

hand. For instance, this would be a good time to take advantage of edge weights by putting in desired values (which can be changed later, if necessary) while constructing the edges of the graph. We have manipulated some of the weights, such as the weight between the p85 and p110 nodes, as they are intended to be directly next to each other. Once constructed, the graph can be saved with the `save` command and stored for later use (which has been done already as part of the *integrinMediatedCellAdhesion* dataset).

```
> library("Rgraphviz")

KernSmooth 2.22 installed
Copyright M. P. Wand 1997

> data("integrinMediatedCellAdhesion")
> IMCAGraph

An object of class "graphNEL"
Slot "edgemode":
[1] "directed"
Slot "edgeData":
An object of class "attrData"
Slot "data":
Slot "data":$`ITGB|ITGA`
Slot "data":$`ITGB|ITGA`$weight
[1] 1

Slot "data":$`ITGB|ILK`
Slot "data":$`ITGB|ILK`$weight
[1] 1

Slot "data":$`ITGB|CAV`
Slot "data":$`ITGB|CAV`$weight
[1] 1

Slot "data":$`ITGB|SHC`
Slot "data":$`ITGB|SHC`$weight
[1] 1

Slot "data":$`ITGB|ACTN`
Slot "data":$`ITGB|ACTN`$weight
[1] 1
```

```
Slot "data":$`ITGB|TLN`  
Slot "data":$`ITGB|TLN`$weight  
[1] 1
```

```
Slot "data":$`ITGA|ITGB`  
Slot "data":$`ITGA|ITGB`$weight  
[1] 1
```

```
Slot "data":$`ILK|ITGB`  
Slot "data":$`ILK|ITGB`$weight  
[1] 1
```

```
Slot "data":$`CAV|ITGB`  
Slot "data":$`CAV|ITGB`$weight  
[1] 1
```

```
Slot "data":$`SHC|FYN`  
Slot "data":$`SHC|FYN`$weight  
[1] 1
```

```
Slot "data":$`SHC|GRB2`  
Slot "data":$`SHC|GRB2`$weight  
[1] 1
```

```
Slot "data":$`SHC|ITGB`  
Slot "data":$`SHC|ITGB`$weight  
[1] 1
```

```
Slot "data":$`FYN|SHC`  
Slot "data":$`FYN|SHC`$weight  
[1] 1
```

```
Slot "data":$`GRB2|SHC`  
Slot "data":$`GRB2|SHC`$weight  
[1] 1
```

```
Slot "data":$`GRB2|SOS`
```

```
Slot "data":$`GRB2|SOS`$weight  
[1] 1
```

```
Slot "data":$`SOS|Ha-Ras`  
Slot "data":$`SOS|Ha-Ras`$weight  
[1] 1
```

```
Slot "data":$`Ha-Ras|Raf`  
Slot "data":$`Ha-Ras|Raf`$weight  
[1] 1
```

```
Slot "data":$`Raf|MEK`  
Slot "data":$`Raf|MEK`$weight  
[1] 1
```

```
Slot "data":$`MEK|ERK`  
Slot "data":$`MEK|ERK`$weight  
[1] 1
```

```
Slot "data":$`ERK|MYLK`  
Slot "data":$`ERK|MYLK`$weight  
[1] 1
```

```
Slot "data":$`ERK|cell proliferation`  
Slot "data":$`ERK|cell proliferation`$weight  
[1] 1
```

```
Slot "data":$`MYLK|MYO`  
Slot "data":$`MYLK|MYO`$weight  
[1] 1
```

```
Slot "data":$`MYO|F-actin`  
Slot "data":$`MYO|F-actin`$weight  
[1] 1
```

```
Slot "data":$`ACTN|VCL`  
Slot "data":$`ACTN|VCL`$weight
```

