PADRE: Package for Analyzing and Displaying Reticulate Evolution Manual Version 2.0

Martin Lott

April 8, 2009

1 Introduction

Overview PADRE is a Java software package allowing the computation and graphical representation of reticulate networks from input multi-labeled (MUL) trees. The main functionality of the program is two-fold. Firstly, it performs the consensus of a collection of multi-labeled trees M. Lott *et al* (2008). Secondly, from a multi-labeled tree a phylogenetic network can be constructed using the network construction approach in K. T. Huber *et al* (2006). The options presented to the user depend on whether a single MUL-tree or collection of MUL-trees are loaded into the program. In the case that a collection of MUL-trees is loaded then the user is first prompted to perform a consensus before running the network construction algorithm.

Installation The program PADRE is freely available from `http://www.uea.ac.uk/cmp/research/cmpbio/PADRE`. This page describes how to install the program and offers a number of test files to run as examples. The only requirements are that the machine on which the software is being run has Java virtual machine 1.5 or later installed and it is capable of displaying a graphical user interface. Any queries about the installation of the program should be directed to martin.lott@uea.ac.uk

Running the program For larger datasets (e.g. over 10 taxa) we recommend using the following command to start the program in the terminal window line:

```
java -jar padre.jar -XmxNNNm
```

`NNN` is the size of the memory to allocate in megabytes, 512 or 1024 is best if they are supported by your computer.

Disclaimer This software is supplied as-is, with no warranty of any kind expressed or implied. We have made every effort to avoid errors in design and execution of this software, but we will not be liable for its use or misuse. The user is solely responsible for the validity and consequences of any results generated.

2 Using the program

Main window When the program first loads you will be presented with a blank window, within which are a number of menus and options. The **File** menu contains the basic operations for loading and saving data and is similar in style and function to other programs with a graphical user interface.

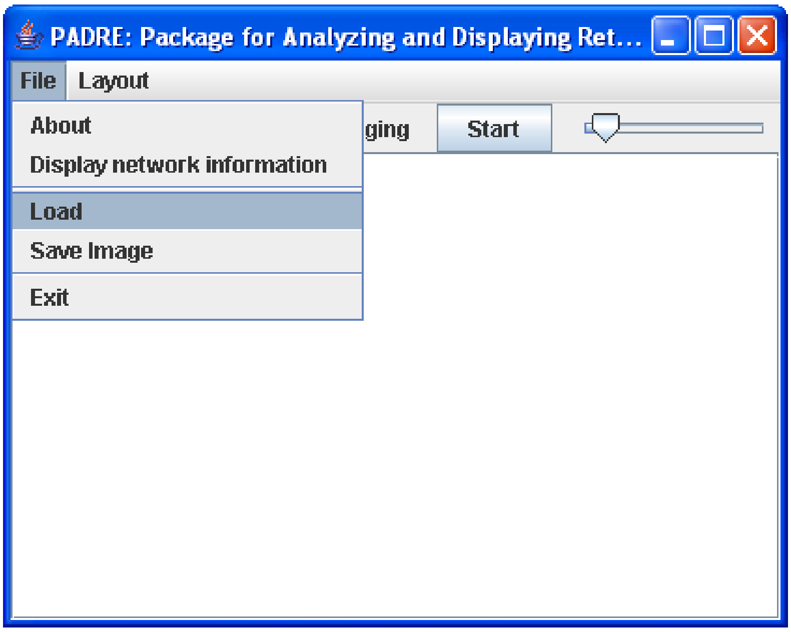

PADRE: Package for Analyzing and Displaying Ret...
File
Layout
About
Display network information
Load
Save Image
Exit
ging
Start

Figure 1: The main program window

The view of this window may vary slightly depending on the operating system of your computer and the resolution of its display.

Loading a MUL-tree The program takes as input a text file containing MUL-trees represented in NEWICK format. Information about the Newick format for representing trees in computer-readable form is provided at `http://evolution.genetics.washington.edu/phylip/newicktree.html`

If the input file contains more than one tree then the user is first prompted to perform a consensus, see section 5. A number of files are provided on our website to test both the consensus and network construction methods. The test files have been used in our publications and hence their corresponding results are also presented in those publications.

