnom <- as.numeric(readline())
}
```

```
while (is.na(agCounts) | agCounts < 1 | agCounts > 2)
{
   message("Do you want to aggregate counts for a range of cal year intervals for a specified lag? <1 for YES, 2 for NO> ")
   agCounts <- as.numeric(readline())
}

while (is.na(showInfo) | showInfo < 1 | showInfo > 2)
{
   message("Do you want to display nearest neighbor summary information at each lag for each cal yr interval? <1 for YES, 2 for NO>
         ")
   showInfo <- as.numeric(readline())
}

if (agCounts == 1)
{
  # the user can enter a specific lag distance for the aggregation
  while (is.na(alag)| alag > lagLast | alag < 50)
     alag=as.numeric(readline("What is lag distance you wish to aggregate? (e.g. 250, rounded down to nearest 50k interval) "))
  alag = alag %/% 50 * 50 # rounded down to the nearest 50k interval
}

#message("All sites with zero values for moran's I, z-values and p-values are retained until the last lag.")
#message("At each lag the significance (p-values) are calculated for the Moran's I and sites that are significant are removed.")
#message("The retained sites are then input into the next lag distance and the moran's I, z-values and p-values are recalculated.")

# information about how the spatial filtering is done
message("")
message("There are 4 possibilities for combinations of Moran's I and Moran's I z-values: ")
message("")
message("positive/positive (criteria 1)")
message("negative/negative (criteria 2)")
message("mixed signs either pos/neg or neg/pos (criteria 3)")
message("all zeroes (criteria 4)")
message("")
message("All rows (sites) that do not meet either criteria 1 or criteria 4 will be removed at each")
message("lag until the last lag where only sites meeting criteria 1 will be retained.")
message("")

# filtering will be done on each cal yr interval in the recon
# then recombine retained sites at each interval and return this matrix at the end

while (selYr <= length(calYear))
{

  # get the subset for subset by cal year interval (selYr)
  reconAnomSubset <- subset(reconAnomFile, reconAnomFile$CalMean == calYear[selYr])

  totSites = nrow(reconAnomSubset)

  message("")
  message(paste("Total sites for",calYear[selYr],"bp:",totSites,sep=" "))

  lagSel = lagFirst # reset this

  while (lagSel <= lagLast)
  {
    if (lagSel == lagFirst)
    {
       sp1 <- SpatialPoints(reconAnomSubset[c(2,3)], proj4string = CRS("+proj=longlat +datum=WGS84"))
       message(paste("Total sites for",calYear[selYr],"bp input into lag distance of ",lagSel,"km:",nrow(reconAnomSubset),sep=" "))
    }else
    {
       sp1 <- SpatialPoints(reconFiltered[c(2,3)], proj4string = CRS("+proj=longlat +datum=WGS84"))
       message(paste("Total sites for",calYear[selYr],"bp input into lag distance of",lagSel,"km:",nrow(reconFiltered),sep=" "))
    }

    # lat/lon values need to be projected in order to calculate moran's I
    sp1Trans <- spTransform(sp1,CRS("+proj=eqdc +lat_0=40 +lon_0=-96 +lat_1=20 +lat_2=60 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +
          units=m +no_defs +towgs84=0,0,0"))

    # lagSel*1000 as lagSel needs to be in metres
    recon_dnn = dnearneigh(sp1Trans,0,lagSel*1000,row.names=NULL,longlat=NULL)

    if (showInfo == 1) # display nearest neighbor info
    {
       message("")
```

```
   summary(recon_dnn)
   message("")
}

# create the spatial weights file here (use row standardized weights, for binary weights, use style="B")
recon_wt = nb2listw(recon_dnn,style="W",zero.policy=TRUE)

# The localmoran function outputs 5 columns of data:
# col 1 is moran's I value
# col 2 is moran's I expected value
# col 3 is moran's I variance
# col 4 is moran's I z-value
# col 5 is moran's I p-value

if (lagSel == lagFirst)
{
   if (reconModAnom == 1)
      locMorIdata <- localmoran(reconAnomSubset$RECON, recon_wt, zero.policy=TRUE, na.action=na.pass, alternative = "two.sided",
             p.adjust.method="none", spChk=NULL)
   else if (reconModAnom == 2)
      locMorIdata <- localmoran(reconAnomSubset$MOD, recon_wt, zero.policy=TRUE, na.action=na.pass, alternative = "two.sided", p
            .adjust.method="none", spChk=NULL)
   else
      locMorIdata <- localmoran(reconAnomSubset$ANOM, recon_wt, zero.policy=TRUE, na.action=na.pass, alternative = "two.sided",
            p.adjust.method="none", spChk=NULL)

}else
{
   if (reconModAnom == 1)
      locMorIdata <- localmoran(reconFiltered$RECON, recon_wt, zero.policy=TRUE, na.action=na.pass, alternative = "two.sided", p
            .adjust.method="none", spChk=NULL)
   else if (reconModAnom == 2)
      locMorIdata <- localmoran(reconFiltered$MOD, recon_wt, zero.policy=TRUE, na.action=na.pass, alternative = "two.sided", p.
            adjust.method="none", spChk=NULL)
   else
      locMorIdata <- localmoran(reconFiltered$ANOM, recon_wt, zero.policy=TRUE, na.action=na.pass, alternative = "two.sided", p.
            adjust.method="none", spChk=NULL)

}

locMorIdata[is.na(locMorIdata)] <- 0 # replace NAs with zeroes

# get the sum of moran's I
allColSums = colSums(locMorIdata)
sumMI = allColSums[1]

message("")
message(paste("Sum of Moran's I is",round(sumMI,digits=2),sep=" "))
message("")

if (lagSel == lagFirst)
   reconUnfiltered <- cbind(reconAnomSubset,locMorIdata[,1],locMorIdata[,4],locMorIdata[,5])
else # recon unfiltered at next lag is assigned recon filtered from previous lag
   reconUnfiltered <- cbind(reconFiltered[,1:10],locMorIdata[,1],locMorIdata[,4],locMorIdata[,5])

# fix column headers
colnames(reconUnfiltered)[11] <- "MI"
colnames(reconUnfiltered)[12] <- "zMI"
colnames(reconUnfiltered)[13] <- "pMI"

fileStr <- paste("recon_unfiltered_",lagSel,"km_",calYear[selYr],"bp.csv",sep="")
# output unfiltered recon at selected lag distance
tryCatch({write.csv(reconUnfiltered,fileStr,row.names=FALSE)}, error = function(e) {message(paste("Cannot write",fileStr,"to
      disk, file may be open!",sep=" "))}, warning = function(w){})

reconSubsetRange <- reconUnfiltered

# always output counts/percentages on unfiltered data
# last lag (z-value) is in selCol + 1

firstCol = 11
firstRow = 2
nSites = totSites
lastRow = nrow(reconUnfiltered)
selRow = firstRow
selCol = firstCol
```

```
ppCount = 0 # pos/pos values
nnCount = 0 # neg/neg values
zzCount = 0 # zero/zero values
pnCount = 0 # mixed values either pos/neg or neg/pos
p1Count = 0 # moran's I pos values
n1Count = 0 # moran's I neg values
z1Count = 0 # moran's I zero values
p2Count = 0 # moran's I pos z-values
n2Count = 0 # moran's I neg z-values
z2Count = 0 # moran's I zero z-values
pSigCount = 0 # p-values that are sig
pNotSigCount = 0 # p-values that are not sig
pZeroCount = 0 # p-values that are zero

# build a new matrix for table data
dataTable = mat.or.vec(lagLast / 50 * 2, 15)
dtColInd = 1 # starting col for output data table
dtRowInd = 1 # starting row for output data table

# dtRowInd = (lagSel / 50) + (lagSel / 50 - 1) # starting row for output data table

# create the column headers
colnames(dataTable) = c("lag (km)","MI (neg)","MI (zero)","MI (pos)","zMI (neg)","zMI (zero)","zMI (pos)","neg/neg","pos/pos","
      mixed","zero","pMI (sig)","pMI (not sig)","pMI (zero)","total sites")

# now calculate the totals of pos, zero & neg moran's I and z-values

ppCount = 0 # pos/pos values
nnCount = 0 # neg/neg values
zzCount = 0 # zero/zero values
pnCount = 0 # mixed values either pos/neg or neg/pos
p1Count = 0 # moran's I pos values
n1Count = 0 # moran's I neg values
z1Count = 0 # moran's I zero values
p2Count = 0 # moran's I pos z-values
n2Count = 0 # moran's I neg z-values
z2Count = 0 # moran's I zero z-values
pSigCount = 0 # p-values that are sig
pNotSigCount = 0 # p-values that are not sig
pZeroCount = 0 # p-values that are zero

# get counts for pos MI, pos zMI, neg MI, neg zMI, zero MI, zero zMI
while (selRow <= lastRow)
{
   if (reconSubsetRange[selRow,selCol] > 0)
      p1Count = p1Count + 1
   if (reconSubsetRange[selRow,selCol+1] > 0)
      p2Count = p2Count + 1

   if (reconSubsetRange[selRow,selCol] < 0)
      n1Count = n1Count + 1
   if (reconSubsetRange[selRow,selCol+1] < 0)
      n2Count = n2Count + 1

   if (reconSubsetRange[selRow,selCol] == 0)
      z1Count = z1Count + 1
   if (reconSubsetRange[selRow,selCol+1] == 0)
      z2Count = z2Count + 1

   if (reconSubsetRange[selRow,selCol+2] < sigValue & reconSubsetRange[selRow,selCol+2] > 0)
      pSigCount = pSigCount + 1
   if (reconSubsetRange[selRow,selCol+2] >= sigValue)
      pNotSigCount = pNotSigCount + 1
   if (reconSubsetRange[selRow,selCol+2] == 0)
      pZeroCount = pZeroCount + 1

   selRow = selRow + 1
}
selRow = 2 # reset selRow here


# now get counts for neg/neg, pos/pos, zero/zero & mixed
# for moran's I and z-values

while (selRow <= lastRow)
```

```
{

    if (reconSubsetRange[selRow,selCol] < 0 & reconSubsetRange[selRow,selCol+1] < 0)
        nnCount = nnCount + 1
    else if (reconSubsetRange[selRow,selCol] > 0 & reconSubsetRange[selRow,selCol+1] > 0)
        ppCount = ppCount + 1
    else if (reconSubsetRange[selRow,selCol] == 0 & reconSubsetRange[selRow,selCol+1] == 0)
        zzCount = zzCount + 1
    else # mixed values
        pnCount = pnCount + 1

    selRow = selRow + 1
}

# now write values to dataTable (selected lag in km, 10 count values, sites retained after filtering
# at previous lag - these sites are input into current lag)
# then next row (%, 10 percentage values, total sites at selected cal yr interval)

dtRowInd = 1 # go to next row in output table
dataTable[dtRowInd,dtColInd] = lagSel
dataTable[dtRowInd,dtColInd + 1] = n1Count
dataTable[dtRowInd,dtColInd + 2] = z1Count
dataTable[dtRowInd,dtColInd + 3] = p1Count
dataTable[dtRowInd,dtColInd + 4] = n2Count
dataTable[dtRowInd,dtColInd + 5] = z2Count
dataTable[dtRowInd,dtColInd + 6] = p2Count
dataTable[dtRowInd,dtColInd + 7] = nnCount
dataTable[dtRowInd,dtColInd + 8] = ppCount
dataTable[dtRowInd,dtColInd + 9] = pnCount
dataTable[dtRowInd,dtColInd + 10] = zzCount
dataTable[dtRowInd,dtColInd + 11] = pSigCount
dataTable[dtRowInd,dtColInd + 12] = pNotSigCount
dataTable[dtRowInd,dtColInd + 13] = pZeroCount
dataTable[dtRowInd,dtColInd + 14] = lastRow

dtRowInd = 2 # go to next row in output table
dtColInd = 1 # reset column in output table

dataTable[dtRowInd,dtColInd] = " % "
dataTable[dtRowInd,dtColInd + 1] = round(n1Count / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 2] = round(z1Count / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 3] = round(p1Count / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 4] = round(n2Count / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 5] = round(z2Count / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 6] = round(p2Count / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 7] = round(nnCount / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 8] = round(ppCount / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 9] = round(pnCount / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 10] = round(zzCount / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 11] = round(pSigCount / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 12] = round(pNotSigCount / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 13] = round(pZeroCount / lastRow, digits = 2)
dataTable[dtRowInd,dtColInd + 14] = nSites

# remove any blank rows in the output table
dataTable <- subset(dataTable,dataTable[,14] != 0)

# here is where we sum the counts for the aggregation counts table
if (agCounts == 1 & lagSel == alag)
{
  while (is.na(addCounts) | addCounts < 1 | addCounts > 2)
  {
      message(paste("Do you wish to add counts for",calYear[selYr],"bp to the aggregation? <1 for YES, 2 for NO> ",sep=" "))
      addCounts <- as.numeric(readline())
  }
  if (addCounts == 1)
  {
     cyCount = cyCount + 1
     calYrStr = paste(calYrStr,"_",calYear[selYr],sep="")
     dtRowInd = 1 # reset row index in output table
     dtColInd = 1 # reset column in output table
     n1CountAg = n1CountAg + as.numeric(dataTable[dtRowInd,dtColInd + 1])
     z1CountAg = z1CountAg + as.numeric(dataTable[dtRowInd,dtColInd + 2])
     p1CountAg = p1CountAg + as.numeric(dataTable[dtRowInd,dtColInd + 3])
     n2CountAg = n2CountAg + as.numeric(dataTable[dtRowInd,dtColInd + 4])
     z2CountAg = z2CountAg + as.numeric(dataTable[dtRowInd,dtColInd + 5])
```

```
    p2CountAg = p2CountAg + as.numeric(dataTable[dtRowInd,dtColInd + 6])
    nnCountAg = nnCountAg + as.numeric(dataTable[dtRowInd,dtColInd + 7])
    ppCountAg = ppCountAg + as.numeric(dataTable[dtRowInd,dtColInd + 8])
    pnCountAg = pnCountAg + as.numeric(dataTable[dtRowInd,dtColInd + 9])
    zzCountAg = zzCountAg + as.numeric(dataTable[dtRowInd,dtColInd + 10])
    pSigCountAg = pSigCountAg + as.numeric(dataTable[dtRowInd,dtColInd + 11])
    pNotSigCountAg = pNotSigCountAg + as.numeric(dataTable[dtRowInd,dtColInd + 12])
    pZeroCountAg = pZeroCountAg + as.numeric(dataTable[dtRowInd,dtColInd + 13])
    nSitesAg = nSitesAg + as.numeric(dataTable[dtRowInd + 1,dtColInd + 14])
  }
  addCounts = 0
}

# write out the table of counts/percentages for selected lag and cal year interval
outFileName <- paste("filtered_recon_stats_",lagSel,"km_",calYear[selYr],"bp.csv",sep="")
message(paste("Writing table of statistics for filtered reconstruction to file",outFileName,"....",sep=" "))
tryCatch({write.csv(dataTable,outFileName,row.names=FALSE)}, error = function(e){message(paste(outFileName,"cannot be written to
       disk, please close the file!",sep=""))}, warning = function(w){})


# do spatial filtering here (keep pos/pos and zeroes until the last lag where only pos/pos sites are retained)
# several options here

if (lagSel < lagLast)
{
 if (posPosOnly == 1) # pos/pos
    reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] >= 0 & reconUnfiltered[,lastCol+2] >= 0)
 else if (posPosOnly == 2) #pos/pos & pos/neg
    reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] >= 0)
}else
{
 if (posPosOnly == 1 & keepZeroes == 1) # pos/pos & retain zeroes
    reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] >= 0 & reconUnfiltered[,lastCol+2] >= 0)
 else if (posPosOnly == 2 & keepZeroes == 1) #pos/pos & pos/neg & retain zeroes
    reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] >= 0)
 else if (posPosOnly == 1 & keepZeroes == 2) #pos/pos & do not retain zeroes
    reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] > 0 & reconUnfiltered[,lastCol+2] > 0)
 else if (posPosOnly == 2 & keepZeroes == 2) #pos/pos & pos/neg & do not retain zeroes
    reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] > 0 & reconUnfiltered[,lastCol+2] != 0)
}

# if (lagSel < lagLast) # keep sites where p-value is not stat sig, except zeroes which are retained for next lag
# reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+2] > 0 & reconUnfiltered[,lastCol+1] > 0 | reconUnfiltered[,
       lastCol+3] == 0)
# else # now get rid of zeroes at the last lag since the are beyond this search radius
# reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+2] > 0 & reconUnfiltered[,lastCol+1] > 0)

#if (lagSel < lagLast) # keep sites where p-value is not stat sig, except zeroes which are retained for next lag
# reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+3] >= sigValue | reconUnfiltered[,lastCol+3] == 0)
#else # now get rid of zeroes at the last lag since the are beyond this search radius
# reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+3] >= sigValue)

message(paste("Total sites for",calYear[selYr],"bp filtered out at lag distance of",lagSel,"km:",nrow(reconUnfiltered)- nrow(
       reconFiltered),sep=" "))
message(paste("Total sites for",calYear[selYr],"bp retained at lag distance of",lagSel,"km:",nrow(reconFiltered),sep=" "))

fileStr2 <- paste("filteredRecon_",lagSel,"km_",calYear[selYr],"bp.csv",sep="")
tryCatch({write.csv(reconFiltered,fileStr2,row.names=FALSE)}, error = function(e) {message(paste("Cannot write",fileStr2,"to
       disk, file may be open!",sep=" "))}, warning = function(w){})

# output anomaly info and (optionally) a map of anomalies (pos/zero/neg)

negAnomSubset <- subset(reconFiltered, reconFiltered[,10] < -0.000001)
zeroAnomSubset <- subset(reconFiltered, reconFiltered[,10] >= -0.000001 & reconFiltered[,10] <= 0.000001 )
posAnomSubset <- subset(reconFiltered, reconFiltered[,10] > 0.000001)

if (mapAnom == 1)
{
   outfile <- paste("anomalyMap_",calYear[selYr],"bp_",lagSel,"km.pdf",sep="")
   pdf(outfile)

   # North America
   lonLimits <- c(-170,-40)
   latLimits <- c(15,90)

   map(database = "world",xlim = lonLimits,ylim = latLimits)
```

```
      map.axes()

      points(negAnomSubset$Lon,negAnomSubset$Lat,pch=21,col="blue",cex=1)
      points(zeroAnomSubset$Lon,zeroAnomSubset$Lat,pch=21,col="darkgrey",cex=1)
      points(posAnomSubset$Lon,posAnomSubset$Lat,pch=21,col="red",cex=1)
      legend("bottomleft",legend=c("Positive","Zero","Negative"),pch =c(21,2),col=c("red","darkgrey","blue"))
      dev.off()
    }

    message(paste("At",calYear[selYr],"bp for a lag distance of",lagSel,"km:",sep=" "))
    message(paste("There are ",length(unique(negAnomSubset[,1]))," sites with negative anomalies",sep=" "))
    message(paste("There are ",length(unique(zeroAnomSubset[,1]))," sites with zero anomalies",sep=" "))
    message(paste("There are ",length(unique(posAnomSubset[,1]))," sites with positive anomalies",sep=" "))

    lagSel = lagSel + lagInt # go to the next lag
    lastCol = 10 # reset this
  }

  # now combine final filtered file from each cal year interval into one file
  if (selYr == 1)
    finalMatrixAllCalYrBP = reconFiltered
  else
    finalMatrixAllCalYrBP <- rbind(finalMatrixAllCalYrBP,reconFiltered)

  selYr = selYr + 1

}

if (agCounts == 1) # if aggregation option is chosen
{
  # build a new matrix for table data (aggregation)
  outTable = mat.or.vec(2, 15)
  dtColInd = 1 # starting col for output data table
  dtRowInd = 1 # starting row for output data table

  # create the column headers
  colnames(outTable) = c("lag (km)","MI (neg)","MI (zero)","MI (pos)","zMI (neg)","zMI (zero)","zMI (pos)","neg/neg","pos/pos","
        mixed","zero","pMI (sig)","pMI (not sig)","pMI (zero)","total sites")

  # now write values to outTable (lag km, 10 values, nSites)
  # repeat 2 rows, 15 cols for each lag dist

  outTable[dtRowInd,dtColInd] = alag
  outTable[dtRowInd,dtColInd + 1] = n1CountAg
  outTable[dtRowInd,dtColInd + 2] = z1CountAg
  outTable[dtRowInd,dtColInd + 3] = p1CountAg
  outTable[dtRowInd,dtColInd + 4] = n2CountAg
  outTable[dtRowInd,dtColInd + 5] = z2CountAg
  outTable[dtRowInd,dtColInd + 6] = p2CountAg
  outTable[dtRowInd,dtColInd + 7] = nnCountAg
  outTable[dtRowInd,dtColInd + 8] = ppCountAg
  outTable[dtRowInd,dtColInd + 9] = pnCountAg
  outTable[dtRowInd,dtColInd + 10] = zzCountAg
  outTable[dtRowInd,dtColInd + 11] = pSigCountAg
  outTable[dtRowInd,dtColInd + 12] = pNotSigCountAg
  outTable[dtRowInd,dtColInd + 13] = pZeroCountAg
  nSitesLag <- pSigCountAg + pNotSigCountAg + pZeroCountAg
  outTable[dtRowInd,dtColInd + 14] = nSitesLag

  dtRowInd = 2 # go to next row in output table
  dtColInd = 1 # reset column in output table

  outTable[dtRowInd,dtColInd] = " % "
  outTable[dtRowInd,dtColInd + 1] = round(n1CountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 2] = round(z1CountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 3] = round(p1CountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 4] = round(n2CountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 5] = round(z2CountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 6] = round(p2CountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 7] = round(nnCountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 8] = round(ppCountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 9] = round(pnCountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 10] = round(zzCountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 11] = round(pSigCountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 12] = round(pNotSigCountAg / nSitesLag, digits = 2)
  outTable[dtRowInd,dtColInd + 13] = round(pZeroCountAg / nSitesLag, digits = 2)
```

```
    outTable[dtRowInd,dtColInd + 14] = nSitesAg
    #outTable[dtRowInd,dtColInd + 14] = paste(cyCount,"intervals",sep=" ")

    outFileName = paste("filtered_recon_stats_aggregation_",alag,"km_",calYrStr,"bp.csv",sep="")
    message(paste("Writing table of aggregated statistics for filtered reconstruction to file",outFileName,sep=" "))
    tryCatch({write.csv(outTable,outFileName,row.names=FALSE)}, error = function(e){message(paste(outFileName,"cannot be written to
          disk, please close the file!",sep=""))}, warning = function(w){})
  }

  # now write out the final data file
  tryCatch({write.csv(finalMatrixAllCalYrBP,"filteredReconstructionAllCalYrBP.csv",row.names=FALSE)}, error = function(e){message("
        filteredReconstruction.csv cannot be written to disk, please close the file!")}, warning = function(w){})
  message("")
  message("Processing completed!!")

  # fix RECON and ANOM col headers
  colnames(finalMatrixAllCalYrBP)[5] = "InterpVal"
  colnames(finalMatrixAllCalYrBP)[8] = "ModClimVal"
  colnames(finalMatrixAllCalYrBP)[10] = "reconSPjoinAnCalc"

  spFilterList <- list(finalMatrixAllCalYrBP, oldCalYrMax, youngCalYrMin)

  # RETURN matrix of all cal yr bp here
  return(spFilterList)

  ### END OF FUNCTION ###
}
}spFilterTest <- function()
{

   require(sp)
   require(spdep)
   require(proj4)
   require(rgdal)

   lagSel = 50
   lastCol = 6

   reconFiltered = NULL
   reconUnfiltered = NULL

   calYr = "MWP-LIA"
   #reconFile = paste("sp_filtering/reconTestXY_",calYr,"_test.csv",sep="")
   reconFile = "avgAnTSdiff.csv"
   reconAnomSubset <- read.csv(reconFile)

   # need to ensure that function works with file created from menu item #8
   # the file from menu item #8 has 10 columns (last column is anomaly)

   # first lag is 0k, last lag is 500k, interval is 50k (these are defaults)

   # ask user for cal yr bp range (e.g. 0 to 2000, 1300 to 1700 or 1500 to 1500)
   # create subset from each interval within range specified by user
   # would the user like to aggregate counts for specified range? (Y/N)

   # also output info about each filtering step (lag & cal year bp)

   # output some basic stats such as median, mode, avg distance (rookcase)

   # how to implement the p-value in the filtering?


   while (lagSel <= 250)
   {
     if (lagSel == 50)
        sp1 <- SpatialPoints(reconAnomSubset[c(2,3)], proj4string = CRS("+proj=longlat +datum=WGS84"))
     else
        sp1 <- SpatialPoints(reconFiltered[c(2,3)], proj4string = CRS("+proj=longlat +datum=WGS84"))

     sp1Trans <- spTransform(sp1,CRS("+proj=eqdc +lat_0=40 +lon_0=-96 +lat_1=20 +lat_2=60 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +
           units=m +no_defs +towgs84=0,0,0"))

     recon_dnn = dnearneigh(sp1Trans,0,lagSel*1000,row.names=NULL,longlat=NULL)

     # summary(recon_dnn)
```

```
recon_wt = nb2listw(recon_dnn,style="B",zero.policy=TRUE)

# The localmoran function outputs 5 columns of data:
   # col 1 is moran's I value
   # col 2 is moran's I expected value
   # col 3 is moran's I variance
   # col 4 is moran's I z-value
   # col 5 is moran's I p-value

if (lagSel == 50)
   locMorIdata <- localmoran(reconAnomSubset$AN, recon_wt, zero.policy=TRUE, na.action=na.pass, alternative = "two.sided", p.
         adjust.method="none", spChk=NULL)
else
   locMorIdata <- localmoran(reconFiltered$AN, recon_wt, zero.policy=TRUE, na.action=na.pass, alternative = "two.sided", p.adjust.
         method="none", spChk=NULL)

locMorIdata[is.na(locMorIdata)] <- 0 # replace NAs with zeroes

if (lagSel == 50)
   reconUnfiltered <- cbind(reconAnomSubset,locMorIdata[,1],locMorIdata[,4],locMorIdata[,5])
else
   reconUnfiltered <- cbind(reconFiltered[,1:6],locMorIdata[,1],locMorIdata[,4],locMorIdata[,5])

# fix column headers
colnames(reconUnfiltered)[7] <- "MI"
colnames(reconUnfiltered)[8] <- "zMI"
colnames(reconUnfiltered)[9] <- "pMI"

write.csv(reconUnfiltered,paste("recon_unfiltered_",lagSel,"km_",calYr,".csv",sep=""),row.names=FALSE)

reconSubsetRange <- reconUnfiltered

# last lag (z-value) is in col ncol
firstCol = 7
firstRow = 2
lastRow = nrow(reconSubsetRange)
nSites = 583
selRow = firstRow
selCol = firstCol

ppCount = 0 # pos/pos values
nnCount = 0 # neg/neg values
zzCount = 0 # zero/zero values
pnCount = 0 # mixed values either pos/neg or neg/pos
p1Count = 0 # moran's I pos values
n1Count = 0 # moran's I neg values
z1Count = 0 # moran's I zero values
p2Count = 0 # moran's I pos z-values
n2Count = 0 # moran's I neg z-values
z2Count = 0 # moran's I zero z-values

# build a new matrix for table data based on # of lags
# 12 columns and at least (2 x (lastLag - firstLag)) rows
# dataTable = mat.or.vec((2 * (lastLag - firstLag)) - 1, 12)
dataTable = mat.or.vec(40, 12)

dtColInd = 1 # starting col for output data table
dtRowInd = 1 # starting row for output data table

# create the column headers
colnames(dataTable) = c("lag (km)","MI (neg)","MI (zero)","MI (pos)","MIz (neg)","MIz (zero)","MIz (pos)","neg/neg","pos/pos","
      mixed","zero",paste(calYr,sep=""))

# start with first lag distance
# spLagDistSel = spLagDistFirst

# now calculate the totals of pos, zero & neg moran's I and z-values

ppCount = 0 # pos/pos values
nnCount = 0 # neg/neg values
zzCount = 0 # zero/zero values
pnCount = 0 # mixed values either pos/neg or neg/pos
p1Count = 0 # moran's I pos values
n1Count = 0 # moran's I neg values
z1Count = 0 # moran's I zero values
p2Count = 0 # moran's I pos z-values
```

```
n2Count = 0 # moran's I neg z-values
z2Count = 0 # moran's I zero z-values

while (selRow <= lastRow)
{
     if (reconSubsetRange[selRow,selCol] > 0)
        p1Count = p1Count + 1
     if (reconSubsetRange[selRow,selCol+1] > 0)
        p2Count = p2Count + 1

     if (reconSubsetRange[selRow,selCol] < 0)
        n1Count = n1Count + 1
     if (reconSubsetRange[selRow,selCol+1] < 0)
        n2Count = n2Count + 1

     if (reconSubsetRange[selRow,selCol] == 0)
        z1Count = z1Count + 1
     if (reconSubsetRange[selRow,selCol+1] == 0)
        z2Count = z2Count + 1
     selRow = selRow + 1
}
selRow = 2


# now get counts for neg/neg, pos/pos, zero/zero & mixed
# for moran's I and z-values

while (selRow <= lastRow)
{

     if (reconSubsetRange[selRow,selCol] < 0 & reconSubsetRange[selRow,selCol+1] < 0)
          nnCount = nnCount + 1
     else if (reconSubsetRange[selRow,selCol] > 0 & reconSubsetRange[selRow,selCol+1] > 0)
          ppCount = ppCount + 1
     else if (reconSubsetRange[selRow,selCol] == 0 & reconSubsetRange[selRow,selCol+1] == 0)
          zzCount = zzCount + 1
     else # mixed values
          pnCount = pnCount + 1

     selRow = selRow + 1
}

# now write values to dataTable (lag km, 10 values, nSites)
# repeat 2 rows, 12 cols for each lag dist

dtRowInd = dtRowInd + 1 # go to next row in output table
dataTable[dtRowInd,dtColInd] = lagSel
dataTable[dtRowInd,dtColInd + 1] = n1Count
dataTable[dtRowInd,dtColInd + 2] = z1Count
dataTable[dtRowInd,dtColInd + 3] = p1Count
dataTable[dtRowInd,dtColInd + 4] = n2Count
dataTable[dtRowInd,dtColInd + 5] = z2Count
dataTable[dtRowInd,dtColInd + 6] = p2Count
dataTable[dtRowInd,dtColInd + 7] = nnCount
dataTable[dtRowInd,dtColInd + 8] = ppCount
dataTable[dtRowInd,dtColInd + 9] = pnCount
dataTable[dtRowInd,dtColInd + 10] = zzCount
dataTable[dtRowInd,dtColInd + 11] = nSites

dtRowInd = dtRowInd + 1 # go to next row in output table
dtColInd = 1 # reset column in output table

dataTable[dtRowInd,dtColInd] = " % "
dataTable[dtRowInd,dtColInd + 1] = round(n1Count / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 2] = round(z1Count / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 3] = round(p1Count / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 4] = round(n2Count / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 5] = round(z2Count / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 6] = round(p2Count / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 7] = round(nnCount / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 8] = round(ppCount / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 9] = round(pnCount / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 10] = round(zzCount / nSites, digits = 2)
dataTable[dtRowInd,dtColInd + 11] = "total sites"

# spLagDistSel = spLagDistSel + spLagDistInt # go to next lag
```

```
    # dtColInd = 1 # reset column in output table
    # selRow = 2 # reset row in data matrix

    # cleanup any blank rows here
    dataTable <- subset(dataTable,dataTable[,12] != 0)

    # at the end write matrix to csv file, e.g. filtered_recon_stats_1300-1700bp.csv
    outFileName <- paste("filtered_recon_stats_",calYr,"_",lagSel,"km.csv",sep="")
    # outFileName = paste("filtered_recon_stats_",youngCalYrMin,"-",oldCalYrMax,"bp_",spLagDistLast,"km.csv",sep="")
    message(paste("Writing table of statistics for filtered reconstruction to file",outFileName,"....",sep=" "))
    tryCatch({write.csv(dataTable,outFileName,row.names=FALSE)}, error = function(e){message(paste(outFileName,"cannot be written to
          disk, please close the file!",sep=""))}, warning = function(w){})


    # do filtering here
    if (lagSel < 250)
       reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] >= 0 & reconUnfiltered[,lastCol+2] >= 0)
       # reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] * reconUnfiltered[,lastCol+2] >= 0)
    else
       reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] > 0 & reconUnfiltered[,lastCol+2] > 0)
       # reconFiltered <- subset(reconUnfiltered, reconUnfiltered[,lastCol+1] * reconUnfiltered[,lastCol+2] > 0)


    write.csv(reconFiltered,paste("filteredRecon_",lagSel,"km_",calYr,".csv",sep=""),row.names=FALSE)

    lagSel = lagSel + 50
    lastCol = 6

  }
  message("Processing completed!!")

}subsetReconByCalYrRange <- function(reconSubset,oldCalYr,youngCalYr)
{
   countSites = 0
   reconSubsetOldCalYr <- subset(reconSubset, reconSubset[,4] <= oldCalYr)
   reconSubsetFinal <- subset(reconSubsetOldCalYr, reconSubsetOldCalYr[,4] >= youngCalYr)

   message("Subset of reconstruction by calendar year BP created!")
   message(paste("Subset has",nrow(reconSubsetFinal),"records"))

   countSites <- unique(reconSubsetFinal[,1])
   message(paste("Subset has",length(countSites),"sites"))

   return (reconSubsetFinal)
}subsetReconBySiteRes <- function(reconSubsetSiteRes,siteRes,siteResColID)
{
   countSites = 0
   reconSubsetFinal <- subset(reconSubsetSiteRes, reconSubsetSiteRes[,siteResColID] <= siteRes)
   message("Subset of reconstruction by fossil pollen site resolution created!")
   message(paste("Subset has",nrow(reconSubsetFinal),"records"))
   countSites <- unique(reconSubsetFinal[,1])
   message(paste("Subset has",length(countSites),"sites"))
   return (reconSubsetFinal)
}subsetReconBySpatialExtent <- function(reconDataFile,lonMax,lonMin,latMax,latMin)
{
    require(geosphere)

    countSites = 0
    # need checks for each time the user subsets
    # max 3 subsets (1 by lat/lon, 1 by oldest cal yr bP, 1 by site resolution)

    while (is.na(lonMax) | lonMax <= -170 | lonMax >= -40 | (lonMax - lonMin > -5))
       lonMax <- as.numeric(readline("Enter maximum (furthest west) longitude [e.g. -90]: " ))
    while (is.na(lonMin) | lonMin <= -170 | lonMin >= -40 | (lonMax - lonMin > -5))
       lonMin <- as.numeric(readline("Enter minimum (furthest east) longitude [e.g. -75]: " ))
    while (is.na(latMin) | latMin >= 90 | latMin <= 15 | (latMax - latMin < 5))
       latMin <- as.numeric(readline("Enter minimum (furthest south) latitude [e.g. 45]: " ))
    while (is.na(latMax) | latMax >= 90 | latMax <= 15 | (latMax - latMin < 5))
       latMax <- as.numeric(readline("Enter maximum (furthest north) latitude [e.g. 60]: " ))

    reconDataFile <- subset(reconDataFile, reconDataFile$LATITUDENORTH <= latMax & reconDataFile$LATITUDENORTH >= latMin &
          reconDataFile$LONGITUDEEAST >= lonMax & reconDataFile$LONGITUDEEAST <= lonMin)

    message("Subset of reconstruction by lat/lon created!")
    message(paste("Subset has",nrow(reconDataFile),"records"))
```

```
    countSites <- unique(reconDataFile[,1])
    message(paste("Subset has",length(countSites),"sites"))

    # create a subset box with coordinates that were entered in
    bbox <- rbind(c(lonMax,latMax), c(lonMax,latMin), c(lonMin,latMin), c(lonMin,latMax))

    # calculate area of polygon with function from geosphere package
    subsetArea <- round(areaPolygon(bbox)/1000000
    message(paste("Subset area bounded by entered coordinates is",subsetArea,"km2"))

    # calculate site density (# of sites per km2)
    siteDensity <- length(countSites) / subsetArea
    message(paste("Site density of this subset area is",siteDensity,"sites per km2"))

    subsetList <- list(reconDataFile, latMax, latMin, lonMax, lonMin)
    return(subsetList)
}subsetSPjoinReconModBySearchRad <- function(reconSPjoin, searchRadius)
{
   # only include rows where modern site was within X degrees of fossil site
   # potentially keep a default of 2 degrees, if search radius is less, more
   # sites will be excluded, increasing the search radius does not matter
   # does not matter if <= searchRadius or < searchRadius is used

   reconSPjoinSubsetSearchRad <- reconSPjoin[reconSPjoin$Distance < searchRadius,]
   return(reconSPjoinSubsetSearchRad)
}tuneYearAD <- function(anByCalYr, yearInt, oldCalYr, youngCalYr)
{
  startYear = 0
  endYear = 0
  tuneDefault2k = 0
  tunedDate = NA
  reconFlag2k = FALSE

  # need to convert cal YR values to tuned AD values (2000 bp becomes 350 AD, 1900 bp becomes 450 AD)
  if (yearInt == 100 & oldCalYr == 2000 & youngCalYr == 0)
  {
    reconFlag2k = TRUE
    while (is.na(tuneDefault2k) | tuneDefault2k < 1 | tuneDefault2k > 2)
       tuneDefault2k <- as.numeric(readline("Reconstruction of the last 2k at 100 yr intervals detected. Use default <1> or manual <2>
             tuning? "))
    # only for 2k reconstruction at 100 year intervals
    if (tuneDefault2k == 1)
    {
      anByCalYr[21,1] <- 350
      anByCalYr[20,1] <- 450
      anByCalYr[19,1] <- 550
      anByCalYr[18,1] <- 650
      anByCalYr[17,1] <- 750
      anByCalYr[16,1] <- 850
      anByCalYr[15,1] <- 950
      anByCalYr[14,1] <- 1050
      anByCalYr[13,1] <- 1150
      anByCalYr[12,1] <- 1250
      anByCalYr[11,1] <- 1350
      anByCalYr[10,1] <- 1450
      anByCalYr[9,1] <- 1550
      anByCalYr[8,1] <- 1600
      anByCalYr[7,1] <- 1650
      anByCalYr[6,1] <- 1700
      anByCalYr[5,1] <- 1750
      anByCalYr[4,1] <- 1800
      anByCalYr[3,1] <- 1850
      anByCalYr[2,1] <- 1900
      anByCalYr[1,1] <- 1950

      startYear = anByCalYr[1,1]
      endYear = anByCalYr[21,1]
    }else
    {
      # allow user to enter in tuned dates for each calendar year BP interval?

      rowIndex = 1

      while (rowIndex <= nrow(anByCalYr))
      {
        # ask user to enter in tuned date for selCalYrBP
```

```
      # no way to check validity of tuned date entered

      tunedDate = NA # reset this
      while (is.na(tunedDate))
      {
        tryCatch({
        tunedDate <- as.numeric(readline(paste("Enter in tuned year AD date for",anByCalYr[rowIndex,1],"bp: ")))
        }, error = function(ex) {
        })
      }

      anByCalYr[rowIndex,1] <- tunedDate

      # go to next row in year BP col
      rowIndex = rowIndex + 1
    }
    startYear <- anByCalYr[1,1]
    endYear <- anByCalYr[nrow(anByCalYr),1]
  }
}
if (!reconFlag2k)
{
  # manual tuning only for all other reconstructions

  rowIndex = 1

  while (rowIndex <= nrow(anByCalYr))
  {
    # ask user to enter in tuned date for selCalYrBP
    # no way to check validity of date entered

    tunedDate = NA # reset this
    while (is.na(tunedDate))
    {
      tryCatch({
      tunedDate <- as.numeric(readline(paste("Enter in tuned year AD date for",anByCalYr[rowIndex,1],"bp (use negative for tuned year
            BC): ")))
      }, error = function(ex) {
      })
    }

    anByCalYr[rowIndex,1] <- tunedDate

    # go to next row in year BP col
    rowIndex = rowIndex + 1
  }
  startYear <- anByCalYr[1,1]
  endYear <- anByCalYr[nrow(anByCalYr),1]
}
tuneList <- list(anByCalYr, startYear, endYear)
return(tuneList)
}writeCustomBin <- function(Directory,StationIds,outFile,inFile = "data.bin"){
options(bigmemory.allow.dimnames=TRUE)
if(!grepl("\\.bin",outFile))stop("please use a bin extension")
dname <- sub("bin","desc",outFile)

BigData <- readBerkeleyData(Directory = Directory,filename= inFile)

Ids <- BigData[,"Id"]

dex <- which(Ids %in% StationIds )

if(length(dex) == 0)stop("bad inventory")

# create a custom filebacked big matrix using only the stations
# in the inventory
cnames <- unlist(dimnames(BigData)[2])

Custom <- filebacked.big.matrix(nrow = length(dex) , ncol = ncol(BigData),
                            dimnames = list(NULL,cnames),
                            init = NULL,
                            backingpath = Directory,
                            backingfile = outFile,
                            descriptorfile = dname,
                            type = "double")
```

```
  dex <- sort(dex)

  newCount <- 0
  for(lineCount in dex) {
    DataLine <- BigData[lineCount,]
    newCount <- newCount + 1
    Custom[newCount,]<-DataLine

  }



}writeReconObj <- function(NA_recons_Obj, climVar, pSum, psCount)
{
  reconFileStr = ""

  if (pSum == 0)
    reconFileStr = paste("PS_CUSTOM_",psCount,"_",toupper(climVar),".recon",sep="")
  else
    reconFileStr = paste("PS", pSum,"_",toupper(climVar),".recon",sep="")

  tryCatch({write.csv(NA_recons_Obj,reconFileStr,row.names=FALSE)}, error = function(e) {message("Error writing NA_recons_Obj to disk
        !")}, warning = function(w){})


  message(paste("Reconstruction file",reconFileStr,"written to working directory!",sep=" "))
}
```

Appendix D

Additional Methods & Scripts

D.1 Overview of reconstruction methodology

The original methodology for the reconstruction is below, developed as a series of excel macros and ESRI ModelBuilder™. All of these steps are incorporated into the final R program.

