

# HiveR: 2D & 3D Hive Plots + Hive Panels

## New Tools for Network Visualization



DEPAUW  
UNIVERSITY

*Est. 1837*



**Bryan A. Hanson**

DePauw University

Dept of Chemistry & Biochemistry

Greencastle Indiana USA

e-mail: [hanson@depauw.edu](mailto:hanson@depauw.edu)

[github.com/bryanhanson/HiveR](https://github.com/bryanhanson/HiveR)

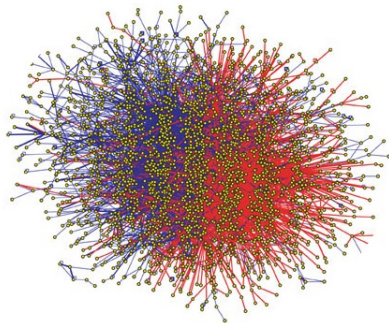
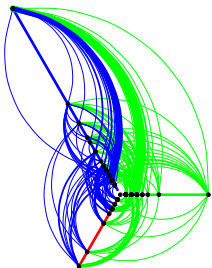
[CRAN.R-project.org/package=HiveR](https://CRAN.R-project.org/package=HiveR)

*Special Thanks to  
Martin Krzywinski!*

*Powered by knitr*

# Inspiration & Motivation

Developed by Martin Krzywinski at the Genome Sciences Center ([www.hiveplot.com](http://www.hiveplot.com))  
Krzywinski et. al. *Briefings in Bioinformatics* doi:10.1093/bib/bbr069 (2011)



Rual et. al. *Nature* vol 437 pg 1173 (2005)

- ▶ "Hairball" style networks  $\neq$  reproducible research
- ▶ Hive Plots use a node coordinate system

## Application Areas

- ▶ Ecology: food webs
- ▶ Social networks
- ▶ Systems biology: protein-protein interactions, gene regulatory networks, genomic architecture
- ▶ Biochemistry: metabolic pathways
- ▶ Computer science
- ▶ In the spirit of Cleveland & Tufte. . .

## Characteristics of Hive Plots

- ▶ Hive Plots are transparent:
  - ▶ Hive Plots are **Rational**: the layout is determined only by the structural properties of the network (no algorithm)
  - ▶ Hive Plots are **Predictable & Reproducible**: network features are mapped to plot features
  - ▶ Hive Plots are **Robust** to changes in the network
- ▶ Hive Plots are practical:
  - ▶ Hive Plots are **Flexible** & can be tuned to show interesting features
  - ▶ **Complexity Scales Well** in a Hive Plot & details can be inspected
  - ▶ Hive Plots of different **Networks can be Directly Compared**
- ▶ Most powerful with large networks, where hidden patterns emerge



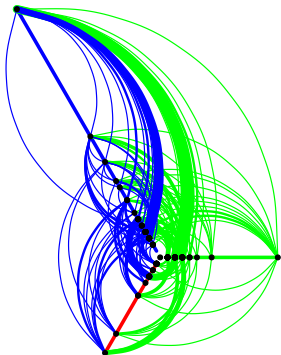
# Early Implementation: FuncMap

December 2010

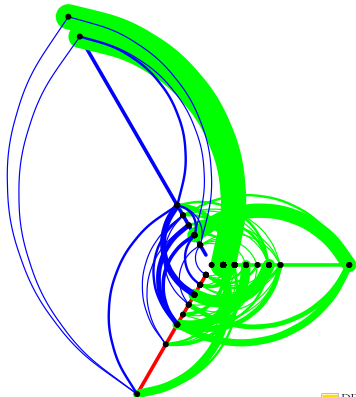
- ▶ Function calls in R packages are divided into 3 types:
  - ▶ Sources, which are functions that only call other functions
  - ▶ Sinks, which are functions that only get called
  - ▶ Managers, which do both
  - ▶ Info from `foodweb` in package `mvbutils`
- ▶ Mapping:
  - ▶ Functions in a package are assigned to an axis by their role
  - ▶ Radius is determined by the number of calls made or received by a function
  - ▶ Width of the edges is determined the same way

