

BOX 1: Hierarchical and modular organization in co-expressed gene networks.

#--- Step 1: load RedeR

```
library(RedeR)
rdp <- RedPort()
callD(rdp)
```

#--- Step 2: get a data frame and an igraph object (i.e. results from differential and co-expression analyses)

```
data(ER.deg)
dt <- ER.deg$dat
sg <- ER.deg$ceg
```

#--- Step 3: map the data frame to the graph

```
sg <- att.mapv(sg, dat=dt, refcol=1)
```

#--- Step 4: set attributes to RedeR (i.e. gene symbols, fold change and ER binding site distance to TSS)

```
sg <- att.setv(sg, from="Symbol", to="nodeAlias")
sg <- att.setv(sg, from="logFC.t3...t0", to="nodeColor", breaks=seq(-2,2,0.4), pal=2)
sg <- att.setv(sg, from="ERbdist", to="nodeSize", nquant=10, isrev=TRUE, xlim=c(5,40,1))
```

#--- Step 5: add graph to the app

```
addGraph(rdp,sg)
```

#--- Step 6: compute a hierarchical clustering using standard R functions on the adjacency matrix

```
hc <- hclust(dist(get.adjacency(sg, attr="weight")))
```

#--- Step 7: superimpose the dendrogram structure onto the network

```
nesthc(rdp, hc, cutlevel=3, nmemb=5, cex=0.3, labels=V(sg)$nodeAlias)
```

#--- Step 8: assign edges to containers

```
mergeOutEdges(rdp,nlev=2)
```

#--- Step 9: relax the network and fine-tune layout interactively!

```
relax(rdp)
```

#--- Step 10: add color and size legends

```
addLegend.color(rdp, sg, title="diff. gene expression (logFC)")
addLegend.size(rdp, sg, title="bd site distance (kb)")
```

#--- Step 11: reset graph

```
resetD(rdp)
```