uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Adrianna Muños

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Computer Science)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Genome Rearrangements: Structural Inference and Functional Consequences

TITRE DE LA THÈSE / TITLE OF THESIS

David Sankoff

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Evangelos Kranakis

WonSook Lee

Benjamin Raphael
Brown University

Marcel Turcotte

Gary W. Slater

Le Doyen de la Faculté des etudes supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Genome Rearrangements: Structural Inference and Functional Consequences

Adriana Muñoz

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Science [1]

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

[1]The program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute for Computer Science

Your file Votre référence
ISBN: 978-0-494-74230-3
Our file Notre référence
ISBN: 978-0-494-74230-3

Abstract

As genomes evolve over hundreds of millions years, the chromosomes become rearranged, with segments of some chromosomes inverted, while other chromosomes reciprocally exchange chunks from their ends. These rearrangements lead to the scrambling of the elements of one genome with respect to another descended from a common ancestor. Multidisciplinary work undertakes to mathematically model these processes and to develop statistical analyses and mathematical algorithms to understand the scrambling in the chromosomes of two or more related genomes. A major focus is the reconstruction of the gene order of the ancestral genomes.

There has been a trend in increasing the phylogenetic scope of genome sequencing without finishing the sequence for each genome. With less interest in completing the sequence, there is an increasing number of genomes being published in scaffold or even contig form. Rearrangement algorithms, including gene order-based phylogenetic tools, require whole genome data on gene order or syntenic block order. Then, for gene order-based comparisons or phylogeny, how can we use rearrangement algorithms to handle genomes available in contig or scaffold form only? For contig data, we develop a model for the behaviour of the genomic distance as a function of evolutionary time, and discuss how to invert this function in order to infer elapsed time. We show how to correct for the effect of chromosomal fragmentation in sets of contigs. We apply our methods to data originating mostly in the 12-genome *Drosophila* project [15]. We compare ten *Drosophila* genomes with two other dipteran genomes

and two outgroup insect genomes.

For scaffolds, our method involves optimally filling in genes missing in the scaffolds, and using the augmented scaffolds directly in the rearrangement algorithms as if they were chromosomes, while making a number of corrections, e.g., we correct for the number of extra fusion/fission operations required to make scaffolds comparable to full assemblies. We model the relationship between scaffold density and genomic distance, and estimate the parameters of this model while comparing the angiosperms genomes *Ricinus communis* and *Vitis vinifera*.

A separate question arises of what the biological consequences of breakpoint creation are, rather than just their structural aspects. The question I will ask is whether proximity to the site of a breakpoint event changes the activity of a gene. I propose to investigate this by comparing the distribution of distances to the nearest breakpoint of genes that change expression in human versus the distribution of genes that do not change expression in human, compared to other primate species (e.g. macaque or chimpanzee).

Keywords: chromosome rearrangement, comparative genomics, phylogenomics, phylogenetic tree, inversion, reciprocal translocation, transposition, DCJ, breakpoint, gene expression.

Acknowledgements

I would like to thank my supervisor Dr. David Sankoff for his invaluable guidance, encouragement and support throughout my research and in writing my Ph.D. thesis. He provided scientific motivation and an excellent research environment for my work.

I would also like to thank the members of my examining committee, Dr. Marcel Turcotte, Dr. Evangelos Kranakis, Dr. Ben Raphael, and Dr. WonSook Lee, for their valuable criticisms, comments and suggestions that helped me to produce the final version of this document.

Finally, I wish to express my sincere gratitude and appreciation to my Father, my Mother, my beloved daughter Daniella, sisters, brothers and the rest of my family and my friends Leonard and Gylliane for their constant support and encouragement throughout the completion of this dream.

Dedication

In memory of my Father, to my Mother and to my beloved daughter Daniella whose unconditional love inspired and motivated me throughout the completion of this dream.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis aims to increase the pertinence of the bioinformatics approach to the comparative structural analysis of genomes, both by extending it to previously intractable genomic data and by exploring a way of embedding it into functional genomics. It will be about the structural changes in chromosomes during the evolution of genomes, focusing on two aspects: how to compare related genomes when the genome assembly is incomplete, and what are the consequences for gene expression for genes in the neighbourhood of chromosome rearrangements. In this chapter and the next, I introduce the background of comparative genomics.

The advent of nuclear genome sequencing in eukaryotes, especially in mammals [1, 2], lent great impetus to the field of comparative structural genomics. During the previous decade, computer scientists had been preparing for this era by developing analyses and algorithms for comparing whole genomes, illustrating them on smaller genomes such as those of mitochondria [3, 4], chloroplasts [5] and prokaryotes [6], and using various sets of ordered markers, usually genes, as the input data, rather than raw sequence.

The central result during this period was the polynomial-time Hannenhalli-Pevzner algorithm [7] for sorting, in a minimum number of steps, a signed genome

(i.e., a signed permutation of $1, 2, \ldots, n$) by successively reversing contiguous fragments of the permutation, where each reversal also switches the polarity of each term in its scope. This allowed the rapid calculation of the genomic distance d between two genomes, since a reversal models the major chromosomal rearrangement process in biology, inversion. At the same time, Caprara [8] showed that most efficient sorting by reversals of unsigned permutations is an NP-hard problem. These results responded to a question that had been raised in various forms in the 1980s [9, 10] and even earlier [11]. Extensions of the Hannenhalli-Pevzner approach [19, 20] allowed the comparison of multi-chromosomal genomes, by modelling the biological processes of reciprocal translocation as well as chromosome fusion and fission.

During this time as well, the first methods of reconstructing ancestral gene orders were proposed. Using the breakpoint metric b, essentially the number of adjacent genes in one genome not adjacent, or at least not with the same sign, in the other, it became possible to infer the optimal ancestral genomes in a phylogeny [21]. And using the genomic distance d, El-Mabrouk devised an algorithm to reconstruct the ancestral form of present-day descendants of a tetraploidization, or whole genome doubling, event [22, 23].

One task that becomes disproportionately more difficult in comparing nuclear genomes, with typically 3×10^9 DNA base pairs in mammals, than with mitochondrial (10^4-10^5 base pairs) or chloroplast (typically 1-3 $\times$ 10^5) genomes is that of identifying the corresponding elements (homologs, orthologs, ancestrally related) to compare. Not only is it extremely difficult and tedious just to identify the genes in a genome, and to distinguish which of potentially many approximate copies of a gene in one genome corresponds to which in the the other, but most of the genome is not occupied by genes at all.

There are two types of response to this difficulty. One is to try to systematically identify conserved segments: homologous lengths of sequence in the two genomes, using alignment techniques and without regard to the genetics or function (i.e., whether

the segment contains genes, parts of genes, or regulatory elements), and then to treat these segments as ordered markers in the comparative analysis. This was first done in 2003 by Pevzner and Tesler [24] focusing on highly conserved (anchor) regions of sequence, attaining thresholds of length and similarity. At the same time Kent *et al.* [25], in building the UCSC genome browser [26], applied a new, nested alignment procedure over all regions of the genome sequence, which could then be analysed at various levels of resolution. Unfortunately, sequence-based approaches to comparative structural genomics not firmly anchored in genetic correspondences are not at all robust to changes in threshold or resolution parameters [27, 28].

The second way of responding to the difficulty in homology identification is simply to wait for the result of genome annotation, i.e., the systematic expert identification of genes and other sequence elements in all the genomes being compared. This task may take years to complete, a frustratingly long delay for those eager to do comparisons, but recent advances in automated or at least computer-assisted gene finding and annotation mean that relatively accurate gene identification is increasingly available and that homologies can be quickly established, enabling genomic distance calculations soon after sequence assembly.

The advances and challenges described in this thesis will be phrased in terms of the second approach. We will characterize genomes as gene orders, and often use these terms interchangeably.

The mathematical details of some of the work discussed here appears in [30], while the field of combinatorics and algorithms for genome rearrangement is the topic of a recent survey volume by Fertin *et al.* [31]. In the next section, we sketch various kinds of genomes and how they are formalized as well as the basic chromosomal rearrangement processes and how they are defined mathematically. This leads to various concepts of genomic distance.

The contributions of this thesis will be to both structural and functional genomics. Since few eukaryotic genomes are available as complete, "polished" sequences,

and since gene identification and ortholog identification are never completely accurate over whole genomes, the conditions for gene order rearrangement analysis, as developed by computer scientists and mathematicians, are rarely met. To be able to handle the bulk of genomic data, then, we must extend this analysis to more realistic types of genomic data. Thus, in Chapter 3, we discuss the genome rearrangement distance problem for incompletely assembled genomes, where the linear order on a chromosome may be fragmented into many smaller linear orders (contigs). We examine the consequences for gene order phylogenetics of treating each of these contigs as if it were a complete chromosome.

A prerequisite for this discussion is a sketch of gene order phylogenetic reconstruction. Rather than digress on this background subject in Chapter 3, we postpone it to a short Chapter 4, where we present the "median problem" as the basis for the reconstruction of genomes at the ancestral nodes of phylogenetic trees.

Since paired ends sequencing is now widely available, incompletely sequenced genomes are now more likely to be in scaffold form, rather than as sets of independent contigs. In Chapter 5, we examine the problem of comparing genomes in scaffold form. The problem becomes one of finding where in the scaffold-internal gaps to optimally insert "missing" genes suggested by comparative evidence.

Part of the goal in this thesis is to explore the functional genomic consequence of gene order rearrangement. We undertake this project in Chapter 6, by studying the relation between rearrangement breakpoints and changes in gene expression.

Chapter 2

Genomes, Distances and Evolutionary Time

In this section, we first will sketch various formalizations of the biological concepts of chromosome and genome, more specifically the mathematical abstractions of genome content, chromosome shape, gene sign and gene order. We introduce comparative genomics in terms of *breakpoints* and the *breakpoint distance*. Then, we will define evolutionary operations that affect genomes and discuss how different combinations of genome structure and permitted evolutionary operations result in computationally different models.

We introduce genomic rearrangement distances and explain how they apply to different kinds of genomes.

We then explore the dependance of expected genomic distance on time, or rather on the number of evolutionary steps under the constant rate-of-change hypothesis. As genomes diverge over time, measures evaluating the differences between these genomes increase. Even in the simplest model, this increase cannot be linear indefinitely over time, since any such measure will have a maximum value, typically the expected difference between two randomly permuted genomes.

2.1 Genomes

We start with a set G of distinct elements called *genes*. A gene g is represented by its two *ends*, its *head* g_h and a *tail* g_t. (In biochemical terms, the heads may correspond to the 5' ends of the genes and the tails the 3' ends, or vice-versa.) An *adjacency* is an unordered pair of gene ends, generally from different genes; a *genome* is a set of adjacencies on G, where no gene end is in more than one adjacency. A gene end that is not in any adjacency is called a *telomere*. Consider a gene g, together with any gene h having an end adjacent to an end of g (there may be 2, 1 or 0 such h), together with any other gene k having an end adjacent to the other end of h, and so on, accumulating genes by transitivity of adjacency. The subset of G thus constructed is called a *chromosome*. If a chromosome contains two telomeres it is a linear chromosome, if it contains no telomere it is circular. A genome with only linear chromosomes is called a *linear* genome.

A genome with only one chromosome is called *unichromosomal*; if it has more than one chromosome it is *multichromosomal*.

As well as the representation in terms of sets of adjacencies, a genome can also be represented as a set of strings, by writing the genes for each chromosome starting with one that has a telomeric end and adding genes according to the adjacencies of their ends. Each gene g whose tail is written first is considered to have positive polarity, while a gene whose head is written first has negative polarity ($-g$). In this way, unichromosomal genomes are equivalent to *signed permutations* by virtue of the head-tail polarity of the gene ends, irrespective of whether they are linear or circular. For each linear chromosome, there are two possible equivalent strings, according to the arbitrary chosen telomere. One string is obtained from the other by reversing the order and switching the signs of all the genes. For circular chromosomes, there are also two possible circular string representations, according to the direction in which the genes are traversed.

For example, consider the two signed permutation 1, 2, 3 and 1, -3, -2 representing two unichromosomal genomes. These are equivalent to the sets of adjacencies $\{\{1_h, 2_t\}, \{2_h, 3_t\}\}$ and $\{\{1_h, 3_h\}, \{3_t, 2_h\}\}$, respectively, where $\{1_t\}$ and $\{3_h\}$ are telomeres in the first genome and $\{1_t\}$ and $\{2_t\}$ are telomeres in the second. [1]

Although we have formulated genomes in terms of sets of distinct genes, in biological reality there are often many copies, identical or almost so, of the same gene in a genome. Incorporating this fact into the mathematics of genome comparison and genome reconstruction complicates the formulation of problems and inevitably worsens their complexity. For this thesis, therefore, we will keep to the single-copy genes case, and leave it to the reader to explore the voluminous literature (starting with [29]) that attempts to generalize to the multicopy case.

2.2 Breakpoints

For two genomes on the same set G containing n genes, denote $\{x, y\}$ to be an adjacency (as defined in Section 2.1) in one of the genomes but not the other. This is called a *breakpoint*. In the example in Section 2.1, $\{1_h, 2_t\}$ is a breakpoint, but $\{2_h, 3_t\}$ is not; it is an adjacency in both genomes.

Let a be the number of adjacencies in common in two genomes Π and Γ, and e be the number of telomeres in common. Then, the *breakpoint distance* is

$$d_{BP}(\Pi, \Gamma) = n - a - \frac{e}{2}. \tag{2.2.1}$$

This definition from [30], applies to the comparison of all the types of genomes mentioned in Section 2.2. It may differ from other definitions slightly, largely in terms of how it accounts for differing sets of telomeres.

[1] A slightly different convention for telomeres will be useful in Chapter 5.

2.3 The Operations

The classical genetics notions of inversion, transposition and reciprocal translocation of chromosome segments, as well as chromosomal fission and fusion, are formalized in such papers as those by Tesler [20], Yancopoulos *et al.* [32], and Bergeron *et al.* [33]. Briefly, using the string representation of a chromosome, e.g., $h_1 \cdots h_l$, where a pair of successive genes $h_u h_{u+1}$ are loosely termed an adjacency if their gene ends constitute an adjacency, we can illustrate:

- an inversion (implying change of sign, i.e., change of strand) of a chromosomal segment:
  $h_1 \cdots h_u \cdots h_v \cdots h_m \rightarrow h_1 \cdots -h_v \cdots -h_u \cdots h_m$, disrupting the two adjacencies $h_{u-1}h_u$ and $h_v h_{v+1}$,

- a transposition of a chromosomal segment:
  $h_1 \cdots h_u \cdots h_v \cdots h_w \cdots h_m$
  $\rightarrow h_1 \cdots h_{u-1}h_v \cdots h_w h_u \cdots h_{v-1}h_{w+1} \cdots h_m$, disrupting the three adjacencies $h_{u-1}h_u$, $h_{v-1}h_v$ and $h_w h_{w+1}$

- a reciprocal translocation between two chromosomes:
  $h_1 \cdots h_u \cdots h_l, k_1 \cdots k_v \cdots k_m$
  $\rightarrow h_1 \cdots k_v \cdots k_m, k_1 \cdots h_u \cdots h_l$, disrupting the two adjacencies $h_{u-1}h_u$ and $k_{v-1}k_v$,

- a chromosome fission:
  $h_1 \cdots h_v \cdots h_l \rightarrow h_1 \cdots h_v, h_{v+1} \cdots h_l$, disrupting the adjacency $h_v h_{v+1}$, and

- the fusion of two chromosomes:
  $h_1 \cdots h_l, k_1 \cdots k_m \rightarrow h_1 \cdots h_l k_1 \cdots k_m$.

The genomic distance is the minimum number of operations of these types (or some specified subset of types) required to transform one of the genomes being compared

into the other. The authors mentioned above also provide rapid algorithms for deriving the distance, given genomes composed of ordered chromosomes represented by the same n genes, markers or segments in the two genomes, assuming the strandedness, or reading direction, of each gene is known.

2.4 Rearrangement Distance

A double-cut-and-join (DCJ) is an operation acting on two adjacencies $\{p, q\}$ and $\{r, s\}$, deleting them and replacing them by $\{p, r\}$ and $\{q, s\}$ or by $\{p, s\}$ and $\{q, r\}$. Also it can act on an adjacency $\{p, q\}$ and a telomere r to produce the adjacency $\{p, r\}$ and a telomere q or the adjacency $\{q, r\}$ and a telomere p. It can also fuse two telomeres to create an adjacency or fission an adjacency to create two telomeres.

A DCJ can have the effect of inverting an interval of a genome, fission one chromosome into two, fusing two chromosomes into a single one, or producing a reciprocal translocation between two chromosomes. Two consecutive DCJ operations may result in a block interchange: two arbitrary segments of the genome exchange their positions, a particular case is that of a transposition, for which the two segments are contiguous. The DCJ operation is thus a very general framework. It was introduced by Yancopoulos *et al.* [32] and was simplified by Bergeron *et al.* [33].

The minimum number of DCJ operations needed to transform one genome into another (on the same set of genes) is the *DCJ distance* d_{DCJ}. It can be quickly calculated by defining a bipartite graph where the adjacencies and telomeres of one genome constitute the vertices on one side of the graph and the adjacencies and telomeres of the other genome are the opposing set of vertices. An edge is drawn between two vertices (from the two genomes) if they are both derived from adjacencies or telomeres containing a same gene end. Then, it is proved in [33], that

$$d_{DCJ} = n - c - \frac{i}{2} \tag{2.4.1}$$

where c is the number of cycles in the graph and i is the number of paths with an odd number of edges.

The reversal/translocation distance was introduced by Hannenhalli and Pevzner [19], and is equivalent to the DCJ distance constrained to linear genomes.

For a linear genome, a *linear* DCJ operation is a DCJ operation that results in a linear genome. This allows reversals, chromosome fusions, fissions, and reciprocal translocations. Other DCJs that create temporary circular chromosomes, are not allowed, so that transpositions (and other block interchanges) may require four operations, instead of two. Chromosomes fusions and fissions are particular cases of translocations in this framework. We call the minimum number of linear DCJ operations that transform one linear genome into another *RT distance* and we denote it by d_{RT}. This distance can be calculated rapidly with formulae analogous to (2.4.1), but which have rather complex form [19, 20].

2.5 The Relation between True and Inferred Distance

Even assuming that rearrangements occur at a relatively constant rate over time and are randomly positioned in the genomes, we have no simple, exact probability relationship between the actual number τ of rearrangements after a certain time has elapsed and the number of rearrangements d inferred by applying the genomic distance algorithms to compare the initial and the derived genomes [34, 35, 36]. We can, however, model the proportion of adjacencies that will be disrupted versus the proportion that will remain intact after τ random rearrangements. For each of the adjacencies in the original genome, the probability that it will remain undisrupted after τ rearrangements is $(1-\lambda/n)^{\tau}$ or approximately $e^{-\lambda\tau/n}$, where λ depends on the proportions of the various kinds of rearrangements in the model. Thus the number of

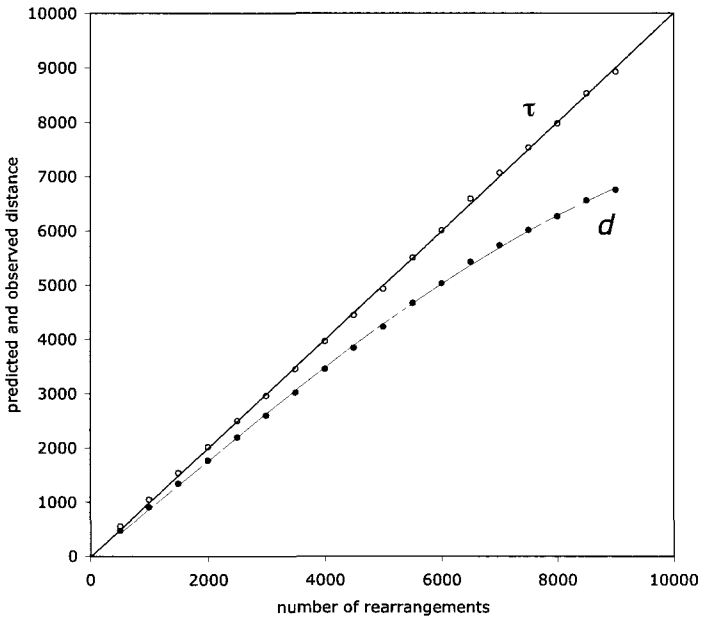

Figure 2.1: Predicted (curve) and observed (dots) values of genomic distance d, and inferred (open dots) values of $\hat{\tau}$ versus true (diagonal line) values.

disrupted adjacencies, the breakpoint distance, will be approximately $n(1 - e^{-\lambda\tau/n})$.

In analogy to the closely related breakpoint distance, we suggested [52] that as a first approximation the expectation

$$\mathrm{E}(\frac{d}{n}) \approx 1 - e^{-\lambda\tau/n}, \tag{2.5.1}$$

where n is the number of genes in both genomes, d is the inferred rearrangement distance and λ is a constant. If we knew λ, we could estimate the true number of rearrangements τ using

$$\hat{\tau} = -\frac{n}{\lambda} \log\left(1 - \frac{d}{n}\right). \tag{2.5.2}$$

In fact, the relationship between the actual and inferred numbers of rearrangements (not shown) deviates considerably from the one-parameter model in (2.5.1) both for small and large τ. We thus add a quadratic correction to the linear term in

the exponent, so that the model becomes

$$\mathrm{E}(\frac{d}{n}) \approx 1 - e^{-\lambda_1 \tau/n - \lambda_2 (\tau/n)^2}, \tag{2.5.3}$$

in which case the estimate of τ becomes

$$\hat{\tau} = \frac{n}{2\lambda_2}\left(-\lambda_1 + \sqrt{\lambda_1^2 - 4\lambda_2 \log\left(1 - \frac{d}{n}\right)}\right) \tag{2.5.4}$$

To estimate the parameters λ_1 and λ_2, we simulated 100 pairs of genomes with almost 9000 genes and τ up to 9000 random rearrangements to derive one genome from the other. We then used a DCJ algorithm [33] to obtain an estimate of $\mathrm{E}(d)$ from the genomes.

