

RSCORE_Tutorials

This is an example of RSCORE. Data comes from NCBI GEO with accession GSE81861. After you download the raw data, we have to do pretreatment. For your data, you can do it by yourself and finally provide a Seurat class object, or you can provide a clean matrix data and use our mat2seurat function.

```
# change the directory to yours
# You need to download the data firstly.
# download.file('https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE81861&format=file&
file=GSE81861%5FCell%5FLine%5FFPKM%2Ecsv%2Egz', destfile = 'RCA_FPKM.csv.gz')
# gunzip('RCA_FPKM.csv.gz', 'RCA_FPKM.csv')
RCA_count <- read.csv('RCA_FPKM.csv', header=T, row.names = 1)
RCA_count <- log(RCA_count+1, 2)
row_names <- strsplit(row.names(RCA_count), '_')
gene_names <- c()
for (i in 1:length(row_names)){
  temp = unlist(row_names[i])
  gene_names[i] = temp[2]
}

row.names(RCA_count) <- make.names(gene_names, unique=TRUE)

RCA_seurat <- CreateSeuratObject(counts = RCA_count, min.cells = 10, min.features = 1000,
                                names.field = 3, names.delim = '_',
                                assay = 'RNA', project = 'RCA')
```

Normalization and feature selection is necessary. Although we have given some default parameters, it depends on your data specifically.

```
RCA_seurat <- ScaleData(object = RCA_seurat)
RCA_seurat <- FindVariableFeatures(object = RCA_seurat, selection.method = 'vst', nfeatures =
8000)
```

PPI data is necessary too. You can provide the adjacent matrix of PPI network by yourself,

```
# change the directory to yours
hs_network <- as.matrix(readRDS(system.file('extdata', '9606_ppi_matrix_BioGRID-
3.5.173.Rda', package = 'RSCORE')))
```

or you can get it by our functions. Then you have to set the parameter ‘PPI’ as ‘String’ or ‘Biogrid’. This means we will download PPI data from STRING or BioGRID (It will cost some time, depends on your Internet speed). Both of these two choices should give the species (default is 9606, Homosapiens).

```
# hs_network <- getPPI_String(object = RCA_seurat, species = 9606, version = '10')
# or
# hs_network <- getPPI_Biogrid(object = RCA_seurat, species = 9606, version = '3.5.173')
```

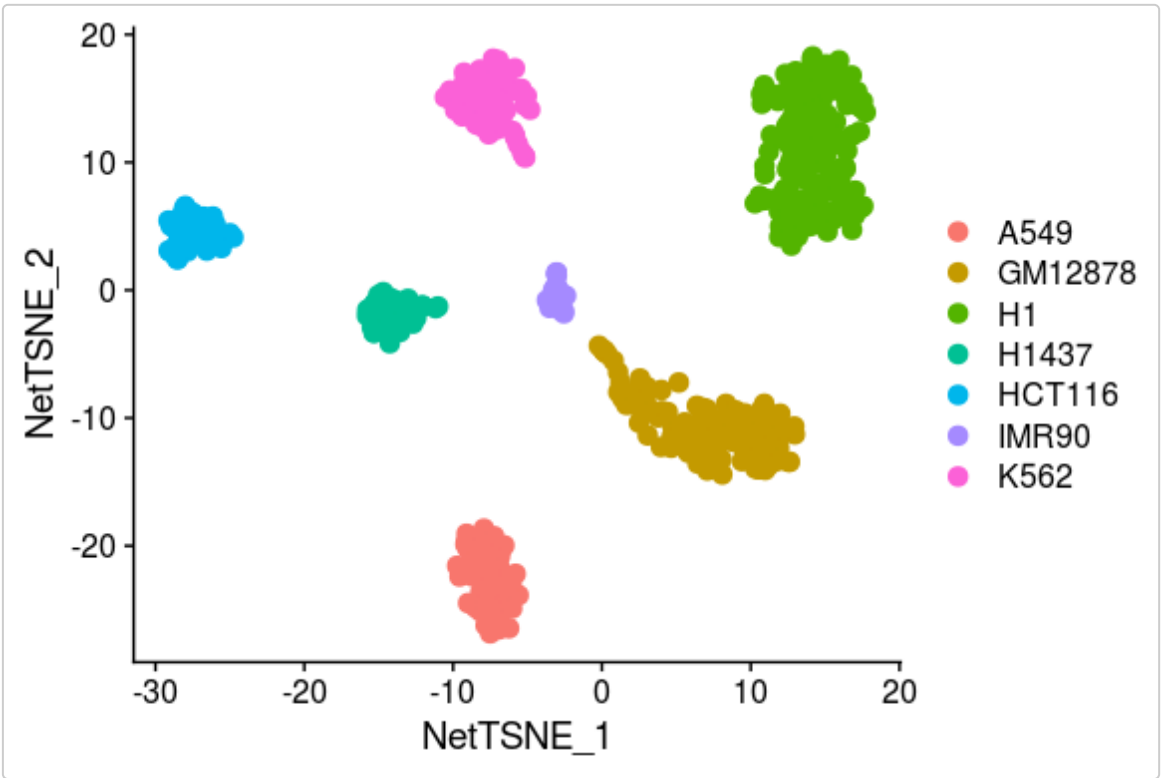
and then the parameter ‘PPI’ is just the matrix.

```
RCA_seurat <- R.SCORE(Data = RCA_seurat, PPI = hs_network)
```

```
#> module num: 1464
```