# Early Implementation: FuncMap

lattice

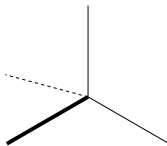


ggplot2

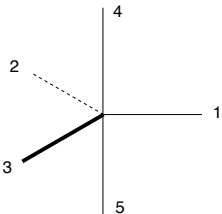


## What Exactly is HiveR?

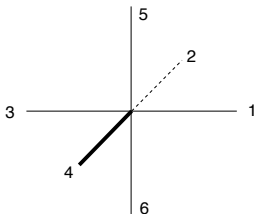
- ▶ HiveR is an implementation of Hive Plots in *R*, not a port of the original *Perl* prototype
- ▶ There are also versions in *Java* & *D3* by others
- ▶ Value added: 3D plots were inspired by the ideas of VSEPR theory in chemistry (more later)



tetrahedral geometry



trigonal bipyramidal geometry



octahedral geometry

## Hive Plot Features Which Can Be Mapped

Axis to which a node is assigned
Radius of a node
Color of a node
Size of a node
Color of an edge
Width of an edge

- ▶ Node assignment can be based upon qualitative or quantitative characteristics:
  - ▶ *1st & hardest task: no jumping or crossing axes allowed*
- ▶ Mapping is limited only by one's creativity & the particular knowledge domain
- ▶ Mapping can be readily tuned
- ▶ Mapping network parameters in this way results in a reproducible plot

## Hive Plots: Axis Units/Scaling Options

After assignment, the 2nd aspect of the node coordinate system

<i>method</i>	<i>axis length</i>	<i>center hole</i>	<i>node behavior</i>
native	$f(\text{node units})$	asymmetric	nodes may overlap
ranked	$\propto \text{rank}(\text{nodes})$	circular	nodes evenly spaced & don't overlap
normed	all equal	circular	nodes may overlap
ranked & normed	all equal	circular	nodes evenly spaced & don't overlap

## HiveR Utilities

- ▶ Generation of random networks (`ranHiveData`)
- ▶ Import data (`dot2HPD`, `adj2HPD`)
- ▶ Extract embedded information (`mineHPD`)
  - ▶ node degree → node radius
  - ▶ node role (source, manager, sink) → node axis
  - ▶ delete orphaned nodes, zero-length edges
- ▶ Scale or invert an axis (`manipAxis`)
- ▶ Check integrity of the HPD (`chkHPD`)
- ▶ Summarize HPD (`sumHPD`)
  - ▶ Find orphaned nodes
  - ▶ Check for edges that jump axes
  - ▶ Check for edges that start & end on the same axis
  - ▶ Report a list of edges to be drawn

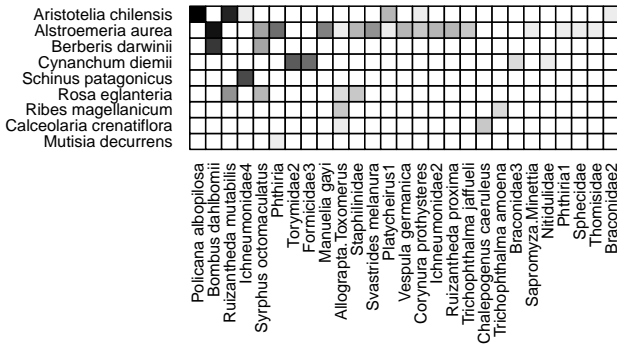
## A Simple Example Using a Plant-Pollinator Network

- ▶ Data set `Safariland`<sup>1</sup> from package `bipartite`
- ▶ Describes plant-pollinator pairs & the number of visits during a fixed observation period
- ▶ Observations at several different habitats/sites
- ▶ Two sites have been converted to HPD format:
  - ▶ `Safari` from an undisturbed area
  - ▶ `Arroyo` from an area grazed by cattle

