

paxtoolsR vignette

Julia Gustavsen

Contents

1 Purpose	1
2 Visualize a network using paxtoolsr	2
2.1 Send network to Cytoscape using RCy3	2
3 Pathway Commons Graph Query	4
3.1 Send network to Cytoscape using RCy3	4
4 Create a subnetwork from a set of proteins	7
4.1 Send network to Cytoscape using RCy3	7
5 Adding metadata to Pathway commons networks	9
5.1 Send network to Cytoscape using RCy3	9
References	12

1 Purpose

This vignette shows one way to visualize networks retrieved using paxtoolsr (Luna et al. 2015) in Cytoscape (P. Shannon et al. 2003 ; Ono et al. 2015) using RCy3. Paxtoolsr leverages tools developed in the java toolkit paxtools to retrieve information from Pathway commons. For example you can retrieve a network or subnetwork related to a particular pathway. For more information please see Luna et al. (2015) and have a look at the paxtoolsr package vignette:

```
library(paxtoolsr)
browseVignettes("paxtoolsr")
```

In this vignette we will visualize a network retrieved from Pathway Commons, query Pathway commons to find the neighbourhood of a specific gene, look at a subnetwork of a set of proteins, and visualize node metadata associated with the networks.

To get started, load packages for use in this tutorial

```
library(paxtoolsr)
library(RCy3)
library(igraph)
library(RColorBrewer)
```

2 Visualize a network using paxtoolsr

We will use paxtoolsr to convert a metabolic pathway file (installed with paxtoolsr) to the Simple Interaction Format (SIF).

```
sif <- toSif(system.file("extdata",  
                        "biopax3-short-metabolic-pathway.owl",  
                        package = "paxtoolsr"))
```

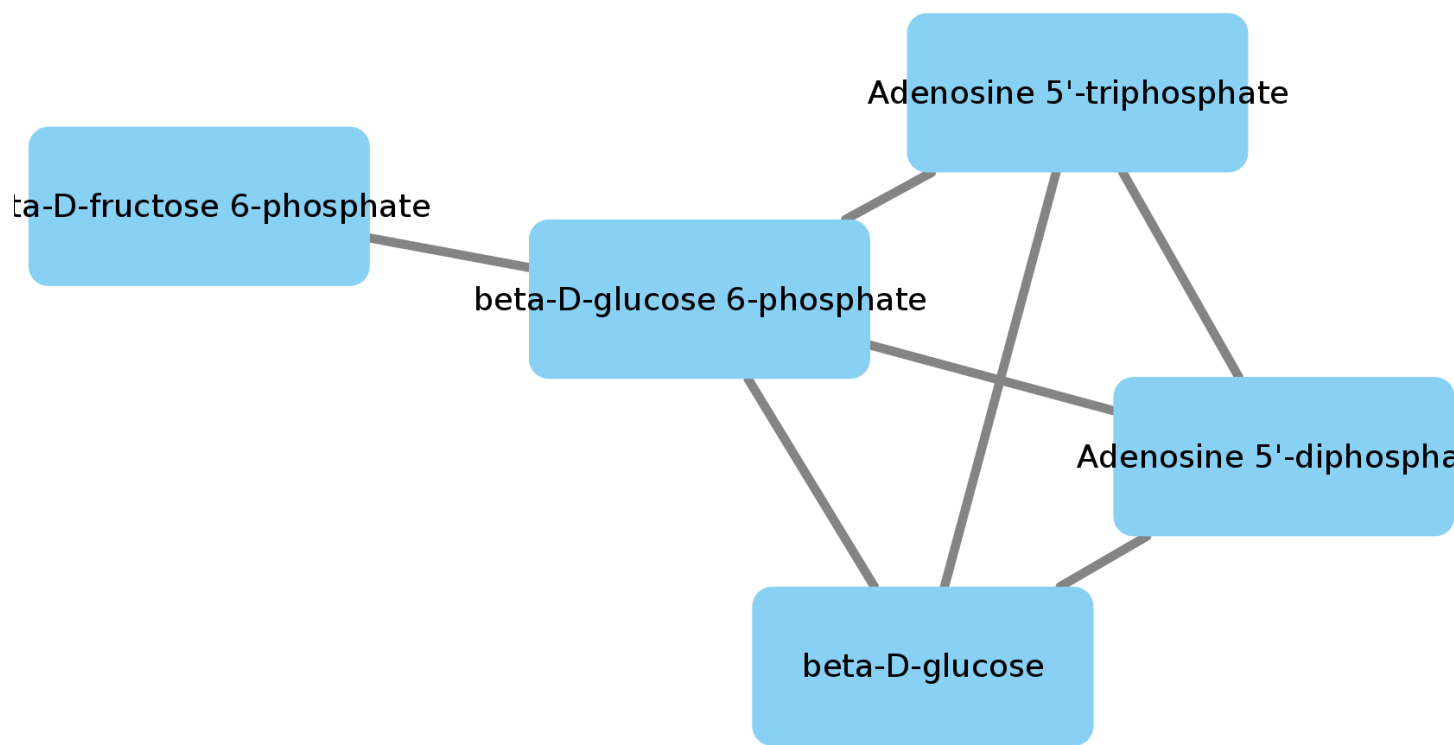
We will use igraph to create a network from the SIF file and then convert that to node and edge lists that can be sent to Cytoscape via RCy3.

```
g <- graph.edgelist(as.matrix(sif[, c(1, 3)]),  
                  directed = FALSE)  
  
g.nodes <- as.data.frame(vertex.attributes(g))  
g.edges <- data.frame(as_edgelist(g))  
names(g.edges) <- c("name.1",  
                   "name.2")  
  
ug <- cyPlot(g.nodes,  
            g.edges)
```