The result is saved in 'Net' assay of RCA_seurat (it has been set as default assay). You can plot the tsne

```
VariableFeatures(RCA_seurat) <- rownames(RCA_seurat)
RCA_seurat <- RunPCA(RCA_seurat, features = rownames(RCA_seurat), npcs = 30, reduction.name =
"NetPCA",
                    reduction.key = "NetPCA_", verbose = F)
RCA_seurat <- RunTSNE(RCA_seurat, reduction = "NetPCA", dims = 1:10,
                    reduction.name = "NetTSNE", reduction.key = "NetTSNE_")
DimPlot(RCA_seurat, reduction = 'NetTSNE', pt.size = 3, group.by = 'orig.ident')
```



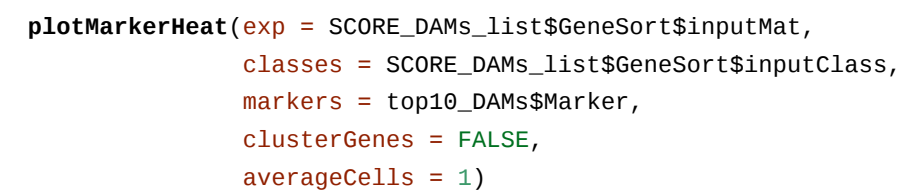
Heatmap of the marker genes and marker modules:

```
library(dplyr)
library(genesorter)
SCORE_DEGs_list <- Find_Markers(object = RCA_seurat, assay = 'RNA', FoldChange = 1.5)

SCORE_DAMs_list <- Find_Markers(object = RCA_seurat, assay = 'Net', FoldChange = 1.5)

#Select the top n markers of each cluster
top10_DEGs <- SCORE_DEGs_list$Markers %>% group_by(Cluster) %>% top_n(n = 10, wt = Gene.Score)
top10_DAMs <- SCORE_DAMs_list$Markers %>% group_by(Cluster) %>% top_n(n = 10, wt = Gene.Score)

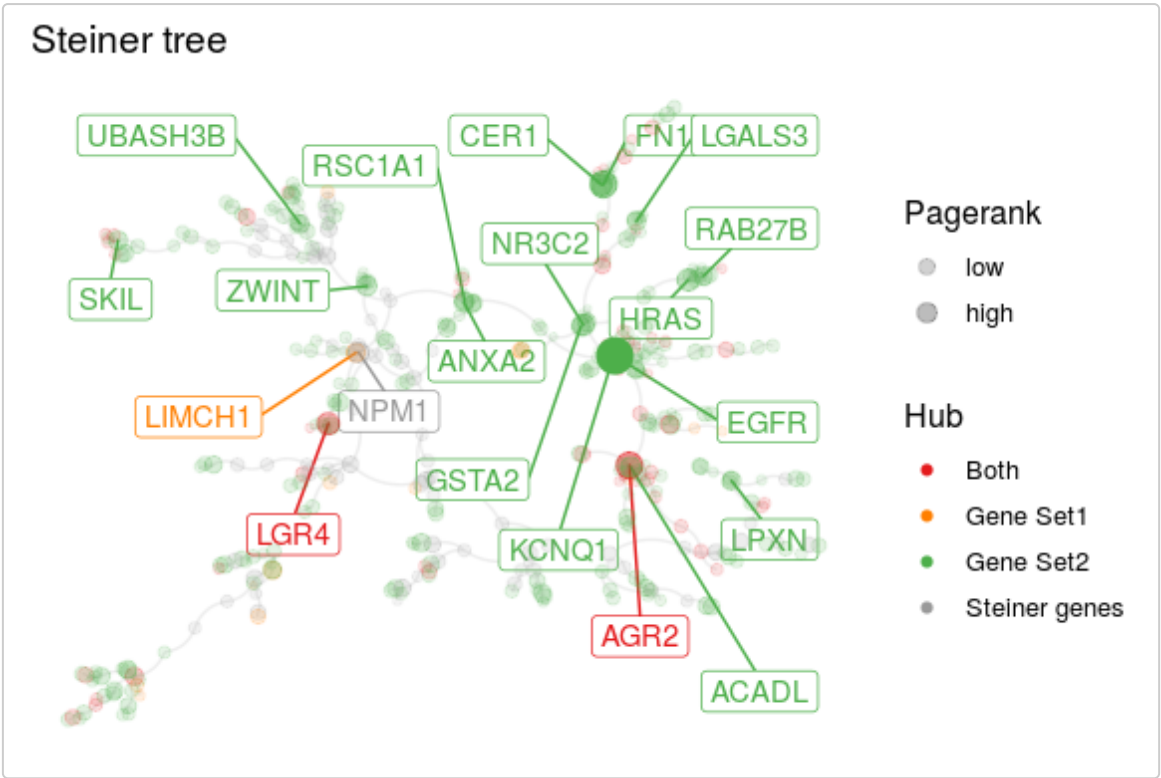
#genesorter plotMarkerHeat function
plotMarkerHeat(exp = SCORE_DEGs_list$GeneSort$inputMat,
               classes = SCORE_DEGs_list$GeneSort$inputClass,
               markers = top10_DEGs$Marker,
               clusterGenes = FALSE,
               averageCells = 1)
```