---

<sup>1</sup>Vazquez & Simberloff, *Ecology Letters* vol 6 pg 1077 (2003)

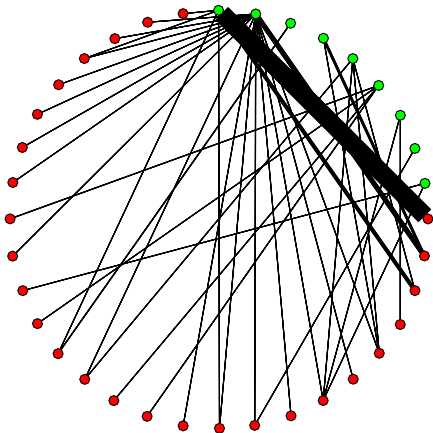
# Plant-Pollinator Network: Function visweb in Package bipartite





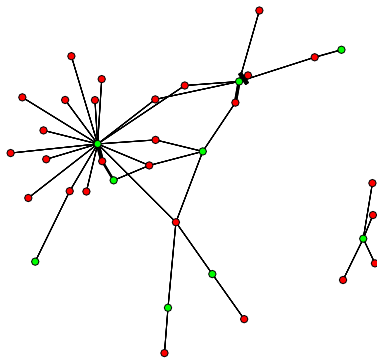
# Plant-Pollinator Network: Function `gplot` in Package `sna`

Plant nodes green, insect nodes red, `mode = circle`



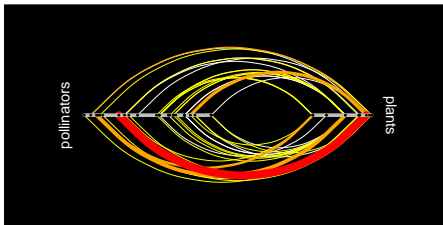
# Plant-Pollinator Network: Function `gplot` in Package `sna`

Plant nodes green, insect nodes red, `mode = fruchtermanreingold`

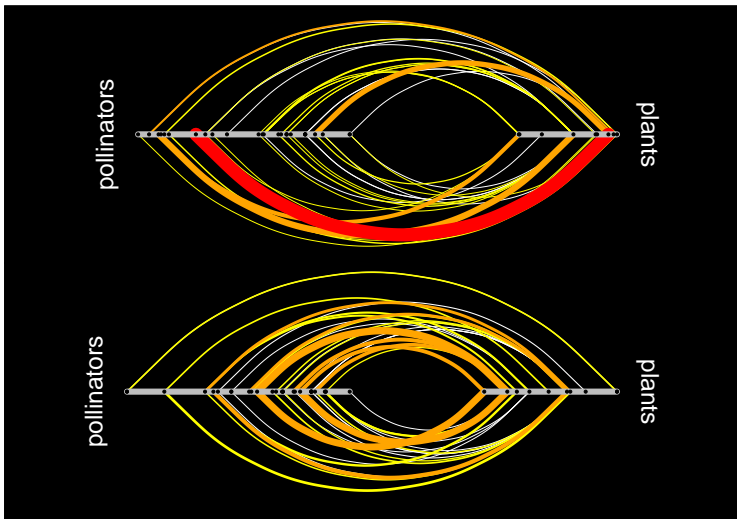


## Plant-Pollinator Network: 2D Hive Plot

- ▶ Plants vs. pollinators
- ▶ Node radius is  $|d'|$
- ▶ Edge weights are  $\propto \sqrt{\text{no. visits}}$
- ▶ Edge colors like weights; redder = more visits



