

BOX 2: Nested structures in time-course gene expression subnetworks.

#--- Step 1: get a data frame and an interactome

```
data(ER.limma)
data(hs.inter)
dt <- ER.limma
gi <- hs.inter
```

#--- Step 2: extract a subgraph and set attributes to RedeR (i.e. logFC of t3-t0 contrast)

#...p.s. not all genes are expected to be present in the interactome!

```
gt3 <- subg(g=gi, dat=dt[dt$degenes.t3!=0,], refcol=1)
gt3 <- att.setv(g=gt3, from="Symbol", to="nodeAlias")
gt3 <- att.setv(g=gt3, from="logFC.t3...t0", to="nodeColor", breaks=seq(-2,2,0.4), pal=2)
```

#--- Step 3: extract another subgraph and set attributes to RedeR (i.e. logFC of t6-t0 contrast)

```
gt6 <- subg(g=gi, dat=dt[dt$degenes.t6!=0,], refcol=1)
gt6 <- att.setv(g=gt6, from="Symbol", to="nodeAlias")
gt6 <- att.setv(g=gt6, from="logFC.t6...t0", to="nodeColor", breaks=seq(-2,2,0.4), pal=2)
```

#--- Step 4: extract another subgraph set attributes to RedeR (i.e. logFC of t12-t0 contrast)

```
gt12 <- subg(g=gi, dat=dt[dt$degenes.t12!=0,], refcol=1)
gt12 <- att.setv(g=gt12, from="Symbol", to="nodeAlias")
gt12 <- att.setv(g=gt12, from="logFC.t12...t0", to="nodeColor", breaks=seq(-2,2,0.4), pal=2)
```

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

```
addGraph(rdp, gt3, gcoord=c(10,25), gscale=20, isNest=TRUE, theme='tm1', zoom=30)
addGraph(rdp, gt6, gcoord=c(20,70), gscale=50, isNest=TRUE, theme='tm1', zoom=30)
addGraph(rdp, gt12, gcoord=c(70,55), gscale=80, isNest=TRUE, theme='tm1', zoom=30)
```

#--- Step 6: nest subgraphs (i.e. overlap time-series)

```
nestNodes(rdp, nodes=V(gt3)$name, parent="N1", theme='tm2')
nestNodes(rdp, nodes=V(gt6)$name, parent="N2", theme='tm2')
nestNodes(rdp, nodes=V(gt3)$name, parent="N4", theme='tm3')
```

#--- Step 7: assign edges to containers

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

#--- Step 8: relax the network

```
relax(rdp,50,400)
```

#--- Step 9: add node colour legend (ps. same scale for all graphs)

```
scl <- gt3$legNodeColor$scale
leg <- gt3$legNodeColor$legend
addLegend.color(rdp, colvec=scl, labvec=leg, title="diff. gene expression (logFC)")
```

#--- Step 10: select a gene

```
selectNodes(rdp,"RET")
```

#--- Step 11: select and get nodes in a container

```
selectNodes(rdp,"N5")
sg <- getGraph(rdp, status="selected")
```

#--- Step 12: reset graph

```
resetd(rdp)
```