2.1 Send network to Cytoscape using RCy3

```
cw <- CytoscapeWindow("Metabolic pathway from paxtoolsr",  
                      graph = ug,  
                      overwriteWindow = TRUE)  
setDefaultNodeFontSize(cw,  
                       7)
```

```
displayGraph(cw)  
layoutNetwork(cw,  
             "force-directed")  
fitContent(cw)
```



3 Pathway Commons Graph Query

Paxtoolsr can also be used to query Pathway Commons for the neighbours of a particular gene. In this case we will examine the gene neighbourhood of gene Brain-derived neurotrophic factor (BDNF) which is involved in neuron growth and survival.

```
gene <- "BDNF"
t1 <- graphPc(source = gene,
              kind = "neighborhood",
              format = "BINARY_SIF",
              verbose = TRUE)
```

```
## URL: http://www.pathwaycommons.org/pc2/graph?kind=neighborhood&source=BDNF&format=BINARY_SIF
```

For our network we only want to visualize interaction where BDNF controls a reactions that changes the state of the second protein (“controls-state-change-of”, see more info on the binary relations in Pathway commons here).

```
t2 <- t1[which(t1[, 2] == "controls-state-change-of"), ]
```

For our example we only want to use a small number of interactions so we will use `filterSIF()` to reduce the number visualized.

```
ids <- unique(c(t2$PARTICIPANT_A,
               t2$PARTICIPANT_B))
t3 <- filterSif(t2,
               ids = sample(ids,
                           25))
```

We will use this filtered dataframe to create a network using igraph’s `graph.edgelist()`

```
g <- graph.edgelist(as.matrix(t3[, c(1, 3)]),
                  directed = FALSE)
```

Format the graph for sending to Cytoscape.

```
g.nodes <- as.data.frame(vertex.attributes(g))
g.edges <- data.frame(as_edgelist(g))
names(g.edges) <- c("name.1",
                  "name.2")

ug <- cyPlot(g.nodes,
            g.edges)
```

3.1 Send network to Cytoscape using RCy3

Reset the default node size

```
setDefaultNodeFontSize(cw,
                      12)
```

```
cw <- CytoscapeWindow("Pathway Commons graph query from paxtoolsr",  
  graph = ug,  
  overwriteWindow = TRUE)
```

```
displayGraph(cw)  
# setLayoutProperties(cw,  
#                   layout.name = "allegro-spring-electric",  
#                   list(gravity = 100,  
#                       scale = 6))  
layoutNetwork(cw,  
  layout.name = "force-directed")  
fitContent(cw)
```