[1] 1

Slot "data":\$`ACTN|TLN`
Slot "data":\$`ACTN|TLN`\$weight
[1] 1

Slot "data":\$`ACTN|ITGB`
Slot "data":\$`ACTN|ITGB`\$weight
[1] 1

Slot "data":\$`ACTN|ZYG`
Slot "data":\$`ACTN|ZYG`\$weight
[1] 1

Slot "data":\$`VCL|ACTN`
Slot "data":\$`VCL|ACTN`\$weight
[1] 1

Slot "data":\$`VCL|TLN`
Slot "data":\$`VCL|TLN`\$weight
[1] 1

Slot "data":\$`VCL|PXN`
Slot "data":\$`VCL|PXN`\$weight
[1] 1

Slot "data":\$`VCL|SH3D5`
Slot "data":\$`VCL|SH3D5`\$weight
[1] 1

Slot "data":\$`VCL|TNS`
Slot "data":\$`VCL|TNS`\$weight
[1] 1

Slot "data":\$`TLN|ACTN`
Slot "data":\$`TLN|ACTN`\$weight
[1] 1

```
Slot "data":$`TLN|VCL`  
Slot "data":$`TLN|VCL`$weight  
[1] 1
```

```
Slot "data":$`TLN|ITGB`  
Slot "data":$`TLN|ITGB`$weight  
[1] 1
```

```
Slot "data":$`TLN|CAPN`  
Slot "data":$`TLN|CAPN`$weight  
[1] 1
```

```
Slot "data":$`TLN|FAK`  
Slot "data":$`TLN|FAK`$weight  
[1] 1
```

```
Slot "data":$`PXN|VCL`  
Slot "data":$`PXN|VCL`$weight  
[1] 1
```

```
Slot "data":$`PXN|BCAR1`  
Slot "data":$`PXN|BCAR1`$weight  
[1] 1
```

```
Slot "data":$`PXN|GIT2`  
Slot "data":$`PXN|GIT2`$weight  
[1] 1
```

```
Slot "data":$`PXN|FAK`  
Slot "data":$`PXN|FAK`$weight  
[1] 1
```

```
Slot "data":$`ZYG|ACTN`  
Slot "data":$`ZYG|ACTN`$weight  
[1] 1
```

```
Slot "data":$`ZYX|VASP`  
Slot "data":$`ZYX|VASP`$weight  
[1] 1
```

```
Slot "data":$`VASP|ZYX`  
Slot "data":$`VASP|ZYX`$weight  
[1] 1
```

```
Slot "data":$`SH3D5|VCL`  
Slot "data":$`SH3D5|VCL`$weight  
[1] 1
```

```
Slot "data":$`TNS|VCL`  
Slot "data":$`TNS|VCL`$weight  
[1] 1
```

```
Slot "data":$`CAPN|CAPNS`  
Slot "data":$`CAPN|CAPNS`$weight  
[1] 1
```

```
Slot "data":$`CAPN|TLN`  
Slot "data":$`CAPN|TLN`$weight  
[1] 1
```

```
Slot "data":$`CAPNS|CAPN`  
Slot "data":$`CAPNS|CAPN`$weight  
[1] 1
```

```
Slot "data":$`CAPNS|SRC`  
Slot "data":$`CAPNS|SRC`$weight  
[1] 1
```

```
Slot "data":$`SRC|FAK`  
Slot "data":$`SRC|FAK`$weight  
[1] 1
```

```
Slot "data":$`SRC|BCAR1`  
Slot "data":$`SRC|BCAR1`$weight  
[1] 1
```

```
Slot "data":$`SRC|CSK`  
Slot "data":$`SRC|CSK`$weight  
[1] 1
```

```
Slot "data":$`FAK|MEK`  
Slot "data":$`FAK|MEK`$weight  
[1] 1
```

```
Slot "data":$`FAK|TLN`  
Slot "data":$`FAK|TLN`$weight  
[1] 1
```

```
Slot "data":$`FAK|PXN`  
Slot "data":$`FAK|PXN`$weight  
[1] 1
```

```
Slot "data":$`FAK|BCAR1`  
Slot "data":$`FAK|BCAR1`$weight  
[1] 1
```

```
Slot "data":$`FAK|p110`  
Slot "data":$`FAK|p110`$weight  
[1] 1
```

```
Slot "data":$`BCAR1|SRC`  
Slot "data":$`BCAR1|SRC`$weight  
[1] 1
```

```
Slot "data":$`BCAR1|CRK`  
Slot "data":$`BCAR1|CRK`$weight  
[1] 1
```

```
Slot "data":$`CSK|SRC`
```


Slot "data":\$`CSK|SRC`\$weight
[1] 1

Slot "data":\$`CRK|DOCK1`
Slot "data":\$`CRK|DOCK1`\$weight
[1] 1

Slot "data":\$`CRK|GRF2`
Slot "data":\$`CRK|GRF2`\$weight
[1] 1

Slot "data":\$`DOCK1|CRK`
Slot "data":\$`DOCK1|CRK`\$weight
[1] 1

Slot "data":\$`GRF2|RAP1`
Slot "data":\$`GRF2|RAP1`\$weight
[1] 1

Slot "data":\$`RAP1|JNK`
Slot "data":\$`RAP1|JNK`\$weight
[1] 1

Slot "data":\$`JNK|cell maintenance`
Slot "data":\$`JNK|cell maintenance`\$weight
[1] 1

Slot "data":\$`GIT2|ARHGEF`
Slot "data":\$`GIT2|ARHGEF`\$weight
[1] 1

Slot "data":\$`ARHGEF|PAK`
Slot "data":\$`ARHGEF|PAK`\$weight
[1] 1

Slot "data":\$`PAK|ARHGEF`
Slot "data":\$`PAK|ARHGEF`\$weight

[1] 1

Slot "data":`\$`PAK|JNK`
Slot "data":`\$`PAK|JNK`\$weight
[1] 1

Slot "data":`\$`PAK|MYLK`
Slot "data":`\$`PAK|MYLK`\$weight
[1] 1

Slot "data":`\$`p85|p110`
Slot "data":`\$`p85|p110`\$weight
[1] 10

Slot "data":`\$`p110|p85`
Slot "data":`\$`p110|p85`\$weight
[1] 1

Slot "data":`\$`p110|Phosphatidylinositol signaling system`
Slot "data":`\$`p110|Phosphatidylinositol signaling system`\$weight
[1] 1

Slot "data":`\$`p110|AKT`
Slot "data":`\$`p110|AKT`\$weight
[1] 1

Slot "data":`\$`Phosphatidylinositol signaling system|VAV`
Slot "data":`\$`Phosphatidylinositol signaling system|VAV`\$weight
[1] 1

Slot "data":`\$`Phosphatidylinositol signaling system|PDPK1`
Slot "data":`\$`Phosphatidylinositol signaling system|PDPK1`\$weight
[1] 1

