

How to use the PrasathGraph package

August 6, 2014

1 Getting Started

In order to demonstrate the usage of `pathwayHeatmap` and the type of output the function will return, we will look at the “EPHB forward signaling pathway” stored in the Protein Interaction Database (PID).

There are several packages in the Bioconductor repository that allow for the creation of `graphNEL` objects from pathways stored in public databases. In this vignette, we will be using the “`graphite`” package to help us create `graphNEL` object of “EPHB forward signaling pathway”.

```
> library(graphite)
> ephb <- nci[["EPHB forward signaling"]]
> ephb <- convertIdentifiers(ephb, "entrez")
> ephbGraph <- pathwayGraph(ephb)
```

The object 'ephbGraph' is a `graphNEL` object containing the information about the “EPHB forward signaling” pathway. For more information about `graphite`, see “[GRAPH Interaction from pathway Topological Environment](#)”.

2 Visualizing Pathway Heatmaps

At this point, it is possible to plot the pathway using `plot()` (Figure 1.).

```
> plot(ephbGraph, "fdp")
```

Now let's imagine that you had identified a number of proteins to be differentially expressed between two conditions and have the \log_2 fold-change values for each of those proteins. Many of these proteins are involved in the “EPHB forward signaling pathway”. The function `pathwayHeatmap` allows for the creation of a network diagram such as the one in Figure 1 but the proteins are colored according to their \log_2 fold-change values.

In this example we will simply make our own log2 fold-change expression values vector. To correctly map the nodes of the graph to the expression values, the log2 fold-change vector must be a named vector where the names are the node identifiers.

```
> ephb.log2fc <- runif(33,-5,5)
> names(ephb.log2fc) <- nodes(ephbGraph)
```

The final parameter that you need before plotting is a vector of colour values. This vector contains the gradient progression for the heatmap. Here we use the package “RColorBrewer” to create our colour vector.

```
> library(RColorBrewer)
> colours <- rev(colorRampPalette(brewer.pal(6,"RdBu"))(9))
```

Executing the following command will produce a graph similar to Figure 2.

```
> library(PrasathGraph)
> pathwayHeatmap(ephbGraph, layout="fdp", de=ephb.log2fc, cols=colours)
```

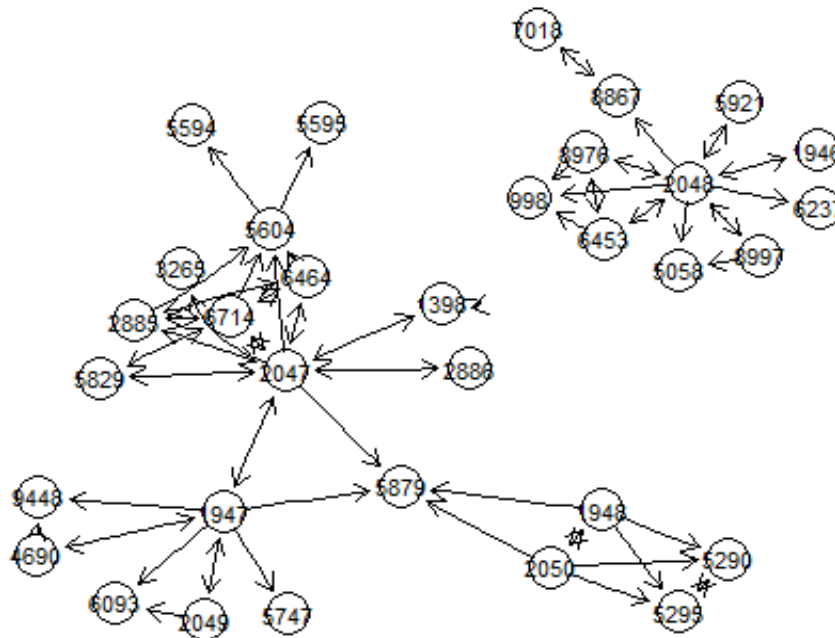


Figure 1. EPHB Forward Signaling Pathway using plot()

