Tutorial for the WGCNA package for R:
I. Network analysis of liver expression data in female mice

6. Exporting a gene network to external visualization software

Peter Langfelder and Steve Horvath

November 25, 2014

Contents

1 Preliminaries: setting up the R session and loading results of previous parts **1**

6 Exporting network data to network visualization software **2**
6.a Exporting to VisANT 2
6.b Exporting to Cytoscape 2

1 Preliminaries: setting up the R session and loading results of previous parts

Here we assume that a new R session has just been started. We load the WGCNA package, set up basic parameters and load data saved in previous parts of the tutorial.

```
# Display the current working directory
getwd();
# If necessary, change the path below to the directory where the data files are stored.
# "." means current directory. On Windows use a forward slash / instead of the usual \.
workingDir = ".";
setwd(workingDir);
# Load the WGCNA package
library(WGCNA)
# The following setting is important, do not omit.
options(stringsAsFactors = FALSE);
# Load the expression and trait data saved in the first part
lnames = load(file = "FemaleLiver-01-dataInput.RData");
#The variable lnames contains the names of loaded variables.
lnames
# Load network data saved in the second part.
lnames = load(file = "FemaleLiver-02-networkConstruction-auto.RData");
lnames
```

We use the network file obtained by the step-by-step network construction and module detection; we encourage the reader to use the results of the other approaches as well.

6 Exporting network data to network visualization software

6.a Exporting to VisANT

The package provides a convenient function for exporting the network to VisANT [1]. We illustrate a simple export of the full weighted network of a single module.

```
# Recalculate topological overlap
TOM = TOMsimilarityFromExpr(datExpr, power = 6);
# Read in the annotation file
annot = read.csv(file = "GeneAnnotation.csv");
# Select module
module = "brown";
# Select module probes
probes = names(datExpr)
inModule = (moduleColors==module);
modProbes = probes[inModule];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into an edge list file VisANT can read
vis = exportNetworkToVisANT(modTOM,
  file = paste("VisANTInput-", module, ".txt", sep=""),
  weighted = TRUE,
  threshold = 0,
  probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol) )
```

Because the brown module is rather large, we can restrict the genes in the output to say the 30 top hub genes in the module:

```
nTop = 30;
IMConn = softConnectivity(datExpr[, modProbes]);
top = (rank(-IMConn) <= nTop)
vis = exportNetworkToVisANT(modTOM[top, top],
  file = paste("VisANTInput-", module, "-top30.txt", sep=""),
  weighted = TRUE,
  threshold = 0,
  probeToGene = data.frame(annot$substanceBXH, annot$gene_symbol) )
```

To provide an example of a VisANT visualization, we loaded the file produced by the above code in VisANT. For better readability we varied the threshold for displaying a link between two nodes. The results are shown in Fig. 1.

6.b Exporting to Cytoscape

Cytoscape [2] allows the user to input an edge file and a node file, allowing the user to specify for example the link weights and the node colors. Here we demonstrate the output of two modules, the red and brown ones, to Cytoscape.

```
# Recalculate topological overlap if needed
TOM = TOMsimilarityFromExpr(datExpr, power = 6);
# Read in the annotation file
annot = read.csv(file = "GeneAnnotation.csv");
# Select modules
modules = c("brown", "red");
# Select module probes
probes = names(datExpr)
inModule = is.finite(match(moduleColors, modules));
modProbes = probes[inModule];
modGenes = annot$gene_symbol[match(modProbes, annot$substanceBXH)];
# Select the corresponding Topological Overlap
modTOM = TOM[inModule, inModule];
```