Figure 2.1 shows the relationship between τ and both $\mathrm{E}(d)$ and $\hat{\tau}$, using the values $\lambda_1 = 0.846$ and $\lambda_2 = 0.576$, found by a least sum of squares criterion applied to the set of τ and $\hat{\tau}$ values. The way τ and d are normalized means that the parameters should not be very sensitive to n.

We found that for the large values of n we studied, the model in (2.5.3) fits the simulated data, including for large and small values of τ, better than the purely curve-fitting model in [36], which also has two parameters.

2.5.1 Comparison with previous work

This analysis resembles the "empirical" approach in Ref [36] to the relationship between d and τ, which also makes use of two parameters, except that our starting point is the intuitive development leading to Eq 2.5.1 at the beginning of this section, whereas Ref [36] takes a purely curve-fitting approach from the outset. These authors propose that

$$\mathrm{E}(d) \approx n \min\left(\frac{\tau}{n}, \frac{\frac{\tau}{n}^2 + b\frac{\tau}{n}}{\frac{\tau}{n}^2 + c\frac{\tau}{n} + b}\right). \tag{2.5.5}$$

They find the parameter values $b = 0.6$ and $c = 0.46$ fit simulated data well when n is in the region of a few dozen, justifying the use of the normalized variable $\frac{\tau}{n}$. When n is a hundred times greater, however, neither these nor any other parameter values allow the model in (2.5.5) to fit the simulated values of d, without the introduction of additional parameters.

Chapter 3

Contigs

3.1 Introduction

While the increasing pace of genome sequencing is adding phylogenetic breadth to the inventory of species available for comparative genomics, the sequencing goal for many of these species is not to produce completely assembled genomes. Instead the published and archived data remain in draft form, with all or many of the numerous *contigs* (continuous sequence of much smaller length than a chromosome) not assigned chromosomal locations, and there are often no resources allocated to further polishing. The price paid for increasing phylogenetic scope in genome sequencing is thus the decreasing sequencing quality for each genome.

While such data may be adequate for many types of comparative genomic studies, they are not directly usable as input to genome rearrangement algorithms. These algorithms require whole genome data, i.e., complete representations of each chromosome in terms of gene order, conserved segment order, or some other marker order, in order to calculate the rearrangement distance d between two genomes. Items whose chromosomal location is unknown cannot be part of the input.

This Chapter deals with gene order-based phylogeny. As such, we will view

contigs as linear array of genes that are contiguous fragments of chromosomes. The question we ask here: Is there any way to use genome rearrangement algorithms to compare genomes available in draft form only? One elegant answer was provided by Gaul and Blanchette [50] for the comparison of two genomes. Other approaches have also been investigated [51]. These methods construct a number of intermediate structures before or during the actual comparison of the genomes. Since in the first instance we will be using distance matrix methods for phylogenetic analysis, these intermediate structures are largely irrelevant; we need distances and not the detailed reconstruction of the structures used in calculating the distance. For these purposes, involving more than two genomes, our suggestion is to use the contigs directly in the rearrangement algorithms as if they were chromosomes. This introduces a number of biases, such as increasing the distance to accommodate the count of extra fusion/fission operations as described in Section 2.3 necessary to compare genomes with different numbers of chromosomes. This bias and other problems with rearrangement distances in general and with contig-based distances in particular must be corrected during the construction of a distance matrix to input into a phylogenetic analysis.

We apply our methods to data originating mostly in the 12-genome *Drosophila* project [15]. We compare ten *Drosophila* genomes with two other dipteran genomes and two outgroup insect genomes. We discuss the data that allows us to construct the contigs in Section 3.2.

In Section 3.3, we performed simulations to estimate the parameters of our model described in Section 2.5 that models the behaviour of the genomic distance as a function of evolutionary time. In Section 3.4 we study the case where one of the two genomes being compared is fully assembled and the other is in contig form. We discuss how the formal mechanisms (Section 2.3) for handling chromosome fusions carries over to the concatenation of contigs to form chromosomes during the application of the genome rearrangement algorithms. Simulations are used to understand the

consequences on evolutionary time inference of using incomplete assemblies. The ideas developed there are then extended to the more complex case where both genomes are fragmented into contigs, in Section 3.5. We can then construct a matrix of corrected evolutionary divergence times between all pairs of genomes in the database and carry out a phylogenetic analysis of the fourteen genomes, in Section 3.6. For the *Drosophila* data only, we compare this phylogeny to one generated by gene order reconstruction algorithms applied to the genomes without any corrections for contigs or distance-time nonlinearities.

3.2 The data

One of the difficulties in using gene order rearrangement algorithms is the lack of curated gene order databases for the higher eukaryotes with sequenced genomes. Because the gene identification and homology identification has already been done in [15], we use a carefully constructed inventory of neighbouring gene pairs (NGPs) in ten *Drosophila* species and four outgroup insects, rather than raw contig data. A.J. Bhuktar provided us with a file listing all NGPs and the genomes in which they

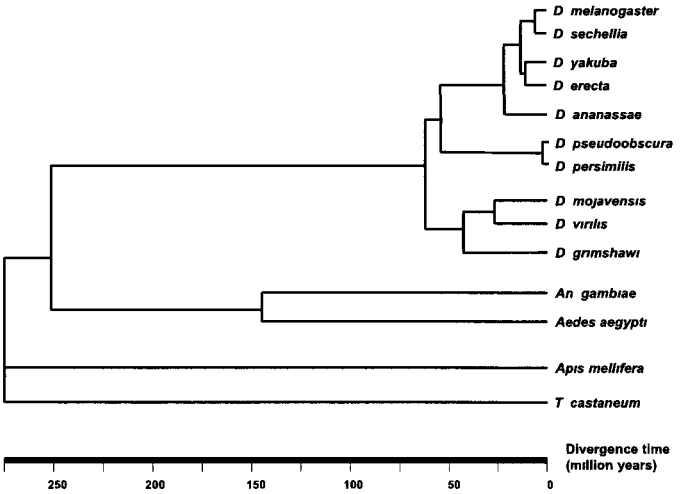

Figure 3.1: Phylogeny of Drosophila and outgroups abstracted from the literature, with divergence times.

appear. By the time of writing, the assembly of these genomes has progressed, but for our purposes, i.e., to show how to handle genomes in contig form, the original data set is preferred.

species (abbreviation)	genes	contigs
D. melanogaster (Dmel)	8867	6
D. sechellia (Dsec)	8851	66
D. yakuba (Dyak)	8809	30
D. erecta (Dere)	8866	9
D. ananassae (Dana)	8844	40
D. pseudoobscura (Dpse)	8778	51
D. persimilis (Dper)	8779	87
D. virilis (Dvir)	8855	32
D. mojavensis (Dmoja)	8853	14
D. grimshawi (Dgri)	8801	35
Anopheles gambiae (Anoph)	6168	6
Aedes aegypti (Aedes)	6318	869
Apis mellifera (Apis)	4898	702
Tribolium castaneum (Trib)	5647	89

Table 3.1: Number of contigs constructed for each genome

We abstracted best-judgement divergence times among the genomes from a number of somewhat contradictory publications [16, 17, 18] available when we carried out our research, as summarized in Figure 3.1. For *Drosophila* this turns out to be the same as the tree now widely accepted (see [14, 12]), except that the time scale is now thought to be 40 % shorter than in Figure 3.1. This is of little pertinence to us since our calculations are all in terms of numbers of rearrangements, not absolute time, which as a first approximation is merely assumed to be proportional to the rearrangement numbers. The position of the mosquitos *Anopheles gambiae* and *Aedes aegypti* relative to *Drosophila* is uncontroversial within the Diptera, while the uncertainty of the branching order of the Hymenoptera represented by *Apis mellifera* and Coleoptera represented by *Tribolium castaneum* leads us to postulate a trichotomy giving rise to these three orders of holometabolous insects.

Bhutkar *et al.* [13, 14] have already used the NGP data for a phylogenetic analysis of *Drosophila*, inferring phylogenies, rearrangements and synteny blocks, but our use of the NGPs here is different. It is simply to reconstruct the gene orders in the contigs; we wish to create a data set for testing our method for gene order-based phylogenetics from genomes in contig form.

For each genome, we constructed contigs by amalgamating overlapping NGPs. Whenever we arrived at a gene in only one NGP in a genome, this terminated a contig. Our reconstruction then, does not necessarily correspond completely to the original contigs in the 12-genome *Drosophila* sequencing project [15], but this has little importance for our work – how the genomes are fragmented into contigs, and into how many, is a methodological question that depends on laboratory resources and techniques and has nothing directly to do with how the genome has evolved. Both contig ends and rearrangement breakpoints may be enriched for duplicated sequence, but this indirect connection has no consequence for the problem we are attacking.

Table 3.1 gives the number of contigs reconstructed for each genome. Note that the reconstructions of *D. melanogaster*, *D. erecta* and *An. gambiae* reflect the complete, or almost complete, assembly of these genomes.

3.3 Genomic distance and evolutionary time

3.3.1 Simulation-based estimates

We refer to our model described in Section 2.5. To estimate the parameters λ_1 and λ_2 found in Eq. 2.5.4, we simulate pairs of genomes with $n = 8867$, the maximum number of genes used in our *Drosophila melanogaster* comparisons, and τ up to 9000 random rearrangements to derive one genome from the other. It is well known (e.g., [14]) that rearrangements in *Drosophila* are almost exclusively inversions within Muller elements (chromosome arms), although this does not pertain to other insects. We

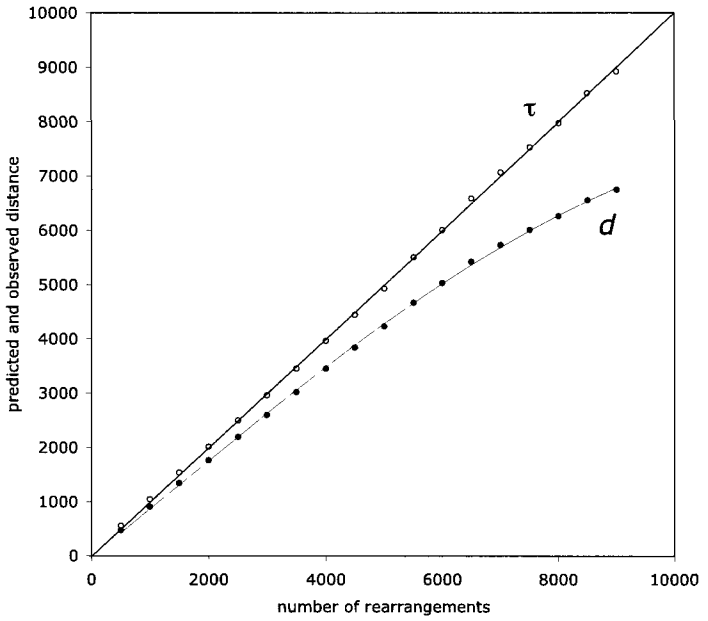

Figure 3.2: Comparison of model and simulations for genomic distance d and of true value and estimated τ. Predicted (curve) and observed (dots) values of d, and inferred (open dots) values of $\hat{\tau}$ versus true (diagonal line) values. See Sections 2.5 and 3.3.1.

carry out our simulations accordingly, with about 99.8% inversions, and only a few reciprocal translocations. We use a DCJ (double-cut-and-join) algorithm [32, 33] to calculate d from the genomes. This is repeated 100 times, and d averaged, to estimate $\mathrm{E}(d)$.

Figure 3.2, repeated from Figure 2.1, shows the relationship between τ and both $\mathrm{E}(d)$ and $\hat{\tau}$, using the values $\lambda_1 = 0.846$ and $\lambda_2 = 0.576$, found by a least sum of squares criterion applied to the set of τ and $\hat{\tau}$ values. The way τ and d are normalized means that the parameters should not be very sensitive to n, though we do not study this here, since all the experimental genomes are of comparable size.

Note that if τ is very small, our model predicts $\mathrm{E}(d) \approx \lambda_1 \tau$ instead of $\mathrm{E}(d) \approx \tau$, though this bias occurs at a scale not visible in Figure 3.2.

3.4 The effect of genome fragmentation

3.4.1 Contigs as chromosomes

Consider one completely assembled genome B and another, A, in contig form only. The basic idea is that if we treat each contig as a chromosome, a rearrangement algorithm will automatically carry out a number of contig concatenations or "fusions" as described in Section 2.3 to assemble the χ_A contigs in A into a small number of inferred chromosomes equal to the number χ_B in B, in calculating d. At the same time it will find other rearrangements, but we know that the number of fusions required will be at least the difference between the number of contigs in A and the number of chromosomes in B. Indeed, in almost all optimal rearrangement scenarios there will not be both fusions *and* chromosome fissions; thus, when we use a rearrangement algorithm to compare a genome A in contig form with an assembled genome B, obtaining a preliminary distance d', it may seem appropriate to correct this to

$$d = d' - (\chi_A - \chi_B), \tag{3.4.1}$$

where we assume $\chi_A > \chi_B$.

This argument holds independent of the other details of the optimal rearrangement scenario, for which there may be many for a particular data set.

3.4.2 Contig fusion and $\frac{d\mathrm{E}(d)}{d\tau}$

We cannot simply substitute correction Eq 3.4.1 into Eq 2.5.2 or 2.5.4 to estimate τ. Even if the number of contig fusions is exactly $\chi_A - \chi_B$, we know that these fusions are done by the inference algorithm in such a way as to minimize the total d', including inversions and other operations. To take account of this, we should only remove a proportion α of $C = \chi_A - \chi_B$ from d'. How large a proportion? **The natural**

hypothesis is that the effect of the C fragmentation operations, which have exactly the same properties as chromosome fission operations, and which create C extra contigs, should have the same effect as the addition of C of any other types of rearrangement to the genome. In any model, such as Eq 2.5.3, where we relate d and τ as if they were real variables (i.e., not just integer variables), the expected rate of increase of d with τ is the derivative $\frac{dE(d)}{d\tau}$. Thus increasing τ by C should increase d by approximately $\frac{dE(d)}{d\tau}C$.

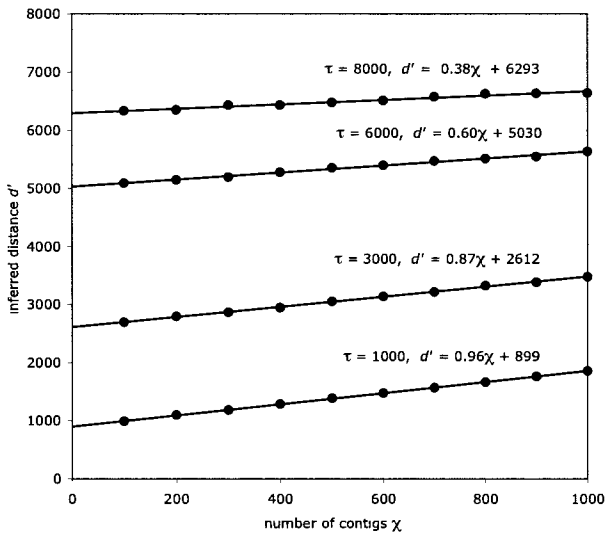

Figure 3.3: Effect of genome fragmentation on genomic distance. For genomes generated by $\tau = 1000, 3000, 6000$ or 8000 rearrangements, broken into $\chi = 100, 200, \ldots, 1000$ contigs: the relationship between uncorrected genomic distance d' and χ, with equations of trend lines.

To verify this hypothesis, we undertook a series of simulations, starting from an initial genome B containing 8867 genes in $\chi_B = 6$ chromosomes, generating 100 rearranged genomes, each through τ random rearrangements applied to B to produce a new genome, and each then fragmented into χ_A contigs. This was repeated for a range of values of τ and χ_A.

The average results for d' are summarized on Figure 3.3. First the linearity of the response to increasing χ_A is clear, at least in the range studied $\chi_A < 1000$, indicating

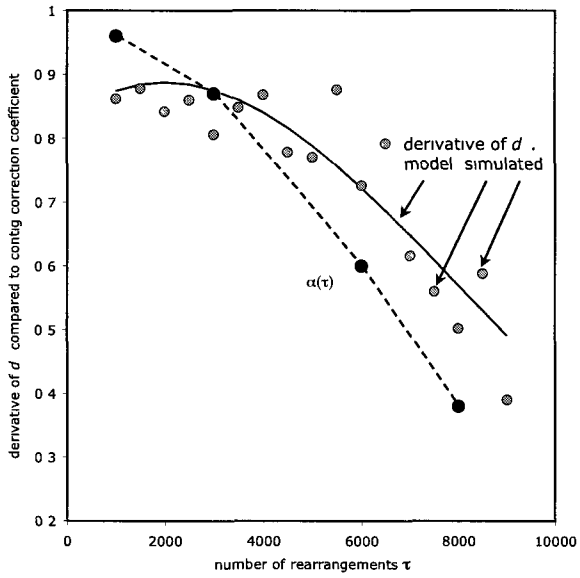

Figure 3.4: The parameter α as a function of τ (large dots and dotted line), with values taken from the coefficient of χ in the trend lines in Fig. 3.3. Solid line represents the derivative of E(d) in Eq 2.5.3 and Fig. 3.2, while the shaded dots represent the simulated values of the derivative, calculated from the data presented in Fig. 3.2. Calculated and simulated derivatives for small values of τ biased downwards, as discussed in Section 3.3.1.

that Eq 3.4.1 should be replaced by

$$d = d' - \alpha(\tau)|\chi_A - \chi_B|, \tag{3.4.2}$$

where $\alpha(\tau)$ is a decreasing function of the number of rearrangements τ. As can be seen from Fig. 3.4 this decrease approximately parallels the theoretical derivative of d, but is somewhat steeper. For purposes of interpolation, we fit $\alpha(\tau)$ with a quadratic function $1 - 0.0276(\tau/1000) - 0.0063(\tau/1000)^2$ by minimizing the sum of squares over the four values of τ where we have calculated α.

Given d', then, we can solve Eqs 2.5.3 and 3.4.2 simultaneously to find τ and d, since $n, \lambda_1, \lambda_2, \chi_A$ and χ_B are known, as is the dependence of α on τ. In practice, this can be done by successive iteration of Eqs 2.5.4 and 3.4.2, which converges rapidly,

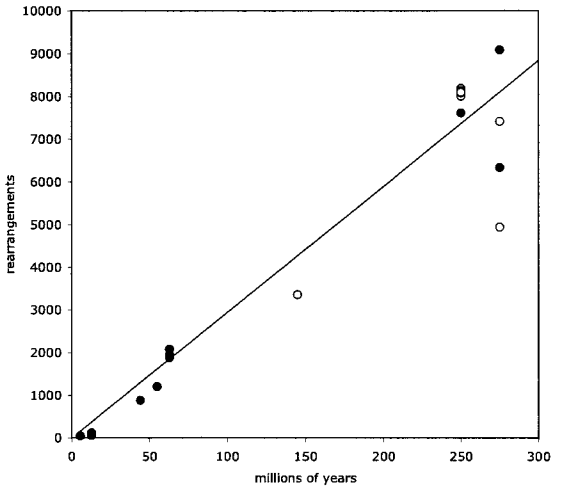

Figure 3.5: Divergence, in total number of genome rearrangements, estimated from genomic distances through Eqs 2.5.3 and 3.4.2, compared to divergence times abstracted from the literature. Estimates of τ for *D. melanogaster* compared with 13 other genomes (solid dots). Estimates for *Anopheles gambiae* with 13 other genomes (open dots). Line represents least squares fit to all points.

initializing with, for example, $\tau_0 = d'$.

Applying this to the comparison of the completely assembled *D. melanogaster* genome with each of the other 13 genomes, and to the comparison of the completely assembled *Anopheles gambiae* genome with each of the other 13 genomes, gives the results shown in Figure 3.5. The high degree of scatter at higher divergence times reflects both the uncertainty of the divergence dates and the inhomogeneity of rearrangement rates both between the fruit-fly and mosquito families within the dipteran order and among the three orders in the class Insecta represented in these data.

3.5 The case of both genomes in contig form

When we compare two incompletely assembled genomes A and B, we may still wish to remove some quantity depending on χ_A and χ_B from d' to account for the fusions (and/or fissions), but this is not as easy to analyze, for two reasons. One is that

we are not comparing a fragmented genome to a complete genome, so we can no longer consider this correction as a way of using the assembled genome as a guide for reconstructing the fragmented genome, simultaneous with the distance calculation. The second problem is that there is no obvious way, within the formula, of combining (adding, multiplying, ...) the number of contigs in one genome with the number in the other. This reflects the lack of intuition on how the contigs increase the distance (because of artificial fusions and fissions) on one hand, and how they decrease it (by multiplying the number of economical but false rearrangements) on the other hand. These reasons lessen the intuitive appeal of the kind of correction we used in the previous section. Nevertheless, we can try to find an appropriate correction using the same simulation approach as in the previous sections.

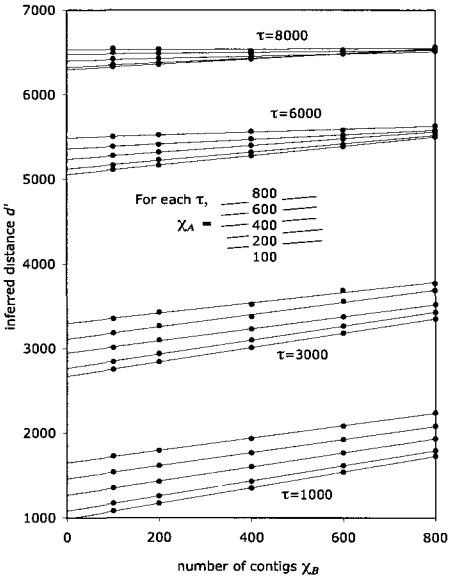

Figure 3.6: Effect of fragmentation of both genomes on genomic distance. For genomes A and B separated by $\tau = 1000, 3000, 6000$ or 8000 random rearrangements, broken into $\chi = 100, 200, 400$ or 800 contigs: the relationship between uncorrected genomic distance d', χ_A and χ_B, with trend lines for each χ_A connecting the values of d' for a range of χ_B.

