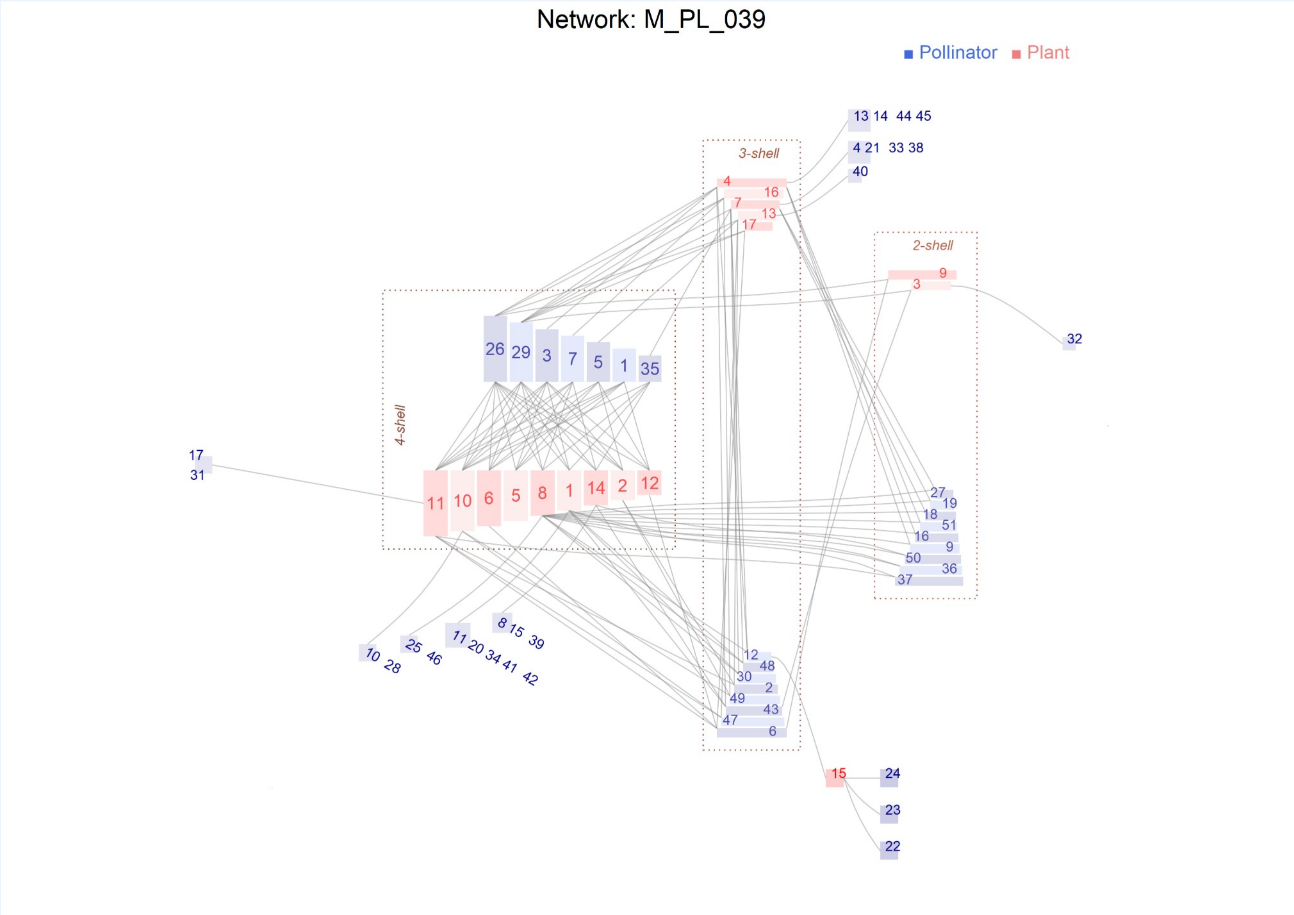
BipartGraph



User Guide

R2.0 January 2025

BipartGaph User guide R2.0

Author: Javier García-Algarra

January 2025 https://github.com/jgalgarra/bipartgraph

Contents

1 Introduction..4

2 Installation...5

2.1 Requirements..5

2.2 First step: download the software..5

2.3 Second step: install R..6

2.4 Third step: setting up the application...8

2.5 Running *BipartGraph*...10

2.6 Installation on Linux..10

3 Data management...12

4 The ziggurat graph..14

4.1 Explore the interactive ziggurat..17

4.2 Modifying the look and feel of the ziggurat plot..18

4.3 Printable ziggurat..24

5 The polar graph...27

Useful hacks...29

References..29

1 Introduction

BipartGraph is an interactive application to visualize bipartite ecological networks, using the *kcore* decomposition method [1,2].

The most common visualization of this kind of networks is the bipartite graph. It underlines the existence of two communities, but links among species are difficult to distinguish even with a reduced number of nodes. When the size grows, the graph becomes a *hairball*, an expression common in visualization for messed plots (fig. 1).

It is an exercise of graphical intuition to identify the central core and almost impossible to guess the centrality of species. As a result, it is uncommon finding graphs of networks with more than ten species per guild in the research literature, although typical mutualistic networks are bigger. This problem is not due to a lack of skills of the user, it is a limitation of the graph itself.

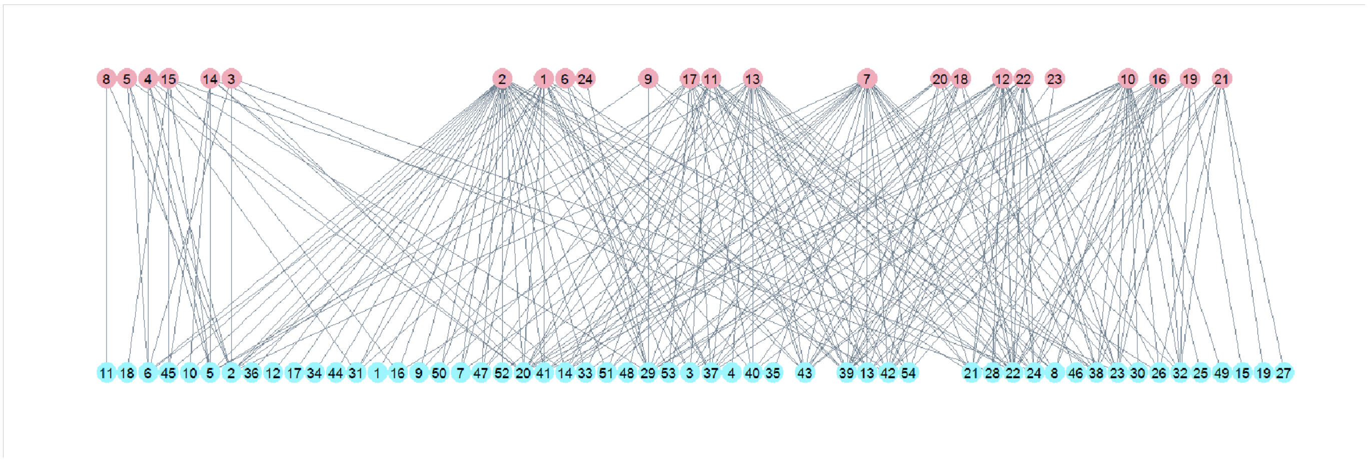


Figure 1: Bipartite graph of a mutualistic community with 78 species [3].

The *k-core* decomposition offers a natural way to group species with similar connectivity and so it enables a more powerful spatial distribution.

You do not need to be an expert in analysis of graphs to enjoy this program, just install it and start to play!

2 Installation

BipartGraph is an interactive application developed in `R` language. If you are familiar with `R`, congratulations, this guide will seem pretty easy to you. If you have not worked with `R` or with any other programming language at all, this document guides you step by step through the installation procedure.

2.1 Requirements

There are only two requirements to install *BipartGraph*, a working `R` environment and any web browser (*Firefox*, *Chrome* or *Safari* if possible). We recommend, at least, 8 GB of RAM in your machine. `R` is an open source project, you will find the download and installation instructions for different operating systems at https://cran.r-project.org/.

Please, read carefully how to run *BipartGraph* in three steps.

2.2 First step: download the software

`Bipartgraph` is available as open source software under `MIT` license in `github`. Open your browser and go to the following URL: https://github.com/jgalgarra/bipartgraph

Notice the green button **Clone or download** that provides two ways to get the software (Fig. 2).

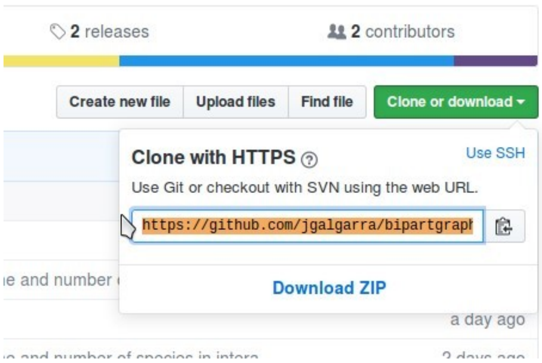


Figure 2: *BipartGraph* repository at github.com

First one is using `git`, the control version tool (https://git-scm.com/). If you know how `git` works just clone the `jgalgarra/bipartgraph` repository.

```
> git clone https://github.com/jgalgarra/bipartgraph
```

A new directory called `bipartgraph` appears. This is the recommended method, because at any time you may update your local software with the `git pull` command.

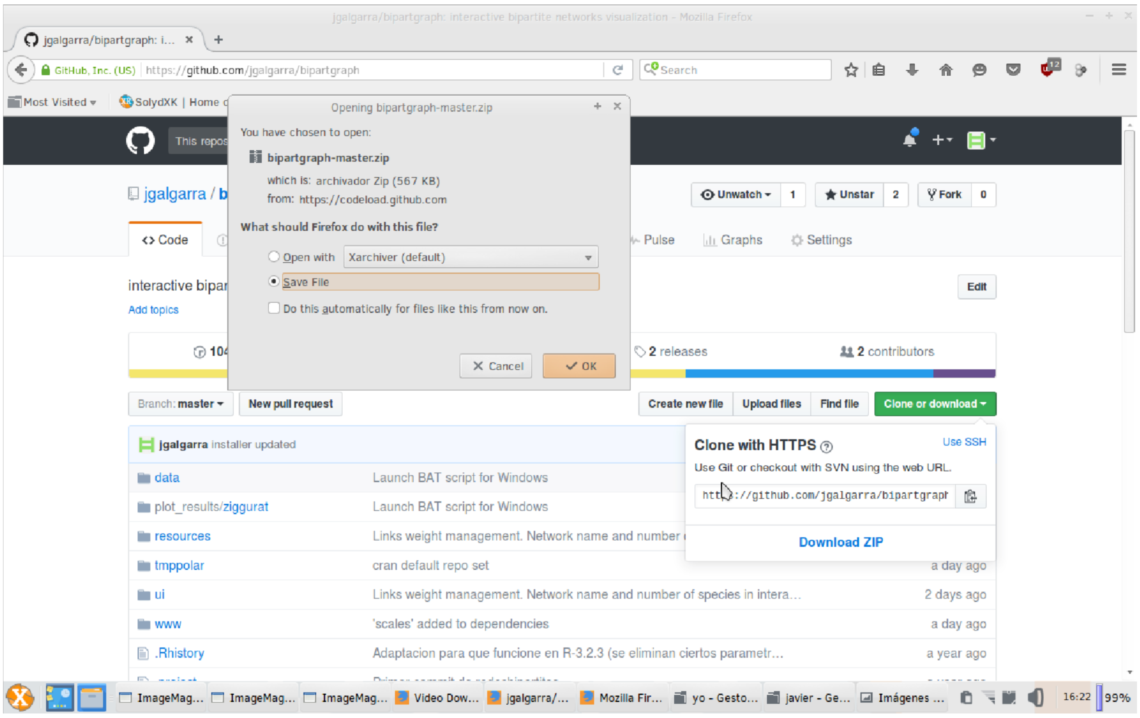


Figure 3: Downloading the *BipartGraph* ZIP file.

OK, but what happens if you are not a software developer and do not want to install `git`? Download the `ZIP` file (fig. 3), uncompress it, and you will have a new directory called `bipartgraph-master` with the same contents than the original repository. The only difference is the directory name.

2.3 Second step: install R

You need the release `R 4.3.0` or later. If you have never installed `R`, pick the last available release from the download page, it will be fine for this purpose.

You need a `gcc` compiler in your machine. If you are a developer and you do have it, `kcorebip` will install smoothly. Otherwise, you will get a message telling that `Rtools` is not available. In that case follow the instructions of the following page to install Rtools:

`https://cran.r-project.org/bin/windows/Rtools/`

A) `Windows`

Figure 4 shows the installer program. *Your attention please!* The installation program will suggest you a path. Do not change it unless you are sure of what you are doing. For instance, in `Windows` sytems, the volume `C:` uses to be restricted to users with `administrator` privileges.

Let the `R` installer to choose the best location for your user. Please, write the name of the installation folder somewhere (the notepad is a good choice), because you need it for the nex step.

Once you install `R`, check that it is ready to run in the default path of your environment. What does this mean? If you type `R` in the command line the application starts. If you are a `Windows` user it is rather probable that you get this disappointing message: `R is not a recognized command as an internal or external command`.

