



Root & Nodule Quantification Software

SOFTWARE USER GUIDE

December 9, 2013

CONTENTS

Introduction.....	3
Getting started	4
Setting the search path in Matlab	4
Running programs in Matlab	4
Troubleshooting in Matlab	5
Root & Nodule Quantification Programs.....	6
ListDataset2	6
GetRes4	6
RootSkel.....	6
PrimaryRoot2.....	7
SmoothPrimary.....	7
LatRoots4.....	7
SegLaterals12.....	8
SmoothLaterals11.....	9
PotentialNodules	10
LinkNods2Roots7	10
NodulesUI7	10
NodExtractData16	10

INTRODUCTION

This software was developed for recording and quantifying nodule distribution patterns in a standardized way, based on image analysis. The software can be used to quantify differences in nodulation patterns among cultivars, under different environmental conditions, and/or between different legume species. Such quantitative phenotyping will make it possible to explore the link between nodulation patterning, root system architecture, and ultimately plant yield.

Please refer to the associated publication for more information. This publication contains essential information to understand how the images need to be taken, how the software works, and what data are extracted.

Please also refer to the **Instructions for Imaging Root Systems for RNQS** before starting your experiment for important details on imaging your plants appropriately for the software. If these instructions are not followed carefully, the software will not work optimally, or may result in errors.

The first sections of this Software User Guide will briefly explain how to get set up and how to run the software in Matlab and other general information you'll need to get started.

The 'Programs' section contains the names and explanations of programs you will run to extract root nodule data.

Many of the programs do not require user input; for those that do, *demonstration htm files* are provided to aid in illustrating how to use them in the folder "RNQS program demonstrations".

You can ignore the Courier font text in the user guide - these are for programming purposes only (indicating what data are saved in each program, the name of the mat-file and variables).

GETTING STARTED

SETTING THE SEARCH PATH IN MATLAB

1. Open Matlab as you would any other program (click on the desktop icon or the programs list in Windows).
2. The Matlab command window is the big blank region where you see ">>". Copy and paste the following statement into the command window and press enter.
`addpath(uigetdir(cd,'Select the Root Nodule Software folder'));`
Alternatively, navigate to the 'Set Path' button (Matlab 2013) or menu option (under 'File' tab in older versions of Matlab).
3. This will open a standard dialogue box. Use it to select the folder
C:\...\Root & Nodule Qunatification Software\RNQS programs
that you downloaded and extracted on your computer.

See [Launch MATLAB and set path demo.htm](#).

RUNNING PROGRAMS IN MATLAB

Now to run any program you can simply type (or copy and paste) the name of the program (see following section) into the Matlab command window and press enter.

See [Intro to using Matlab with ListDataSet demo.htm](#) .

TROUBLESHOOTING IN MATLAB

If you get any errors while running the programs, please copy the error message (i.e. red text appearing in the command window) and provide this to

arolland@uottawa.ca or laurenremmler@gmail.com

- If you get the error
??? Undefined function or variable 'FunctionName'
 - o Check that you did not make a typo in the function name. Note that function names are case sensitive.
 - o If the function name was entered correctly, try re-setting the path (see last step of "Before running programs for the first time").

- To interrupt the execution of a program press Ctrl+Pause.

- If the program crashes or gives an "Out of memory error", consult your IT specialist. You may need to increase the amount of RAM on your computer. We have processed the images without problem using 3.4GB RAM on Windows XP.

- Some errors could be the result of inadequate photo quality or problem with the way the sample was prepared. If this seems to be the case, try removing this image from the data set.

- IMPORTANT NOTE IF YOU COPY YOUR MAIN PROJECT FOLDER TO A NEW LOCATION DURING DATA PROCESSING:
The software generates a file called dataset.mat which records information on where files are stored. This file is generated by the first program (listdataset2) and accessed by later programs. If you happen to copy the folder containing your project to another location (e.g. another computer) after that file has been generated, you will get an error when running all subsequent programs. To fix this issue and update the dataset.mat file after having moved/copied your folder to a new location, you have to run the program named convertpaths (just type convertpaths in the command window, you will be prompted to select your main project folder).

ROOT & NODULE QUANTIFICATION PROGRAMS

This software package consists of a number of programs to be run, as follows. Unless otherwise indicated, it is important the programs are run in the order presented.

Many of the programs do not require user input; for those that do, *demonstration files* are provided in the folder “RNQS program demonstrations” to aid in illustrating how to use the programs.

LISTDATASET2

TO MAKE A LIST OF ENTIRE DATA SET FOR PROGRAMS TO RUN THROUGH

You will be prompted by the program to show where the main project folder is by selecting that folder when prompted.

Make sure the image folder is called ‘Images’ (case sensitive).