Slot "data":`\$`VAV|RAC`
Slot "data":`\$`VAV|RAC`\$weight
[1] 1

```
Slot "data":$`PDPK1|AKT`  
Slot "data":$`PDPK1|AKT`$weight  
[1] 1
```

```
Slot "data":$`AKT|cell maintenance`  
Slot "data":$`AKT|cell maintenance`$weight  
[1] 1
```

```
Slot "data":$`RAC|PAK`  
Slot "data":$`RAC|PAK`$weight  
[1] 1
```

```
Slot "data":$`RAC|cell motility`  
Slot "data":$`RAC|cell motility`$weight  
[1] 1
```

```
Slot "data":$`CDC42|RAC`  
Slot "data":$`CDC42|RAC`$weight  
[1] 1
```

```
Slot "data":$`CDC42|PAK`  
Slot "data":$`CDC42|PAK`$weight  
[1] 1
```

```
Slot "data":$`RHO|PI5K`  
Slot "data":$`RHO|PI5K`$weight  
[1] 1
```

```
Slot "data":$`RHO|ROCK`  
Slot "data":$`RHO|ROCK`$weight  
[1] 1
```

```
Slot "data":$`PI5K|Phosphatidylinositol signaling system`  
Slot "data":$`PI5K|Phosphatidylinositol signaling system`$weight  
[1] 1
```

```
Slot "data":$`ROCK|MYO-P`  
Slot "data":$`ROCK|MYO-P`$weight  
[1] 1
```

```
Slot "data":$`MYO-P|MYO`  
Slot "data":$`MYO-P|MYO`$weight  
[1] 1
```

```
Slot "data":$`F-actin|MYO`  
Slot "data":$`F-actin|MYO`$weight  
[1] 1
```

```
Slot "defaults":  
Slot "defaults":$weight  
[1] 1
```

```
Slot "nodeData":  
An object of class "attrData"  
Slot "data":  
list()  
Slot "defaults":  
list()  
Slot "nodes":  
[1] "ITGB"  
[2] "ITGA"  
[3] "ILK"  
[4] "CAV"  
[5] "SHC"  
[6] "FYN"  
[7] "GRB2"  
[8] "SOS"  
[9] "Ha-Ras"  
[10] "Raf"  
[11] "MEK"  
[12] "ERK"  
[13] "MYLK"  
[14] "MYO"  
[15] "ACTN"  
[16] "VCL"  
[17] "TLN"  
[18] "PXN"  
[19] "ZYG"
```

[20] "VASP"
 [21] "SH3D5"
 [22] "TNS"
 [23] "CAPN"
 [24] "CAPNS"
 [25] "SRC"
 [26] "FAK"
 [27] "BCAR1"
 [28] "CSK"
 [29] "CRK"
 [30] "DOCK1"
 [31] "GRF2"
 [32] "RAP1"
 [33] "JNK"
 [34] "GIT2"
 [35] "ARHGEF"
 [36] "PAK"
 [37] "p85"
 [38] "p110"
 [39] "Phosphatidylinositol signaling system"
 [40] "VAV"
 [41] "PDPK1"
 [42] "AKT"
 [43] "RAC"
 [44] "CDC42"
 [45] "RHO"
 [46] "PI5K"
 [47] "ROCK"
 [48] "MYO-P"
 [49] "cell maintenance"
 [50] "cell motility"
 [51] "F-actin"
 [52] "cell proliferation"
 Slot "edgeL":
 Slot "edgeL":\$ITGB
 Slot "edgeL":\$ITGB\$edges
 [1] 2 3 4 5 15 17

Slot "edgeL":\$ITGA
 Slot "edgeL":\$ITGA\$edges
 [1] 1

Slot "edgeL":\$ILK
 Slot "edgeL":\$ILK\$edges

[1] 1

Slot "edgeL":\$CAV
Slot "edgeL":\$CAV\$edges
[1] 1

Slot "edgeL":\$SHC
Slot "edgeL":\$SHC\$edges
[1] 6 7 1

Slot "edgeL":\$FYN
Slot "edgeL":\$FYN\$edges
[1] 5

Slot "edgeL":\$GRB2
Slot "edgeL":\$GRB2\$edges
[1] 5 8

Slot "edgeL":\$SOS
Slot "edgeL":\$SOS\$edges
[1] 9

Slot "edgeL":\$`Ha-Ras`
Slot "edgeL":\$`Ha-Ras`\$edges
[1] 10

Slot "edgeL":\$Raf
Slot "edgeL":\$Raf\$edges
[1] 11

Slot "edgeL":\$MEK
Slot "edgeL":\$MEK\$edges
[1] 12

Slot "edgeL":\$ERK
Slot "edgeL":\$ERK\$edges
[1] 13 52

```
Slot "edgeL":$MYLK
Slot "edgeL":$MYLK$edges
[1] 14
```

```
Slot "edgeL":$MYO
Slot "edgeL":$MYO$edges
[1] 51
```

```
Slot "edgeL":$ACTN
Slot "edgeL":$ACTN$edges
[1] 16 17 1 19
```

```
Slot "edgeL":$VCL
Slot "edgeL":$VCL$edges
[1] 15 17 18 21 22
```

```
Slot "edgeL":$TLN
Slot "edgeL":$TLN$edges
[1] 15 16 1 23 26
```

```
Slot "edgeL":$PXN
Slot "edgeL":$PXN$edges
[1] 16 27 34 26
```

```
Slot "edgeL":$ZYX
Slot "edgeL":$ZYX$edges
[1] 15 20
```

```
Slot "edgeL":$VASP
Slot "edgeL":$VASP$edges
[1] 19
```

```
Slot "edgeL":$SH3D5
Slot "edgeL":$SH3D5$edges
[1] 16
```

```
Slot "edgeL":$TNS
Slot "edgeL":$TNS$edges
[1] 16
```

```
Slot "edgeL":$CAPN
Slot "edgeL":$CAPN$edges
[1] 24 17
```

```
Slot "edgeL":$CAPNS
Slot "edgeL":$CAPNS$edges
[1] 23 25
```

```
Slot "edgeL":$SRC
Slot "edgeL":$SRC$edges
[1] 26 27 28
```

```
Slot "edgeL":$FAK
Slot "edgeL":$FAK$edges
[1] 11 17 18 27 38
```

```
Slot "edgeL":$BCAR1
Slot "edgeL":$BCAR1$edges
[1] 25 29
```

```
Slot "edgeL":$CSK
Slot "edgeL":$CSK$edges
[1] 25
```

```
Slot "edgeL":$CRK
Slot "edgeL":$CRK$edges
[1] 30 31
```

```
Slot "edgeL":$DOCK1
Slot "edgeL":$DOCK1$edges
[1] 29
```



```
Slot "edgeL":$GRF2
Slot "edgeL":$GRF2$edges
[1] 32
```

```
Slot "edgeL":$RAP1
Slot "edgeL":$RAP1$edges
[1] 33
```

```
Slot "edgeL":$JNK
Slot "edgeL":$JNK$edges
[1] 49
```

```
Slot "edgeL":$GIT2
Slot "edgeL":$GIT2$edges
[1] 35
```

```
Slot "edgeL":$ARHGEF
Slot "edgeL":$ARHGEF$edges
[1] 36
```

```
Slot "edgeL":$PAK
Slot "edgeL":$PAK$edges
[1] 35 33 13
```

```
Slot "edgeL":$p85
Slot "edgeL":$p85$edges
[1] 38
```

```
Slot "edgeL":$p110
Slot "edgeL":$p110$edges
[1] 37 39 42
```

```
Slot "edgeL":`$Phosphatidylinositol signaling system`
Slot "edgeL":`$Phosphatidylinositol signaling system`$edges
[1] 40 41
```

```
Slot "edgeL":$VAV
```