# Safari (undisturbed) vs. Arroyo (grazed): Hive Panel



## Plant-Pollinator Network: 2D Hive Plot

- ▶ The degree of specialization in each network is different
- ▶ A greater number of visits (wider, redder edges) occur between more specialized species (nodes at larger radii) in Safari than Arroyo
- ▶ Plants in Arroyo have a larger range of specialization: the plant axis is longer
- ▶ The huge number of visits encoded in red in Safari (the ungrazed site) is missing in Arroyo, which was an interesting finding in the study
- ▶ Note that deleting nodes & edges does not affect those remaining

## Nuances of Hive Plots

- ▶ Hive Plots are radially-arranged parallel coordinate plots
- ▶ Assigning the nodes is the most time-consuming task (no jumping/crossing of axes)
- ▶ Nodes cannot be assigned w/o thinking about the edges as well (next slide)
- ▶ Not all data sets can be made into Hive Plots
- ▶ Hive Plots are directionally agnostic. Almost.
- ▶ If drawn using native or normed coordinates, nodes may overlap. The nodes "on top" & showing are the last drawn nodes. You may wish to sort the nodes before drawing to get a certain effect – the same is true for edges
- ▶ Only deals with one network at a time

## Nuances of Hive Plots

As with any parallel coordinate plot, the order of axes has a dramatic effect

### ▶ 2D Hive Plots

- ▶ With 2 or 3 axes, edges cannot jump axes. For 4+ axes, you must guard against this:
- ▶ Edges should go  $1 \rightarrow 2$ ,  $2 \rightarrow 3$ ,  $\dots$   $5 \rightarrow 6$ , but not  $1 \rightarrow 5$

### ▶ 3D Hive Plots

- ▶ No edges can start & end on the same axis
- ▶ For 5 or 6 axes, edges may not start on one axis & end on a co-linear axis
- ▶ For 4 axes, edges cannot jump axes. For 5 or 6 axes, you must guard against this.

## A More Complex Example: *E. coli* Gene Regulatory Network<sup>2</sup>

- ▶ Raw data is composed of genes that regulate each other via transcription factors
- ▶ Steps to Create a Hive Plot:
  1. Using dot2HPD & helper files, create the initial HPD
  2. Edges colored by role (activator, repressor, dual) during import
  3. Assign node radius based upon degree
  4. Assign axis based upon source, sink, manager role
  5. Remove zero-length edges
  6. Sort edges

---

<sup>2</sup>Yan *et. al.* *PNAS* vol 107 pg 9186 (2010), based upon RegulonDB



## *E. coli* Network: Sample Dot File

Only a very small portion of the Dot standard is followed at this time

```
## [1] "zrar [label=nonpersistent];"  
## [2] "zras [label=nonpersistent];"  
## [3] "zwf [label=nonpersistent];"  
## [4] "arca -- phantom_gene [type=0];"  
## [5] "crp -- phantom_gene [type=0];"  
## [6] "fnr -- phantom_gene [type=0];"  
## [7] "fur -- phantom_gene [type=0];"  
## [8] "arca -- acea [type=1];"  
## [9] "crp -- acea [type=0];"
```

# Importing Dot Files: Sample Mapping Instructions

## Node Mapping Instructions

dot.tag	dot.val	hive.tag	hive.val
label	persistent	color	red
label	nonpersistent	color	black

## Edge Mapping Instructions (used in next example)

dot.tag	dot.val	hive.tag	hive.val
interaction	repressor	color	red
interaction	activator	color	green
interaction	dual	color	orange

## *E. coli* Network: Create the Initial HPD

The edge mapping instructions color the edges in this process

```
sumHPD(EC1)
```

```
## E coli gene regulatory network (RegulonDB)  
## This hive plot data set contains 1597 nodes on 1 axes and 3893 edges.  
## It is a 2D data set.  
##  
## Axis 1 has 1597 nodes spanning radii from 1 to 1  
##  
## Axes 1 and 1 share 3893 edges  
##
```