```
Step 1: Setup of pollen data files
A) decide on pollen sum (35, 63, 134, 135)
B) setup fossil pollen percentages file (from latest version of NAPD in Neotoma)*
C) setup modern pollen percentages file (from latest version of Whitmore v1.72)*

* run macro to convert counts to percentages, also macro to split into PINUS and PINUS_SE (optional)

Step 2 and 3: Run MATTOOLS in R
A) use html page (with javascript) to generate MATTOOLS code in R
B) run the R code generated in step 2A (this will include the following)
i) read modern pollen percentages file
ii) read fossil pollen percentages file
iii) mat.dissim to use the MAT to reconstruct
iv) finally cbind recon to modern column
v) write.csv for climate variable of interest
C) output file has 30,861 records excluding column headers (all cal yr BP), number of columns depends on number of analogs selected in
      Step 2B(iii)

Step 4: Linear interpolation and formatting
A) sort CalYrBP field ascending and select records for time period of interest (e.g. last 2k, which will have 8,278 records excluding
      column headers)
B) select fields (siteID, lat, lon, CalYrBP, InterpVal  there could be InterpVal2, InterpVal3, etc. if doing more analogs than only the
       top analog)
C) copy to 2nd worksheet where either (i) linear interpolation formulas are already setup and interpolated values will be automatically
       calculated or (ii) call linear interpolation function (use the code from completed macro) to do the interpolation (ask user
      which method to use: case 1, G to K with dummy and flag rows - simply copy the correct columns of data from worksheet 1 to
      worksheet 2; case 2, create new worksheet, copy data then run linear interpolation code**

** this only needs to be done the first time the reconstruction is run, assuming the same time period is used, which will mean the
      starting number of records will be the same (as in Step 3A)

D) for the last 2k, the 100 yr interpolation of ~8,278 records will result in ~10,887 records
E) copy this output (5 columns as in Step 3B excluding flag -999 and dummy row with zeroes) to new worksheet
F) format the lat/lon columns to number and 5 decimal places
G) save this worksheet as a .csv file

Step 5: Spatial join of reconstruction to modern to calculate anomalies
A) x-y map the csv file from step 3G
B) export the events layer to a shapefile
C) spatial join of reconstruction shapefile to modern climate data shapefile
```

```
*** (reconstruction is target, modern climate is join features, keep all common features, search method is CLOSEST with a 2 degree
      search radius)

*** Note that 133 records do not join with a search radius of 2 degrees

D) create new field (type double) for anomaly of climate variable (and chosen analog) and use field calculator [reconstructed value
      modern value]

Step 6: calculate average of anomalies for all sites by 100 year cal yr BP intervals (Excel macro "calcMTWAanomaliesByCalYrBP" does the
       following)
A) select field with cal yr BP intervals (usually column F)
B) sort this column descending
C) go to last row + 1 and add flag of -999 in cal yr BP field
D) calculate average of anomalies by cal Yr BP for climate variable, excluding records that did not join in Step 4C (133 records for
      the last 2k)
E) create a time series plot from the output on worksheet #2 (optional)
F) save the .dbf file as an Excel workbook
```

A.1 LOCATION OF FILES:

NOTE: all scripts are located in the "R:/LPC/LPC PAPERS/NAM_2k/scripts" folder, macros are in the "vba_macros" subfolder

R scripts are in the "scripts" folder, linear interpolation scripts are in the "interpolation" subfolder

To run a macro open the excel file, press ALT + F11, then paste code of macro in "ThisWorkbook". Macro security needs to be set to no higher than medium in order to run these macros.

A.2 DATA FILES NEEDED:

These are located in: "R:/LPC/LPC PAPERS/NAM2k/db_files/FINAL"

Modern pollen database: "whitmore_modernpollen_final_1-7-3.csv"

Ensure you have the latest version, it is available at: http://www.lpc.uottawa.ca/data/modern/

The latest version of Neotoma was used for the fossil pollen database (NAPD).

If a new version is available, download it, and create a new workbook from the POLLEN and CLIMATE+BIOCLIMATE worksheets, save it as a XLS file. Then run the MS Excel™macro to convert the pollen counts into percentages (paste the code below into VB editor in MS Excel™using ALT + F11). Delete the worksheet with the counts and save the worksheet with the percentages as a csv file.

Ensure that the database files are in csv rather than an xls format.
e.g. "NA_recons_fossilpollen_2009_final.csv"

Otherwise a non-numeric error will be generated when loading the file in R.

A.3 ADDITIONAL SETUP INFORMATION (before running MATTOOLS):

MS Excel™setup: While working with the pollen database file it is best to show column numbers in MS Excel™. Do this using [Tools - Options] then [General] tab. Under settings check the [R1C1 reference style] box.

CSV file format: number of columns of taxa for modern and fossil pollen databases need to match exactly before running MATTOOLS. Also, check the column number references for both modern and fossil database files.

For example:
35 - (cols 14:49 for taxa)
64 - (cols 14:76 for taxa)
134 - (cols 14:147, 16:149 for taxa)
135 - (cols 14:148, 16:150 for taxa)

MATTOOLS setup in R:

From the "Packages" menu in R select "Install package(s)" and select MATTOOLS. Once the package is installed, use the command "library(MATTOOLS)" to

load the package.

R code for MATTOOLS can be generated using the html/javascript page at:
R:\LPC\LPC PAPERS\NAM_2k\scripts\code_generator_tool\NA_recons_mattools_code_generator.html

The columns with modern climate variables from Whitmore need to be referenced correctly in the cbind code:

Pollen Sum 35 - 61 to 129
Pollen Sum 64 - 88 to 156
Pollen Sum 134 - 160 to 228
Pollen Sum 135 - 161 to 229

Running the mat.dissim function on the two database files takes approx 5-15 min depending on the speed of the computer.

Section I. Selecting Taxa, fossil & modern files setup and running MATTOOLS

1. Select taxa for pollen sum according to paper by Williams and Shuman (2008).
2. If pollen sum is less than 134/135, it will be necessary to calculate a new total pollen sum for all taxa for each site in both the modern and fossil sites.
3. The resultant file will have the counts, may need to split PINUS taxa into PINUS and PINUS_SE, depending on pollen sum used (run macro PinusXtoPinusSE_fossil or PinusXtoPinusSE_modern to do this).

   Pollen taxa column refs need to be modified depending on pollen sum used (35, 64, 134, 135). This needs to be done for both the modern and fossil pollen databases.

   135 has all taxa plus PinusSE (Williams and Shuman, 2008)
   Columns are 14 to 148

A macro was used to put the PINUS percentages flagged with a 1 into a new taxa group PinusSE

```
Sub PinusToPinusSE()
' at each fossil site with a pinus count, check lat/lon of site
' if lon is >= -97.5 and lat is <= 40.8 (modern ranges) then do the following:
' 1. copy Pinus count to new column PinusSE
' 2. replace Pinus count with a zero
' 3. if lat/lon is outside of boundaries
' 4. leave pinus count as is, write a zero to PinusSE column
'
' Script written by Matthew Ladd
' Revised: February 25, 2011

Dim wb As Workbook
Dim curRow, lastRow, pinus1Col, pinus2Col, pinusSECol As Long
Dim siteLat, siteLon As Single

Set wb = ThisWorkbook

' first row with counts
curRow = 2

' last row with counts
lastRow = 30862

' lat column is 4
siteLat = 4

' lon column is 5
siteLon = 5

' PINUS columns to be added to SE are 55 & 57 (DIP & UNDIFF)
pinus1Col = 55
pinus2Col = 57

' PINUS_SE column is 58
pinusSECol = 58

' go through all rows with counts
Do While curRow <= lastRow
    ' first check if site has a Pinus DIP & UNDIFF count > 0
    If wb.Worksheets(1).Cells(curRow, pinus1Col).Value > 0 Or wb.Worksheets(1).Cells(curRow, pinus2Col).Value > 0 Then
    ' then check lat/lon of site to see if it's within the modern range of Pinus SE
        If wb.Worksheets(1).Cells(curRow, siteLat).Value <= 40.8 And wb.Worksheets(1).Cells(curRow, siteLon).Value >= -97.5 Then
            ' copy appropriate value to PINUS_SE column
            wb.Worksheets(1).Cells(curRow, pinusSECol).Value = wb.Worksheets(1).Cells(curRow, pinus1Col).Value + wb.Worksheets(1).Cells(
                  curRow, pinus2Col).Value
            ' remove duplicate value from PINUS columns, replace with zero
            wb.Worksheets(1).Cells(curRow, pinus1Col).Value = 0
            wb.Worksheets(1).Cells(curRow, pinus2Col).Value = 0
        Else 'if site not in PinusSE range, then just write a zero to PINUS_SE column
            wb.Worksheets(1).Cells(curRow, pinusSECol).Value = 0
        End If
    Else ' write a zero in PinusSE for any blank cells
        wb.Worksheets(1).Cells(curRow, pinusSECol).Value = 0
    End If
    ' go to next row
    curRow = curRow + 1
Loop

MsgBox "PINUS and PINUS_SE counts have been updated!"

End Sub
```

4. The counts need to be converted to percentages (run convertCountsToPercMacro).

```
Sub countsToPerc()
' converts pollen counts to percentages in modern pollen DB (Whitmore)
' Script written by Matthew Ladd
```

```
Dim wb As Workbook
Dim curRow, lastRow, curCol, lastCol, pollSum, pollCount As Long
Dim pollPerc As Double

Set wb = ThisWorkbook

' first row with counts
curRow = 2

' last row with counts (change this if neccessary)
lastRow = 4834

' first col of pollen taxa (change this if neccessary)
' note that col 13 'M' is pollen sums (change this if neccessary)
curCol = 14

' last col of pollen taxa (change this if neccessary)
lastCol = 76

' First worksheet has counts, Second worksheet will have percentages
wkCount = 1
wkPerc = 2

' loop for taxa in each column
Do While curCol <= lastCol
    ' go through all rows with counts (reset current row)
    curRow = 2
    Do While curRow <= lastRow
        ' get the count value for cell
        pollCount = wb.Worksheets(wkCount).Cells(curRow, curCol).Value
        ' divide it by the POLLSUM value in col 13 (M) to get the percentage value
        pollSum = wb.Worksheets(wkCount).Range("M" & curRow).Value
        pollPerc = pollCount / pollSum * 100
        ' write this value to proper column on new worksheet (format cells to 1 dec place)
        wb.Worksheets(wkPerc).Cells(curRow, curCol).NumberFormat = "0.0"
        wb.Worksheets(wkPerc).Cells(curRow, curCol).Value = pollPerc
        ' go to next row
        curRow = curRow + 1
    Loop
    'go to next column
    curCol = curCol + 1
 Loop

MsgBox "Calculation of pollen percentages is complete. Please copy the percentages (2nd worksheet) to first worksheet and re-save
      workbook (use save as)."

End Sub
```

5. Check the file carefully and ensure the excel file is saved as a CSV (MATTOOLS requires input files in CSV format).

6. Now open the html file to generate the R code (the html file in the "R:/LPC/LPC PAPERS/NAM_2k/scripts/code_generator_tool" Folder, see Appendix D.8), select the fossil and modern files (same pollen sum for each, number of taxa columns must be identical).

7. Paste the R code into a new script file in TINN-R and save it then start R using the R workspace file in scripts folder.
   (NOTE: TINN-R is a free script editor available at http://sourceforge.net/projects/tinn-r/)

The script generated in step 6 will look like the one below:

EXAMPLE RECONSTRUCTION (Pollen Sum 134):

Load the DB files:

```
Modpoll <- read.csv("R:\\LPC\\LPC PAPERS\\NAM_2k\\db_files\\_NEW\\FINAL\\whitmore_modernpollen_final_v1-7-2_134_types.csv", header=TRUE
      , sep=",", na.strings="NA", dec=".", strip.white=TRUE)
NA_fossil <- read.csv("R:\\LPC\\LPC PAPERS\\NAM_2k\\db_files\\_NEW\\FINAL\\NA_recons_fossilpollen_2009_final_134_types.csv", header=
      TRUE, sep=",",na.strings="NA", dec=".", strip.white=TRUE)
```

Run mat.dissim:

```
NA_recons_SCD=mat.dissim(inFossil=NA_fossil,inModern=Modpoll,llMod=10:11,modTaxa=14:147,llFoss=4:5,fosTaxa=16:149,numAnalogs=20,counts=
      T,dist.method="spherical")
```

Make sure to run the following command:

```
x = NA_recons_SCD
```

Run script to format the output files:
Change column (i.e. 160) each time script is run to reconstruct a different climate variable: e.g. in Pollen Sum 134, TFEB is 160, TMAR is 160, etc.

```
source("H:\\_NA_recons\\scripts\\NA_recons_allVars.r")

Code courtesy of Dr. Mike Sawada:
ReconsTJAN=cbind(NA_fossil[,c(2,4:6,9:11)],t(NA_recons_SCD$sqdist[1:20,]),
ID1=Modpoll[x$position[1,],1],ID2=Modpoll[x$position[2,],1],
ID3=Modpoll[x$position[3,],1],ID4=Modpoll[x$position[4,],1],
ID5=Modpoll[x$position[5,],1], ID6=Modpoll[x$position[6,],1],
ID7=Modpoll[x$position[7,],1], ID8=Modpoll[x$position[8,],1],
ID9=Modpoll[x$position[9,],1],ID10=Modpoll[x$position[10,],1],
ID11=Modpoll[x$position[11,],1], ID12=Modpoll[x$position[12,],1],
ID13=Modpoll[x$position[13,],1], ID14=Modpoll[x$position[14,],1],
ID15=Modpoll[x$position[15,],1], ID16=Modpoll[x$position[16,],1],
ID17=Modpoll[x$position[17,],1], ID18=Modpoll[x$position[18,],1],
ID19=Modpoll[x$position[19,],1], ID20=Modpoll[x$position[20,],1],
TJAN1=Modpoll[x$position[1,],160],TJAN2=Modpoll[x$position[2,],160],
TJAN3=Modpoll[x$position[3,],160],TJAN4=Modpoll[x$position[4,],160],
TJAN5=Modpoll[x$position[5,],160],TJAN6=Modpoll[x$position[6,],160],
TJAN7=Modpoll[x$position[7,],160],TJAN8=Modpoll[x$position[8,],160],
TJAN9=Modpoll[x$position[9,],160],TJAN10=Modpoll[x$position[10,],160],
TJAN11=Modpoll[x$position[11,],160],TJAN12=Modpoll[x$position[11,],160],
TJAN13=Modpoll[x$position[13,],160],TJAN14=Modpoll[x$position[14,],160],
TJAN15=Modpoll[x$position[15,],160],TJAN16=Modpoll[x$position[16,],160],
TJAN17=Modpoll[x$position[17,],160],TJAN18=Modpoll[x$position[18,],160],
TJAN19=Modpoll[x$position[19,],160],TJAN20=Modpoll[x$position[20,],160])
write.csv(ReconsTJAN," R:\\LPC\\LPC PAPERS\\NAM_2k\\scripts\\ReconsTJAN.csv",quote=F,row.names=F)
```

NOTE: the path can be changed depending on the variable!

8. Now run this script in R as follows:

```
source("R:\\LPC\\LPC PAPERS\\NAM_2k\\scripts\\new_script.r")
Modpoll <- read.csv("R:\\LPC\\LPC PAPERS\\NAM_2k\\db_files\\FINAL\\whitmore_modernpollen_perc_final_v1-7-2_135_types.csv", header=TRUE,
       sep=",", na.strings="NA", dec=".", strip.white=TRUE)

NA_fossil <- read.csv("R:\\LPC\\LPC PAPERS\\NAM_2k\\db_files\\FINAL\\NA_recons_fossilpollen_perc_final_135_types.csv", header=TRUE, sep
      =",",na.strings="NA", dec=".", strip.white=TRUE)
```

```
NA_recons_SCD=mat.dissim(inFossil=NA_fossil,inModern=Modpoll,llMod=10:11,modTaxa=14:148,llFoss=4:5,fosTaxa=16:150,numAnalogs=20,counts=
      F,dist.method="spherical")

x = NA_recons_SCD

ReconsTJAN=cbind(NA_fossil[,c(2,4:6,9:11)],t(NA_recons_SCD$sqdist[1:20,]),
ID1=Modpoll[x$position[1,],1],ID2=Modpoll[x$position[2,],1],
ID3=Modpoll[x$position[3,],1],ID4=Modpoll[x$position[4,],1],
ID5=Modpoll[x$position[5,],1], ID6=Modpoll[x$position[6,],1],
ID7=Modpoll[x$position[7,],1], ID8=Modpoll[x$position[8,],1],
ID9=Modpoll[x$position[9,],1],ID10=Modpoll[x$position[10,],1],
ID11=Modpoll[x$position[11,],1], ID12=Modpoll[x$position[12,],1],
ID13=Modpoll[x$position[13,],1], ID14=Modpoll[x$position[14,],1],
ID15=Modpoll[x$position[15,],1], ID16=Modpoll[x$position[16,],1],
ID17=Modpoll[x$position[17,],1], ID18=Modpoll[x$position[18,],1],
ID19=Modpoll[x$position[19,],1], ID20=Modpoll[x$position[20,],1],
TJAN1=Modpoll[x$position[1,],160],TJAN2=Modpoll[x$position[2,],160],
TJAN3=Modpoll[x$position[3,],160],TJAN4=Modpoll[x$position[4,],160],
TJAN5=Modpoll[x$position[5,],160],TJAN6=Modpoll[x$position[6,],160],
TJAN7=Modpoll[x$position[7,],160],TJAN8=Modpoll[x$position[8,],160],
TJAN9=Modpoll[x$position[9,],160],TJAN10=Modpoll[x$position[10,],160],
TJAN11=Modpoll[x$position[11,],160],TJAN12=Modpoll[x$position[11,],160],
TJAN13=Modpoll[x$position[13,],160],TJAN14=Modpoll[x$position[14,],160],
TJAN15=Modpoll[x$position[15,],160],TJAN16=Modpoll[x$position[16,],160],
TJAN17=Modpoll[x$position[17,],160],TJAN18=Modpoll[x$position[18,],160],
TJAN19=Modpoll[x$position[19,],160],TJAN20=Modpoll[x$position[20,],160])
write.csv(ReconsTJAN,"R:\\LPC\\LPC PAPERS\\NAM_2k\\db_files\\OUTPUT\\PS135\\PS135_RECONS_TJAN.csv",quote=F,row.names=F
```

The output file will have 30,862 rows (including the header row). Once you have the output file, you will see columns with siteID, lat, lon, altitude, sampleID, Age and CalyrBP. Columns with ID 1-20 and variable name 1-20, and 1-20. The 1-20 columns contain the SCD values based on the top 20 analogs (top analog has the lowest SCD value). The ID 1-20 columns have the site ID of the closest modern match to the fossil site. The variable name (e.g. TJAN) 1-20 columns have the values of the variable for the top 20 analogs.

Save a copy of the db file (in csv format) and keep only the first 7 columns (siteID to CalyrBP) and the 1, ID1, variable name 1 columns (because you are using the top analog).

This file is the reconstruction that will be used with the site averaging macro. This macro (C.4.) will calculate the average value for the specified variable at each site.

Section II. Import into MS Access™, export from MS Access™and linear interpolation of files

9. Check the output files and move them into a separate directory before doing

the next pollen sum. Also, be sure to run the macro formatLatLonColumns on all csv files before the import into MS Access™. The reason is that that we want MS Access™to recognize that lat and lon are both fields with decimal places when importing so the macro sets the numberformat of those 2 columns to 0.0 (only 1 decimal place is needed). Or you can simply set the format of the cells to number with 4 decimal places. If the macro is not run then MS Access™will end up rounding the lat values to whole numbers. Column headers should not have any spaces which is important for x-y mapping the data in GIS software. The macro formatColHeaders will remove spaces in the headers if you have a large number of columns to change.

Note about decimal places: The climate variable of interest should only be formatted as numeric but no decimal places set (leave the default cell formatting). These values should not be rounded until producing the final time series. Lat/Lon is potentially an issue as the fossil sites have 5 significant figures whereas the modern sites have only 3 significant figures. The fossil sites can be formatted to 1 significant figure and this will not affect the results.

10. Import CSV files into MS Access™database (run macro importMultipleCSVintoAccess).The database may be compacted using [Tools - Database Utilities - Compact and Repair Database] to reduce its size.

```
Sub Import_multiple_csv()

Dim fs, fldr, fls, fl
Set fs = CreateObject("Scripting.FileSystemObject")
Set fldr = fs.getfolder("R:\LPC\LPC PAPERS\NAM_2k\db_files\OUTPUT\PS_35\")
Set fls = fldr.files

For Each fl In fls

   If Right(fl.Name, 4) = ".csv" Then
      DoCmd.TransferText acImportDelim, , fl.Name, "R:\LPC\LPC PAPERS\NAM_2k\db_files\OUTPUT\PS_35\" & fl.Name, True
   End If

Next fl

End Sub
```

11. Once in Access, under queries, create a new query in design view. Select the table (e.g. PS35_RECONS_MTWA_csv) then add the fields: SiteID, LatitudeNorth, LongitudeEast, CalYrBp and then MTWA1 to MTWA20 (variable of interest with 20 analogs).

For the CalYrBP select the time period of interest (e.g. criteria is $\geq$ -100 and $\leq$ 2100 for the last 2,000 years). Save this query and then run it checking the output. The total number of records will be 8,279 records for the period above (including the column header row).

12. Export the query as an Excel 97-2003™file and then open that file and save as CSV, the file should be saved to the directory of spath below: "R:/LPC/LPC PAPERS/NAM_2k/scripts/interpolation/SOURCE_FILES/"

13. Run the linear interpolation script on selected climate variable for each pollen sum (linear_interp_test_v2.r but this script needs to be modified depending on the climate variable used, as follows):

    NOTE: the linear interpolation macro for MS Excel™can be used instead, however it takes longer to run than the R script, there should be 10,888 records after the interpolation (including the column header row).

```
spath <- "R:\\LPC\\LPC PAPERS\\NAM_2k\\scripts\\interpolation\\SOURCE_FILES\\MTCO2000.csv"
dpath <- "R:\\LPC\\LPC PAPERS\\NAM_2k\\scripts\\interpolation\\OUTPUT\\MTCO\\"

vName = "MTCO" # name of climate variable to be reconstructed

# function written by Mike Sawada and modified by Matthew Ladd (March 15, 2011)
# not neccessary to change anything in this function!

# modifed to read and write lat/lon values, neccessary for mapping in GIS software
reconTest = function(fossilDataFrame,colFirstTaxa=C1,colLat=2,colLon=3,colCalmean=4,colE.=1,target.dates=targets)
{
  holdnames<-as.vector(unlist(names(fossilDataFrame)))

  # get enumbers and the date column plus the columns with the variables to interpolate.
  datainterp1 <- as.matrix(fossilDataFrame[,c(colE.,colLat,colLon,colCalmean,colFirstTaxa)])

  #make no dimnames to speed up calculations
  dimnames(datainterp1)<-NULL

  #get list of enums for each site
  enumbers<-as.numeric(unique(datainterp1[,1]))

  #newcolend<-dim(datainterp1)[6]
  outtable<-vector("numeric")

  for(i in 1:length(enumbers))
  {
    datainterp<-datainterp1[which(datainterp1[,1] == enumbers[i]), ]
    if(length(dim(datainterp)) == 2)
    {

      interpolatedPrcts<-data.frame(approx(datainterp[,4],datainterp[,5],target.dates))

      inter<-cbind(E. = rep(fossilDataFrame[which(fossilDataFrame[,colE.] == enumbers[i])[1],colE.],length(target.dates)),Lat = rep(
            fossilDataFrame[which(fossilDataFrame[,colE.] == enumbers[i])[1],colLat],length(target.dates)),Lon = rep(fossilDataFrame[
            which(fossilDataFrame[,colE.] == enumbers[i])[1],colLon],length(target.dates)),interpolatedPrcts)

      outtable <- rbind(outtable,inter[which(is.na(inter[,5])==F), ])
    }
  }
  return (outtable)
}
# END OF FUNCTION #

# This part written by Matt Ladd
```

```
# now read in the source file, set destination file path (change if neccessary)
spath <- "R:\\LPC\\LPC PAPERS\\NAM_2k\\scripts\\interpolation\\SOURCE_FILES\\mipt2000.csv"
dpath <- "R:\\LPC\\LPC PAPERS\\NAM_2k\\scripts\\interpolation\\OUTPUT\\MIPT\\"
recon <- read.csv(spath, header=TRUE, sep=",", na.strings="NA", dec=".", strip.white=TRUE)

# get number of cols, create data frame for matrix read in (recon)
nc = ncol(recon)
reconDF <- data.frame(recon)

# in calYrBP
startYr = 0 # this is the the most recent year
endYr = 10000 # this is the the oldest year
yrInt = 100 # time step for interpolations (best to use 50 or 100)

fn=0 # counter for file names


vName = "MIPT" # name of climate variable to be reconstucted
ftype = ".csv" # output file type (best to use .csv or .txt)


for (k in 5:nc) # first climate var col is 5, last column is nc
{
   # run function and write output file
   fn=fn+1 # variable name for each analog starts at 1
   reconOut = reconTest(reconDF,k,,,,,seq(startYr,endYr,yrInt))
   outpath = paste(dpath,vName,fn,"_INTERP",ftype)
   write.table(reconOut,outpath,sep=",",quote=F)
}

### END OF SCRIPT ###
```

Check that the CSV files output in dpath to make sure they are correct. There should be 20 of them if using 20 analogs. If only using the top analog this step can be skipped.

14. Open combine_interp_csv.xls and then run the macro to combine interpolation files into one CSV file for each climate variable for each pollen sum. When the select folder dialog appears, select the same path as dpath above. After script finishes, check the output excel file and re-save as CSV in the appropriate directory.

```
' Author: Matthew Ladd
' Notes: adapted from code at: http://www.rondebruin.nl/copy3.htm
' Last modified: March 19, 2011

Sub combineInterpCSVfiles()

Dim MyPath, fldName As String
Dim FilesInPath As String
Dim MyFiles() As String
Dim SourceRcount As Long
Dim Fnum, Cnum, Cind As Long
Dim mybook As Workbook
Dim basebook As Workbook
Dim myrange As Range
Dim Msg As String
Dim fldr As FileDialog
Dim sItem As String
Set fldr = Application.FileDialog(msoFileDialogFolderPicker)


'Default path\folder where the files are on your machine
' NOTE: THIS CAN BE CHANGED IF NECCESSARY
strPath = "R:\LPC\LPC PAPERS\NAM_2k\scripts\interpolation\OUTPUT\"
```

```
With fldr
   .Title = "Select a Folder..."
   .AllowMultiSelect = False
   .InitialFileName = strPath
   If .Show <> -1 Then GoTo NextCode
   MyPath = .SelectedItems(1)
End With
NextCode:
GetFolder = sItem
Set fldr = Nothing


'Add a slash at the end if the user forget it
If Right(MyPath, 1) <> "\" Then
   MyPath = MyPath & "\"
End If

'If there are no Excel files in the folder exit the sub
FilesInPath = Dir(MyPath & "*.csv")

If FilesInPath = "" Then
   MsgBox "No files found"
   Exit Sub
End If

On Error GoTo CleanUp

Application.ScreenUpdating = False
Application.DisplayAlerts = False 'disable all alert prompts

Set basebook = ThisWorkbook

'Fill the array(myFiles)with the list of Excel files in the folder
Fnum = 0

Do While FilesInPath <> ""
   Fnum = Fnum + 1
   ReDim Preserve MyFiles(1 To Fnum)
   MyFiles(Fnum) = FilesInPath
   FilesInPath = Dir()
Loop

'Loop through all files in the array(myFiles)

For Fnum = LBound(MyFiles) To UBound(MyFiles)

   Set mybook = Workbooks.Open(MyPath & MyFiles(Fnum))

   ' then count the number of columns
   Set myrange = Rows(1)
   ncol = Application.WorksheetFunction.CountA(myrange)

   ' determine last row number of imported sheet
   lastRow = mybook.ActiveSheet.UsedRange.SpecialCells(xlCellTypeLastCell).Row

   ' get the name of the variable (used for column headers)
   fldName = Left(mybook.Name, 6)

   ' need to ensure column of data is pasted into correct column of combined worksheet
   ' note that each climate variable has 20 analogs hence 20 columns
   ' first 4 cols are siteID, lat, lon & calMean

   'Array of files are ordered as follows:
   '1,10,11,12,13,14,15,16,17,18,19,2,20,3,4,5,6,7,8,9

   Select Case (Fnum)

       Case 1 ' eg - TJUL1
         newaddress = "E1" '6th column is E

       Case 2 'eg - TJUL10
         newaddress = "N1"

       Case 3 'eg - TJUL11
         newaddress = "O1"
```

```
    Case 4 'etc as above
      newaddress = "P1"

    Case 5
      newaddress = "Q1"

    Case 6
      newaddress = "R1"

    Case 7
      newaddress = "S1"

    Case 8
      newaddress = "T1"

    Case 9
      newaddress = "U1"

    Case 10
      newaddress = "V1"

    Case 11
      newaddress = "W1"

    Case 12 ' eg - TJUL 2
      newaddress = "F1"

    Case 13
      newaddress = "X1"

    Case 14
      newaddress = "G1"

    Case 15
      newaddress = "H1"

    Case 16
      newaddress = "I1"

    Case 17
      newaddress = "J1"

    Case 18
      newaddress = "K1"

    Case 19
      newaddress = "L1"

    Case 20
      newaddress = "M1"

    'Case Else
      ' not needed here

End Select


If Fnum = 1 Then
    'copy columns needed and paste into base worksheet
    mybook.Worksheets(1).Range("B1:F1").EntireColumn.Copy
    basebook.Sheets(1).Paste

Else 'Fnum > 1

    'first climate var column should be copied to first workbook
    mybook.Worksheets(1).Range("F1").EntireColumn.Copy
    'ensure that column data is pasted to next column over
    basebook.Sheets(1).Paste basebook.Sheets(1).Range(newaddress)

End If


On Error Resume Next

ActiveSheet.Name = fldName & "_INTERP"
```

```
   On Error GoTo 0

   ' You can use this if you want to copy only the values
   ' With ActiveSheet.UsedRange
   ' .Value = .Value
   ' End With

   mybook.Close savechanges:=False

Next Fnum

' now write column headers
basebook.Sheets(1).Range("A1").Value = "siteID"
basebook.Sheets(1).Range("B1").Value = "LAT"
basebook.Sheets(1).Range("C1").Value = "LON"
basebook.Sheets(1).Range("D1").Value = "calMean"

' columns count is 1 to 20
' column index for variables is 5 to 24
Cnum = 1
Cind = 5

Do While Cnum <= 20
   'write column header based on the column index (Cind) and number of variables (Cnum)
   basebook.Sheets(1).Cells(1, Cind).Value = fldName & Cnum
   ' keep track of column index and number of columns with variables
   Cnum = Cnum + 1
   Cind = Cind + 1
Loop

CleanUp:
Application.ScreenUpdating = True
Application.DisplayAlerts = True

End Sub


'### END OF SCRIPT ###
```

Section III. Method of calculating anomalies for the variables of interest

15. Map the variables of interest and calculate anomalies in ESRI ArcGIS ™(see GIS methods below):

Note: The faster method has been proven to be accurate to within 0.01°C with a reconstruction of MTWA64 for the last 2,000 years. Hence, the shapefile with all reconstruction data for the last 2,000 can be spatially joined (step 15B) without splitting the shapefiles (step 15A with details in steps 30-34) and recombining the files (steps 16A to 16D). There will be 133 records that do not join. This means 133 records (out of 10,887) over the last 2,000 years do not have a modern record match when using a 2 search radius.

A- Create shapefiles for each time interval for each reconstructed climate variable for each pollen sum (35, 64, 134, 135)

B- Spatially join shapefiles (Note this is different from a field to field join which can be done in MS Access™) from A with modern climate data (Whitmore) in order to calculate anomalies (0 - modern, 100 - modern, etc.)

NOTE: Only MTWA, MTCO, MIPT and ANNP completed up to this point. GDD5, TJAN and TJUL completed to step 15A.

ERRORS TO NOTE: MIPT135 has only 0-1900 cal yr BP. Also, MTWA64 only has the top analog. This will not affect the results as we are only using the top analog of MTWA64.

C- Create gridded maps for each time interval for each reconstructed climate variable for each pollen sum (using interpolation)

16. Calculate T-S of anomalies created in 15c (run calcMTWAmeanByCalYr then plot data)

```
Sub calcMTWAmeansByCalYrBP()

Dim MTWAsum As Single
Dim MTWAmean As Single
Dim startRow, endRow, curRow, curRowWS2, vcount, calYr, nextCalYr As Integer
Dim calYrCell, nextCalYrCell As String

' Make sure reconstruction file is interpolated and is sorted by CalYrBP first!
' written by Matthew Ladd
' last modified: March 28, 2011

' initialize the variables
startRow = 2
endRow = 9958 'change this depending on # of rows
curRow = 2 'keep track of rows on 1st worksheet
curRowWS2 = 2 'keep track of rows on 2nd worksheet

MTWAsum = 0
MTWAmean = 0

vcount = 1 'keep track of the number of MTWA values for a given site

' put a header for the output on 2nd worksheet
Worksheets(2).Range("A1").Value = "calMean"
Worksheets(2).Range("B1").Value = "meanMTWA"

While curRow <= endRow

' get the calYr in the current row
calYrCell = "D" & curRow
calYr = Worksheets(1).Range(calYrCell).Value

' get the calYr of the next row
nextCalYrCell = "D" & curRow + 1
nextCalYr = Worksheets(1).Range(nextCalYrCell).Value

' ensure we have the correct cells for the MTWA and MTWA mean
```

```
MeanMTWAcell = "B" & curRowWS2 ' values written in 2nd col on 2nd worksheet
MTWAcell = "E" & curRow ' change if not using first analog

' keep a running sum of the MTWA values
MTWAsum = MTWAsum + Worksheets(1).Range(MTWAcell).Value

' if new site on next row calculate the mean MTWA and output to 2nd worksheet
If nextCalYr <> calYr Or nextCalYr = Null Then
    MTWAmean = MTWAsum / vcount
    Worksheets(2).Range(MeanMTWAcell).NumberFormat = "0.0"
    Worksheets(2).Range(MeanMTWAcell).Value = MTWAmean
    Worksheets(2).Range("A" & curRowWS2).Value = calYr
    ' reset this to zero since next row is new calYrBP value
    MTWAsum = 0
    MTWAmean = 0
    curRow = curRow + 1
    curRowWS2 = curRowWS2 + 1
    vcount = 1
Else
    ' if same value of CalYrBP in next row just go to that row
    curRow = curRow + 1
    vcount = vcount + 1

End If

Wend

End Sub


' from http://social.msdn.microsoft.com/Forums/en-US/exceldev/thread/6b416858-fad2-4f51-8688-b3f063debfcf

' Note: need to combine anonamly shapefile tables (DBFs) using macro then run macro to get
' average of anomalies for all sites at each time interval

Sub OpenDBF()

Set Masterbk = ThisWorkbook
Set MasterSht = Masterbk.Sheets(1)

'get first empty column in master worksheet
MasterLastCol = MasterSht.Cells(1, Columns.Count).End(xlToLeft).Column
If MasterLastCol <> 1 Then
 MasterNewCol = MasterLastCol + 1
Else
 MasterNewCol = 1
End If

Folder = "R:\LPC\LPC PAPERS\NAM_2k\mapping\MTWA\anomalies\_to_combine\test\"

 'make sure backslash is in string

FName = Dir(Folder & "*.dbf")

Do While FName <> ""
 Set bk = Workbooks.Open(Filename:=Folder & FName)

 Set NewSht = bk.Sheets(1)


 With NewSht

  'use column A to determine last row of database
  DatabaseLastRow = .Range("A" & Rows.Count).End(xlUp).Row

  MasterLastRow = MasterSht.Range("A" & Rows.Count).End(xlUp).Row
  MasterNewRow = MasterLastRow + 1
  LastCol = .Cells(1, Columns.Count).End(xlToLeft).Column


  For ColCount = 1 To LastCol
   Header = .Cells(1, ColCount)
   Set DataBaseColumn = _
   .Range(.Cells(2, ColCount), .Cells(DatabaseLastRow, ColCount))

   With MasterSht
```

```
  'check if header exists
  Set c = .Rows(1).Find(what:=Header, _
   LookIn:=xlValues, lookat:=xlWhole)
  If c Is Nothing Then
   .Cells(1, MasterNewCol) = Header
   DataBaseColumn.Copy _
    Destination:=.Cells(MasterNewRow, MasterNewCol)
   MasterNewCol = MasterNewCol + 1
  Else
   DataBaseColumn.Copy _
    Destination:=.Cells(MasterNewRow, c.Column)
  End If

  End With
 Next ColCount
End With

bk.Close savechanges:=True

FName = Dir()
Loop
MsgBox ("DONE!")
End Sub
```

Steps to calc average of site anomalies for each time interval:

A Place all anomaly DBF files in a separate folder (if necessary).

B Open macro "combine_multDBF" and modify path to files in A first.

C Run macro and verify the output.

D Re-save as XLS format.

E Copy macro code (calcMTWAmeanByCalYrBP) into the XLS file from D to calculate average of site anomalies for each time interval. Make sure that the last blank row after the last record has a -999 value in the calMean field. Otherwise the anomaly for 0 cal yr BP will not be calculated.

F Macro will need to be modified to reference calYr and anomaly columns properly.

G Sort the file (descending so that time series will read left to right) by calYr and expand the selection (useful to autofit the width of the columns to check data [Format - Columns - Autofit Selection])

H Now run the macro, output is on second worksheet (paste into notepad, row with -999 can be discarded).

I Re-save the output then re-run for second file (paste into a text editor such as Notepad).

J Finally copy output from notepad back into MS Excel™(blank workbook) & plot the time series (or use another program such as R for plotting the time series).

17. Repeat steps 15 for ANNP, GDD5, MIPT, MTCO, TJAN and TJUL for all

pollen sums (35, 64, 134, 135).

Section IV. Creating regional time-series using anomaly data and adapted Fedorova ecoregions (NOTE: other ecoregions can be used)

1. First create a csv file with the AVHRR (Fedorova) data from Whitmore (also, include siteID ID1, lat and lon). X-Y map this in ESRI ArcGIS ™and export as a shapefile.

2. Start ESRI ArcGIS™and add this shapefile with the Whitmore sites categorized by Fedorova region (see *). Add the shapefile for variable of interest (e.g. MTWA64_0-2000.shp).

3. Open the spatial join tool and join the Fedorova data to the variable of interest. Join features is Fedorova, target is variable of interest, keep all features and select option "CLOSEST" or "INTERSECT".

4. After the join is complete, check the output and close ESRI ArcGIS™.

5. Now you open the Excel file and sort the data by Fedorova column (sorting method does not matter).

6. Copy the data for each Fedorova region to a new worksheet (leave one blank worksheet in between).

7. Now you run the macro from step 14E on each region. Note that the script reads from worksheet 1 and outputs to worksheet 2 so the data and blank worksheet for the selected region will need to be moved to the far left in front of the other worksheets).

8. Copy all T-S data for each region and plot each of them on separate graphs.

   NOTE: A GIF image of the Fedorova raster (see Fedorova *et al.*, 1994) was imported into ESRI ArcGIS™and then georeferenced to North America. The Fedorova ecoregions are aggregated in Whitmore such that there are only 10 regions plus no data. Arctic and Mediterranean regions were not included due to lack of data.

Forest-Tundra
Conifer/Hardwood
Boreal Forest

Deciduous Forest
Prairies
Arctic
Coastal
Mountain vegetation
Desert
Mediterranean
No data (blank and -99 values, not used)

Section V. X-Y mapping the shapefiles, splitting each shapefile by time interval, and mapping at specific time intervals

The first step is to bring in the CSV files and create point shapefiles for selected variable at specific cal yr dates. For x-y mapping to work the csv file should not be open in another program (e.g. MS Excel™) and the column header (field) names should not have any spaces in them.

9. Run macro formatLatLonColumnsAndColHeadersMacro to fix this (e.g. MTCO 1 becomes MTCO1).