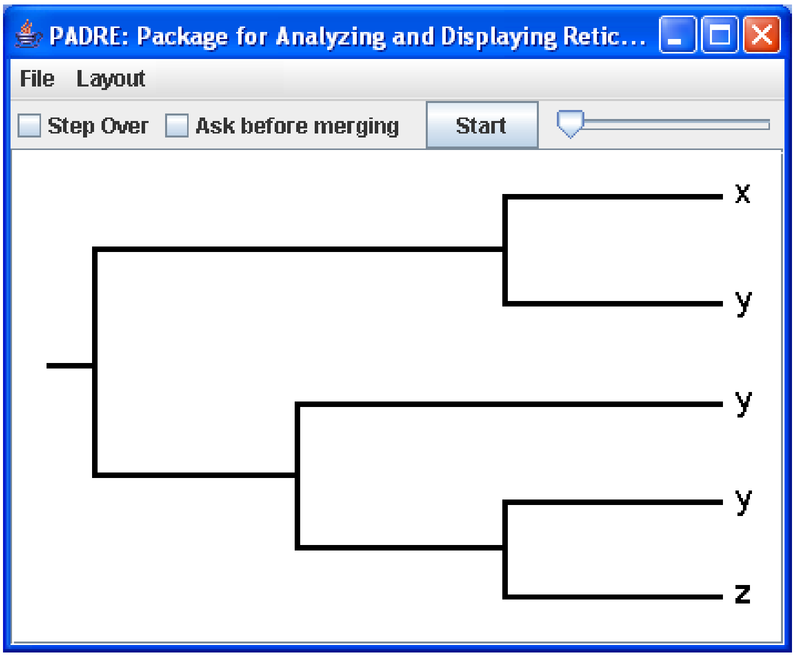


Figure 2: An example of a multi-labeled tree loaded into PADRE.

Exporting an image Any input MUL-tree, consensus MUL-tree or phylogenetic network displayed on screen can be exported into EPS format using the **Save image** command under the **File** menu. JPEG and PNG formats are provided as an alternative to EPS but we recommend using EPS for ease of inclusion into publications. The size and layout of the network saved to disk will be exactly that which is currently being displayed in the main window.

Please note: if images are to be used in scientific papers we ask that you cite our paper, K. T. Huber *et al* (2006).

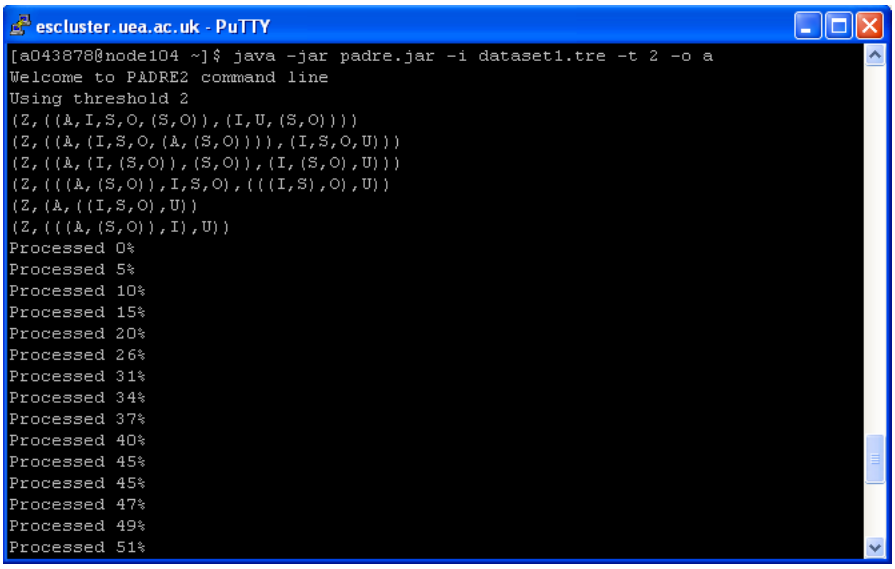
3 Command Line Interface

A command line version of our software is available that may be used to process difficult datasets on a cluster, or as part of a pipeline. The software takes as input any newick file which contains a collection of MUL-trees and outputs a newick file with the resulting consensus MUL-tree(s). The output file has `-result` appended to the name, for example `dataset1-result.tre`

After downloading the software from our website, simply type `java -jar padre-cli.jar`

A list of options will be presented for performing the method as follows.