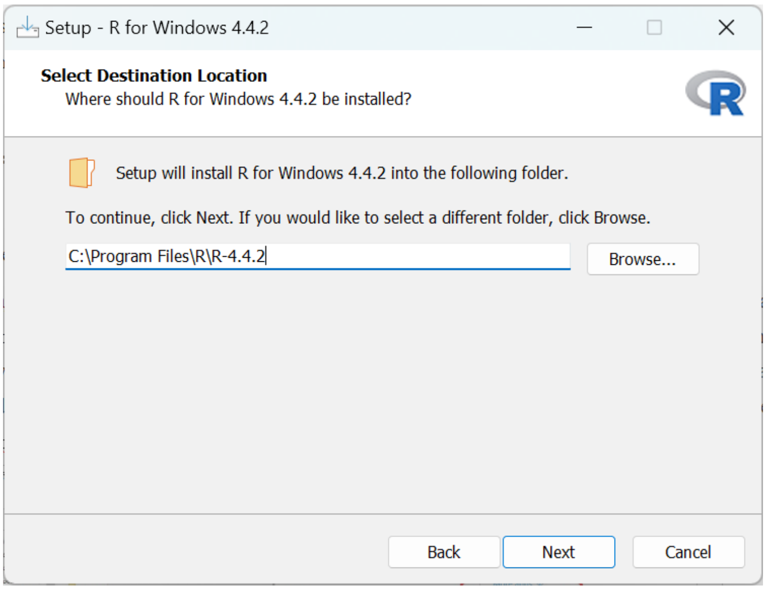
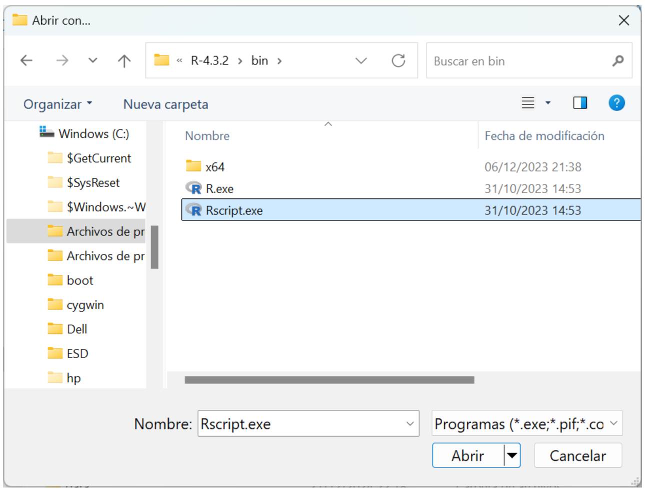


Figure 4: Installer program for Windows. The name of the installation folder is important to set the PATH variable in Windows.

There are two methods to overcome this problem in `Windows`:

1. Simple and safe. Go to the directory where you installed `bipartgraph`. Select the file called `global`, right click and select "Open with" and then "Choose another app" (the option name may be slight different, this screen was captured in `Windows 11`). Navigate to the installation folder that you wrote in your notepad. Inside this folder click on the `bin` folder and then select `Rscript`.



After you perform this action the `R` icon is displayed by the name of the scripts.

2. Include the `R bin` folder in the path variable. In the example of figure 4, it is `C:\Program Files\R\R-4.3.2\bin`. You have to add \bin to the name of the installation folder you copied in the notepad. To modify the value of the variable `Path`

   in `Windows` we suggest you this useful page, the procedure depends on the release http://www.dowdandassociates.com/blog/content/howto-set-an-environmentvariable-in-windows-gui/ .

B) UX like systems (including `Linux` an `MacOS`)

Your attention, please. For Debian-based distributions there is a detailed set of instructions if you find problems to install `R 4.2` or later. Check section 2.6.

Figure 5 is a screen capture of your terminal after you invoke `R` if you work in `Linux`. The appearance in `Windows` is quite the same.

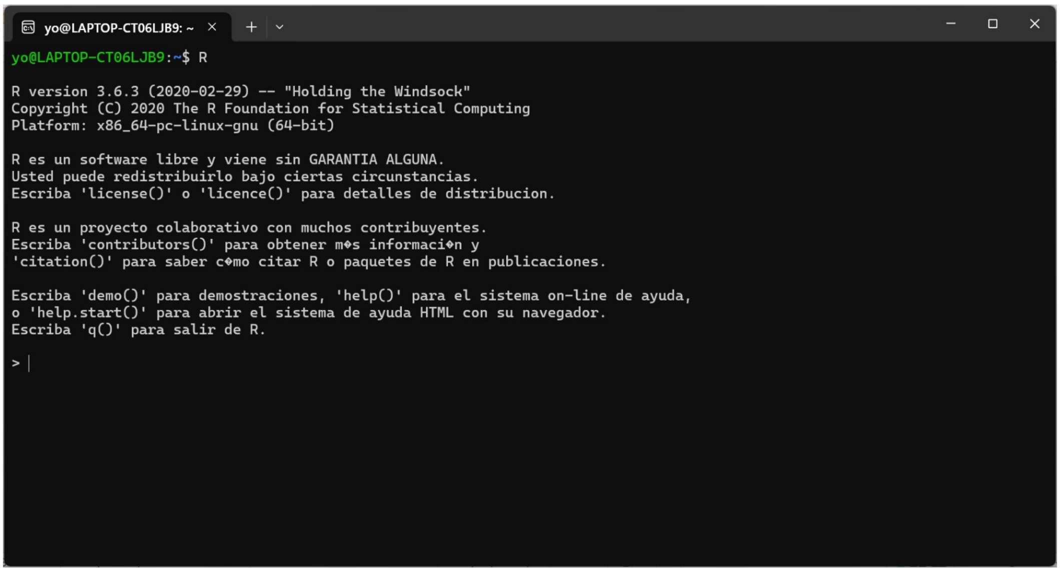


Figure 5: Launching R from the command shell. The release number (3.3.0 in this example) appears in the first row of information.

2.4 Third step: setting up the application

Once you have downloaded and uncompressed the `ZIP` file from `github.com` or performed a `git clone` of the `bipartgraph` repository, the software is ready for the final set up. Move to the `bipartgraph` directory (`bipartgraph-master` if you installed from the `ZIP`).

A) Windows

Double-click on install_bipartgraph.

If you have set up R in your Path variable you may also install with the commands of the UX like style

B) UX like

Type in the terminal:

```
Rscript install_bipartgraph.R
```

Installation procedure is very fast in Windows because all packages are precompiled. In Linux it may take several minutes because your machine will compile several packages. Please, be patient and do not close the terminal until the procedure finishes.

Workaround

Depending on your user privileges, install_bipartgraph may fail in Linux if the current user has no permissions to write in the packages directory. If so and ony if so, type instead:

```
sudo Rscript install_bipartgraph.R
```

Once the script finishes, the installation procedure is complete. Check that everything was installed properly. Open a new terminal (or command line), go to to the bipartgraph directory, or bipartgraph-master if you installed from the ZIP file. Double-click on the plot_test (Windows) icon or type in the command line (UX like):

```
Rscript plot_test.R
```

Can you see a window with figure 6? If so, the installation was OK.

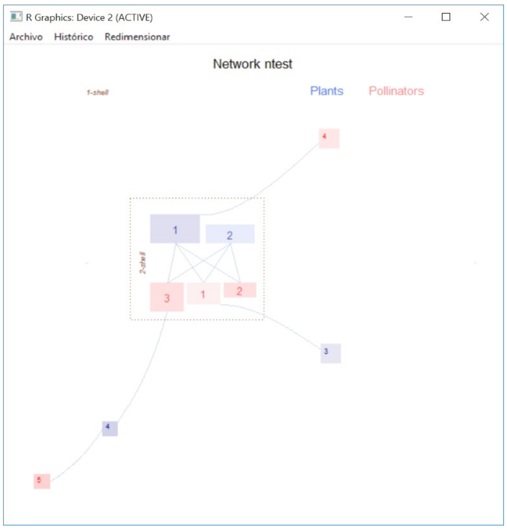


Figure 6: Result after running the script plot_test.R

2.5 Running *BipartGraph*

The application is now ready to run. Launch the script double clicking on `bipartgraph` icon (`Windows`) or write in the command line (`UX like`):

```
Rscript bipartgraph.R
```

Your default web broser will open a new tab with a local `URL`. If everything went right you have reached the landing page.

Click on **Data** (main menu) and then click on the **Data file** box and select `M_PL_008.csv`. Click on the **Interactive Ziggurat** tab (main menu) and you will see the network plot of figure 7.

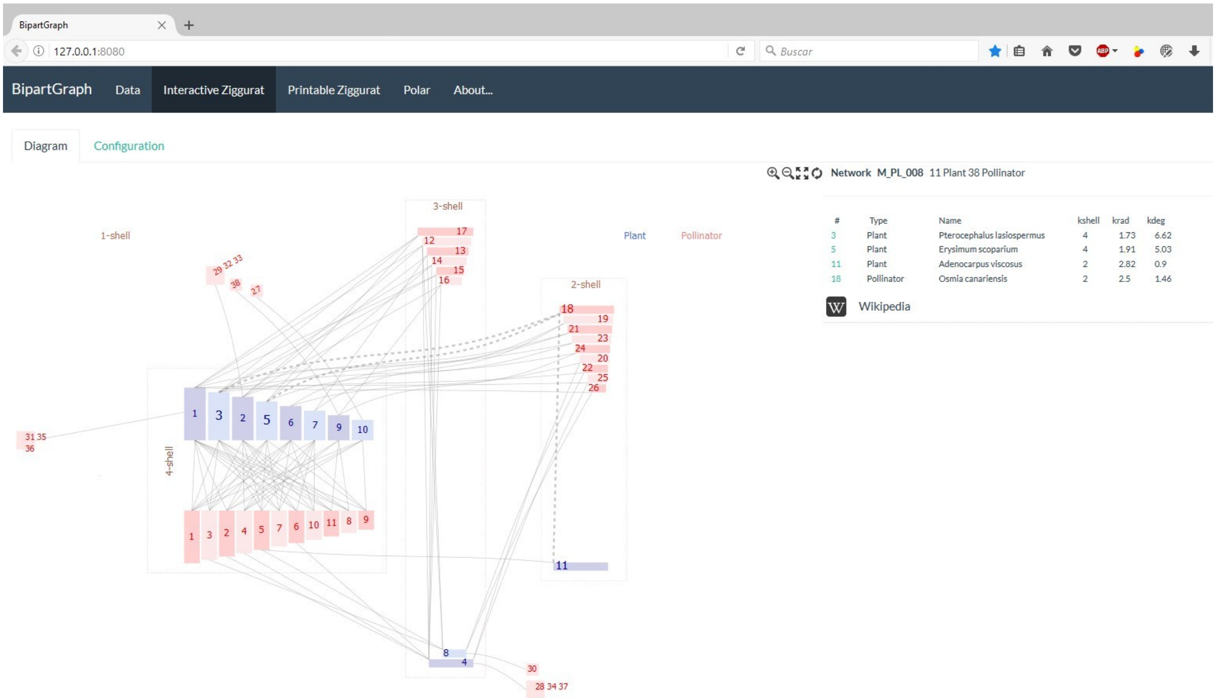


Figure 7: Ziggurat plot

2.6 Installation on Linux

These section explains how to install bipartgrah on a Debian based Linux. With minor modifications for install commands is valid for any other distro.

Run the script linux_dependencies

```
> sudo sh < linux_dependencies
```

This script downloads and install the R runtime environment and the libraries to compile dependencies.

Run this command tin install the gcc compiler

```
> sudo apt install build-essential
```

DOWNLOAD ZIP from `https://github.com/jgalgarra/bipartgraph` (then UNZIP and let "bipartgraph-master" folder on any path of your user home; go to that folder) or git clone the repository and go to the "bipartgraph-master" folder

```
> Rscript install_bipartgraph.R
```

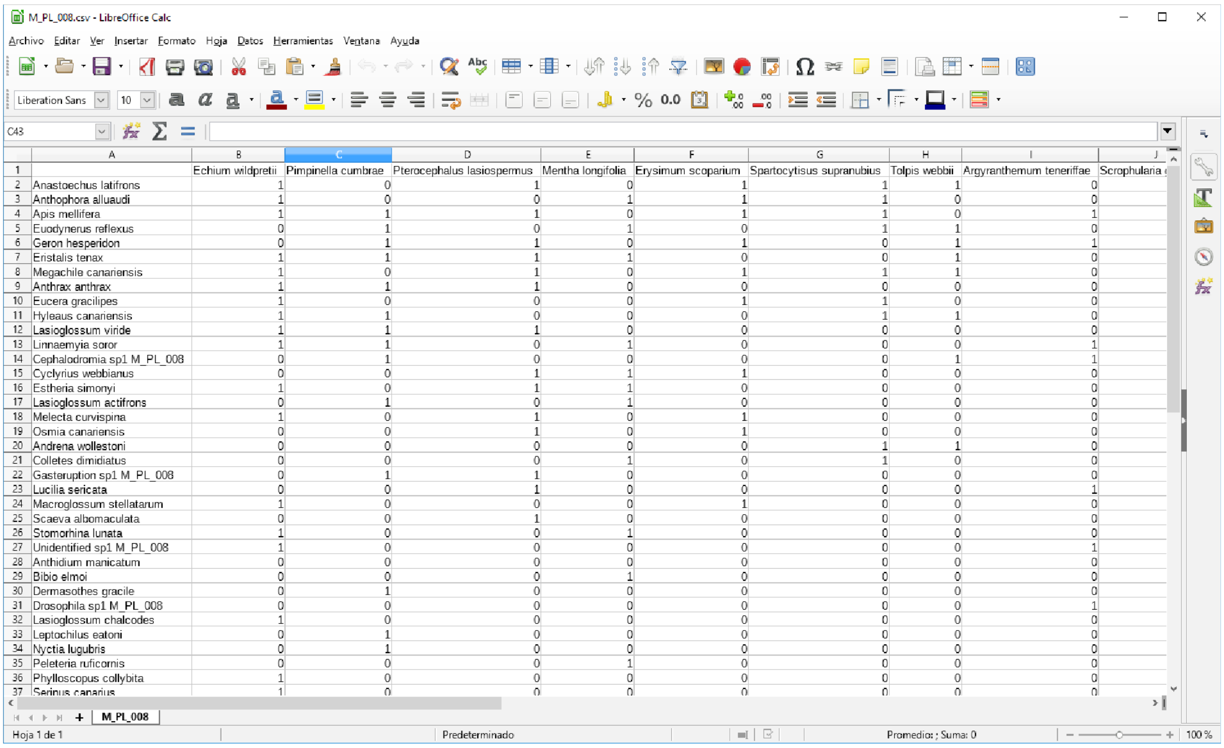
If you get... "This is the end, my friend. Installation completed" , everything went OK :-)

You can check that installation is fine with the following command:

```
> Rscript plot_test.R
```

3 Data management

BipartGraph is a tool to plot ecological communities, but you may use it with any bipartite network. In this guide we use examples of mutualism and parasitism well documented in the web of life database [4].



