SPRINT: Package for Calculating and Visualising the Minimal Hybrid Number of a Phylogenetic Network Manual

Liam J Maher[1], Taoyang Wu[1], and Katharina T Huber[1]

[1]University of East Anglia, School of Computing Sciences

22 September, 2022

1 Introduction

Overview Polyploidy is an important evolutionary driver which can be found in organisms ranging from plants to fish and fungi. Existing algorithms for reconstructing the evolutionary history of polyploid species can be found in Huber et. al. (2006) and Lott et. al. (2009), however, these methods rely on the existence of so called multi-labelled trees which may not be readily available. There is currently no tool available which aims to compute the minimum number of hybrid vertices (vertices of indegree at least two) possible to explain a given set of ploidy levels based on the ploidy levels alone, where hybrid vertices represent polyploidization events.

Here, we present the *Species Ploidy Realisation of Integers with Networks Tool* (SPRINT) software package that has been developed to circumvent the reliance on multi-labelled trees when obtaining an evolutionary picture from a polyploid dataset.

Installation The program SPRINT is freely available from the UEA Comp. Bio. webpage and also on GitHub. On the webpage and on GitHub is a text file (install_instructions.txt) which describes how to install the program and the page offers a number of test files to run as examples. The machine on which the software is being run must have Python virtual machine 3.9 or later installed and is capable of displaying a graphical user interface. Also, provided with the software is a file containing the other requirements (matplotlib, networkx, Pillow, PySimpleGUI) which can be installed using the command: **pip install -r requirements.txt**. The authors recommend using pygraphviz to visualise the networks generated by SPRINT and to do so will require the installation of Homebrew (see the install_instructions.txt file included in the package for a step by step guide on how to do so from the command line interface). If this is a problem for the user, we also provide another file (SPRINT-nxspring.py) in the same SPRINT package which instead visualises the networks with the spring layout from networkx. Any queries about the installation of the program should be directed to liamjmaher96@gmail.com.

Running the program See the text file install_instructions.txt.

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

Data entry window When the program first loads you will be presented with a data entry window within which are a number of options to enter data and how users would like to visualise data, Figure 1.

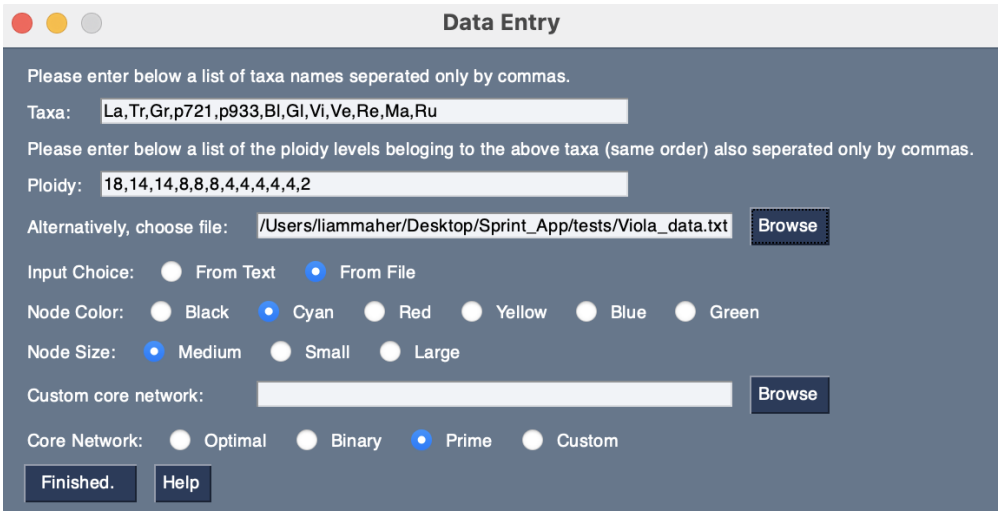

Figure 1: Data entry window.

The view of this window may vary slightly depending on the operating system and display resolution of an individual user.