4 Create a subnetwork from a set of proteins

We will examine a network composed of the genes AKT serine/threonine kinase 1 (“AKT1”), Insulin receptor substrate 1 (“IRS1”), mechanistic target of rapamycin (“MTOR”) and Insulin Like Growth Factor 1 Receptor (“IGF1R”). The paths between these genes can represent metabolic and signaling pathways, molecular and genetic interactions or gene regulation.

```
genes <- c("AKT1",
           "IRS1",
           "MTOR",
           "IGF1R")
t1 <- graphPc(source = genes,
              kind = "PATHSBETWEEN",
              format = "BINARY_SIF",
              verbose = TRUE)
```

```
## URL: http://www.pathwaycommons.org/pc2/graph?kind=PATHSBETWEEN&source=AKT1&source=IRS1&source=MTOR&
```

We will again filter our network to visualize interactions where the genes control a reactions that changes the state of the second gene protein (“controls-state-change-of”, see more info here).

```
t2 <- t1[which(t1[, 2] == "controls-state-change-of"),]
```

Create graph using igraph.

```
g <- graph.edgelist(as.matrix(t2[, c(1, 3)]),
                   directed = FALSE)
```

Convert graph to node and edges lists to send to Cytoscape.

```
g.nodes <- as.data.frame(vertex.attributes(g))
g.edges <- data.frame(as_edgelist(g))
names(g.edges) <- c("name.1",
                   "name.2")

ug <- cyPlot(g.nodes,
             g.edges)
```

4.1 Send network to Cytoscape using RCy3

```
cw <- CytoscapeWindow("Subnetwork of Pathway Commons graph query from paxtoolsr",
                      graph = ug,
                      overwriteWindow = TRUE)
```

```
displayGraph(cw)
layoutNetwork(cw,
              layout.name = "force-directed")
```



5 Adding metadata to Pathway commons networks

It can be useful to overlay data from experiments or other sources onto the nodes and edges of a network. We will do this by simulating some data that will be used to colour the nodes of the network.

```
# Generate a color palette that goes from white to red
# that contains 10 colours
numColors <- 10
colors <- colorRampPalette(brewer.pal(9, "Reds"))(numColors)

# Generate values that could represent some experimental values
values <- runif(length(V(g)$name))

# Scale values to generate indices from the color palette
xrange <- range(values)
newrange <- c(1,
              numColors)

factor <- (newrange[2] - newrange[1])/(xrange[2] - xrange[1])
scaledValues <- newrange[1] + (values - xrange[1]) * factor
indices <- as.integer(scaledValues)
```

After the colours and values for the nodes have been generated, we will add this information to the network that we used in the previous section.

```
g <- cw@graph
g <- initNodeAttribute(graph = g,
                       'indices',
                       "numeric",
                       0)
nodeData(g, nodes(g), "indices") <- indices

cw <- CytoscapeWindow("Coloured network paxtoolsr",
                      graph = g,
                      overwriteWindow = TRUE)
displayGraph(cw) # cw's graph is sent to Cytoscape
```

```
## [1] "label"
## [1] "indices"
```

