

The HiveR Package (v 0.1-1)

Bryan A. Hanson

DePauw University
Department of Chemistry & Biochemistry
Greencastle Indiana USA

e-mail: hanson@depauw.edu

github.com/bryanhanson/HiveR

October 22, 2011

This document describes some features of the HiveR package including current features and future plans. The current release contains what I believe to be the core set of needed functions and I believe they are robust, but time will tell. Many more features remain to be added; see the sections below. Details on functions can be found by typing `?HiveR` in the R console after installing HiveR.

1 Inspiration and Motivation

HiveR was inspired by the concept of hive plots as developed by Martin Krzywinski at the Genome Science Center (www.hiveplots.net). Hive plots are reaction to "hair ball" style networks in which the layout of the network is somewhat arbitrary. The key innovation in a hive plot is that nodes are assigned to axes based upon some parameter, and the position of the node along the axis is also a function of a parameter. "Parameters" as used here refers to either qualitative or quantitative, but reproducible, characteristics relevant to the system at hand. In a hive plot edges are drawn in a pretty standard way, but may be colored or have a width/weight which encodes an interesting value. Overall, in a hive plot properties of a system can be assigned as follows. The mapping is limited only by one's creativity.

- The axis to which a node is assigned.
- The position/radius of a node.
- Color of a node.
- Size of a node.
- Color of an edge.
- Width/weight of an edge.

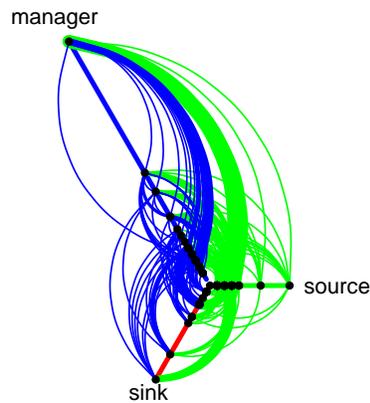
Inspired by the examples given by Krzywinski in his materials on the web, I created the R package `FuncMap` in December 2010. This single function package maps the function calls made by an R package into 3 types: sources, which are functions that make only outgoing calls, sinks, which take only incoming calls, and managers, which do both. Figure 1 shows an example of such a plot which is basically a type of hive plot.

HiveR takes things a step further. HiveR is intended as an implementation of hive plots in R, not a port of `linnet` (Krzywinski's program that actually draws hive plots). As such, it does some things differently, and not all features are implemented (and they may or may not be in the future). The key innovation in HiveR is that for 4 to 6 axes, the plots are 3D and interactive. These 3D plots were inspired by VSEPR theory in chemistry in that the framework of these 3D plots are tetrahedral, trigonal bipyramidal or octahedral respectively.

Figure 1: FuncMap for package lattice

Hive Plot Function Map of lattice Package

142 functions total; 32 are stand alone



position along axis is count of total calls

2 HiveR Features

2.1 Internal Storage

HiveR stores the information needed to create a hive plot in a `HivePlotData` object which is an S3 class. Utilities are provided to summarize the contents of these objects and to check their integrity (functions `sumHPD` and `chkHPD` respectively).

2.2 Generation of Random Network Data Sets

HiveR has the ability to generate random network data sets with between 2 and 6 axes, using function `ranHiveData`. These are useful for testing and demonstration purposes and will be used in the examples below.

2.3 Import of Real Data Sets

Work has begun on a largely automated means of importing real data sets in `.dot` format. Look for this feature in a future release.

2.4 Making Hive Plots

Figures 2 and 3 show hive plots of random networks using absolute scaling. Note that for 4 to 6 axes, plots are interactive and cannot be shown here. See the help page for `plot3dHive` for an example you can run. Figure 4 shows the 3 axis example with the nodes displayed by rank.