See *Launch MATLAB and set path demo.htm* .

```
save([... '\dataset.mat'], 'dataset', 'imgpth', 'resfol')  
save([.../picinfo.mat'], 'groupID', 'pltID', 'age', 'plane', 'c_imgpth')
```

GETRES4

TO EXTRACT IMAGE RESOLUTION USING SCALE BAR

Enter the size of the scale bar in inches then press enter. For each plant an image will be shown. Click on opposite sides of the label, such that you get a straight line across the width of the label.

See *GetRes demo.htm*.

ROOTSKEL

TO EXTRACT ROOT SYSTEM SKELETON THROUGH AUTOMATIC IMAGE PROCESSING

No input needed.

```
save([... '/skel.mat'], 'mask', 'skel', 'allx', 'ally');
```

PRIMARYROOT2

TO GET PRIMARY ROOT COORDINATES

Click on the top and then the bottom coordinates of the primary root. The program will then attempt to trace the primary root.

Make sure you are very consistent about what you click as the top of the root (right at the base of the cotyledons) for all planes.

You can edit it if necessary using the commands displayed.

- Left click to force it through the primary root if it took the wrong path
- Right-click on part of the current path that is wrong to force it to look for another path (I find the former option usually works better than this).
- Backspace to start over (try to avoid doing this too much though).

The end of the primary root is hard to find sometimes because it is hidden by a mess of other roots (the dissecting pin is meant to point toward it, if that helps). Don't worry if you're not 100% sure, about it, it will mainly be used to measure the primary root length.

For P3 try to get it accurate enough to correctly measure the length of the primary root.

For P1-2 don't waste time fixing anything below the nodulation zone (i.e. anywhere below where there are nodules), as this data will not be used.

See [PrimaryRoot demo.htm](#).

```
save(['.../primary.mat'], 'tipx', 'tipy', 'topx', 'topy', 'prirc')
```

SMOOTHPRIMARY

TO SMOOTH OUT PRIMARY ROOT COORDINATES

No input needed.

Just visually check that the smoothed coordinates (pink) look okay.

LATROOTS4

TO EXTRACT LATERAL ROOTS THROUGH AUTOMATIC IMAGE PROCESSING

No input needed.

TO IDENTIFY INDIVIDUAL LATERAL ROOTS

To begin, it may be helpful to clarify and define some terms that will be used to explain this section:

- **Extracted coordinates** of the lateral root system (white): these are simply all of the points (or pixels) that the previous programs automatically extracted from the image as being part of the root system.
 - Note that sometimes not all of the coordinates of a root system will be automatically extracted (such as when a root is very thin or dark).
 - The coordinates are used to identify lateral roots, but alone they do not provide any information about lateral roots because there is nothing that defines which coordinates make up which root.
 - These points are plotted in order to clearly show the user which coordinates were automatically extracted from the image, which is necessary for the user to determine the best way to add lateral roots, as described more in Part 2.
- **Lateral root**: a root consists of a specific set of coordinates that define the path of the root.
- **Tentative lateral roots (green)**: This program will iteratively show you each lateral root and ask you to verify that it is okay. The automatic identification and tracing of roots can only use the extracted coordinates (in white), but you will have the option to add more points manually (i.e. by clicking with the mouse) if necessary.
- **Confirmed lateral roots (yellow)**: When a lateral root's coordinates have been confirmed by the user they are plotted in yellow. I.e. only lateral roots that are plotted in yellow are included in the analysis.

The program begins by displaying the jpeg image then plotting all the automatically extracted lateral root coordinates over it in white. Primary root coordinates are also plotted in yellow.

PART 1: Displays each auto-identified root in green for you to verify (or fix).

Next the program will show you the lateral roots that it has automatically identified for you to verify.

It will individually plot each auto-identified root in green (with the attachment point to the primary root in red).

After each root is displayed, the program will ask: "Hit enter if this root is okay as is, 0 if not okay at all, 1 to add more points to the end:"

- ENTER = okay as is: the program has traced a full lateral root correctly
- 0 = not okay at all: the root is not traced correctly (you can come back later to trace the root manually).
- 1= add more points to the end: the root was traced correctly except it is missing some points

If you chose:

- ENTER = okay as is: the program will now move on to plotting the next auto-identified root
- 0 = not okay at all: this “root” (what the program thought was a root) is deleted, and the program will move on to the next auto-identified root
- 1= add more points to the end: you will be prompted to click more points along the part of the root that was missed, starting from the point that the automatically extracted part left off, down to the root tip. You can add as many or few points as are necessary to trace out the path of the remaining root.

PART 2: adding any remaining lateral roots with user input.

If any root was not identified in the previous step (i.e. it is not plotted in yellow), you can now add it, using one of the methods described below...

- If **none or too little of the root in question is extracted** (i.e. plotted in white): The root can be traced out by the user by clicking points along the root. Make sure when you are plotting the first point, that it is right on top of the primary root traced out by the program in yellow.
- If the root was not identified but **its coordinates (or at least most of them) were extracted** (i.e. plotted in white): You will be prompted to click the attachment point of the root on the primary root, and then the root tip. Note that you do not need to worry about being terribly precise about selecting the tip - *the program will always chooses the extracted coordinate that is closest to the point you clicked* - so as long as there aren't other root coordinates very close by you can just roughly click somewhere just below the tip of the root in question.
 - Or, if the extracted coordinates don't run the full length of the root, instead of the tip you will select the point of the automatically extracted coordinates that is closest to the tip –you will have the option to add more points to the end as in Part 1.