	Echium wildpretii	Pimpinella cumbrae	Pterocephalus lasiospermus	Mentha longifolia	Erysimum scoparium	Spartocytisus supranubius	Tolpis webbii	Argyranthemum teneriffae
Anastoechus latifrons	1	0	1	0	1	1	1	0
Anthophora alluaudi	1	0	0	1	1	1	0	0
Apis mellifera	1	1	1	0	1	1	0	1
Euodynerus reflexus	0	1	0	1	0	1	1	0
Geron hesperidon	0	1	1	0	1	0	1	1
Eristalis tenax	1	1	1	1	0	0	1	0
Megachile canariensis	1	0	1	0	1	1	1	0
Anthrax anthrax	1	1	1	0	0	0	0	0
Eucera gracilipes	1	0	0	0	1	1	0	0
Hyleaus canariensis	1	1	0	0	0	1	1	0
Lasioglossum viride	1	1	1	0	0	0	0	0
Linnaemyia soror	1	1	0	1	0	0	0	1
Cephalodromia sp1 M_PL_008	0	1	0	0	0	0	1	1
Cyclyrius webbianus	0	0	1	1	1	0	0	0
Estheria simonyi	1	0	1	1	0	0	0	0
Lasioglossum actifrons	0	1	0	1	0	0	0	0
Melecta curvispina	1	0	1	0	1	0	0	0
Osmia canariensis	0	0	1	0	1	0	0	0
Andrena wollestoni	0	0	0	0	0	1	1	0
Colletes dimidiatus	0	0	0	1	0	1	0	0
Gasteruption sp1 M_PL_008	0	1	1	0	0	0	0	0
Lucilia sericata	0	0	1	0	0	0	0	1
Macroglossum stellatarum	1	0	0	0	1	0	0	0
Scaeva albomaculata	0	0	1	0	0	0	0	0
Stomorhina lunata	1	0	0	1	0	0	0	0
Unidentified sp1 M_PL_008	1	0	0	0	0	0	0	1
Anthidium manicatum	0	0	0	0	0	0	0	0
Bibio elmoi	0	0	0	1	0	0	0	0
Dermasothes gracile	0	1	0	0	0	0	0	0
Drosophila sp1 M_PL_008	0	0	0	0	0	0	0	1
Lasioglossum chalcodes	1	0	0	0	0	0	0	0
Leptochilus eatoni	0	1	0	0	0	0	0	0
Nyctia lugubris	0	1	0	0	0	0	0	0
Peleteria ruficornis	0	0	0	1	0	0	0	0
Phylloscopus collybita	1	0	0	0	0	0	0	0
Serinus canarius	1	0	0	0	0	0	0	0

Figure 8: Interaction matrix of network M_PL_008 [5]

Data are stored as `.csv` files. Fields are separated by commas, first row stores the binomial names of guild A species and first colum those of guild B. When you upload your own data, remember that the application waits to found there that information. If you do not have it, or it does not make sense at all in your environment, leave the first row and the firsr column filled with void strings. *BiparthGraph* will label species following the order in the input file. If the interaction matrix is binary, the cell of *species A* `m`; *species B* `n` will be set to `1`. If it is weighted, to a real number different of `0`. We provide examples with the default installation software. To upload your own data, click on the Data tab of the main menu and then on Manage Files.

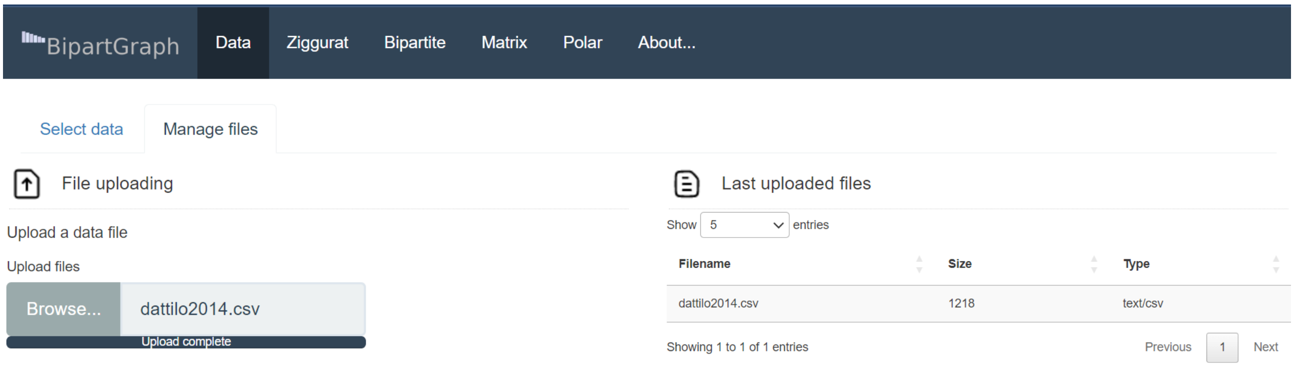


Figure 9: Uploading a file

Click again on Select Data and choose any file. The table shows the inteaction matrix. You can change Guild A and Guild B labels, *BipartGraph* will remember it next time you work with the same file.

The **Network Anaylisis** button creates a `.csv` file with the individual *k-magnitudes* of all nodes. With Reset all, session restarts and all configuration parameters are set to their default values.

The Species names button downloads just the species names of a network, this list may be quite handy for the caption of a `LaTeX` image, for instance.

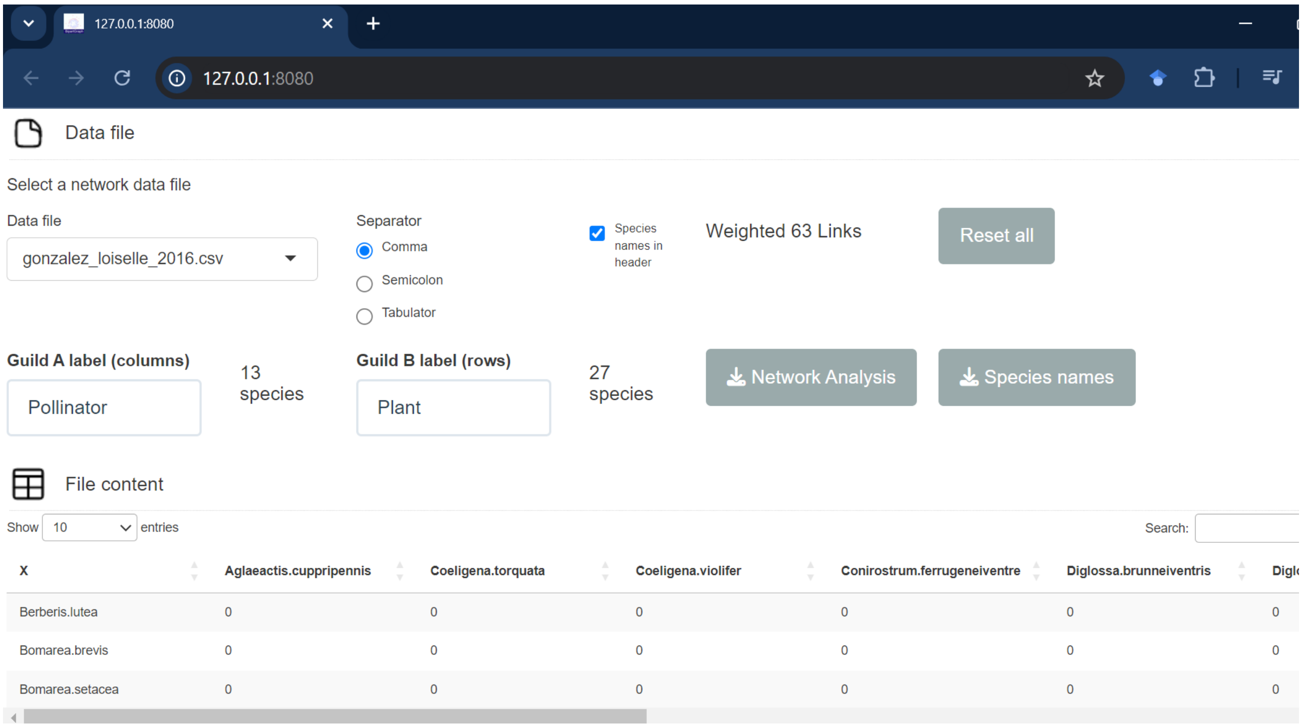


Figure 10: Network data

4 The ziggurat plot

The ziggurat graph is an original kind of visualization. The idea behind it, is splitting species in sets by their *k-shell* numbers. Each of these groups are represented as small ziggurats.

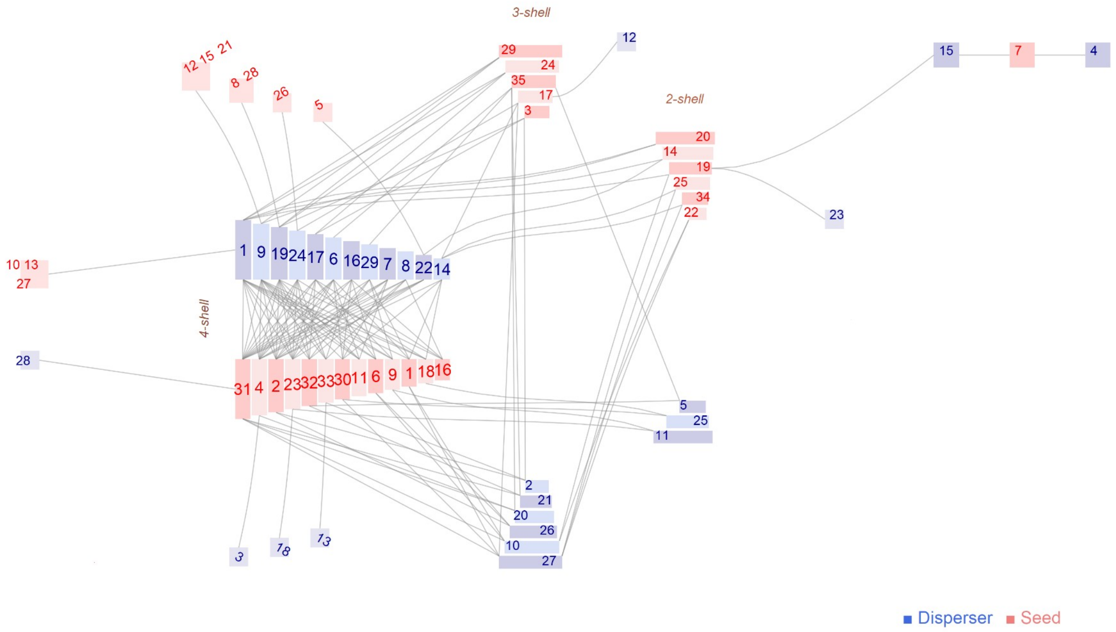


Figure 11: Ziggurat plot of an eating fruit birds network in Brazil [4]

Networks are very different in size and connectivity but they all share building units that will help you to get a clean view of the structure (fig. 12). The building blocks are ***nodes*** and ***links***. Bipartite networks have two kind of nodes that conventionally we call ***guilds***. In this example there are frugivore birds, playing the role of seed *dispersers* (guild A), and plants that yield those *seeds* (guild B). If two species interact, there is a link between them, for instance *bird 18* feeds on fruits of *plant 23*. The bird fosters its reproduction by seed dispersal. Links among species of the same guild cannot exist in bipartite networks.

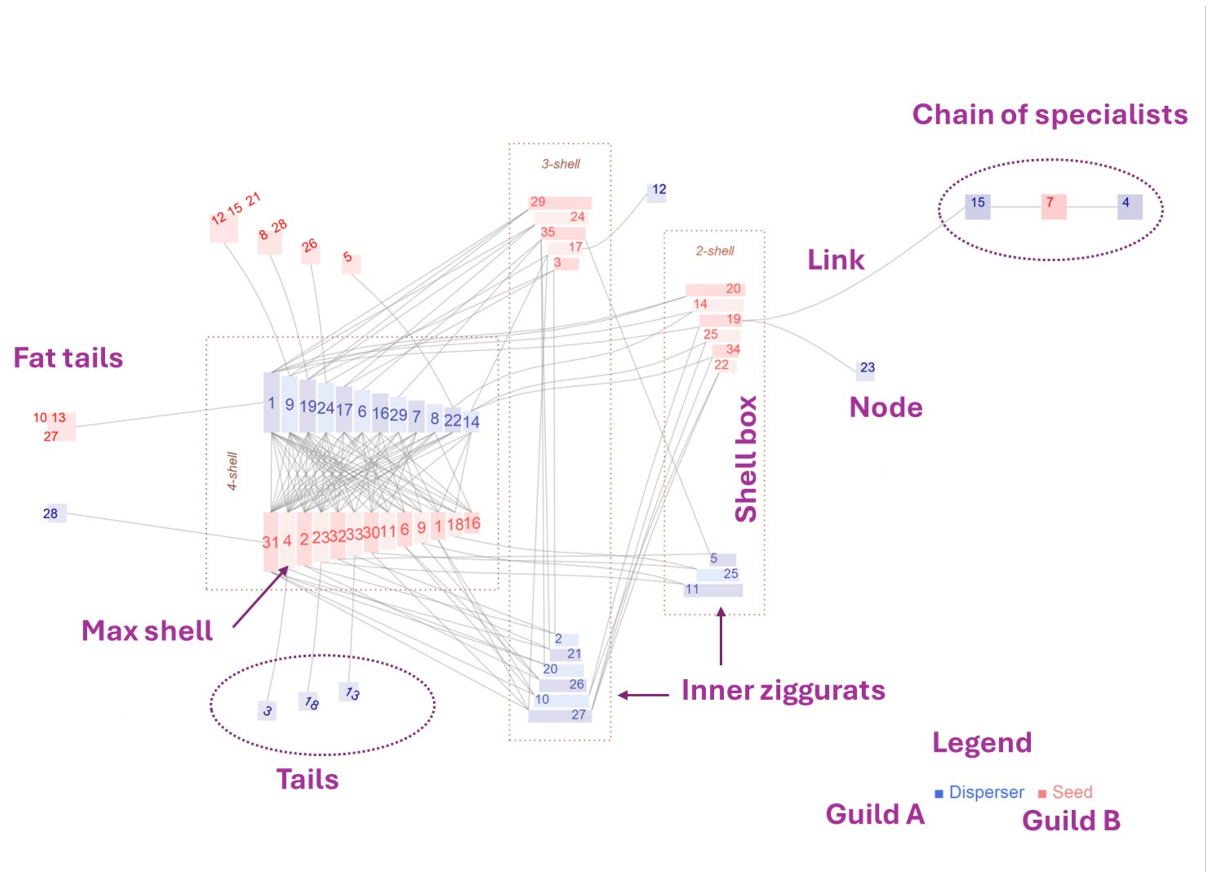


Figure 12: Elements of the ziggurat plot

All links have the same color and no label. Nodes are filled with the color of their guild, with two slightly different tones to help visual perception. Each species is labeled with its order in the input file. So, *plant 1* is the species recorded in column 1 and *bird 5* in row 5. This is the only meaning of this number.