8/10/19, 7:18 PM

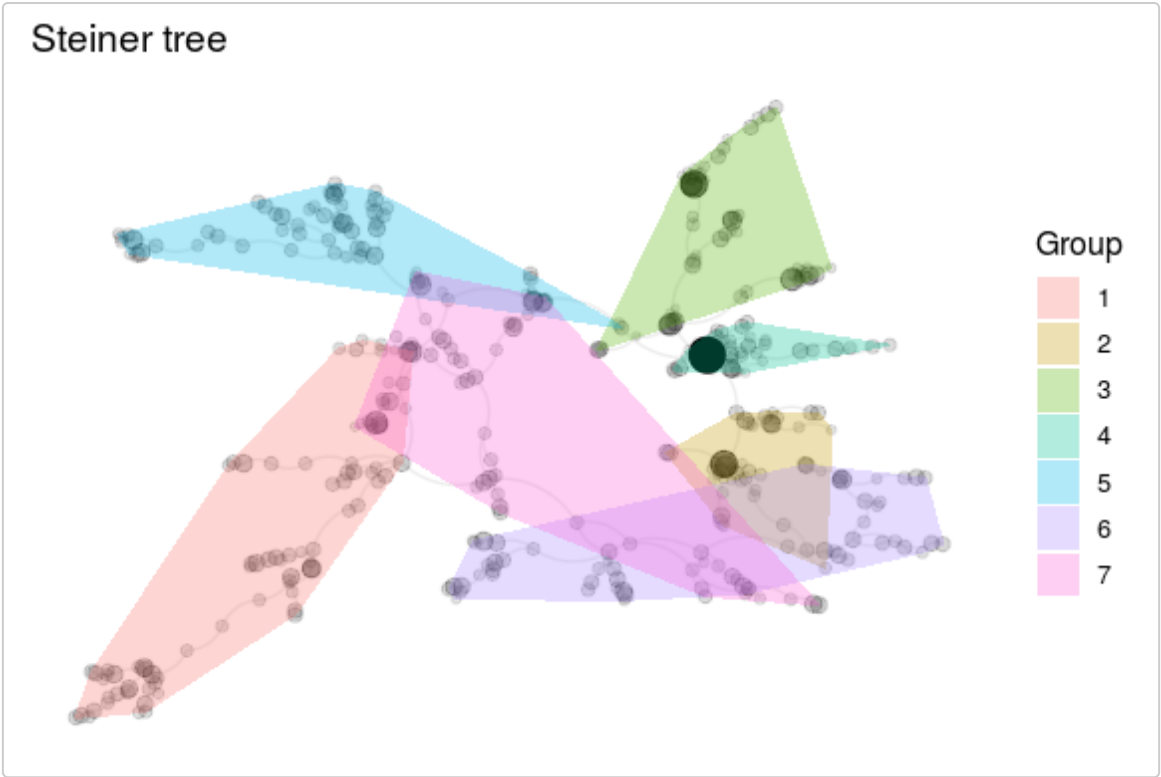
```
DEGs <- SCORE_DEGs_list$Markers[SCORE_DEGs_list$Markers$Cluster==ident,]$Marker
DAMs <- SCORE_DAMs_list$Markers[SCORE_DAMs_list$Markers$Cluster==ident,]$Marker
DAMGs <- unique(rownames(table(unlist(RCA_seurat@misc$geneSets[DAMs]))))

st_res <- PlotSteinertree(RCA_seurat, geneset1 = DEGs, geneset2 = DAMGs)
#> calculate tree
print(st_res$plot)
```