```
' Author: Matthew Ladd
' Notes: adapted from code at: http://www.rondebruin.nl/copy3.htm
' Last modified: April 5, 2011

Sub formatLatLonColumns()
Dim MyPath, fldName As String
Dim FilesInPath As String
Dim MyFiles() As String
Dim SourceRcount As Long
Dim Fnum, Cnum, Cind, Clnumber As Long
Dim mybook As Workbook
Dim basebook As Workbook
Dim myrange As Range
Dim Msg As String
Dim fldr As FileDialog
Dim sItem, strCell As String


Set fldr = Application.FileDialog(msoFileDialogFolderPicker)
'Default path\folder where the files are on your machine
' NOTE: THIS CAN BE CHANGED IF NECCESSARY
strPath = "R:\LPC\LPC PAPERS\NAM_2k\scripts\interpolation\_FINAL\"
With fldr
   .Title = "Select a Folder..."
   .AllowMultiSelect = False
   .InitialFileName = strPath
   If .Show <> -1 Then GoTo NextCode
   MyPath = .SelectedItems(1)
End With
```

```
NextCode:
GetFolder = sItem
Set fldr = Nothing
'Add a slash at the end if the user forget it
If Right(MyPath, 1) <> "\" Then
   MyPath = MyPath & "\"
End If
'If there are no Excel files in the folder exit the sub
FilesInPath = Dir(MyPath & "*.csv")
If FilesInPath = "" Then
   MsgBox "No files found"
   Exit Sub
End If
On Error GoTo CleanUp
Application.ScreenUpdating = False
Application.DisplayAlerts = False 'disable all alert prompts
Set basebook = ThisWorkbook
'Fill the array(myFiles)with the list of Excel files in the folder
Fnum = 0
Do While FilesInPath <> ""
   Fnum = Fnum + 1
   ReDim Preserve MyFiles(1 To Fnum)
   MyFiles(Fnum) = FilesInPath
   FilesInPath = Dir()
Loop
'Loop through all files in the array(myFiles)
For Fnum = LBound(MyFiles) To UBound(MyFiles)

   Set mybook = Workbooks.Open(MyPath & MyFiles(Fnum))
   ' determine last row number of imported sheet
   lastRow = mybook.ActiveSheet.UsedRange.SpecialCells(xlCellTypeLastCell).Row

   'ensure LAT values have one decimal place
   mybook.Worksheets(1).Range("B2:B" & lastRow).NumberFormat = "0.00"


   'ensure LON values have one decimal place
   mybook.Worksheets(1).Range("C2:C" & lastRow).NumberFormat = "0.00"


   ' now remove spaces in column header names (e.g. ANNP 1 becomes ANNP1, etc.)
   For Clnumber = 5 To 24
     strCell = Cells(1, Clnumber).Value
     strCell = Replace(strCell, " ", "")
     Cells(1, Clnumber).Value = strCell
   Next Clnumber


   On Error Resume Next


   On Error GoTo 0



   mybook.Close savechanges:=True

Next Fnum

CleanUp:
Application.ScreenUpdating = True
Application.DisplayAlerts = True
End Sub
'### END OF SCRIPT ###
```

10. X-Y map each CSV file (e.g. MTWA for pollen sum 35, etc.) using 'Make XY Event Layer' tool (right click tool for batch mode).

```
1. X-Y table is the CSV file to map (can right click & browse for multiple CSV files located in R:\LPC\LPC PAPERS\NAM_2k\scripts\
      interpolation\_FINAL)
2. X field is LON
3. Y field is LAT
4. Layer Name or Table View is output name for events layer
```

5. For Spatial Reference use GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]];IsHighPrecision
6. For steps 4B to 4E, right click the field and click fill (Layer Name in step 4D for each X-Y table will need to be modified)

NOTE: it is important to check that a coordinate system for this shapefile (WGS 84) is set before proceeding (in step 10 above).

11. After X-Y mapping check that the lat/lon columns have at least 1 (preferably 2) decimal place. If not, the numberformat of the lat/lon columns in the CSV file(s) will have to be fixed manually as it is "General" by default. Make sure to set both fields to Number with 2 decimal places.

(NOTE that decimal places of lat/lon columns should be checked at every step until the shapefile is created.)

12. Now each of the X-Y event layers will need to be exported as shapefiles. This can only be done manually (one at a time) by right clicking each X-Y event layer and selecting [Data - Export Data].

13. Now extract by calendar year BP mean using the geoprocessing toolbox Split Layer by Attributes at http://arcscripts.esri.com/details.asp?dbid=14127

NOTE: there is a newer version for ESRI ArcGIS™(v10.0+)

14. Unzip package and move the files to "C:/Program Files/ArcGIS/ArcToolboxes/Toolboxes/" then add this new toolbox in ESRI ArcGIS™and then run the script associated with it.

15. Before running the tool, each shapefile (e.g. MTWA_PS-64) must have a new calMean field (e.g. CaYrBPmean) of type long integer. Use the "Add Field" tool in batch mode to do this. Otherwise output shapefile names will be 0000.0 or 1000.0.

A. Input table should be the shapefile as in step 7.
B. Field Name should be CaYrBPmean.
C. Field Type should be LONG.
D. Leave other optional fields as is.

16. This new field CaYrBpmean must have the same values as calMean so use the tool "Calculate Field" in batch mode to copy values from calMean to CaYrBP-mean.

    A. Input table should be the shapefile as in step 7.
    B. Field Name should be CaYrBPmean.
    C. For Expression open up the calculator and select calMean so that CaYrBP = [calMean].
    D. Leave other optional fields as is.

17. Now run the "Split Layer by Attributes" tool in batch mode.

    A. Feature Layer is each shapefile (e.g. MTWA_PS-64, etc., use browse to select multiple shapefiles)
    B. Field to query is CaYrBPmean
    C. Output filename is almost the same as step 9A (e.g. MTWA64ps no longer than 9 characters, 4 will be added on by the script: 0100, 0200, etc.)
    D. output folder should be a separate subdirectory

18. Now do the join of MTWA0, 100, etc. with modern climate data to enable calculation of the anomalies (e.g. 0 - mod, 100 - mod, etc.). Under ArcToolBox, select [Analysis Tools - Overlay - spatial join] (right click on the tool to run in batch mode)

19. Choose the whitmore_mod_climate layer for the join features, each shapefile (e.g. MTWA_PS35_100, these will need to added to map document first) for the target features, select an appropriate output path (output feature class), join operation is JOIN_ONE_TO_ONE, keep all target features is FALSE (unchecked), field map of join features is blank, match option is CLOSEST, search radius is 2 decimal degrees, and distance field name is DIST. Some of these options can be changed if necessary.

20. Create a field in each shapefile for the anomaly (should be double or float) using the "Add Field" tool in batch mode. (e.g. MTWA1_an)

A. Input table should be the shapefile as in step 11.

B. Field Name will depend on climate variable (and analog) being reconstructed (e.g. MTWA1_an, MTCO1_an)

C. Field Type should be the same as original field (with the same number of significant figures).

D. Leave other optional fields as is.

21. Now calculate the anomalies using Calculate Field tool in batch mode. For example, the anomaly field value would be equal to MTWA1 (first analog) - mtwa (modern value).

A. Input table should be the shapefile as in step 11.

B. Field Name should be the anomaly field (e.g. MTWA1_an).

C. For Expression open up the calculator and select (in this case) field names such that MTWA1_an = [MTWA1] [mtwa].

D. Leave other optional fields as is.

1. Now interpolate the points to create gridded maps for MTWA at 0 BP, 100 BP, 200 BP, 300 BP, etc.
2. Start the Geostatistical Wizard (add this toolbar first, if not visible).
3. Select local polynomial and MTWA1 is the field to be interpolated.
4. Follow the steps and select 70
5. Once complete then convert interpolation to raster using [data - export] to raster.
6. This raster should be clipped to the North American land area.
7. Under spatial analyst options select the mask as world.shp, working directory is current directory, then for extent use same as world.shp, for cell size use the same as mtwa100.
8. Now start the raster calculator under the spatial analyst menu and choose mtwa100 and calculate.
9. Need to make clipped raster permanent ([data-make permanent]).
10. Now open up these rasters in the nam_2k_proj mxd and classify them (right click [raster - properties] then choose the symbology tab.

11. Select [classified] after computing the histogram and click the [classify] button. It depends on the climate variable but for MTWA, select [defined interval] with a size of 5, this will be 5c intervals.

12. Choose an appropriate colour ramp.

13. Switch to layout view to add a legend.

14. Export map as jpeg (or another format).

15. Repeat steps 40-52 for each time interval (0 BP, 100 BP, etc)

16. Repeat steps 26-53 for each climate variable for each pollen sum (MTWA 64, MTWA 134, MTWA 135, MTCO 35, etc.)

17. Ensure that mxd filed is saved for older versions of ESRI ArcGIS™(if necessary)

Section VI. ANOMALY MAPS FOR MWP, LIA and MWP - LIA

18. Make sure MWP folder only has DBF files for 650-1050 AD (900 to 1300 cal Yr BP) and LIA folder only has DBF files for 1350-1750 AD (200 to 600 cal Yr BP).

    NOTE: time intervals may need to be modified depending on the time period of interest

19. In Excel open a blank workbook and paste macro "combineMultDBF" into the code window. Run macro to combine DBF files (change path in script first).

20. Sort this file by modern site ID (does not matter if ascending or descending, make sure to expand selection).

21. Open the code window and replace code with macro calcMTWAavgAnomaly-BySite to average all anomalies at each site for the time intervals in step 31. Ensure the correct columns are being referenced for modern site ID, modern lat/lon, and anomaly!

```
Sub calcMTWAavgAnomolybySite()

Dim MTWAsum, MTWAmean, lonVal, latVal As Single
Dim startRow, endRow, curRow, curRowWS2, vcount, calYr, nextCalYr As Integer
Dim calYrCell, nextCalYrCell, latCell, lonCell As String

' Make sure reconstruction file is interpolated and is sorted by siteID first!
' written by Matthew Ladd
' last modified: April 8, 2011

' initialize the variables
```

```
startRow = 2
endRow = 17071 'change this depending on # of rows
curRow = 2 'keep track of rows on 1st worksheet
curRowWS2 = 2 'keep track of rows on 2nd worksheet

MTWAsum = 0
MTWAmean = 0

vcount = 1 'keep track of the number of MTWA values for a given site

' put a header for the output on 2nd worksheet
Worksheets(2).Range("A1").Value = "siteID"
Worksheets(2).Range("B1").Value = "LAT"
Worksheets(2).Range("C1").Value = "LON"
Worksheets(2).Range("D1").Value = "meanMTWA"

While curRow <= endRow

' get the calYr in the current row
calYrCell = "BZ" & curRow
calYr = Worksheets(1).Range(calYrCell).Value

' get the lat/lon values
latCell = "CA" & curRow
latVal = Worksheets(1).Range(latCell).Value

lonCell = "CB" & curRow
lonVal = Worksheets(1).Range(lonCell).Value

' get the calYr of the next row
nextCalYrCell = "BZ" & curRow + 1
nextCalYr = Worksheets(1).Range(nextCalYrCell).Value

' ensure we have the correct cells for the MTWA and MTWA mean
MeanMTWAcell = "D" & curRowWS2 ' values written in 2nd col on 2nd worksheet
MTWAcell = "CX" & curRow ' change if not using first analog

' keep a running sum of the MTWA values
MTWAsum = MTWAsum + Worksheets(1).Range(MTWAcell).Value

' if new site on next row calculate the mean MTWA and output to 2nd worksheet
If nextCalYr <> calYr Or nextCalYr = Null Then
    MTWAmean = MTWAsum / vcount
    'Worksheets(2).Range(MeanMTWAcell).NumberFormat = "0.00"
    Worksheets(2).Range(MeanMTWAcell).Value = MTWAmean
    Worksheets(2).Range("A" & curRowWS2).Value = calYr

    ' output lat/lon values here
    Worksheets(2).Range("B" & curRowWS2).Value = latVal
    Worksheets(2).Range("C" & curRowWS2).Value = lonVal

    ' reset this to zero since next row is new calYrBP value
    MTWAsum = 0
    MTWAmean = 0
    curRow = curRow + 1
    curRowWS2 = curRowWS2 + 1
    vcount = 1
Else
    ' if same value of CalYrBP in next row just go to that row
    curRow = curRow + 1
    vcount = vcount + 1

End If

Wend

MsgBox ("DONE!")

End Sub
```

22. Run this script and check the output on the second worksheet.

23. Copy this data to a new worksheet and save as CSV.

24. X-Y map this in ESRI ArcGIS™, export as a shapefile, and produce final map (see interpolation steps below).

    A. use geostatistical wizard, ordinary prediction kriging spherical, use mean, no anisotropy or smoothing, specify an ellipse major semi-axis of 2 x 2 degrees. No transformation is needed.

    B. Export to raster (2 degree cell size), classify (make sure to have zero), 5 categories with appropriate intervals. Also, take the opposite values of anomalies when mapping (i.e. multiply values by -1).

1. Repeat previous steps for LIA period.

2. As there are a different number of points in the MWP versus LIA file, a simple difference calculation in MS Excel™will not work. It is possible to grid the data using a program such as MatLab™but mapping the new grid in ESRI ArcGIS™can be problematic. The solution is to do a spatial join of the MWP and LIA anomaly data in ESRI ArcGIS™.

3. Add the MWP and LIA anomaly shapefiles created (for variable of interest) to an empty map document.

4. Open the spatial join tool and join the LIA anomaly data to the MWP anomaly data. Join features is LIA anomaly for variable of interest, target is MWP anomaly for variable of interest, keep all features and select option "CLOSEST" or "INTERSECT".

5. Check the output shapefile and add a new field MWP-LIA (type double). Using the field calculator the value of this new field will be the MWP anomaly LIA anomaly.

6. Now, map this difference using the same interpolation steps as before.

7. Create final versions of these anomaly maps in layout view (add scale bar, legend, graticule, and projection if necessary). Regional time series plotted in R can be exported as metafiles to Corel Draw™where they can be imported and overlaid with the exported basemap from a map document.

8. The last step is to create a map of the fossil and modern sites. Use the NAM2k basemap and add the necessary data. The final maps are produced in layout view and then imported into Corel Draw™.

D.2 Calculation of site resolution in a reconstruction

Table 14: Number of sites through time for reconstruction of MTWA (using 35 pollen sum)

CalBP	SiteCount
0	457
100	557
200	579
300	583
400	584
500	583
600	584
700	581
800	582
900	578
1000	573
1100	563
1200	563
1300	556
1400	541
1500	526
1600	499
1700	473
1800	434
1900	338
2000	163

Method to verify the resolution:

- Only 82 sites have a resolution $\leq$ 50 yrs
- Only 156 sites have a resolution $\leq$ 100 yrs
- Run macro on raw data (last 2k)
- Save first 2 worksheets as csv (raw data & macro output)
- Import into MS Access™

- Query for resolutions ≤ 100 and ≥ 50 yrs
- Join raw data to each of these two queries
- Export two new queries as Excel 97-2003™
- Open these files and save as csv
- Then can run linear interp script: linear_interp_test_v3.r
- Spatial join with whitmore modern climate data
- Then can create T-S using same methodology as before (MS Excel™macro)

```
Sub calcResolution()

' written by Matthew Ladd (New version)
' last modified: February 10, 2012

Dim sampCount, siteID, ageMin, ageMax, calYr, nextCalYr, totalSampCount, vcount As Long
Dim lat, lon, avgSampRes, sampRes, allSampRes As Single
Dim yrCell, nextYrCell As String

' initialize the variables
startRow = 2
endRow = 30862 'change this depending on # of rows or use -999 for last cal yr value
curRow = 2
curRow2 = 2
currRow = 2
sampCount = 0 'keep track of the number of samples
vcount = 0

' need to ask user which columns contain appropriate data
calYrFieldCol = Application.InputBox _
    (Prompt:="Enter the letter of the Cal Yr BP column (e.g. K):", _
    Title:="SPECIFY COLUMN", Type:=2)
On Error GoTo 0

calyroldval = Application.InputBox _
    (Prompt:="Enter the oldest cal Yr BP interval (must be a multiple of 1000):", _
    Title:="OLDEST CAL YR BP", Type:=1)
On Error GoTo 0

lastColLetter = Application.InputBox _
    (Prompt:="Enter the last column of the data set (e.g. BX):", _
    Title:="LAST COLUMN", Type:=2)
On Error GoTo 0

siteIDCol = Application.InputBox _
    (Prompt:="Enter the column with siteID values (e.g. B):", _
    Title:="SITEID COLUMN", Type:=2)
On Error GoTo 0

' ensure that the proper cal yr BP values are included
' e.g. user enters in 2033, therefore need to round to the nearest 1000
' but the last 2ka should include data up to 2099 cal yr BP
calYrStop = Int(calyroldval / 1000) * 1000 + 99
calyroldval = calYrStop - 99

Sheets.Add.Name = "OUTPUT" ' this is worksheet #1

' sort cal yr BP field ascending
Worksheets(2).Range(calYrFieldCol & "2").Sort Key1:=Worksheets(2).Columns(calYrFieldCol), Order1:=xlAscending, Header:=xlYes

' find the last row of the data set based on cal Yr entered
endRow = Worksheets(2).UsedRange.Rows.Count

'need to get value from first row
```

```
calYrCell = calYrFieldCol & currRow
calYr = Worksheets(2).Range(calYrCell).Value

' get the calYr in the current row
While currRow <= endRow And calYr < calYrStop
    calYrCell = calYrFieldCol & currRow
    calYr = Worksheets(2).Range(calYrCell).Value
    currRow = currRow + 1
Wend

Worksheets(2).Range("A1:" & lastColLetter & currRow).Copy _
Destination:=Worksheets(1).Range("A1")

' now sort this by siteID ascending
Worksheets(1).Range(siteIDCol & "2").Sort Key1:=Worksheets(1).Columns(siteIDCol), Order1:=xlAscending, Header:=xlYes

'add -999 flag to last row + 1 of cal yr BP field
lastrow = Worksheets(1).UsedRange.Rows.Count
Worksheets(1).Range(calYrFieldCol & lastrow + 1).Value = -999

' final output will be on this worksheet 'this is now worksheet #1,
' OUTPUT is worksheet #2, original data is on worksheet #3
Sheets.Add.Name = "FINAL"

' write headers on 2nd worksheet
Worksheets(1).Range("A1").Value = "SiteID"
Worksheets(1).Range("B1").Value = "LatDD"
Worksheets(1).Range("C1").Value = "LonDD"
Worksheets(1).Range("D1").Value = "SampleRes"
Worksheets(1).Range("E1").Value = "TotalDates"
Worksheets(1).Range("F1").Value = "FirstDate"
Worksheets(1).Range("G1").Value = "LastDate"


latCellCol = Application.InputBox _
    (Prompt:="Enter the column with latitude values (e.g. D):", _
    Title:="LATITUDE COLUMN", Type:=2)
On Error GoTo 0

lonCellCol = Application.InputBox _
    (Prompt:="Enter the column with longitude values (e.g. E):", _
    Title:="LONGITUDE COLUMN", Type:=2)
On Error GoTo 0


While curRow <= endRow

    ' get the cal Yr value in the current row (check for NAs)
    yrCell = calYrFieldCol & curRow
    If IsNumeric(Worksheets(2).Range(yrCell).Value) Then
        calYr = Worksheets(2).Range(yrCell).Value
        sampCount = sampCount + 1
        ' first sample in group is ageMin
        If sampCount = 1 Then
           ageMin = calYr
        End If
    Else
      yrCell = calYrFieldCol & curRow + 1
    End If


    ' get the cal Yr value of the next row
    nextYrCell = calYrFieldCol & curRow + 1
    If IsNumeric(Worksheets(2).Range(nextYrCell).Value) Then
      nextCalYr = Worksheets(2).Range(nextYrCell).Value
    Else
      nextYrCell = calYrFieldCol & curRow + 1
    End If


    If nextCalYr < calYr Then

       vcount = vcount + 1
       ageMax = calYr ' last sample in group is ageMax

       siteID = Worksheets(2).Range(siteIDCol & curRow).Value
```

```
        lat = Worksheets(2).Range(latCellCol & curRow).Value
        lon = Worksheets(2).Range(lonCellCol & curRow).Value


        ' if new site on next row calculate the resolution (ageMax - ageMin) / sampCount
        If (ageMax <> ageMin) Then
            sampRes = (ageMax - ageMin) / sampCount
        Else
            sampRes = ageMax
        End If

        ' write the values to 2nd worksheet
        Worksheets(1).Range("A" & curRow2).Value = siteID
        Worksheets(1).Range("B" & curRow2).NumberFormat = "0.0"
        Worksheets(1).Range("B" & curRow2).Value = lat
        Worksheets(1).Range("C" & curRow2).NumberFormat = "0.0"
        Worksheets(1).Range("C" & curRow2).Value = lon
        Worksheets(1).Range("D" & curRow2).NumberFormat = "0"
        Worksheets(1).Range("D" & curRow2).Value = sampRes
        Worksheets(1).Range("E" & curRow2).NumberFormat = "0"
        Worksheets(1).Range("E" & curRow2).Value = sampCount
        Worksheets(1).Range("F" & curRow2).NumberFormat = "0"
        Worksheets(1).Range("F" & curRow2).Value = ageMin
        Worksheets(1).Range("G" & curRow2).NumberFormat = "0"
        Worksheets(1).Range("G" & curRow2).Value = ageMax

        ' keep track of all samples resolution
        allSampRes = allSampRes + sampRes

        ' keep track of total samples
        totalSampCount = totalSampCount + sampCount

        ' counter for 2nd worksheet
        curRow2 = curRow2 + 1

        ' reset this to zero since next row is new group of samples
        sampCount = 0

        If (nextCalYr = -999) Then
            avgSampRes = Round(allSampRes / vcount)
            MsgBox ("PROCESSING FINISHED! There are " & totalSampCount & " samples at all sites. The average sample resolution for all
                  sites is " & avgSampRes & " years for the last " & calyroldval & " years.")
            ' save this worksheet as a .csv file
            ActiveWorkbook.SaveAs Filename:="sampleResolution.csv", FileFormat:=xlCSV, CreateBackup:=False
            Exit Sub
        End If

    End If

    curRow = curRow + 1

Wend

End Sub
```

Table 15: Resolution in years for selected pollen sites

SiteID	LatDD	LonDD	SampleRes	TotalDates	FirstDate	LastDate
7	47	-80.1	154	13	-26	1976
8	34	-120	134	15	-40	1972
18	29.6	-97.5	179	11	120	2089
25	52.5	-57	30	1	30	30
29	44.8	-94.5	33	16	-12	513
196	46.2	-89.7	27	75	-10	2003
197	46.3	-89.9	404	3	254	1466
203	36	-85.5	155	12	0	1858
205	46.7	-85	13	13	-33	141

D.3 Calibration of radiocarbon dates using IntCal '09

1. Download and install Calib Rev 6.0.1™.
2. Setup template file with radiocarbon dates (age field) from MTWA64 file (output file from MATTOOLS). Make sure to select CalYR under [options - calibration]. Use SD=1 and NH terrestrial sample for the template file. Other fields are left as is.
3. Run the calibration in batches of 5,000 radiocarbon dates. Using more dates will cause Calib to crash.
4. Combine the files created in step 3 into one large file.
5. Extract columns Lab code or Sample code, 14C age and median probability (this can be renamed to IntCal09).
6. Filter all 3 columns by unique values (copy to new blank columns) and then save as csv.
7. Import into MS Access™along with MTWA64 as tables. Ensure that lat/lon fields are set as number with one decimal place.
8. Create a new query with a join on these two tables (join lab code to sample ID).
9. Add all fields from each table then export as an Excel 97-2003™file.
10. Open the file in MS Excel™, and sort by Cal Yr age ($\leq$ 4100 for a 4 ka reconstruction). And delete any rows where C14age $\geq$ 4100.
11. Keep only columns SiteID, Lat, Lon, Cal Yr, and reconstructed variable of interest (e.g. MTWA1).
12. Now the data file is ready for the linear interpolation script (Appendix C.4.).

D.4 Excel Macros for linear interpolation and averaging anomalies

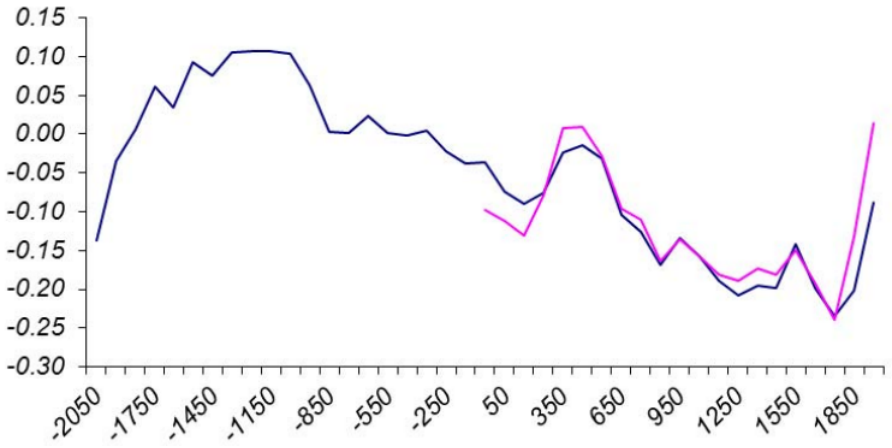


Figure 30: Comparison of IntCal '09 (navy blue) versus IntCal '04 (magenta) time-series. Note that the Viau *et al.* (2006) reconstruction used IntCal '98.

```
Sub reconsPrePostProcessing()

' written by Matthew Ladd (LPC)
' last modified: October 31, 2011
' more information about interpolation using Excel (the FORECAST formula)
' at http://www.blueleafsoftware.com/Products/Dagra/LinearInterpolationExcel.php

' For the first option (2nd worksheet with linear interpolation formulas already setup)
' ***Make sure reconstruction file has 5 columns: (A)siteID, (B)Lat, (C)Lon, (D)CalYrBP,
' and (E) reconstructed climate variable to be interpolated (e.g. MTWA1, MIPT1, etc)
' Also, first row should be a 'dummy' record (e.g. all zeroes)
' And last siteID should have data that span at least two 100 year intervals

Dim startRow, endRow, curRow, curRowWS2, siteID, nextSiteId, calYr, nextCalYr As Long
Dim calYrInterp, interpStart, interpEnd, interpLast, vcount, CalYrInt As Long
Dim calYrStopVal, calYrStop, interpInterval, decPlaces, currRow As Long
Dim siteIdCell, nextSiteIdCell As String

' initialize the variables
startRow = 2
endRow = 65536 'change this depending on # of rows **OR** use -999 for last siteID (see below)
curRow = 2 'keep track of rows on 1st worksheet
curRowWS2 = 2 'keep track of rows on 2nd worksheet (or actually 2nd group of output rows on first worksheet)
vcount = 0 ' keep track of number of rows for each site
CalYrInt = 0 ' start at zero bp

' first row has column headers
currRow = 2

'Application.DisplayAlerts = False

' set INTERPOLATION INTERVAL
Dim interpQuestion As VbMsgBoxResult

interpQuestion = MsgBox("Do you wish to interpolate at 100 year intervals? If no, interpolation interval will be 50 years.", vbQuestion
       + vbYesNo)

If interpQuestion = vbYes Then
    interpInterval = 100
ElseIf interpQuestion = vbNo Then
    interpInterval = 50
End If


' set INTERPOLATION RANGE
On Error Resume Next

calYrStopVal = Application.InputBox _
```

```
    (Prompt:="Enter the oldest cal yr BP value to interpolate (acceptable values are between 1000 and 30000, value must be a multiple
          of 1000, youngest will always be 0):", _
    Title:="INTERPOLATION RANGE", Type:=1)
On Error GoTo 0


If calYrStopVal = Null Or calYrStopVal > 30000 Then
    Exit Sub
Else
    'must be 1000 cal Yr BP intervals
    calYrStop = Int(calYrStopVal / 1000) * 1000
End If

' set INTERPOLATION PRECISION

On Error Resume Next

decPlaces = Application.InputBox _
    (Prompt:="Enter the number of decimal places to interpolate to (1-5), Note that N/A values will always be excluded:", _
    Title:="INTERPOLATION PRECISION", Type:=1)
On Error GoTo 0

If decPlaces = Null Or decPlaces > 5 Or decPlaces < 1 Then
    Exit Sub
End If

'Application.DisplayAlerts = True


' sort cal yr BP field ascending
Worksheets(1).Range("G2").Sort Key1:=Worksheets(1).Columns("G"), Order1:=xlAscending, Header:=xlYes

' find the last row of the data set based on cal Yr entered
endRow = Worksheets(1).UsedRange.Rows.Count

'need to get value from first row
calYrCell = "G" & currRow
calYr = Worksheets(1).Range(calYrCell).Value

calYrStop = calYrStop + 99
' get the calYr in the current row
While currRow <= endRow And calYr < calYrStop
    calYrCell = "G" & currRow
    calYr = Worksheets(1).Range(calYrCell).Value
    currRow = currRow + 1
Wend
calYrStop = calYrStop - 99

'adjust row count to account for two extra rows
currRow = currRow - 2

' copy to 2nd worksheet where either (1) linear interpolation formulas are already setup and interpolated values
' will be automatically calculated or (2) create new worksheet and call linear interpolation function to do the
' interpolation - ask user which method to use

Dim Response As VbMsgBoxResult
Response = MsgBox("Is 2nd worksheet already set up for linear interpolation? Format is Col G has siteID, col H has lat, col I has lon,
      col J has calYrBP interval and col K has climate variable of interest with formulas, also 2nd row must have all zeroes - dummy
      row. If not, a new worksheet will be created, the data copied, and linear interpolation will be run. This could take a long time
      .", vbQuestion + vbYesNo)
If Response = vbYes Then

      ' *** CASE #1 ***
      ' select 5 columns (A,B,C,G,AV) for time period of interest (e.g. last 2k has 8280 records)
      ' reconstructed value (1st analog is column 48, 2nd analog is column 49)
      Worksheets(1).Range("A2:C" & currRow & ",G2:G" & currRow & ",AV2:AV" & currRow).Select

      Worksheets(1).Range("A2:C" & currRow & ",G2:G" & currRow & ",AV2:AV" & currRow).Copy _
      Destination:=Worksheets(2).Range("M3")

      Worksheets(2).Range("M3").Sort Key1:=Worksheets(2).Columns("M"), Order1:=xlAscending, Header:=xlNo

      Worksheets(2).Range("M3:Q" & currRow + 1).Copy _
      Destination:=Worksheets(2).Range("A4")

      Worksheets(2).Range("M2:Q" & currRow + 1).Delete
```

```
      Worksheets(2).Name = "Output"

      MsgBox ("Process complete!")

Else 'response = vbNo

      ' *** CASE #2 ***
      Sheets.Add.Name = "Output"

      currRow = currRow + 1

      ' copy the correct data
      Worksheets(2).Range("A1:C" & currRow & ",G1:G" & currRow & ",AV1:AV" & currRow).Copy _
      Destination:=Worksheets("Output").Range("A1")

      Worksheets("Output").Range("A1").Sort Key1:=Worksheets("Output").Columns("A"), Order1:=xlAscending, Header:=xlYes

      ' add dummy row of zeroes
      Worksheets("Output").Rows(1).EntireRow.Insert
      ncol = 1
      While ncol <= 5
         Worksheets("Output").Cells(2, ncol).Value = 0
         ncol = ncol + 1
      Wend

      ' add 2nd dummy row of zeroes
      Worksheets("Output").Rows(1).EntireRow.Insert
      ncol = 1
      While ncol <= 5
         Worksheets("Output").Cells(2, ncol).Value = 0
         ncol = ncol + 1
      Wend

      ' add column headers to Output Worksheet
      Worksheets("Output").Cells(1, 1).Value = "siteID"
      Worksheets("Output").Cells(1, 2).Value = "Lat"
      Worksheets("Output").Cells(1, 3).Value = "Lon"
      Worksheets("Output").Cells(1, 4).Value = "CalYrBP"
      Worksheets("Output").Cells(1, 5).Value = "InterpVal"

      ' add -999 flag to last row + 1 of cal yr BP field
      lastrow = Worksheets("Output").UsedRange.Rows.Count
      Worksheets("Output").Range("A" & lastrow + 1).Value = -999

   curRow = 3 ' because of two dummy rows of zeroes

   While curRow <= endRow

      ' get the siteID in the current row
      siteIdCell = "A" & curRow
      siteID = Worksheets("Output").Range(siteIdCell).Value

      ' get the siteID of the next row
      nextSiteIdCell = "A" & curRow + 1
      nextSiteId = Worksheets("Output").Range(nextSiteIdCell).Value

      ' get the calYr in the current row
      calYrCell = "D" & curRow
      calYr = Worksheets("Output").Range(calYrCell).Value

      ' get the calYr of the next row
      nextCalYrCell = "D" & curRow + 1
      nextCalYr = Worksheets("Output").Range(nextCalYrCell).Value

      ' reset this variable
      CalYrInt = 0


      ' If new site on next row, start the interp for this group of values
      ' NOTE: place a -999 in the empty cell after the last siteID

      If nextSiteId <> siteID Then

      'set min range for interp
     interpStart = curRow - vcount + 1
```

```
 'set max range for interp
interpEnd = curRow

' will handle case 3 below
calYrInterp = -999
interpLast = -999

Do While CalYrInt <= calYrStop
' set start date for interp
' this will remove #N/A values and avoid extrapolations

    If Worksheets(1).Range("D" & interpStart).Value <= CalYrInt Then
       calYrInterp = CalYrInt
       Exit Do ' exit the do while once we have a value here
    End If

    If interpInterval = 100 Then
      CalYrInt = CalYrInt + 100
    ElseIf interpInterval = 50 Then
       CalYrInt = CalYrInt + 50
    End If
Loop

CalYrInt = calYrStop
Do While CalYrInt >= 0
    ' set end date for interp
    ' this will remove #N/A values and avoid extrapolations

    If Worksheets(1).Range("D" & curRow).Value >= CalYrInt Then
       interpLast = CalYrInt
       Exit Do
    End If

    If interpInterval = 100 Then
       CalYrInt = CalYrInt - 100
    ElseIf interpInterval = 50 Then
       CalYrInt = CalYrInt - 50
    End If ' exit the do while once we have a value here
Loop

While calYrInterp <= interpLast

     ' CASES TO HANDLE HERE:
     '1. reconstruction value at 0, 100, 200 etc (do not interpolate, just use the value)
     '2. one reconstruction value between 2 intervals (e.g. 67, cannot interpolate to either 0 or 100 bp,
     ' therefore the value should be excluded)
     '3. two or more reconstruction values between 2 intervals (e.g. 37 and 80 -
     ' cannot interpolate to either 0 or 100 here, therefore value should be excluded)
     '4. several reconstruction values around one interval (e.g. -21, -7, 4, 23, 49), interpolate at 0 bp only)
     '5. reconstruction values starting at >= 100 and ending < 2000 (only interpolate intervals in between starting
     ' dates of reconstruction (this is handled by the do while loops above)


     ' case 1: if there is a reconstruction value at 0,100, 200 bp, etc then do not interpolate, simply use reconstruction value
     If (Worksheets("Output").Range("D" & curRow).Value = calYrInterp) Then

           Worksheets("Output").Range("G" & curRowWS2).Value = Worksheets("Output").Range("A" & curRow).Value
           Worksheets("Output").Range("H" & curRowWS2).Value = Worksheets("Output").Range("B" & curRow).Value
           Worksheets("Output").Range("I" & curRowWS2).Value = Worksheets("Output").Range("C" & curRow).Value
           Worksheets("Output").Range("J" & curRowWS2).Value = calYrInterp

           If decPlaces = 5 Then 'this is the max number of decial places
              Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.00000"
           ElseIf decPlaces = 4 Then
              Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.0000"
           ElseIf decPlaces = 3 Then
              Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.000"
           ElseIf decPlaces = 2 Then
              Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.00"
           ElseIf decPlaces = 4 Then
              Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.0"
           End If
        Worksheets("Output").Range("K" & curRowWS2).Value = Worksheets("Output").Range("E" & curRow).Value
        ' increase row count for output
        curRowWS2 = curRowWS2 + 1
```

```
      Else
         ' case 2: only interpolate if more than 1 reconstructed value
         ' case 3: calYrInterp and interpLast will have values of -999
         ' case 4: calYrInterp and interpLast might be both 0 or 100 or another interval

         If (interpStart <> interpEnd And calYrInterp >= 0 And interpLast >= 0) Then
            'write SiteID, Lat, Lon & CalYrMean to first worksheet, then do the interp
            Worksheets("Output").Range("G" & curRowWS2).Value = Worksheets("Output").Range("A" & curRow).Value
            Worksheets("Output").Range("H" & curRowWS2).Value = Worksheets("Output").Range("B" & curRow).Value
            Worksheets("Output").Range("I" & curRowWS2).Value = Worksheets("Output").Range("C" & curRow).Value
            Worksheets("Output").Range("J" & curRowWS2).Value = calYrInterp

            If decPlaces = 5 Then 'this is the max number of decial places
               Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.00000"
            ElseIf decPlaces = 4 Then
               Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.0000"
            ElseIf decPlaces = 3 Then
               Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.000"
            ElseIf decPlaces = 2 Then
               Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.00"
            ElseIf decPlaces = 4 Then
               Worksheets("Output").Range("K" & curRowWS2).NumberFormat = "0.0"
            End If

            ' this is the interpolation formula: DO NOT TOUCH THIS!!!
            Worksheets("Output").Range("K" & curRowWS2).Formula = "=FORECAST(" & calYrInterp & ",OFFSET($E$" & interpStart & ",
                  MATCH(" & calYrInterp & ",$D$" & interpStart & ":$D$" & interpEnd & ",1)-1,0,2),OFFSET($D$" & interpStart & ",
                  MATCH(" & calYrInterp & ",$D$" & interpStart & ":$D$" & interpEnd & ",1)-1,0,2))"

            ' increase row count for output
            curRowWS2 = curRowWS2 + 1

         End If

      End If

      If (interpInterval = 100) Then
         calYrInterp = calYrInterp + 100
      Else ' 50 yr interpolation interval
         calYrInterp = calYrInterp + 50
      End If

   Wend

   'reset this to interpolate group of reconstructed values at next site
   vcount = 0

   ' stop interpolating if siteID = -999
   If nextSiteId = -999 Then

      ' copy this output to new worksheet
      ' exclude interpolations of dummy row
      ' and exclude #N/A values

      lastRowWS2 = curRowWS2
      curRowWS2 = 1
      curRow = 2

      Sheets.Add.Name = "FINAL"

      While curRowWS2 <= lastRowWS2
         If (Worksheets("Output").Range("G" & curRowWS2).Value = 0) Or (IsNumeric(Worksheets("Output").Range("K" & curRowWS2).
               Value) = False) Then
         Else
            Worksheets("Output").Rows(curRowWS2).Copy
            Worksheets("FINAL").Rows(curRow).PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False, Transpose:=
                  False
            curRow = curRow + 1
         End If
         curRowWS2 = curRowWS2 + 1
      Wend

      ' format the lat/lon columns to number and 5 decimal places
      Worksheets("FINAL").Range("H2").NumberFormat = "0.00000"
      Worksheets("FINAL").Range("I2").NumberFormat = "0.00000"
```

```
          ' delete extra columns
          Worksheets("FINAL").Range("A1:F" & curRow).Delete

        ' add column headers to Output Worksheet
         Worksheets("FINAL").Range("A1").Value = "siteID"
         Worksheets("FINAL").Range("B1").Value = "LatDD"
         Worksheets("FINAL").Range("C1").Value = "LonDD"
         Worksheets("FINAL").Range("D1").Value = "CalYrBP"
         Worksheets("FINAL").Range("E1").Value = "InterpVal"

         ' save this worksheet as a .csv file
         ActiveWorkbook.SaveAs Filename:="final_reconstruction.csv", FileFormat:=xlCSV, CreateBackup:=False
         MsgBox ("DONE! Formatted interpolated output on FINAL worksheet saved as a CSV file.")
         Exit Sub
       End If

    End If

    curRow = curRow + 1
    vcount = vcount + 1

  Wend

End If

End Sub
```

These are the verification results of the linear interpolation macro. 74 random sites were selected for testing purposes. If the output is satisfactory: passed = y, otherwise passed = n.

Script for calculating anomalies for all sites in North America by CalYrBP:

```
Sub calcMTWAanomaliesByCalYrBP()


' Make sure reconstruction file is interpolated and is sorted by CalYrBP first!
' written by Matthew Ladd
' last modified: November 3, 2011


' initialize the variables
Dim anomalySum, anomalyMean, MTWAmean, MTWAsum As Double
Dim startRow, endRow, lastrow, curRow, currRow, curRowWS2, vcount, fcount, calYr, jcValue, nextCalYr, siteID, nextSiteID As Long
Dim calYrCell, nextCalYrCell, jcCell, anomalyCell, meanAnomalyCell, siteIDCell, nextSiteIDCell, fedorova, nextfedorova As String


startRow = 2
endRow = 65536 'change this depending on # of rows OR use -999 flag for last record
curRow = 2 'keep track of rows on 1st worksheet
curRowWS2 = 2 'keep track of rows on 2nd worksheet