- -i The name of the input NEWICK text file for the consensus method. For example, in Windows `-i ‘‘C:\Program Files\Padre\dataset1.tre’’`, alternatively in Linux/Unix `-i /Padre/dataset1.tre`
- -t Threshold on the number of trees each cluster must be exhibited by. For example, `-t 2` would require that each cluster is exhibited by at least two input trees.
- -o Output type (Optional). By default one of the best trees is arbitrarily selected for output. -o a outputs all possible consensus trees. -o b outputs all best consensus trees i.e. those with minimal hybrid number. -o s outputs the collection of (possibly unresolved) MUL-trees found.
- -a Additional options (Optional). -a a Considers all clusters removed by the threshold.



```
[a043878@node104 ~]$ java -jar padre.jar -i dataset1.tre -t 2 -o a
Welcome to PADRE2 command line
Using threshold 2
(Z,((A,I,S,O,(S,O)),(I,U,(S,O))))
(Z,((A,(I,S,O,(A,(S,O)))),(I,S,O,U)))
(Z,((A,(I,(S,O)),(S,O)),(I,(S,O),U)))
(Z,(((A,(S,O)),I,S,O),(((I,S),O),U))
(Z,(A,((I,S,O),U))
(Z,(((A,(S,O)),I),U))
Processed 0%
Processed 5%
Processed 10%
Processed 15%
Processed 20%
Processed 26%
Processed 31%
Processed 34%
Processed 37%
Processed 40%
Processed 45%
Processed 45%
Processed 47%
Processed 49%
Processed 51%
```

Figure 3: An example of using the command line interface in PuTTY.

4 Graphical User Interface

The main window displays either the input MUL-trees, consensus MUL-trees or phylogenetic networks depending on the stage of the algorithm execution. Furthermore, the network construction algorithm can be run in steps and the results are displayed at each step. Note that some options need to be selected in order to view the network at each step. Initially the MUL-tree is scaled to the screen width and height, however the slider at the top right allows a range of sizes to be selected.

Tree and network structures All nodes and edges can be moved by dragging them with the mouse to the required point on the screen. In addition, the entire subtree rooted at a node being dragged will also be repositioned by default. Note that any node which has a path leading to it from the node being dragged is considered to be part of its subtree. Therefore, as the algorithm executes, nodes may become part of more than one subtree as their ancestors have been merged. To be able to move nodes in isolation from their descendants an option called **Don't move subtrees** can be selected under **Layout**. Subtrees can be moved at any point during and after the network algorithm executes.

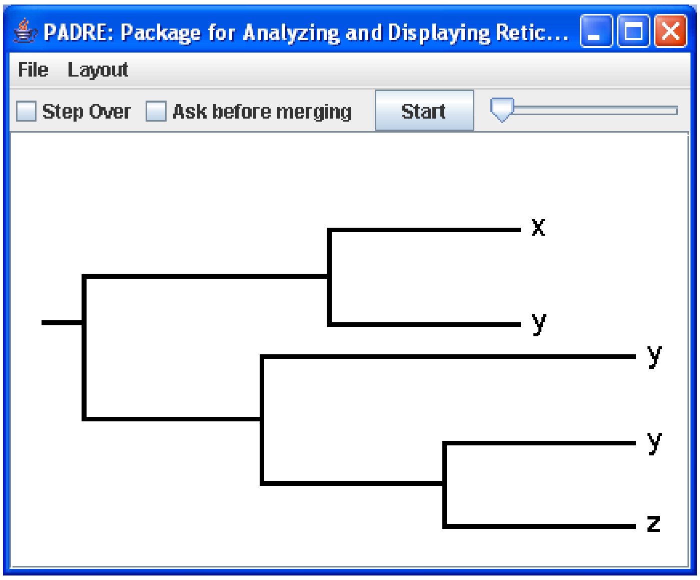


Figure 4: The result of performing a drag and drop on one of the vertices from the tree presented in Figure 3

Flagging nodes and edges Any tree node or edge may be 'flagged' by the user simply by clicking it once, a flagged node is different colored and this option can be useful for highlighting part of the tree considered interesting for biological reasons. By default, vertices are not painted however, selecting **Paint vertices** under the **Layout** menu will draw them with a circle. The algorithm also uses this flagging feature to show subtrees under consideration for merging when the **Ask before merging** option is selected for the network construction algorithm.

Note: Edges are colored in bright green by the software in the case that two or more completely overlap i.e. a species is thought to have separated into two subspecies which then recombined into a new species.