Slot "edgeL":\$VAV\$edges
[1] 43

Slot "edgeL":\$PDPK1
Slot "edgeL":\$PDPK1\$edges
[1] 42

Slot "edgeL":\$AKT
Slot "edgeL":\$AKT\$edges
[1] 49

Slot "edgeL":\$RAC
Slot "edgeL":\$RAC\$edges
[1] 36 50

Slot "edgeL":\$CDC42
Slot "edgeL":\$CDC42\$edges
[1] 43 36

Slot "edgeL":\$RHO
Slot "edgeL":\$RHO\$edges
[1] 46 47

Slot "edgeL":\$PI5K
Slot "edgeL":\$PI5K\$edges
[1] 39

Slot "edgeL":\$ROCK
Slot "edgeL":\$ROCK\$edges
[1] 48

Slot "edgeL":\$`MYO-P`
Slot "edgeL":\$`MYO-P`\$edges
[1] 14

Slot "edgeL":\$`cell maintenance`
Slot "edgeL":\$`cell maintenance`\$edges

```
numeric(0)
```

```
Slot "edgeL":$`cell motility`  
Slot "edgeL":$`cell motility`$edges  
numeric(0)
```

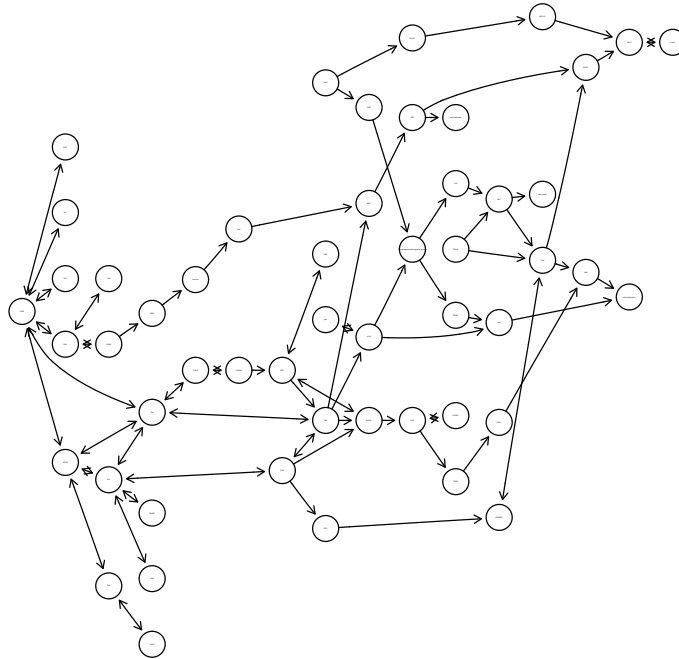
```
Slot "edgeL":$`F-actin`  
Slot "edgeL":$`F-actin`$edges  
[1] 14
```

```
Slot "edgeL":$`cell proliferation`  
Slot "edgeL":$`cell proliferation`$edges  
numeric(0)
```

3 Laying out the graph

Laying out a pathway graph is much like dealing with any other graph, except that typically we want to as closely emulate the officially laid out graph (or at least make it look like an actual pathway - the Graphviz layout methods were not designed with this task in mind). A lot of experimentation comes into play, in order to find the right combination of attributes, although there are some general tips that can help out. The first thing to know is that we will almost always want to use the *dot* layout, as that will provide the closest base to work off. Likewise, the *rankdir* attribute should be set to *LR*, to give us the left to right look of the graph. To see our starting point, here is the *IMCAGraph* with just those settings.

```
> plot(IMCAGraph, attrs = list(graph = list(rankdir = "LR")))
```



Note that `IMCAAttrs$defAttrs` is simply just the `rankdir` attribute for `graph`, so we will be using that in place of the `list` call from now on.

This plot is not terrible, in that it conveys the proper information, but the formatting is quite different from the layout at KEGG, and can be difficult to get a coherent idea of what is going on. Furthermore, smaller things like the coloration of the nodes and the shape of the phosphatidylinositol signaling system are not being represented here.