MTWAsum = 0
MTWAmean = 0

vcount = 0 'keep track of the number of MTWA values for a given site

calYrFieldCol = Application.InputBox _
    (Prompt:="Enter the letter of the Cal Yr BP column (e.g. H):", _
    Title:="SPECIFY COLUMN", Type:=2)
On Error GoTo 0

anomalyCol = Application.InputBox _
    (Prompt:="Enter the letter of the anomaly column (e.g. CL):", _
    Title:="SPECIFY COLUMN", Type:=2)
On Error GoTo 0

joincountCol = Application.InputBox _
```

Table 16: Linear interpolation test results (part 1)

SiteID	AgeMin	AgeMax	InterpMin	InterpMax	Range	Passed
8	-40	6000	0	6000	6000	y
29	-12	513	0	500	500	y
203	0	28780	0	28700	28700	y
205	-33	141	0	100	100	y
217	0	13713	0	13700	13700	y
222	-29	11319	0	11300	11300	y
244	444	26480	500	26400	25900	y
268	0	12939	0	12900	12900	y
275	0	12734	0	12700	12700	y
295	20118	29892	20200	29800	9600	y
326	-33	220	0	200	200	y
336	0	29028	0	29000	29000	y
338	133	12583	200	12500	12300	y
344	-10	12977	0	12900	12900	y
360	727	11502	800	11500	10700	y
362	0	7706	0	7700	7700	y
509	10183	16417	10200	16400	6200	y
511	438	7842	500	7800	7300	y
531	0	906	0	900	900	y
558	254	18975	300	18900	18600	y
656	40	13592	100	13500	13400	y
657	0	16163	0	16100	16100	y
665	0	7714	0	7700	7700	y
757	0	12515	0	12500	12500	y
800	-30	12757	0	12700	12700	y
808	-35	10183	0	10100	10100	y
816	-40	9655	0	9600	9600	y
826	-38	15359	0	15300	15300	y
834	4760	11910	4800	11900	7100	y
845	0	13011	0	13000	13000	y
945	0	1027	0	1000	1000	y
1077	8989	15388	9000	15300	6300	y
1106	13	3221	100	3200	3100	y
1483	10	14093	100	14000	13900	y

Table 17: Linear interpolation test results (part 2)

SiteID	AgeMin	AgeMax	InterpMin	InterpMax	Range	Passed
1503	1598	18555	1600	18500	16900	y
1530	-34	696	0	600	600	y
1546	0	12466	0	12400	12400	y
1594	10	12960	100	12900	12800	y
1664	-12	320	0	300	300	y
1677	-32	13699	0	13600	13600	y
1709	260	10389	300	10300	10000	y
1711	-35	13705	0	13700	13700	y
1738	10144	19418	10200	19400	9200	y
1759	0	13208	0	13200	13200	y
1761	-40	2371	0	2300	2300	y
1771	-9	1006	0	1000	1000	y
1804	20	28509	100	28500	28400	y
1824	-14	6062	0	6000	6000	y
1901	12	13652	100	13600	13500	y
1920	26	19083	100	19000	18900	y
1945	577	6604	600	6600	6000	y
1946	538	6330	600	6300	5700	y
1955	0	29184	0	29100	29100	y
1957	0	13263	0	13200	13200	y
1975	158	13904	200	13900	13700	y
2194	0	18597	0	18500	18500	y
2209	0	10200	0	10200	10200	y
2267	11864	16048	11900	16000	4100	y
2274	32	15246	100	15200	15100	y
2290	0	12732	0	12700	12700	y
2518	-9	646	0	600	600	y
2520	33	19877	100	19800	19700	y
2526	43	13127	100	13100	13000	y
2527	44	30081	100	30000	29900	y
2532	0	12162	0	12100	12100	y
2550	6	11884	100	11800	11700	y
2555	3033	7138	3100	7100	4000	y
2800	706	26474	800	26400	25600	y
2828	0	9284	0	9200	9200	y
2839	0	22984	0	22900	22900	y
2846	-41	2450	0	2400	2400	y
2847	-36	2090	0	2000	2000	y
2941	-36	1393	0	1300	1300	y
2948	0	10293	0	10200	10200	y

```
    (Prompt:="Enter the letter of the join count column (e.g. C):", _
    Title:="SPECIFY COLUMN", Type:=2)
On Error GoTo 0

climVarName = Application.InputBox _
    (Prompt:="Enter the abbrievated climate variable name (e.g. MTWA, MIPT, etc.):", _
    Title:="ENTER VARIABLE NAME", Type:=2)
On Error GoTo 0

' extract time slice of data or use all data
Dim interpQuestion As VbMsgBoxResult

interpQuestion = MsgBox("Do you wish to use all data? If no, you will be asked for start/end dates for the time slice of data. For the
      time-slice option the appropriate data will be copied to the FINAL worksheet.", vbQuestion + vbYesNo)

If interpQuestion = vbYes Then
     allData = True
ElseIf interpQuestion = vbNo Then
     allData = False
End If

If allData = False Then

    Sheets.Add.Name = "OUTPUT"

    calYrYoungVal = Application.InputBox _
        (Prompt:="Enter the youngest cal Yr BP interval (min is 0):", _
        Title:="TIME-SLICE MIN", Type:=1)
    On Error GoTo 0

    calYrOldVal = Application.InputBox _
        (Prompt:="Enter the oldest cal Yr BP interval (max must match your data set):", _
        Title:="TIME-SLICE MAX", Type:=1)
    On Error GoTo 0

    ' sort cal yr BP field ascending
    Worksheets(2).Range(calYrFieldCol & "2").Sort Key1:=Worksheets(2).Columns(calYrFieldCol), Order1:=xlAscending, Header:=xlYes

    currRow = 2
    Do While currRow <= endRow
        calYrCell = calYrFieldCol & currRow
        calYr = Worksheets(2).Range(calYrCell).Value
        If (calYr = calYrYoungVal) Then
           firstrow = currRow
           Exit Do
        End If
        currRow = currRow + 1
    Loop

    currRow = 2
    Do While currRow <= endRow
        calYrCell = calYrFieldCol & currRow
        calYr = Worksheets(2).Range(calYrCell).Value
        If (calYr = calYrOldVal) Then
           lastrow = currRow
        End If
        currRow = currRow + 1
    Loop

    ' then copy appropriate column headers paste to worksheet FINAL
    Worksheets(2).Range("A1:" & anomalyCol & "1").Copy _
    Destination:=Worksheets(1).Range("A1")

    ' then copy appropriate data and then paste to worksheet FINAL
    Worksheets(2).Range("A" & firstrow & ":" & anomalyCol & lastrow).Copy _
    Destination:=Worksheets(1).Range("A2")

    Sheets.Add.Name = "FINAL"

    ' initialize the variables
    startRow = 2
    endRow = 65536 'change this depending on # of rows
    curRow = 2 'keep track of rows on 1st worksheet
    curRowWS2 = 2 'keep track of rows on 2nd worksheet

    anomalySum = 0
```

```
anomalyMean = 0

vcount = 1 'keep track of the number of MTWA values for a given site

' put a header for the output on 3rd worksheet
Worksheets(1).Range("A1").Value = "siteID"
Worksheets(1).Range("B1").Value = "LATDD"
Worksheets(1).Range("C1").Value = "LONDD"
Worksheets(1).Range("D1").Value = "mean" & climVarName

siteIDCol = Application.InputBox _
    (Prompt:="Enter the column with siteID values (e.g. E):", _
    Title:="SITEID COLUMN", Type:=2)
On Error GoTo 0

latCellCol = Application.InputBox _
    (Prompt:="Enter the column with latitude values (e.g. F):", _
    Title:="LATITUDE COLUMN", Type:=2)
On Error GoTo 0

lonCellCol = Application.InputBox _
    (Prompt:="Enter the column with longitude values (e.g. G):", _
    Title:="LONGITUDE COLUMN", Type:=2)
On Error GoTo 0

' sort cal yr BP field ascending
Worksheets(2).Range(siteIDCol & "2").Sort Key1:=Worksheets(2).Columns(siteIDCol), Order1:=xlAscending, Header:=xlYes

'add -999 flag to last row + 1 of cal yr BP field
lastrow = Worksheets(2).UsedRange.Rows.Count
Worksheets(2).Range(siteIDCol & lastrow + 1).Value = -999

While curRow <= endRow

    ' get the siteID in the current row
    siteIDCell = siteIDCol & curRow
    siteID = Worksheets(2).Range(siteIDCell).Value

    ' get the siteID of the next row
    nextSiteIDCell = siteIDCol & curRow + 1
    nextSiteID = Worksheets(2).Range(nextSiteIDCell).Value

    ' get the lat/lon values
    latCell = latCellCol & curRow
    latVal = Worksheets(2).Range(latCell).Value

    lonCell = lonCellCol & curRow
    lonVal = Worksheets(2).Range(lonCell).Value

    ' ensure we have the correct cells for the MTWA and MTWA mean
    meanAnomalyCell = "D" & curRowWS2 ' values written in 2nd col on 2nd worksheet
    anomalyCell = anomalyCol & curRow ' change if not using first analog

    ' keep a running sum of the MTWA values
    anomalySum = anomalySum + Worksheets(2).Range(anomalyCell).Value

    ' if new site on next row calculate the mean MTWA and output to 2nd worksheet
    If nextSiteID > siteID Or nextSiteID = -999 Then

        anomalyMean = anomalySum / vcount

        Worksheets(1).Range(meanAnomalyCell).NumberFormat = "0.00000"
        Worksheets(1).Range(meanAnomalyCell).Value = anomalyMean
        Worksheets(1).Range("A" & curRowWS2).Value = siteID

        ' output lat/lon values here
        Worksheets(1).Range("B" & curRowWS2).Value = latVal
        Worksheets(1).Range("C" & curRowWS2).Value = lonVal

        ' reset this to zero since next row is new calYrBP value
        anomalySum = 0
        anomalyMean = 0
        curRow = curRow + 1
        curRowWS2 = curRowWS2 + 1
        vcount = 1
```

```
            If (nextSiteID = -999) Then
               MsgBox ("PROCESSING FINISHED!")
               Exit Sub
            End If
        Else
            ' if same value of CalYrBP in next row just go to that row
            curRow = curRow + 1
            vcount = vcount + 1

        End If

    Wend


Else ' allData = true

    Sheets.Add.Name = "FINAL"

    ' now ask to calculate anomalies by Cal Yr BP either for all data or by Fedorova regions
    interpQuestion = MsgBox("Do you wish to split the output by Fedorova ecoregions? If no, the anomalies will be calculated using all
          of the data. Fedorova regions of -99 and no data will be excluded.", vbQuestion + vbYesNo)

    If interpQuestion = vbYes Then

        ' ask which column has Fedorova region
        fedorovaCol = Application.InputBox _
            (Prompt:="Enter the letter of the Fedorova column (e.g. CK):", _
            Title:="SPECIFY COLUMN", Type:=2)
        On Error GoTo 0

        ' sort cal yr BP field descending
        Worksheets(2).Range(calYrFieldCol & "2").Sort Key1:=Worksheets(2).Columns(calYrFieldCol), Order1:=xlDescending, Header:=xlYes

        ' sort Fedorova field ascending
        Worksheets(2).Range(fedorovaCol & "2").Sort Key1:=Worksheets(2).Columns(fedorovaCol), Order1:=xlDescending, Header:=xlYes


        ' put a header for the output on 2nd worksheet
        Worksheets(1).Range("A1").Value = "calMean"


        'add -999 flag to last row + 1 of cal yr BP field
        lastrow = Worksheets(2).UsedRange.Rows.Count
        Worksheets(2).Range(calYrFieldCol & lastrow + 1).Value = -999

        fcount = 2 'keep track of fedorova regions (2nd column is first fedorova region)

        While curRow <= endRow

            ' get the joincount in the current row
            jcCell = joincountCol & curRow
            jcValue = Worksheets(2).Range(jcCell).Value

            ' get the calYr in the current row
            calYrCell = calYrFieldCol & curRow
            calYr = Worksheets(2).Range(calYrCell).Value

            ' get the calYr of the next row
            nextCalYrCell = calYrFieldCol & curRow + 1
            nextCalYr = Worksheets(2).Range(nextCalYrCell).Value

            ' get the fedorova in the current row
            fedorovaCell = fedorovaCol & curRow
            fedorova = Worksheets(2).Range(fedorovaCell).Value

            ' get the fedorova in the next row
            nextfedorovaCell = fedorovaCol & curRow + 1
            nextfedorova = Worksheets(2).Range(nextfedorovaCell).Value


            ' ensure we have the correct cells for the MTWA and MTWA mean
            MeanMTWAcell = "B" & curRowWS2 ' values written in 2nd col on 2nd worksheet
            MTWAcell = anomalyCol & curRow ' change if not using first analog


            ' keep a running sum of the MTWA values
```

```
' only include records that were successfully joined (i.e first join count column (A)
' where joincount=1) after the reconstruction was spatially joined with the modern

If jcValue = 1 Then
   MTWAsum = MTWAsum + Worksheets(2).Range(MTWAcell).Value
   vcount = vcount + 1
End If


' if new site on next row calculate the mean MTWA and output to 2nd worksheet
' make sure blank row after last record has a value of -999 for calMean

If nextCalYr <> calYr Then

    If fedorova = "Mediterranian" Or fedorova = "Arctic" Or fedorova = "" Or fedorova = -99 Then
       ' exclude Arctic, Mediterranian, -99 and no data fedorova regions
       ' these are data sparse regions (esp Mediterranian)

       ' reset these to zero since next row is new calYrBP and fedorova value
        MTWAsum = 0
        MTWAmean = 0
        vcount = 0

        curRow = curRow + 1

        flagFed = False

    Else

        flagFed = True

        If vcount > 0 Then 'avoid division by zero
           MTWAmean = MTWAsum / vcount
        End If

        ' write the cal Yr BP interval

        If fcount = 2 Then
           Worksheets(1).Range("A" & curRowWS2).Value = calYr
        End If


        ' ensure header will identify anomalies by fedorova region
        Worksheets(1).Cells(1, fcount).Value = climVarName & "_" & fedorova

        ' keep this precision
        Worksheets(1).Cells(curRowWS2, fcount).NumberFormat = "0.00000"

        ' write the mean anomaly
        Worksheets(1).Cells(curRowWS2, fcount).Value = MTWAmean

        ' reset this to zero since next row is new calYrBP and fedorova value
        MTWAsum = 0
        MTWAmean = 0
        vcount = 0

        curRow = curRow + 1
        curRowWS2 = curRowWS2 + 1

    End If


Else
    ' if same value of CalYrBP in next row just go to that row
    curRow = curRow + 1
End If

If nextfedorova <> fedorova Then

    ' reset this
    curRowWS2 = 2

    If (flagFed = True) Then
        ' next Cal YR BP cell is also next Fedorova region
        fcount = fcount + 1
    End If
```

```
        End If

        If (nextCalYr = -999) Then
            MsgBox ("PROCESSING FINISHED!")
            Exit Sub
        End If

    Wend

ElseIf interpQuestion = vbNo Then

    ' put a header for the output on 2nd worksheet
    Worksheets(1).Range("A1").Value = "calMean"
    Worksheets(1).Range("B1").Value = "mean" & climVarName

    ' sort cal yr BP field descending
    Worksheets(2).Range(calYrFieldCol & "2").Sort Key1:=Worksheets(2).Columns(calYrFieldCol), Order1:=xlDescending, Header:=xlYes

    'add -999 flag to last row + 1 of cal yr BP field
    lastrow = Worksheets(2).UsedRange.Rows.Count
    Worksheets(2).Range(calYrFieldCol & lastrow + 1).Value = -999

    While curRow <= endRow

        ' get the joincount in the current row
        jcCell = joincountCol & curRow
        jcValue = Worksheets(2).Range(jcCell).Value


        ' get the calYr in the current row
        calYrCell = calYrFieldCol & curRow
        calYr = Worksheets(2).Range(calYrCell).Value


        ' get the calYr of the next row
        nextCalYrCell = calYrFieldCol & curRow + 1
        nextCalYr = Worksheets(2).Range(nextCalYrCell).Value


        ' ensure we have the correct cells for the MTWA and MTWA mean
        MeanMTWAcell = "B" & curRowWS2 ' values written in 2nd col on 2nd worksheet
        MTWAcell = anomalyCol & curRow ' change if not using first analog


        ' keep a running sum of the MTWA values
        ' only include records that were successfully joined (i.e first join count column (A)
        ' where joincount=1) after the reconstruction was spatially joined with the modern


        If jcValue = 1 Then
           MTWAsum = MTWAsum + Worksheets(2).Range(MTWAcell).Value
           vcount = vcount + 1
        End If


        ' if new site on next row calculate the mean MTWA and output to 2nd worksheet
        ' make sure blank row after last record has a value of -999 for calMean


        If nextCalYr <> calYr Then


            If vcount > 0 Then 'avoid division by zero
               MTWAmean = MTWAsum / vcount
            End If

            Worksheets(1).Range(MeanMTWAcell).NumberFormat = "0.00000" 'keep this precision
            Worksheets(1).Range(MeanMTWAcell).Value = MTWAmean
            Worksheets(1).Range("A" & curRowWS2).Value = calYr

            ' reset this to zero since next row is new calYrBP value
            MTWAsum = 0
            MTWAmean = 0
            curRow = curRow + 1
            curRowWS2 = curRowWS2 + 1
            vcount = 0
```

```
            If (nextCalYr = -999) Then
              MsgBox ("PROCESSING FINISHED!")
              Exit Sub
            End If
        Else
            ' if same value of CalYrBP in next row just go to that row
            curRow = curRow + 1
        End If

    Wend

  End If

  ' save the dbf as a new workbook
  ActiveWorkbook.SaveAs Filename:="recon_anomalies.xls", FileFormat:=xlWorkbookNormal, CreateBackup:=False
  MsgBox ("WORKBOOK SAVED!")

End If

End Sub
```

Procedure to format variable of interest for mapping (average of all samples at each site):
NOTE: replace GDD5 with variable of interest

1. Open the file (e.g. ReconsGDD5.xls) in MS Excel™.
2. Sort by field "CalyrBP".
3. Select the time interval of data need (e.g. 5500 to 6500) & copy to new worksheet.
4. In the new worksheet sort the data by siteID.
5. Remove extra columns for analogs 2-20 which are not needed (e.g. 2-20, ID2-ID20, GDD51-GDD520).
6. Open Excel macro "calcSiteMeansMacro" (see below) and paste in VB editor window (alt + F11 in Excel then paste code into "ThisWorkbook" for current worksheet).
7. Check number of rows in worksheet and edit line 8 of macro (endRow should equal last row of data in worksheet). Also line 39 (NumberFormat = 0.0) may need to be modified depending on the number of decimal places of the variable.
8. Create a new column beside last column (col J) with data for average of samples at each site (e.g. header should be GDD5 in col K).
9. Run macro and delete col J, last column (GDD5 in this case) should have one average value for variable of interest for each site.

10. Sort this column and delete rows with no value or blank rows.

11. Finally save the worksheet with an easy to identify name.

12. The data file is now ready to be x-y mapped in GIS software.

D.5 Spatial filtering steps & Excel macro to spatially filter reconstructions

1. Before spatial filtering, convert the lat/lon values into metres (x-y coordinates) using an Albers Equidistant conic transformation. See the web resource at http://www.shamskm.com/files/lat-long-to-northing-and-easting.pdf

2. Install Rookcase add-in for Excel (see documentation at http://www.lpc.uottawa.ca/data/scripts/)

3. Open the CSV file in Excel and copy the x, y, z values (z being the variable of interest) to a new worksheet.

4. Start rookcase and determine the optimal lag distance for the data. Check Moran's I values (and Z values)

5. Once lag distance is decided, do comparisons (same signs for Moran's I/Geary's C and z-values, see table below).

   Key (for Nkm_lag_code worksheets) where N=50 to N=500

   1 = col DEF all negative
   2 = col DEF all positive
   3 = col DEF combination of neg/pos
   4 = col DEF all zero

   NOTE: only keep sites where CODE = 1 or CODE = 2

6. Remove sites that do not meet the criteria and re-do the anomaly maps.

7. Compare filtered reconstructions to unfiltered reconstructions.

Table 18: Positive, zero, and negative values and combinations for Moran's I and Moran's I (z-values) and the number of neighbors & percentages for sites at different lag distances

–	MI	–	–	zMI	–	–
lag_km	neg	zero	pos	neg	zero	pos
50	90	227	192	86	227	196
%	0.176817	0.445972	0.37721	0.168959	0.445972	0.385069
100	148	105	256	79	105	325
%	0.290766	0.206287	0.502947	0.155206	0.206287	0.638507
150	181	49	279	71	49	389
%	0.355599	0.096267	0.548134	0.139489	0.096267	0.764244
200	201	30	278	67	30	412
%	0.394892	0.058939	0.546169	0.131631	0.058939	0.80943
250	202	17	290	65	17	427
%	0.396857	0.033399	0.569745	0.127701	0.033399	0.8389
300	195	13	301	61	13	435
%	0.383104	0.02554	0.591356	0.119843	0.02554	0.854617
350	205	8	296	58	8	443
%	0.40275	0.015717	0.581532	0.113949	0.015717	0.870334
400	229	6	274	58	6	445
%	0.449902	0.011788	0.53831	0.113949	0.011788	0.874263
450	237	5	267	57	5	447
%	0.465619	0.009823	0.524558	0.111984	0.009823	0.878193
500	243	3	263	56	3	450
%	0.477407	0.005894	0.516699	0.11002	0.005894	0.884086

Table 19: Cumulative frequencies from the table of results above

Bin	Freq	Cumulative	Bin	Freq	Cumulative
0	227	44.60%	0	227	44.60%
1	126	69.35%	1	126	69.35%
2	80	85.07%	2	80	85.07%
3	32	91.36%	3	32	91.36%
4	22	95.68%	4	22	95.68%
5	10	97.64%	5	10	97.64%
6	9	99.41%	6	9	99.41%
7	1	99.61%	7	1	99.61%
8	1	99.80%	8	1	99.80%
9	1	100.00%	9	1	100.00%
More	0	100.00%	More	0	100.00%

Table 20: Same as above except for Geary's C (z-value)

lag_km	neg	zero	pos
50	246	2	36
%	0.48330	0.44597	0.07073
100	382	105	22
%	0.75049	0.20629	0.04322
150	446	49	14
%	0.87623	0.09627	0.02751
200	468	30	11
%	0.91945	0.05894	0.02161
250	487	17	5
%	0.95678	0.03340	0.00982
300	493	13	3
%	0.96857	0.02554	0.00589
350	499	8	2
%	0.98035	0.01572	0.00393
400	501	6	2
%	0.98428	0.01179	0.00393
450	502	5	2
%	0.98625	0.00982	0.00393
500	505	3	1
%	0.99214	0.00589	0.00197

Table 21: Code assigned to combinations of values (see key below)

LocalMoransI	zLocalMoransI	zLocalGearysC	CODE
1.57450	1.13201	1.04305	2
-11.94034	-6.06285	4.84090	3
0	0	0	4
-0.65485	-0.32873	-0.17307	1

Table 22: Coded sites at different lag distances based on key below

code	50k	100k	150k	200k	250k	300k
1	50	46	46	36	40	36
%	0.09823	0.09037	0.07662	0.07073	0.07859	0.07073
2	5	2	1	0	0	0
%	0.00982	0.00393	0.00197	0.00000	0.00000	0.00000
3	257	377	436	455	459	466
%	0.50491	0.74067	0.85658	0.89391	0.90177	0.91552
4	197	84	33	18	10	7
%	0.38703	0.16503	0.06483	0.03536	0.01965	0.01375

8. Produce map to show heterogeneity of unfiltered reconstruction. This method simply maps the anomalies and classifies them either as negative, zero, or positive. The break values are -0.000001, 0.000001 and a value higher than the highest anomaly to produce three classes.

9. Also, look into Getis-Ord (Gi*) stat for magnitude of clusters (optional). High (low) Getis values indicate clusters with values higher (lower) than the mean value.

10. Comparison of gridded values versus point values within grid (optional).

```
Sub filterSitesByCode()

Dim MIval, zMIval, zGCval As Single
Dim endRow, curRow As Integer
Dim MIcell, zMIcell, zGCcell, codeCell As String

' Make sure file has columns X,Y,Z followed by localMoransI, zlocalMoransI, and zLocalGearysC
' written by Matthew Ladd
' last modified: July 9, 2011

' initialize the variables
endRow = 510 'change this depending on # of rows
curRow = 2 'keep track of rows (row Index)

While curRow <= endRow ' for each site

' get the local Moran's value
MIcell = "D" & curRow
MIval = Worksheets(1).Range(MIcell).Value

' get the local Moran's z-value
zMIcell = "E" & curRow
zMIval = Worksheets(1).Range(zMIcell).Value

' get the local Geary's C z-value
zGCcell = "F" & curRow
zGCval = Worksheets(1).Range(zGCcell).Value

' cell reference to write code
codeCell = "G" & curRow

' code based on MIval, zMIval, zGCval: -,-,- (1), +,+,+ (2), 0,0,0 (4), else (3)
If MIval < 0 And zMIval < 0 And zGCval < 0 Then
   Worksheets(1).Range(codeCell).Value = 1 ' keep site
ElseIf MIval > 0 And zMIval > 0 And zGCval > 0 Then
   Worksheets(1).Range(codeCell).Value = 2 ' keep site
ElseIf MIval = 0 And zMIval = 0 And zGCval = 0 Then
   Worksheets(1).Range(codeCell).Value = 4 ' discard site
Else ' signs are mixed positive and negative
   Worksheets(1).Range(codeCell).Value = 3 ' discard site
End If

curRow = curRow + 1

Wend

End Sub
Sub countCodes()

Dim endRow, curRow, cCount1, cCount2, cCount3, cCount4 As Integer
Dim codeCell As String

' Make sure file has columns X,Y,Z followed by localMoransI, zlocalMoransI, and zLocalGearysC, code is column G
' written by Matthew Ladd
' last modified: July 10, 2011

' initialize the variables
endRow = 510 'change this depending on # of rows
```

```
curRow = 2 'keep track of rows (row Index)

cCount1 = 0
cCount2 = 0
cCount3 = 0
cCount4 = 0


While curRow <= endRow ' for each site

' get the CODE value
codeCell = "G" & curRow
codeVal = Worksheets(1).Range(codeCell).Value

If codeVal = 1 Then
    cCount1 = cCount1 + 1
ElseIf codeVal = 2 Then
    cCount2 = cCount2 + 1
ElseIf codeVal = 3 Then
    cCount3 = cCount3 + 1
ElseIf codeVal = 4 Then
    cCount4 = cCount4 + 1
End If

curRow = curRow + 1

Wend


' now write the totals by code & percentages
curRow = 2
codeCell = "C" & curRow

Worksheets(2).Range(codeCell).Value = cCount1
curRow = curRow + 1
codeCell = "C" & curRow

Worksheets(2).Range(codeCell).Value = cCount1 / 509
curRow = curRow + 1
codeCell = "C" & curRow

Worksheets(2).Range(codeCell).Value = cCount2
curRow = curRow + 1
codeCell = "C" & curRow

Worksheets(2).Range(codeCell).Value = cCount2 / 509
curRow = curRow + 1
codeCell = "C" & curRow

Worksheets(2).Range(codeCell).Value = cCount3
curRow = curRow + 1
codeCell = "C" & curRow

Worksheets(2).Range(codeCell).Value = cCount3 / 509
curRow = curRow + 1
codeCell = "C" & curRow

 Worksheets(2).Range(codeCell).Value = cCount4
curRow = curRow + 1
codeCell = "C" & curRow

Worksheets(2).Range(codeCell).Value = cCount4 / 509
curRow = curRow + 1
codeCell = "C" & curRow


End Sub
```

Methodology for creating a matrix in MS Access™with output from resolution calculations for all sites (sample resolution and total dates) and the anomaly (of MTWA64) at that site if it is still kept after filtering at each calendar year BP interval.

Table 23: Sample output from Rookcase. 4th column is local Moran's I with a 50 km lag distance.

X	Y	anomaly	Local_50k
-2085015	-439122	0.366667	0
1134008	922220.8	-1.58889	0.178991
111591.9	567946.9	-0.4	0.027292
894269	-420951	0	0
790556	837410.4	0.9	-0.04753
-1587487	3112217	0	0.00202
-2598288	3908358	0	0.007525
1775917	1124274	0	0.00202

Method is as follows: Need to sort 2 tables from above by X or Y then copy the site values to 2nd table, then select local and zlocal for lag distance of interest, then insert a column for code value (this column can be deleted later) and run macro to filter out sites (keep only those with CODE=1 or CODE=2), and re-save at csv, then import into MS Access™, open matrix query, add table for selected calendar year BP, set up join (type 2) from siteID of allSites table to selected table, and re-save query, re-run query, repeat all steps for each calendar year BP, check the final output and export to MS Excel™.

D.6 Steps to acquire and use GHCN and WorldClim data for modern climate data in a reconstruction

All GHCN station data (adjusted) monthly means compiled into one file. The methodology is described below:

Steps to build GHCN station data file:

1. Download GHCN data file
2. Install GHCN package for R
3. Run the following code

```
path='R:\\LPC\\CLIMATE DATA\\station_data\\GHCN\\ghcnm.v3.0.0.20110912\\ghcnm.tavg.v3.0.0.20110912.qca.dat'
```

```
v3Mean <- readV3Data(filename = path,output = "Zoo")

[1] "Removing station with all data Missing"
Stations read in 7279
Stations with more than 1 year of data 7278

path2 = 'R:\\LPC\\CLIMATE DATA\\station_data\\GHCN\\ghcnm.v3.0.0.20110912\\ghcnm.tavg.v3.0.0.20110912.qca.inv'
inventory <- readInventory(filename = path2)
```

These steps are optional:

```
> V3Anomaly <- createAnomaly(v3Mean, criteria = list(Start = 1961, End = 1990, Years = 30, Threshold = 12))
> Data <- intersectInvData(inventory,V3Anomaly)
```

4. Do a write.csv for each of the v3Mean and inventory files.
5. Open the v3Mean file in MS Excel™then transpose it.
6. Run macro to fill in headers with Month & Year (data starts at Jan 1801).
7. Now merge inventory file and v3Mean file in MS Excel™.
8. Calculate July 1961-1990 anomalies for each station.
9. Export the file as CSV.
10. Now bring into ESRI ArcGIS™and spatially join to Whitmore modern.

For WorldClim data (http://www.worldclim.org) the steps are as follows:

1. Download worldclim data rasters for ESRI ArcGIS™
2. Import selected raster (∼2 GB) into ESRI ArcGIS™
3. Create new field in raster (temp = value / 10)
4. Now use Hawth's tools add in for ESRI ArcGIS™to intersect whitmore site with grid value (temp)
5. Now this modern climate file can be joined to the reconstructed file

So I first took the Whitmore pollen sum 64 (percentages) file and did a point intersect (ignoring differences in spatial reference) with the worldclim long term July mean raster (using Hawth's tools add in for ESRI ArcGIS™, cannot do a spatial join between a raster and a shapefile). Then I ran the MATTOOLS as usual with this modern file and specify the WorldClim column when running the "cbind"' function. Then I did the linear interpolation and then imported it into ESRI ArcGIS™to create the shapefile. I made sure lat/lon values and MTWA1 values had 2 decimal places. This shapefile is "intersected" with the same raster as before so that the anomalies

can be calculated.

The only issue is that the fine resolution of the WorldClim data relative to the Whitmore sites causes 60 sites to appear off the grid or over water instead of land. This could be fixed with more precise modern pollen sites. As a temporary solution, the New *et al.* (2002) July mean temperature values were assigned to these sites. Finally, I ran the macro in MS Excel™to average the anomalies by cal yr to create the time series.

D.7 Steps for revising figures in Chapter 2 (Viau *et al.*, 2012)

Figure 2. Hatching was added using GIS software to account for the areas where the reconstruction is less certain. These areas were adapted from Sawada *et al.* (2004). The areas are located in Alaska, Western and the Southwestern USA and the Canadian Arctic.

Figure 3. The T-S of mtwa reconstructions for each eco-region needed uncertainty curves. Uncertainties were calculated using Viau *et al.* (2006) Eq. 1 on the data file with MTWA modern and reconstructed values for 100 year intervals over the last 2,000 years. The excel macro writeOverlaps ensures that modern values are only used at sites where there is a reconstructed value for 0 cal Yr BP (i.e. blank cells are skipped). The uncertainties were added and subtracted from the reconstructed mtwa value at 100 year intervals for each eco-region.

Figure 4. This was a new figure (2 pages) using C2. Each page would have (A) the SCD through time for the top 5 analogs, (B) MTWA reconstruction for 2k, (C) site density, (D) dates and (E) samples through time. Panels A through E would be repeated for each of the 8 eco-regions: (1) coastal, (2) mountain, (3) prairies, (4) desert, (5) forest tundra, (6) boreal, (7) hardwood and (8) deciduous regions. The GPC paper provides more details on these eco-regions.

The reconstructions by eco-region were already completed so it was straightforward to calculate the number of sites through time (at 100 year intervals). Both of these data files were opened in C2 and plots created. For the radiocarbon dates, they had

to be extracted from the raw data (NAPD) and exported to a file called GPC_dates. A column called dates was added next to the CalYrBp column. These dates were plotted in C2. The samples through time data was calculated by taking the reconstructed file after running MATTOOLS (columns calYrBp and samples) and then adding another column with a 1 in each row for each sample number. The calYrBP column was sorted then a macro was run to total the 1s for each radiocarbon date. The samples column was replaced with the totals from the new column of data from the macro, then sorted. Any rows with blank values for samples were deleted, then the calYrBP column was resorted. Now the file has the number of samples at each date and can be plotted with C2.

D.8 Javascript tool to generate MATTOOLS code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Generate MATTOOLS &quot;R&quot; Code for North American Pollen Reconstructions</title>
<script language="javascript" type="text/javascript">
<!--

// written by Matthew Ladd
// last modified on October 26, 2011

function selectAll()
{
  for (var i=0;i<document.ReconForm.climVarList.options.length;i++)
  {
      document.ReconForm.climVarList.options[i].selected = true;
  }

}

function clearAll()
{
  for (var i=0;i<document.ReconForm.climVarList.options.length;i++)
  {
     document.ReconForm.climVarList.options[i].selected = false;
  }

}

function generateCode()
{
       // code written and tested by Matthew Ladd from the LPC
     // modified: October 26, 2011

       // first fix the file strings (double backslashes for file paths is needed)
       strFile1 = document.ReconForm.modFile.value
       var strFile1rep = strFile1.replace(/\\/g, "\\\\")

       strFile2 = document.ReconForm.fosFile.value
       var strFile2rep = strFile2.replace(/\\/g, "\\\\")


     climVarIndex = 0
       selClimVarName = ""
       codeString = ""
     strCounts = ""
       fileOutName = ""
       fileOutPath = ""

       fileOutName = document.ReconForm.fileOutputName.value
```

```
 fileOutPath = document.ReconForm.fileOutputPath.value

 LLfossil = document.ReconForm.flatlon.value
 LLmodern = document.ReconForm.mlatlon.value
 TaxaMod = document.ReconForm.mtaxa.value
 TaxaFos = document.ReconForm.ftaxa.value

 if (document.ReconForm.pollCounts.checked)
 {
        strCounts = "Counts=T"

 }
 else
 {
        strCounts = "Counts=F"
 }

climVarIndex = document.ReconForm.climVarCol1.value


 analogs = document.ReconForm.numOfAnalogs.value
 distMethod = document.ReconForm.distMethod.value

codeString = "Modpoll <- read.csv(\"" + strFile1rep + "\", header=TRUE, sep=\",\", na.strings=\"NA\", dec=\".\", strip.white=TRUE
      )" +
         "\n\nFospoll <- read.csv(\"" + strFile2rep + "\", header=TRUE, sep=\",\", na.strings=\"NA\", dec=\".\", strip.white=TRUE
               )" +
         "\n\nNA_recons_SCD=mat.dissim(inFossil=Fospoll,inModern=Modpoll,llMod=" + LLmodern + ",modTaxa=" + TaxaMod + ",llFoss="
               + LLfossil + ",fosTaxa=" + TaxaFos + ",numAnalogs=" + analogs + "," + strCounts + ",dist.method=\"" + distMethod +
                "\")" +
         "\n\nx = NA_recons_SCD"

 var curAnalog = 1

for (var i=0;i<document.ReconForm.climVarList.options.length;i++)
{

         var selClimVarName = document.ReconForm.climVarList.options[i].value

       // check if the variable is selected from the drop-down list
     if (document.ReconForm.climVarList.options[i].selected)
     {
                codeString += "\n\nRecons" + selClimVarName + "=cbind(NA_fossil[,c(2,4:6,9:11)],t(NA_recons_SCD$sqdist[1:" +
                      analogs + ",]),"

         while (curAnalog <= analogs) {
                if (curAnalog % 2 == 0) {
                       codeString += "ID" + curAnalog + "=Modpoll[x$position[" + curAnalog + ",],1], "
                       }
                else {
                       codeString += "\nID" + curAnalog + "=Modpoll[x$position[" + curAnalog + ",],1], "
                }
                curAnalog ++
         }
         curAnalog = 1

                while (curAnalog <= analogs)
                {
                if (curAnalog == analogs) {
                     if (curAnalog % 2 == 0) {
                                    codeString += selClimVarName + curAnalog + "=Modpoll[x$position[" + curAnalog + ",]," +
                                          climVarIndex + "])"
                              }
                              else {
                              codeString += "\n" + selClimVarName + curAnalog + "=Modpoll[x$position[" + curAnalog + ",]," +
                                     climVarIndex + "])"
                     }
                }
                else {
                       if (curAnalog % 2 == 0) {
                              codeString += selClimVarName + curAnalog + "=Modpoll[x$position[" + curAnalog + ",]," +
                                     climVarIndex + "],"
                       }
                       else {
                              codeString += "\n" + selClimVarName + curAnalog + "=Modpoll[x$position[" + curAnalog + ",]," +
                                     climVarIndex + "],"
```

```
                        }

                    }
                    curAnalog ++
                }
                    curAnalog = 1
                codeString += "\nwrite.csv(Recons" + selClimVarName + ",\"" + fileOutPath + fileOutName + "_" + selClimVarName + ".csv
                    \",quote=F,row.names=F)\n"
                }
            climVarIndex++
        }
        document.ReconForm.output.value += codeString
}

function MM_jumpMenu(targ,selObj,restore)
{ //v3.0
  eval(targ+".location='"+selObj.options[selObj.selectedIndex].value+"'");
  if (restore) selObj.selectedIndex=0;
}
//-->
</script>

<style type="text/css">
<!--
.style1 {
        color: #FF0000;
        font-weight: bold;
}
.style2 {
        color: #0000FF;
        font-weight: bold;
}
body {
        background-color: #CCCC99;
}
.style5 {color: #0000FF}
-->
</style>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></head>

<body onload="clearAll()">
<form action="" method="" name="ReconForm">
<h2>Generate MATTOOLS &quot;R&quot; Code for North American Pollen Reconstructions</h2>
<hr size="5" color="black">
<form method="POST" enctype="multipart/form-data" name="form" onsubmit="generateCode();">
  <p class="style1">NOTE: taxa in modern pollen DB file should match the taxa fossil pollen DB file exactly! </p>
  <p><strong>Select the modern pollen (Whitmore) DB file in CSV format</strong>: </p>
  <p>
    <input name="modFile" type="file" size="120">
  </p>
  <p>  </p>
  <p><strong>Select the fossil pollen DB file in CSV format: </strong></p>
  <p>
    <input name="fosFile" type="file" size="120">
  </p>
  <p> </p>
  <p><strong>File output name [e.g. PS134_RECONS_ + var name (TJAN)]</strong></p>
  <p>
    <input name="fileOutputName" type="text" id="fileOutputName" value="PS64_RECONS">
  </p>
<p> </p>
  <p><strong>Where to write output files (e.g. - H:\\NA_recons\\):</strong></p>
  <p>
    <input name="fileOutputPath" type="text" id="fileOutputPath" size="100" value="R:\\LPC\\LPC PAPERS\\NAM_2k\\db_files\\FINAL\\">
  </p>

  <p><strong>Specify column ranges for Modern/Fossil taxa and LAT/LON:</strong></p>
  <p class="style1">e.g. - Format is 4:5 to specify columns 4 and 5</p>
  <table width="307" border="0">
    <tr>
      <td width="93"> </td>
      <td width="97">Modern DB </td>
      <td width="95">Fossil DB </td>
    </tr>
    <tr>
```

```
      <td>LAT/LON:</td>
      <td><input name="mlatlon" type="text" id="mlatlon" value="10:11" size="15"></td>
      <td><input name="flatlon" type="text" id="flatlon" value="4:5" size="15"></td>
    </tr>
    <tr>
      <td>TAXA:</td>
      <td><input name="mtaxa" type="text" id="mtaxa" value="14:147" size="15"></td>
      <td><input name="ftaxa" type="text" id="ftaxa" value="16:149" size="15"></td>
    </tr>
  </table>
  <p>First column of climate variables in modern pollen file:
<br>
    <input name="climVarCol1" type="text" value="160" size="10"><br>
<strong>NOTE: In the original Whitmore file, variables are indexed from 160 to 228 (229 is a flag).
Check the file to make sure your starting column is 160, if not enter the new starting column.</strong>
If using one or more additional columns of modern climate data enter the starting column above (e.g. 230) and then
select TJAN if one extra column, TJAN and TFEB is two extra columns, etc. <u>Check your file (use R1C1 reference
style option in Excel under options -> general to get the correct column number.</u><br>
  </p>
  <p>Number of analogs
    <input name="numOfAnalogs" type="text" value="20" size="10">
  counts
  <input name="pollCounts" type="checkbox" value="checkbox">
  (do not check if DB files contain percentages)
</p>
  <p>Distance method
    <input name="distMethod" type="text" value="spherical" size="20">
</p>
  <p><strong>Climate variables to reconstruct :</strong></p>

  <p class="style2"><a href="javascript:selectAll();">Select all climate variables</a> | <a href="javascript:clearAll();">Clear
        selection of climate variables</a> |
<a href="javascript:generateCode();"><strong>Click here to generate code</strong></a></p>
  <p><em>
<Table><tr><td valign="bottom">
    <select name="climVarList" size="30" multiple id="climVarList">
    <!-- <select name="climVarList" size="20" multiple id="climVarList" onChange="MM_jumpMenu('parent',this,0)"> -->
<option value="TJAN" selected>160 TJAN</option>
<option value="TFEB" selected>161 TFEB</option>
<option value="TMAR" selected>162 TMAR</option>
<option value="TAPR" selected>163 TAPR</option>
<option value="TMAY" selected>164 TMAY</option>
<option value="TJUN" selected>165 TJUN</option>
<option value="TJUL" selected>166 TJUL</option>
<option value="TAUG" selected>167 TAUG</option>
<option value="TSEP" selected>168 TSEP</option>
<option value="TOCT" selected>169 TOCT</option>
<option value="TNOV" selected>170 TNOV</option>
<option value="TDEC" selected>171 TDEC</option>
<option value="PJAN" selected>172 PJAN</option>
<option value="PFEB" selected>173 PFEB</option>
<option value="PMAR" selected>174 PMAR</option>
<option value="PAPR" selected>175 PAPR</option>
<option value="PMAY" selected>176 PMAY</option>
<option value="PJUN" selected>177 PJUN</option>
<option value="PJUL" selected>178 PJUL</option>
<option value="PAUG" selected>179 PAUG</option>
<option value="PSEP" selected>180 PSEP</option>
<option value="POCT" selected>181 POCT</option>
<option value="PNOV" selected>182 PNOV</option>
<option value="PDEC" selected>183 PDEC</option>
<option value="SJAN" selected>184 SJAN</option>
<option value="SFEB" selected>185 SFEB</option>
<option value="SMAR" selected>186 SMAR</option>
<option value="SAPR" selected>187 SAPR</option>
<option value="SMAY" selected>188 SMAY</option>
<option value="SJUN" selected>189 SJUN</option>
<option value="SJUL" selected>190 SJUL</option>
<option value="SAUG" selected>191 SAUG</option>
<option value="SSEP" selected>192 SSEP</option>
<option value="SOCT" selected>193 SOCT</option>
<option value="SNOV" selected>194 SNOV</option>
<option value="SDEC" selected>195 SDEC</option>
<option value="TAVE" selected>196 TAVE</option>
<option value="TMAX" selected>197 TMAX</option>
<option value="TMIN" selected>198 TMIN</option>
```

```
<option value="GDD0" selected>199 GDD0</option>
<option value="GDD5" selected>200 GDD5</option>
<option value="CHILL" selected>201 CHILL</option>
<option value="MTCO" selected>202 MTCO</option>
<option value="MTWA" selected>203 MTWA</option>
<option value="MIPT" selected>204 MIPT</option>
<option value="AAETPT" selected>205 AAETPT</option>
<option value="APETPT" selected>206 APETPT</option>
<option value="MIPTEV" selected>207 MIPTEV</option>
<option value="AAETPTEV" selected>208 AAETPTEV</option>
<option value="APETPTEV" selected>209 APETPTEV</option>
<option value="MIPTDC" selected>210 MIPTDC</option>
<option value="AAETPTDC" selected>211 AAETPTDC</option>
<option value="APETPTDC" selected>212 APETPTDC</option>
<option value="TOTSNOPT" selected>213 TOTSNOPT</option>
<option value="ANNP" selected>214 ANNP</option>
<option value="PJANPANN" selected>215 PJANPANN</option>
<option value="PJULPANN" selected>216 PJULPANN</option>
<option value="TMAM" selected>217 TMAM</option>
<option value="TJJA" selected>218 TJJA</option>
<option value="TSON" selected>219 TSON</option>
<option value="TDJF" selected>220 TDJF</option>
<option value="PMAM" selected>221 PMAM</option>
<option value="PJJA" selected>222 PJJA</option>
<option value="PSON" selected>223 PSON</option>
<option value="PDJF" selected>224 PDJF</option>
<option value="SMAM" selected>225 SMAM</option>
<option value="SJJA" selected>226 SJJA</option>
<option value="SSON" selected>227 SSON</option>
<option value="SDJF" selected>228 SDJF</option>
</select>
 </em><!--<strong>select all</strong><em> <input name="checkbox" type="checkbox" value="checkbox" checked>-->
 </em></p></td><td>
  <p><strong>The final code to run in R is shown below:</strong> <br>
      <textarea name="output" cols="100" rows="30" wrap="wrap" id="output"></textarea></td></tr>
<tr><td></td><td><input type="reset" value="CLEAR CODE WINDOW"></td></tr>
   </table><br>
   </p>
</form>
</body>
</html>
```