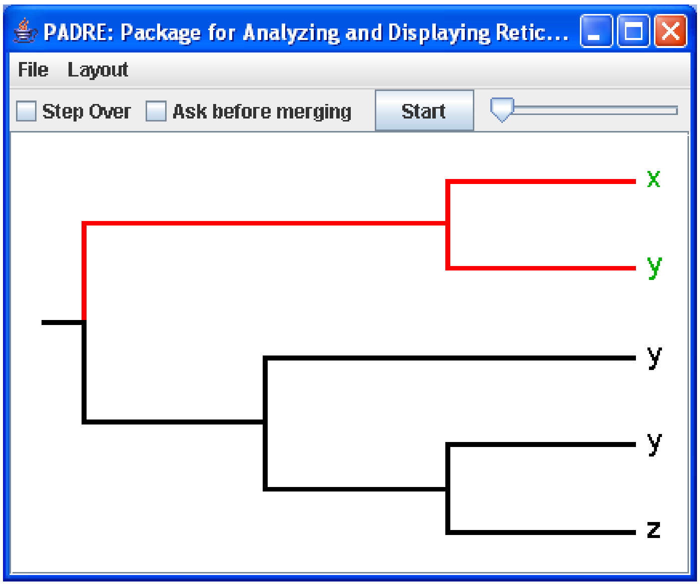


Figure 5: A tree with some flagged vertices and edges

Slanted and rectangular edges Mathematicians and computer scientists prefer to use *slanted* edges as depicted in Figure 7 below whereby a line is drawn directly connecting a given pair of vertices. Biologists however, prefer to use *rectangular* edges as in Figures 1 to 6 whereby the edge follows a right angle which is first vertical and then horizontal. Given that this program is intended primarily for use by Biologists, edges are drawn using rectangular edges by default. The **Layout** menu contains the options to switch to entirely slanted edges by selecting the **Use rectangular edges** option.

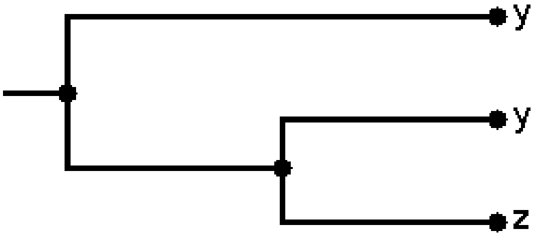


Figure 6: A tree represented with rectangular edges

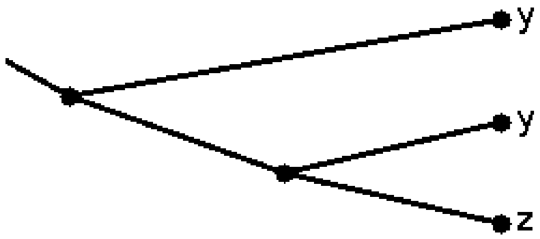


Figure 7: The same tree presented in Figure 6 with slanted edges

Scaling the tree or network Although the network is scaled by default to fit exactly onto the user screen, it may be the case that for larger networks this is simply too small. For this reason, an option is provided to scale the network both larger and smaller at any point before, during and after the execution of our algorithm.

Scaling is achieved by using the slider at the top right of the program window. By default, scaling takes place on the X axis only, however with the **Maintain aspect ratio** option selected scaling will simultaneously take place along both the X and Y axis. If the structure is too big to fit on the screen appropriate horizontal and vertical sliders may be used to focus on part of the tree/network.