We simulated 50 runs each of two genomes of size $n = 8867$ separated by $\tau =$

1000, 3000, 6000 and 8000 random rearrangements as before, but with both genomes independently and randomly fragmented into $\chi = 100, 200, 400, 600$ or 800 contigs, i.e., $5 + \binom{5}{2} = 15$ pairs of contig configurations for each degree of rearrangement. We applied the DCJ algorithm and calculated the mean d' for each configuration. The results are summarized in Figure 3.6.

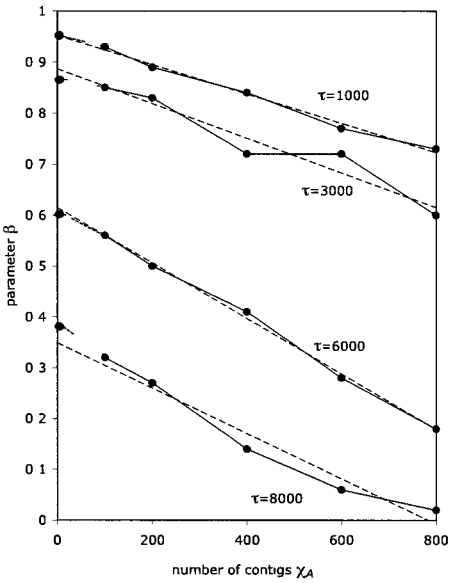

Figure 3.7: The coefficient β of the linear dependence of d' on χ_B, as a function of χ_A. Dotted segments connect $\beta(\tau, 100)$ to $\beta(\tau, 0) = \alpha(\tau)$ from Figure 3.3. Dashed line is the linear trend line, not taking account of $\beta(\tau, 0)$.

We observe in Figure 3.6 that for fixed τ and χ_A, the response of d' to increasing χ_B is systematically linear. This is clear up to $\tau = 6000$ and only starts to break down for $\tau = 8000$ and $\chi_A \geq 600$, where examination of the data on an expanded scale shows that d' actually decreases somewhat initially, then increases, as χ_B increases (not discernible in Figure 3.6). The linear rate of increase of d', plotted as $\beta(\tau, \chi_A)$ in Figure 3.7, is the same as the $\alpha(\tau)$ in Figure 3.3 for small values of χ_A. In fact, d' shows the same linear increase as a function of $\chi_A + \chi_B$ up to moderate values of this sum, as in Figure 3.9, depending on τ, after which the rate of increase drops off

markedly. As both genomes are increasingly fragmented, then, the mutational process becomes saturated so that inference of distance can only be very approximate.

Nevertheless, as with the case of only one genome fragmented into contigs studied in Section 3.4, we can infer d and τ from observed values of d' by solving Eq 2.5.4 simultaneously with

$$d = d' - \alpha(\tau)\chi_A - \beta(\tau, \chi_A)\chi_B, \tag{3.5.1}$$

where $\beta(\tau, \chi_A) = \alpha(\tau) - (.00027 - .00003\tau)\chi_A$, and where the coefficient of χ_A is estimated by a least squares fit to the slopes of the four trend lines in Figure 3.6 (bottom).

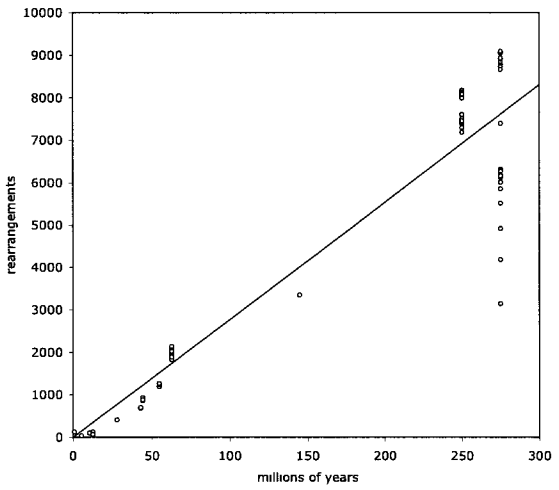

Figure 3.8: Divergence, in total number of genome rearrangements, estimated from genomic distances through Eqs 2.5.3 and 3.4.2, compared to divergence times abstracted from the literature. Pairwise comparison of all pairs of 14 genomes, as discussed in Section 3.5. Line represents least squares fit.

Plotting the values of τ inferred from Eq 3.5.1 against values extracted from the literature produced the results shown in Figure 3.8. Note that Eq 3.5.1 is asymmetric with respect to A and B, which could have consequences for the analysis in the next section. However, this is avoided if A always denotes the more fragmented of the two genomes.

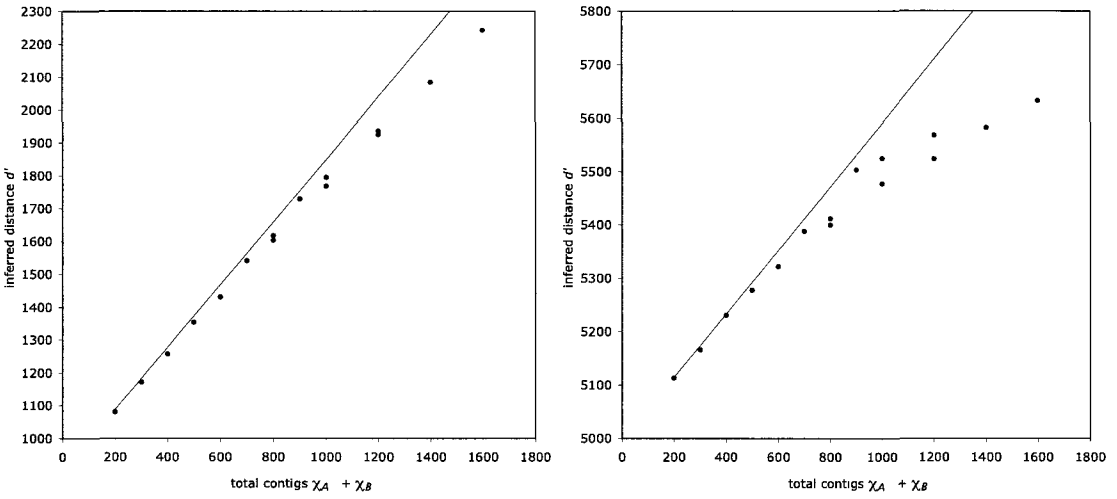

Figure 3.9: Dependence of d' on the total number of contigs in the two genomes, for $\tau = 1000$ (left) and $\tau = 6000$ (right). Straight lines represent $d' = d + \alpha(\tau)(\chi_A + \chi_B)$ where $\alpha(\tau)$ is as in Figure 3.3.

3.6 Phylogeny.

3.6.1 Using a distance matrix.

If we input the inferred pairwise values of τ into a neighbour-joining routine, we produce the phylogeny in Figure 3.10. When this is compared to Figure 3.1, the only structural difference is at one node where we see *D. sechellia* branching off just before *D. melanogaster* rather than branching off together as sister groups. More striking is the long branch leading to the *Drosophila* group, suggesting a rapid rate of evolution at the moment of divergence from other Diptera. Note that using the uncorrected matrix of d' as input to neighbour joining does not show this rate effect as clearly as τ and also introduces other structural errors into the phylogeny.

3.6.2 Phylogeny with reconstruction of ancestral genomes.

While the use of neighbour-joining on a distance matrix is convenient and rapid, it does not infer anything about the ancestral genomes in the resulting phylogeny. Tools

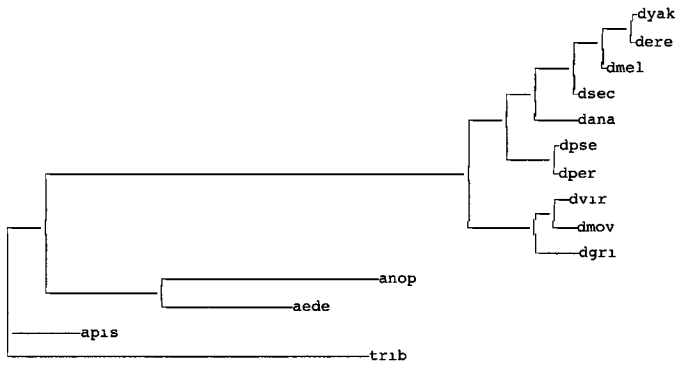

Figure 3.10: Neighbour-joining phylogeny based on matrix of inferred number of rearrangements τ.

are available, however, for inferring these ancestral genomes, given the topology of the tree.

As schematized in Fig. 3.11 and as will be discussed in Chapter 4, we solved the gene-order median problem at each node of the ancestral tree, using the path groups approach [67], iterating over all ancestral nodes of the tree until convergence. We did this separately for the trees in Figs. 3.1 and 3.10, using genomes containing the 8500 genes in common among all the *Drosophila* species, ignoring the outgroups for this analysis. This method attempts to minimize the sum of d over all branches. The path groups median is constructed by greedily building three "breakpoint graphs" simultaneously, relating the genome under construction M to each of A, B, and C in Fig. 3.11 (top), and attempting to maximize the total number of cycles over the three separate graphs. In the last iterations, the construction employs a look-ahead routine to increase accuracy, at the cost of increasing individual median calculations from a minute or so with 8560 genes, to an hour or two depending on how different the three genomes are.

The *Drosophila* part of the tree in Fig. 3.10 proved to be substantially less costly in terms of rearrangement operations (3862 rearrangements versus 3872 for the tree in Fig. 3.1), showing that the incorrect early branching of *D. sechellia* in Fig. 3.10 is

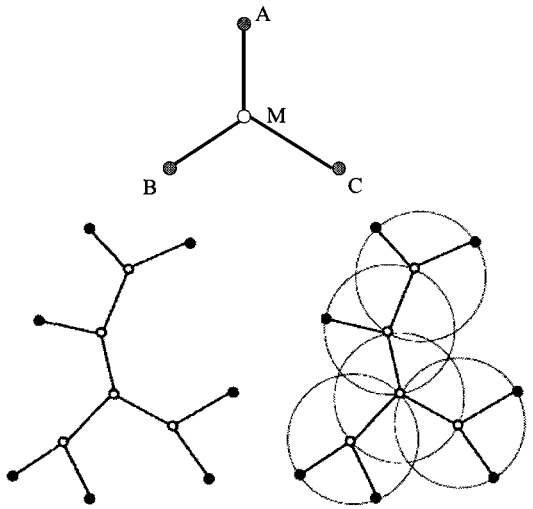

Figure 3.11: (top) Median problem: given genomes A, B, C, find M such that $d(A, M) + d(B, M) + d(C, M)$ is minimized. (left) Example of unrooted phylogeny with given present-day genomes at terminal nodes (dark dots) and genomes to be inferred at the ancestral nodes (white dots). (right) Inference of genomes at ancestral nodes found by iterating through the ancestral vertices, solving a median problem at each step.

inherent in the gene-order data and is not an artefact of the distance correction.

The rearrangement analysis was carried out on the raw data, namely the genomes assembled or partially assembled from the NGP data. The only preprocessing was to discard the small number (200-300 per genome) genes not in common to all genomes. The median program contains a bias towards fusions in the direction of the median, though this does not bias the total cost of the median analysis. Thus the ancestral genomes contain relatively few contigs.

How can we validate the rearrangement analysis? If we knew the gene order of the ancestral *Drosophila*, we could compare the reconstructed ones with it. Data provided in Additional file 7 in Ref [13] include the 1000+ syntenic blocks that Bhutkar *et al.* were able to reconstruct using conservative criteria based on 12 *Drosophila* genomes. Treating these blocks as contigs of the ancestral genome "A", we calculated the rearrangement distance between A and each of our 10 data and 8 ancestral genomes (reconstructed according to Fig. 3.1), after removing from our genomes the several thousand genes absent in A. The distance in Figure 3.12 is plotted against elapsed time from A as in Fig. 3.1. Despite the large amount of noise in the analysis, largely due to large number of contigs in A, it is clear that our reconstructed ancestors tend to be closer to A than are the data genomes. We may conclude that the rearrangement phylogeny is reconstructing aspects of ancestral gene order that are not apparent in all the individual data genomes.

3.7 Conclusion

We have developed a principled approach to correcting genome rearrangement distance when comparing genomes in contig form. Features of this include:

- A model for the τ—d relationship motivated by an intuitive negative exponential connection between asymptotic genomic distance ($\mathrm{E}(d) \nearrow n$) and adjacency

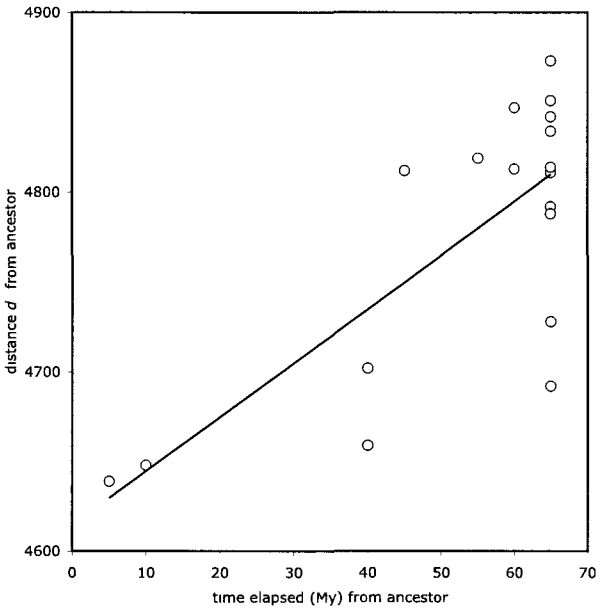

Figure 3.12: Rearrangement distance between Bhutkar *et al.* *Drosophila* ancestor [13] and 10 data (vertically aligned on right edge of graph) and 8 reconstructed ancestral genomes. Least squares line shown.

retention, including an empirically motivated quadratic correction term to improve the fit to simulated values.

- A reasoned procedure for subtracting artificial fusions and fissions due to the fragmentation of one or both of the genomes into contigs.

- The discovery and quantitative characterization of the linear relation between the uncorrected distance and the number of contigs, when only one or both of the genomes are fragmented into contigs. These linearities hold for a wide range of τ, up to 6000 for genomes of size around $n = 9000$, and up to $\chi = 1000$ contigs, though saturation is more of a problem when both genomes are highly fragmented.

- Improved phylogenetic reconstruction for a data set on 14 insect genomes. We recovered a tree that accurately reflects almost all the phylogenetic information

extracted from the literature, and pinpointed a period of evolutionary acceleration on one lineage.

- A validation approach based on the reconstruction of uncorrected ancestral genomes using gene-order medians biased towards contig fusion operations.

As argued in Section 3.3, the values of the parameters λ_1 and λ_2 are not likely to be very sensitive to n, especially for n in the thousands, since the model relates the normalized variables τ/n and d/n. Nor should they depend on details of the rearrangement model such as the number of chromosomes or the proportions of different types of rearrangement, assuming the latter are naturally weighted as in the double-cut-and-join framework. Thus, the values we have estimated through simulation for λ_1, λ_2 and α (considered as a function of τ/n) should hold for a range of data sets. This stability reassures us that our methods should be widely applicable beyond the *Drosophila* data we have used, but only partly mitigates the main shortcoming of this and other models such as in Ref [36], namely that they are not analytically derived. Thus, the mathematical foundation of probability models and statistical analyses of genomic problems like the one addressed here would benefit more from advances like those in Ref [35] than by further characterization of empirical models such Eq 2.5.3.

Chapter 4

The Median Problem

4.1 Introduction

In Section 3.6.2, I mentioned use of gene order "median" as the key technique at the heart of ancestral genome reconstruction. Rather than discuss the necessary background on this topic then, which would have a distracting discussion, I postponed it to this short chapter. the material in the next Section should then be considered as technical background rather than new contributions.

4.2 The Median and the Small Phylogeny Problem

The evolution of species, especially eukaryotic species, is most often represented by a phylogenetic tree. The problem of reconstructing or inferring a tree from data on present-day species may be conceptually (and often methodologically) separated into two parts. The *large* phylogenetic problem is one of finding the topology, or branching pattern, of the tree connecting the given species represented by the leaves, or terminal nodes, of the tree. The *small* phylogenetic problem is the inference, for a given tree, of the nature or content of the ancestral species identified with each of the non-terminal

nodes of the tree based on the nature or content of the given species associated with the leaves of the tree. In this section we will deal with the small problem in the case where the data on the present-day species are the orders of the genes on their chromosomes.

There are a variety of aspects of gene order that can be rapidly reconstructed, e.g., in [37, 38, 39], while monitoring the linearity of the reconstructed chromosomes through maintenance of bandwidth or a PQ-tree structure.

Here we will concentrate on solving the global problem by minimizing total branch length over a phylogeny while reconstructing optimal ancestral gene orders. This is basically the Steiner problem for the genome rearrangement distance over the set of all genomes having the same set of genes. This approach has been recently proven to be superior to local techniques when confronted with a manually validated ancestor gene order [39]. Formally, let $\mathcal{P}$ be a phylogeny where each of the N_t terminal nodes is labelled by a known gene order on the same n genes, and let d be a metric on the set of gene orders. Each branch of $\mathcal{P}$ may be incident to at most one terminal node and at least one of the N_a ancestral nodes. Each non-terminal node is of degree at least three. We want to reconstruct $R = (G_1 \ldots, G_{N_a})$, a set of gene orders at the ancestral nodes that minimize

$$L(R) = \sum_{\text{branch } XY \in \mathcal{P}} d(XY). \tag{4.2.1}$$

We use a hill-climbing procedure to find a local optimum for $L(R)$. This is illustrated in Fig. 4.1, repeated from Fig. 3.11. The archetypical (unrooted) phylogeny has three or more leaves and exactly one non-terminal node, as on the top of the figure. The problem becomes that of reconstructing a single gene order M, the sum of whose distances to the given gene orders is minimal. This problem has a relatively long history, with an early algorithm [40] for the breakpoint median based on d_{BP}. Technical speedups were described by Cosner *et al.* [5] and incorporated into the

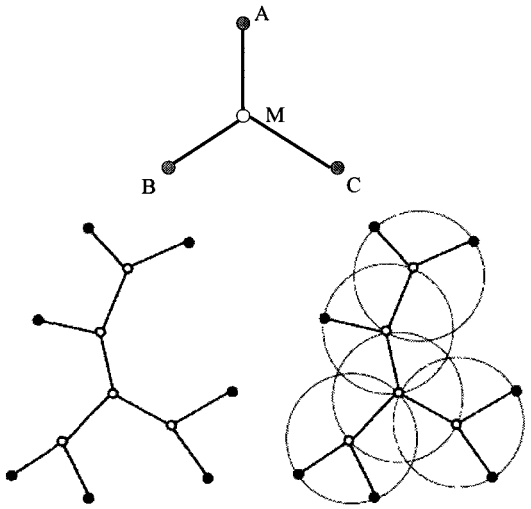

Figure 4.1: (top) Median problem: given genomes A, B, C, find M such that $d(A, M) + d(B, M) + d(C, M)$ is minimized. (left) Example of unrooted phylogeny with given present-day genomes at terminal nodes (dark dots) and genomes to be inferred at the ancestral nodes (white dots). (right) Inference of genomes at ancestral nodes found by iterating through the ancestral vertices, solving a median problem at each step.

GRAPPA software [41]. Siepel [42] and Caprara [43] gave exact median algorithms for small instances of d_{RT} and Bourque [44] released a heuristic web application for this version of the problem. Much progress has been made recently on exact algorithms capable of handling large or moderate size genomes [45, 46] for d_{DCJ}.

For most formulations, in terms of different kinds of genome and different distances, the median problem is known (or thought to be) to be NP-hard; recently, however, for the case of breakpoint distance on multichromosomal genomes not restricted to be linear, Tannier *et al* [30] have given a polynomial-time algorithm, and this has been implemented [47] as a rapidly executing program.

Focusing on the more general small phylogeny problem with more than one ancestral node, the current heuristic strategy is based on the ability of the median algorithm to achieve a fairly accurate solution in a reasonable time on a large proportion of instances. As illustrated at the bottom of Fig. 4.1, the phylogeny at the left is decomposed on the right into a set of overlapping median configurations, with one non-terminal, i.e., ancestral, node as median, and all its (three or more) co-linear nodes, terminal or non-terminal. The heuristic consists of solving each of the median problems in turn, updating the median at each step only if it diminishes the sum of the lengths of the branches incident to the median, and iterating. This eventually converges to a local minimum. The quality of the solutions may depend on the initialization of the ancestral gene orders [21], e.g., by random gene orders, or by copying some of the present-day gene orders to the ancestral nodes. It may also depend on various techniques for escaping from local minima [48].

Chapter 5

Scaffolds

5.1 Introduction

As mentioned in Chapter 3, the dramatic drop in the expense of genome sequencing has two somewhat contradictory effects on comparative study of gene order. On one hand it greatly increases the range of genomes available for genomic analysis, including comparative studies and phylogenomics. On the other hand, however, it encourages the final release of the genomes in unfinished (*standard* or *high-quality* draft) form, since the cost of finishing has not decreased at nearly the same rate as the cost of random sequencing [60]. The use of draft genomes makes many analyses and interpretations tentative and prone to error, and leads to particular problems in the comparative study of gene order. Many algorithms for studying genome rearrangement require whole genome data, i.e., complete representations of each chromosome in terms of gene order, conserved segment order, or some other marker order, in order to calculate the rearrangement distance d between two genomes. Items whose chromosomal location is unknown cannot be part of the input. As mentioned in Chapter 3, this puts the many "draft" genomes outside the scope of currently available comparison technology, although these data are suitable to most of the other goals of genomics.