## *E. coli* Network: Assign Node Degree to Radius

```
EC2 <- mineHPD(EC1, option = "rad <- tot.edge.count")
sumHPD(EC2)

## E coli gene regulatory network (RegulonDB)
## This hive plot data set contains 1597 nodes on 1 axes and 3893 edges.
## It is a 2D data set.
##
## Axis 1 has 1597 nodes spanning radii from 1 to 434
##
## Axes 1 and 1 share 3893 edges
##
```

## *E. coli* Network: Assign Nodes to Axis

```
EC3 <- mineHPD(EC2, option = "axis <- source.man.sink")
sumHPD(EC3)

## E coli gene regulatory network (RegulonDB)
## This hive plot data set contains 1597 nodes on 3 axes and 3893 edges.
## It is a 2D data set.
##
## Axis 1 has 45 nodes spanning radii from 1 to 83
## Axis 2 has 1416 nodes spanning radii from 1 to 11
## Axis 3 has 136 nodes spanning radii from 2 to 434
##
## Axes 1 and 2 share 400 edges
## Axes 1 and 3 share 21 edges
## Axes 3 and 2 share 3158 edges
## Axes 3 and 3 share 314 edges
##
```

## *E. coli* Network: Clean Up Some Problems

```
EC4 <- mineHPD(EC3, option = "remove zero edge")

##
## 125 edges that start and end on the same point were removed

sumHPD(EC4)

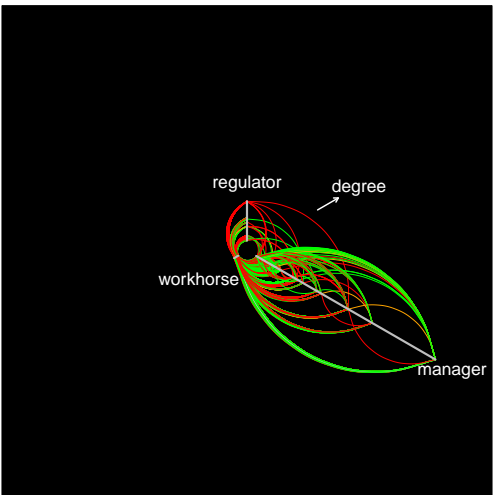
## E coli gene regulatory network (RegulonDB)
## This hive plot data set contains 1597 nodes on 3 axes and 3768 edges.
## It is a 2D data set.
##
## Axis 1 has 45 nodes spanning radii from 1 to 83
## Axis 2 has 1416 nodes spanning radii from 1 to 11
## Axis 3 has 136 nodes spanning radii from 2 to 434
##
## Axes 1 and 2 share 400 edges
## Axes 1 and 3 share 21 edges
## Axes 3 and 2 share 3158 edges
## Axes 3 and 3 share 189 edges
##
```

## *E. coli* Network: Interpreting the Hive Plot

- ▶ Nodes to axes according to role: regulator, workhorse, or manager
- ▶ Node radius = edge count/degree
- ▶ Edges colored according to role:
  - ▶ repressors: red
  - ▶ activators: green
  - ▶ dual function: orange