Here is where using other attributes can start to have a positive effect. We can set the colors of each node (another piece that is necessary to enter in manually) and change the shape of the phosphatidylinositol signaling system node to be an ellipse. We have done this for this graph in the `IMCAAttrs$nodeAttrs` data:

```
> IMCAAttrs$nodeAttrs$shape
Phosphatidylinositol signaling system
    "ellipse"

> IMCAAttrs$nodeAttrs$width
Phosphatidylinositol signaling system
    1.5

> IMCAAttrs$nodeAttrs$height
```

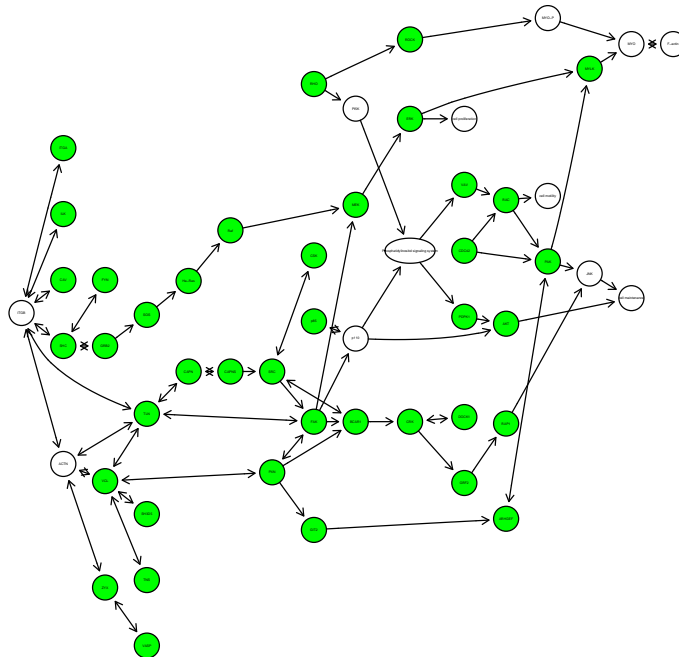
```
Phosphatidylinositol signaling system
0.75
```

```
> IMCAAttrs$nodeAttrs$fillcolor[1:10]
```

```
ITGB ITGA ILK CAV SHC FYN GRB2 SOS Ha-Ras Raf
"white" "green" "green" "green" "green" "green" "green" "green" "green" "green"
```

Using these attributes to plot the graph will get us a bit closer to our goal:

```
> plot(IMCAGraph, attrs = IMCAAttrs$defAttrs, nodeAttrs = IMCAAttrs$nodeAttrs)
```



Here the color scheme is now the same as on KEGG, and using an ellipse helps with the rendering of the phosphatidylinositol signaling system node. However, we're still left with the issue that the layout itself doesn't convey the same meaning as the original. The output nodes are scattered about, there's not a clear sense of where the membrane nodes are, and many nodes that are intended to be close to each other simply are not. This is where the use of subgraphs and clusters can help. In Graphviz, a subgraph is an organizational method to note that a set of nodes and edges belong in the same conceptual space, sharing attributes and the like. While there is some tendency to have nodes be laid out near each other in a subgraph, there is no guarantee of this, and the results can be highly dependent on the layout method (*dot*, *neato*, etc). A Graphviz cluster is a subgraph which is laid out as a separate graph and then

introduced into the main graph. This provides a much stronger guarantee of having the nodes clustered together visually. For a description of how to specify subgraphs in *Rgraphviz*, please see the vignette `HowTo Render A Graph Using Rgraphviz`.

So here we will define four subgraphs: One will be the membrane nodes, one will be the output nodes, one will be the F-actin block and the last will be everything else. It would be possible to specify more subgraphs to try to help keep things more blocked together like the original graph, but for the purposes of this document, these are what will be used.

```
> nodes <- nodes(IMCAGraph)
> sg1 <- subGraph(nodes[c(2, 1, 3, 4)], IMCAGraph)
> sg2 <- subGraph(c(nodes[5:14], nodes[19:48], nodes[52]), IMCAGraph)
> sg3 <- subGraph(nodes[49:51], IMCAGraph)
> sg4 <- subGraph(nodes[15:18], IMCAGraph)
```

While we have the subgraphs defined, we still have not determined whether to use these as subgraphs or clusters in Graphviz. Ideally, we would like to use clusters, as that guarantees that the nodes will be laid out close together. However, it would also be useful to utilize the *rank* attribute for the membrane and output nodes, specifically using the values *source* and *sink* respectively. That will help to get the verticle line up that we see in the KEGG graph and create more of the left to right pathway feel. The problem is that *rank* only works with subgraphs and not clusters. So for the membrane and output subgraphs, we will be defining them as Graphviz subgraphs, and the other two subgraphs will be defined as clusters. We have already prepared all of this as *IMCAAttrs\$subGList*:

```
> IMCAAttrs$subGList

[[1]]
[[1]]$graph
An object of class "graphNEL"
Slot "nodes":
[1] "ITGA" "ITGB" "ILK" "CAV"
Slot "edgeL":
Slot "edgeL":$ITGA
Slot "edgeL":$ITGA$edges
2
2

Slot "edgeL":$ITGA$weights
2
1

Slot "edgeL":$ITGB
```

```

Slot "edgeL":$ITGB$edges
1 3 4
1 3 4

Slot "edgeL":$ITGB$weights
1 3 4
1 1 1

Slot "edgeL":$ILK
Slot "edgeL":$ILK$edges
2
2

Slot "edgeL":$ILK$weights
2
1

Slot "edgeL":$CAV
Slot "edgeL":$CAV$edges
2
2

Slot "edgeL":$CAV$weights
2
1

Slot "edgemode":
[1] "undirected"

[[1]]$cluster
[1] FALSE

[[1]]$attrs
  rank
"source"

[[2]]
[[2]]$graph
An object of class "graphNEL"
Slot "nodes":
 [1] "SHC"
 [2] "FYN"