D.9 Example of Climate Data Operator (CDO) commands to process a NetCDF data file

```
UCAR PDSI netcdf file has 135 yrs (1870-2005) therefore 136 timesteps x 12 months = 1632 total timesteps

Use Excel formulas to create a list of timesteps needed.

For annual average of PDSI from 1961-1990, timesteps are 1093 to 1452, cdo code is as follows:

cdo seltimestep,1093/1452 pdsi_ucar.nc pdsi_out.nc

For July 1961-1990, cdo code is as follows:
cdo seltimestep,1099,1111,1123,1135,1147,1159,1171,1183,1195,1207,1219,1231,1243,1255,1267,1279,1291,
1303,1315,1327,1339,1351,1363,1375,1387,1399,1411,1423,1435,1447 pdsi_ucar.nc pdsi_30yrs.nc

We now have 30 timesteps of July so now simply average the timesteps:
cdo yearmean pdsi_30yrs.nc pdsi_30yrmean.nc

For July 1950-1979, cdo code is as follows:
cdo seltimestep,31,43,55,67,79,91,103,115,127,139,151,163,175,187,199,211,223,235,247,259,271,283,295,307,319,331,343,355,367,379 ncep.
      air.mon.mean.nc ncep_july_avgtemp.nc
```

D.10 Overview of data-model comparison methods

There has been an ongoing effort in paleoclimate data-model comparisons to improve our ability to estimate future climate: Cheddadi *et al.* (1996), Bartlein *et al.* (1998), Prentice *et al.* (1998), Webb *et al.* (1998), Kageyama *et al.* (2001), Sawada *et al.* (2004), Brewer *et al.* (2006), Brewer *et al.* (2007) and Otto-Bliesner *et al.* (2009). The premise is that if climate models can reproduce climate patterns for both modern and past conditions, we will have higher confidence in their future predictions of climate (Gajewski, 2008). It is suggested in Bartlein *et al.* (2011) to focus on GDD5, mean temperature of the coldest month of the year (MTCO) and mean annual precipitation (MAP) for data-model comparisons.

One method is to visually compare the maps for agreement/disagreement, but this is a more qualitative approach (Gajewski, 2008). In section 2.4 of Sawada *et al.* (2004), both qualitative and quantitative methods are used. The main methods are point to point (Pinot *et al.* 1999; Texier *et al.* 1997) or grid to grid (Webb *et al.* 1987; Liao *et al.* 1994; Vettoretti *et al.* 2000). In Bonfils *et al.* (2004) an innovative cluster analysis method is used, where climate reconstruction output is clustered and compared to climate model output.

The main problem with data-model comparisons is that the gridded output from these models is different from GIS output. However, with file formats such as NetCDF and GIS tools to convert the NetCDF format to rasters or point shapefiles, the data-model comparisons are more feasible now, but there is also the question of how to do the statistical validation (i.e. to determine uncertainty) for the data-model comparison.

Section 2.4 of Sawada *et al.* (2004) describes how the uncertainty estimates of model output are computed; however, it is more difficult to calculate uncertainty for reconstruction output. In addition, there are areas of extrapolations and hence uncertainty when comparing the reconstructions to the model output (Edwards *et al.*, 2007; Sawada *et al.*, 2004). Therefore it is suggested that data-model comparisons be confined to discrete sites (i.e. implying a point-to-point comparison) as interpolated areas can be data-sparse (Broccoli and Marciniak, 1996).

Appendix E

Supplementary Plots, Maps & Tables

Table 24: Calendar year BP dates tuned to BCE/CE dates.

Cal Yr BP	2500	2400	2300	2200	2100	2000	1900	1800	1700
Tuned CE	-150	-50	50	150	250	350	450	550	650
Cal Yr BP	1600	1500	1400	1300	1200	1100	1000	900	800
Tuned CE	750	850	950	1050	1150	1250	1350	1450	1550
Cal Yr BP	700	600	500	400	300	200	100	0	
Tuned CE	1600	1650	1700	1750	1800	1850	1900	1950	

Table 25: Number and percentage of sites retained after the spatial filtering procedure is applied. The average, median, min and max are for 0-2500 BP at 100 year time steps.

lagKM	ALLbp	0	500	1000	1500	2000	2500	avg	med	min	max
50	11272	348	451	474	448	452	442	451	453	439	479
%	0.76	0.76	0.77	0.81	0.78	0.79	0.78	0.79	0.79	0.76	0.82
100	9942	281	375	408	369	388	374	382	386	351	423
%	0.67	0.62	0.64	0.7	0.64	0.68	0.66	0.67	0.68	0.62	0.73
150	8649	245	320	361	332	346	314	333	337	286	369
%	0.58	0.54	0.55	0.62	0.58	0.61	0.55	0.58	0.58	0.49	0.63
200	7891	232	302	333	267	313	287	304	312	263	335
%	0.53	0.51	0.52	0.57	0.46	0.55	0.51	0.53	0.54	0.45	0.57
250	7119	199	272	310	243	286	268	274	285	237	313
%	0.48	0.44	0.47	0.53	0.42	0.5	0.47	0.48	0.50	0.40	0.54
300	6496	188	251	296	217	211	249	250	250	209	304
%	0.44	0.41	0.43	0.51	0.38	0.37	0.44	0.44	0.44	0.36	0.52
350	5959	182	235	273	193	192	223	229	228	187	290
%	0.4	0.4	0.4	0.47	0.34	0.34	0.39	0.40	0.40	0.32	0.49
400	5628	176	229	253	181	183	209	216	215	174	272
%	0.38	0.39	0.39	0.43	0.31	0.32	0.37	0.38	0.37	0.30	0.46
450	5262	167	212	239	164	175	197	202	200	162	255
%	0.35	0.37	0.36	0.41	0.28	0.31	0.35	0.35	0.35	0.28	0.43
500	4935	164	203	225	151	152	185	190	187	151	245
%	0.33	0.36	0.35	0.38	0.26	0.27	0.33	0.33	0.33	0.26	0.42

Key for Whitmore climate variables in reconstruction plots (Figure 31, Figure 32, Figure 33 which follow - order is left to right, top to bottom, starting with the top row of each panel):

AAETPT: Actual evapotranspiration (AE)
AAETPTDC: Annual actual evapotraspiration for deciduous assimilation period (when pseudo-daily temperatures are greater than 5C) [calculated using Cramer & Prentice approach]
AAETPTEV: AE calculated for the deciduous pft assimilation period
APETPT: Potential evapotranspiration (PE)
APETPTDC: Annual potential evapotraspiration for deciduous assimilation period (when pseudo-daily temperatures are greater than 5C) [calculated using Cramer & Prentice approach]
APETPTEV: PE calculated for the deciduous pft assimilation period
GDD0: Growing-degree days (0C base)
GDD5: Growing-degree days (5C base)
PJANPANN: January/Annual precipitation ratio
MIPT: Priestley-Taylor (alpha) parameter (AE/PE)
MIPTDC: Annual Priestly-Taylor moisture index (alpha) for deciduous assimilation period (when pseudo-daily temperatures are greater than 5C) [calculated using Cramer & Prentice approach]
MIPTEV: Alpha calculated for the evergreen pft assimilation period (days when the pseudo-daily temperature > -4C
MTWA: Mean temperature (in C) of the warmest month of the year
MTCO: Mean temperature (in C) of the coldest month of the year
ANNP: Total annual precipitation (in mm)
CHILL: Chilling requirement (number of days when pseudo-daily temperature was below 5C)
PJULPANN: July/Annual precipitation ratio
TOTSNOPT: Total snow-water equivalent from bucket model (computed)
SJJA: Mean sunshine (%) for June, July, August
SDJF: Mean sunshine (%) for December, January, February
SMAM: Mean sunshine (%) for March, April, May
SSON: Mean sunshine (%) for September, October, November
SJAN: Mean monthly percent possible sunshine for January
SFEB: Mean monthly percent possible sunshine for February
SMAR: Mean monthly percent possible sunshine for March
SAPR: Mean monthly percent possible sunshine for April
SMAY: Mean monthly percent possible sunshine for May
SJUN: Mean monthly percent possible sunshine for June
SJUL: Mean monthly percent possible sunshine for July
SAUG: Mean monthly percent possible sunshine for August
SSEP: Mean monthly percent possible sunshine for September
SOCT: Mean monthly percent possible sunshine for October
SNOV: Mean monthly percent possible sunshine for November
SDEC: Mean monthly percent possible sunshine for December
PDJF: Mean precipitation (in mm) for December, January, February
PMAM: Mean precipitation (in mm) for March, April, May
PJJA: Mean precipitation (in mm) for June, July, August
PSON: Mean precipitation (in mm) for September, October, November
PJAN: Mean monthly precipitation for January in mm
PFEB: Mean monthly precipitation for February in mm
PMAR: Mean monthly precipitation for March in mm
PAPR: Mean monthly precipitation for April in mm
PMAY: Mean monthly precipitation for May in mm
PJUN: Mean monthly precipitation for June in mm
PJUL: Mean monthly precipitation for July in mm
PAUG: Mean monthly precipitation for August in mm
PSEP: Mean monthly precipitation for September in mm
POCT: Mean monthly precipitation for October in mm
PNOV: Mean monthly precipitation for November in mm
PDEC: Mean monthly precipitation for December in mm
TDJF : Mean temperature (in C) for December, January, February
TMAM: Mean temperature (in C) for March, April, May
TJJA: Mean temperature (in C) for June, July, August
TSON: Mean temperature (in C) for September, October, November
TJAN: Mean monthly temperature for January in C
TFEB: Mean monthly temperature for February in C
TMAR: Mean monthly temperature for March in C
TAPR: Mean monthly temperature for April in C
TMAY: Mean monthly temperature for May in C
TJUN: Mean monthly temperature for June in C
TJUL: Mean monthly temperature for July in C
TAUG: Mean monthly temperature for August in C
TSEP: Mean monthly temperature for September in C
TOCT: Mean monthly temperature for October in C
TNOV: Mean monthly temperature for November in C
TDEC: Mean monthly temperature for December in C
TMAX: Absolute maximum temperature in C
TMIN: Absolute minimum temperature in C
TAVE: Average annual temperature in C (see p.243 for plot)

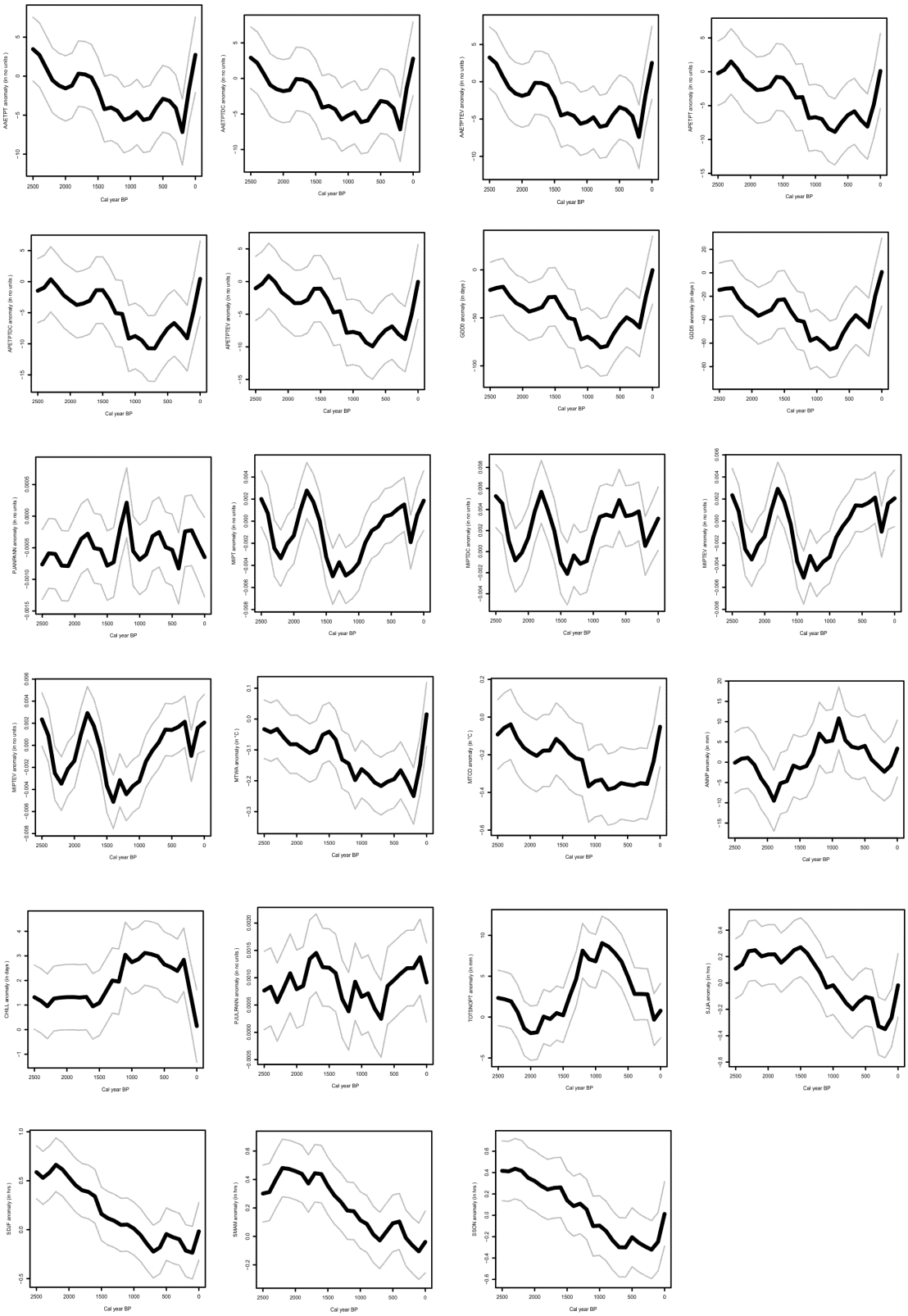


Figure 31: Reconstruction of Whitmore climate variables (Panel 1)

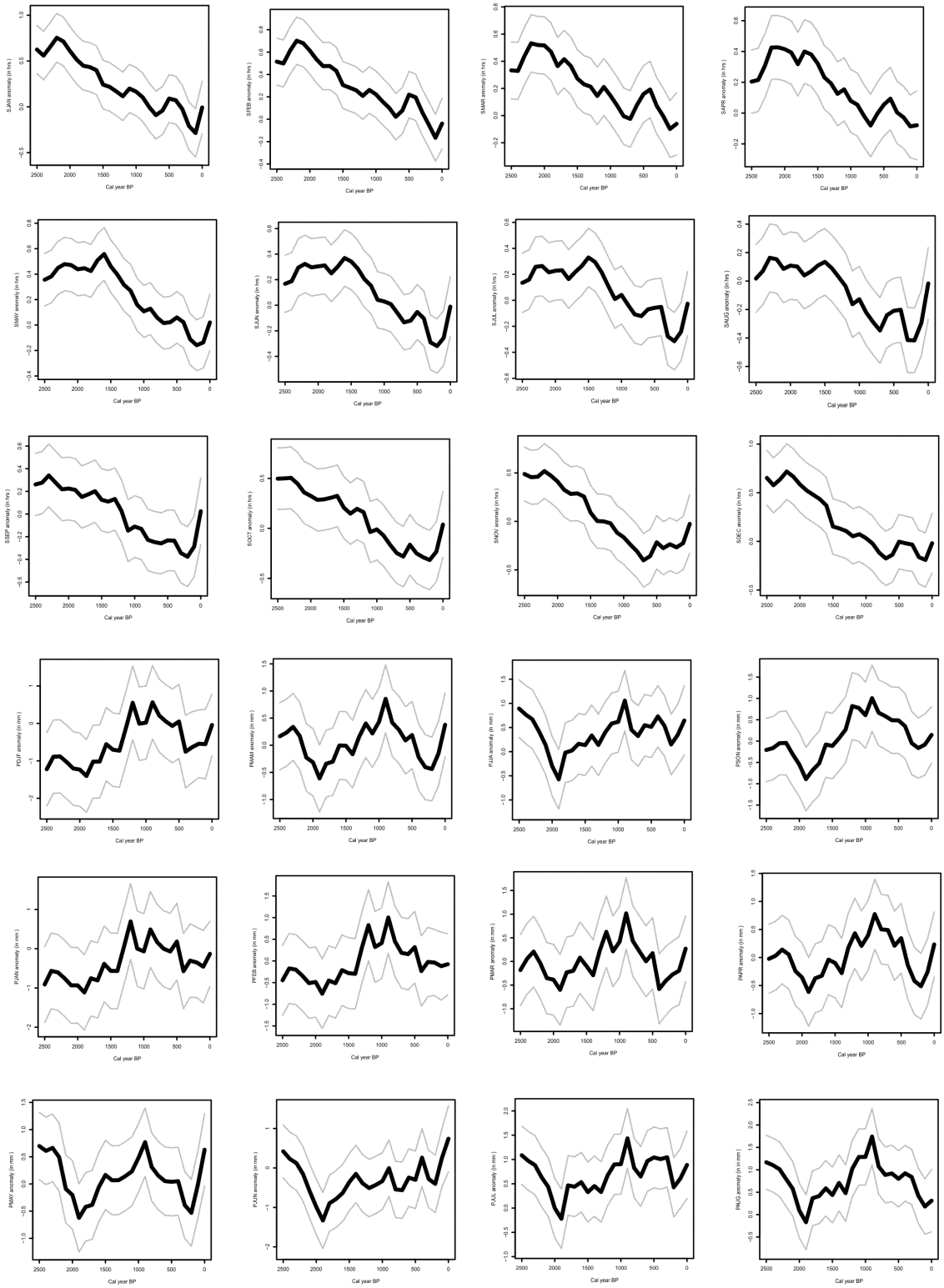


Figure 32: Reconstruction of Whitmore climate variables (Panel 2)

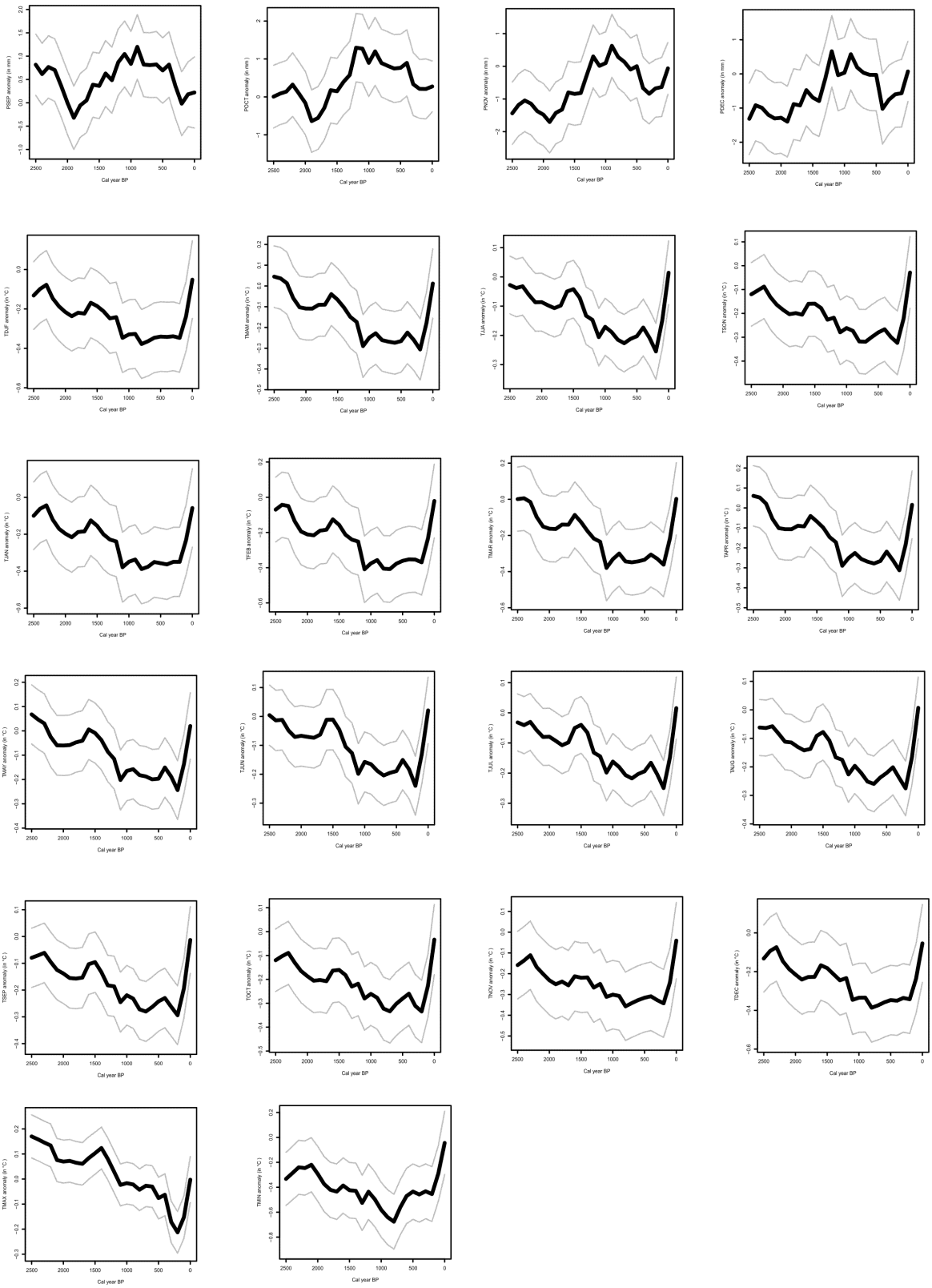


Figure 33: Reconstruction of Whitmore climate variables (Panel 3)

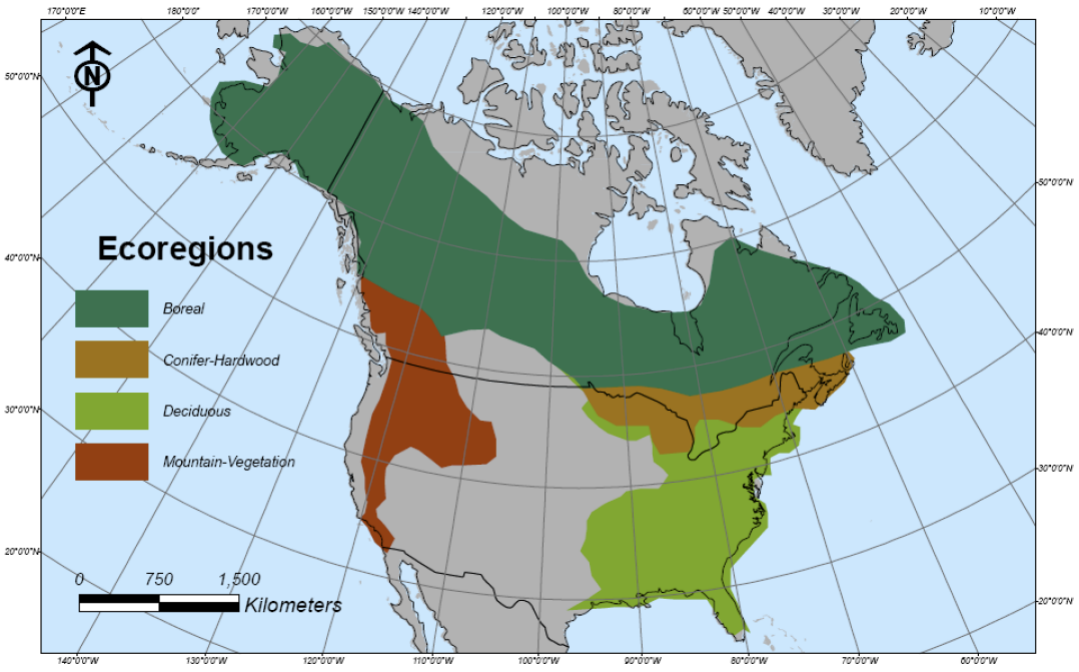


Figure 34: Map of four major ecoregions of North America (adapted from Fedorova *et al.*, 1994).

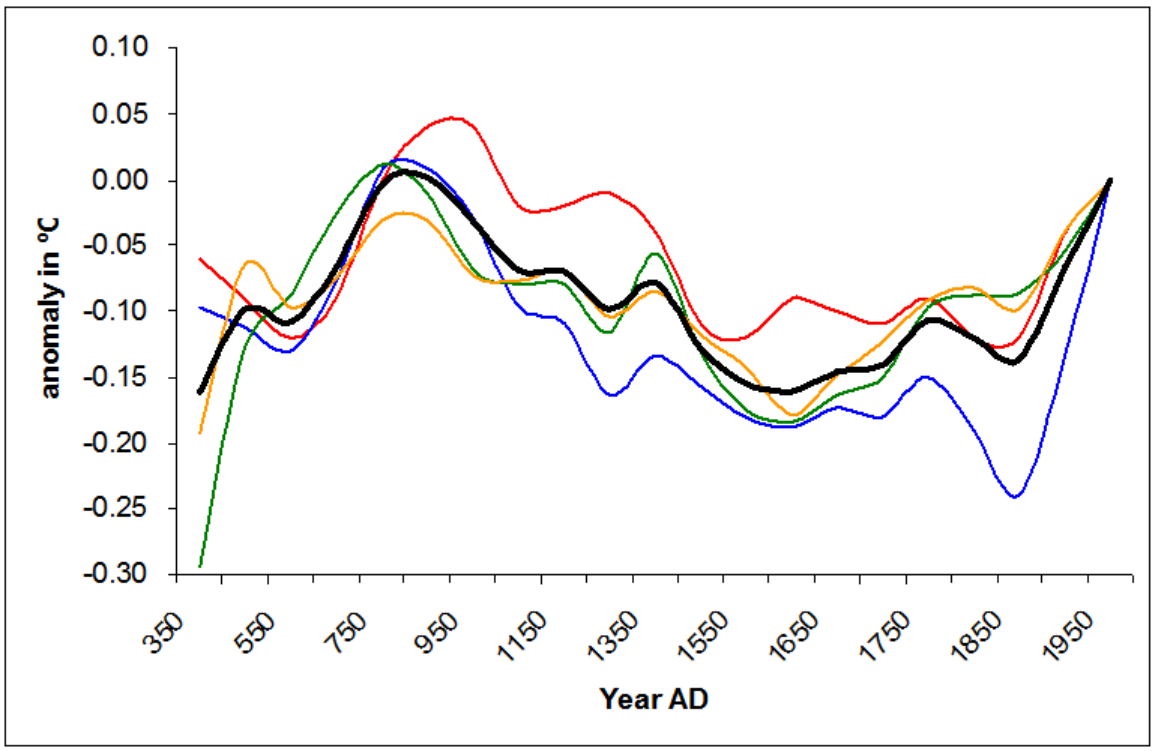


Figure 35: Reconstruction of MTWA (anomalies in °C) using 35 (red), 64 (blue), 134 (green), 135 (gold) pollen sums and average of all 4 pollen sums (black).

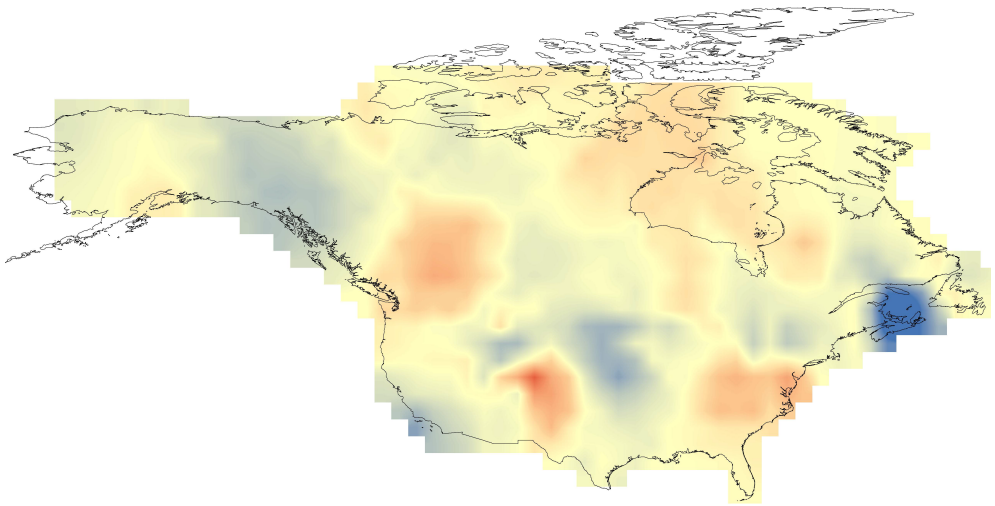


Figure 36: MTWA reconstructed anomaly map at 100 BP (range of anomalies is +/- 2°C).

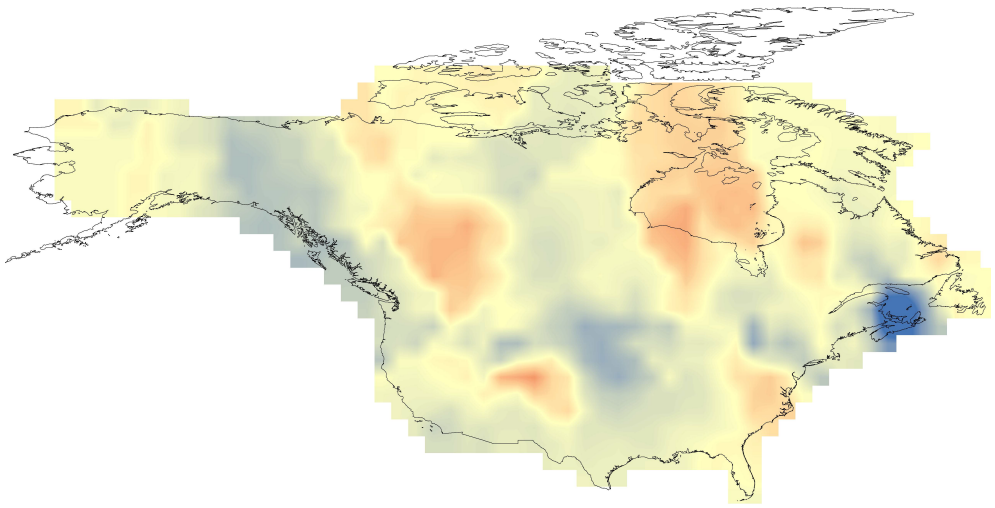


Figure 37: MTWA reconstructed anomaly map at 200 BP (range of anomalies is +/- 2°C).

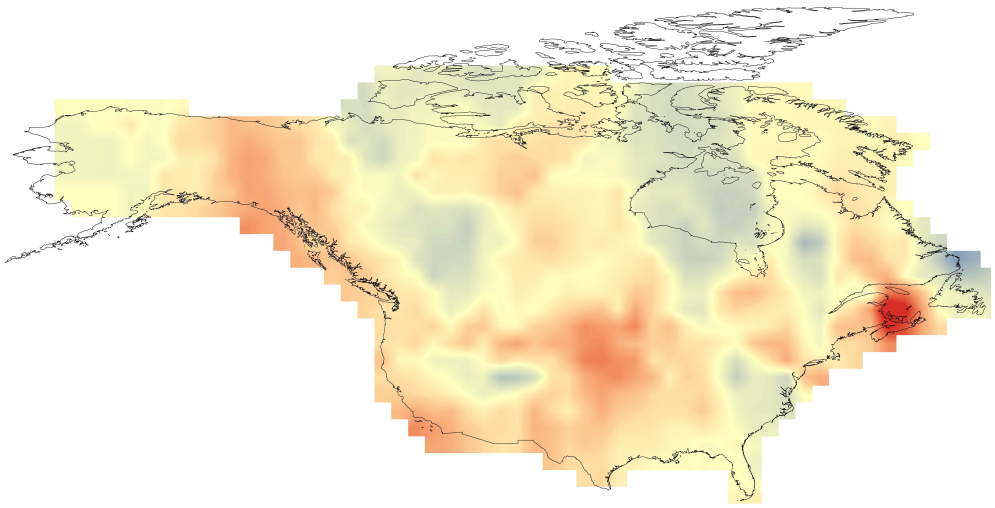


Figure 38: MTWA reconstructed anomaly map at 300 BP (range of anomalies is +/- 2°C).

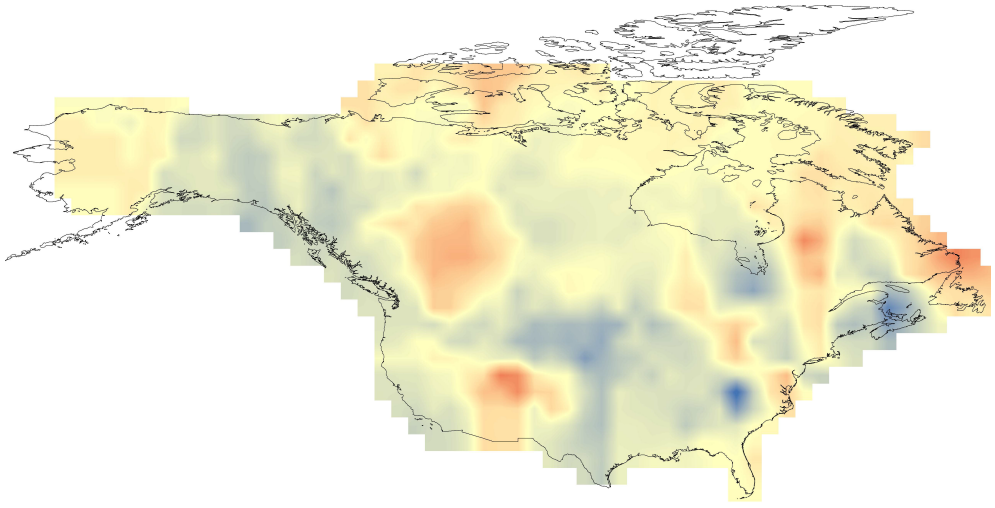


Figure 39: MTWA reconstructed anomaly map at 400 BP (range of anomalies is +/- 2°C).

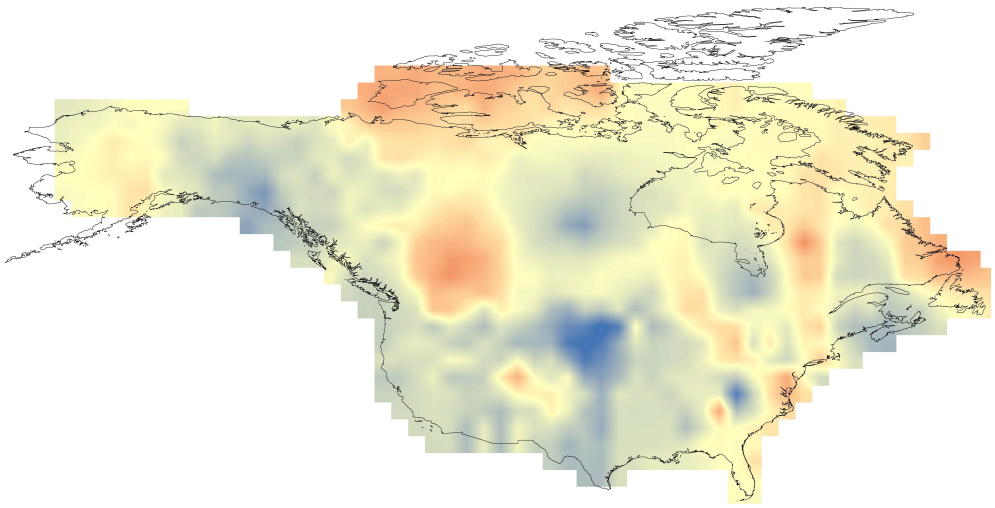


Figure 40: MTWA reconstructed anomaly map at 500 BP (range of anomalies is +/- 2°C).