# *E. coli* Gene Regulatory Network

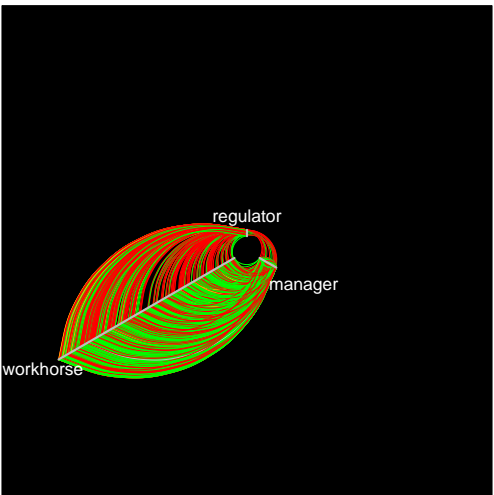
Axes plotted using native units





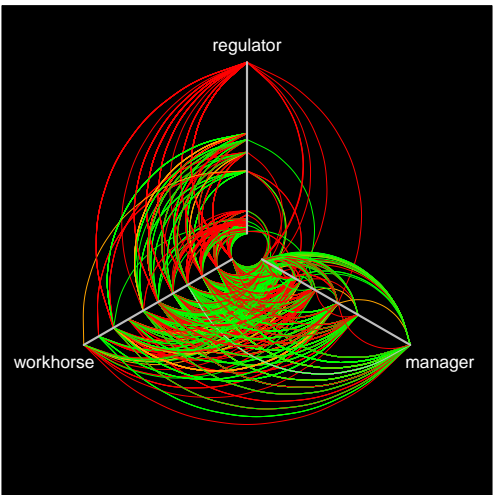
# *E. coli* Gene Regulatory Network

Axes plotted using ranked units



# *E. coli* Gene Regulatory Network

Axes plotted using normalized units

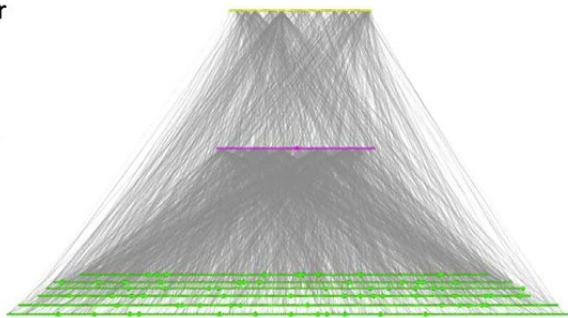


# *E. coli* Gene Regulatory Network: Compare to Yan *et. al.*

master regulator

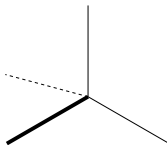
middle manager

workhorse

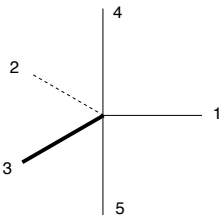


## 3D Hive Plots

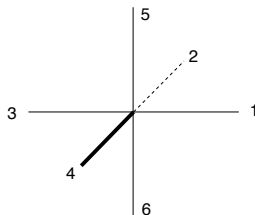
- ▶ Interactive using `rgl` graphics
- ▶ More adjacent axes than for 2D Hive Plots
  - ▶ Tetrahedron: 8 adjacent axis pairs, *crossings impossible*
  - ▶ Trigonal bipyramid: 9 adjacent axis pairs
  - ▶ Octahedron: 12 adjacent axis pairs
- ▶ Other approaches to 3D networks...



tetrahedral  
geometry



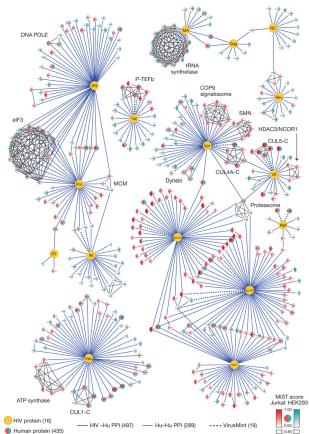
trigonal bipyramidal  
geometry



octahedral  
geometry

# Let's Make a 3D Hive Plot

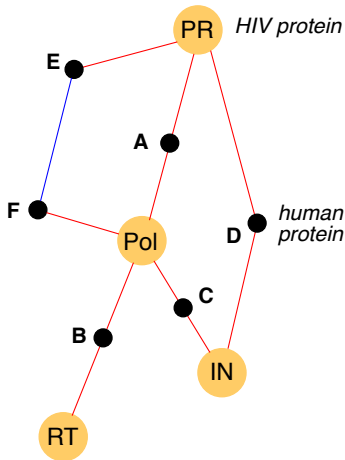
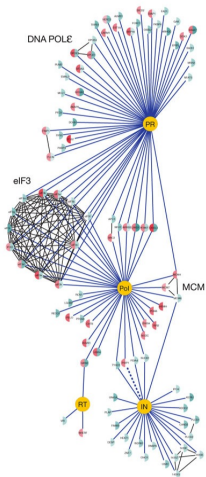
## Global Landscape of HIV-Human Protein Complexes<sup>3</sup>