```

[3] "GRB2"
 [4] "SOS"
 [5] "Ha-Ras"
 [6] "Raf"
 [7] "MEK"
 [8] "ERK"
 [9] "MYLK"
 [10] "MYO"
 [11] "ZYX"
 [12] "VASP"
 [13] "SH3D5"
 [14] "TNS"
 [15] "CAPN"
 [16] "CAPNS"
 [17] "SRC"
 [18] "FAK"
 [19] "BCAR1"
 [20] "CSK"
 [21] "CRK"
 [22] "DOCK1"
 [23] "GRF2"
 [24] "RAP1"
 [25] "JNK"
 [26] "GIT2"
 [27] "ARHGEF"
 [28] "PAK"
 [29] "p85"
 [30] "p110"
 [31] "Phosphatidylinositol signaling system"
 [32] "VAV"
 [33] "PDPK1"
 [34] "AKT"
 [35] "RAC"
 [36] "CDC42"
 [37] "RHO"
 [38] "PI5K"
 [39] "ROCK"
 [40] "MYO-P"
 [41] "cell proliferation"
 Slot "edgeL":
 Slot "edgeL":\$SHC
 Slot "edgeL":\$SHC\$edges
 2 3
 2 3

 Slot "edgeL":\$SHC\$weights

2 3
1 1

Slot "edgeL":\$FYN
Slot "edgeL":\$FYN\$edges
1
1

Slot "edgeL":\$FYN\$weights
1
1

Slot "edgeL":\$GRB2
Slot "edgeL":\$GRB2\$edges
1 4
1 4

Slot "edgeL":\$GRB2\$weights
1 4
1 1

Slot "edgeL":\$SOS
Slot "edgeL":\$SOS\$edges
5
5

Slot "edgeL":\$SOS\$weights
5
1

Slot "edgeL":\$`Ha-Ras`
Slot "edgeL":\$`Ha-Ras`\$edges
6
6

Slot "edgeL":\$`Ha-Ras`\$weights
6
1

Slot "edgeL":\$Raf
Slot "edgeL":\$Raf\$edges

```
7
7

Slot "edgeL":$Raf$weights
7
1

Slot "edgeL":$MEK
Slot "edgeL":$MEK$edges
8
8

Slot "edgeL":$MEK$weights
8
1

Slot "edgeL":$ERK
Slot "edgeL":$ERK$edges
9 41
9 41

Slot "edgeL":$ERK$weights
9 41
1 1

Slot "edgeL":$MYLK
Slot "edgeL":$MYLK$edges
10
10

Slot "edgeL":$MYLK$weights
10
1

Slot "edgeL":$MYO
NULL

Slot "edgeL":$ZYX
Slot "edgeL":$ZYX$edges
12
12
```

Slot "edgeL":\$ZYX\$weights
12
1

Slot "edgeL":\$VASP
Slot "edgeL":\$VASP\$edges
11
11

Slot "edgeL":\$VASP\$weights
11
1

Slot "edgeL":\$SH3D5
NULL

Slot "edgeL":\$TNS
NULL

Slot "edgeL":\$CAPN
Slot "edgeL":\$CAPN\$edges
16
16

Slot "edgeL":\$CAPN\$weights
16
1

Slot "edgeL":\$CAPNS
Slot "edgeL":\$CAPNS\$edges
15 17
15 17

Slot "edgeL":\$CAPNS\$weights
15 17
1 1

Slot "edgeL":\$SRC
Slot "edgeL":\$SRC\$edges
18 19 20
18 19 20

```
Slot "edgeL":$SRC$weights
18 19 20
1 1 1
```

```
Slot "edgeL":$FAK
Slot "edgeL":$FAK$edges
7 19 30
7 19 30
```

```
Slot "edgeL":$FAK$weights
7 19 30
1 1 1
```

```
Slot "edgeL":$BCAR1
Slot "edgeL":$BCAR1$edges
17 21
17 21
```

```
Slot "edgeL":$BCAR1$weights
17 21
1 1
```

```
Slot "edgeL":$CSK
Slot "edgeL":$CSK$edges
17
17
```

```
Slot "edgeL":$CSK$weights
17
1
```

```
Slot "edgeL":$CRK
Slot "edgeL":$CRK$edges
22 23
22 23
```

```
Slot "edgeL":$CRK$weights
22 23
1 1
```

```
Slot "edgeL":$DOCK1
```

Slot "edgeL":\$DOCK1\$edges

21

21

Slot "edgeL":\$DOCK1\$weights

21

1

Slot "edgeL":\$GRF2

Slot "edgeL":\$GRF2\$edges

24

24

Slot "edgeL":\$GRF2\$weights

24

1

Slot "edgeL":\$RAP1

Slot "edgeL":\$RAP1\$edges

25

25

Slot "edgeL":\$RAP1\$weights

25

1

Slot "edgeL":\$JNK

NULL

Slot "edgeL":\$GIT2

Slot "edgeL":\$GIT2\$edges

27

27

Slot "edgeL":\$GIT2\$weights

27

1

Slot "edgeL":\$ARHGEF

Slot "edgeL":\$ARHGEF\$edges

28

28

```
Slot "edgeL":$ARHGEF$weights
28
1
```

```
Slot "edgeL":$PAK
Slot "edgeL":$PAK$edges
27 25 9
27 25 9
```

```
Slot "edgeL":$PAK$weights
27 25 9
1 1 1
```

```
Slot "edgeL":$p85
Slot "edgeL":$p85$edges
30
30
```

```
Slot "edgeL":$p85$weights
30
10
```

```
Slot "edgeL":$p110
Slot "edgeL":$p110$edges
29 31 34
29 31 34
```

```
Slot "edgeL":$p110$weights
29 31 34
1 1 1
```

```
Slot "edgeL":$`Phosphatidylinositol signaling system`
Slot "edgeL":$`Phosphatidylinositol signaling system`$edges
32 33
32 33
```

```
Slot "edgeL":$`Phosphatidylinositol signaling system`$weights
32 33
1 1
```