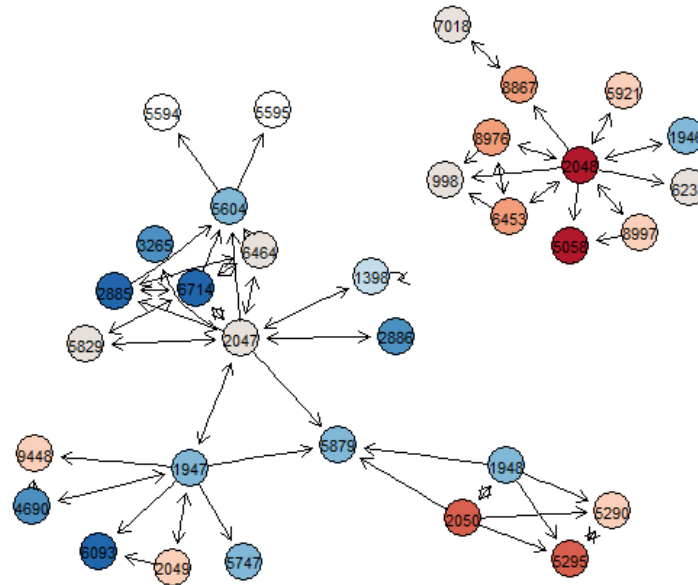


Figure 2. EPHB Forward Signaling Pathway using pathwayHeatmap()

Adjusting Graph Attributes

The “pathwayHeatmap” function is built on top of the plot.graph() function of the “RGraphviz” package. Consequently, “pathwayHeatmap” will pass on user-defined node attributes and global attributes to the plotting function. For more information on manipulation of these attributes, see “[How To Plot A Graph Using Rgraphviz](#)”.

3 Manipulating Pathway Graphs

The functions “upstreamSearcher” and “downstreamSearcher” allow for parsing of a pathway graphNEL object. Given a node within a pathway, “upstreamSearcher” and “downstreamSearcher” will return all the nodes that exist upstream or downstream of said node, respectively.

Here we extract all the nodes that are upstream of the node with the entrezID “2885” in the “EPHB forward signaling pathway”.

```
> upstreamSearcher(ephbGraph, "2885")
```

```
[1] "2885" "6714" "6464" "2047" "5829" "1398" "1947" "2886" "4690" "2049"
```

In conjunction with the “subGraph” function, you can create a new graphNEL object containing only the nodes that are connected upstream of node “2885” in “EPHB forward signaling pathway”.

```
> ephbGraph.sub <- subGraph(upstreamSearcher(ephbGraph,"2885"),ephbGraph)
```

Plotting ephbGraph.sub with pathwayHeatmap produces Figure 3.

```
> pathwayHeatmap(ephbGraph.sub, layout="fdp", de=ephb.log2fc, cols=colours)
```

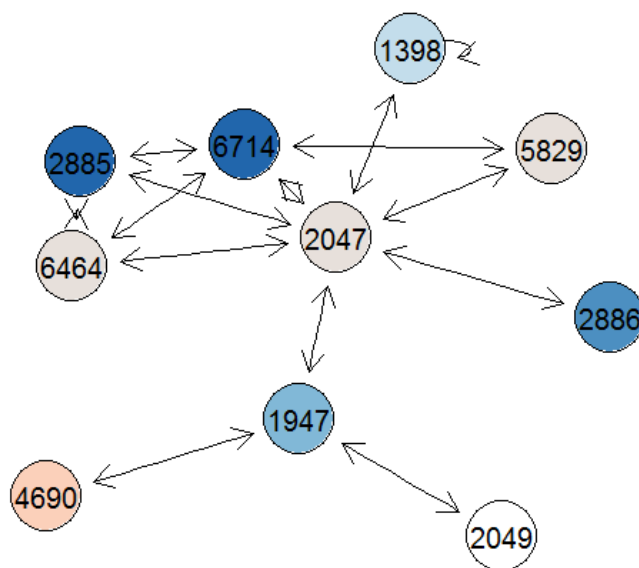


Figure 3. Upstream interactions of node "2885" in EPHB forward signaling pathway.