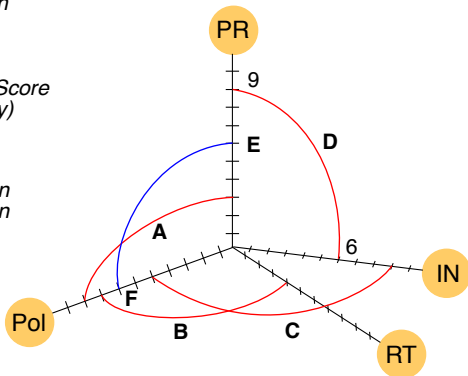
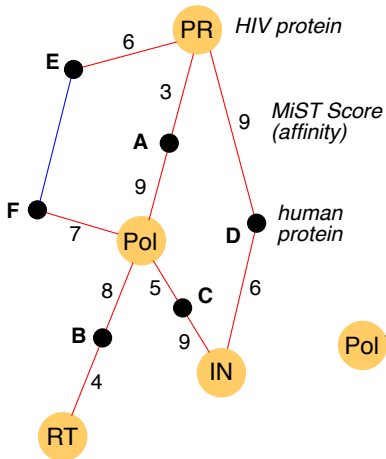
- ▶ Available data:
  - ▶ HIV- human protein interactions
  - ▶ Human-human protein interactions
  - ▶ MiST scores (strength of protein-protein affinity)
  - ▶ x 2 cell lines

<sup>3</sup>Jäger et. al. *Nature* vol 481 pg 365 (2012)

# Focus on One Subnet & Conceptualize the Mapping

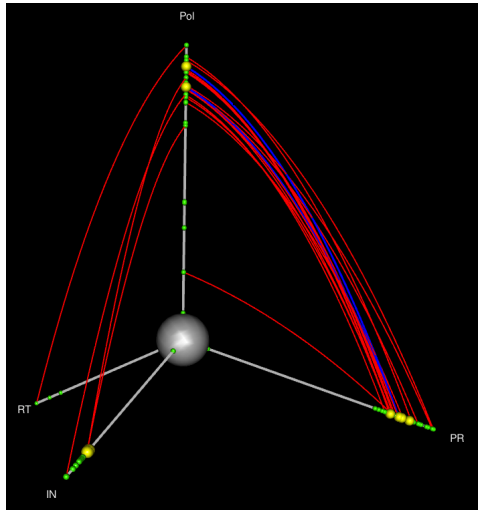


# Details of the Mapping Process



# 3D Hive Plot: Static View

HIV-human interactions in red, radius = MiST score with HEK cells, native units  
HIV → human → human → HIV interactions in blue  
Proteins w/  $\geq 2$  edges shown as larger yellow nodes, otherwise green