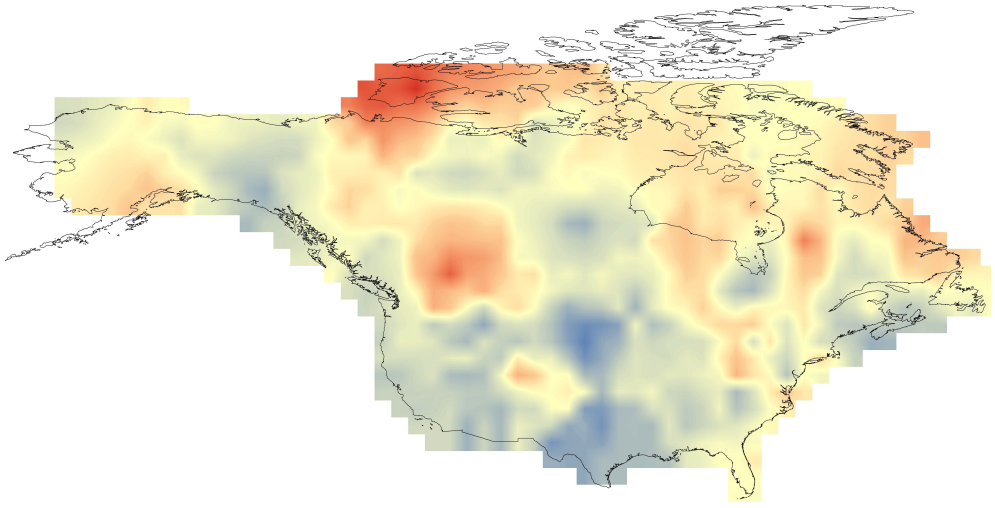


Figure 41: MTWA reconstructed anomaly map at 600 BP (range of anomalies is +/- 2°C).

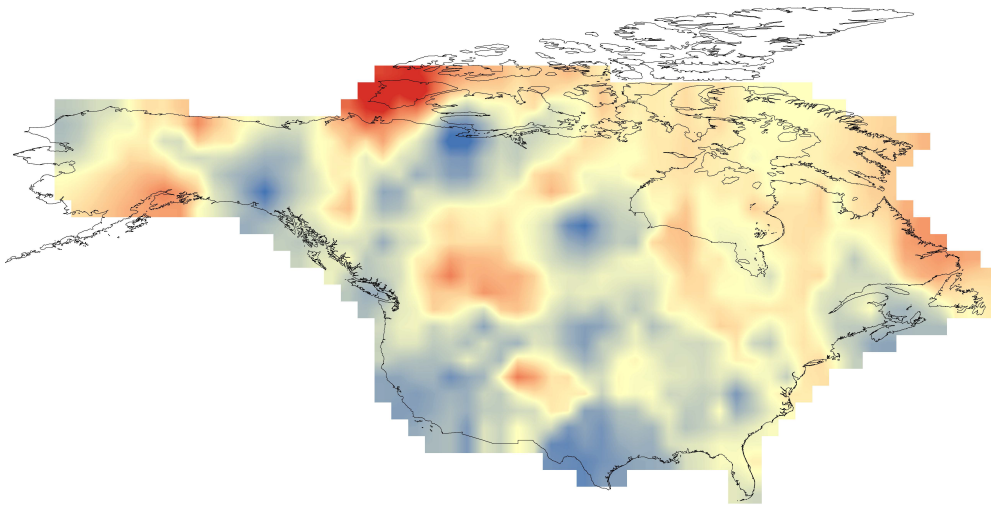


Figure 42: MTWA reconstructed anomaly map at 700 BP (range of anomalies is +/- 2°C).

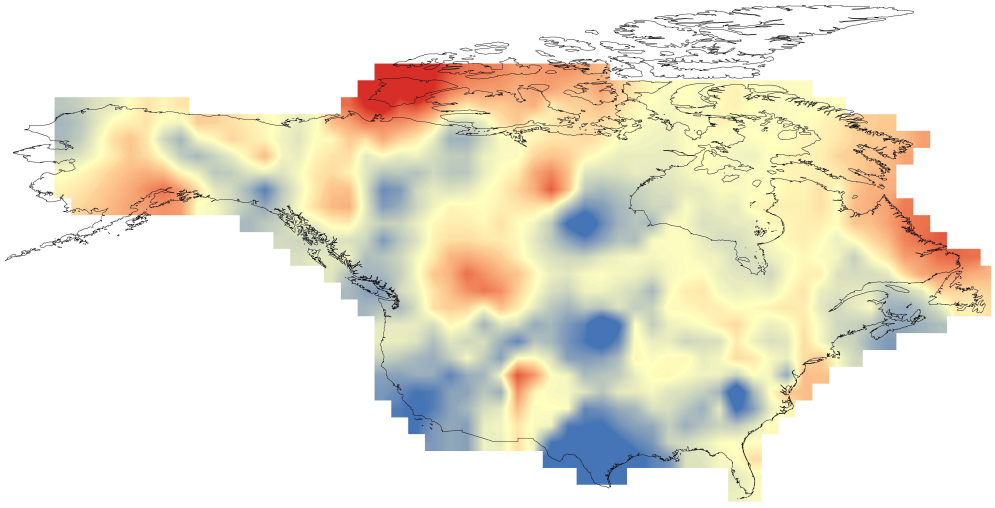


Figure 43: MTWA reconstructed anomaly map at 800 BP (range of anomalies is +/- 2°C).

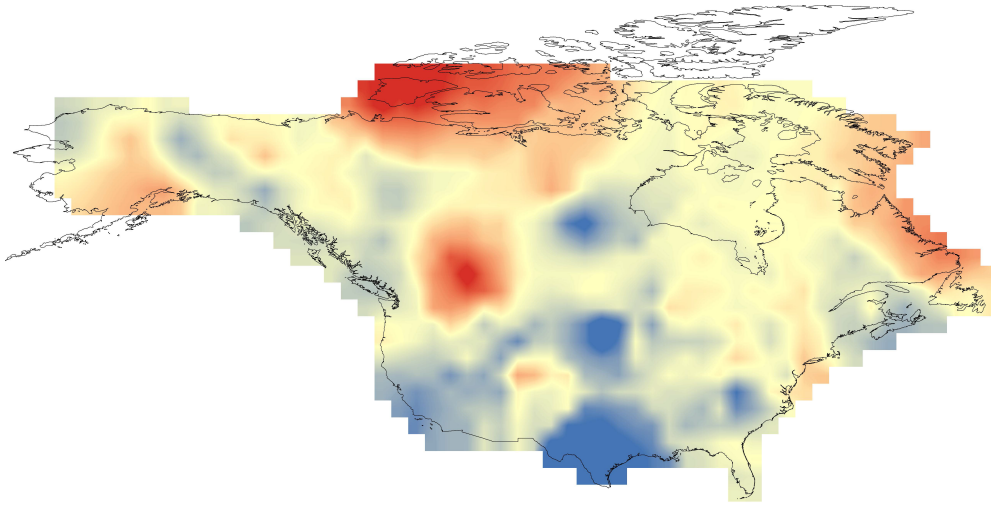


Figure 44: MTWA reconstructed anomaly map at 900 BP (range of anomalies is +/- 2°C).

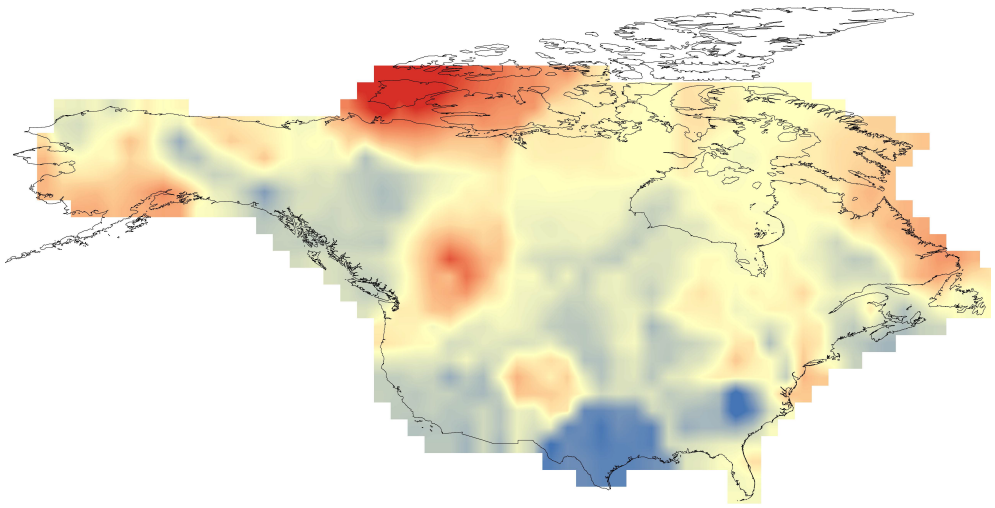


Figure 45: MTWA reconstructed anomaly map at 1000 BP (range of anomalies is +/- 2°C).

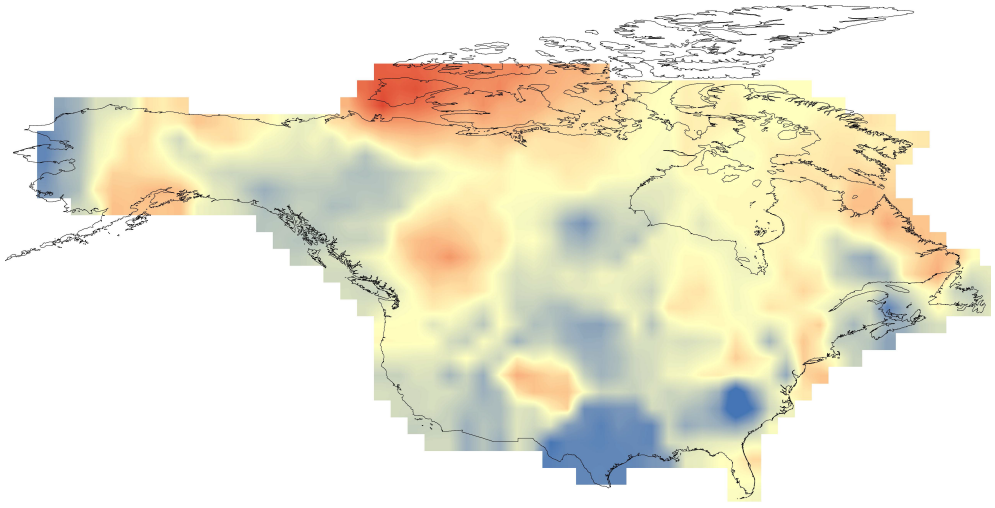


Figure 46: MTWA reconstructed anomaly map at 1100 BP (range of anomalies is +/- 2°C).

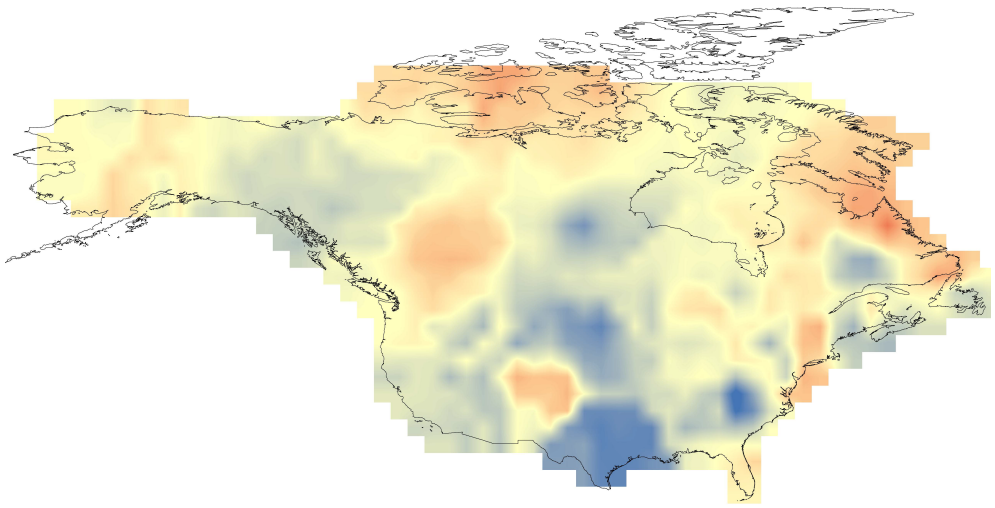


Figure 47: MTWA reconstructed anomaly map at 1200 BP (range of anomalies is +/- 2°C).

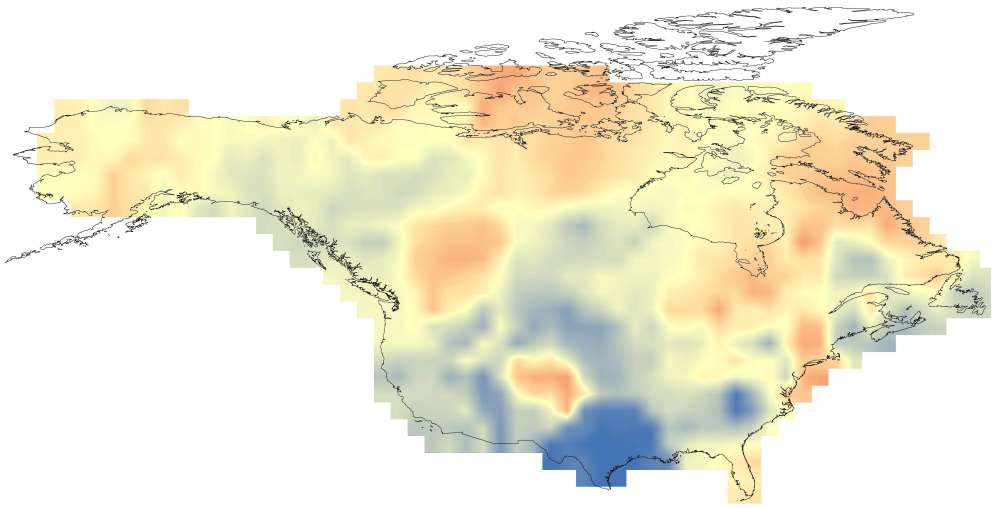


Figure 48: MTWA reconstructed anomaly map at 1300 BP (range of anomalies is +/- 2°C).

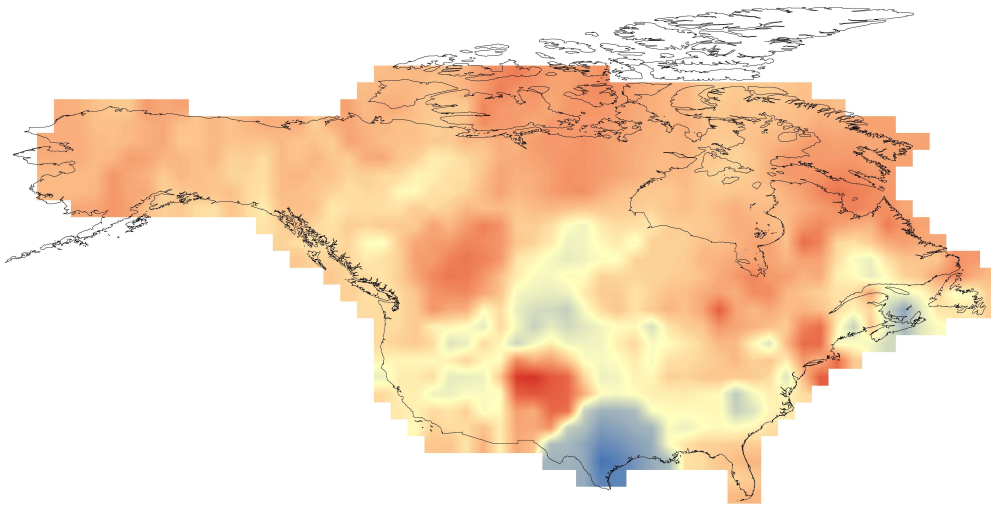


Figure 49: MTWA reconstructed anomaly map at 1400 BP (range of anomalies is +/- 2°C).

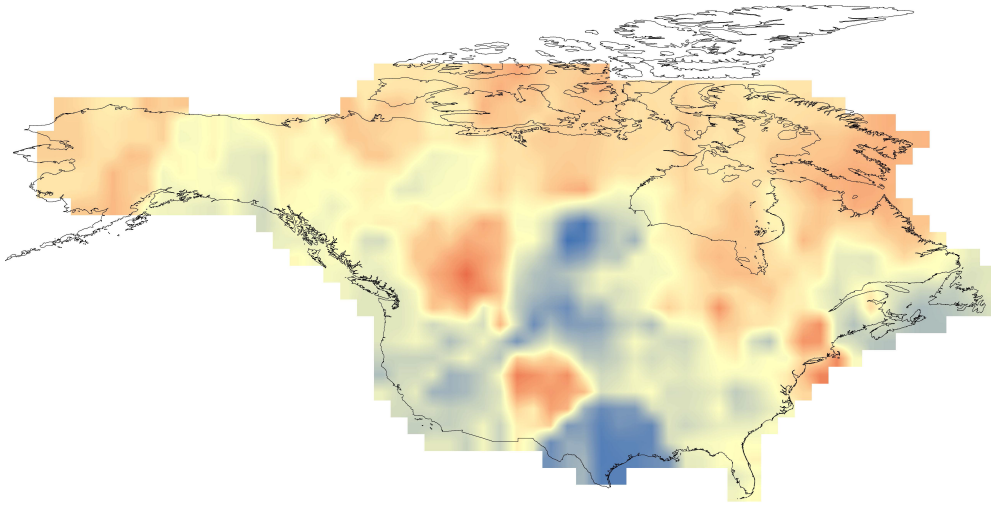


Figure 50: MTWA reconstructed anomaly map at 1500 BP (range of anomalies is +/- 2°C).

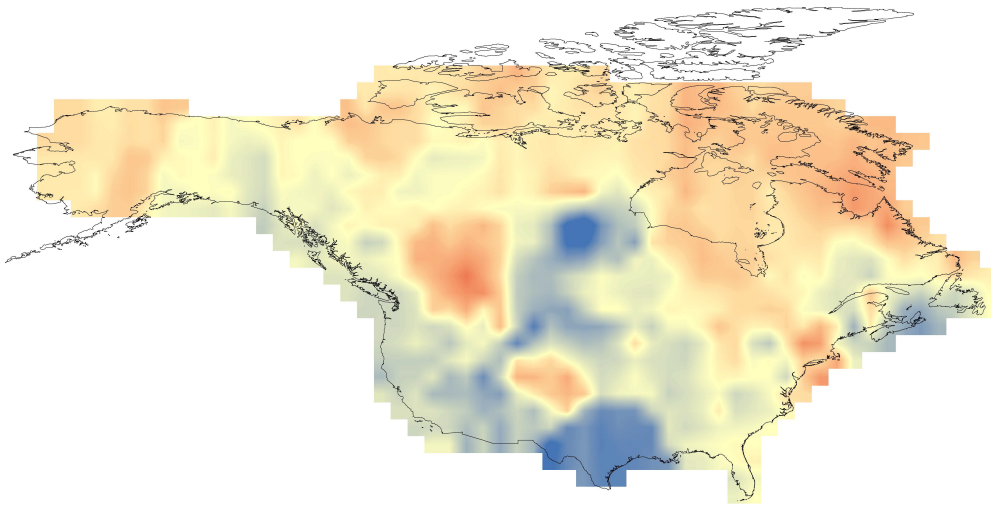


Figure 51: MTWA reconstructed anomaly map at 1600 BP (range of anomalies is +/- 2°C).

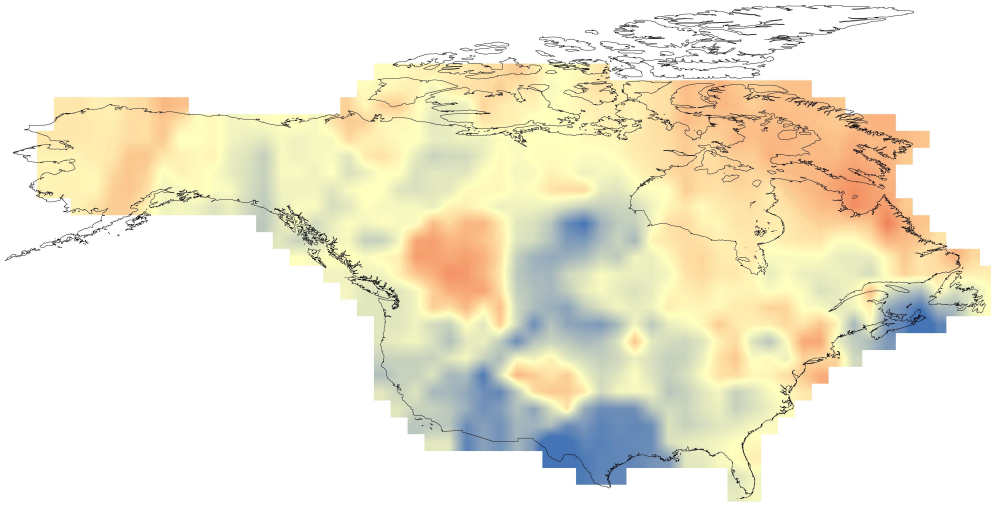


Figure 52: MTWA reconstructed anomaly map at 1700 BP (range of anomalies is +/- 2°C).

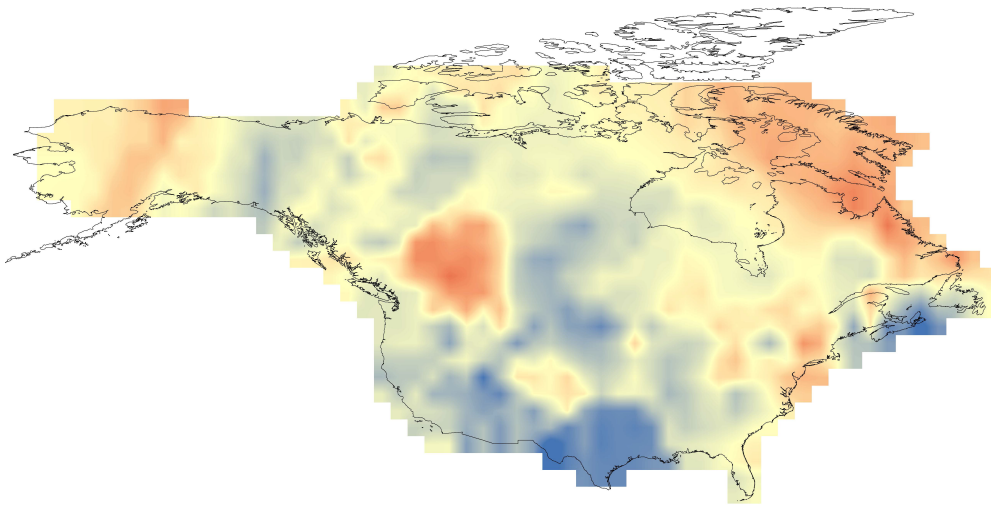


Figure 53: MTWA reconstructed anomaly map at 1800 BP (range of anomalies is +/- 2°C).

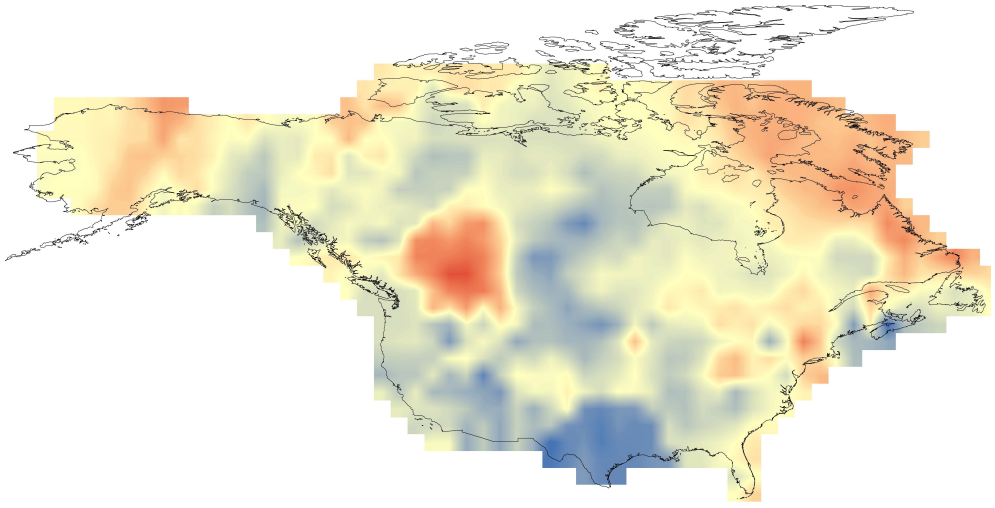


Figure 54: MTWA reconstructed anomaly map at 1900 BP (range of anomalies is +/- 2°C).

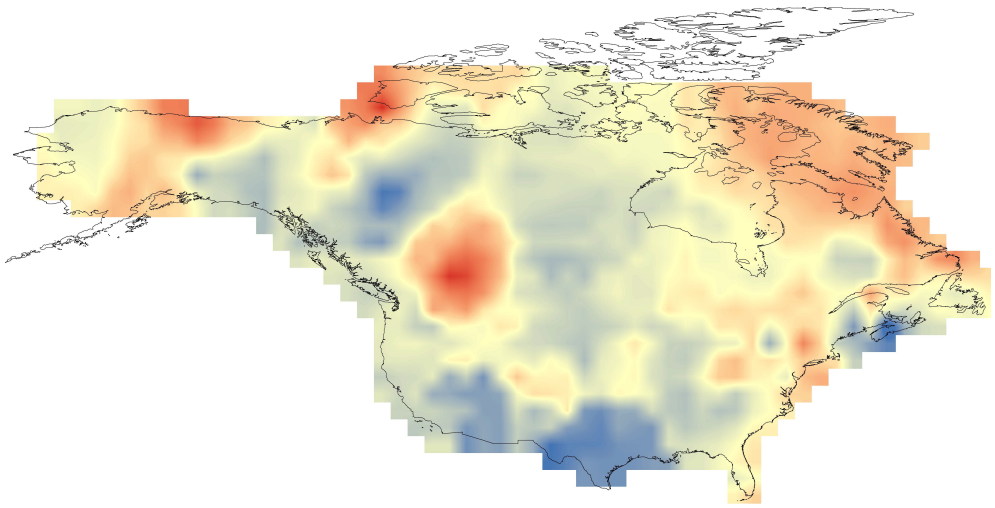


Figure 55: MTWA reconstructed anomaly map at 2000 BP (range of anomalies is +/- 2°C).

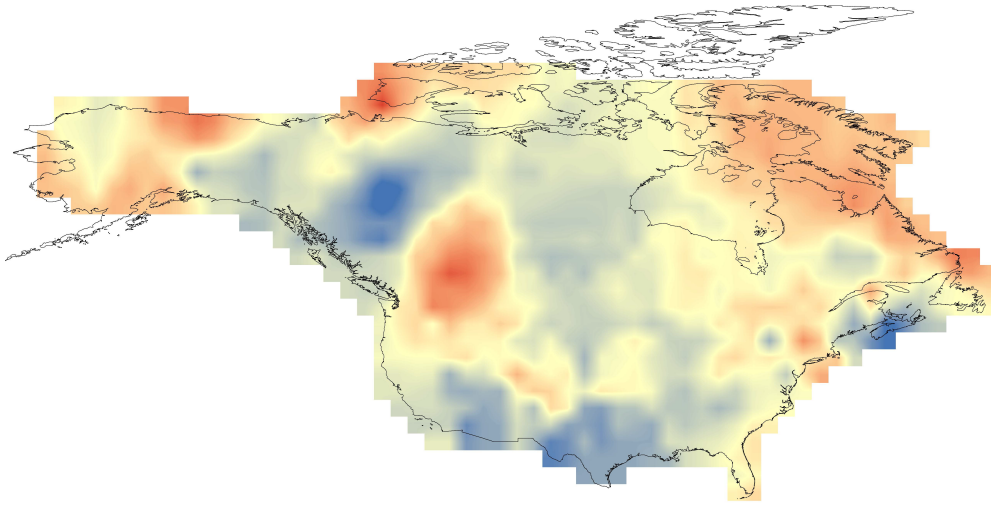


Figure 56: MTWA reconstructed anomaly map at 2100 BP (range of anomalies is +/- 2°C).

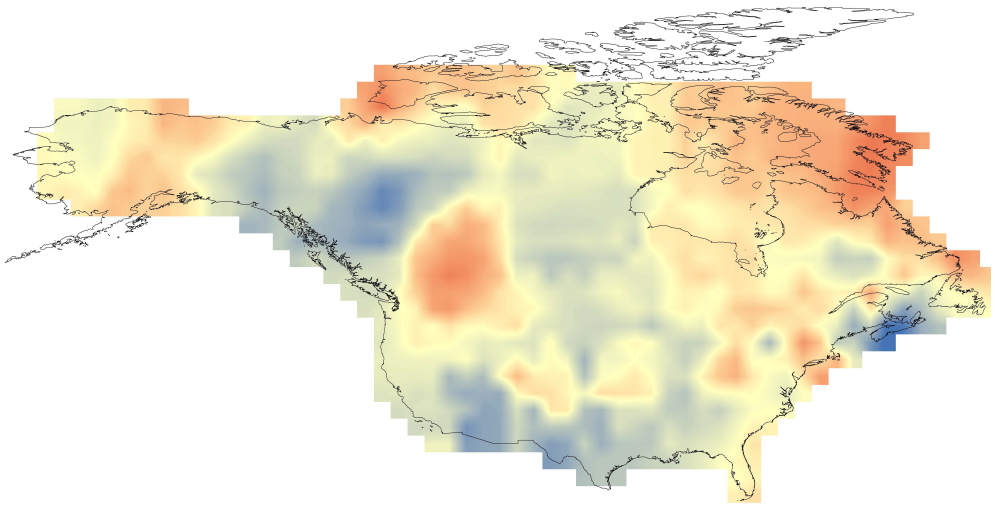


Figure 57: MTWA reconstructed anomaly map at 2200 BP (range of anomalies is +/- 2°C).

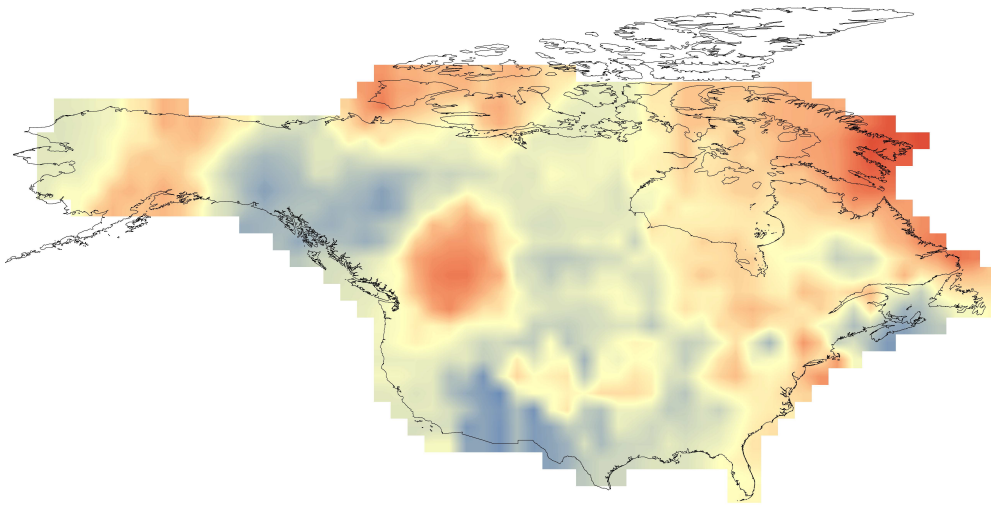


Figure 58: MTWA reconstructed anomaly map at 2300 BP (range of anomalies is +/- 2°C).

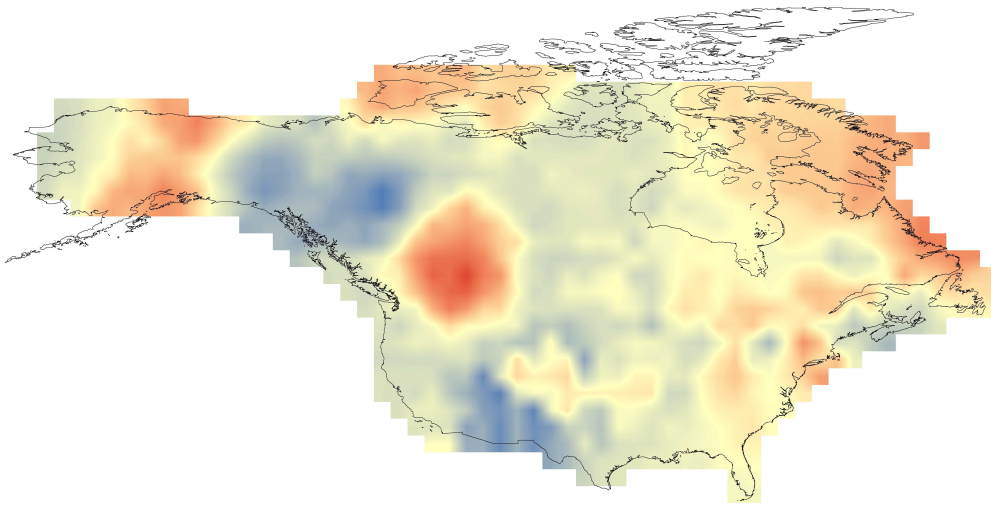


Figure 59: MTWA reconstructed anomaly map at 2400 BP (range of anomalies is +/- 2°C).

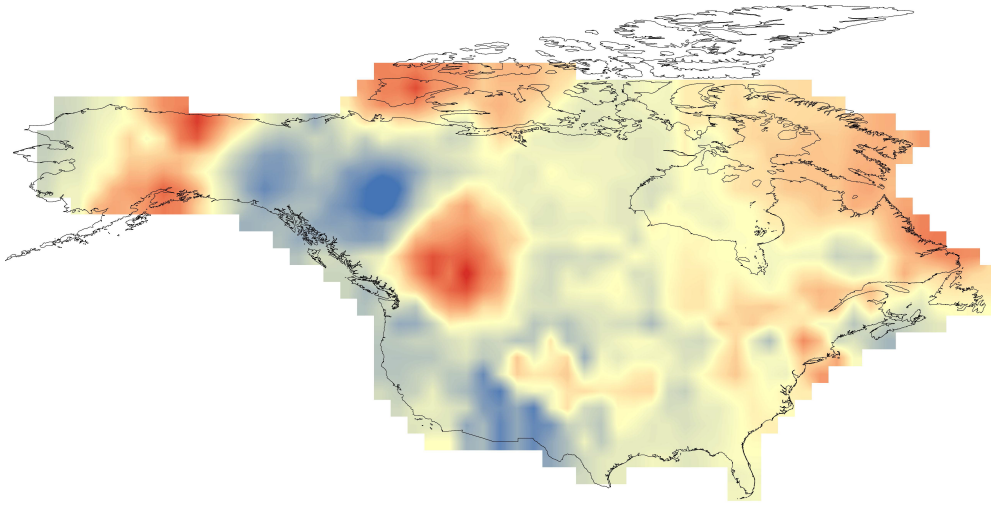


Figure 60: MTWA reconstructed anomaly map at 2500 BP (range of anomalies is +/- 2°C).

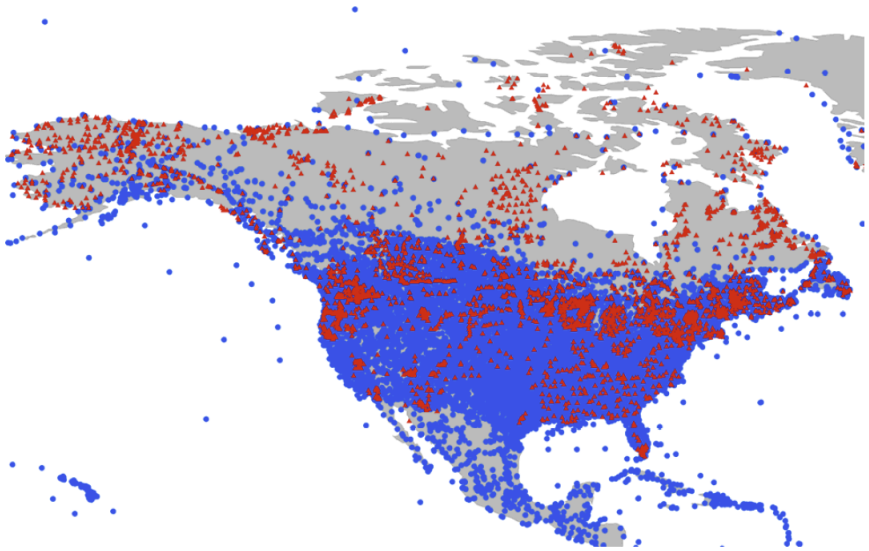


Figure 61: Location of BEST climate stations in North America (blue) and location of modern pollen sites (red).

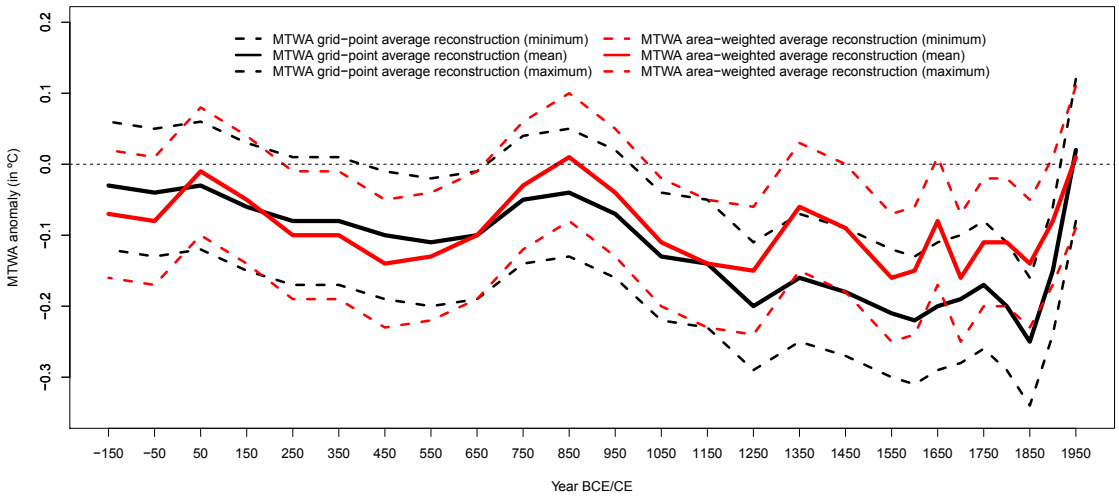


Figure 62: Minimum, mean and maximum of grid point average (black) versus area-weighted average (red) MTWA/TJUL anomalies through time.

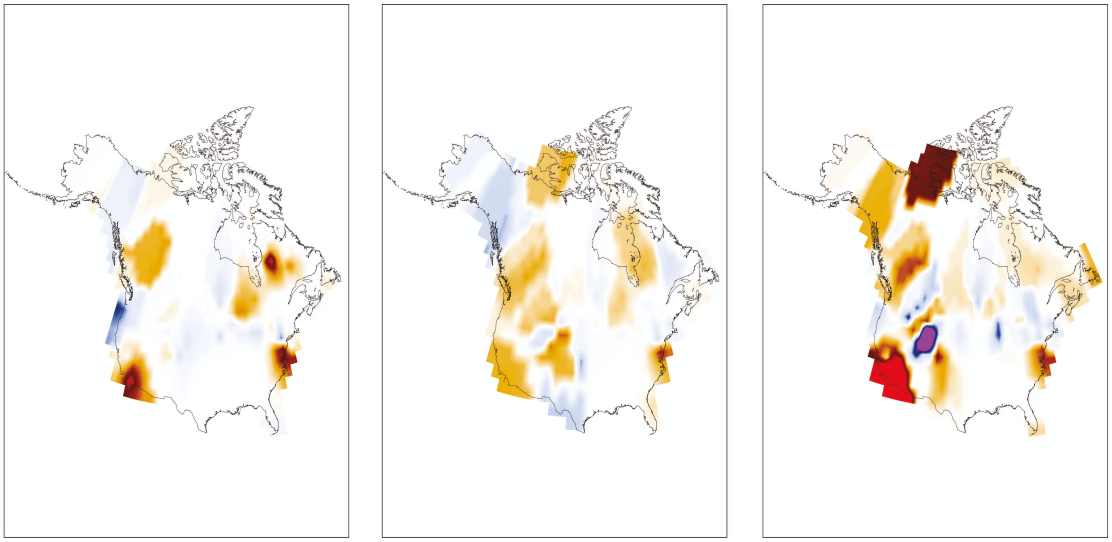


Figure 63: Reconstructed anomaly map of MTWA/TJUL at 0 BP (1950 CE) using modern TJUL data sets with highest (NARR) (A) versus lowest (NCEP1) (C) standard deviations. Whitmore (B) is close to the median standard deviation of all data sets.