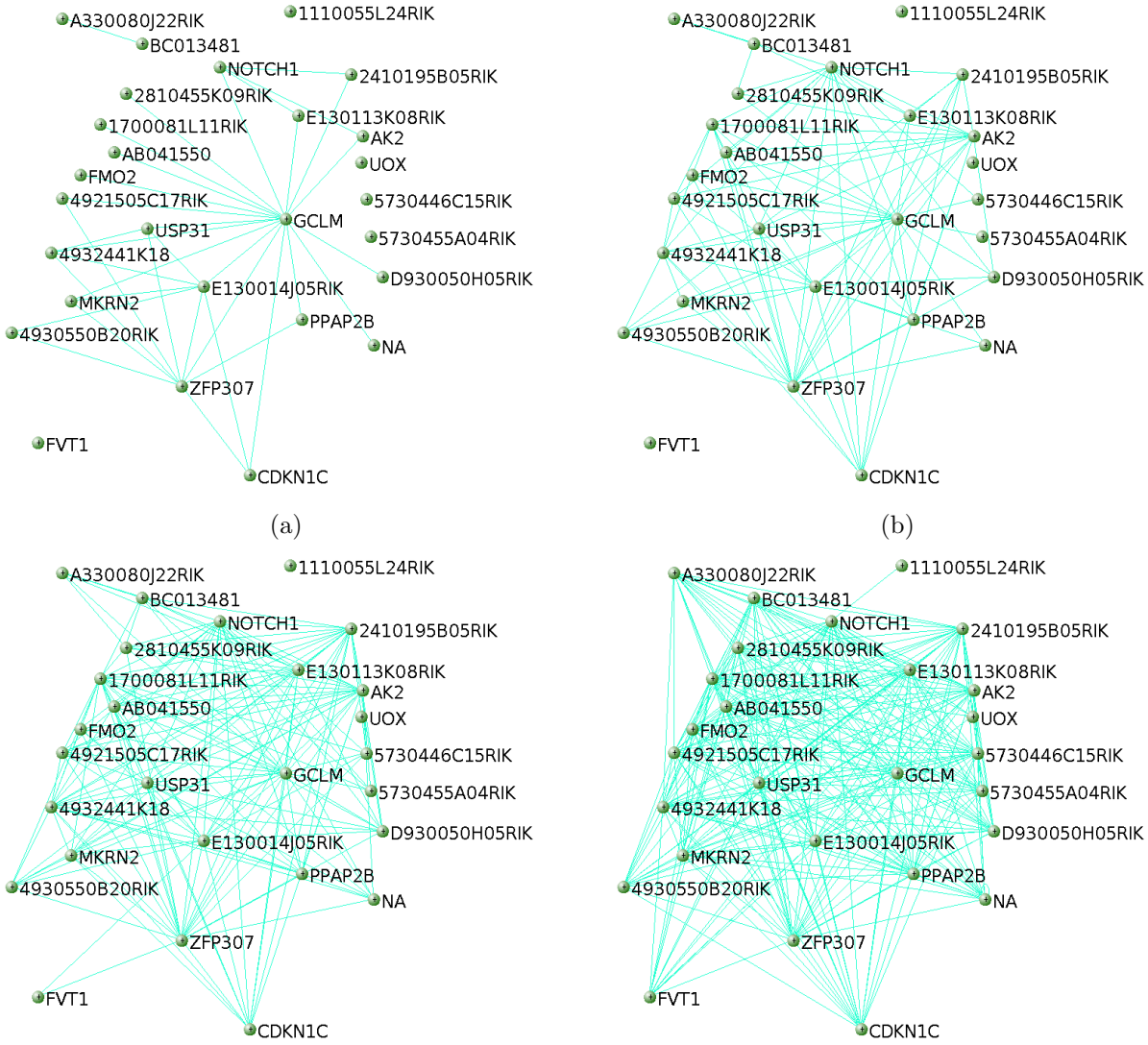

Figure 1: Visualization of the network connections among the most connected genes in the brown module, generated by the VisANT software. The four plots (a), (b), (c), and (d) show network connections whose topological overlap is above the thresholds of 0.10, 0.08, 0.06, and 0.03.

```
dimnames(modTOM) = list(modProbes, modProbes)
# Export the network into edge and node list files Cytoscape can read
cyt = exportNetworkToCytoscape(modTOM,
  edgeFile = paste("CytoscapeInput-edges-", paste(modules, collapse="-"), ".txt", sep=""),
  nodeFile = paste("CytoscapeInput-nodes-", paste(modules, collapse="-"), ".txt", sep=""),
  weighted = TRUE,
  threshold = 0.02,
  nodeNames = modProbes,
  altNodeNames = modGenes,
  nodeAttr = moduleColors[inModule]);
```

Note that network input to Cytoscape is a bit more involved and the user should take care to select all necessary options for the edge and node files to be interpreted correctly. We refer the reader to Cytoscape documentation for all the necessary details.

References

[1] Zhenjun Hu, Evan S. Snitkin, and Charles DeLisi. VisANT: an integrative framework for networks in systems biology. *Brief Bioinform*, 9(4):317–325, 2008.

[2] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, 13(11):2498–2504, 2003.