Text entry The first option to enter data into the program is composed of two input text boxes. One is for the species taxa and the other for those species' ploidy levels. For taxa, users input alphanumeric strings, one for each taxa of interest, separated by a comma (no spaces). For ploidy levels, users input integers greater than 0 for each taxa entered above, in the same order as entered above and separated by a comma (no spaces).

File browser The second option to enter data into the program is by searching on your system for a .txt file via the 'Browse' button found within the data entry window. The .txt file should be of the form (see Figure 2):

- **First line** For taxa, users input alphanumeric strings, one for each taxa of interest, separated by a comma (no spaces).
- **Second line** For ploidy levels, users input integers greater than 0 for each taxa entered above, in the same order as entered above and separated by a comma (no spaces).

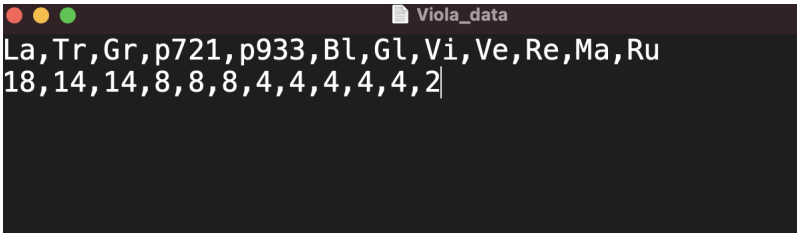

Figure 2: File browser input example for the Viola dataset used for the output in Figure 5.

Input choice Once the user has either input data into the data entry text boxes or alternatively the user has browsed for and selected a .txt file containing the data and therefore the file path has been displayed, the user then selects which method they have chosen from the radio buttons labelled 'Input Choice:' by selecting 'From Text' or 'From File', respectively.

Note: selecting the 'From File' radio button before the file path has been displayed will cause the program to fail and result in an error message.

Node color The radio buttons beside the label 'Node Color:' gives the user six different color options. When the program produces the phylogenetic network visualisation of the input data, the choice of color made by the user will be reflected in the color of the nodes/vertices in the output figure. The default option is 'Black' but any color can be chosen with no effect on the run time of the program.

Node size The radio buttons beside the label 'Node size:' gives the user three different size options. When the program produces the phylogenetic network visualisation of the input data, the choice of size made by the user will be reflected in the size of the nodes/vertices in the output figure. The default option is 'medium' but any size can be chosen with no effect on the run time of the program.

Core network The radio buttons beside the label 'Core Network:' gives the user different options from which to initialize the programs algorithm. Radio buttons 'Binary' and 'Prime' refers to the core network construction methods Binary representation and Prime factor decomposition respectively, see (Huber and Maher, 2022) for more information on core network construction methods. The default option, 'Optimal', will compare the number of hybrid vertices between two different phylogenetic networks constructed with the core network construction methods listed above. The program will then choose the network with the fewest hybrids (prime factor is used in the

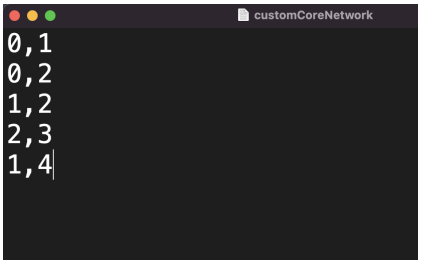

Figure 3: Format of a custom core network for the ploidy profile $(2, 1)$ entered into a .txt file.

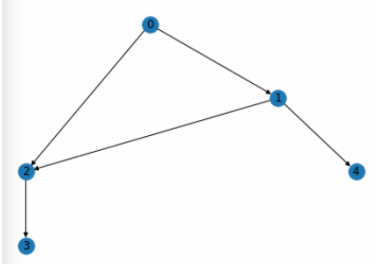

Figure 4: Example of a custom core network drawn in networkx for the ploidy profile $(2, 1)$.

case of a tie) to initialize the remainder of the construction.

Custom core network Should the user prefer to enter a custom core network instead of using one of the above methods of construction built into SPRINT, it is possible to do so. First create a .txt file which contains a pair of integers on each line separated by a comma. Each integer represents the vertex label and each line will represent an edge in the users choice of network. See Figure 3 for an example file for the ploidy profile $(2, 1)$.