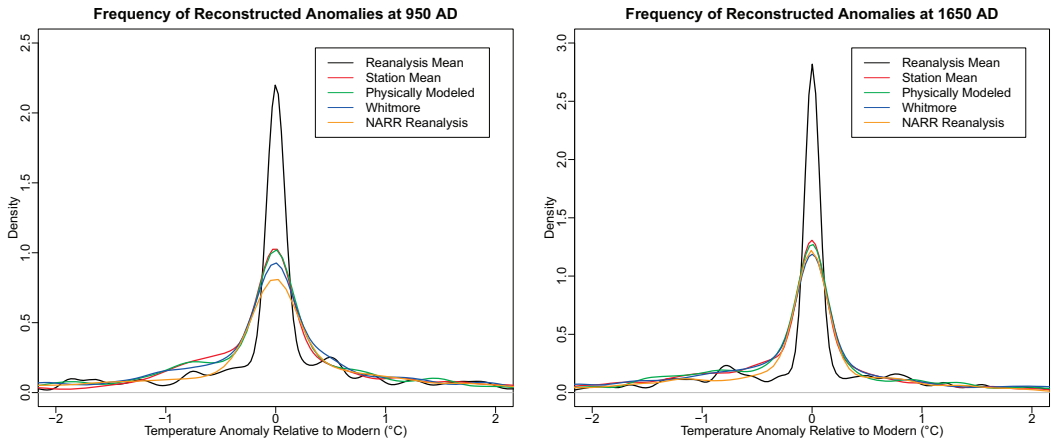


Figure 64: Frequency of reconstructed MTWA/TJUL anomalies at 950 and 1650 CE (courtesy of R.G. Way).

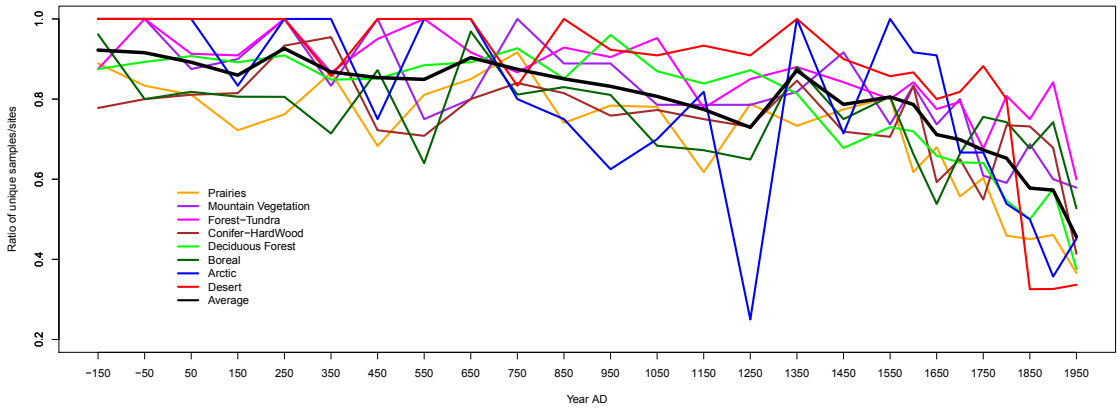


Figure 65: Ratio of unique fossil pollen sites divided by fossil pollen samples through time (by ecoregion).

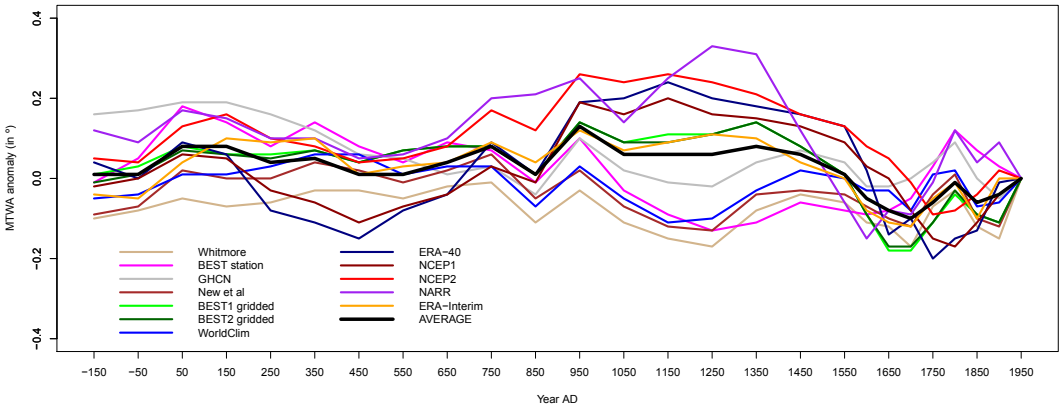


Figure 66: Reconstructions of MTWA/TJUL anomalies (in °C) for the last 2,000 years using all modern data sets (composite of sites in boreal forest region only).

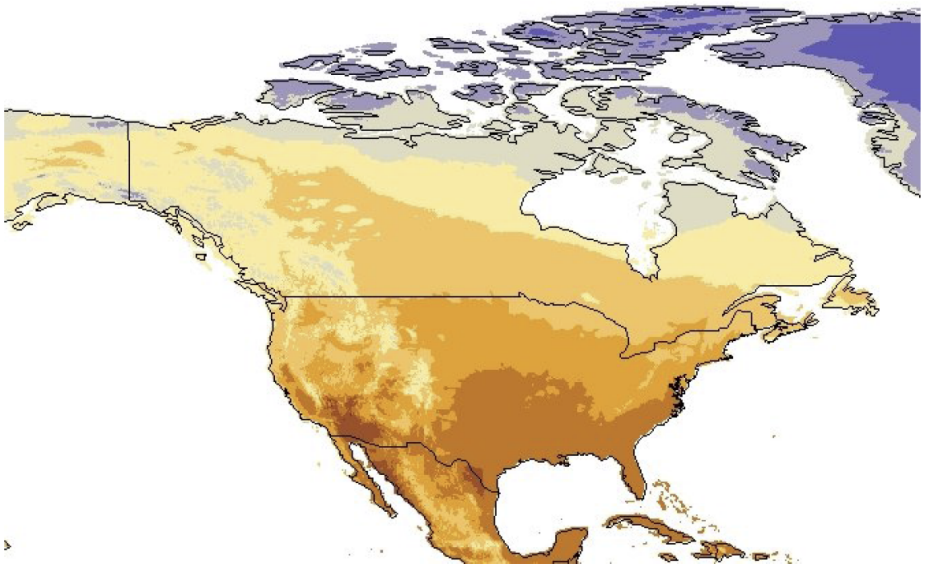


Figure 67: Long term mean July temperature map (1950 to 2000 CE) from WorldClim data (TMEAN7 raster layer) for North America.

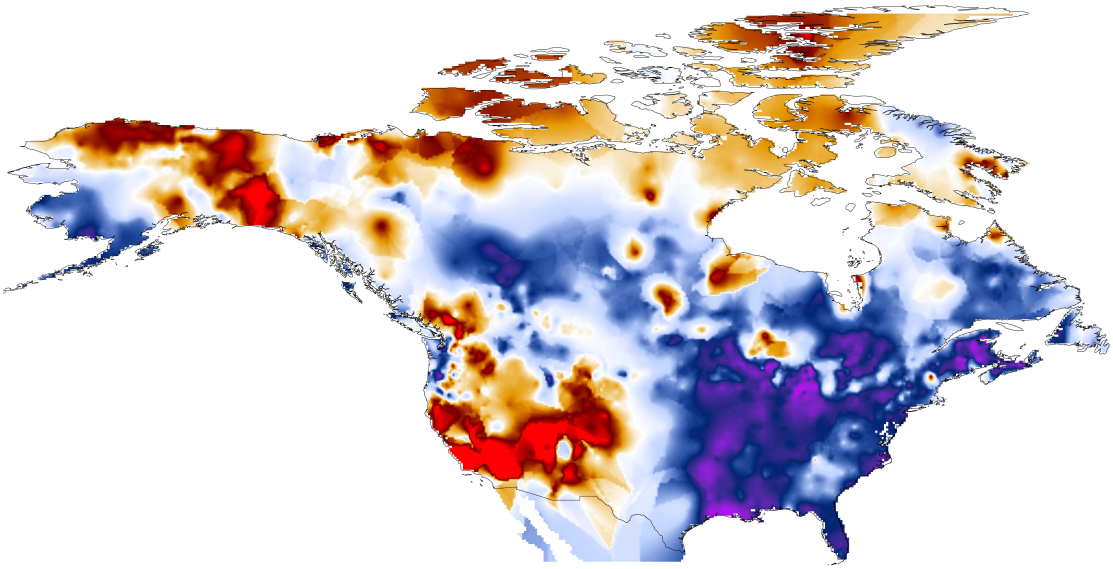


Figure 68: Standard deviation of all modern TJUL data sets (1961–1990) at each modern pollen site. Lower (higher) standard deviations are indicated with colder (wamer) colours.

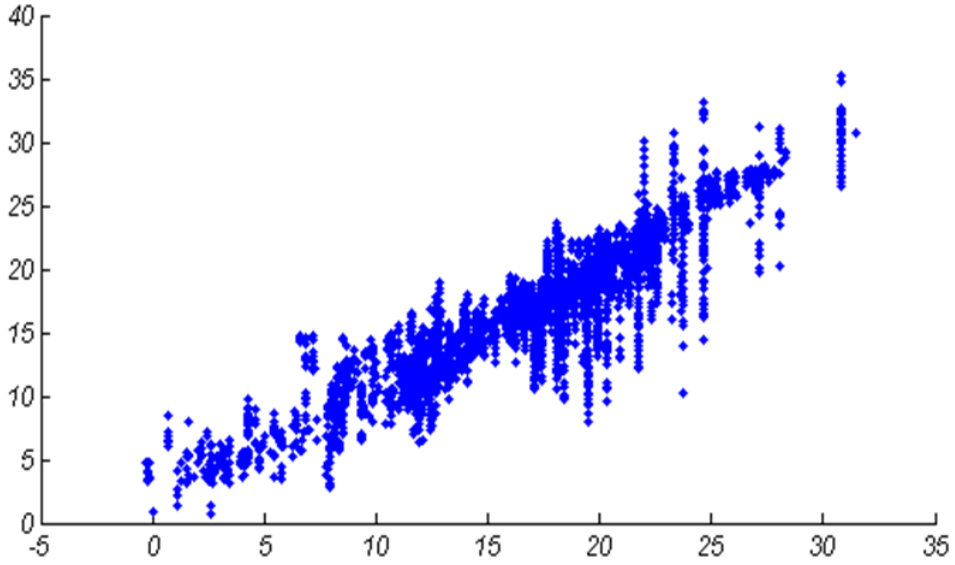


Figure 69: Scatterplot of Whitmore MTWA (TJUL) versus NCEP TJUL modern (1961-1990).

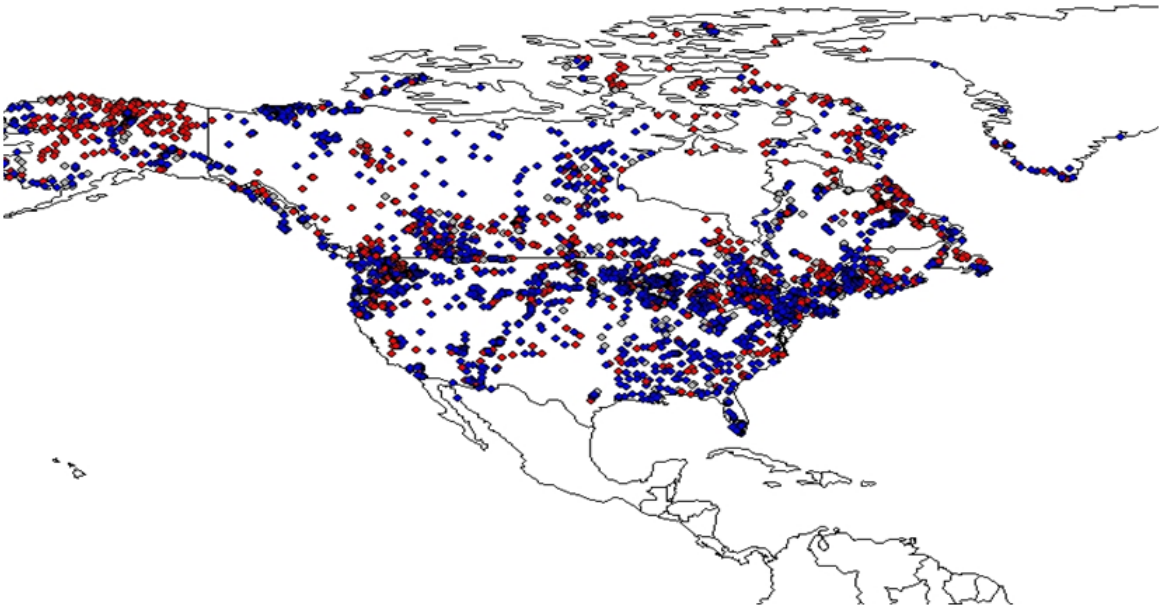


Figure 70: Difference in modern TJUL for the nearest WorldClim grid point to each modern pollen site. Red (blue) modern pollen sites are warmer (colder) than nearest WorldClim grid point.

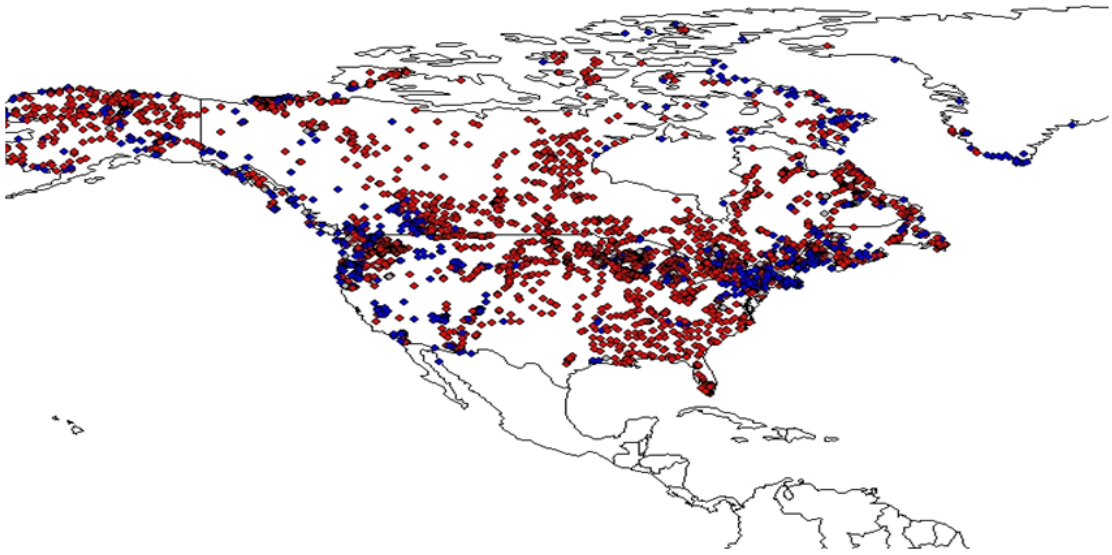


Figure 71: Difference in modern TJUL for the nearest NARR grid point to each modern pollen site. Red (blue) modern pollen sites are warmer (colder) than nearest NARR grid point.

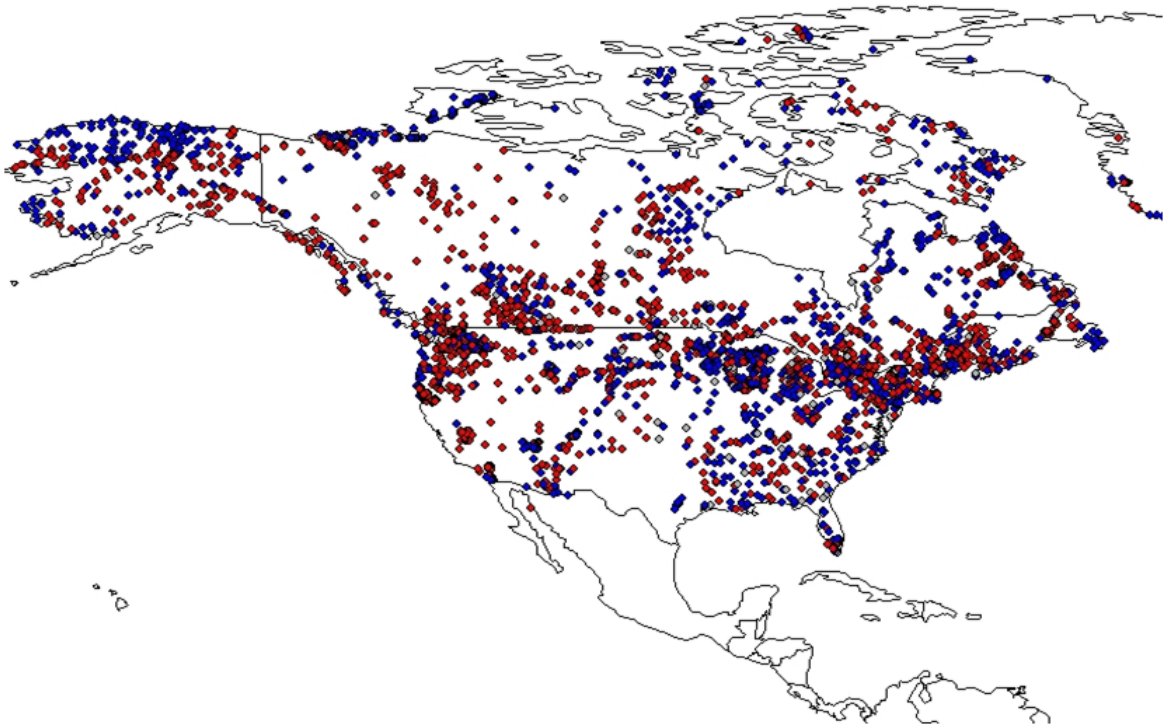


Figure 72: Difference in modern TJUL for the nearest GHCN station to each modern pollen site. Red (blue) modern pollen sites are warmer (colder) than nearest GHCN station.

tunedAD	Whitmore	BESTS1	GHCN	NEWetal	BESTG1	BESTG2	WorldCli	ERA-40_1	ERA-40_2	NCEP1	NCEP2	NARR	ERA_INT
1950	0.02	0.03	0.02	0.02	0.01	0.00	0.03	0.04	0.03	0.00	0.01	0.07	0.07
1900	-0.15	-0.09	-0.16	-0.16	-0.12	-0.12	-0.12	-0.03	-0.03	-0.11	-0.08	-0.01	-0.03
1850	-0.25	-0.14	-0.21	-0.25	-0.17	-0.17	-0.20	-0.09	-0.08	-0.16	-0.13	-0.06	-0.11
1800	-0.20	-0.11	-0.21	-0.22	-0.13	-0.13	-0.15	-0.11	-0.10	-0.16	-0.13	-0.05	-0.07
1750	-0.17	-0.08	-0.19	-0.18	-0.08	-0.08	-0.11	-0.12	-0.12	-0.11	-0.09	-0.05	-0.05
1700	-0.19	-0.07	-0.20	-0.20	-0.08	-0.08	-0.13	-0.05	-0.05	-0.04	-0.02	-0.07	-0.03
1650	-0.20	-0.07	-0.26	-0.20	-0.11	-0.10	-0.14	-0.07	-0.07	-0.06	-0.05	-0.09	-0.06
1600	-0.22	-0.09	-0.27	-0.22	-0.08	-0.08	-0.18	-0.02	-0.02	-0.07	-0.05	-0.12	-0.07
1550	-0.21	-0.12	-0.25	-0.21	-0.06	-0.06	-0.19	-0.01	-0.01	-0.08	-0.05	-0.12	-0.05
1450	-0.18	-0.12	-0.20	-0.18	0.00	0.00	-0.17	0.05	0.04	-0.06	-0.03	-0.08	0.00
1350	-0.16	-0.12	-0.16	-0.16	-0.01	-0.01	-0.15	0.06	0.05	-0.04	-0.01	-0.05	-0.01
1250	-0.20	-0.13	-0.18	-0.21	-0.04	-0.04	-0.18	0.04	0.03	-0.04	-0.01	-0.05	-0.04
1150	-0.14	-0.08	-0.12	-0.17	-0.02	-0.03	-0.14	0.08	0.06	-0.01	0.02	-0.03	0.00
1050	-0.13	-0.06	-0.10	-0.15	-0.03	-0.04	-0.11	0.08	0.06	-0.02	0.00	-0.08	-0.01
950	-0.07	0.00	-0.05	-0.08	0.01	0.01	-0.02	0.09	0.08	0.02	0.04	0.05	0.04
850	-0.04	0.05	0.00	-0.05	0.03	0.02	0.00	0.08	0.07	0.04	0.07	0.10	0.07
750	-0.05	0.05	0.02	-0.05	0.01	0.01	0.00	0.05	0.05	0.04	0.07	0.06	0.07
650	-0.10	0.03	-0.03	-0.10	-0.02	-0.03	-0.04	-0.03	-0.04	-0.02	0.02	0.01	0.01
550	-0.11	-0.02	-0.02	-0.13	-0.03	-0.04	-0.06	-0.03	-0.03	-0.02	0.02	-0.03	-0.01
450	-0.10	0.00	-0.05	-0.12	-0.04	-0.05	-0.05	-0.08	-0.07	-0.04	0.00	-0.05	-0.03
350	-0.08	0.02	-0.04	-0.10	-0.01	-0.02	-0.03	-0.04	-0.04	-0.03	0.01	-0.02	0.01
250	-0.08	0.01	-0.07	-0.10	-0.01	-0.01	-0.03	-0.02	-0.03	-0.02	0.01	-0.03	0.02
150	-0.06	0.06	-0.04	-0.07	0.01	0.01	-0.01	0.06	0.06	0.04	0.07	0.00	0.06
50	-0.03	0.11	-0.01	-0.04	0.05	0.04	0.01	0.12	0.11	0.07	0.09	0.03	0.07
-50	-0.04	0.09	0.00	-0.07	0.04	0.02	0.00	0.09	0.07	0.03	0.05	0.00	0.05
-150	-0.03	0.06	0.01	-0.08	0.03	0.02	-0.01	0.10	0.08	0.04	0.06	0.01	0.05

Figure 73: MTWA (TJUL) anomalies through time (composite average of all sites in North America) using different modern TJUL data sets.

SITENAME	LON	LAT	NARR_ TJUL	NCEP_ TJUL	Newetal_ TJUL	WorldClim_ TJUL	nearest_GHCN_ TJUL	mtwa_ whitmore
Little Thunder Reservoir	-105.38	43.69	25.5	20.8	21.3	21.5	21.3	21.4
BCSU32	-123.33	48.5	16.3	17.1	16.1	15.5	15.4	16.3
PANM10	-117.25	36.5	31.4	22	27.5	31.5	38.4	25.4
BOREA24, Mile 49	-105.53	54.43	18.6	16.3	16.6	16.6	16.9	16.3
Squa Pan Pond	-68.33	46.55	18.5	17.5	18.6	17.9	18.7	18.6
Watchtower Basin	-117.08	52.78	10.4	10.7	8.9	7.1	14.1	9.9
61PMA	-90.1	43	22.9	22.1	21.7	21.5	21.5	21.7
Jack Lake	-81.77	47.32	18.4	17.7	18	16.7	17.2	18
BKS102	-140.5	61.5	-0.9	6.8	8.2	8.5	14	10
Qaqssiarssuk, Isthmus	-45.55	61.15	6	4.3	9	9.3	7.6	9.7
NEWSU 25	-58.63	48.55	16.3	14.4	15.4	15.4	16.1	15.5
Tiinkdhul Lake	-143.15	66.58	19.1	11.9	12.7	14.2	16.5	12.8
Vermnt18	-89.83	43.08	22.9	22.1	21.6	21.9	21.5	21.7
Yarger Lake	-141.63	62.97	17	6.6	14.4	14.3	14.9	14.6
RS48	-93.63	75.15	7.6	2.3	4.1	4.3	4	4.1
Slide Mountain Site 4	-74.37	42.03	21.1	19.9	18.7	18.4	21.7	19.4
GRW16	-116.8	34.15	19.1	23.7	21	16.2	25.3	20.4
ABD184	-161.67	62.17	14.2	13	12.7	12.7	12.8	12.7
RB03A	-76.31	57.76	12.4	8.4	9.1	9.1	9.1	9.1
Onion Lake	-123.75	41.38	16.8	18.1	16.2	14.4	21.8	15.5
Site 22 [Whitehead & Jackson 1990]	-73.98	44.12	16.5	18.6	16.1	16.1	17.4	16.4
NMHL23	-105	36	21.6	23.8	19.4	19.1	20.2	21.7
10PMA	-91.92	44.88	22.2	20.4	22	22.3	22	22.1
LM01	-74.58	55.63	13	11.4	10.8	10.7	13.2	10.8
45A	-88.87	44.82	21.3	20.5	20.3	20.2	19.4	20.2
CNTA14	-111.5	33.5	35.3	30.8	30.9	31.2	32.1	32.5
Fahi Pond	-69.9	44.9	19.5	18.2	19.3	19	19.1	19.4

Figure 74: Range of modern TJUL values in °C assigned to each modern pollen site.

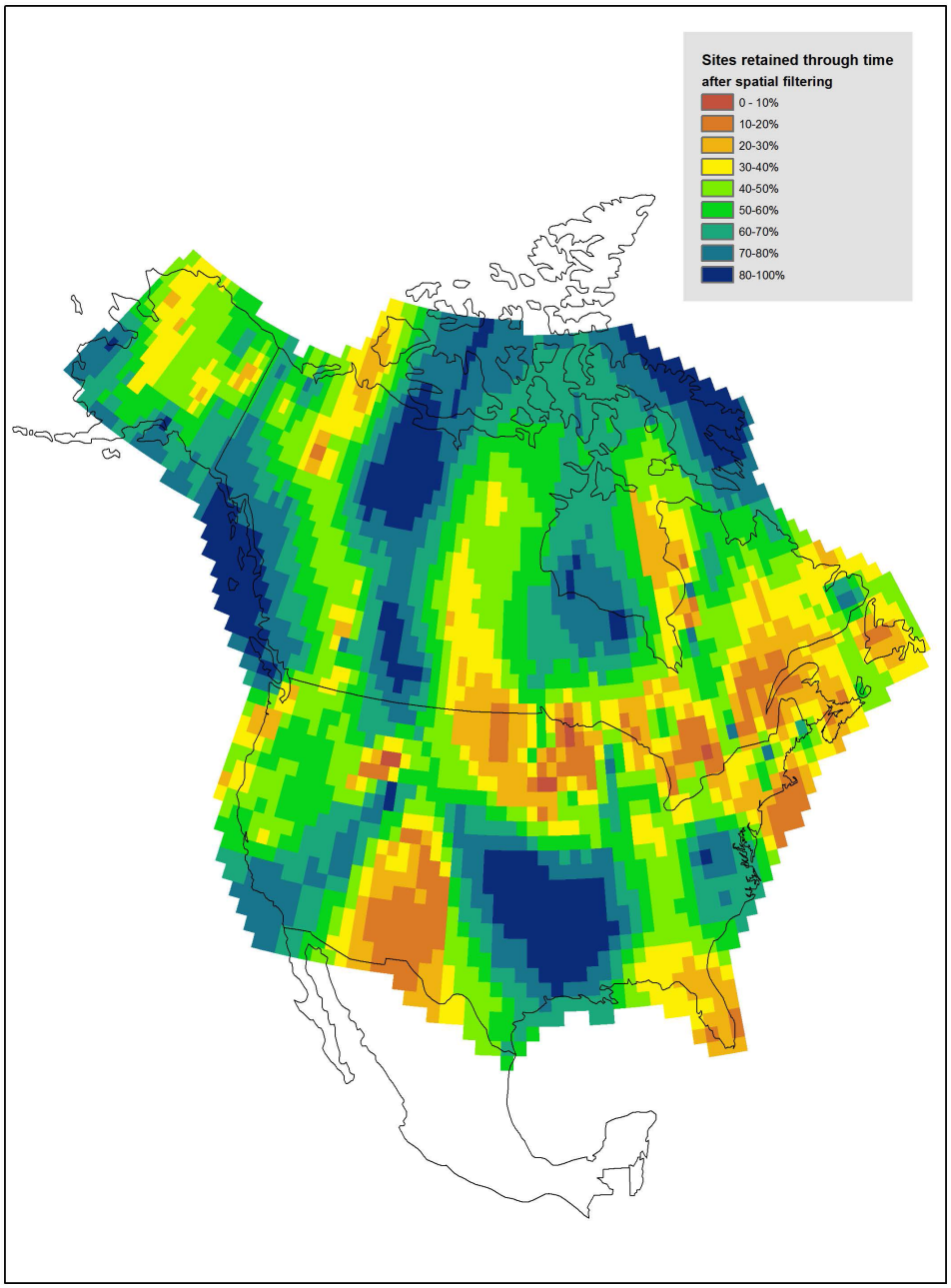


Figure 75: Percentage of sites retained through time for the ANNP reconstruction after the application of spatial filtering.

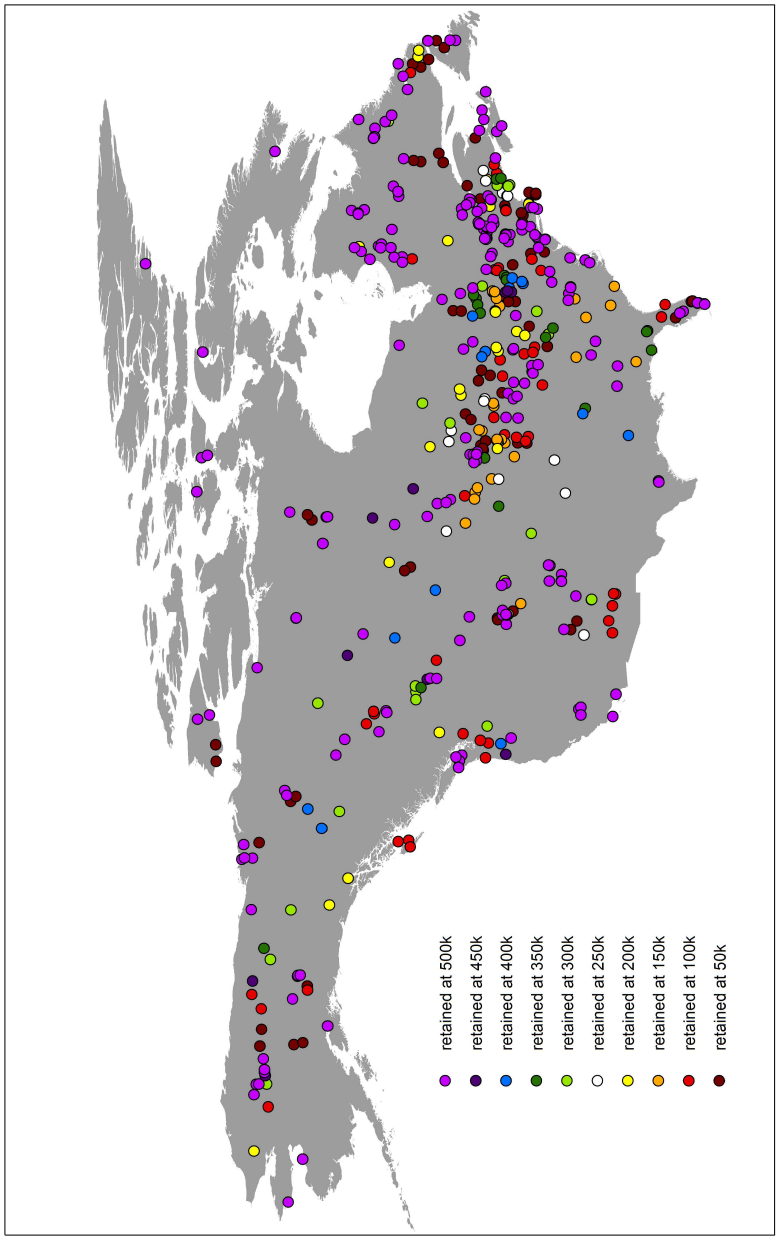


Figure 76: Sites retained for ANNP reconstruction after spatial filtering at different lag distances from 50 km to 500 km.

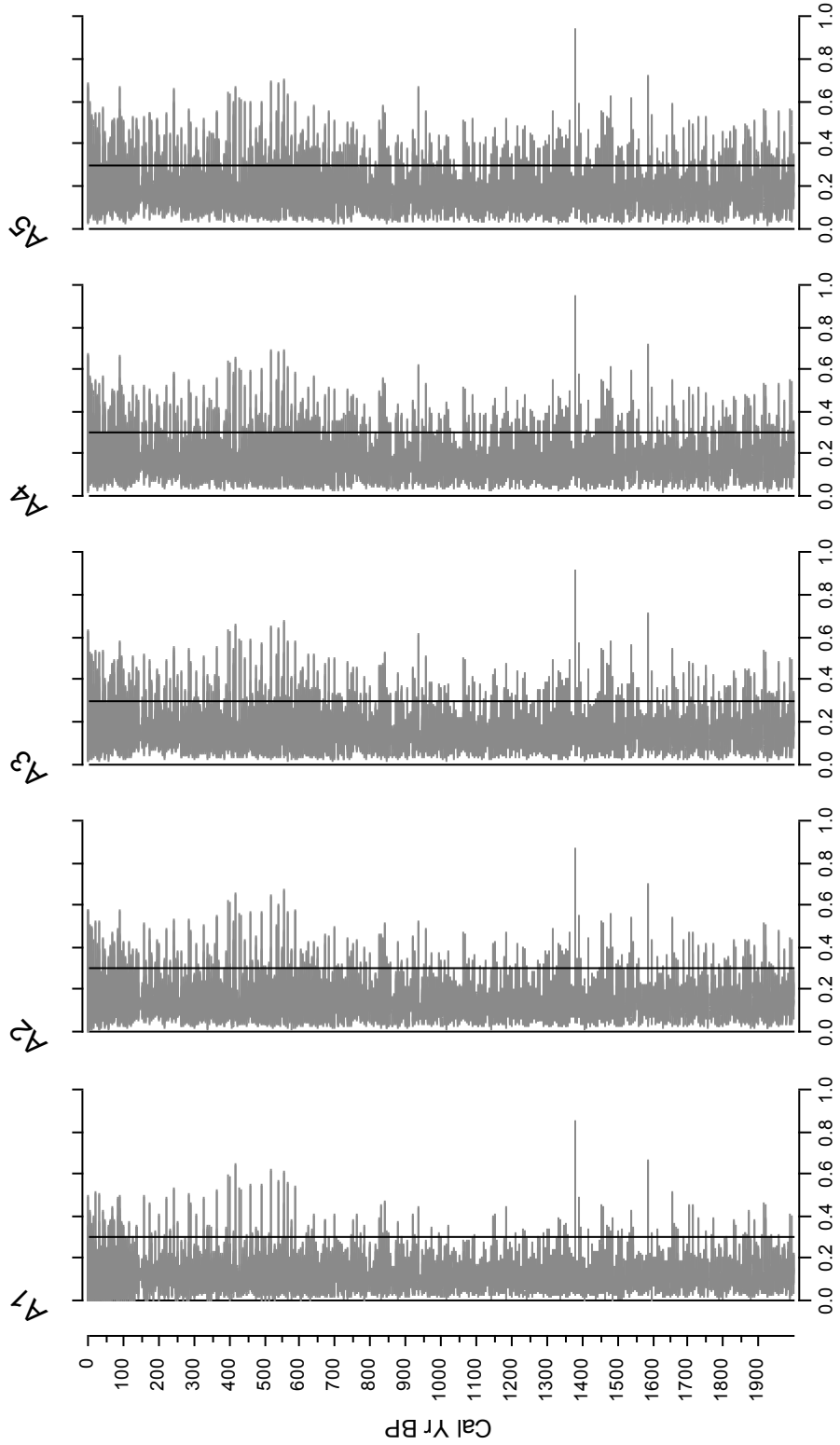


Figure 77: Squared Chord Distance (SCD) plot of the top five analogs for the reconstruction of ANNP.

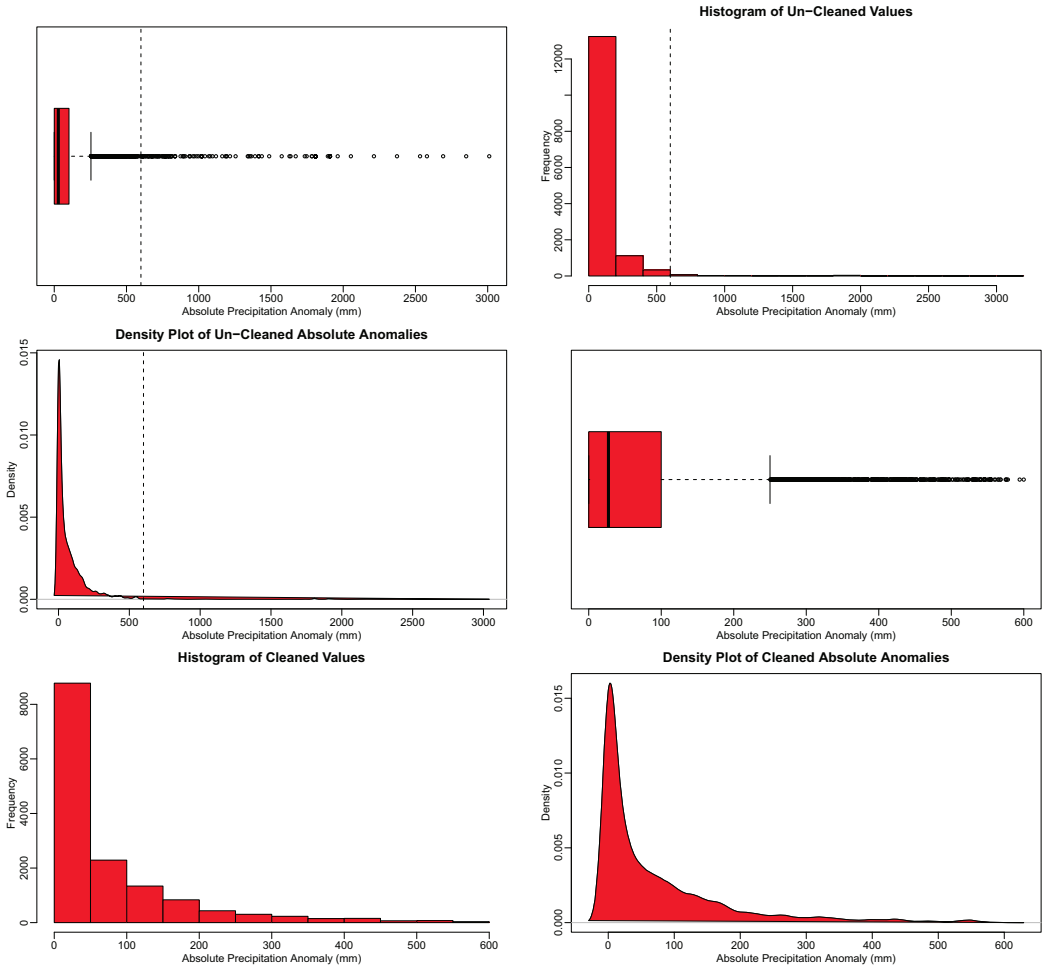


Figure 78: Frequency distribution of ANNP reconstruction data points (outliers not removed) versus ANNP reconstruction data points (outliers removed) [courtesy of R.G. Way].