Node size does not convey any information in the ziggurat graph, shapes just make it more readable. Although species of a same guild and shell appear stacked in the ziggurat, they do not have share ***any*** other property. *Plants 2, 10, 20, 26* and *27* belong to the *3-shell*, this is the only fact relevant for this plot.

The ***legend*** tells us how we choose to name each guild, the left element is always *guild A*. Ziggurats may optionally appear enclosed by dotted rectangles that we call ***shell boxes*** with a box label. The shell with the highest k-index (4 in the example) is the ***max shell***. This shell is located at the centre of the plot, with the species of both guilds in two mirrored triangles. Shells of lower indexes (3 and 2) are arranged following an almond distribution. We call them ***inner ziggurats***.

Nodes of *1-shell* are scattered around the plot. They may be part of two structures, ***tails*** and ***specialists chains***. A tail is just one species of 1-shell linked to one species of a higher shell of the opposite guild, for instance, *bird 5* and *plant 3* are tails.

When multiple species of *1-shell* are tied to the same node of any of the ziggurats, they are clustered to reduce the number of lines. *Plants 12, 15* and *21* are linked to the generalist *bird 9*.

Specialists chains are much less common than tails. They are set of species of *1-shell* linked to species of *1-shell* of the other guild. *Bird 17–plant 7–bird 4* is an example of a chain of specialists.

With this organization, we have a clear view of structure and interconnections. The central almond shape area leaves a wide space for the links, so they do not over-cross the boxes. Other structural details are easy to catch with this visualization. It is also clear why *bird species 4* and *plant 7* are in a dangerous position. They depend on another specialist, *bird 15*, that has only one link with *2-shell*. Sometimes, the field observers record species not linked to the giant component. These ***outsider*** species are plotted as a subnetwork under the ziggurat plot (fig. 13).

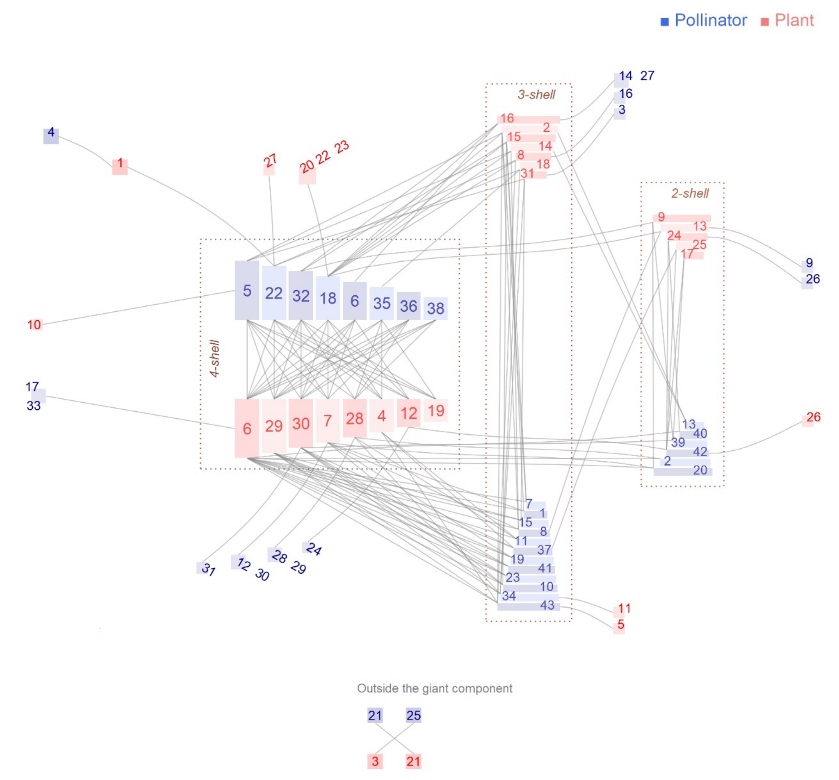


Figure 13: Plant – pollinator community in Syndicate, Dominica Island [6]. M_PL_041

4.1 Explore the interactive ziggurat

Go to Data and select the `M_PL_008.csv` and the click on Interactive Ziggurat. The plot is displayed on the left, inside the Plot panel.

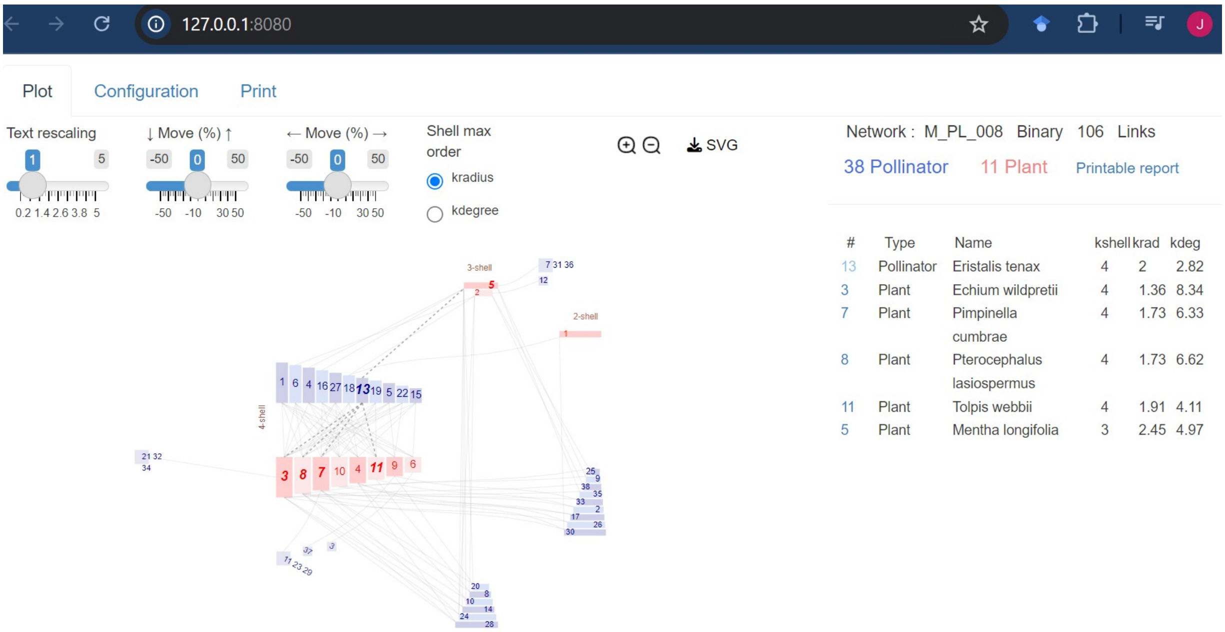
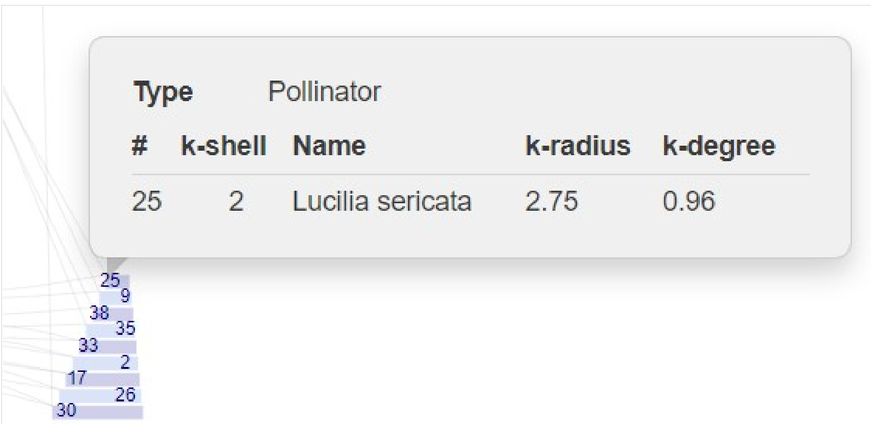


Figure 14: Interactive Ziggurat

When you pass the mouse over one species box, a tooltip shows its binomial name and *kparameters*.



If you click on one of these boxes, the links to other species are highlighted as dashed lines, and the labels of those neighbors look bigger.

On the right panel, you may see the information table of the selected species (*disperser 13* and all its partners in fig. 14). Clicking on the species number in the table, the application opens the `Wikipedia` information about that species. If the binomial name is not properly written in the data file you will get an error. Sometimes, `Wikipedia` may redirect you to some disambiguation page.

4.2 Modifying the look and feel of the ziggurat plot

The elements of a ziggurat are configurable for the sake of clarity and beauty. Let's start with the simple ones, ***links***. Please, select the file RA_HP_001.csv, a host-parasite network, with 18 parasite species, 10 host species and 61 links. If it is the first time you load it, change the Guild A label to Parasite and Guild B label to Host in the Data panel. *BipartGraph* will remember this setting next time you load these data. Click Interactive Ziggurat on the main menu, and you will get a fancy default visualization of this small network (fig. 15).

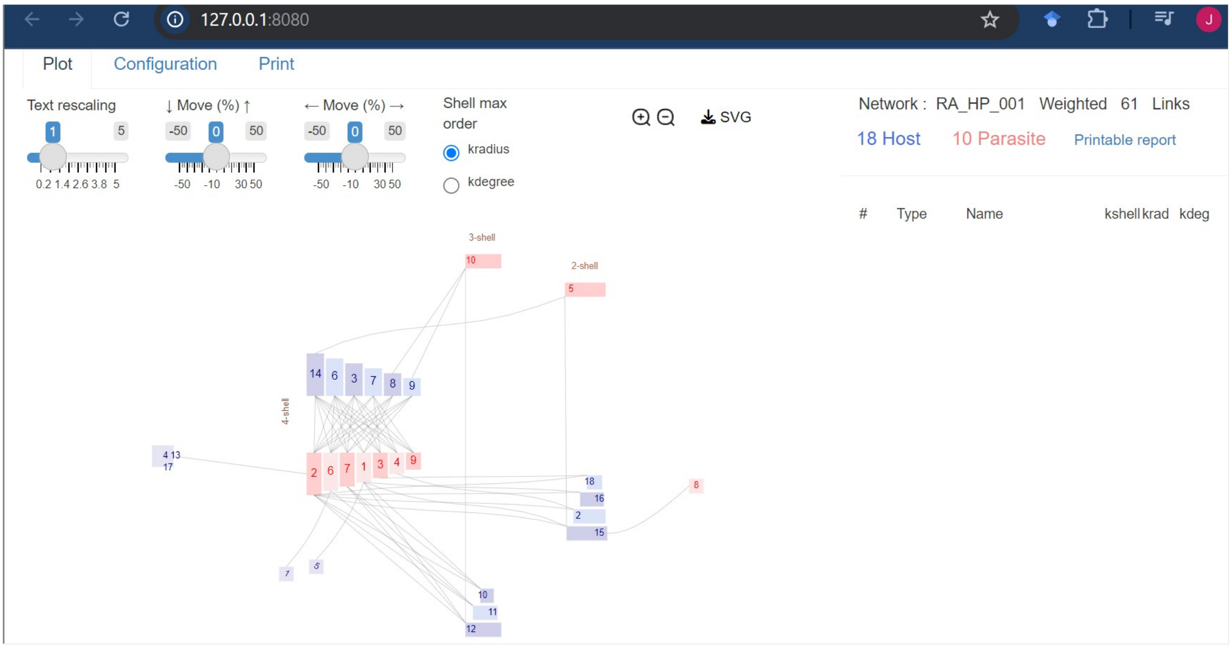


Figure 15: Host – parasite network in Azharia [7].

Click on Configuration tab and the first thing you see are the link controls. There are 5 of them (fig. 16).

- Show links. This box is checked by default. If you unchek it, links will hide. This may be useful when working with very big network, because link processing is the most time consuming task. You may decide hide them while you configure other elements.
- Use spline. Links come in two flavors, straight lines (*parasite 7 – host 6*) and splines, smoothly curved lines (*parasite – host 10*). If you uncheck, all links will be straight shaped and the plot will have a more *rude* look.
- Spline points. Number of points of each spline, the bigger the smoother and more time consumig. The default value is 50, do not increase it very much unless you are producing a very high quality plot for a journal.
- Link width. If the network is binary (there are only 1's and 0's in the in the interaction matrix) all links have the same width. If weighted, Weight aggregation lets you choose to make their width proportional to the interaction strength (ln or log10 of these values) or ignore weight (no) and plot them as if the network were binary.