5.1.1 Strategy

To overcome this hindrance to the exploitation of much of the genome sequence data produced now and in the future, we have undertaken a program of adapting genome rearrangement methodology to partially sequenced and incompletely assembled genomes. In earlier studies on *Drosophila* [58], and Chapter 3 in this thesis, we investigated the case when one or both of the genomes being compared are given only in contig form. A preliminary study had been carried out on *Carica papaya* [61]. Though we did find manage to find appropriate genomic data to test our methods, most sequencing projects are able to order some or all of the contigs, with intervening gaps, in scaffolds, which contain more information than unordered sets of contigs. In Section 5.2.1 we model how contigs are organized into scaffolds in the two current approaches to sequencing. In Section 5.3 we formalize scaffolded genome comparison, where one of the genomes is known only in scaffold form, as a combinatorial optimization problem for inserting missing genes in the scaffold gaps in such a way as to minimize the rearrangement distance. We devise an exact polynomial-time solution for this problem. In Section 5.5, we assess how this algorithm performs on simulated data after applying it in Section 5.4.1 to compare the scaffolded genome of castor bean *Ricinus communis* to the fully sequenced genome of grapevine *Vitis vinifera*.

Although using comparative evidence has long been commonplace in predicting gene location and indeed is one of the original motivations for model organism genomics, we believe this to be the first effort to predict the locations of large numbers of genes simultaneously using combinatorial optimization, while detecting and taking account of genome rearrangements.

5.2 Background

5.2.1 Partial sequencing scenarios

By *contig* we understand a completely sequenced fragment of a chromosome. This is assembled through identifying significantly overlapping reads of sequencing reactions. By *scaffold* we mean a set of ordered contigs (the order reflecting that on the chromosome) separated by unsequenced DNA which may be of known or unknown length. An anchored scaffold or contig is one whose location on the chromosome is known, thanks to any one of a number of different types of evidence.

In an idealized completely sequenced and gene-identified genome, complete gene orders would be known for each chromosome (Figure 5.1). When genome sequencing is not supplemented by finishing techniques, however, three different types of incomplete gene order data can result. When a strategy such as shotgun sequencing of unordered clones is employed, we have only isolated contigs constructed from overlapping reads, which would contain no internal gaps but could be relatively short assemblies (Figure 5.1b). Contigs-only assemblies could also involve much longer sequence fragments produced by complete, polished, sequencing of BACs or other chromosome fragments, which are not yet numerous enough to have been assembled into full chromosomes. When *paired ends* reads with unsequenced inserts are included with shorter complete reads, some of the contigs may then be ordered into scaffolds, with unsequenced gaps intervening between successive contigs, as in Figure 5.1c. Finally, detailed physical maps may be available to anchor all scaffolds to precise chromosomal locations, so that the scaffolds for a given chromosome become, in effect, a single scaffold or *pseudomolecule* (Figure 5.1d)

In practice, sequencing projects may use both BAC and shotgun methods as well as sequence obtained by other means. Not all BACs are necessarily anchored and some contigs produced by shotgun methods may be anchored. Nevertheless,

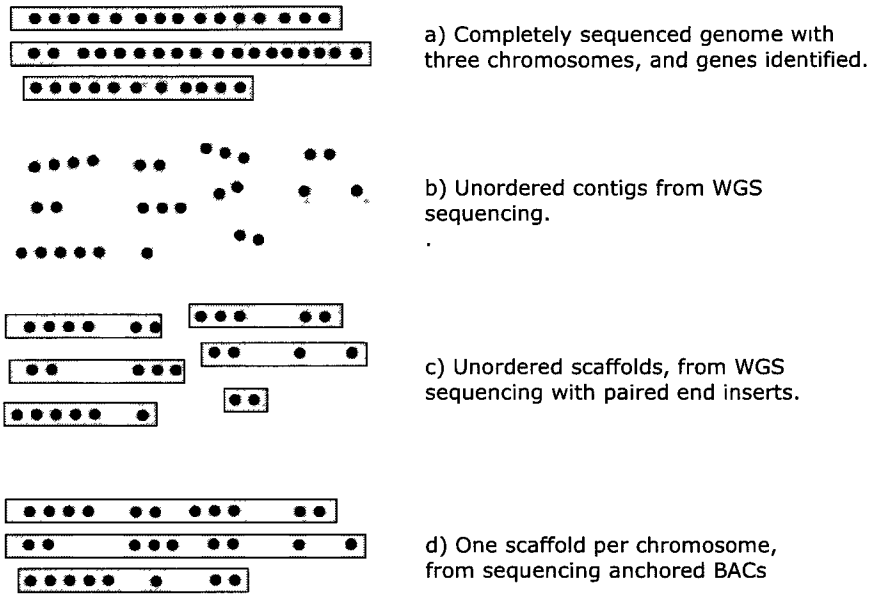

Figure 5.1: Types of partially sequenced and incompletely assembled genomes. Shaded areas represent sequenced contigs. Dots represent identified genes. The set of contigs within each outlined portion has a known order.

from our viewpoint, the three abstractions represented in Figure 5.1 b), c) and d) capture the essential distinctions between contig and scaffold and between anchored and unanchored.

5.2.2 Genomic distance

As discussed in Section 2.4, the *rearrangement distance* or *genomic distance* $d(G_1, G_2)$ is a metric counting the number of rearrangement operations necessary to transform one signed multichromosomal gene order G_1 into another G_2. In the simplest case, we require that the two genomes both contain the same n genes, with no duplicate genes. The positive or negative sign associated with a gene indicates its reading direction (or strandedness). To calculate d efficiently, we use the *breakpoint graph* of G_1 and G_2 as follows and as illustrated in Figure 5.2.

In a first step, each gene g with a positive sign is replaced by its tail and head

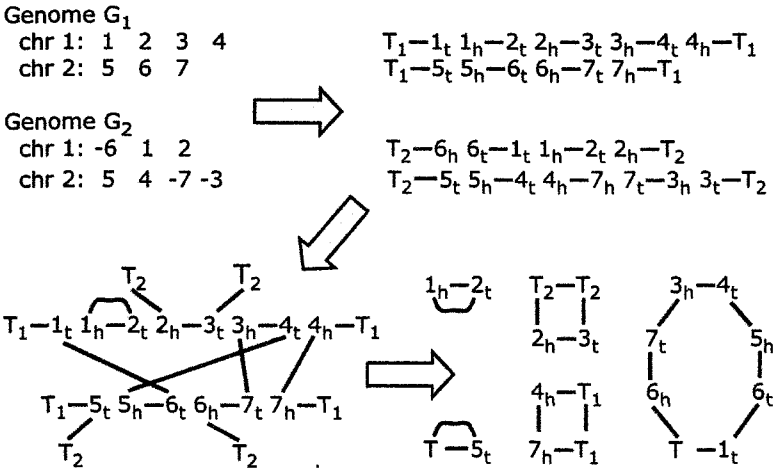

Figure 5.2: Construction of breakpoint graph. Upper left: Signed genomes G_1 and G_2. Upper right: Vertices and edges of individual genome graphs. Lower left: Cycles and paths after identifying vertices of two genome graphs. Lower right: Cycles in completed breakpoint graph.

vertices in the order g_t, g_h; for $-g$ we would put g_h, g_t. Each pair of successive genes in the gene order defines an adjacency, namely the pair of vertices that are adjacent in the vertex order thus induced. For example, if $i, j, -k$ are three neighbouring genes on a chromosome then the unordered pairs $\{i_h, j_t\}$ and $\{j_h, k_h\}$ are the two adjacencies they define. There are two special vertices called telomeres for each linear chromosome, namely the first vertex from the first gene and the second vertex from the last gene.

We convert all the telomeres in genome G_1 and G_2 into adjacencies with new vertices all labelled T_1 or T_2, respectively. We define a blue edge connecting the vertices in each adjacency in G_1 and a red edge for each adjacency in G_2.

In the next step in Figure 5.2, we start constructing the breakpoint graph by identifying (i.e., superimposing) each vertex in G_1 with the identically labelled vertex in G_2. In the last step depicted in Figure 5.2, we make a cycle of any path ending in two T_1 or two T_2 vertices, connecting them by a red or blue edge, respectively, while

for a path ending in a T_1 and a T_2, we collapse them to form one T vertex.

Each vertex is now incident to exactly one blue and one red edge. This bicoloured graph decomposes uniquely into κ' alternating cycles. If n' is the number of blue edges,

$$d(G_1, G_2) = n' - \kappa' \tag{5.2.1}$$

and the optimizing rearrangements are rapidly recovered by operations on the graph [32, 33].

In Section 5.3 below, we will refer to Tesler's [20] mathematically equivalent formulation of the breakpoint graph, where the final step in Figure 5.2, turning paths into cycles, is not carried out. Instead, there are only $\kappa \leq \kappa'$ cycles and a certain number π of the paths, namely those with at least one T_1 endpoint, are called *good* paths. Then

$$d(G_1, G_2) = n + \chi_1 - \kappa - \pi, \tag{5.2.2}$$

where χ_1 is the number of chromosomes in G_1. Although the breakpoint graphs, and d, are equivalent in the two formulations, Tesler does not call d "genomic distance". This difference is due to our inclusion of transpositions of chromosomal segments in the repertoire of rearrangements permitted in calculating d, together with the inversions, reciprocal translocations, chromosome fusions and fissions allowed by Tesler.

5.3 Methods

There are two different aspects of the comparison of a completely assembled genome G_1 with a genome in scaffold form G_2. One is *scaffold filling*, which predicts where in G_2 to locate potential genes that have not been identified in the sequence but are present in G_1. The second is *contig fusion*, which suggests how to piece G_2 contigs together to form chromosomes. In Figure 5.1, only scaffold filling is necessary for scenario (d) and only contig fusion is required for scenario (b). Scenario (c) requires

both.

We have shown how to handle the contig fusion problem in a previous publication on *Drosophila* [58] and in Chapter 3 of this thesis. A preliminary study had been carried out on *Carica papaya* [61]. This will be reviewed in section 5.3.4 below. In the present Chapter, we design and analyze an efficient exact algorithm for scaffold filling that simultaneously carries out contig fusion. We use this algorithm to analyze real and simulated data.

5.3.1 Filling in scaffolds

When G_2 is only partially sequenced, and is missing some orthologs with G_1 (cases (c) and (d) in Figure 5.1), we cannot complete the breakpoint graph since the red edges cannot be drawn to the two vertices corresponding to each missing gene, though these vertices are present in the graph and are incident to blue edges. At the same time, although we can draw a red edge between the last gene in one contig of a scaffold and the first gene in the next contig, we know that in reality there may be genes in the unsequenced gap between the contigs, and that once these genes are identified, the red edge will have to be "cut" and replaced by two or more gene vertices and two or more other red edges.

5.3.2 Statement of the combinatorial optimization problem

G_1 consists of χ chromosomes, each of which is an ordered set of signed genes.

A contig in G_2 is an ordered set of one or more signed genes, each orthologous to a gene in G_1. A scaffold in G_2 is an ordered set of contigs. Then G_2 consists of a number of scaffolds, each of which is an ordered set of genes interrupted occasionally by a gap.

Then with reference to Figure 5.1 (c) and (d), the problem becomes: Find an assignment of the missing genes to the gaps in the scaffolds or at the ends of the

scaffolds of G_2, thus transforming the scaffolds into contigs, such that the resulting set of contigs $\overline{G}_2$ is at a minimum rearrangement distance from G_1.

Implicit in our definitions is that between every pair of successive contigs in a scaffold is a gap large enough to contain genes. Where this is not the case, we can simply create a larger contig by disregarding the gap and concatenating the contigs on either side. We also disregard contigs without genes, so that they too may be subsumed in a gap. Note these are basically terminological conventions, rather than restrictions on the data.

5.3.3 A polynomial-time algorithm

The exact, linear-time, algorithm we have devised completes the breakpoint graph, only partially determined by G_1 and by the scaffolds of G_2, by means of insertions of missing genes into the gaps of G_2.

Terminology

We have hitherto used the term *path* only to refer to alternating-colour sequences of edges connecting some of the bivalent vertices in the breakpoint graph, with telomeres at either end, that are eventually turned into cycles by joining or collapsing these two telomeres. In what follows, however, a path more generally may be any such connected set of edges, with or without telomeres, and may consist of only one (blue) edge. Paths with two telomeres will be called *complete* paths. A *free end* is a vertex in the graph that has no incident red edges, only a blue one. Thus when we say that that G_1 and the scaffolds of G_2 *partially determine* a breakpoint graph, we mean that there are paths not ending in two T vertices, but in at least one free end.

A *half path* is a path ending in one telomere and one free end. A *pseudopath* is a structure consisting of two half paths where the two telomeres are deemed to be adjacent, though not by means of a red or blue edge. Pseudopaths will sometimes be

treated as if they were paths, with the two free ends being the free ends from the two constituent half paths.

Initially, a *cuttable edge* is a red edge drawn between vertices of two successive genes in a scaffold that are not in the same contig, i.e., there is an unsequenced gap between the genes. Subsequently, if a red edge is disrupted during gene insertion, new red edges are created as will be specified in the algorithm presented below.

A *bundle* is a subset of the paths in the breakpoint graph of G_1 and G_2. Each bundle is associated with one or more of the missing genes. The vertices corresponding to each missing gene, its free ends, must be in the same bundle and must be endpoints of two paths, or the two ends of one path. An *open* bundle contains at least one cuttable edge; a *closed* one has no cuttable edges. As the breakpoint graph is completed by the algorithm, the bundles also change.

A sketch of the algorithm

We have divided the algorithm into three parts. The first, the main algorithm **fillScaffolds**, constructs the partial breakpoint graph determined by G_1 and the scaffolds of G_2, and then partitions the paths in this graph (except the complete paths, and not including the cycles) among a number of bundles, some open and some closed. Initially, a bundle can contain either zero or two telomeres. If they are present, the half-paths, which are the two paths ending in telomeres, are linked together to become a pseudopath.

Although the missing genes represented by the free ends in an open bundle will eventually be inserted in an optimal way by manipulating cuttable edges, this is not possible within closed bundles. **fillScaffolds** thus calls the second algorithm **combineBundles**, which subsumes all closed bundles within open ones, as in Figure 5.3, thus creating larger open bundles, including some which contain more than two telomeres. This is done in such a way as to minimize the eventual genomic distance between

G_1 and $\overline{G}_2$. This step requires interchanging the half paths of the pseudopaths in the two bundles being combined, through changes in telomere adjacencies, to maximize the number of good paths according to the Tesler formulation in Equation (5.2.2). Finally, **fillScaffolds** calls **completeBundle**, which makes the connections between the free ends and the cuttable edges within each of the open bundles. The output of the algorithm includes cycles, each containing at most one pair of "adjacent" telomeres, which become the two endpoints of a complete path within the breakpoint graph.

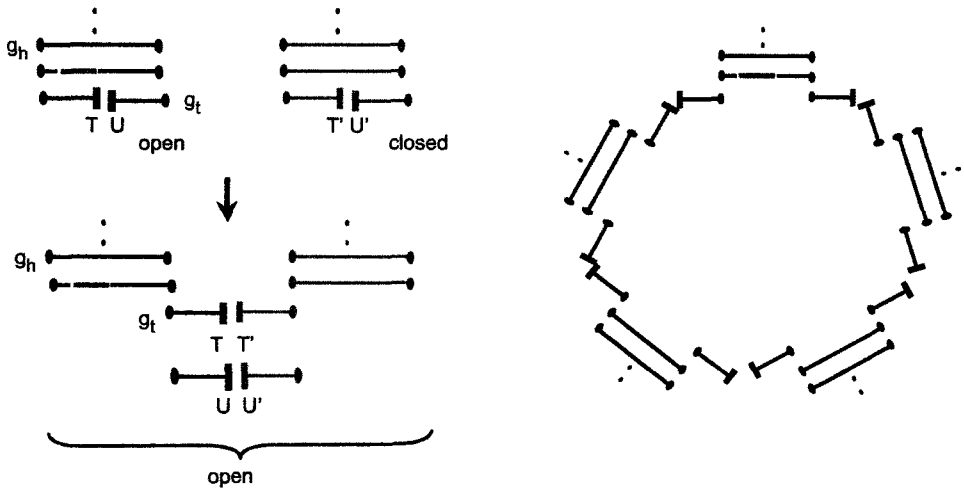

Figure 5.3: (left) Combining an open bundle (in black) and a closed bundle (in blue) by exchanging half paths. Dots represent free ends, rectangular blocks represent T vertices in half paths. Cuttable edge in red. This may be iterated to incorporate more closed bundles in a linear or circular structure as in the large open bundle on the right. Cuttable edge is in original open bundle.

After presenting the algorithm, we state and prove a theorem establishing its correctness.

Algorithm fillScaffolds

Input: A fully sequenced and assembled (without gaps) genome G_1, and

a genome G_2 made up of scaffolds containing some of the genes in G_1 and gaps.

Output: A completed form of G_2, denoted $\overline{G}_2$ where the missing genes from G_1 are inserted into the gaps in such a way as to minimize $d(G_1, \overline{G}_2)$, and the associated breakpoint graph.

1. Construct the breakpoint graph based on genome G_1 (blue edges) and G_2 (red edges), including *cuttable* red edges between consecutive genes in G_2 scaffolds separated by a gap. We include T_1 vertices at the telomeres of G_1 chromosomes and T_2 vertices at the end of G_2 scaffolds. We do not complete the third step of Figure 5.2, so the graph may contain cycles, complete paths and other paths.
2. We construct the initial bundles as follows. We choose any free end not already in any bundle as the seed of a new bundle. Then if a path containing free end g_t is in a bundle B, then we also include the path with g_h as a free end, and vice versa.
3. There can be zero or two T vertices in an initial bundle. If there are two, we consider the two half paths as if they were one path where the two T are adjacent, even though there is no red or blue edge connecting them.
4. We use **combineBundles** to remove all the closed bundles by merging them with open bundles, or with complete paths or cycles with cuttable edges, resulting in larger open bundles. We do this in such a way as to minimize $d(G_1, \overline{G}_2)$.
5. Complete each bundle, using **completeBundle**.

Algorithm combineBundles

Input: The set of open and closed bundles as well as the set S of complete paths and cycles with cuttable edges.

Output: A set of open bundles, and a subset S' of the complete paths and cycles. The open bundles contain all the vertices in the input bundles plus those vertices in $S \backslash S'$, the paths and cycles not included in S'.

1. **while** there is a closed bundle with a T_1T_1 adjacency and a open bundle, or complete path with a cuttable edge, with a T_2T_2 adjacency, combine them by switching the adjacencies between T vertices, i.e., by exchanging two half-paths. This results in a larger open bundle and also increases the number of good complete paths by one.
2. **while** there is a closed bundle with a T_2T_2 adjacency and a open bundle, or complete path with a cuttable edge, with a T_1T_1 adjacency, combine them by switching adjacencies. This results in a larger open bundle and also increases the number of good complete paths by one.
3. **while** there is a closed bundle with a T_2T_2 adjacency and closed bundle with a T_1T_1 adjacency, combine them by switching adjacencies. This results in a larger *closed* bundle and increases the number of good complete paths by one. The closed bundle eventually has to be combined with an open bundle or cycle or complete path.
4. **while** there is a closed bundle with a TT adjacency and a open one with a TT adjacency, combine them by switching adjacencies. To maintain the number of good paths, if the adjacencies are T_1T_2, and $T_1'T_2'$, then after the switching the adjacencies they should be T_1T_2' and $T_1'T_2$.
5. **while** there is a closed bundle, combine it with an open bundle or

cycle or complete path by adding a pair of cuttable edges, as in Figure 5.4:

i. Find two free ends g_h and g_t in the closed bundle.

ii. Choose a cuttable edge kl in some open bundle, or path or cycle.

iii. Replace kl by two cuttable edges kg_h and and g_tl.

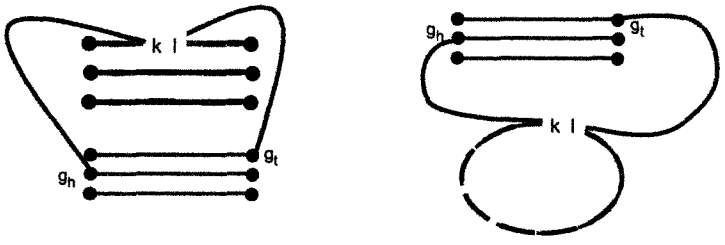

Figure 5.4: Combining a closed bundle, represented by blue incomplete paths, with an open bundle (left) and with a cycle (right) with cuttable edge kl, shown here after being replaced by two cuttable edges kg_h and g_tl).

Algorithm completeBundle

Input: a good bundle.

Output: a number of cycles.

while there remain paths in the bundle as in Figure 5.5

1. Choose a path containing a cuttable edge kl, with endpoint g_t, where l is not on the subpath between k and g_t.
2. Find the path with endpoint g_h, possibly the same path.
3. Replace kl by kg_t and g_hl, which are red cuttable edges. This results in a cycle containing kg_t and a path containing g_hl, unless g_t and g_h are on the same path, in which case the operation produces two cycles.

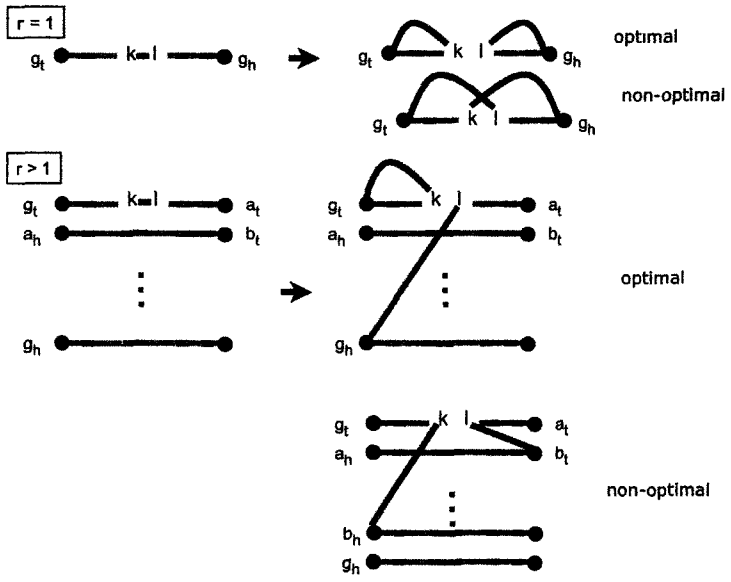

Figure 5.5: First step in completing a good bundle, maximizing the number of cycles. r is the number of incomplete paths in the bundle. Dots represent free ends before the step. kl is a cuttable edge in the input bundle, as are kg_t and g_hl in the new bundle. Output consists of two cycles (for $r = 1$) or one cycle and an open bundle with one fewer incomplete path (for $r > 1$).