Slot "edgeL":\$VAV
Slot "edgeL":\$VAV\$edges
35
35

Slot "edgeL":\$VAV\$weights
35
1

Slot "edgeL":\$PDPK1
Slot "edgeL":\$PDPK1\$edges
34
34

Slot "edgeL":\$PDPK1\$weights
34
1

Slot "edgeL":\$AKT
NULL

Slot "edgeL":\$RAC
Slot "edgeL":\$RAC\$edges
28
28

Slot "edgeL":\$RAC\$weights
28
1

Slot "edgeL":\$CDC42
Slot "edgeL":\$CDC42\$edges
35 28
35 28

Slot "edgeL":\$CDC42\$weights
35 28
1 1

Slot "edgeL":\$RHO
Slot "edgeL":\$RHO\$edges
38 39

38 39

Slot "edgeL":\$RH0\$weights

38 39

1 1

Slot "edgeL":\$PI5K

Slot "edgeL":\$PI5K\$edges

31

31

Slot "edgeL":\$PI5K\$weights

31

1

Slot "edgeL":\$ROCK

Slot "edgeL":\$ROCK\$edges

40

40

Slot "edgeL":\$ROCK\$weights

40

1

Slot "edgeL":\$`MYO-P`

Slot "edgeL":\$`MYO-P`\$edges

10

10

Slot "edgeL":\$`MYO-P`\$weights

10

1

Slot "edgeL":\$`cell proliferation`

NULL

Slot "edgemode":

[1] "undirected"

[[3]]

[[3]]\$graph


```

An object of class "graphNEL"
Slot "nodes":
[1] "cell maintenance" "cell motility" "F-actin"
Slot "edgeL":
Slot "edgeL":$`cell maintenance`
NULL

Slot "edgeL":$`cell motility`
NULL

Slot "edgeL":$`F-actin`
NULL

Slot "edgemode":
[1] "undirected"

[[3]]$cluster
[1] FALSE

[[3]]$attrs
  rank
"sink"

[[4]]
[[4]]$graph
An object of class "graphNEL"
Slot "nodes":
[1] "ACTN" "VCL" "TLN" "PXN"
Slot "edgeL":
Slot "edgeL":$ACTN
Slot "edgeL":$ACTN$edges
2 3
2 3

Slot "edgeL":$ACTN$weights
2 3
1 1

Slot "edgeL":$VCL
Slot "edgeL":$VCL$edges
1 3 4
1 3 4

Slot "edgeL":$VCL$weights

```

```
1 3 4
1 1 1
```

```
Slot "edgeL":$TLN
Slot "edgeL":$TLN$edges
1 2
1 2
```

```
Slot "edgeL":$TLN$weights
1 2
1 1
```

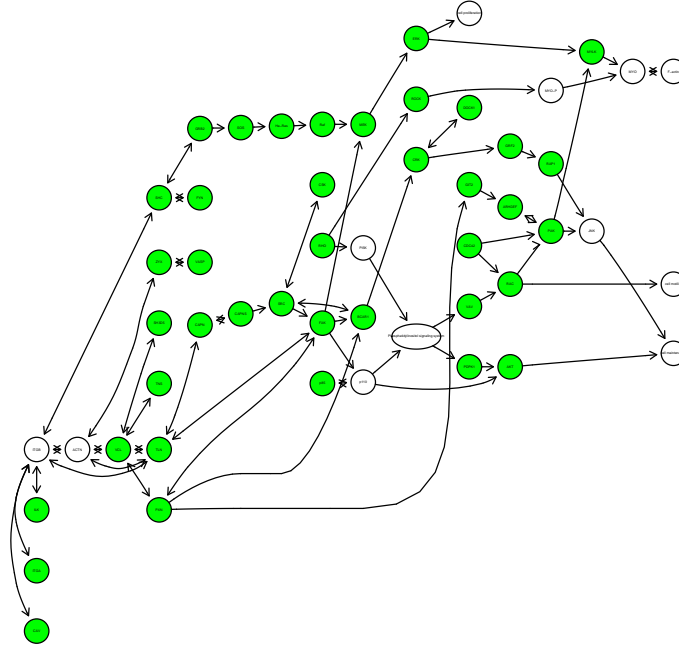
```
Slot "edgeL":$PXN
Slot "edgeL":$PXN$edges
2
2
```

```
Slot "edgeL":$PXN$weights
2
1
```

```
Slot "edgemode":
[1] "undirected"
```

You can see that subgraphs 1 and 3 have the *cluster* parameter set to *FALSE* as well as having a *rank* attribute set appropriate. Subgraphs 2 and 4 simply have the subgraph itself, and will be laid out as a cluster without any special attributes. Using this subgraph list, we now get:

```
> plot(IMCAGraph, attrs = IMCAAttrs$defAttrs, nodeAttrs = IMCAAttrs$nodeAttrs,
+      subGList = IMCAAttrs$subGList)
```



While this is still not identical to the image on KEGG (and for most graphs, it will be impossible given current abilities to do so), this layout is now much closer to providing an accurate visual rendition of the pathway. We can see the output nodes are now to the right end of the graph, and while not neatly stacked on the left hand side the membrane nodes are to the left side of the rest. We can also see the F-actin group in the lower left portion of the graph, representing one of the clusters.