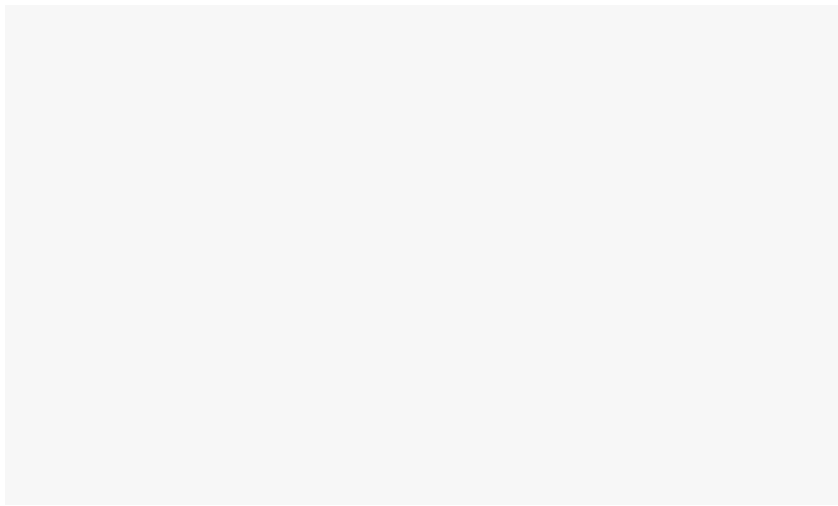


# 3D Hive Plot

HIV-human interactions in red, radius = MiST score with HEK cells, native units

HIV → human → human → HIV interactions in blue

Proteins w/  $\geq 2$  edges shown as larger yellow nodes, otherwise green



## Possible Extensions (the *To-Do* list)?

- ▶ Subtract 2 hive plots & display the result
- ▶ Automatically permute the axes in 3D mode with 5 or 6 axes so that the best option can be selected
- ▶ Hovering brings up node or edge label (may need to go to a different graphics system)
- ▶ More ways to import/mine (better Dot compliance)
- ▶ Change splines to bezier curves
- ▶ What would you like to see?

## HiveR vs. *Perl* Prototype

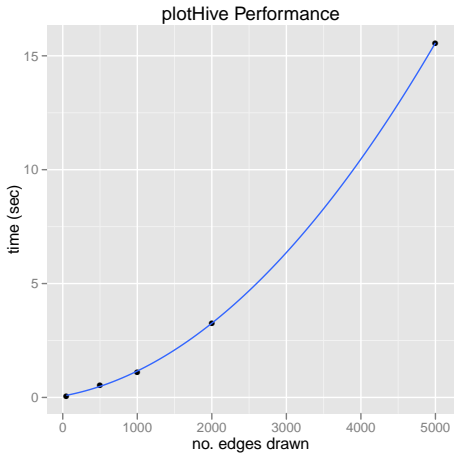
1. In the prototype one can clone an axis to show connections that would start and end on the same axis.
2. In HiveR, one can simply add a new axis based upon some property of the system. Alternatively, for 2D hive plots, HiveR is able to show edges that start & end on the same axis.
3. No segmentation of an axis is currently possible with HiveR
4. The prototype uses bezier curves to create the edges; HiveR uses splines with a single control point

## HivePlotData Objects

\$nodes	\$id	int	Node identifier
	\$lab	chr	Node label
	\$axis	int	Axis
	\$radius	num	Node Radius
	\$size	num	Node size
	\$color	chr	Node color
\$edges	\$id1	int	Starting node id
	\$id2	int	Ending node id
	\$weight	num	Width of edge
	\$color	chr	Edge color
\$type		chr	2D or 3D plot
\$desc		chr	Description
\$axis.cols		chr	Colors for axes
- attr		chr	"HivePlotData"

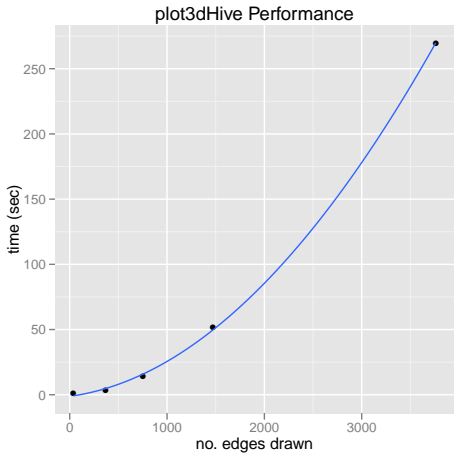
# Performance: 2D Hives Using grid Graphics

MacBook Pro running OSX 10.7.4 using 8 GB RAM & an Intel i7 chip at 2 GHz



# Performance: 3D Hives Using rg1 Graphics

MacBook Pro running OSX 10.7.4 using 8 GB RAM & an Intel i7 chip at 2 GHz



# 3D Spline Curves