- Color and Transparency define how the links are filled. The transparency ranges from 0.1 (almost invisible) to 1 (opaque color).

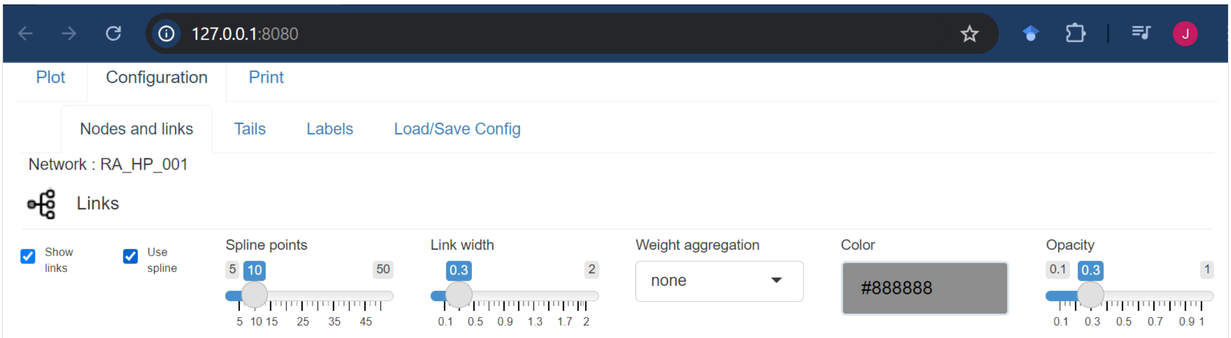


Figure 16: Link configuration control.

Let's go back to figure 15 to make some adjustments. First, uncheck the Use spline box (the Spline points control has no meaning if you use straight line links). We want to make the width proportional to the interaction strength but just in this moment we do not remember if the network is weighted or not. Click **Data** on the main menu and observe the **File Content**. The interaction matrix is not binary.

Click again on **Interactive Ziggurat / Configuration**. We select ln, that means that the width of each link will be proportional to the natural logarithm of its strength (the value recorded in the interaction matrix).

To test the effect of other configuration paramenter, increase the Link width slider up to 0.5, move the Transparency control to 0.2 and pick a green tone color #087007 (you can write it inside the Color box).

That's all, now it's time to see how the ziggurat changes. Click on the **Plot** tab and the software will plot again the network (fig. 17). Perhaps it will not win a plot beauty contest, colors pop up your eyes and straigth links make it less pleasant than the default fig. 15. On the other hand, you can detect that *host 7* is much prone to have parasites of *species 6* than of *species 10*. We have learned how to modify ***links***, now we move on to more exciting elements of the graph.

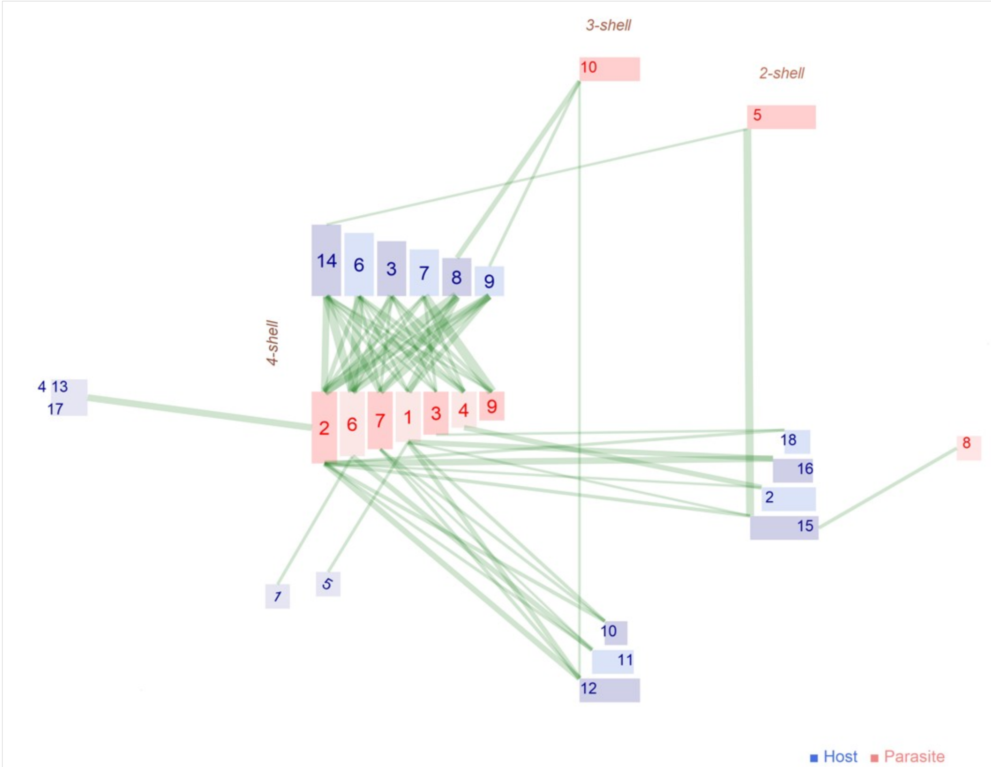
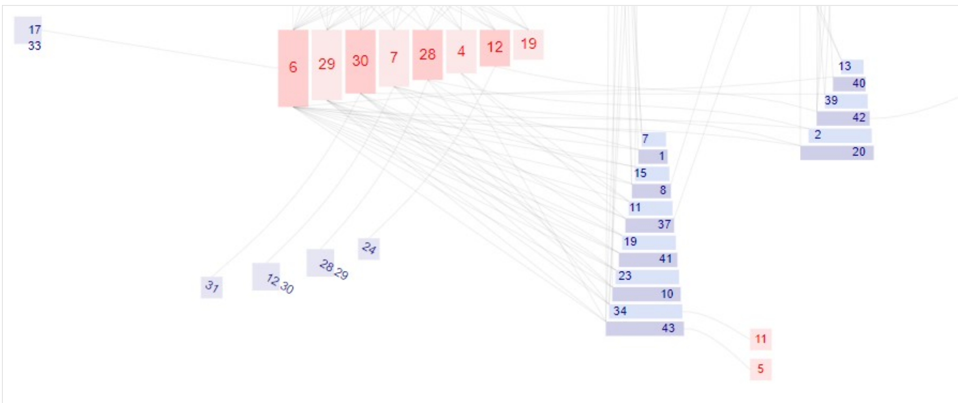


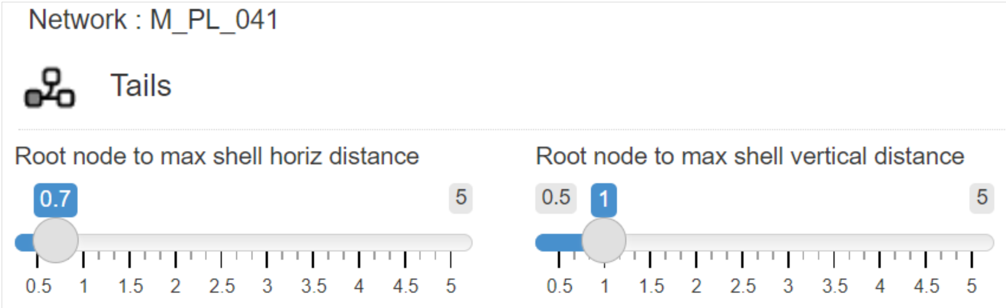
Figure 17: Host – parasite network in Azharia [6], with modified link properties.

The network of figure 13 is bigger. There are some details that need improvement, for instance the height of the nodes of the ***inner ziggurats***, they look smashed. Go to Configuration, move the Nodes height slider to 1.5 and click on Plot again.

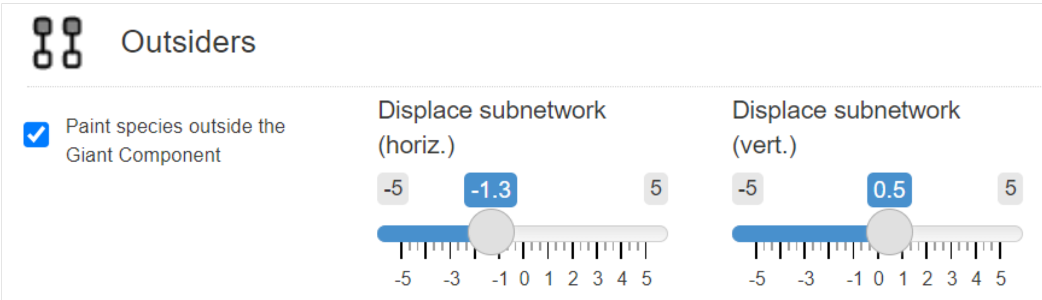


Boxes are fancier now, but links from *2-shell* towards *4-shell* now cross the ziggurat of pollinators of *3-shell*. We can move it upwards or downwards with the Inner ziggurats vertical displacement control. The ziggurat that is troubling you is the *3-shell* of guild A (if you do not know wich guild you are dealing with, the color of the legend will tell you).

Please, move the 3-shell A slider to -0.1 (roughly move it down a 10% of the original position). The ***chain of specialists*** in the upper left end of figure 13 is a bit long. Click the Tails tab and reduce the distance.



Finally, move the ***outsiders***, clicking on the Nodes and links tab:



After you apply all these changes, the ziggurat plot you get is the figure 18. Of course, you could keep on tuning parameters.

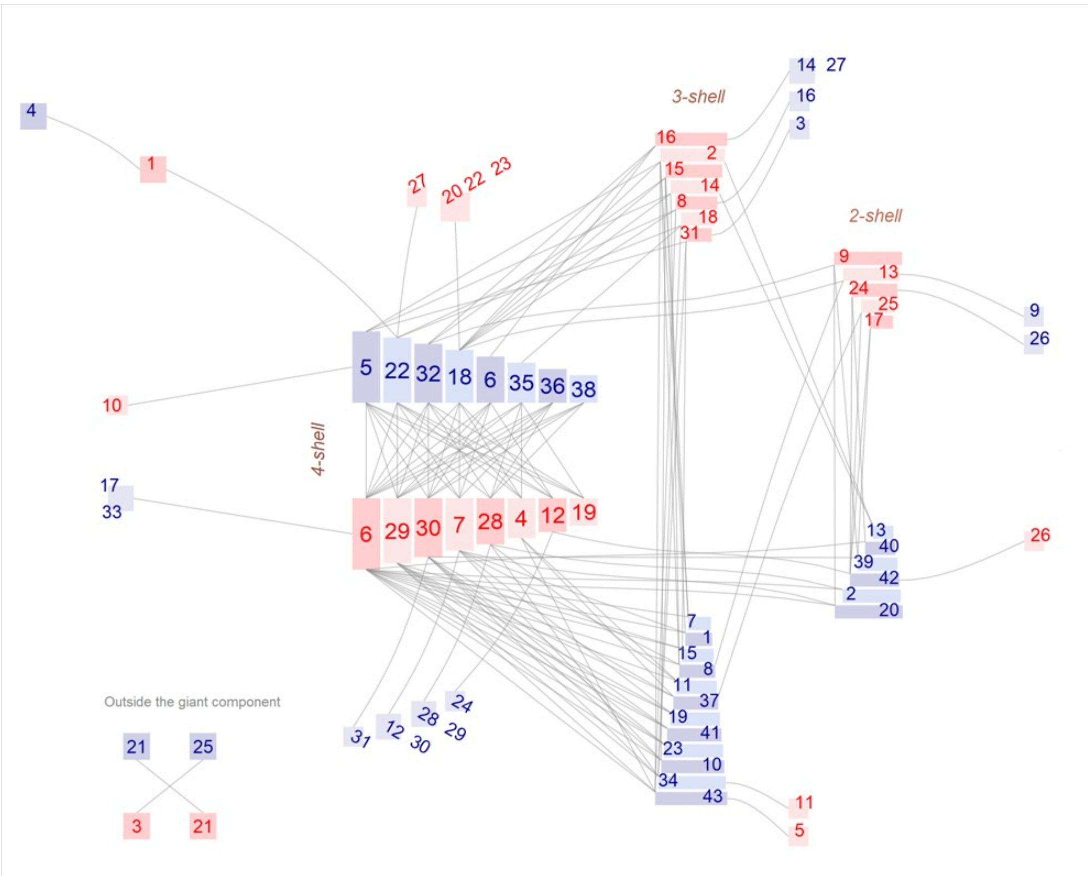


Figure 18: Ziggurat graph of a plant – pollinator community in Syndicate, Dominica Island [5] after visual adjustments.

For instance, outsiders may be a bit disturbing, and you may hide them or reduce their area. This network is weighted, try to change link properties as we did in the previous example and see how the plot tells a different story.

A new example will show how to deal with other elements of the ziggurat. Go to Data and before selecting a new file push the **Reset all** button. This action restores the default values of all the controls. Now, select the `M_PL_012.csv` file and display the interactive ziggurat.