Output Once the data has been entered and the user has selected their preferences in the Data entry window, click the 'Finished' button to proceed. For example, if the .txt file found in the documents of this program was used then the output would look similar to Figure 5.

The title of the output GUI window displays the ploidy profile used as input. The top of the window contains a figure which displays a phylogenetic network. This network realises the input ploidy profile and contains the fewest possible number of hybrid vertices given the input ploidy profile.

Shown below the figure is the simplification sequence, starting from the input ploidy profile at the top and continuing until the final element is a simple ploidy profile, from which the core network is constructed and the traceback begins.

Below the simplification sequence, the minimal number of hybrid vertices for the input ploidy profile is displayed. In Figure 5, the minimal hybrid number is 6. Also displayed is confirmation whether the network generated by SPRINT is optimal (provided the initialisation is optimal) or not. This message will instead provide an error in the case that operation a) (from the application note) is used twice in a row during the traceback of the simplification sequence. Advice on how to obtain the optimal network is provided within this message.

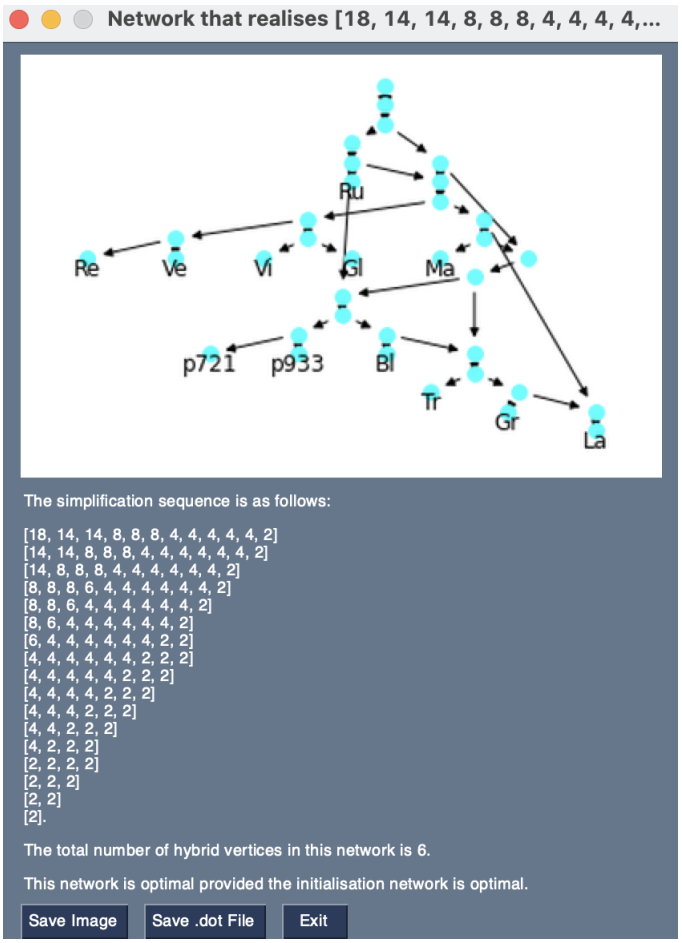

Figure 5: Output window.

3 Legal

Whilst this implementation of the algorithm presented in K. T. Huber and L. J. Maher (2022) is complete we still continue to actively work on SPRINT. SPRINT 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.

4 References

[1] Huber, K. T. and Oxelman, B. and Lott, M. and Moulton, V. *Reconstructing the evolutionary history of polyploids from multilabeled trees* Mol. Biol. Evol. 23(9) (2006) 1784–91.

[2] Lott, M. and Spillner, A. and Huber, K. T. and Petri A. and Oxelman B. and Moulton V. *Inferring polyploid phylogenies from multiply-labeled gene trees*, BMC Evol. Biol. 9 (2009) 216.

[3] Huber, K. T. and Moulton, V. *Phylogenetic networks from multi-labelled trees*, J. Math. Biol. 52(5) (2006) 613-32.

[4] Huber, K. T. and Maher, L. J. *The Hybrid Number of a Ploidy Profile*, J. Math. Biol. 85 (2022) article number: 30.