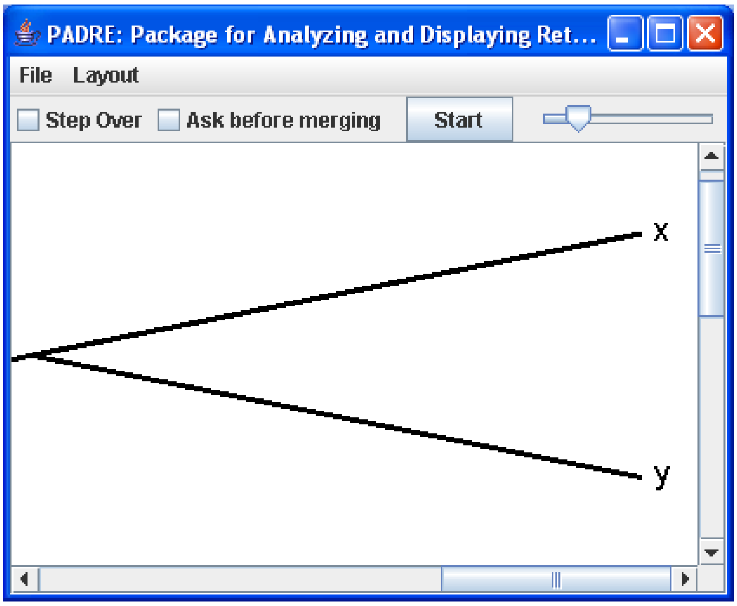
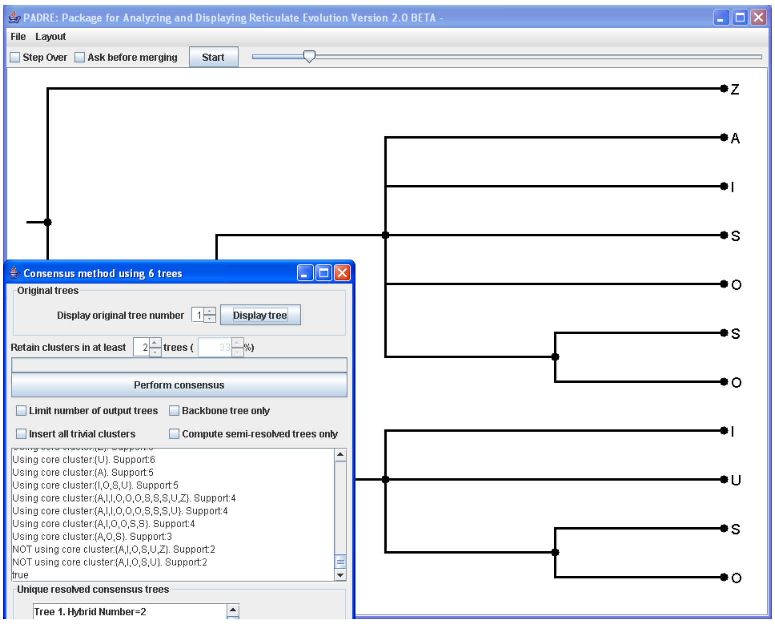


Figure 8: Zoomed in on a particular part of the tree by making it larger and moving the sliders

5 Performing a consensus from a collection of MUL-trees

Overview If the input file contains several MUL-trees the user is first presented with a window that contains all of the options for the consensus method (see Figure 9). The consensus problem itself is computationally intractable K.T. Huber *et al* (2008) and so the time taken may increase considerably for larger datasets (e.g. more than 20 taxa). Since exact solutions take too long to compute, we have developed and implemented an approximation algorithm which aims to find a reasonable solution in a relatively short amount of time M. Lott *et al* (2008).

Running the method The options for the consensus method include those to display any one of the input trees or choose the threshold for the consensus method. The user can also select one of the consensus trees which are ranked according to their 'hybrid number', that is, the number of hybridisation events they hypothesize.


PADRE: Package for Analyzing and Displaying Reticulate Evolution Version 2.0 BETA -
File
Layout
□ Step Over
□ Ask before merging
Start
Z
A
I
S
O
S
O
I
U
S
O
Consensus method using 6 trees
Original trees
Display tree
Perform consensus
Unique resolved consensus trees

Display original tree number 1

Retain clusters in at least 2 trees (33 %)

□ Limit number of output trees
□ Backbone tree only
□ Insert all trivial clusters
□ Compute semi-resolved trees only

```
Using core cluster:{U}. Support:6
Using core cluster:{A}. Support:5
Using core cluster:{I,O,S,U}. Support:5
Using core cluster:{A,I,I,O,O,O,S,S,S,U,Z}. Support:4
Using core cluster:{A,I,I,O,O,O,S,S,S,U}. Support:4
Using core cluster:{A,I,O,O,S,S}. Support:4
Using core cluster:{A,O,S}. Support:3
NOT using core cluster:{A,I,O,S,U,Z}. Support:2
NOT using core cluster:{A,I,O,S,U}. Support:2
true
```

Tree 1. Hybrid Number=2

Figure 9: One of the input trees to the consensus method, the consensus options window is shown in the bottom left corner.