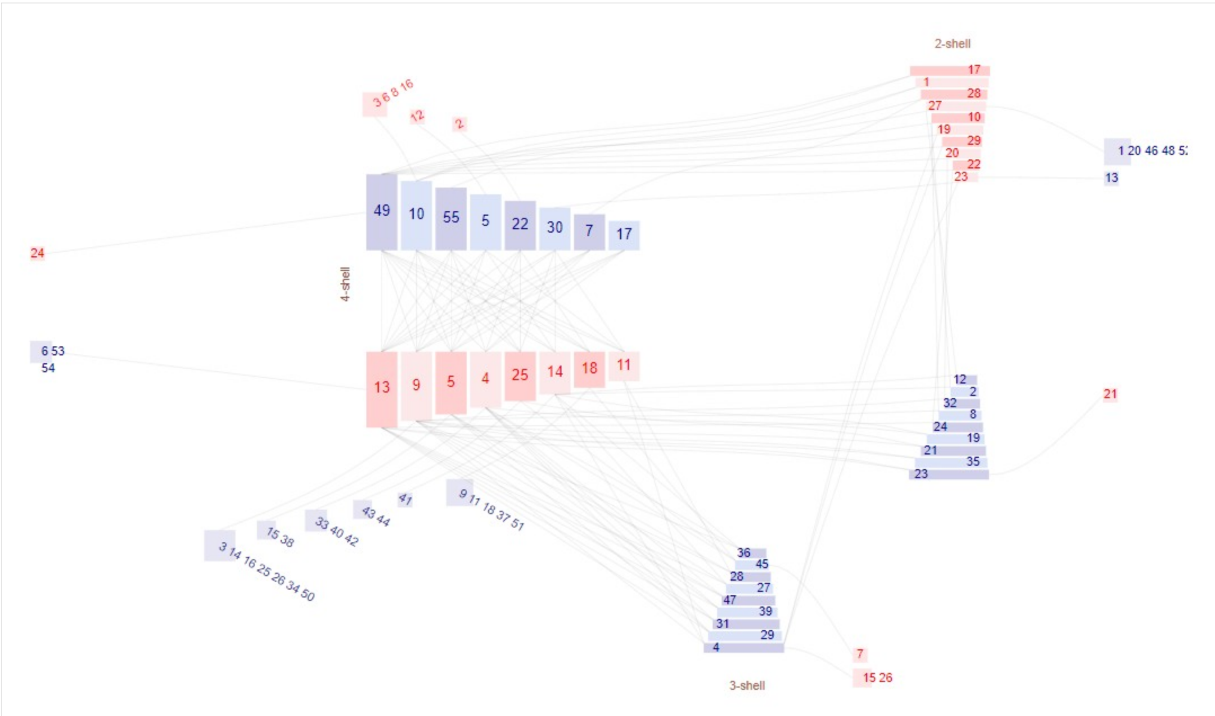
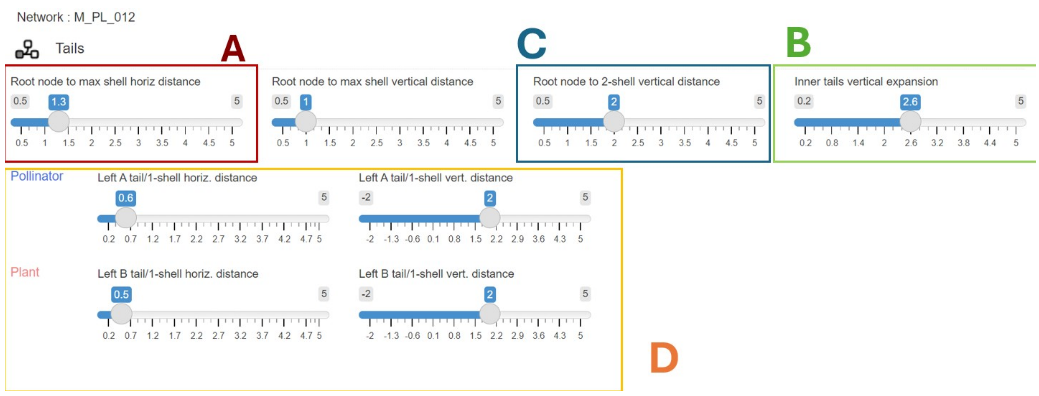


Figure 19: Ziggurat graph of a plant – pollinator in Garajonay (Spain), recorded by Jens Olesen. Unpublished.

This community has a high number of species in *1-shell*, grouped in ***tails***. Pay attention to the tails that are above and below the ***max shell***. The graph is overcrowded and it would be nice making additional room for them, expanding their horizontal distance (**A**). Then, we increase the vertical distance among tails connected to inner ziggurats (**B**), like plants 7*, 15* and *26*. The third step is widening the gap of the two tails connected to plants *2-shell* (**C**). The last modification pulls up/down and right the left tails, those connected to the leftmost species of the ***max shell*** (**D**).



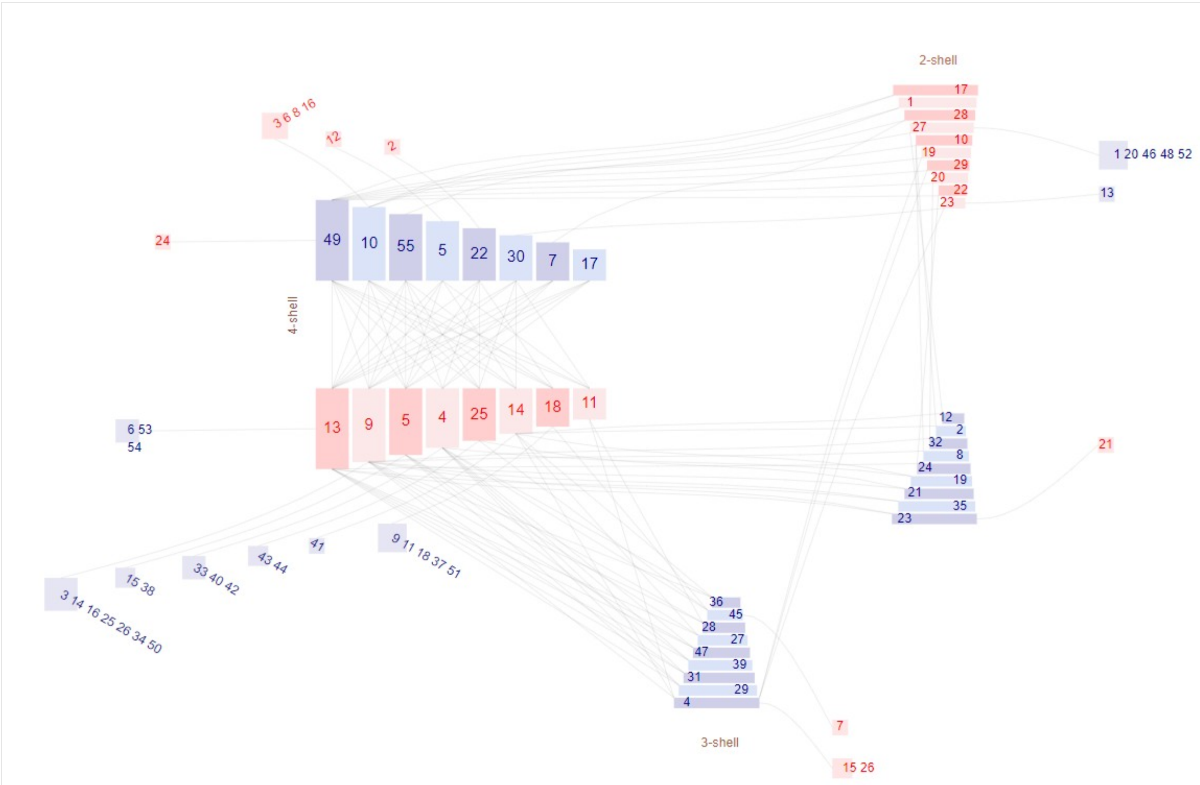
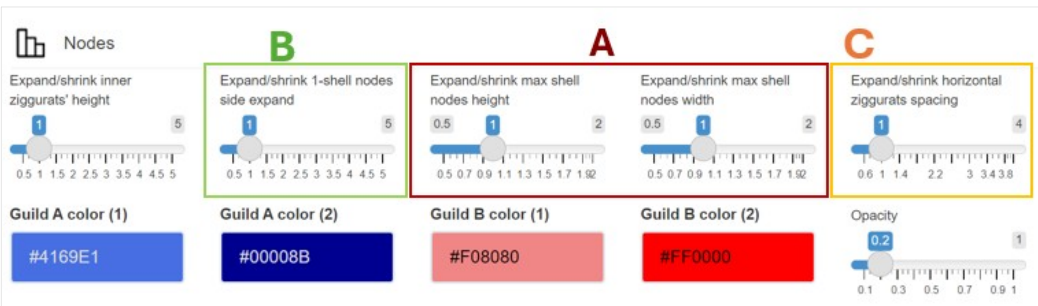


Figure 20: Ziggurat graph of a plant – pollinator in Garajonay (Spain), after visual adjustments.

The dimensions of the ***max shell*** triangle are controlled with two sliders Max shell height expand and Max shell width expand (**A**). The area of all boxes of *1-shell* may be increased by the factor 1-shell nodes expand (**B**). There are four color pickers for guild nodes, please use it with care. *BipartGraph* will remember your choice next time you load the same network. Transparency works as explained above when talking about ***links***. The horizontal inner shells separation widens the horizontal gap between ziggurats (**C**).



Let's take a small portion of network `M_PL_031` to show the behavior of ***specialists*** controls (fig. 21). They expand or contract these chains. The distance of the root node is modified by First node horiz/vert distance to network if tied to nodes in any ziggurat except *2-shell*, that has their own controls.

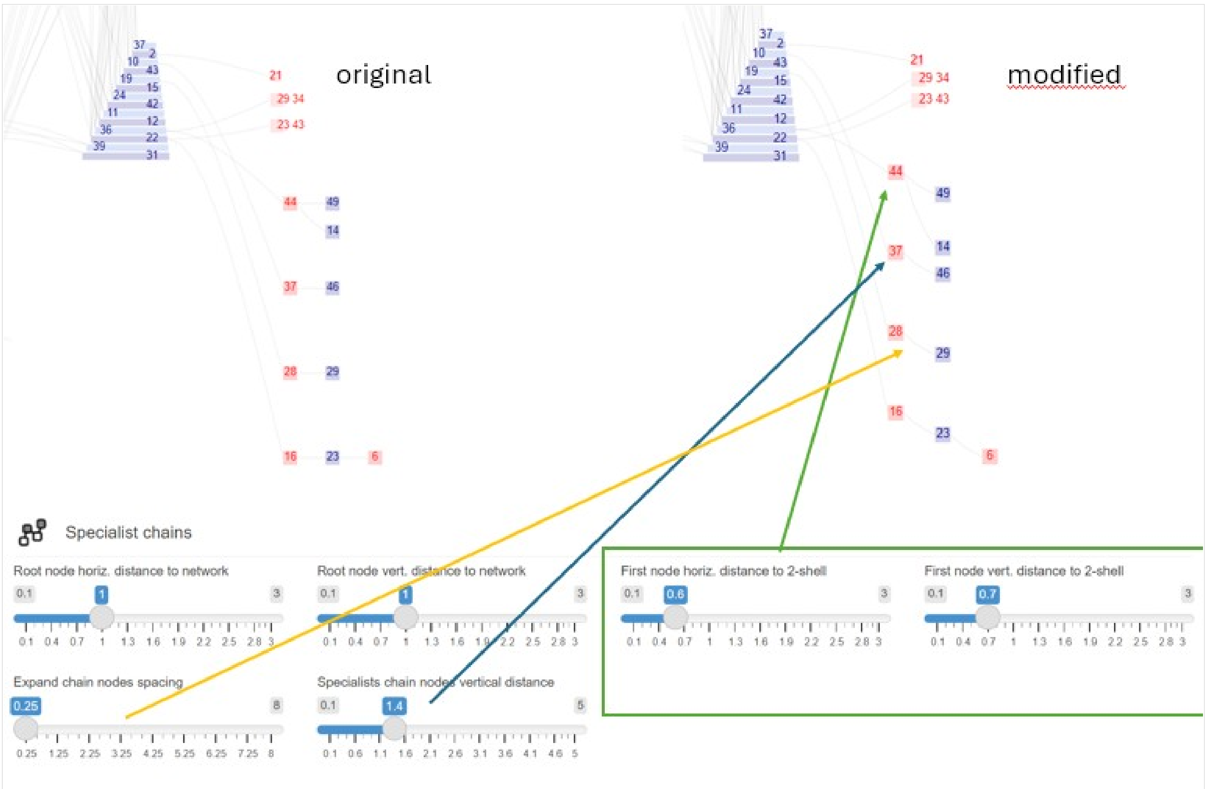


Figure 21: Changing the configuration of the chains of specialists.

Expand chain boxes separation controls the horizontal distance among nodes of any chain and Specialists chain nodes vertical distance the vertical gap.

4.3 Printable ziggurat

Once you have got a clean and nice ziggurat it is quite possible that you need to produce a high quality plot for your research. That is what the Print tab of the main menu will do for you.

Select again the network `M_PL_012.csv` and visualize the default Interactive Ziggurat. Now, go to Print and click the button **Plot Download**. Your broser will download a file called `M_PL_012-ziggurat.png` (fig. 22). Compare it with the interactive plot (fig. 19). It is almost identical, but there are minor visual differences. In the printed plot the name of the network and legend can be displayed, and perhaps the sizes of labels and legend are slightly different.

The interactive and printed plots are built with two different techonologies, the first one is an `SVG` object, the second a `ggplot2` object. These details are not important for you as an end user of the application, but depending on the resolution of your screen and the installed fonts, sizes may be different. In the Plot view you have trhee controls to rescale at once all label size, and to displace the SVG plot. The default value or rescaling is `1`, if you increase it, the size of labels and width of the links of the interactive plot will grow compared to the printed ones. It may be useful to adjust it once if dimensions are rather different.