See [SegLaterals demo.htm](#).

```
save([c_resfol '/individroots.mat'], 'attg', 'xyg')
```

SMOOTHLATERALS11

TO SMOOTH OUT THE LATERAL ROOT COORDINATES AND PUT THEM IN THE RIGHT ORDER

No input needed. You can visually verify the results.

Troubleshooting tip: If you see something you need to add or correct, go to the project folder and into the Mat-files folder, find the folder containing the mat files for the plant-plane that is problematic and delete the file made by the previous program, individroots.mat. You can now run SegLaterals12 again in Matlab and re-do this program for this specific image. When you are done run SmoothLaterals11 again.

```
save([c_resfol '/individrootsG.mat'], 'attg', 'xygg');
```

POTENTIALNODULES

TO EXTRACT POTENTIAL NODULE COORDINATES

No input needed.

```
save([... '/nodules_auto.mat'], 'nx', 'ny', 'nodmap');
```

LINKNODS2ROOTS7

TO LINK AUTO-IDENTIFIED NODULES TO SPECIFIC ROOTS

No input needed.

```
save([... '/nodAttchmts.mat'], 'nID', 'natt', 'nx', 'ny');
```

NODULESUI7

TO VERIFY AND EDIT NODULE COORDINATES

First you will use your cursor to encircle the nodulation zone (be sure to include all nodules here) so that the program can zoom in, making it easier to see individual nodules.

The program will now automatically move through each lateral (plus the primary) root, one at a time, highlighting the root in green and plotting any auto-identified nodules on that root with a pink circle.

- If any of the auto-IDed nodules are erroneous, you can remove them by right-clicking on them.
- If any nodules on the root were not auto-IDed, you can add them by left-clicking on them.

See [NodulesUI demo.htm](#).

```
save([... '/nodattchG.mat'], 'noddatt');
```

NODEXTRACTDATA16

TO GENERATE EXCEL FILES AND MAKE FIGURES.

No user input is required.

[Information on Excel files:](#)

Excel files generated will be stored in: ...\\Main project folder\\Excel files.

The main excel files, as described in Table 1 in Remmler et al. 2014, correspond to:

- Data at the whole plant level (1 row gives data for one plant): WholePlantData.xls.
- Data at the root level (1 row gives data for one root): RootData.xls.
- Data at the nodule level (1 row gives data for one nodule): NoduleCoordinates.xls.

Other excel files are generated too, that keep a record of the numerical values of each square grid for each map shown on Figure 5 in Remmler et al., 2014:

- Map of mean nodule density ie mean Number of nodules per cm²(=per grid square).xls (corresponding to Figure 5a in the paper, 5d or 5g in the paper).
- Map of standard error of nodule density.xls (corresponding to Figure S1a, S1d or S1g in the paper).
- Map of root density ie mean Root length in cm per cm²(=per grid square).xls (corresponding to Figure 5b in the paper, 5e or 5h in the paper).
- Map of standard error of root density.xls (corresponding to Figure S1b, S1e or S1h in the paper).
- Map of nodule-to-root ratio ie mean nb nodule per cm of Root.xls (corresponding to Figure 5c in the paper, 5f or 5i in the paper).
- Map of standard error of nodule-to-root ratio.xls (corresponding to Figure S1c, S1f or S1i in the paper).

Descriptions of the variable names used in the Excel files can be found in the folder C:/ ...Root & Nodule Quantification Software/Excel file variable descriptions.

Information on Figure files:

Figures will be stored in: ...\\Main project folder\\Figures.

Various figures are generated (refer to Remmler et al. 2014):

Figures of each RSE (one figure per plant) are saved as .fig and named using the following convention: Hull2D_RSEfigNN.fig where NN is a two digits code representing the plant ID number. Each of those figures corresponds to the example given in Figure 3a in the paper. However, on top of showing the hull of the roots within the nodulation zone for the plant in red, the hull of the roots within the nodulation zone for the dataset is also shown (in green).

The figure showing all extracted roots and nodules for the dataset (corresponding to Figure 3b in the paper) is saved as .eps. .png and .fig.

Average maps (corresponding to Figure 5 and Figure S1 in the paper) are saved as .eps. .png and fig.

NOTE on file formats of figures:

- .eps: vector graphics for making publication quality figures (use vector-based illustration software such as CorelDRAW for instance)
- .png: useful for quick viewing, MSWord, ppt slides
- .fig: for viewing in Matlab. Useful for modifying figure properties such as axis limits, colour coding or labels (refer to Matlab help for information on how to set figure properties).