5.1 Send network to Cytoscape using RCy3

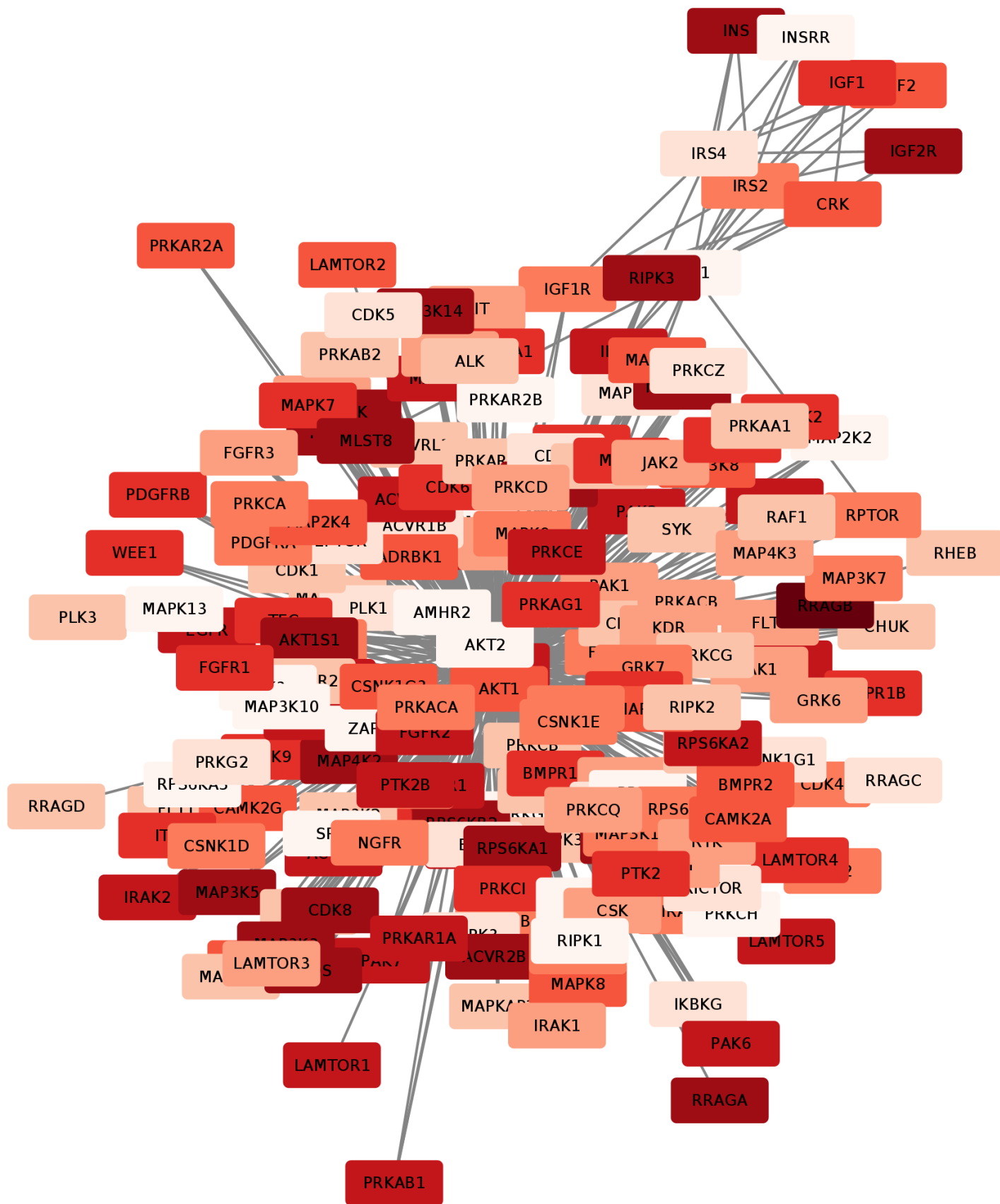
```
displayGraph(cw)
fitContent(cw)
```

Now use the information contained in the “indices” column to add the colours to the nodes in Cytoscape.

```
layoutNetwork(cw,
              layout.name = "force-directed")
setNodeColorRule(cw,
```











```
    "indices",  
    control.points = as.numeric(c(1.0:10.0)), # needs to match type of column in Cytoscape  
    colors,  
    "lookup",  
    default.color="#ffffff")
```

```
## Successfully set rule.
```



Visual Legend for default

Node Fill Color Mapping

Node Fill Color	indices
	1.0
	2.0
	3.0
	4.0
	5.0
	6.0
	7.0
	8.0
	9.0
	10.0

References

Luna, A., Babur, O., Aksoy, B. A, Demir, E., Sander, and C. 2015. "PaxtoolsR: Pathway Analysis in R Using Pathway Commons." *Bioinformatics*, December.

Ono, K, T Muetze, G Kolishovski, P Shannon, and B Demchak. 2015. "CyREST: Turbocharging Cytoscape Access for External Tools via a RESTful API [Version 1; Referees: 2 Approved]." *F1000Research* 4 (478). doi:10.12688/f1000research.6767.1.

Shannon, Paul, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada

Amin, Benno Schwikowski, and Trey Ideker. 2003. "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks." *Genome Research* 13 (11): 2498–2504. <http://genome.cshlp.org/content/13/11/2498.short>.