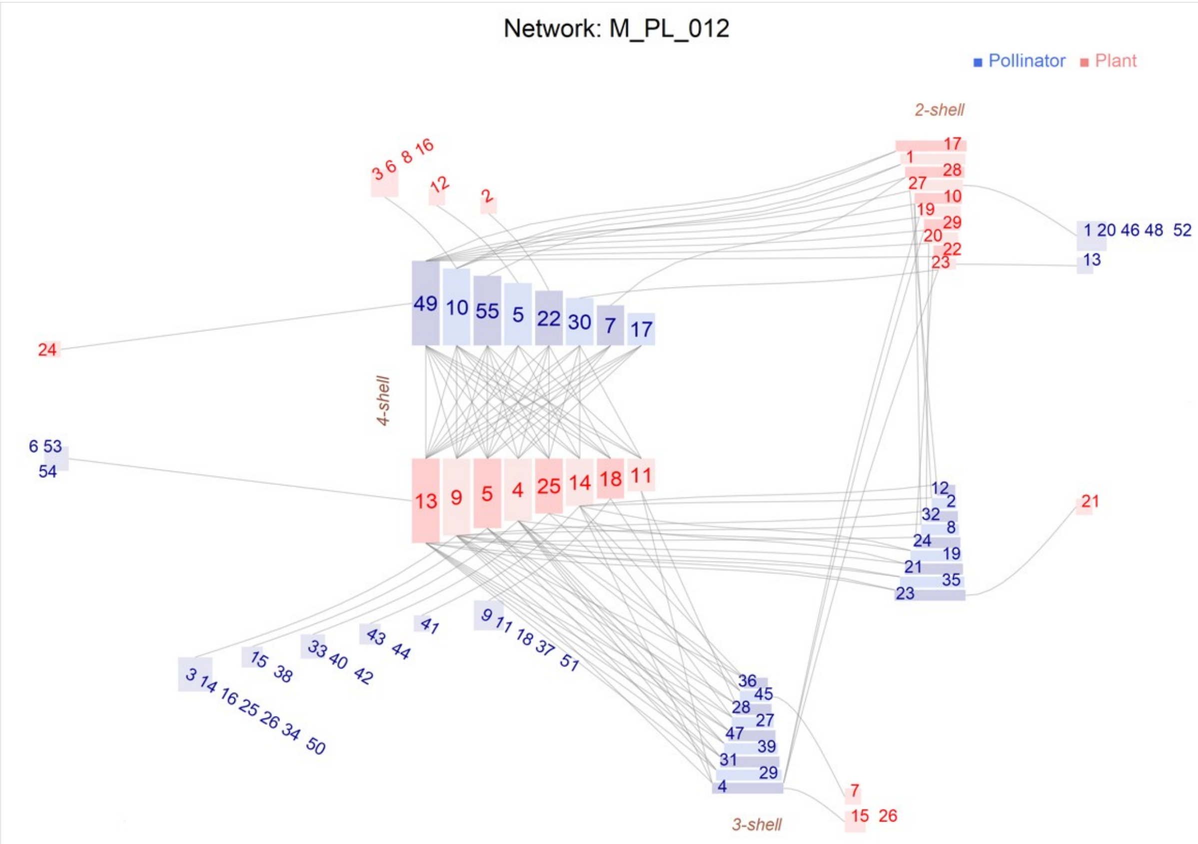


Figure 22: Printed Ziggurat graph of a plant – pollinator in Garajonay (Spain).

Go back to the Print pane. Options are quite straightforward, you may change the Paper size, orientation (Landscape), resolution and title and legend options. The Aspect Ratio is a special control that only affects the printed version. If it is bigger than `1` the graph is stretched in the vertical direction, the opposite if it is smaller. You can choose four file formats for the printed plot: `png, eps. tiff` and `svg`.

When you get a very nice printed ziggurat you will want to save the configuration for reproducibility. This is what the **Dowload generating code** button does. Click on it and you will see the full invocation of the `R ziggurat_graph()` function. The meaning of each invocation parameter is explained in the `kcorebip` user manual. This package was installed in your computer during the process and the guide explains how to run the code:

https://github.com/jgalgarra/kcorebip/blob/master/inst/doc/kcorebip_man.pdf

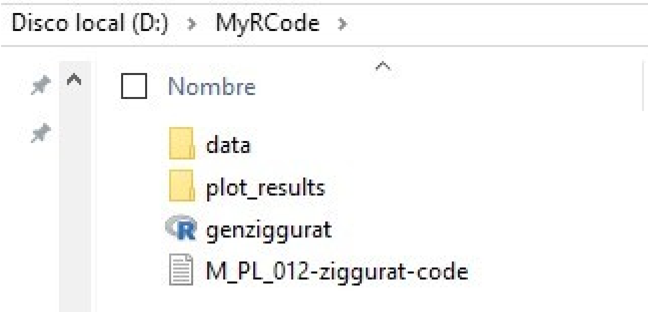
If you are not a programmer but want to get the printed ziggurat again using the code, this is what you have to do. Create a directory called `MyRCode` where you want, and copy there the file `M_PL_012-ziggurat-code.txt` that the application generated. Create a folder called `data` inside that directory and copy there the input data files, for instance the `M_PL_012.csv` that you used as an example and lives in `biparthgraph/data`.

Create inside `MyRCode` the folder `plot_results`, and inside plot_results the folder `ziggurat`.

Copy and paste this code and save it in a file called `genziggurat.R` in the `MyRCode` directory.

```
library(kcorebip) source("M_PL_012-ziggurat-code.txt")
```

Your new directory should look like that, with the file `M_PL_012.csv` in the data folder and an empty `ziggurat` folder inside `plot_results`.



Go to the command line and run

```
> Rscript genziggurat.R
```

When the script finishes the `M_PL_012_ziggurat.png` file is available inside `plot_results/ziggurat`.

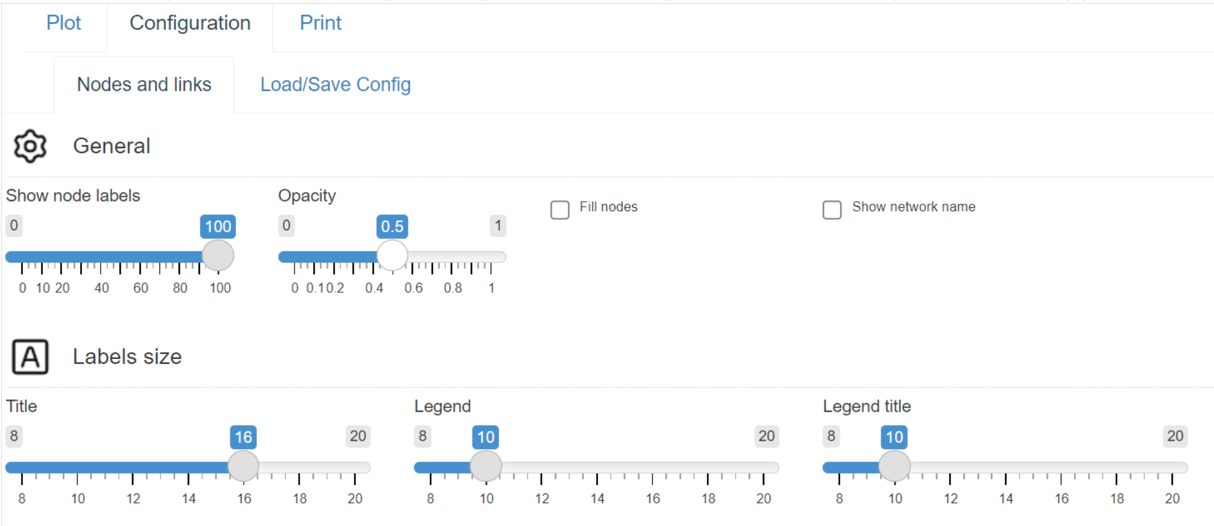
If you are not a developer but you want to save the tuned parameters for the future, then go to the **Config/Load Save Config** pane. Use the **Save plot configuration** to save the values in a text file. You can read it later with the **Browse** dialogue, to choose any configuration file. Checking “Show contents” will show the values of each parameter that has been stored in that paticular file.

5 The polar plot

The goal of the Polar graph is showing species centrality and network compactness. It provides a quick overview of network distribution. It was inspired by the fingerprint-like graph, developed by Alvarez-Hamelin et al. [8] to plot very large k-decomposed networks.

Species are plotted at *k-radius* from the center. Angles are assigned by the visualization algorithm to reduce overlapping, with the condition that each guild lies inside one of the half planes. The area of each node is proportional to its *k-degree* and color represents the *k-shell*. This visualization does not include links. The user may choose adding the histograms of *k-magnitudes*, a handy option because they convey a wealth of structural information. Polar graphs are specially useful when comparing different networks side by side, even if they are of very different sizes.

The configuration of polar plot is quite easy compared to the configuration of ziggurat.



The color is set by the algorithm, you may choose the level of transparency and if you want to fill them. The pixels box controls the size of the image in screen and may be useful to adapt it to your equipment.

If you want to add the node number, use Show node labels. It tells the application how many node labels to show, from highest degree to lowest. If the network is big the labels may spoil the plot. You can choose to fill the nodes and the level of opacity.

The Labels size control plays the same role than for the ziggurat plot.

Once the plot is displayed you may download it or download its generating code, just the same that happens with the ziggurat plot. An important difference is that the polar plot shown in the browser and the printed version are the same object, so you do not need to worry about your screen resolution.

Figure 23: Polar graph of a plant – pollinator in Garajonay (Spain), including labels and histograms. Compare with fig. 19

6 The bipartite plot

The bipartite plot is the conventional way to depict a bipartite network. Nodes of both guilds are aligned in two parallel lines (horizontal or vertical is a matter of convenience), ordered by degree. A straight line between two nodes of different guilds means that there is a link and its width may be proportional to the weight of the interaction. The bipartite plot becomes a hair ball as the number nodes and links increases. There are some minor mitigation strategies to overcome this problem. This package offers the legacy style for bipartite plot, following the traditional layout, but distributing evenly the nodes of the smallest guild. This solution makes the diagram more readable.

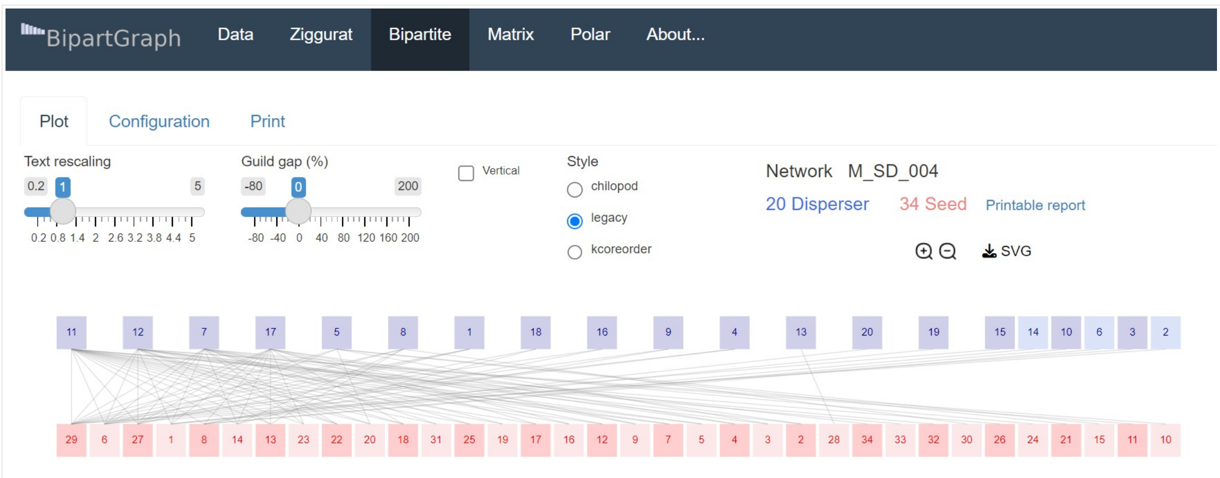
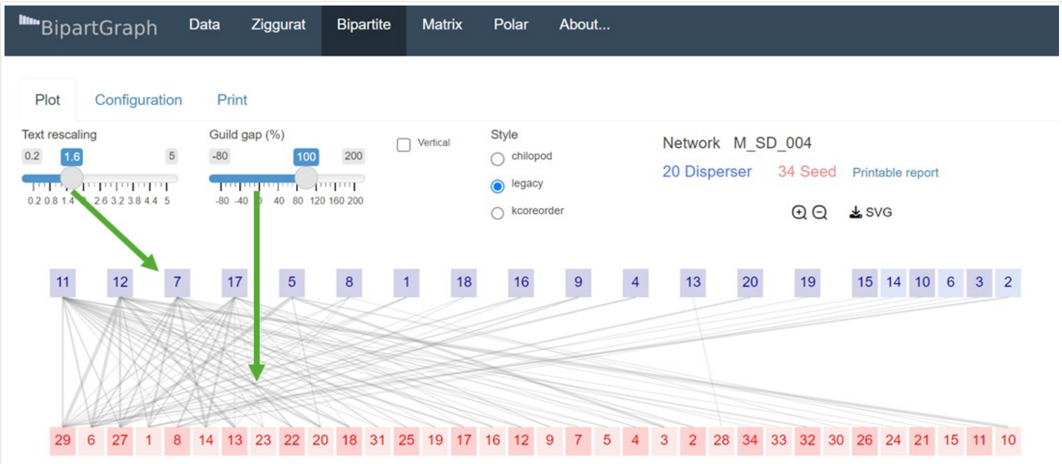


Figure 24: Legacy bipartite plot of disperser network M_SD_004

There are some details that we can deduce at a glance. The more generalist species are those that lay on the left side of the plot and the pair disperser 13/plant 28 is disconnected of the giant component of the network. We can get further visual information tunning input parameters.



Now, it is easy to spot the different k-shells, with species of 4-shell being the most interconnected.

On their left, we find 1-shell, on their right, 3-shell, 2-shell and finally, species disconnected of the giant component. The only difference between the two styles is the order of the species, but information is still concentrated in the two lines of nodes.