Proving the algorithm

After the first three steps of **fillScaffolds**, suppose we have constructed γ open bundles with $r_1, \cdots, r_\gamma$ paths, β closed bundles not containing T vertices with $q_1, \cdots, q_\beta$ paths, and $\delta - \beta$ closed bundles containing T vertices with $q_{\beta+1}, ..., q_\delta$ paths. Let $\epsilon = 0$ unless $\delta - \beta > 0$ but there are no open bundles containing T vertices nor any complete paths with cuttable edges, in which case $\epsilon = 1$. Suppose there were κ^* cycles and p^* complete paths in the original breakpoint graph of G_1 and G_2.

Theorem: There are

$$\kappa + p = \kappa^* + p^* + \sum_{i=1}^{\gamma} r_i + \sum_{i=1}^{\delta} q_i + \gamma - \beta - \epsilon \tag{5.3.1}$$

cycles and complete paths in the final breakpoint graph constructed by **fillScaffolds**. Moreover, not only is the number of cycles κ maximal over all ways of inserting the missing genes, but so is the number of good complete paths $\pi \leq p$. Thus the algorithm also implicitly produces the value of $d(G_1, G_2)$.

Proof: We first show that in completing an open bundle with r paths, we obtain $r+1$ cycles. Later, we will show that each of these cycles has at most two T vertices.

Consider the case $r = 1$. Figure 5.5 shows that completing this bundle in the optimal way creates two cycles. It also shows that for $r > 1$, we obtain a open $r-1$-bundle plus one cycle. Thus, completing an open bundle with r paths produces a total of $r + 1$ cycles.

It is thus never advantageous to draw a pair of red edges between two open bundles with r and s edges, since this cannot create a cycle, only a bundle with $r+s-1$ edges. When completed this will only give $r+s$ cycles instead of the $r+s+2$ if we had completed them separately.

On the other hand, to be processed toward completion, it is necessary for a each closed bundle to be combined with either an open bundle, or a cycle or a complete path with a cuttable edges, since a closed bundle has no cuttable edges by itself. The optimum ways to do this are illustrated in Figs. 5.3 and 5.4. In the former case, where both bundles have T vertices, switching adjacencies allows a closed bundle with r paths to contribute r paths to the open bundle, and eventually to be responsible for r cycles. If one of the bundles has no T vertices, on the other hand, the closed bundle can contribute only $r-1$ of its r paths in combining with the open bundle (Figure 5.4).

Now the numbers of open bundles, closed bundles with T vertices, closed bundles without T are fixed at the outset, and we can also find out if there are open bundles with T (or complete paths with cuttable edges) or not at the initial stage. Counting the number of cycles given by each type we arrive a the first claim of the theorem. Since each combination and completion is done optimally in the algorithm, the result

for κ is best possible. So is π, through the operations minimizing the number of T_2T_2 edges in **combineBundles**.

It remains to show that the cycles output by the algorithm have no T vertices, i.e., are the kind of cycles appearing in the breakpoint graph in the second to last stage of the construction of Figure 5.2, or exactly two adjacent T, i.e., are the kind of complete paths (upon dissolution of the TT adjacency) appearing breakpoint graphs. Otherwise, the values of κ and π that we obtain in this theorem would not be those required for Equation (5.2.2).

To prove this, we refer to Figure 5.6, which integrates aspects of Figure 5.3, 5.4 and 5.5. The case by case analysis illustrated there shows that if there are more than one TT adjacency in a path, these adjacencies will necessarily be incorporated at most one at a time into cycles. Cycles without TT adjacencies are also cycles in the breakpoint graph between G_1 and the augmented genome $\overline{G}_2$ and the cycles with TT adjacencies become complete paths, either good or bad, in this breakpoint graph. This completes the proof.

The construction of the optimal breakpoint graph by **fillScaffolds** inserts the missing orthologs in the scaffold gaps and at the ends of scaffolds in a way that minimizes the number of rearrangements intervening between G_1 and the optimal G_2 thus constructed. Once the optimal breakpoint graph is known, these rearrangements can be recovered rapidly by standard manipulations on the graph [32], as mentioned in our discussion of Equation (5.2.1). The construction of breakpoint graphs is of linear complexity, and this extends to the identification of bundles and their manipulations in **fillScaffolds**. This includes the placement of missing genes. The recovery of minimizing rearrangements can be implemented in subquadratic time [32].

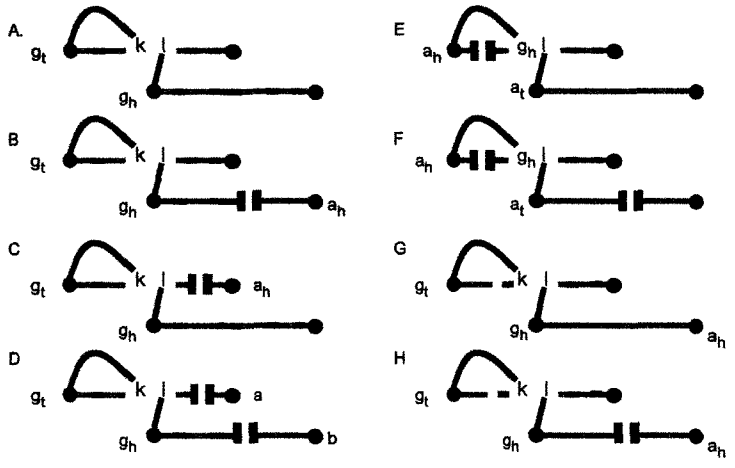

Figure 5.6: Illustrating proof that cycles formed by **completeBundle** can contain only zero or one TT adjacency. $g_h l$ is the cuttable edge formed when kl is destroyed as k is incorporated into a cycle. Case A reflects the first step in Figure 5.5 ($r > 1$), not involving any TT adjacencies. Cases B and C show how the paths from an originally closed bundle enter into the completion process thanks to the switched adjacency of a pair of half paths illustrated in Figure 5.3. Case D shows how two (or more) such adjacencies can accumulate in one path, but always with a cuttable edge between them. Cases E and F show how a single adjacency is incorporated in a cycle, through deletion of the cuttable edge between the adjacency and any other adjacency in the path. Cases G and H show a step in the completion of an open bundle without T, constructed as in Figure 5.4, also leading to the entry of originally closed bundle paths into the completion process.

.

5.3.4 Contig fusion

The algorithm in the preceding section fills in the gaps between the scaffold whenever this is justified, so that by our definitions, the scaffolds become contigs. For unanchored scaffolds, as they are filled in by our algorithm described above, they are also being assembled into chromosomes. In doing this, our method based on the breakpoint graph treats the incorporation of each scaffold/contig as if it were a chromosomal fusion operation.

We previously found [58] (see Chapter 2.5 above) through simulations that for

ordinary genomes, i.e., complete gene orders, if there are τ rearrangements, but the genomic distance algorithm infers d rearrangements, then the expectation

$$\mathrm{E}(d) \approx n(1 - e^{-\lambda_1 \tau/n - \lambda_2(\tau/n)^2}), \tag{5.3.2}$$

An estimate of τ is

$$\hat{\tau} = \frac{n}{2\lambda_2}\left(-\lambda_1 + \sqrt{\lambda_1^2 - 4\lambda_2 \log\left(1 - \frac{d}{n}\right)}\right) \tag{5.3.3}$$

where $\lambda_1 \approx 0.85$ and $\lambda_2 \approx 0.58$. λ_1 and λ_2 are parameters that depend on how the rearrangements are generated.

When one of the genomes consists of unanchored contigs (or filled-in scaffolds), we have to correct the output of the genomic distance algorithm d_S before using (5.3.3) to take into account the number of scaffold "fusions" necessary to optimally piece together the scaffolds into chromosomes. The corrected distance is (see Chapter 3.4.2)

$$d = d_S - \alpha(\tau)|\chi_A - \chi_B|, \tag{5.3.4}$$

where $\alpha(\tau)$ is a decreasing function of the number of rearrangements τ, approximately paralleling the derivative of d, namely $\frac{dd_S}{d\tau}$.

5.3.5 Missing genes: absent or just unsequenced?

We will use G_1 and G_2 here to refer to the genomes that are the source of the gene order data. By definition in our method, unsequenced genes must be located in gaps between the contigs or at the ends of scaffolds. We assume any genes within contigs have been identified. However, many or even most genes that are in G_1 but have no ortholog in the G_2 data may actually be absent from the latter genome either because over time they have been deleted from G_2, or because they were acquired by G_1 but

not by G_2 since the two lineages diverged.

The scaffold filling algorithm is designed to enhance sequence assembly, and cannot distinguish one type of missing gene from another. Indeed, were gene models are available from cDNA or EST data, we could simply discard the missing genes from G_1 that are not reflected in the set of gene models for G_2. In general, however, we do not have this information, and the best we can hope for is to be able to estimate quantitatively how many of the missing genes are present in the genome, but unsequenced.

Let $\underline{G}_1$ represent the genome G_1 with all the genes missing from G_2 deleted. The remaining genes are ordered in the chromosomes in the same way as in G_1. One way to estimate the proportions of the two types of missing genes is to compare the genomic distance $\underline{d} = d(\underline{G}_1, G_2)$, where only the genes in common in the data from the two genomes are considered, with the distance $\overline{d} = d(G_1, \overline{G}_2)$ after G_2 is augmented to $\overline{G}_2$ by the scaffold-filling procedure. As detailed below, we have found in extensive simulations that if all unsequenced genes were originally located in regions that are gaps after the (partial) sequencing and assembly are finished, the distances $\underline{d}$ and $\overline{d}$ are identical, or almost so, over a wide range of genome sizes, rearrangement distances and missing gene sets. If on the other hand, many of the missing genes are in reality absent from the G_2 genome, a major proportion of these, approximately equal to the coverage of the genome sequencing, will have been in syntenic contexts in G_1 that are in contigs in G_2. Thus forcing these genes to be in gaps, as the scaffold-filling algorithm does, will tend to increase the rearrangement distance $\overline{d}$. Then if m is the number of missing genes

$$d' = (\overline{d} - \underline{d})/m \tag{5.3.5}$$

is a measure of how the proportion of missing genes are not actually in the G_2 genome.

The value of d' depends on how much the contigs are already rearranged in the independent evolution of the two genomes. If the contigs are highly rearranged

compared to G_1, then there is no necessary increase in d' when the missing gene is forced into a gap. But if the syntenic context of a missing gene is intact in a contig, then forcing this gene into a gap remote from this context will necessarily increase d'.

Our strategy for evaluating this dependence requires us to manipulate the overall degree of syntenic context conservation while keeping d fixed. In the simulations, in Section 5.5 below, we accomplish this by using fixed length inversions. By generating the genomic divergence with very short inversions, we require more inversions to attain the same inferred d, but we also guarantee the existence of a good number of conserved segments (conserved syntenic contexts) and allow d' to increase. By fixing the inversion length at successively higher values, the scope of each inversion becomes longer and it is less likely a conserved segment will remain undisrupted, and d' will tend not to increase.

5.4 Connecting scaffold filling and contig fusion

In this section, we describe how to apply the scaffold filling and contig fusion methods by comparing the draft genome of *Ricinus communis* with the more complete genome of *Vitis vinifera*. We will do this in three stages. First we will give a brief description of the phylogenetic relationship of these two angiosperms and a preliminary bioinformatic comparison of their genome sequences in Section 5.4.1. This will give us 14,033 presumptive orthologous genes in the two genomes, plus 4,267 genes *Vitis* genes which are not in the *Ricinus* data, either because they are in the unsequenced parts of the genome, or because they have simply been deleted from, or never acquired in, the *Ricinus* lineage. We call these *missing* genes. We calculate statistics about how the missing genes are distributed in *Vitis*, as singletons, pairs, triples or longer runs. We also calculate for *Ricinus* the distribution of the number of genes per contig and per scaffold, the number of contigs per scaffold and the total numbers of contigs and scaffolds.

In the second stage, we use these distributions to simulate random pairs of genomes having the same characteristics as the *Ricinus-Vitis* data set. We model the number and distribution of missing genes as being due to three types of process:

- the evolutionary divergence of gene complement between the two species,
- the variability of conserved segment size as chromosome inversions disrupt gene order over time, and
- the distribution of contig and scaffold sizes produced during the sequencing project.

The simulations that are presented in Section 5.5 enable us to predict how these factors affect the results of scaffold filling.

Finally, in the third step, we apply our scaffold-filling algorithm and contig fusion analysis to the *Ricinus-Vitis* data and interpret the results in the light of missing gene models we elaborate and the simulations we carry out. These results are discussed in Section 5.6.

5.4.1 The castor bean genome

Sequenced by the Sanger method to a depth of 4X, the castor bean genome exemplifies the kind of final product that we can increasingly expect of draft genome sequencing projects, with a large number of scaffolds ($> 28,000$) not anchored to any chromosome. (Indeed, later genomes sequenced with the 454 and Solexa methods will have shorter reads and have perhaps even shorter scaffolds.) Almost all of the genes, however, are found on a smaller number ($\approx 1,600$) of the larger scaffolds (> 10 Kbp). To illustrate our method, we wish to pick a completely sequenced genome with which to compare *Ricinus*, one from a not too distantly related angiosperm species, so that it is likely to share a large majority of its gene complement and gene order with *Ricinus*.

More distant relatives might also work, but divergent gene complement and decreasing synteny would lead to more ambiguous and less reliable results. Moreover, there is another, more stringent, condition. On the two lineages from their common ancestor leading to the two genomes, there should be no whole genome duplication (WGD) event. Though we know how to compare gene orders of such former tetraploids with diploids that diverged before the WGD [65, 61], in the first instance we would like to avoid such complexities in testing our new procedure. This eliminates *Arabidopsis, Oriza, Populus* and *Medicago* among the high-quality genome sequences available. It also eliminates the closely related *Hevea brasiliensis* (rubber) genome, in the same family (Euphorbiaceae) as *Ricinus*, for which the draft sequence has been announced, but which is a recent tetraploid or, more accurately, an amphidiploid [66], p. 278.

Fortunately, there seems to be no WGD in the lineages leading to *Vitis vinifera* and *Ricinus* since their last common ancestor, and so we can use *Vitis* as G_1 in our method and *Ricinus* as G_2. Although Burleigh *et al.* [56] have suggested that there have been one or more WGD events in the rosid clade rooted after the divergence of *Vitis vinifera*, in the lineage leading the Euphorbiaceae family, which contains *Ricinus*, the evidence presented in that paper, namely a large number of gene families originating in the early period, is not at all statistically significant, may be a methodological artifact as acknowledged by the authors, and, *pace* reference [64], is uncorroborated in the literature (cf the recent survey of the many angiosperm WGD events by Soltis *et al.* [62]). In addition, though a relatively recent WGD has been proposed for *Vitis* [64], this suggestion has not met with general acceptance [57, 62] either. Thus we may provisionally treat the *Vitis-Ricinus* relationship as being uninterrupted by WGD. Finally, there is evidence that *Vitis* gene order has evolved relatively slowly, e.g., Reference [61].

We extracted scaffold, contig and gene level data on *Ricinus communis* from GenBank as well as chromosomal gene order data on *Vitis vinifera*. Of the 18,300 *Vitis* genes, 14,033 showed up as best reciprocal hits (BRH), using BLAST and a 1e-5

threshold to compare the proteins, among the 31,221 possible protein genes suggested by the *Ricinus* sequence. We discarded the rest of the *Ricinus* gene models.

Key statistics are given in Table 5.1. To the 4,267 missing orthologs we add 339 genes that were found on *Ricinus* scaffolds with no other genes, i.e., since they contribute no gene order information, so that a total of 4,606 genes are to be placed relative to the *Ricinus* gaps. The remaining 13,694 of the 14,033 *Ricinus* orthologs were organized into 748 (=1,087-339) scaffolds each with two or more genes, i.e., containing at least some order information. The scaffolds also contained a total of 2,527 gaps. Note that our algorithm automatically places additional gaps at the two ends of each scaffold, so that we need not worry separately about placing genes between the scaffolds.

genes per contig	number of contigs	contigs per scaffold	number of scaffolds	genes per deletion	number of deletions
1	1699	1	469	1	2253
2	612	2	181	2	548
3	337	3	121	3	167
4	210	4	80	4	36
5	140	5	67	5	23
6	107	6	29	6	8
7	91	7	25	7	5
8	55	8	19	8	1
9	38	9	27	9	1
10	36	10	10	10	0
>10	289	>10	59	>10	3
total: 14033	3614	total: 3614	1087	total: 4267	3045

Table 5.1: (left) Contigs in *Ricinus* data. (middle) Scaffold structure of *Ricinus* data. (right) Size of *Vitis* genes blocks deleted from *Ricinus*.

The distance $d(\underline{G_1}, G_2)$, where only the orders of the 13,694 genes in both G_1 and the scaffolds of G_2 are considered, is 8283. The distance $d(G_1, \overline{G_2})$, which compares G_1 to the augmented version of G_2, namely $\overline{G_2}$, after the scaffold-filling procedure has been applied, so that the orders of all 18,300 genes are considered, is 9,931.

5.5 Simulations

We performed two different sets of simulations that we called Simulations Experiment 1 and Simulations Experiment 2 that will be described in the following sections.

5.5.1 Simulations Experiment 1

Simulating evolution

We simulate pairs of genomes with $n = 18,000$. The first, G_1, simply has the genes evenly distributed among the 10 chromosomes. We used between 4,000 and 10,000 random rearrangements to derive the second genome from the other. These parameters are motivated by the details of the *Ricinus* - *Vitis* comparison to be reported in the next section. We assume the rearrangements are preponderantly inversions (around 90%), a common tendency in gene order evolution.

Simulating scaffolds

The input to the scaffolding simulation program consists of

- the original genome G_1 with χ_1 chromosomes and the rearranged genome G_2 on χ_2 chromosomes, both containing n genes, with all n orthologies known. For simplicity, and because it has little impact on the results, we simulate with $\chi_1 = \chi_2 = \chi$, a single value. Additional input parameters are:

- c: the expected number of non-telomeric contig breaks, where $1 \leq c < n - \chi$, so that the number of contigs will be around $c + \chi$. Set $p = c/(n - \chi)$ the proportion of non-telomeric genes which should terminate a contig, i.e., the probability that a gene will the final gene in a contig. (This does not include the last gene in a chromosome, which is always the end of a contig.)

- ω: the "density" of the scaffolds, where $0 < \omega < 1$ and

- s: a parameter that helps determine the proportion of gaps that are also scaffold breaks. $0 < s < 1$.

Values of these parameters are chosen to imitate the *Ricinus* - *Vitis* comparative data or, in the case of experimental parameters like the density or the number of rearrangements, to bracket the *Ricinus* - *Vitis* values

In the simulation itself, then, running through all the genes from the first in each chromosome of the rearranged genome to the second-to-last in that chromosome, for each gene we randomly decide with probability p whether this gene ends a contig, and the next gene is the first gene in a new contig. The first gene in the chromosome is of course the first gene in the first contig of that chromosome, and the last gene in the chromosome also is the last gene in a contig.

We can expect this procedure to produce $c + \chi$ contigs.

We then proceed to discard contigs at random, to simulate unsequenced parts of the genome. Independently for each contig we discard it with probability $1 - \omega$, and define a "gap" to exist in its place. If two or more contiguous contigs are discarded, we create double gaps, triple gaps, and so on. We also determine for the discarded contigs whether they also define the end of a scaffold. Each gap is defined to be the end of a scaffold with probability s. If there are m contiguous gaps they are considered to determine a scaffold break with probability $1 - (1 - s)^m$. Note that at a scaffold break we discard the information about which of the two flanking scaffolds is ordered before the other. We also discard all gaps at the beginning and the end of each scaffold, so that all scaffolds start and end with contigs. The output is a sequence of all the genes in each scaffold in order, with a indication of where they are separated by gaps, and no particular order of the list of all the scaffolds in the genome, or information about chromosomal assignment.

Simulation results

We use the algorithm in Section 5.3.3 to calculate d from G_1 and the completed G_2 in each run. We carried out simulations to assess the effect of the scaffold density ω and the rearrangement distance d on the quality of the reconstruction of the scaffolded genome.

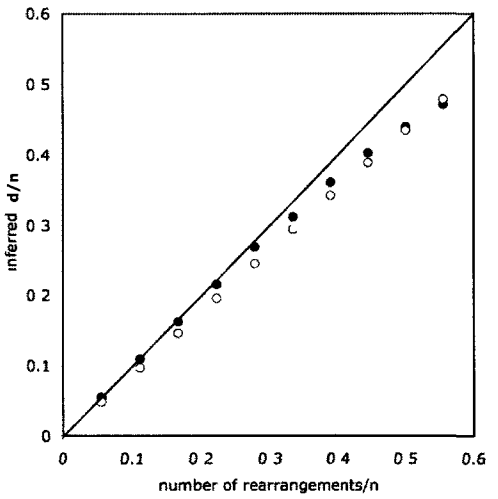

Figure 5.7: Fit of equation (5.3.2) (filled dots) and the normalized number of rearrangements inferred (open dots), after correction for number of scaffold fusions.

First, however, we wanted to compare the relationship between the inferred number of rearrangements, corrected for the number of scaffolds in G_2, and the actual number of random rearrangements τ used in generating this genome. Fig. 5.11 shows that this relationship is close to that discussed for contigs in equation (5.3.2) described in Section 5.3.4

This allows us to estimate τ using equation (5.3.3), for the eventual purposes of distance-based phylogeny.