Figure 2: A randomly generated hive plot with 2 axes

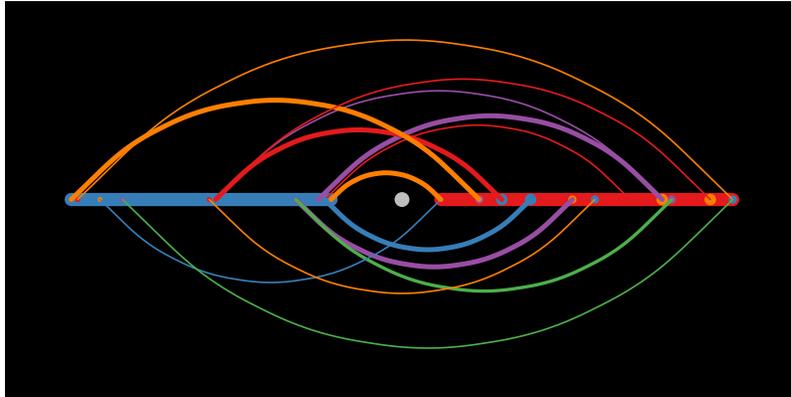


Figure 3: A randomly generated hive plot with 3 axes

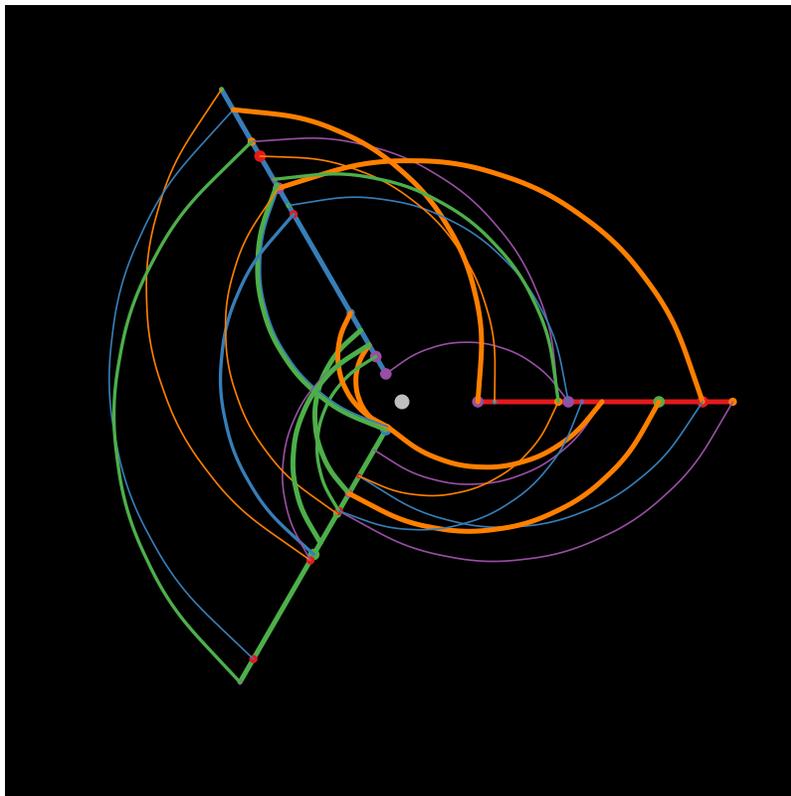
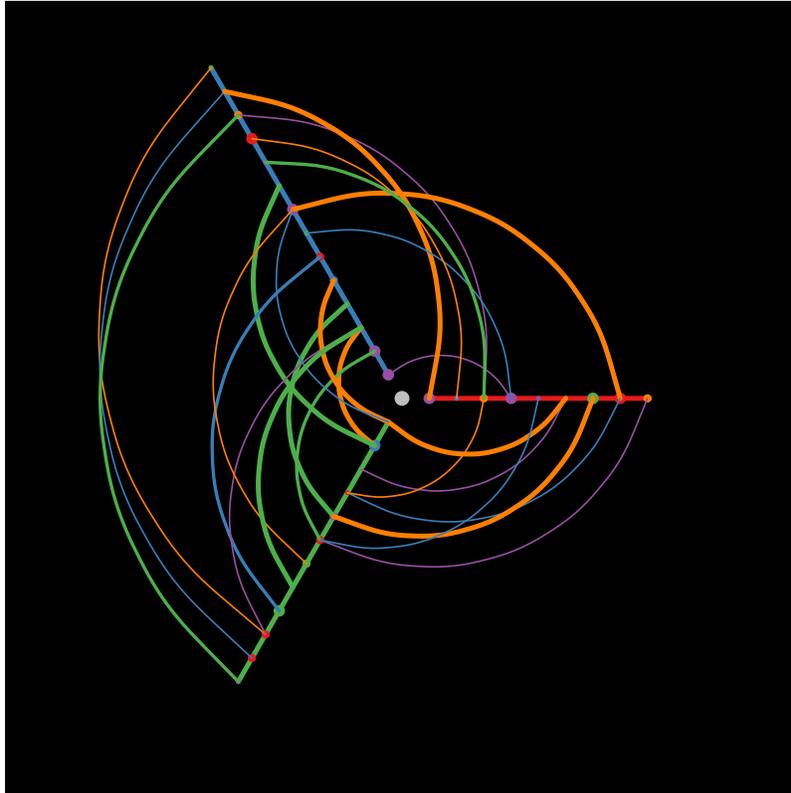