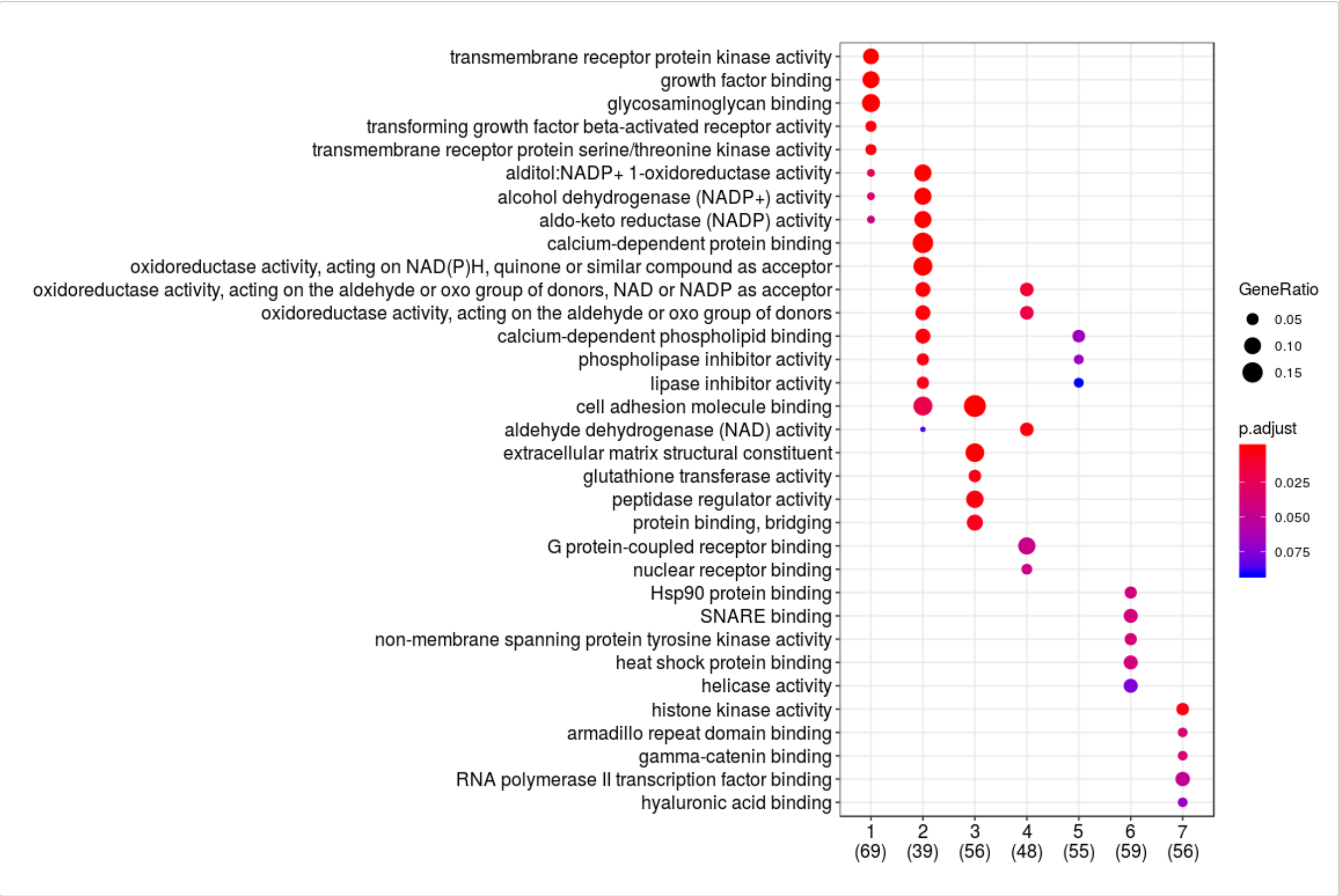
And then you can do GO enrichment analysis of the genes in the Steiner tree. First you need to divide the Steiner tree into several groups,

```
st_res <- cut_steiner_tree(st_res, k = 7)
```



then you can do GO enrichment analysis of all the groups

```
en_res_all <- get_enrich_plot(st_res)
print(en_res_all$plot)
```



or a specified group of genes

```
en_res <- get_enrich_plot(st_res, group = 1)
print(en_res$plot)
```

Steiner tree