In Fig. 5.8, we see that both a low rate of recovery of genes in the scaffolds and high genomic distance contribute to the insertion of missing orthologs where they cause an increase in genomic distance. We loosely call this "incorrect placement" of

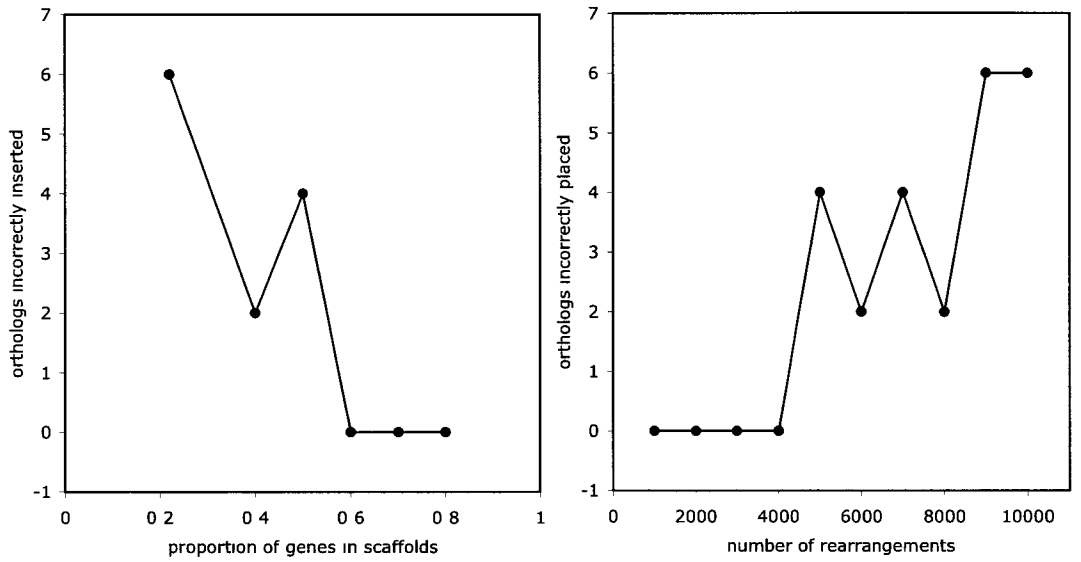

Figure 5.8: Effect of reducing the number of genes in scaffolds (left) or increasing the rearrangement distance (right) on the number of "incorrectly" placed orthologs.

the gene, but there is no error involved; it is just a matter of there being no place to insert the gene without it causing an increase of one or more in the genomic distance. Note that if G_1 contains genes g h k in that order where h is not found in the G_2 scaffolds, and g occurs directly before k in G_2 with a gap in between, any optimal solution will insert h in that gap and will not cause an increase in the rearrangement distance. Even if g and k are remote from each other in G_2 but g is followed by a gap or k is preceded by a gap, gene h can be inserted in that gap without causing any increase in genomic distance. And there are other such cases. But there are also cases, such as when g and k are adjacent within a G_2 contig, where any insertion will likely cause an increase in the distance.

The main observation from these simulations is that even when many genes are absent from G_2 or when the genomes are very highly rearranged, the scaffold filling algorithm correctly places all but a tiny number of missing genes. But this is not what we found in Section 5.4.1 with respect to *Vitis* and *Ricinus*. This motivated us to undertake a new series of simulations better able to model the incorrect placement

of missing genes by the algorithm.

5.5.2 Simulations Experiment 2

Simulating evolution

We simulated pairs of genomes with number of genes $n = 18,300$. The first, G_1, simply has the genes evenly distributed among the 10 chromosomes. Blocks totalling 4,267 genes, distributed as in Table 5.1, to be eventually deleted in forming G_2, were chosen at random along the genome, constrained only from overlapping or even touching. At first these genes were only marked, but not deleted. For a range of values of τ, we applied τ random rearrangements to G_1 and then deleted the marked genes. We assumed that rearrangements are preponderantly inversions (around 90%), a common tendency in gene order evolution, and we chose the two breakpoints for each rearrangement randomly along the chromosome.

All deletions create gaps

Each deletion event created a gap between two contigs. In addition random contig breaks were inserted to make sure that the number of contigs totaled 3,614, as in the *Ricinus* data in Table 5.1. Adjacent contigs were then assembled randomly into scaffolds in such a way as to produce the same distribution of contigs per scaffold as in Table 5.1. Single-gene scaffolds were identified and removed from the lists of scaffolds and contigs and transferred to the list of missing genes, as in the *Ricinus* analysis.

Applying the **fillScaffolds** algorithm to these data, for $\tau = 3000, 6000, \cdots, 15000$, twenty runs for each τ, demonstrated that under the model where missing genes are entirely due to incomplete sequencing, the distance $d(G_1, \overline{G}_2)$ was exactly the same as $d(\underline{G}_1, G_2)$ in 90% of the runs, and 2 rearrangements more costly (out of 5000 or more) in the remaining cases. Thus we can conclude that **fillScaffolds** generally inserts the

4,267 missing genes (plus a variable number of genes from single-gene scaffolds) at virtually no cost, in terms of genomic distance. This holds over a wide range of genomic distances. It also hold for a range of models of rearrangement; for example, if instead of the two breakpoints being randomly chosen over the chromosome, we restrict inversions to involve only a small number of genes, the difference between pre- and post-scaffold-filling is less than 0.5%.

We note that the simulations, including the use of our implementation of the **fillScaffolds** algorithm, took on the order of a minute each on a MacBook Pro with 3.06 GHz processor speed.

Some deletions do not create gaps

How can we model the subset of missing genes that are not those unsequenced genes in G_2 that cause gaps between the contigs, but genes that are not in the G_2 genome at all? To do this, we delete some proportion of the genes marked at the beginning of the simulation as before, but do not create a gaps between contigs at the deletion point. Insofar as the syntenic context of the absent G_1 gene is conserved in a G_2 contig, this should cause an increase in $d(G_1, \overline{G}_2)$ over $d(\underline{G}_1, G_2)$, due to the rearrangement cost of moving the gene from its original context to a gap. It will not tend to cause an increase if the syntenic context in G_1 has already been rearranged in G_2, e.g., if the absent gene is at the breakpoint of an inversion or translocation. Because this effect involves the interaction of synteny conservation and rate of non-gap-creating deletions, we set up simulations as described in Section above, varying both of these processes. We carried out simulations with from 60% to 100% non-gap-creating deletions and with fixed-length inversions from 1 to 6 genes long. Each of the 30 simulation conditions (6 conservation settings times 5 deletion types) is represented by the average of 20 simulation trials.

The simulations show that the value of d' increases with greater conserved syn-

teny, and with higher proportions of non-gap-creating deletions. This is depicted in two ways in Figures 5.9 and 5.10. Of particular interest will be the case of $d' = 0.37$ indicated by the dashed line in both graphs. This case corresponds to the *Ricinus-Vitis* comparison as reported in Section 5.6 below.

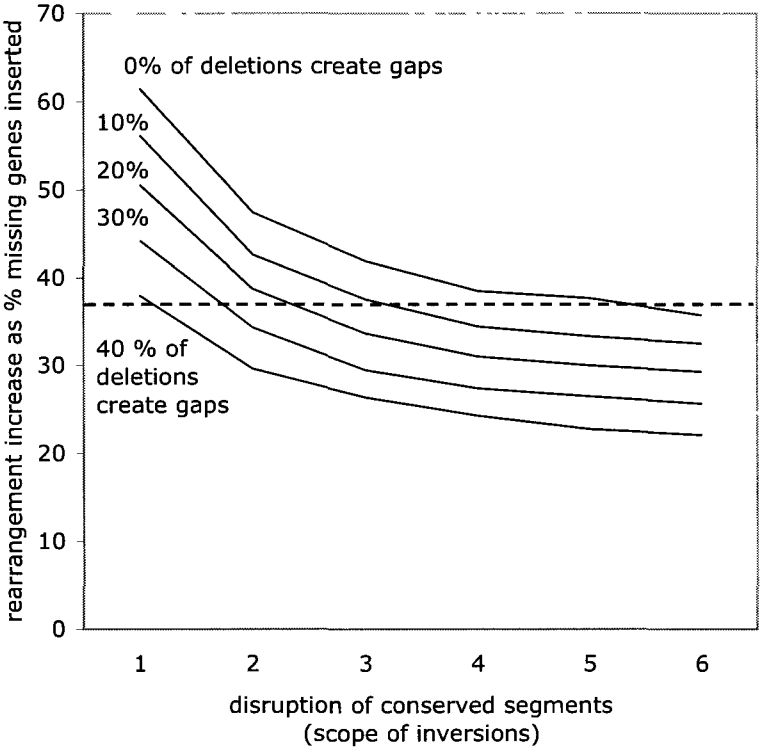

Figure 5.9: Effect on d' of decreasing syntenic conservation for different proportions of non-gap-creating deletions.

Distances

We compare the relationship between the inferred number of rearrangements, corrected for the number of scaffolds in G_2, and the actual number of random rearrangements τ used in simulating this genome. Before the deletion of the genes from the gaps and the creation of the scaffolds, i.e., when the genomes contain 18,300 orthologs, equation (2.5.3) closely predicts the observed distances. This is illustrated

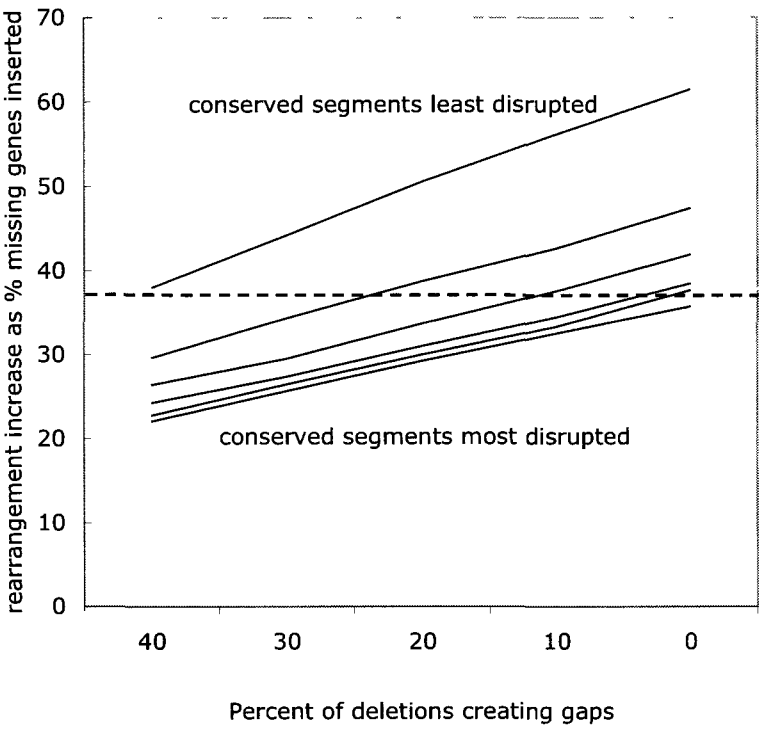

Figure 5.10: Effect on d' of increasing non-gap-creating deletions for different levels of syntenic conservation.

in Figure 5.11, which is based on the average of 20 simulated trials per data point.

After the deletions of 4,267 genes representing those absent from G_2, as well as the variable number (usually less than 200) genes in single-gene scaffolds, following scaffold creation in the "all deletions create gaps" model, the observed distance is less than that predicted by equation (2.5.3), especially when the simulated rearrangements become numerous. This is also illustrated in Figure 5.11. The observed distance (averages of 20 trials) is corrected (downwards) for the 935 chromosomal fusions necessary to assemble the contigs (filled scaffolds) into 10 chromosomes, using $\alpha(\tau) = \frac{dd}{d\tau}$ in equation (3.4.2), but the inferred distance is smaller than predicted even without this correction.

These results indicate that estimating τ using equation (2.5.4), e.g., for the purposes of distance-based phylogeny, is likely to underestimate this genomic distance to

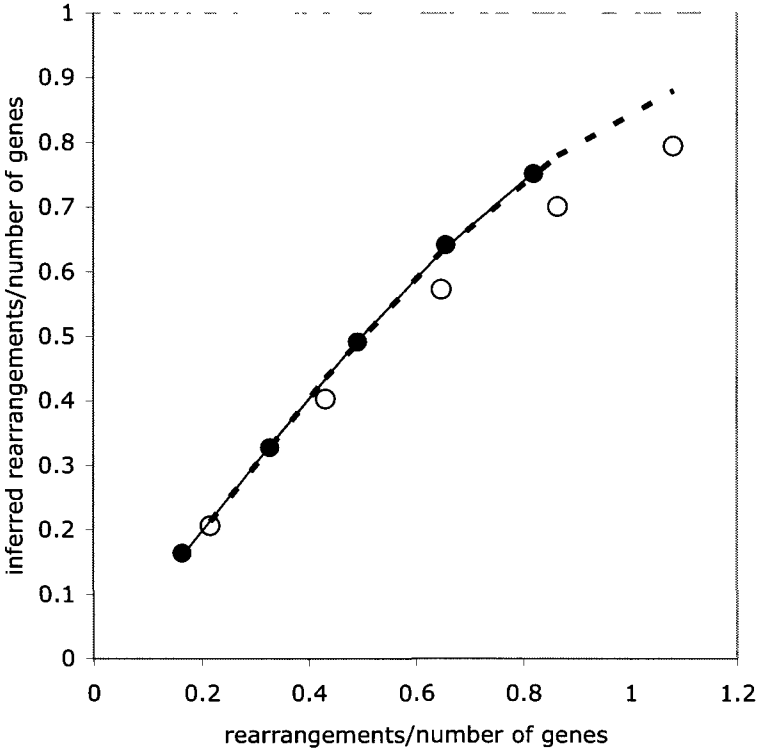

Figure 5.11: Fit of equation (2.5.3) (solid line) to the normalized number of rearrangements inferred (filled dots), before deletion of missing genes; $\lambda_1 = 0.899, \lambda_2 = 0.988$. Fit of same equation, when taking into account only genes remaining after deletions and scaffold construction (dashed line), to normalized number of rearrangements inferred (open dots) after correction for the number of scaffold fusions. The position of each dot is based on an average of 20 trials.

some extent.

5.6 Results on *Ricinus*

Our algorithm found a distance of 9,931 operations between *Vitis* and the reconstructed *Ricinus* genome, corrected for fusions to 9,365 (cf Section 5.3.4). Previous work [65] estimated the distance between *Populus* and *Vitis*, which should represent the same divergence time, given that *Populus* and *Ricinus* presumably shared a common ancestor since the divergence of the *Vitis* lineage. The distance between *Populus*

and *Carica papaya* was also estimated [61]. This should represent a divergence time smaller than *Vitis* - *Ricinus*. Making these comparisons (Table 5.2) is reasonable, although the *Populus* rearrangements occurred after a WGD event.

When all the data are taken into account, and each distance normalized by the number of genes in the comparison, the *Vitis* - *Ricinus* distance is comparable to the *Populus* - *Carica* one, and both are greater than *Populus* - *Vitis*. This slight disproportion between *Vitis* - *Ricinus* and *Vitis* - *Populus* is attributable, in unknown proportions, to

- the use of a method more refined than BRH, namely OrthoMCL [68], to identify *Populus* - *Vitis* orthologs. For *Vitis* - *Ricinus* we used BRH without any validation by chromosomal context or by gene ontology.
- generation time difference in different lineages, as argued in [61].
- the proportion of non-gap-creating deletions, which is a function of the divergence in gene complement.

Only the first of these is directly amenable to computational improvements, without further biological input.

The key result in Table 5.2 is the rate of correct placement of the missing orthologs. Some 63 % percent of the orthologs were inserted without any increase in rearrangement distance. This is comparable to the 57 % - 64 % in the previous studies, even though the latter each benefited from evidence from two syntenic contexts rather the single *Vitis* contexts used for orthologs placement in *Ricinus*. With reference to Figures 5.9 and 5.10, it suggests that around 75% of missing genes are not attributable to incomplete sequencing, but rather to divergent gene complement in the two genomes. Table 5.2 and Figure 5.9 and 5.10 are also compatible with the fact that almost all the missing genes in the *Populus* comparisons are attributable to divergent gene complement.

Comparison	without missing homologs n	d	d/n	missing homologs replaced n	d	d/n	Δn	Δd	$\frac{\Delta d}{\Delta n}$
Populus-Vitis [65]	2104	1092	0.52	6144	2545	0.41	4040	1453	0.36
Populus-Carica [61]	2590	1461	0.56	7222	3466	0.48	4632	2005	0.43
Ricinus-Vitis	13694	8283	0.60	18300	9931	0.54	4606	1682	0.37
R.-V. corrected	13694	7715	0.56	18300	9365	0.51	4606	1684	0.37

Table 5.2: Normalized distances and insertion costs for three comparisons. Last row contains corrections to distance due to scaffold fusions. *Populus* comparisons include rearrangements during the re-diploidization of the ancestral tetraploid. Note that the missing homologs in the *Ricinus* comparisons are simply not sequenced, while in the previous comparisons, generally missing homologs are actually absent from the genome.

5.7 Conclusions

One methodological difficulty inherent in our comparison of *Ricinus* and *Vitis* is that of ortholog identification. BRH, which we used, is the simplest approach to this problem, using only sequence similarities, but there are many others available such as OrthoMCL [68], Inparanoid [69] and MSOAR [70], which can also make use of local order and gene ontology information.

Aside from improvements in orthology identification, which is a major roadblock to all gene order reconstruction problems, not only the scaffold assembly problem discussed here, there are a number of immediate possibilities to extending our technique. One is to take into account gap sizes on the scaffolds and gene sizes for the orthologs. As it is our reconstruction does not limit how many genes of whatever size can go into a gap.

A second, associated, problem would be to allow overlapping scaffolds, in cases where the paired ends data might not be resolved enough to preclude this configuration. We have already done this to some extent, in treating the single-gene scaffolds in the same way as missing genes. These small scaffolds are thus being inserted into

the gaps in other scaffolds. Allowing more general overlapping might complicate the algorithm, but in practice this could be a rare occurrence.

In the present work, we have assumed G_1 to be fully sequenced and G_2 to be in scaffolds. This is reasonable even though there are some gaps in the *Vitis* genome; there are not likely to be a large proportion of genes that remain unsequenced as there are in *Ricinus*. In other contexts, however, it might be desirable to expand our theoretical and practical considerations to allow both genomes to be in scaffold form. Here it may be necessary to insert missing orthologs in both directions, from G_2 to G_1 as well as from G_1 to G_2.

We have devoted much effort to differentiating between unsequenced genes and genes that are truly absent from the genome. Our goal here has been to predict the location of those genes that are missing because of incomplete sequencing or unsuccessful gene identification, not those genes that are absent because they have been deleted from *Ricinus* over time or acquired by *Vitis* since the divergence of the two lineages. Yet the latter class of genes are forced into our *Ricinus* reconstruction, because we have no *a priori* way of knowing they are actually absent from *Ricinus*. Our procedure would work equally well if instead of using all the missing *Vitis* genes, we used only those for which we had unigene, EST, RNA sequence or other cDNA evidence of their existence in *Ricinus*. We could then apply our algorithm to reconstruct the *Ricinus* gene order based on that of a reduced version of *Vitis* where all the genes with no *Ricinus* ortholog would be deleted at the outset from the *Vitis* gene order.

We discussed the case of BAC sequencing where the scaffolds are anchored on chromosomes so that there is no issue of optimal scaffold fusion. Gaps can still occur between BACs, and even inside BAC sequence assemblies, depending on the strategies and policies of the sequencers. Here our algorithm would require no modification to do a rearrangement analysis and ortholog insertion.

There are many occurrences of non-uniqueness in rearrangement inference and ortholog insertion in applying methods such as ours. This precludes a straightforward

comparison of $\overline{G}_2$ with the pre-deletion simulated genome to validate the method. However, non-uniqueness can sometimes be partially resolved by examining elements in common from many optimal solutions.

It bodes well for future use of this methodology that our algorithm was efficient enough to solve the problem with over 18,000 genes in less than a minute of computing time on a laptop computer, putting virtually all genomes within range of this technology.

Chapter 6

Genome Rearrangements, Evolutionary Breakpoints and Gene Expression: The hypothesis of neutrality

6.1 Introduction

The phenotypic consequences of genome rearrangements in humans, such as infertility or developmental pathologies when these mutations occur in the germ line, and cancer when they occur in somatic cells, are well documented [78] and often understood down to the level of changes in gene expression. The classic example is the Philadelphia t(9;22)(q34;q11) translocation creating the Philadelphia chromosome [79] and the BCR-Abl fusion gene whose tyrosine kinase product has wide-ranging molecular interactions ultimately responsible for chronic myeloid leukemia. The situation with the homozygotic rearranged genomes of reproductively isolated populations is quite different. The breakpoints of the evolutionary rearrangements differentiating

these genomes are known to co-occur with a large number of genomic features, such as regions that are gene-rich regions, GC-rich, hypomethylated, duplicated, pericentromeric or subtelomeric, as often reviewed (e.g., [80]), but the functional consequences of individual breakpoints remain virtually unknown, and there are no direct genome-wide studies of breakpoints from this point of view.

In the previous chapters, we were concerned with the creation of breakpoints and their position in the chromosome. The question here is what the biological consequences of breakpoint creation are, rather than just their structural aspects. Obviously, we cannot expect the effects of pathology-related rearrangement in somatic cells or germ-line cells to be perpetuated in evolutionary changes, simply because they are lethal or deleterious mutations. Since this type of change can be ruled out, are the observed evolutionary rearrangements selectively neutral? Or do they tend to represent selective advantages? Since any such advantage will be mediated by changes of gene expression, this will be the focus of my study. This work requires computational methods different from those in the earlier chapters, namely the integration of datasets and the development of statistical methods for genomics.

The question I will ask is whether proximity to the site of a breakpoint event changes the activity of a gene. In this chapter, I propose a paradigm for this type of investigation. The idea is basically to compare any changes of expression of genes that are close to, or even disrupted by, chromosomal breakpoints in the comparison of two genomes with changes affecting the gene complement more generally, controlled of course for tissue and experimental conditions. This is not a trivial exercise. There are now high-resolution techniques to identify breakpoint regions [76, 71], and thousands of data sets containing the results of whole-genome microarray assays, but comparative, whole genome data sets, controlled for tissue, with orthologous chromosomal positions specified for two species, are not easy to come by [77].