Figure 4: A randomly generated hive plot with 3 axes (nodes by rank)



3 A Simple Demonstration Using a Plant-Pollinator Network

HiveR currently contains two built-in data sets, `Safari` and `Arroyo`.¹ These are plant-pollinator data sets which were derived from Vasquez and Simberloff, 2003 [1]. These are two-trophic level systems that have almost exactly the same plants and pollinators. `Safari` is from an undisturbed area, while `Arroyo` is from a nearby location grazed by cattle. In these data sets, plants are on one axis, and pollinators are on the other. Each organism was assigned a radius on the axis based by calculating d' using function `dfun` in package `bipartite`. d' is an index of specialization; higher values mean the plant or pollinator is more specialized. Edge weights were assigned proportional to the square root of the normalized number of visits of a pollinator to a plant. Thus the width of the edge drawn is an indication of the visitation rate. The number of visits was divided manually into 4 groups and used to assign edge colors ranging from white to red. The redder colors represent greater numbers of visits, and the color-coding is comparable for each data set. Thus both the edge color and the edge weight encode the same information. It would of course be possible to encode an additional variables by changing either edge color or weight, or node size.

Figures 5 and 6 are two means of plotting `Safari` using package `bipartite`. Figures 7 and 8 show `Safari` and `Arroyo` respectively, using `plotHive`. These plots show a rich amount of information not available from the more standard plots. It is very clear that the network connections are fundamentally different:

- The degree of specialization with each network is different, with many more visitors between specialized plants and pollinators in `Safari`.
- There are more plants in `Arroyo` (plants are on the short axis).
- The huge number of visits encoded in red in `Safari` (the ungrazed site) is missing in `Arroyo`.

¹Be warned: I am not an ecologist and these data sets and plots are merely an example