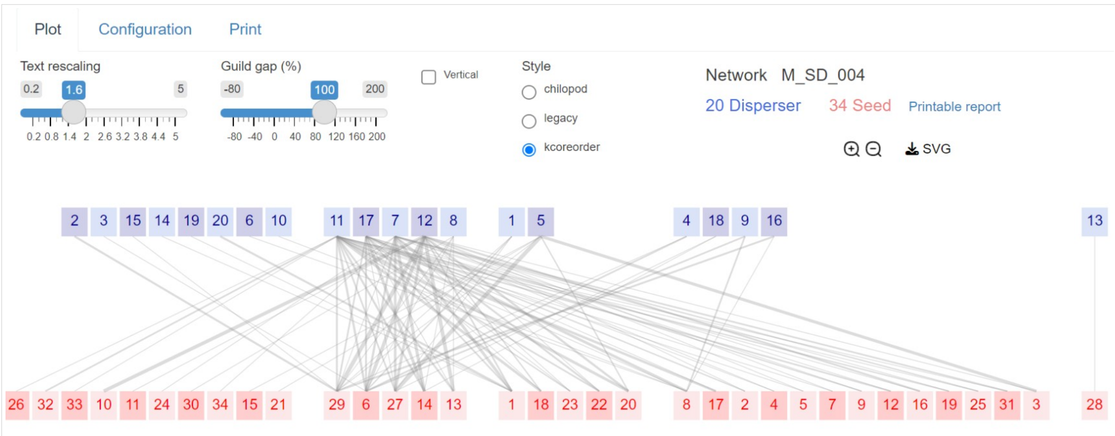


Figure 25: kcoreorder bipartite plot of disperser network M_SD_004

What if we could keep the whole idea of bipartite plot but spreading part of the visual information in two dimensions? That is the idea behind the third style, the chillopod plot. The name refers to the resemblance of this plot with a centipede. Nodes of 1-shell that we call tails in the ziggurat plot are mived away from the guild lines, and appear as short appendices. We apply the same grouping idea that in ziggurat's fat tails, nodes that has just one link and are tied to the same species of a higher shell, are plotted together.

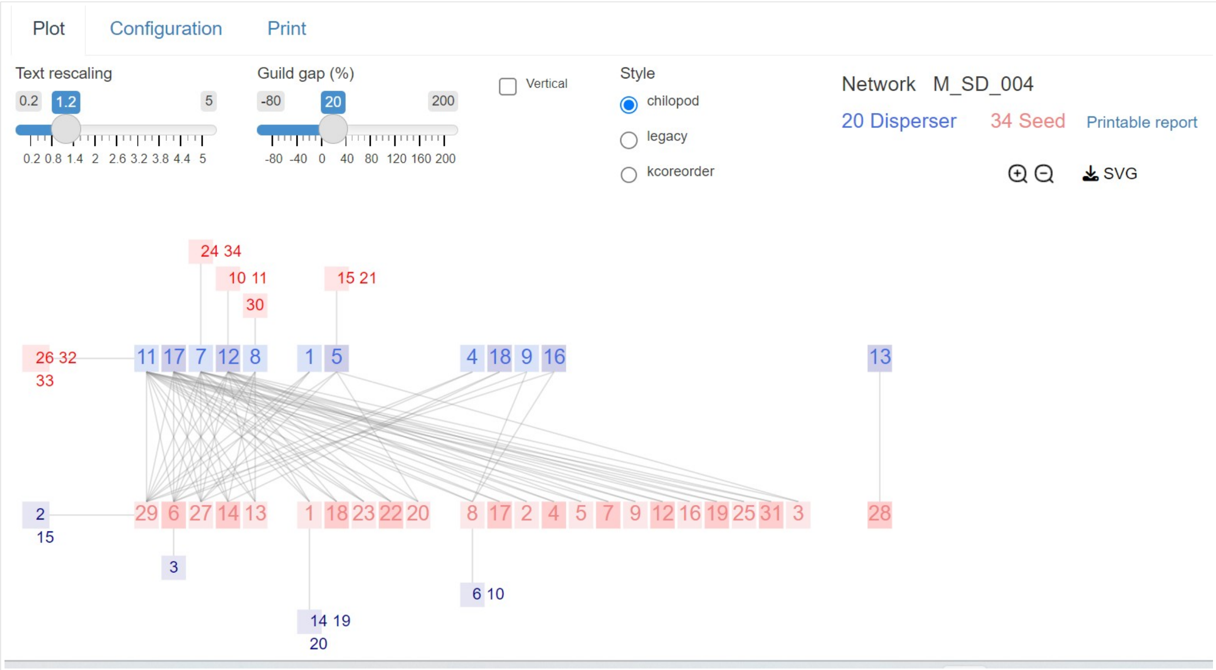


Figure 26: kcoreorder bipartite plot of disperser network M_SD_004

7 The matrix plot

The Matrix Plot is another traditional way of plotting bipartite networks. It is just the depiction of the interaction matrix, with species of guild A as columns and species of guild B as rows. For binary matrixes, the cell i, j is filled with solid color if there is a link between species i of guild A and species j of guild B. For weighted networks, the interaction strength is usually encoded as a color gradient. Nodes are sorted by their degree in descending order. This kind of plot is simple, and is useful to spot spatial properties as nestedness, but it is pretty hard to discover chains of specialists.

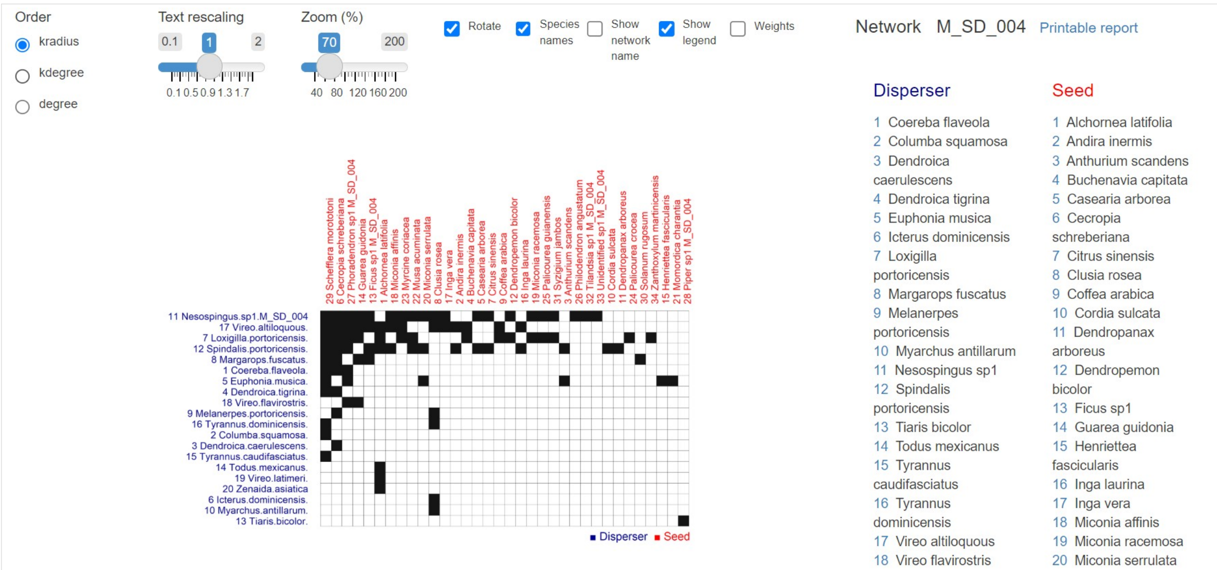
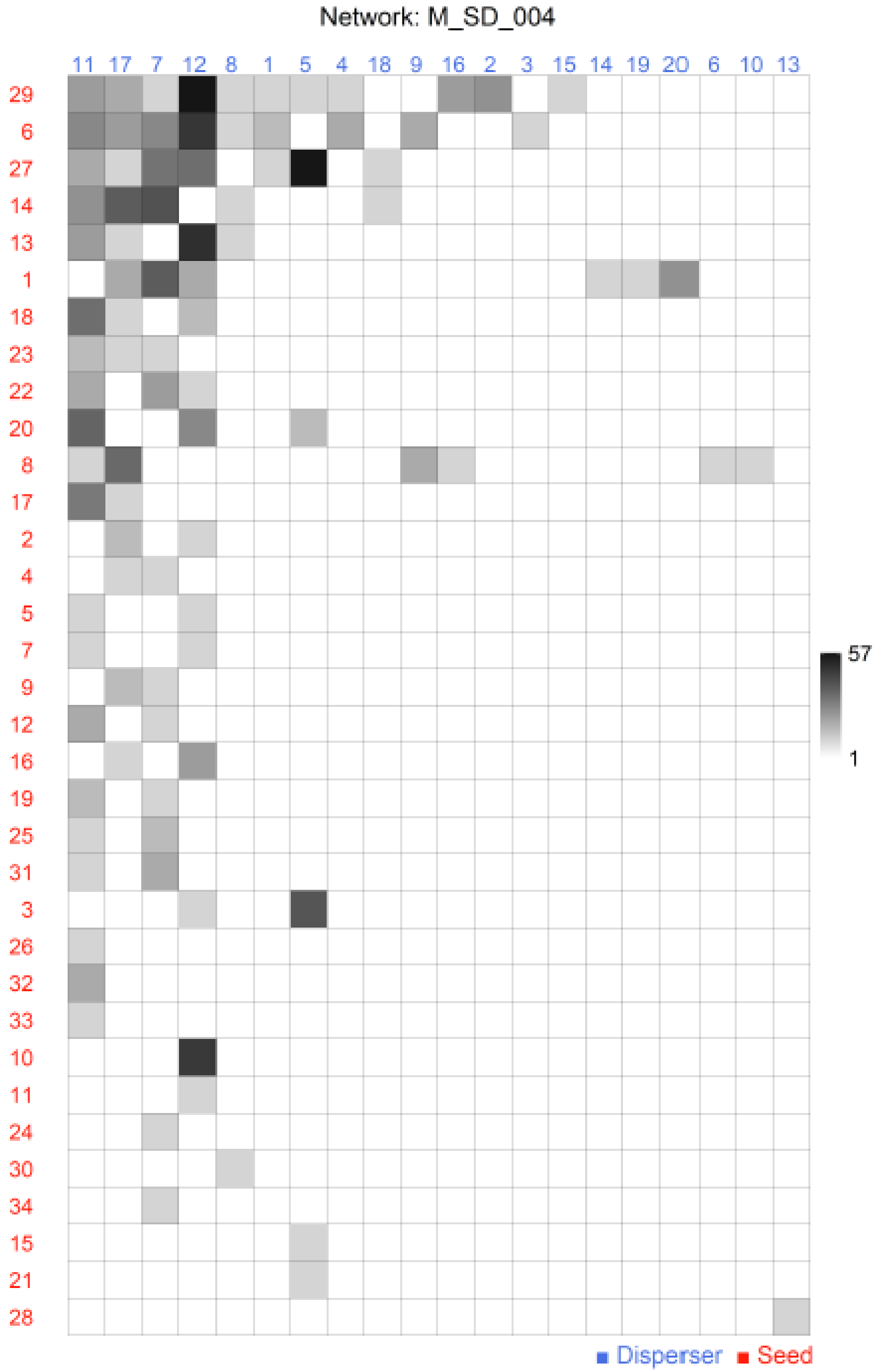


Figure 27: Matrix plot of disperser network M_SD_004

The function allows to flip the matrix to show it in landscape layout for convenience, as in this example. Besides the traditional order by degree, kcorebip offers order by kradius and kdegree. Species names can be removed, and this feature is of special interest when the network is big.


Network: M_SD_004
11 17 7 12 8 1 5 4 18 9 16 2 3 15 14 19 20 6 10 13
29 6 27 14 13 1 18 23 22 20 8 17 2 4 5 7 9 12 16 19 25 31 3 26 32 33 10 11 24 30 34 15 21 28
57
1
Disperser
Seed

Useful hacks

In your `bipartgraph` directory there is a file called `CONFIG.txt` with the following contents:

```
LANGUAGE;PORT;LabelA;LabelB;ColorGuildA1;ColorGuildA2;ColorGuildB1;ColorGuildB2

"en";8080;"Plant";"Pollinator";"#4169E1";"#00008B";"#F08080";"#FF0000"
```

You may change the default language, application TCP port, labels of guilds and colors. It would be nice if you make a copy before editing it. In case you delete by accident this file, the application starts with these default values but you can't change them.

References

[1] GARCÍA-ALGARRA J, PASTOR JM, IRIONDO JM, GALEANO J., 2017. Ranking of critical species to preserve the functionality of mutualistic networks using the k-core decomposition. PeerJ 5:e3321 https://doi.org/10.7717/peerj.3321

[2] GARCÍA-ALGARRA J, PASTOR JM, MOURONTE ML, GALEANO J, 2018. A structural approach to disentangle the visualization of bipartite biological networks. Complexity, 2018. https://www.hindawi.com/journals/complexity/2018/6204947/

[3] BURKLE, Laura A.; MARLIN, John C.; KNIGHT, Tiffany M. Plant-pollinator interactions over 120 years: loss of species, co-occurrence, and function. Science, 2013, vol. 339, no 6127, p. 1611-1615.

[4] BASCOMPTE, Jordi. Disentangling the web of life. Science, 2009, vol. 325, no 5939, p. 416-419.

[5] GALETTI, M. and PIZO, MA., 1996. Fruit eating birds in a forest fragment in southeastern Brazil. Ararajuba, vol. 4, no. 2, p. 71-79.

[6] INGVERSEN, T. T., 2006. Plant-pollinator interactions on Jamaica and Dominica: The centrality, asymmetry and modularity of networks. Master's thesis, Univeristy of Aarhus, Denmark.

[7] HADFIELD, Jarrod D., et al. A tale of two phylogenies: comparative analyses of ecological interactions. The American Naturalist, 2013, vol. 183, no 2, p. 174-187

[8] ALVAREZ-HAMELIN, J. Ignacio, et al. Large scale networks fingerprinting and visualization using the k-core decomposition. Advances in neural information processing systems, 2006, vol. 18, p. 41.