I propose to investigate the functional consequences of rearrangements by comparing the distribution of distances to the nearest breakpoint of genes that change

expression in human versus the distribution of genes that do not change expression in human, compared to other primate species, specifically rhesus macaque.

This is an exploratory project in several ways, since although some researchers have wondered about a change of activity near a breakpoint, nobody has systematically investigated it before. Whether the appropriate data even exists is not *a priori* clear.

We have been able to make use of one, relatively early, tissue-controlled comparison [72] of orthologs in humans and non-human primates, despite its lack of chromosomal positioning of genes, and its obsolete gene nomenclature. In the next section we describe a breakpoint dataset, an ortholog expression dataset and a human genome database necessary to relate the two other datasets. Next, we describe the method for linking the breakpoint dataset with the expression dataset and its implementation (Section 6.3). In Section 6.4, we elaborate the null hypothesis, also known as the neutral mutation model, in simple formal terms. In Section 6.5, we present the statistical results of our study on change of expression near breakpoints. We find little evidence for rejecting the neutralist hypothesis, but attribute this to sparse data and relatively crude measures of fold changes. Then, in the Conclusions, we discuss the potential for larger scale studies within this paradigm.

6.2 The data

Perhaps, the hardest part of this research is to locate, among the many hundreds of bioinformatics databases, the ones that contain the pertinent information. This needs to be in a form which enables us to link the variables in the hypothesis in a consistent way. One of the difficulties in comparing gene expression on eukaryote genomes has been the lack of a breakpoint database for the higher eukaryotes with sequenced genomes. Fortunatedly, however, the identification of breakpoints in the human genome as compared to other eukaryote species has recently been done

in [71]. This research carefully constructed an inventory of breakpoints in three primate species including human, and three non-primate mammalian species.

6.2.1 The breakpoint database

Lemaitre et al. [71] compared the genome of human and five sequenced eutherian mammals that are listed in Table 6.1 using a methodology that allowed them to delineate evolutionary breakpoint regions along the human genome with a finer resolution than observed previously. They excluded chromosome Y from their analysis because it was not available for all listed species in Table 6.1.

They defined a breakpoint region *BRP* in the human genome as "a region that underwent at least one large chromosomal structural change, or is orthologous to such region in a non-human lineage" .

species genomes	releases of genome assemblies	databases
human	May, 2004	UCSC browser
chimp	March, 2006	UCSC browser
macaque	February, 2006	UCSC browser
mouse	December, 2005	UCSC browser
rat	December, 2004	UCSC browser
dog	May, 2005	UCSC browser

Table 6.1: Datasets for genome sequences

They performed pairwise comparisons between human and other mammals and identified 622 non-intersecting BRPs ranging from 1 to 2,887,673 nucleotides with a mean size of 104 kb. Those 622 BRPs are stored in a database of sets of coordinates of breakpoints, organized by chromosome in the following format:

<chromosome, begin of BRP, end of BRP, evolutionary branch assignment>

An example of their database is illustrated in Table 6.2.

To compare the macaque genome to the human, we extracted only those breakpoints, 92 of them, on evolutionary branches leading to these species from their most

chromosome	begin	end	evolutionary branch assignment
chr1	10382322	10382387	dog
chr1	109923784	109923788	chimp
chr1	143495190	143766399	macaca
chr1	144691208	144707142	primates
chr1	144850157	145574145	chimp
chr1	225101534	225105289	human-chimp
chr3	126855424	127207816	primates
chr3	128287101	128299344	macaca
chr5	102756311	102787215	mouse
chr5	110090786	110287080	rodents
chr5	112304457	112304458	rat
chr22	34277622	34286037	rodents
chr22	37056914	37068605	rat

Table 6.2: Database of BRPs

recent common ancestor, namely those labelled in the dataset as human, human-chimp or macaca (macaque), as illustrated in Fig. 6.1. All other breakpoints are found in both human and macaque or in neither.

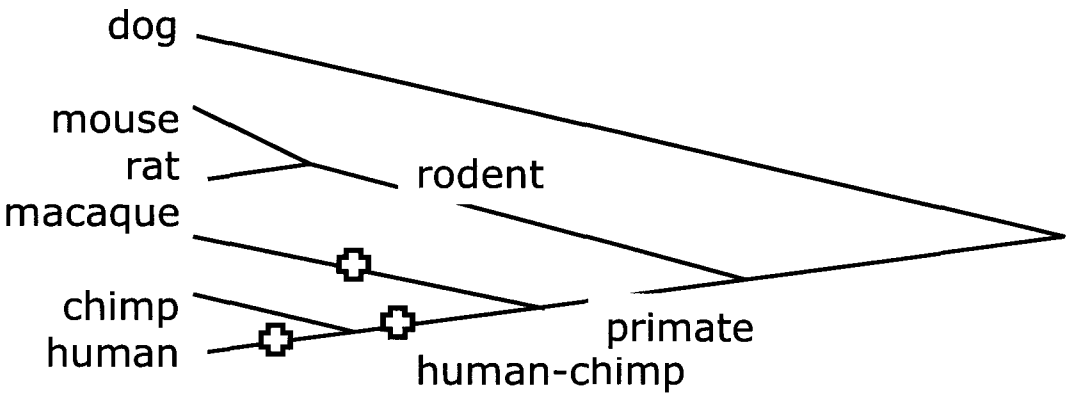

Figure 6.1: Phylogeny of species in the breakpoint database, with branches pertinent to the human-macaque comparison indicated.

It is important to note that there is no systematic accounting of gene expression or even of gene information in the BRP database, although these features of the human genome (but not in other genomes) played a role in the characterization of

BRP regions in [71].

6.2.2 The gene expression database

Dillman et al. [72] analyzed human whole blood tissue and three closely related non-human primates (NHP) namely the rhesus macaque (*Macaca mulatta*), the cynomolgous macaque (*Macaca fascicularis*), and the African green monkey (*Chlorocebus aethiops*). Each of their probe sets was defined by 54,000 probes, representing 38,500 genes from the completely sequenced human genome (2004 release).

The gene expression profiles for non-human primates (NHP) and human whole blood tissue were compared using a variety of statistical techniques (principal components, hierarchical clustering, analysis of variance) in order to find genes differentially expressed in humans and NHPs. They reported in [72] those genes that were found significant to a Bonferroni-corrected $p < 0.05$. They performed a 4-wise interspecies comparison of the four species that revealed those genes differentially expressed in human whole blood tissue compared with NHP whole blood tissue. The results, which include genetic elements identified as genes, mRNAs and ESTs, are stored in a database with the following format:

<Probe set Id, Accession Id, Gene Title, Bonferroni-corrected p value, Fold change>

where *Fold change* represents a human:NHP ratio

An extract of their database is illustrated in Table 6.3

Note that where these data tell us a gene is expressed more in one genome than the other, it does not tell us whether expression increased in the first genome since divergence from a common ancestor, or whether it decreased in the other.

We extracted 317 genetic elements with significant fold change from this table to use in testing our hypotheses. It is important to note that there is no gene coordinate or BRP information in the gene expression database. Thus the crux of our investiga-

Probe Set ID	Accession ID	Gene Title	corrected p value	Fold change
1553570_x_at	NM_173705	cytochrome c oxidase subunit II	$9.58E-12$	2.566336
1553538_s_at	NM_173704	—	0.00148793	2.49123
238199_x_at	AI708524	similar to OK/SW-CL.16	$8.98E-10$	2.415496
208450_at	NM_006498	lectin, galactoside-binding, soluble, 2 (galectin 2)	$5.67E-09$	1.7072
208581_x_at	NM_005952	metallothionein1X	0.0103346	-1.239609
208308_s_at	NM_000175	glucose phosphate isomerase	0.00294922	-1.497637

Table 6.3: Database of genetic elements differentially expressed in human whole blood tissue compared with NHP whole blood tissue.

tion is to relate these unpositioned expression data associated with gene names (see Table 6.3) to the breakpoint data (see Table 6.2), which is simply positional, with no gene names. To do so, we require a database containing both name and positions of human genes. The relation between those databases will be described in section 6.3.1.

6.2.3 The genome database

Since the breakpoint data are expanded in coordinates specified by the UCSC genome browser, we will cite the entire set of human genes from the browser as a baseline with which to test our differentially expressed genes. These human genes are stored in a database that has the following format:

<name, chrom, strand, txStart, txEnd, cdsStart, cdsEnd, exonCount, exonStarts, exonEnds, proteinID, alignID>

An extract from the UCSC human genome database is illustrated in Table 6.4.

name	chrom	strand	txStart	txEnd	cdsStart	cdsEnd
NM_024796	chr1	-	801449	802749	801942	802434
NM_001005484	chr1	+	58953	59871	58953	59871
NM_001005221	chr1	+	407521	408460	407521	408460
NM_001005221	chr1	-	660958	661897	660958	661897
NM_006498	chr22	-	36290753	36300524	36290769	36300412

Table 6.4: Database of human genes found in the human genome (May, 2004), UCSC browser.

6.3 Making connections

In this section, we first sketch the general protocol for linking the breakpoint dataset with the expression data set via the UCSC gene browser. We then describe how we implemented this in a way that can handle data sets much larger than those available for the present study. As quantitative measures of gene expression become more accurate, as data on multiple tissues are generated, and as we compare more highly rearranged genomes, it will be useful to have a high throughput system to do generate the data for statistical analysis.

6.3.1 Linking the breakpoint and gene expression databases via gene name and gene position

The protocol is as follows.

1. Scan the gene expression database for genes showing significant fold change in the human-macaque comparison and extract the human gene name (e.g. *NM_006498*) using the *Accession ID* field (cf. in Table 6.3).

2. Locate the records for these differentially expressed genes in the UCSC browser, matching gene names with the *name* field shown in Table 6.4 (e.g. *NM_006498* gene). This step is not fully automated since a good proportion of the "names"

in the expression database are not gene names at all, but are ESTs or transcripts of part of the gene, which can be located in other UCSC browser files, or obsolete gene names, which have to be tracked down by web search. A full 50 of the 317 differentially expressed elements did not have hits at all in the UCSC browser, and had to be dropped from our analysis.

3. Extract the chromosome number and coordinates of these genes in the human genome, using the fields *chrom*, *cdsStart* and *cdsEnd*, as in Table 6.4.

4. Having the coordinates *cdsStart* and *cdsEnd* and chromosome *chrom* of each differentially expressed gene extracted, we can now compute the distance in nucleotides to the closest BRP (*begin*, *end*) in the BRP database (Table 6.2), referring to the *chromosome* field.

5. Similarly, for all the human genes in the UCSC browser that do not match those differentially expressed genes previously identified, compute the distance in nucleotides to the closest BRP.

Having extracted all these data on 267 differentially expressed genes from the expression database, as well as the corresponding information on the rest of the human gene complement, we are now in a position to treat them statistically.

6.3.2 Implementation

We designed a relational database schema, implemented in PostgreSQL [81], to integrate the three different kinds of dataset: BRPs, differentially expressed genetic elements and all human genes. We loaded this with the data described in Sections 6.2.1, 6.2.2 and 6.2.3. We also loaded UCSC browser counterparts of the mRNA and EST entries found in the differentially expressed genes database. In addition, we loaded the entire set of human genes from the UCSC Known Genes Table.

We queried the relational database with a series of SQL statements implementing the different steps described in Section 6.3.1 in order to link the information and compute the distance d between each differentially expressed gene and its closest BRP, as well as the distance between each gene in the remainder of the human gene complement, and its closest BRP. After that, we loaded those distances d into spreadsheets in order to do the statistical analysis and produce the histograms that will be presented in Section 6.5. The construction of the histograms was based on the $log_{10}(d)$ and on an equal distribution of the frequencies of $log_{10}(d)$ into 9 integer values ranging from 1 to 9. After calculating the distribution of frequencies of $log_{10}(d)$ for the differentially expressed genes, those frequencies were normalized dividing them into the number of differentially expressed genes. A histogram was constructed for the not differentially expressed genes in a similar way. Then, both histograms were combined into a single chart (e.g. Figure 6.3) in order to compare the distribution of frequencies of distances for the nearest breakpoint for differentially and not differentially expressed genes.

6.4 The neutral model

Were there no association between breakpoint creation and change of expression of neighbouring genes, we would expect changed-expression genes to be spatially distributed independently of breakpoint positions. Consider the interval determined by the position a_1 and a_2 of the two breakpoints on either side of a changed- expression gene. Let $u = |a_1 - a_2|/2$. The position of the gene, considered as a random variable y should be uniformly distributed in the interval $[y_{min}, y_{min} + 2u]$ where $y_{min} = \min(a_1, a_2)$. The distance x to the closest breakpoint will then be distributed as a uniform variable on the interval $[0, u]$.

For visualization purposes, since the scale of intergenic distances is of the order of hundredths or thousandths of inter-breakpoint distances, we will study the distribution of $z = \log x$ rather than of x. Since x is uniform on $[1, u]$, the probability

density of z will have the form of a truncated positive exponential distribution

$$p(z) = e^{z-u}, \tag{6.4.1}$$

for $0 \leq z \leq u$, as in Fig. 6.2a.

Since the distance $2u$ between the breakpoints will itself be distributed randomly (as the distance between two order statistics, namely a negative exponential) and depend on the length of the chromosome and the number of breakpoints, the empirical distribution of distances is predicted by a sum of variables, all with density $p(z)$ but with different parameters u, as in Figure 6.2b.

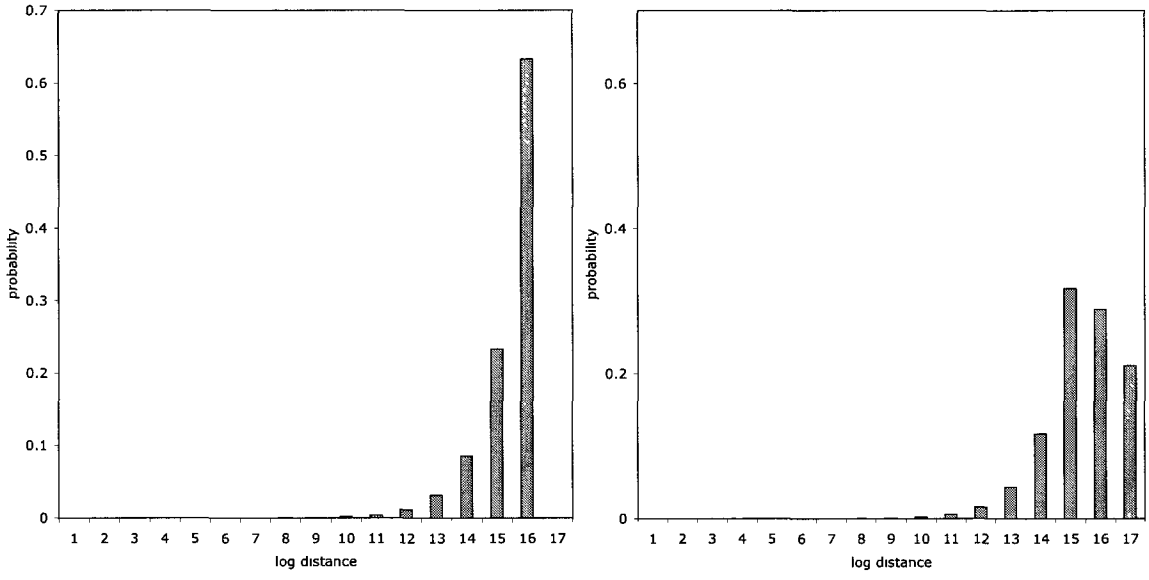

Figure 6.2: a) Distribution of z = log distance to nearest breakpoint, under the neutral hypothesis, with $u = e^{16}$. b) Predicted empirical frequency distribution, based on equally weighted $u = e^{15}, e^{16}, e^{17}$.

6.5 Results

Fig. 6.3 compares the distance to the nearest breakpoint of differentially expressed genes to that of the entire set of human genes. While the shape of the distribution is generally as expected from our model in Section 6.4, it is clear that there is little difference between the distributions for the differentially expressed genes and the rest of the human gene complement. This is what we would expect if rearrangement is generally neutral with respect to gene expression. However, this is scarcely a conclusive test, given the available data, on just one pair of species (though, strictly speaking, the NHP data comes from three very closely related genomes), with few breakpoints, and with measures of gene expression that are relatively crude compared to today's methodology.

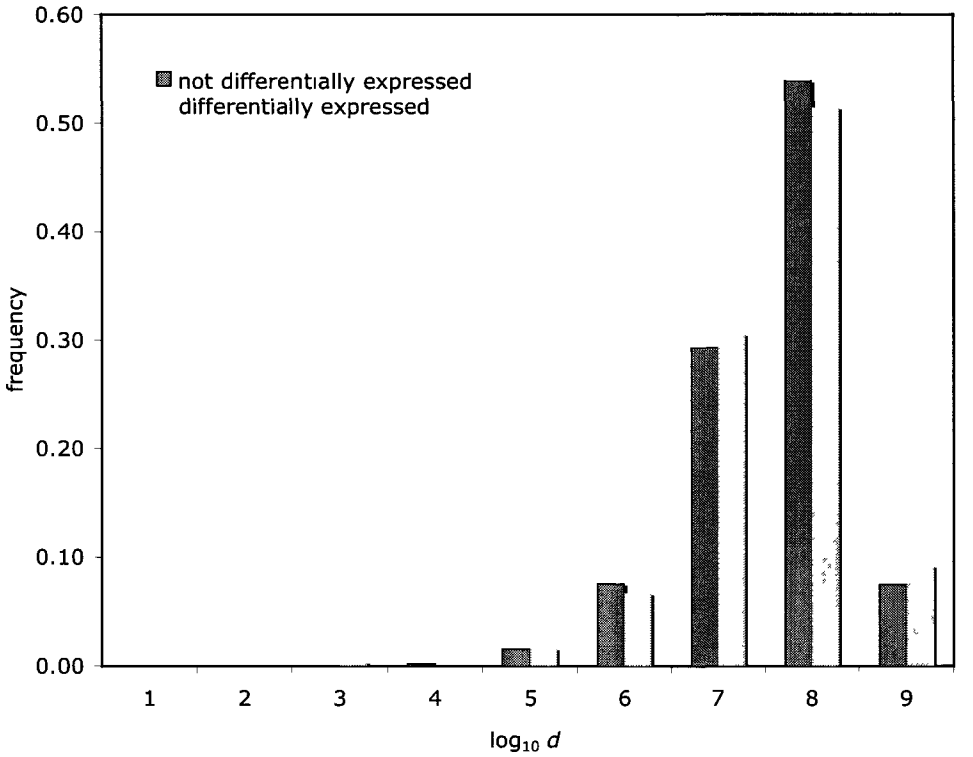

Figure 6.3: Histogram of distances from genes to closest BRP for differentially expressed genes versus not differentially expressed genes for all chromosomes

We did identify a number of differentially expressed genes close to BRPs. one in chromosome 16 where $d = 216$ and four elsewhere with $d < 10^5$. As a visualization tool, our distribution on a log scale depicts this neatly, as in Fig. 6.4. for genes in chromosome 16 and chromosome 1.

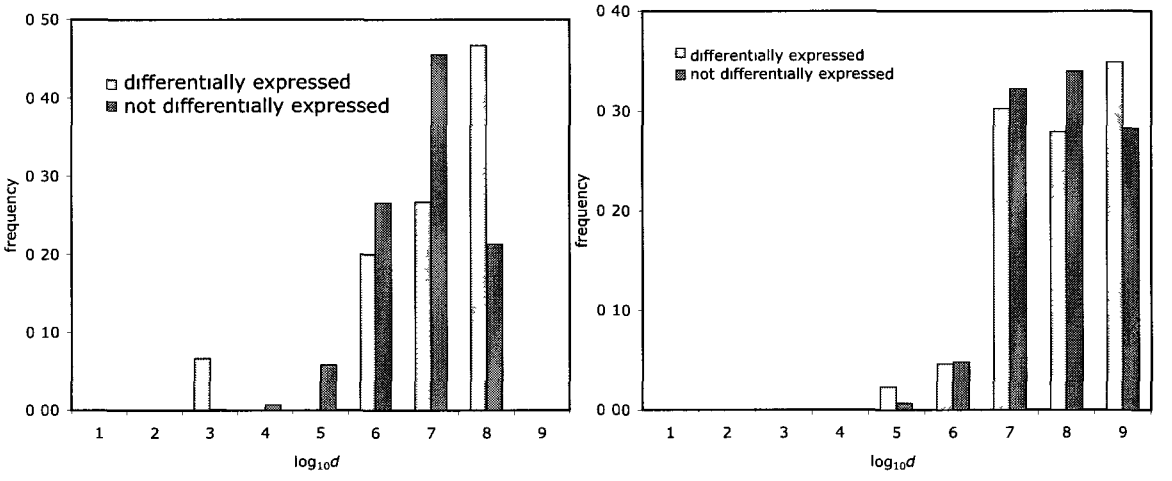

Figure 6.4: Histogram of distances from genes to closest BRP for differentially expressed genes versus not differentially expressed genes for chromosome 16 and for chromosome 1

Though the few differentially expressed genes close to breakpoints that we found do not seem to be functionally related, it is of interest that one of those on chromosome 1, the *NBPF10* gene, is member 10 of the Neuroblastoma breakpoint gene family. This novel gene family named *NBPF* for Neuroblastoma breakpoint gene family was identified by Vandepoele et al. [82] who speculated that *NBPF* genes played a role in the evolution of primates, including human because a recent expansion of the *NBPF* sequences indicated in primate genomes resulted in species-specific genes.

6.6 Conclusions

Genome rearrangement research has been concerned with the creation of breakpoints and their position in the chromosome. The question arises of what the biological

consequences of breakpoint creation are, rather than just their structural aspects.

We have asked whether proximity to the site of a breakpoint event changes the activity of a gene. We investigated this by comparing the distribution of distances to the nearest breakpoint of genes that change expression after rearrangement with the same distribution for those that do not change. This question has not been investigated previously on a genome-wide basis.