Figure 5: Safariland data set using visweb

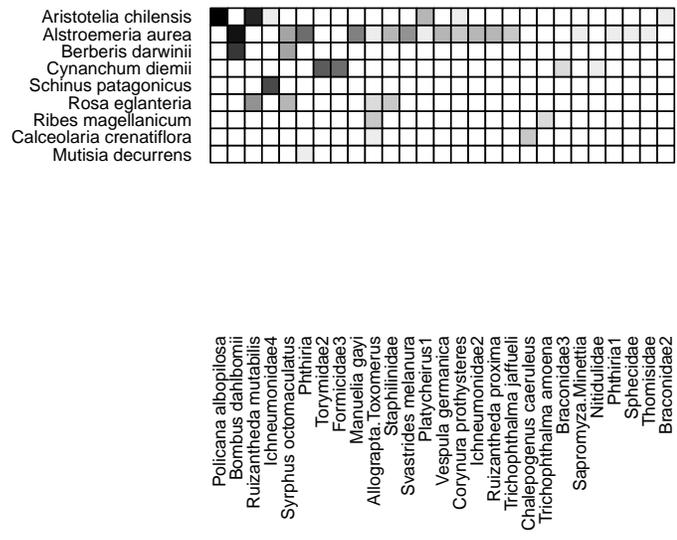


Figure 6: Safariland data set using plotweb

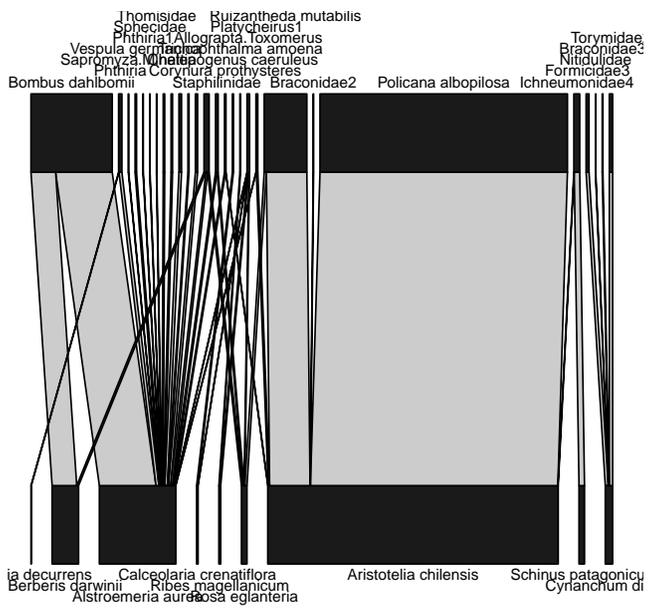


Figure 7: Safari data set using plotHive

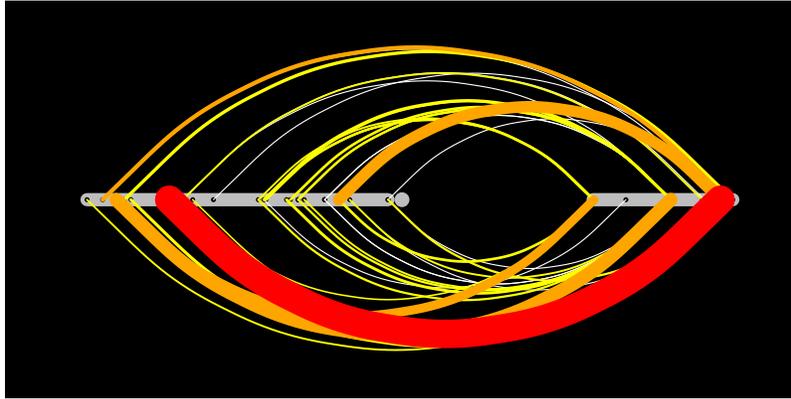
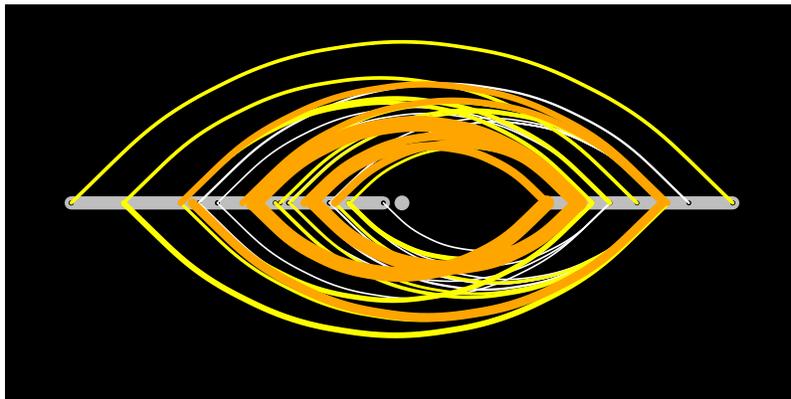


Figure 8: Arroyo data set using plotHive



4 Comparison to linnet

linnet (for linear networks) is the program written by Krzywinski that draws hive plots. Here are some notes about how HiveR compares to linnet.

1. To show more information, in linnet one can clone an axis. In HiveR, the same notion exists but rather than a clone of an existing axis, one simply adds a new axis based upon some property of the system.
2. No segmentation of an axis is currently possible with HiveR.

5 Things to Do

1. Add magnification of an axis. Use `manipAxis`.
2. Add inversion of an axis. Use `manipAxis`.
3. Tasks Related to Input of Real Data
 - (a) Need a parser for `.dot` format (in progress).
 - (b) Write `map2Axis` which will assign nodes to a position on a particular axis and make good configuration choices.
 - (c) Find a medium sized real data set.
 - (d) Bring in the *E. coli* data set.
4. Work on speed: for 3D hive plots, drawing many edges can be prohibitively slow. Testing has shown that the problem is not `rcsr`, so need to look other places when profiling the code.

6 Features to Add Long Term

1. Hive Panels: set up a GUI or other display mechanism which can display multiple views of the same hive from different perspectives, or different hives from the same perspective.
2. The current 3D spline calculation produces an asymmetric spline. It could be made more symmetric.
3. Add the ability to subtract 2 hive plots and display the result.
4. Set up animations. Perhaps include the possibility of running two animations of related hives side by side.
5. Set up a mechanism to automatically permute the axes when $n_x = 5$ or 6 so that the best option can be selected.
6. Set up mouse controls.

7 Notes, Technical and Other

1. Hive plots are not directed graphs

References

- [1] D. P. Vazquez and D. Simberloff, "Changes in interaction biodiversity induced by an introduced ungulate," *Ecology Letters*, vol. 6, pp. 1077–1083, 2003.