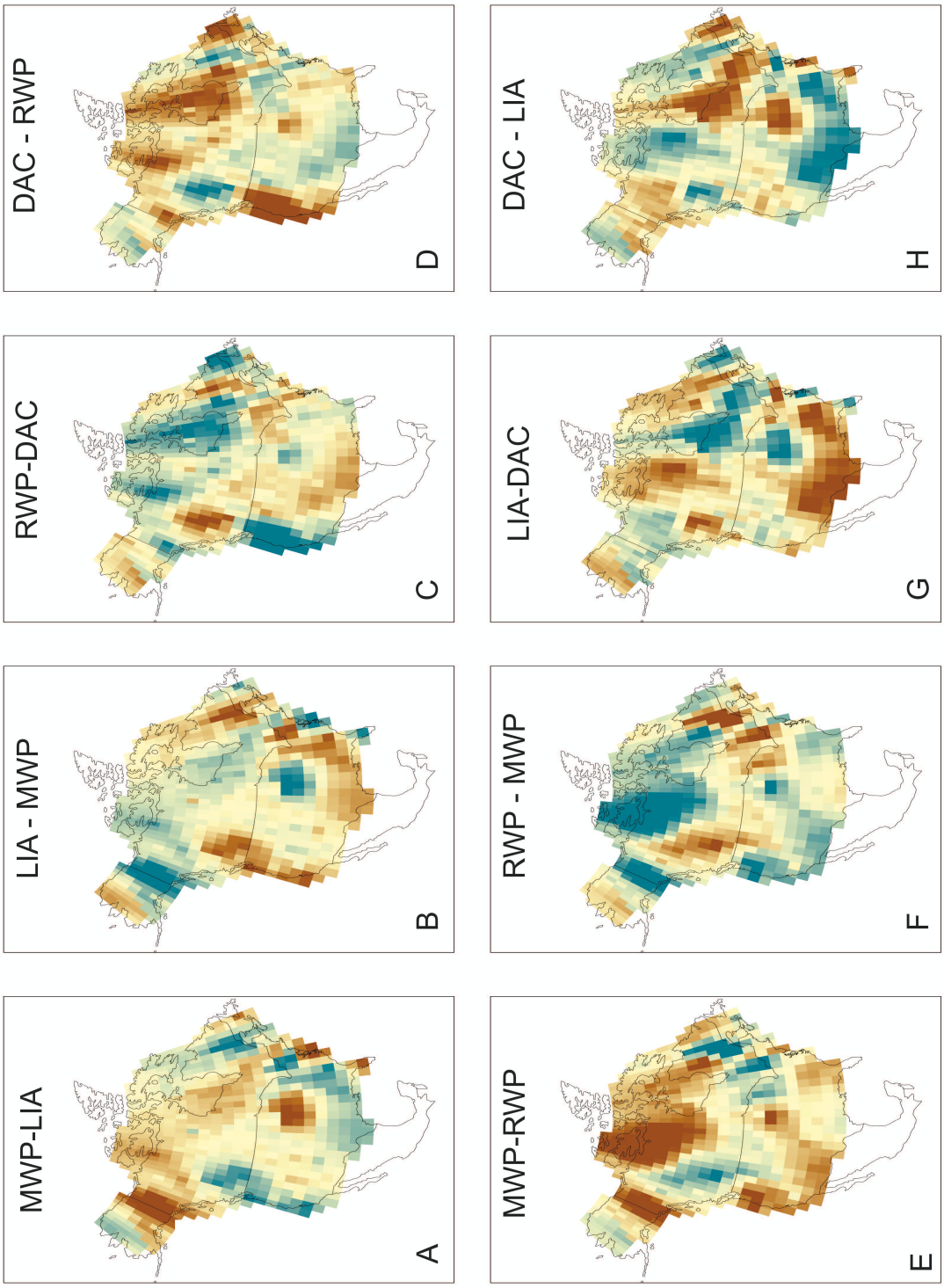


Figure 79: ANNP anomalies (in mm) for the MWP (950–1250 CE) minus LIA (1550–1850 CE) (A), LIA minus MWP (B), RWP (150 BCE–150 CE) minus DAC (350–650 CE) (C), DAC minus RWP (D), MWP minus RWP (E), RWP minus MWP (F), LIA minus DAC (G) and DAC minus LIA (H). This was the original version of Figure 15.

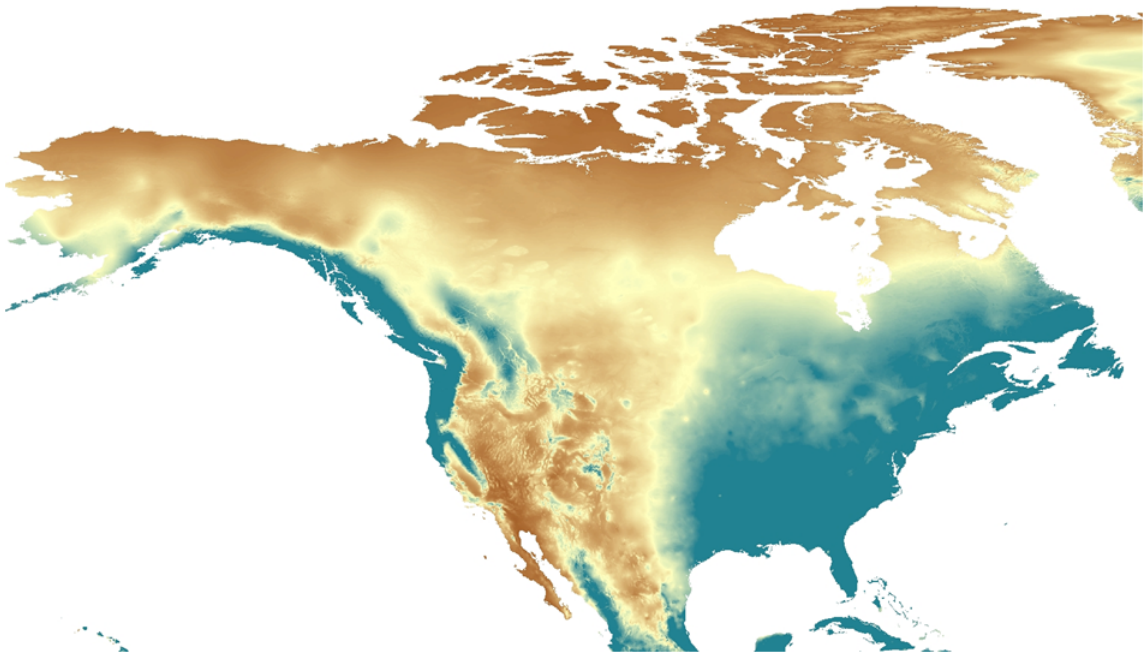


Figure 80: Long term total annual precipitation map (1950 to 2000 CE) from WorldClim data (BIOCLIM12 raster layer) for North America. Aqua (brown) areas are wetter (drier).

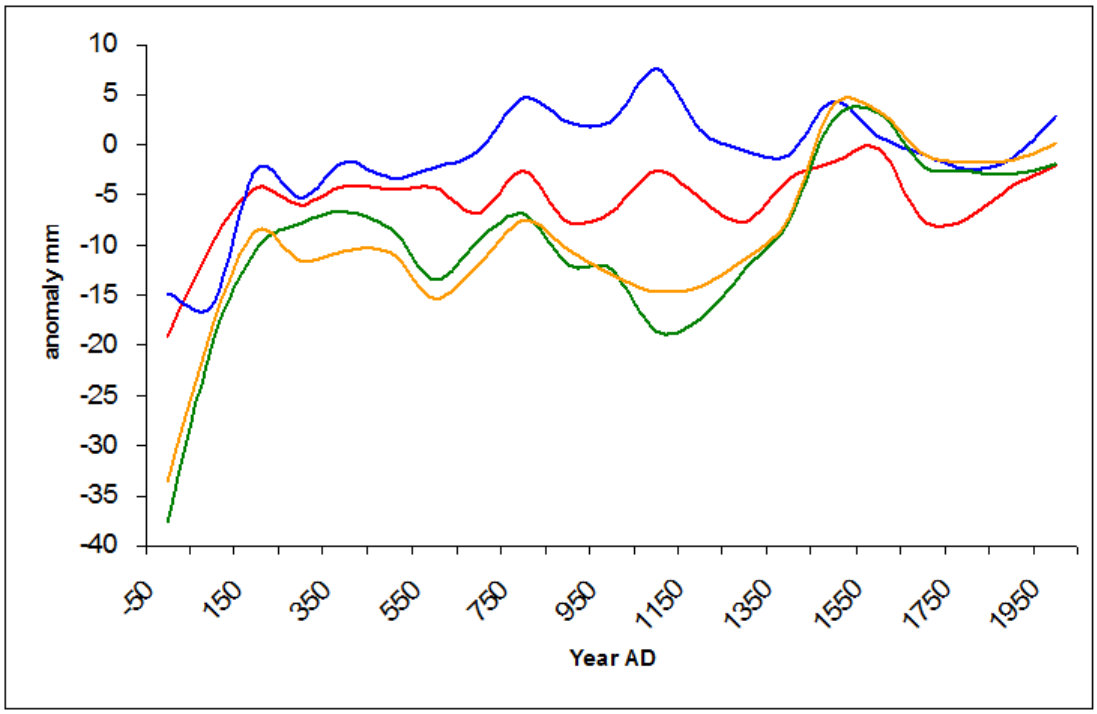


Figure 81: Reconstruction of ANNP (anomalies are in mm/yr) using 35 (red), 64 (blue), 134 (green), 135 (gold) pollen sums and average of all 4 pollen sums (black).

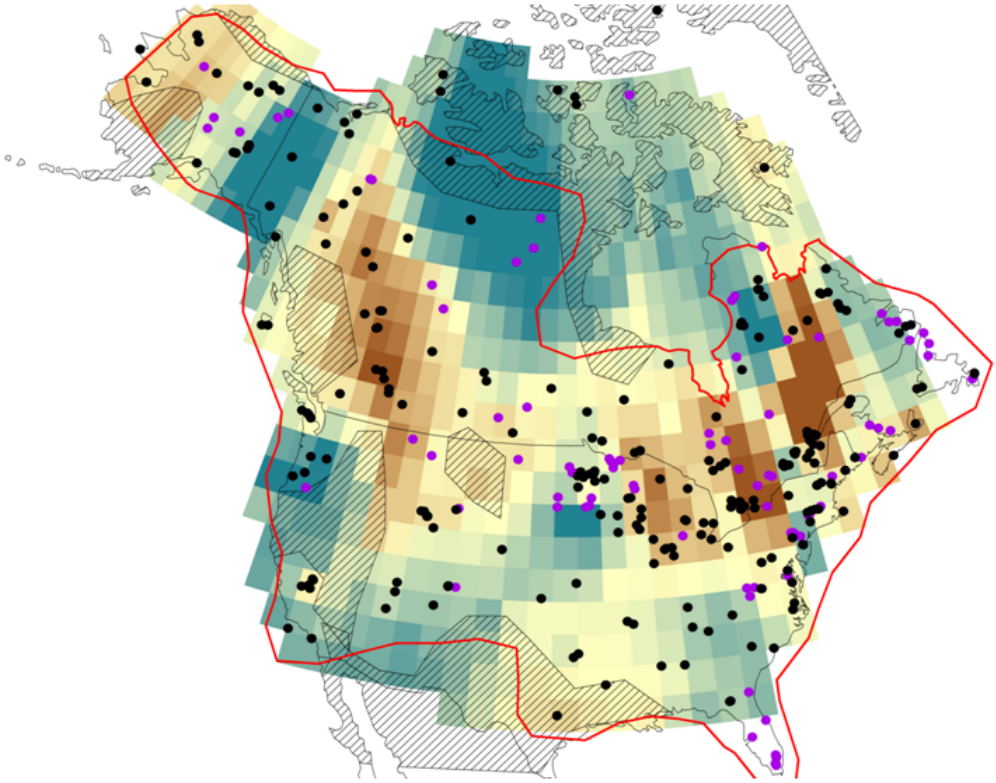


Figure 82: Reconstruction of ANNP example with site distribution and hatched areas indicating low confidence reconstructions and sample land mask (red border). Note that purple sites are not retained after filtering.

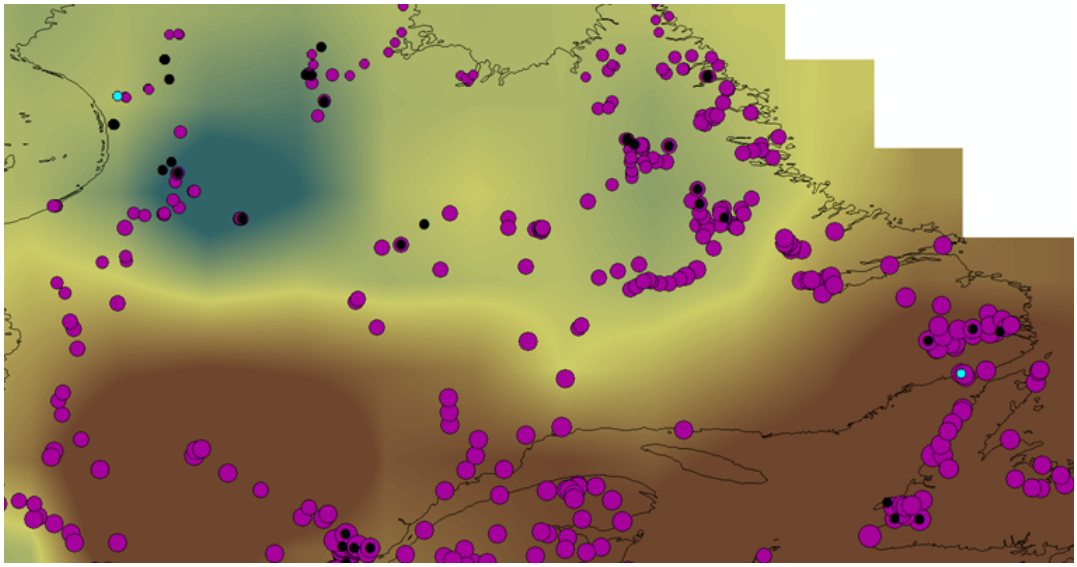


Figure 83: Example of the large distance between a fossil pollen site and the closest modern analog (modern pollen site) for the reconstruction of ANNP in Eastern Canada.



Figure 84: Flowchart of the spatial filtering methodology that was developed by the author.

	Moran's I			Moran's I (z-value)			neg/neg	pos/pos	pos/neg	zero	total
lag (km)	neg	zero	pos	neg	zero	pos					
50	90	227	192	86	227	196	86	192	4	227	509
%	0.176817	0.445972	0.37721	0.168959	0.445972	0.385069	0.168959	0.37721	0.007859	0.445972	
100	148	105	256	79	105	325	79	256	69	105	509
%	0.290766	0.206287	0.502947	0.155206	0.206287	0.638507	0.155206	0.502947	0.13556	0.206287	
150	181	49	279	71	49	389	71	279	110	49	509
%	0.355599	0.096267	0.548134	0.139489	0.096267	0.764244	0.139489	0.548134	0.21611	0.096267	
200	201	30	278	67	30	412	67	278	134	30	509
%	0.394892	0.058939	0.546169	0.131631	0.058939	0.80943	0.131631	0.546169	0.263261	0.058939	
250	202	17	290	65	17	427	65	290	137	17	509
%	0.396857	0.033399	0.569745	0.127701	0.033399	0.8389	0.127701	0.569745	0.269155	0.033399	
300	195	13	301	61	13	435	61	301	134	13	509
%	0.383104	0.02554	0.591356	0.119843	0.02554	0.854617	0.119843	0.591356	0.263261	0.02554	
350	205	8	296	58	8	443	58	296	147	8	509
%	0.40275	0.015717	0.581532	0.113949	0.015717	0.870334	0.113949	0.581532	0.288802	0.015717	
400	229	6	274	58	6	445	58	274	171	6	509
%	0.449902	0.011788	0.53831	0.113949	0.011788	0.874263	0.113949	0.53831	0.335953	0.011788	
450	237	5	267	57	5	447	57	267	180	5	509
%	0.465619	0.009823	0.524558	0.111984	0.009823	0.878193	0.111984	0.524558	0.353635	0.009823	
500	243	3	263	56	3	450	56	263	187	3	509
%	0.477407	0.005894	0.516699	0.11002	0.005894	0.884086	0.11002	0.516699	0.367387	0.005894	

Figure 85: Table of results for spatial filtering on a reconstruction of MTWA for the last 2,000 years. Counts/percentages are based on sites retained when spatial filtering at different lag distances in km.

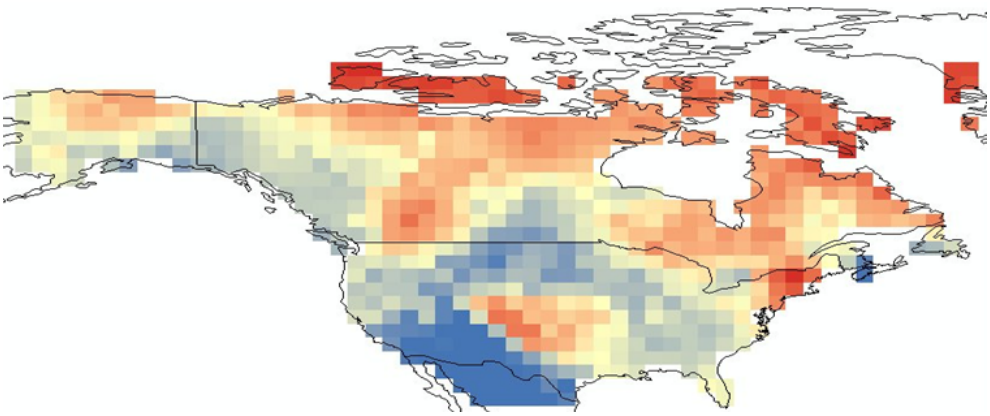


Figure 86: Reconstruction of MTWA (filtered) at 1300 BP and gridded at 2°. Anomalies in red (blue) are warmer (colder) than modern.

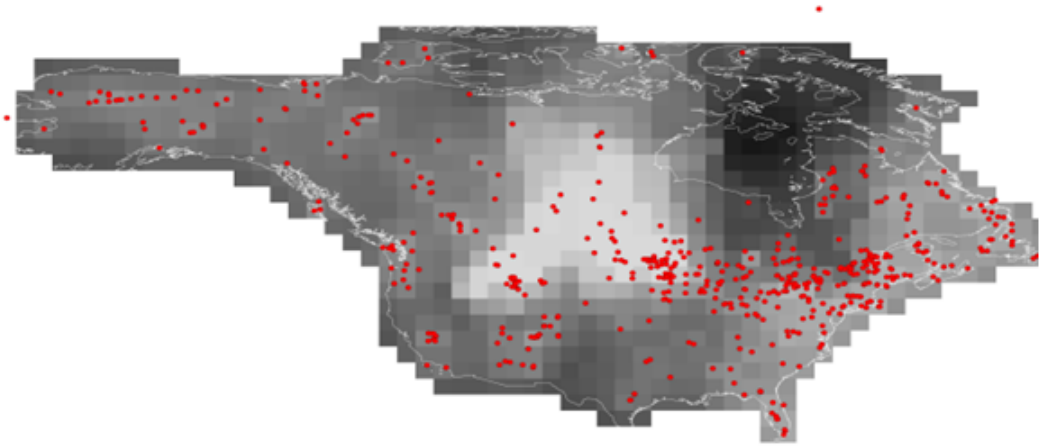


Figure 87: Standard error of prediction map for unfiltered reconstruction of PDSI at 2500 BP. Black (white) areas have a higher (lower) standard error of prediction.

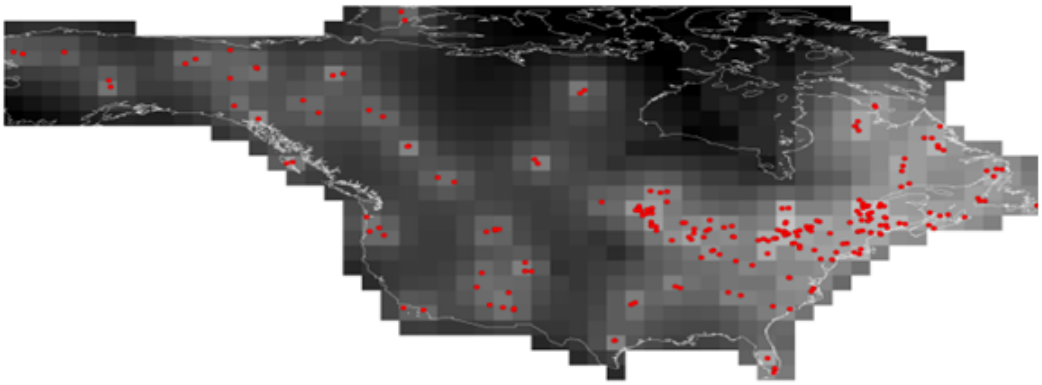


Figure 88: Example of filtered reconstruction of PDSI at 2500 BP. Black (white) areas have a higher (lower) standard error of prediction.

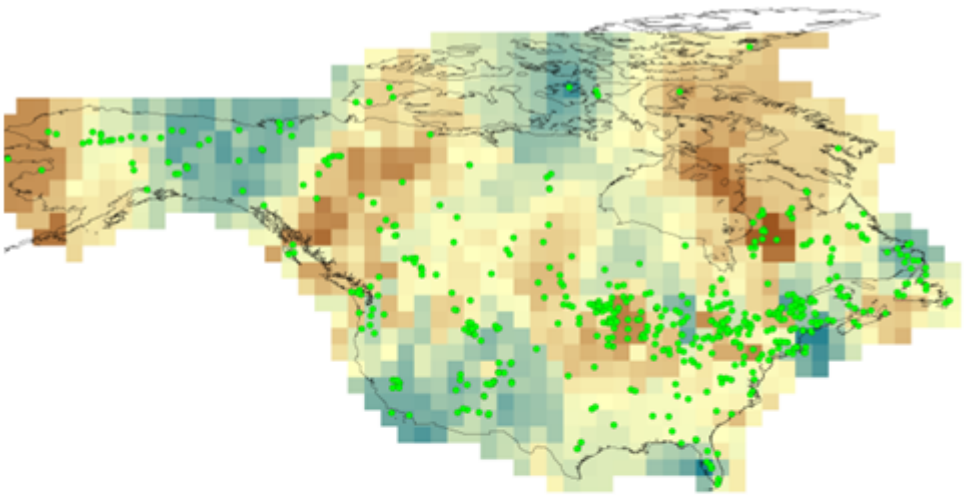


Figure 89: Unfiltered reconstruction of MIPT at 500 BP with site distribution. Brown (aqua) areas are drier (wetter).

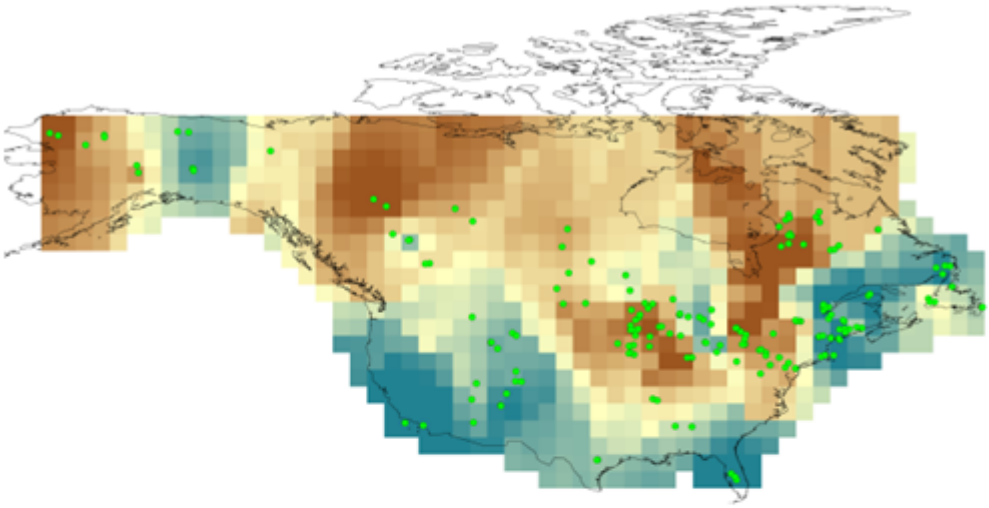


Figure 90: Filtered reconstruction of MIPT at 500 BP with site distribution. Brown (aqua) areas are drier (wetter).

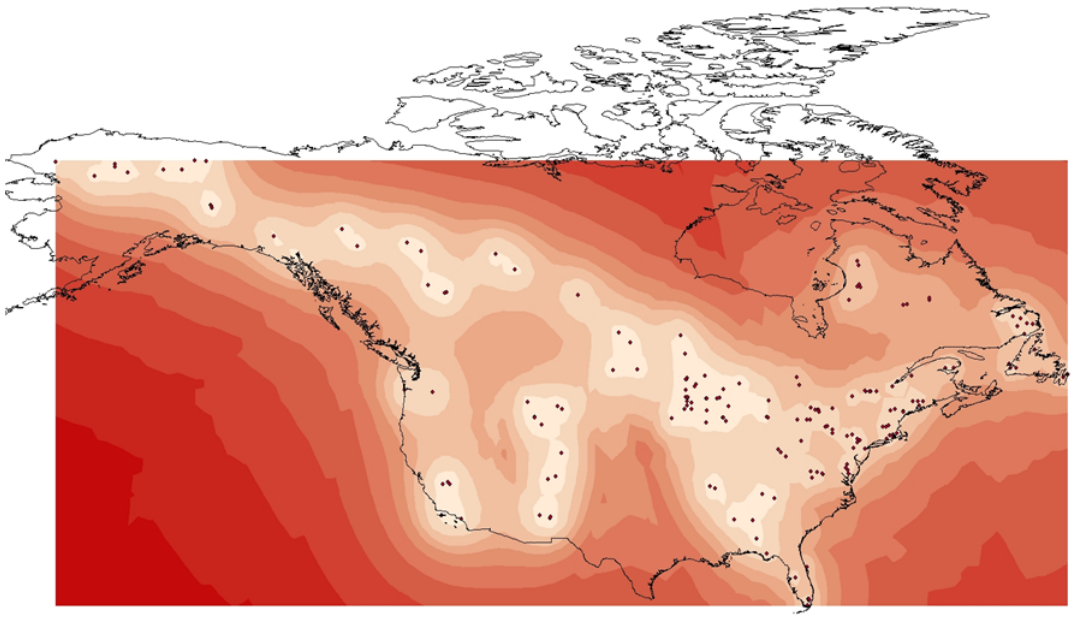


Figure 91: Standard error of prediction map with site distribution for filtered reconstruction of MIPT at 500 BP. Red (white) areas have a higher (lower) standard error of prediction.

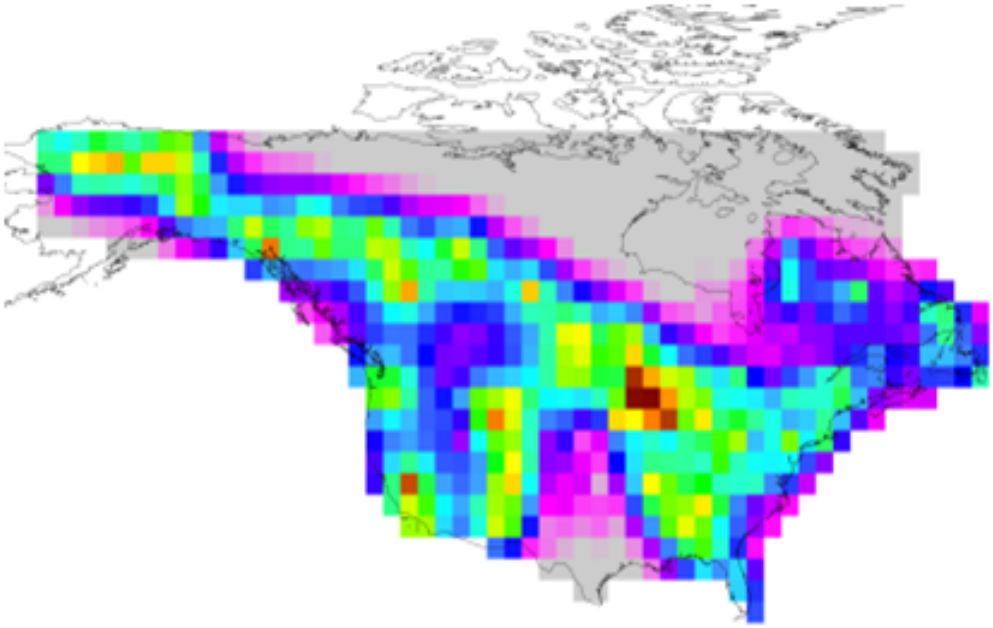


Figure 92: Filtered reconstruction of MIPT at 500 BP with areas in grey having anomalies exceeding +/-1 standard deviation from the average standard error of prediction for the reconstruction at 500 BP.

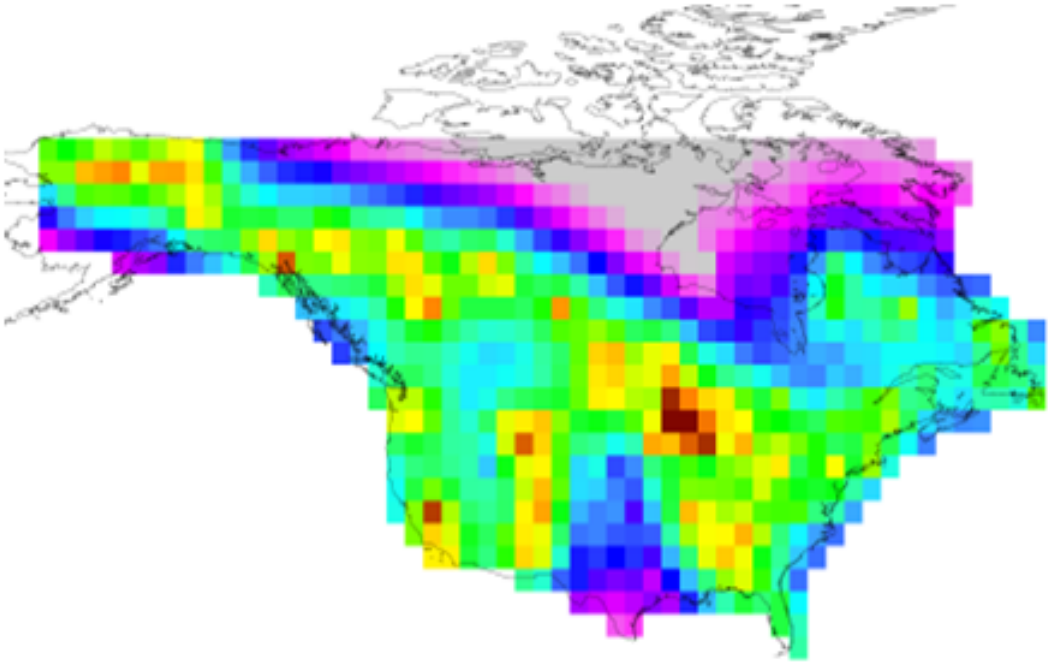


Figure 93: Filtered reconstruction of MIPT at 500 BP with areas in grey having anomalies exceeding +/-2 standard deviations from the average standard error of prediction for the reconstruction at 500 BP.

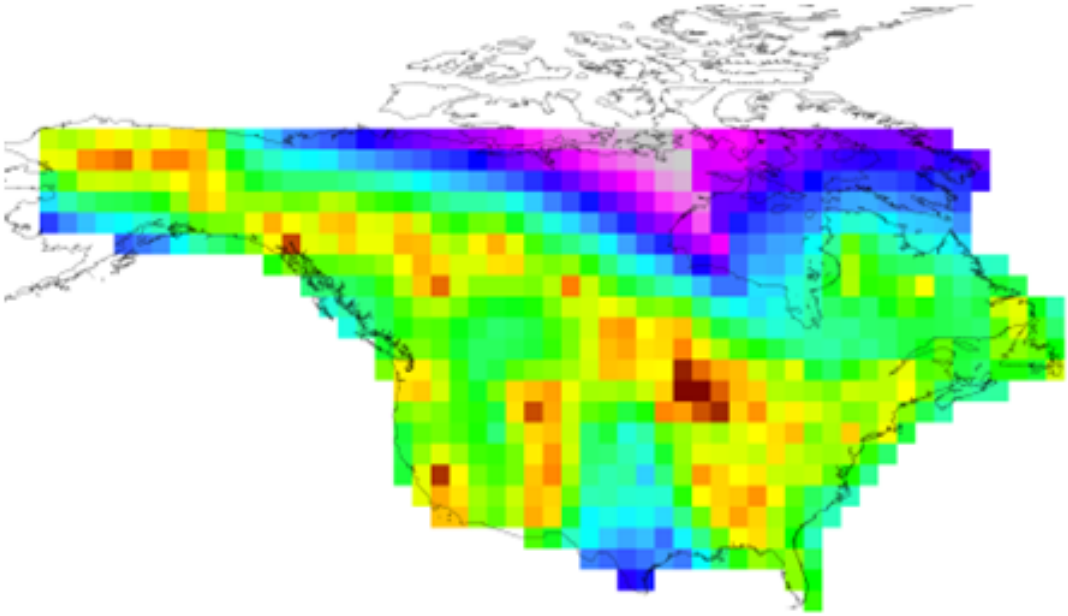


Figure 94: Filtered reconstruction of MIPT at 500 BP with areas in grey having anomalies exceeding +/-3 standard deviations from the average standard error of prediction for the reconstruction at 500 BP.

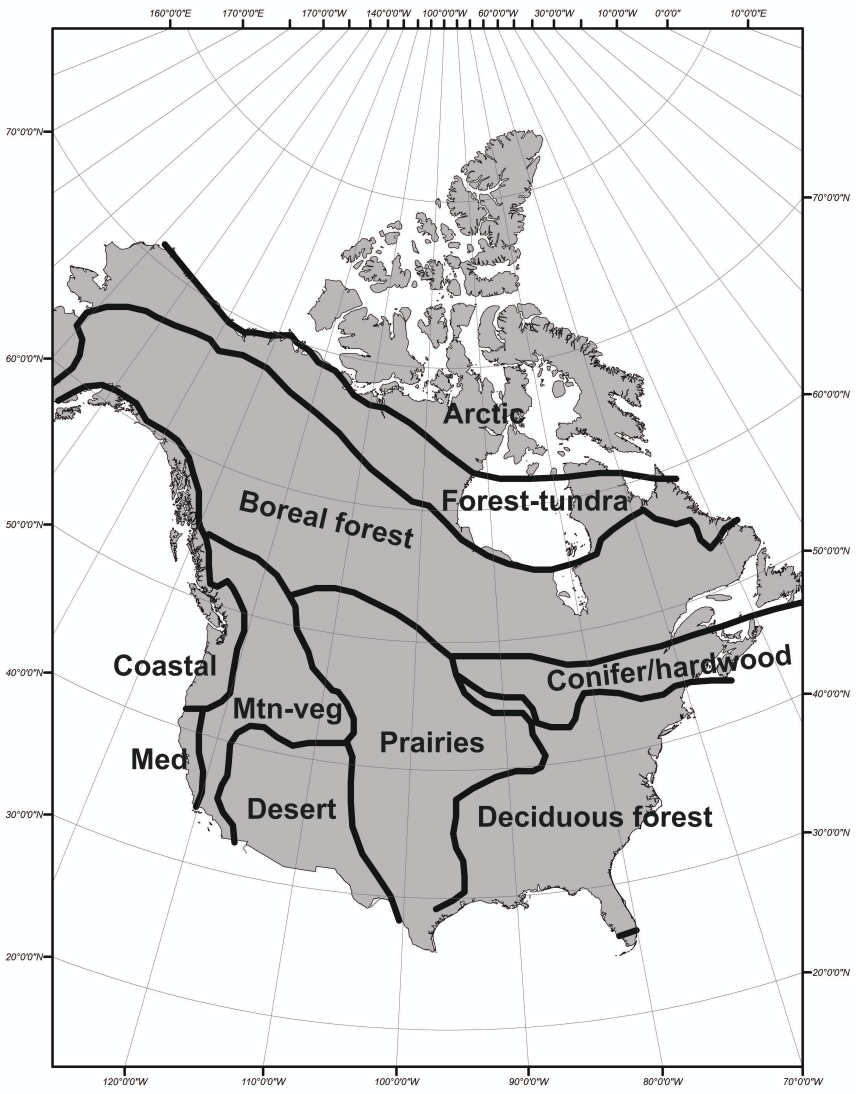


Figure 95: Map of ecoregions of North America (adapted from Fedorova *et al.*, 1994).

Appendix F

All Appendix References

Bartlein, P. J., K. H. Anderson, P. M. Anderson, M. E. Edwards, C. J. Mock, R. S. Thompson, R. S. Webb, T. Webb III, and C. Whitlock (1998), Paleoclimate simulations for north america over the past 21,000 years: features of the simulated climate and comparisons with paleoenvironmental data, *Quaternary Science Reviews*, *17*(6-7), 549–586.

Bartlein, P. J., S. J. Harrison, S. Brewer, S. Connor, B. A. S. Davis, K. Gajewski, J. Guiot, T. I. Harrison-Prentice, A. Henderson, O. Peyron, I. C. Prentice, M. Scholze, H. Seppä, B. Shuman, S. Sugita, R. S. Thompson, A. E. Viau, J. Williams, and H. Wu (2011), Pollen-based continental climate reconstructions at 6 and 21 ka: a global synthesis, *Climate Dynamics*, *37*(3-4), 775–802.

Bonfils, C., N. de Noblet-Ducoundr, J. Guiot, and P. Bartlein (2004), Some mechanisms of mid-holocene climate change in europe, inferred from comparing pmip models to data., *Climate Dynamics*, *23*(1), 79–98.

Brewer, S., J. Guiot, and F. Torre (2006), Mid-holocene climate change in europe: a data-model comparison, *Climate of the Past Discussions*, *2*(6), 1155–1186.

Brewer, S., J. Guiot, and F. Torre (2007), Mid-holocene climate change in europe: A data-model comparison, *Climate of the Past*, *3*(3), 499–512.

Broccoli, A. J., and E. P. Marciniak (1996), Comparing simulated glacial climate and paleodata: a reexamination, *Paleoceanography*, *11*(1), 3–14.

Cheddadi, R., G. Yu, J. Guiot, S. Harrison, and I. Prentice (1996), The climate of europe 6000 years ago, *Climate Dynamics*, *13*(1), 1–9.

Edwards, T. L., M. Crucifix, and S. P. Harrison (2007), Using the past to constrain the future: How the paleorecord can improve estimates of global warming, *Progress in Physical Geography*, *31*(5), 481–500.

Fedorova, I., Y. Volkova, and D. Varlyguin (1994), World vegetation cover: Digital raster data on a 30-minute cartesian orthonormal geodetic (lat./long.) 1,080× 2,160 grid, *Global Ecosystems Database Version*, *2*.

Gajewski, K. (2008), The global pollen database in biogeographical and palaeoclimatic studies, *Progress in Physical Geography*, *32*(4), 379–402.

Kageyama, M., O. Peyron, S. Pinot, P. Tarasov, J. Guiot, S. Joussaume, and G. Ramstein (2001), The last glacial maximum climate over europe and western siberia: a pmip comparison between models and data, *Climate Dynamics*, *17*(1), 23–43.

Liao, X., F. Street-Perrott, and J. Mitchell (1994), Gcm experiments with different cloud parameterization: comparisons with palaeoclimatic reconstructions for 6000 years bp, *Data Model*, *1*, 99–123.

New, M., D. Lister, M. Hulme, and I. Makin (2002), A high-resolution data set of surface climate over global land areas, *Climate research*, *21*(1), 1–25.

Otto-Bliesner, B., R. Schneider, E. Brady, M. Kucera, A. Abe-Ouchi, E. Bard, P. Braconnot, M. Crucifix, C. Hewitt, M. Kageyama, O. Marti, A. Paul, A. Rosell-Mel, C. Waelbroeck, S. Weber, M. Weinelt, and Y. Yu (2009), A comparison of pmip2 model simulations and the margo proxy reconstruction for tropical sea surface temperatures at last glacial maximum, *Climate Dynamics*, *32*(6), 799–815.

Pinot, S., G. Ramstein, S. Harrison, I. Prentice, J. Guiot, M. Stute, and S. Joussaume (1999), Tropical paleoclimates at the last glacial maximum: comparison of paleoclimate modeling intercomparison project (pmip) simulations and paleodata, *Climate Dynamics*, *15*(11), 857–874.

Prentice, I., S. Harrison, D. Jolly, and J. Guiot (1998), The climate and biomes of europe at 6000 yr bp: comparison of model simulations and pollen-based reconstructions, *Quaternary Science Reviews*, *17*(6-7), 659–668.

Sawada, M., A. Viau, G. Vettoretti, W. Peltier, and K. Gajewski (2004), Comparison of north-american pollen-based temperature and global lake-status with cccma agcm2 output at 6ka, *Quaternary Science Reviews*, *23*(3), 225–244.

Texier, D., N. De Noblet, S. Harrison, A. Haxeltine, D. Jolly, S. Joussaume, F. Laarif, I. Prentice, and P. Tarasov (1997), Quantifying the role of biosphere-atmosphere feedbacks in climate change: coupled model simulations for 6000 years bp and comparison with palaeodata for northern eurasia and northern africa, *Climate Dynamics*, *13*(12), 865–881.

Vettoretti, G., W. Peltier, N. McFarlane, and P. p. groups (2000), The simulated response of the climate system to changes in soil moisture parameterization under paleoclimatic boundary conditions at 6000 years before present, *Canadian Journal of Earth Sciences*, *37*(5), 635–660.

Viau, A., K. Gajewski, M. Sawada, and P. Fines (2006), Millennial-scale temperature variations in north america during the holocene, *Journal of Geophysical Research: Atmospheres (1984-2012)*, *111*(D9).

Webb III, T., P. Bartlein, and J. Kutzbach (1987), Climatic change in eastern north america during the past 18,000 years; comparisons of pollen data with model results, *The Geology of North America*, *3*, 447–462.

Webb III, T., K. H. Anderson, P. Bartlein, and R. Webb (1998), Late quaternary climate change in eastern north america a comparison of pollen-derived estimates with climate model results, *Quaternary Science Reviews*, *17*(6-7), 587–606.

Whitmore, J., K. Gajewski, M. Sawada, J. Williams, B. Shuman, T. Bartlein, P.J .and Minckley, A. Viau, T. Webb III, S. Shafer, P. Anderson, and L. Brubaker (2005), Modern pollen data from north america and greenland for multi-scale paleoenvironmental applications, *Quaternary Science Reviews*, *24*(16), 1828–1848.

Williams, J., and B. Shuman (2008), Obtaining accurate and precise environmental reconstructions from the modern analog technique and north american surface pollen dataset, *Quaternary Science Reviews*, *27*(7), 669–687.