4 Working with the layout

One of the benefits of using *Rgraphviz* to perform your layout as opposed to using the static layouts provided by sites like KEGG, is the ability to work with outside data and visualize it using your graph. The `plotExpressionGraph` function in *geneplotter* can be used to take expression data and then color nodes based on the level of expression. By default, this function will color nodes blue, green or red, corresponding to expression levels of 0-100, 101-500, and 501+ respectively. Here we will use this function along with the *fibroEset* and *hgu95av2* data packages and the *IMCAAttrs\$IMCALocuLink* data which maps the nodes to their LocusLink ID values.

```
> require("geneplotter")
```

```
[1] TRUE
```

```

> require("fibroEset")

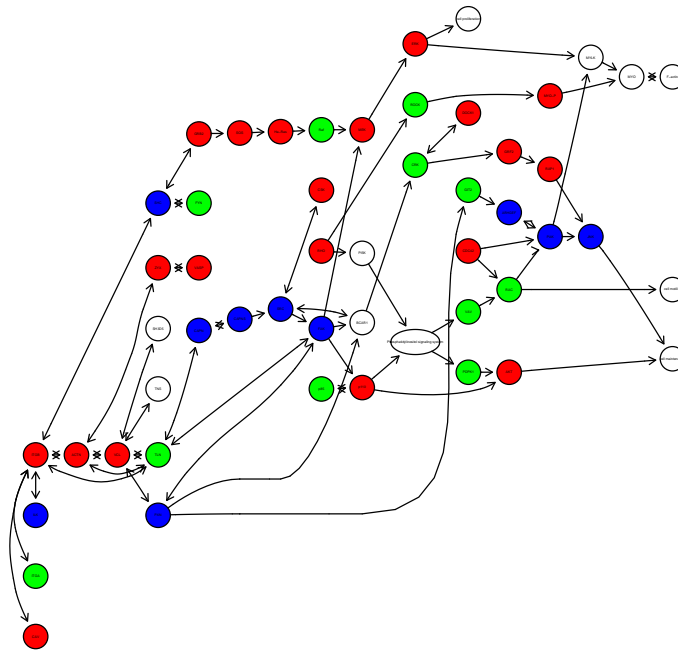
[1] TRUE

> require("hgu95av2")

[1] TRUE

> data("fibroEset")
> plotExpressionGraph(IMCAGraph, IMCAAttrs$LocusLink, exprs(fibroEset)[,
+   1], hgu95av2LOCUSID, attrs = IMCAAttrs$defAttrs, subGList = IMCAAttrs$subGList,
+   nodeAttrs = IMCAAttrs$nodeAttrs)

```



One can also simply choose to layout the pathway based on the needs and desires of a particular situation. For instance, the following layout could be used in situations where the node names are the important visual cue, as opposed to the previous example where the nodes themselves are being used to demonstrate values:

```

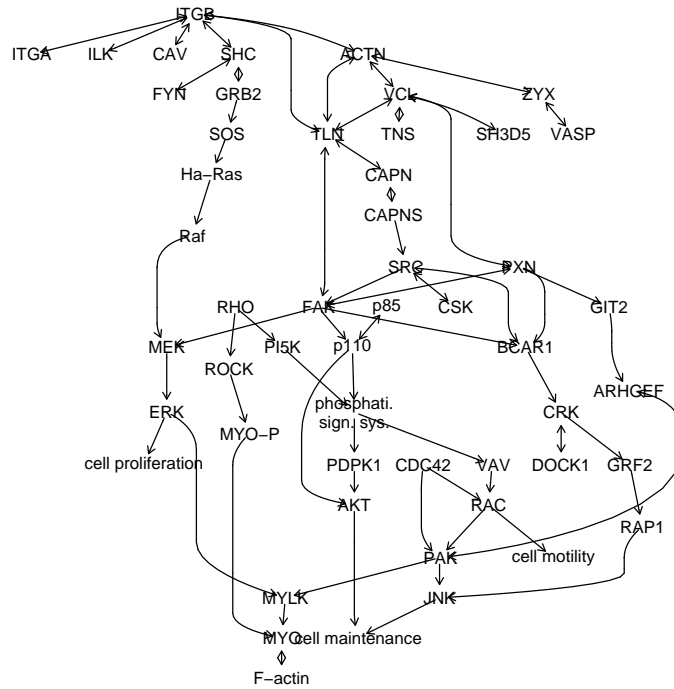
> z <- IMCAGraph
> nodes(z)[39] <- c("phosphati.\nsign. sys.")
> nag <- agopen(z, name = "nag", attrs = list(node = list(color = "white",
+   fontcolor = "white"), edge = list(arrowsize = 2.8, minlen = 3)))
> nagxy <- getNodeXY(nag)

```

```

> plot(nag)
> text(nagxy, label = nodes(z), cex = 0.8)

```



5 Conclusions

At this time, laying out a pathway can provide good visual information for users, although it isn't yet able to be completely automated nor is it a perfect science. Yet with a bit of work and experimentation, one can get a fairly close rendition of what is available on sites like KEGG and have the ability to directly manipulate the graphs and customize the outputs to demonstrate a variety of effects. Hopefully as time goes on, we will be able to provide more in the way of automation in our tools, but even as it exists now, laying out pathways can provide a valuable tool for users.