The data currently available on individual gene expression change across entire genomes for different species is limited. That we found little evidence for rejecting the neutralist hypothesis is attributable to sparse data and to relatively crude measures of fold changes. With the advent of Next Generation Sequencing, quantitative RNA sequence data on many tissues from related species should soon become available. [1] Our methodology may be of utility at that time.

[1] such as the expression data controlled for species, developmental stage and tissue, being developed by H. Kaessmann of EPFL in Lausanne.

Chapter 7

Conclusions and future work

My goal in this thesis was to increase the pertinence of the bioinformatics approach to the comparative structural analysis of genomes, first by extending it to previously unusable genomic data and second by exploring a way of embedding it into functional genomics. I focused on two aspects: how to compare related genomes when the genome assembly is incomplete, and what are the consequences for gene expression for genes in the neighbourhood of chromosome rearrangements.

I first proposed an approach to correcting genome rearrangement distance when comparing genomes in contig form. This involved a careful treatment of contig concatenation imitating the evolutionary process of chromosome fusion. Since the parameters in my model do not depend on the number of genes nor the details of the rearrangement model such as the number of chromosomes or the proportions of different types of rearrangement, assuming the latter are naturally weighted as in the double-cut-and-join framework, my methods should be widely applicable beyond the *Drosophila* data.

In this work I posited a new relationship between the true number of rearrangements and the number inferred. The main shortcoming of this and other models such as in Ref [36] is that they are not analytically derived. Thus the mathematical

foundation of probability models and statistical analyses of genomic problems like the one addressed here would benefit more from advances like those in Ref [35] than by further characterization of empirical models such Eq. 2.5.3.

The contig data were not the typical output from a genome project. I used data on neighbouring gene pairs (NGP) to generate the contigs. It is unlikely that complete NGP data will be constructed for many genome projects as they were for *Drosophila*. Genomes in contig form may nevertheless be produced if the strategy of sequencing paired ends of long insert size are not available or not used for some reason. However, we may expect that most projects will try to construct long scaffolds as better representation of overall chromosome structure. This led me to focus next on the scaffold-filling problem.

In my work on scaffold filling, I assumed a genome G_1 (e.g. *Vitis*) to be fully sequenced and a genome G_2 (e.g. *Ricinus*) to be in scaffolds. In other contexts, however, it might be desirable to expand our theoretical and practical considerations to allow both genomes to be in scaffold form. Here it may be necessary to insert missing orthologs in both directions, from G_2 to G_1 as well as from G_1 to G_2. This suggestion has been taken up by Binhai Zhu of Montana State University (personal communication).

The scaffold filling problem superficially resembles the problem of comparing unequal genomes, but it is quite different in fact. Comparing unequal genomes minimizes the number of block insertion and deletions of genes, whereas insertions and deletions cost nothing in scaffold filling; only genome rearrangements are minimized. Nevertheless there are likely research avenues combining the two problems, such as to choose among the multiple optimal solutions to one problem using the criteria of the other.

In the area of research relating genome rearrangements, evolutionary breakpoints and gene expression to study the functional consequences of breakpoint creation, I found little evidence for rejecting the neutralist hypothesis. I do not consider these

results definitive, however, since they may well be attributable to sparse data and to relatively crude measures of fold changes. Indeed, it is surprising how limited is the data currently available on individual gene expression change across entire genomes for different species, given the profusion of data sets on gene expression generated for other purposes.

With the advent of Next Generation Sequencing, quantitative RNA sequence data on many tissues from related species should soon become available that will give the opportunity to explore the potential for larger scale studies within this paradigm.

Publication Record and Collaborations

The following list of publications have been produced solely or partly due to the work described here. In each case, I append an assessment of the contribution of myself and of my collaborators, except for my supervisor, David Sankoff, who has played his advisory role in all aspects of this research.

Papers Refereed and Published in Journals

1. Muñoz A., Sankoff D. Rearrangement phylogeny of genomes in contig form. *IEEE/ACM Transactions in Computational Biology and Bioinformatics* 2010, Vol. 7 (4): 579-587.

2. Muñoz A., Zheng C., Zhu Q., Albert V.A., Rounsley S., Sankoff D. Scaffold filling, contig fusion and comparative gene order inference. *BMC Bioinformatics* 2010, **11**:304.

   The research in this paper is part of a larger project on the comparison of angiosperm genomes in collaboration with V. Albert and S. Rounsley. I coordinated this work and formulated the problem as an extension of my previous work on contigs. The extraction of the *Ricinus* and *Vitis* data was done by Q. Zhu, and the scaffold filling algorithm was designed by C. Zheng. Albert

and Rounsley suggested the test genomes. I connected the scaffold filling and the contig fusion methods, set up and carried out the simulations, and did the interpretation and write-up.

3. Sankoff D., Zheng C., Muñoz A., Yang Z, Adam Z., Warren R., Choi V., Zhu Q.. *Issues in the Reconstruction of Gene Order Evolution.* Journal of Computer Science and Technology, 2010, Vol. 25 (1): 10-25

   For this paper, I contributed Sections 2.5 and 5.2.

Papers Refereed and Published in Conferences

4. Muñoz A., Sankoff D. Rearrangement phylogeny of genomes in contig form. In *Bioinformatics Research and Applications (ISBRA). Fifth International Symposium* (Ion Măndoiu I, Narasimhan G, Zhang Y, eds), Lecture Notes in Computer Science 5542, Springer, 2009, pp. 160–172.

Papers Submitted to Conferences

5. Muñoz A., Sankoff D. Evolutionary breakpoints and gene expression: The hypothesis of neutrality. *RECOMB Satellite Workshop on Comparative Genomics* 2010 (submitted).

Presentations at Conferences

6. Muñoz A., Sankoff D.: Scaffolds, contigs and gene order comparison. *PAG'10*, San Diego, U.S.A., Jan/2010

7. Muñoz A.: Rearrangement phylogeny of genomes in contig form. *Robert Cedergren Bioinformatics Colloquium'09*, Montreal, Canada, Nov/2009

8. Muñoz A, Sankoff D: Comparaison de genomes sous form de contig. *ACFAS'09*, Ottawa, Canada, May/2009

Posters Presented at Conferences

9. Muñoz A., Zheng C., Zhu Q., Albert V.A., Rounsley S., Sankoff D. Gene Order Comparison With Contigs And Scaffolds. *ISMB* 2010, Boston, U.S.A.

10. Muñoz A.: Rearrangement phylogeny of genomes in contig form. *Robert Cedergren Bioinformatics Colloquium'09*, Montreal, Canada, Nov/2009

11. Muñoz A., Sankoff D.: Rearrangement phylogeny of genomes in contig form. *ISMB'09*, Stockholm, Sweden, June/2009

12. Muñoz A., Sankoff D.: Rearrangement phylogeny of genomes in contig form. *RECOMB Satellite Workshop on Comparative Genomics'08*, Paris, France, October/2008

13. Muñoz A., Sankoff D.: Rearrangement Algorithms and the Comparison of Genomes in Contig form. *ISMB'08*, Toronto, Canada, July/2008

Bibliography

[1] Venter J C, Adams M D, Myers E W, Li P W, Mural R J, Sutton G G *et al.* 2001. The sequence of the human genome. *Science*, 2001, 291: 1304–1351.

[2] Mouse Genome Sequencing Consortium. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 2002, 420: 520–562.

[3] Sankoff D, Leduc G, Antoine N, Paquin B, Lang B F, Cedergren R. Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome. *Proceedings of the National Academy of Sciences USA*, 1992, 89: 6575–6579.

[4] Sankoff D. Edit distance for genome comparison based on non-local operations. In *Combinatorial Pattern Matching (CPM). Third Annual Symposium* (Apostolico A, Crochemore M, Galil Z, Manber U, eds), Lecture Notes in Computer Science 644, Springer, 1992, pp. 121–135.

[5] Cosner M E, Jansen R K, Moret B M E, Raubeson L A, Wang L-S, Warnow T, Wyman S. An empirical comparison of phylogenetic methods on chloroplast gene order data in Campanulaceae. In *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families* (Sankoff D, Nadeau, J, eds), Kluwer Academic Publishers, Dordrecht, 2000, pp. 99–121.

[6] Ajana Y, Lefebvre J-F, Tillier E R M, El-Mabrouk N. Exploring the set of all minimal sequences of reversals - an application to test the replication-directed reversal hypothesis. In *Algorithms in Bioinformatics (WABI). Second International Workshop* (Guigo R, Gusfield D, eds), Lecture Notes in Computer Science 2452, Springer, 2002, pp. 300–315.

[7] Hannenhalli S, Pevzner P A. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 1999, 46: 1–27.

[8] Caprara A. Sorting permutations by reversals and Eulerian cycle decompositions *SIAM Journal on Discrete Mathematics*, 1999, 12: 91–110.

[9] Watterson G A, Ewens W J, Hall T E, Morgan A., The chromosome inversion problem. *Journal of Theoretical Biology*, 1982, 99: 1–7.

[10] Sankoff D. Mechanisms of genome evolution: models and inference. *Bulletin of the International Statistical Institute*, 1989, 47(3): 461–475.

[11] Sturtevant A H, Novitski E. The homologies of the chromosome elements in the genus *Drosophila. Genetics*, 1941, 26: 517–541.

[12] Assembly/alignment/annotation of 12 related Drosophila species. http://rana.lbl.gov/drosophila/

[13] Bhutkar A, Gelbart WM, Smith TF. 2007. Inferring genome-scale rearrangement phylogeny and ancestral gene order: a *Drosophila* case study. *Genome Biology* **8**: R236

[14] Bhutkar A, Schaeffer SW, Russo SM, Xu M, Smith TF, Gelbart WM. 2008. Chromosomal rearrangement inferred from comparisons of 12 *Drosophila* genomes. *Genetics* **179**: 1657–80.

[15] *Drosophila* 12 Genomes Consortium. Clark AG et al. 2007. Evolution of genes and genomes on the *Drosophila* phylogeny. *Nature* **450**: 203–218

[16] Krzywinski J, Grushko OG, Besansky NJ. 2006. Analysis of the complete mitochondrial DNA from *Anopheles funestus*: An improved dipteran mitochondrial genome annotation and a temporal dimension of mosquito evolution. *Molecular Phylogenetics and Evolution 2006* **39**: 417–423.

[17] Savard J, Tautz D, Richards S, Weinstock GM, Gibbs RA, Werren JH,Tettelin H, Lercher MJ. 2006. Phylogenomic analysis reveals bees and wasps (Hymenoptera) at the base of the radiation of Holometabolous insects. *Genome Research* **16**: 1334–8.

[18] Severson DW, DeBruyn B, Lovin DD, Brown SE, Knudson DL, Morlais I. 2004. Comparative genome analysis of the yellow fever mosquito *Aedes aegypti* with *Drosophila melanogaster* and the malaria vector mosquito *Anopheles gambiae*. *Journal of Heredity* **95**:103–13.

[19] Hannenhalli S, Pevzner P A. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Foundations of Computer Science (FOCS). Thirty-Sixth Annual Symposium*, IEEE Computer Society, 1995, pp. 581–592.

[20] Tesler G. Efficient algorithms for multichromosomal genome rearrangements, *Journal of Computer and System Sciences*, 2002, 65: 587–609.

[21] Sankoff D, Blanchette M. Multiple genome rearrangement and breakpoint phylogeny. *Journal of Computational Biology*, 1998, 5:555–570.

[22] El-Mabrouk N, Bryant D, Sankoff D. Reconstructing the pre-doubling genome. In *Computational Molecular Biology (RECOMB). Third Annual International Conference* (Istrail S, Pevzner P, Waterman M, eds), ACM Press, 1999, pp. 154–163.

[23] El-Mabrouk N, Sankoff D.The reconstruction of doubled genomes. *SIAM Journal on Computing*, 2003, 32: 754–792.

[24] Pevzner P, Tesler G. Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution. *Proceedings of the National Academy of Sciences USA*, 2003, 100:7672–7677.

[25] Kent W J, Baertsch R, Hinrichs A, Miller W, Haussler D. Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proceedings of the National Academy of Sciences USA*, 2003, 100:11484–11489.

[26] Kent W J, Sugnet C W, Furey T S, Roskin K M, Pringle T H, Zahler A M, Haussler A D. The Human genome browser at UCSC. *Genome Research*, 2002, 12: 996–1006.

[27] Mazowita M, Haque L, Sankoff D. Stability of rearrangement measures in the comparison of genome sequences. *Journal of Computational Biology*, 2006, 13: 554–566.

[28] Sinha A U, Meller J. Sensitivity analysis for reversal distance and breakpoint reuse in genome rearrangements. *Pacific Symposium on Biocomputing*, 2008, 13:37–38.

[29] Jiang T. Some algorithmic challenges in genome-wide ortholog assignment. *Journal of Computer Science and Technology*, 2010, Vol. 25 (1): 42-52

[30] Tannier E, Zheng C, Sankoff D. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics*, 2009, 10: 120.

[31] Fertin G, Labarre A, Rusu I, Tannier E, Vialette S. *Combinatorics of Genome Rearrangements*. Cambridge, Massachusetts: The MIT Press, 2009.

[32] Yancopoulos S, Attie O, Friedberg R. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 2005, 21: 3340–3346,

[33] Bergeron A, Mixtacki J, Stoye J. A unifying view of genome rearrangements. In *Algorithms in Bioinformatics (WABI). Sixth International Workshop*, (Bücher P, Moret BME, eds), Lecture Notes in Computer Science 4175, Springer, 2006, pp. 163–173

[34] Dalevi D, Eriksen N. Expected gene-order distances and model selection in bacteria. *Bioinformatics*, 2008, 24: 1332–1338.

[35] Eriksen N. Hultman A. Estimating the expected reversal distance after a fixed number of reversals. *Advances in Applied Mathematics*, 2004, 32: 439–453.

[36] Wang L-S, Warnow T. Distance-based genome rearrangement phylogeny. Ch. 13 in *Mathematics of Evolution and Phylogeny* (Gascuel O, ed), Oxford, 2005, pp. 353–383.

[37] Adam Z, Turmel M, Lemieux C, Sankoff D. Common intervals and symmetric difference in a model-free phylogenomics, with an application to streptophyte evolution. *Journal of Computational Biology*, 2007, 14: 436–445.

[38] Zhu Q, Adam Z, Choi V, Sankoff D. Generalized gene adjacencies, graph bandwidth, and clusters in yeast evolution. *Transactions on Computational Biology and Bioinformatics*, 2009, 6(2): 213–220.

[39] Tannier E. Yeast ancestral genome reconstructions: the possibilities of computational methods. In *Comparative Genomics (RECOMB CG). Seventh Annual Workshop*, (Ciccarelli F D, Miklós I, eds.), Lecture Notes in Computer Science 5817, Springer, 2009.

[40] Sankoff D, Blanchette M. The median problem for breakpoints in comparative genomics. *Computing and Combinatorics (COCOON). Third Annual Conference*, (Jiang T, Lee D T, eds), Lecture Notes in Computer Science 1276, Springer, 1997, pp. 251–263.

[41] *HPCwire*. GRAPPA runs In a record time. November 23, 2000, 9(47).

[42] Siepel A C. *Exact algorithms for the reversal median problem.* Master's thesis, University of New Mexico, 2001.

[43] Caprara A. On the practical solution of the reversal median problem. In *Algorithms in Bioinformatics (WABI). First International Workshop* (Gascuel O, Moret B M E, eds), Lecture Notes in Computer Science 2149, Springer, 2001, pp. 238–251.

[44] Bourque G, Pevzner P A. Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Research*, 2002, 12: 26–36.

[45] Xu A W. A fast and exact algorithm for the median of three problem – a graph decomposition approach. In *Comparative Genomics (RECOMB CG). Sixth Annual Workshop* (Nelson C E, Vialette S, eds), Lecture Notes in Computer Science 5267, Springer, 2008, pp. 184–197.

[46] Xu A W, Sankoff D. Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. In *Algorithms in Bioinformatics (WABI). Eighth International Workshop* (Crandall K A, Lagergren J, eds), Lecture Notes in Computer Science 5251, Springer, 2008, pp. 25–37.

[47] Adam Z, Sankoff D. A statistically fair comparison of ancestral genome reconstructions, based on breakpoint and rearrangement distances. In *Comparative Genomics (RECOMB CG). Seventh Annual Workshop*, (Ciccarelli F D, Miklós I, eds.), Lecture Notes in Computer Science 5817, Springer, 2009, pp. 193–204.

[48] Adam Z, Sankoff D. The ABCs of MGR with DCJ. *Evolutionary Bioinformatics*, 2008, 4: 69–74.

[49] Chen X, Cui Y. An approximation algorithm for the minimum breakpoint linearization problem. *Transactions on Computational Biology and Bioinformatics*, 2009, 6: 401–409.

[50] Gaul E, Blanchette M. Ordering partially assembled genomes using gene arrangements. In *Comparative Genomics (RECOMB CG). Fourth Annual Workshop* (Bourque G, El-Mabrouk N, eds), Lecture Notes in Computer Science 4205, Springer, 2006, pp. 113–128.

[51] Bhutkar A, Russo S, Smith T F, Gelbart W M. Techniques for multi-genome synteny analysis to overcome assembly limitations. *Genome Informatics*, 2006, 17: 152–161.

[52] Muñoz A, Sankoff D. Rearrangement phylogeny of genomes in contig form. In *Bioinformatics Research and Applications (ISBRA). Fifth International Symposium* (Ion Măndoiu I, Narasimhan G, Zhang Y, eds), Lecture Notes in Computer Science 5542, Springer, 2009, pp. 160–172.

[53] Zheng C, Zhu Q, Sankoff D. Removing noise and ambiguities from comparative maps in rearrangement analysis. *Transactions on Computational Biology and Bioinformatics*, 2007, 4: 515–522.

[54] Choi V, Zheng C, Zhu Q, Sankoff D. Algorithms for the extraction of synteny blocks from comparative maps. In *Algorithms in Bioinformatics (WABI). Seventh International Workshop* (Giancarlo R, Hannenhalli S, eds), Lecture Notes in Computer Science 4645, Springer, 2007, pp. 277–288.

[55] Sankoff D, Haque L. Power boosts for cluster tests. In *Comparative Genomics (RECOMB CG). Fifth Annual Workshop* (McLysaght A, Huson D, eds), Lecture Notes in Bioinformatics 3678, Springer, 2005. pp. 121–130.

[56] Burleigh JG, Bansal MS, Wehe A, Eulenstein O. 2009. Locating large-scale gene duplication events through reconciled trees: implications for identifying ancient polyploidy events in plants. *Journal of Computational Biology* **16**: 1071–1083.

[57] Jaillon O, Aury JM, Noel B, *et al.* 2007. The grapevine genome sequence suggests ancestral hexaploidization in major angiosperm phyla. *Nature* **449**:463–467.

[58] Muñoz A, Sankoff D: Rearrangement phylogeny of genomes in contig form. *IEEE/ACM Transactions in Computational Biology and Bioinformatics* 2010, Vol. 7 (4): 579-587.

[59] Sankoff D, Zheng C, Zhu Q. The collapse of gene complement following whole genome duplication. *BMC Genomics*, 2010, **11**: 313.

[60] Chain PSG, Grafham DV, Fulton RS, FitzGerald MG, Hostetler J, Muzny D, Ali J, *et al.*: Genome project standards in a new era of sequencing. *Science* 2009, **326**:236–237.

[61] Sankoff D, Zheng C, Wall P K, dePamphilis C, Leebens-Mack J, Albert V A. 2009. Towards improved reconstruction of ancestral gene order in angiosperm phylogeny. *Journal of Computational Biology* **16**.

[62] Soltis DE, Albert VA, Leebens-Mack J, Bell CD, Paterson AH, Zheng C, D. Sankoff D, dePamphilis CW, Wall PK, Soltis PS. 2009. Polyploidy and angiosperm diversification. *American Journal of Botany* **96**:336–348.

[63] Thomas BC, Pedersen B, Freeling M. 2006. Following tetraploidy in an Arabidopsis ancestor, genes were removed preferentially from one homeolog leaving clusters enriched in dose-sensitive genes. *Genome Research* **16**:934–946.

[64] Velasco R, Zharkikh A, Troggio M, Cartwright DA, Cestaro A, Pruss D, Pindo M, *et al.* 2007. A high quality draft consensus sequence of the genome of a heterozygous grapevine variety. *PLoS ONE* **2**: e1326.

[65] Zheng C, Wall PK, Leebens-Mack J, dePamphilis C, Albert VA, Sankoff D. 2009. Gene loss under neighbourhood selection following whole genome duplication and the reconstruction of the ancestral populus genome. *Journal of Bioinformatics and Computational Biology* **27**: 499–520.

[66] Seguin M, Flori A, Legnaté H, Clément-Demange A: **Rubber tree.** In *Genetic diversity of cultivated tropical plants.* Edited by Hamon P, Seguin M, Perrier, X, Glaszmann, C. Montpellier: CIRAD; 2003:277–306.

[67] Zheng C. Path groups: a dynamic data structure for genome reconstruction problems. *Bioinformatics*, 2010, , **26**:1587-1594.

[68] Li L., Stoeckert CJ Jr, Roos DS: OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Research* 2003, **13**:2178–2189.

[69] O'Brien KP, Remm M, Sonnhammer EL: Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Research* 2005, **33**:D476–80.

[70] MSOAR. A high-throughput ortholog assignment system [http://msoar.cs.ucr.edu/]

[71] Lemaitre C, Zhagloul L, Sagot MF, Gautier C, Arneodo A, Tannier E, Audit B. Analysis of fine-scale mammalian evolutionary breakpoints provides new insight into their relation to genome organisation. *BMC Genomics*, 2009, 10: 335.

[72] Dillman JF III, Phillips CS. Comparison of Non-Human Primate and Human Whole Blood Tissue Gene Expression Profiles. *Toxicological Sciences*, 2005, 87(1), 306314.

[82] Vandepoele K, Van Roy N, Staes K, Speleman F, Van Roy F. 2005. A Novel Gene Family: *NBPF* Intricate Structure Generated by Gene Duplications During Primate Evolution. *Mol. Biol. Evol.* 22(11): 2265-2274.