

# HapStar Version 0.5 (C) FoAM vzw 2010

GPLv3 Licence

Support contact: dave@fo.am

## Overview

HapStar takes the network connection output data generated from Arlequin (or a simple user-generated input file), and uses a force-directed algorithm to automatically lay out the network for easy visualisation. In addition, this program is able to use the alternative connections generated by Arlequin to create a minimum spanning tree. Publication-ready figures can be exported simply, or the exported vector-format files can be further manipulated in graphics software.

## Installation

HapStar is written in Python, and so before beginning, if you are using Linux or Windows, the latest version of Python must be installed onto the computer that will be used. Python is free to download from <http://python.org/download/>. If you are using a Mac, Python should be already installed as part of OSX.

After the installation of Python, HapStar can be downloaded for free from <http://www.fo.am/hapstar> - separate versions are available for Windows and for Linux/Mac OSX. A zip or tar.gz file will be downloaded to the location chosen.

### To run HapStar:

*Windows:* Open the downloaded file and double click on “hapstar.py” icon. A terminal window will open and then the HapStar program window will open on top of this – the terminal window should be left open but can be ignored.

*Mac:* Open the downloaded file, hold ctrl and click on the “hapstar.py” icon, a menu should appear, go to "Open With" and click on "Python Launcher". Unfortunately the window will pop up underneath some smaller terminal windows, move them out of the way to see the program.

*Linux:* Start the program by double clicking on the “hapstar.py” icon, or running “python hapstar.py” in the terminal.

### Zip file contents:

README: This file.

LICENCE: The GPLv3 licence.

hapstar.py: Executable file.

examples/example.txt: A simple example.

examples/big.txt: A synthetically generated test.

examples/complex.rtf: A complex example, run with speed set to less than 10.

examples/loop.txt: A simple example of a looped network.

## Input file

Example input files are available within the download. The input file is simply a list of haplotypes and the connection lengths between them. Arlequin (Excoffier et al. 2005) will output this format directly, and this can be copied and pasted from the Arlequin results file into a text file. The input should be in the following form, where the first two columns are the names of the haplotypes, and the third column is the connection distance between them:

Haplotype1 Haplotype2 3  
Haplotype2 Haplotype3 1  
Haplotype 1 Haplotype 4 2  
etc...

The name of the haplotype can be text or numeric, and will appear in the network nodes as it is written in the input file. The input file should be saved as a text file (ascii or rich text format), and the values can be space or tab separated.

## Running the program

Click on the 'Load' button in the HapStar window to locate and load your input file. Once this is done, the starting network should be visible within the window. Click on 'Run' to start the network layout optimization. You should see the connection lines start to move around and the shape changing. The same 'Run' button can be clicked to stop the process, at this point the haplotype nodes and labels will become visible. Once stopped, the user can manually drag haplotype nodes (using left click and drag on the mouse) to a preferred position, and the optimization can be restarted from this position by clicking the 'Run' button again. This is sometimes necessary, for example if long branches cross each-other. The program will take longer for larger networks, but should not take longer than a few minutes at most to complete the layout. The program does not automatically stop when it has reached an optimal layout – it is up to the user to stop the process when they are happy with the shape. The 'Restart' button can also be used to reset the network to the starting shape.

## Viewing options

The window can be resized to view larger networks, and in addition it is possible to zoom in and out by clicking on the + and - buttons. You can also hold the right mouse button down and drag the mouse up and down. The network can be moved around the viewing window by left clicking on the network, and dragging it around using the mouse.

The radius of the node circles ('Circle Radius' field) and the text size ('Text Size' field) can be individually controlled for the whole network to make it easier to view. Type the new values into the text fields and press enter for them to take effect.

## Exporting graphics

HapStar produces networks that conform to standard haplotype network formatting, with branch lengths that are representative of connection distance, and small filled black nodes representing missing haplotypes. Once you are happy with the network, it can be exported as a Scalable Vector Graphic (.svg) by clicking the 'Save SVG' button, or as a textfile representation of the graph (.dot) by clicking the 'Save DOT' button. For further editing, such as the addition of colour or labels, we recommend exporting a .svg file and editing in Inkscape (free graphics software, <http://www.inkscape.org/>). Additionally, within Inkscape, the .svg file can be exported as other file formats, and the resolution set according to publication guidelines. Alternatively, a .dot file can be used in software such as Graphviz (free graphing software, <http://www.graphviz.org/>) if you want to edit the overall format of the network.

## Other options

*Altering the speed for larger networks:*

The layout algorithm works by repelling the nodes from each-other until they are optimally separated. Some larger networks can suffer from oscillation of the nodes when running, this can be stopped by using the 'Speed' slider to reduce the amount of force the algorithm uses (in other

words, the nodes repel each-other with less force). The default value will be suitable for the majority of networks. This is the only parameter that affects the layout optimization directly – in general, lower speeds are better for large and complex networks with many nodes, whilst higher speeds achieve a faster optimization for small or straightforward networks. The speed can be adjusted while the optimization is running, and the effect should be visible.

#### *Minimum Spanning Tree:*

A tick-box option is available to switch between a standard network and a minimum spanning tree ('MST') of the network. This option can be useful for simplifying complex networks. If alternative connections are provided in the input file, then this option will create a network that connects all haplotypes in a single graph without cycles. Minimum spanning trees are calculated using Prim's Algorithm (Prim 1957). If two connection routes are of equal length, this algorithm will choose arbitrarily but consistently which route to keep. Caution should be applied when using this option, as this will only produce one possible minimum spanning tree, rather than all potential trees.

### **Citing HapStar**

In publications, please cite the use of HapStar as follows:

\*\*\*citation to be inserted on publication\*\*\*

### **References**

Excoffier, L. G. Laval, and S. Schneider (2005) Arlequin ver. 3.0: An integrated software package for population genetics data analysis. *Evolutionary Bioinformatics Online* 1:47-50.

Prim, RC 1957, 'Shortest connection networks and some generalizations', *Bell System Technical Journal*, Vol. 36, pp 1389-1401.