We now briefly describe the options for the consensus method.

Limit the number of output trees Since the number of consensus trees could be very large, this option provides an arbitrary cut-off when 3000 non-unique are found by the method. The intention is to limit space and time requirements due to the computationally intractable nature of the problem.

Backbone tree only Whilst there may be many different consensus trees they all share a common backbone, this option stops the method once the backbone tree has been constructed.

Compute semi-resolved trees only There may not be enough support to construct a bifurcating consensus tree, if this the case the method checks all possible resolutions. This option causes the method to stop before such resolutions are checked and hence the user is presented with a list of semi-resolved trees rather than a list of resolved trees. All (semi-resolved) trees are scored according to the number of hybridisation events they hypothesis regardless of whether this option is used or not.

6 Running the network construction algorithm

Overview In contrast to the consensus method, the network construction problem is more tractable and our algorithm is expected to use minimal computing power hence it will display results within a few minutes for 100 or less taxa. A formal description of the algorithm is given in K. T. Huber *et al* (2006). We now briefly describe the options for the network construction method.

Ask before merging There are a couple of simple options available for concerning the way in which the algorithm is run. The first of these **Ask before merging** allows the user to receive a prompt so he/she may choose whether to accept that two maximal inextendible subtrees should be merged. Once all maximal inextendible subtrees $T_2 \dots T_n$ equivalent to T_1 have been accepted or declined for the process of subdivision, identification and pruning takes place.

Step over This option allows the user to run the algorithm step by step, when selected the algorithm pauses after T_1 is merged with $T_2 \dots T_n$ until the **Resume** button is clicked. Once the resume button is clicked the next collection of maximal inextendible subtrees are merged and again the program pauses until the button is clicked. The order in which the subtrees are merged can clearly be seen. Whilst the algorithm is paused, you have full control to rearrange the graphical representation of the network and to export images as you do at any other time.

7 Editing the results

Resolving interaction vertices An interaction vertex (any vertex with in-degree greater than two) can be resolved by flagging two or more of the incoming edges, these can then be 'pushed up' by selecting **Push up edges from interaction vertex**. In order to perform this operation, at least three edges which share the same head vertex need to be selected.

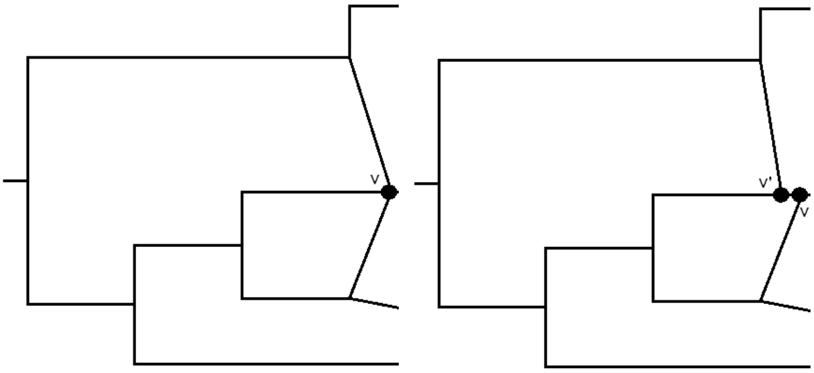


Figure 10: (Left) Before resolving the edges incoming into a vertex v. (Right) After performing this operation, a new vertex v' is created.

8 Legal

Whilst this implementation of the algorithm presented in K. T. Huber *et al* (2006) is complete we still continue to actively work on PADRE. PADRE will always remain free of charge however, during this development process the source code available on the website may not be the most recent version.

References

K. T. Huber and V. Moulton (2006). Phylogenetic networks from multi-labelled trees, Journal of Mathematical Biology 52(5), 2006, 613-632.

K. T. Huber, B. Oxelman, M. Lott and V. Moulton (2006). Reconstructing the evolutionary history of polyploids from multi-labeled trees, Molecular Biology and Evolution, 23(9), 2006, 1784-1791.

K. T. Huber, M. Lott, V. Moulton and A. Spillner (2008). The complexity of deriving multi-labeled trees from bipartitions, Journal of Computational Biology, 15(6), 2008, 639-651.

M. Lott, A. Spillner, K. T. Huber, A. Petri, B. Oxelman and V. Moulton (2008). Inferring polyploid phylogenies from multiply-labeled gene trees, *submitted.*