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

5. Network visualization using WGCNA functions

Peter Langfelder and Steve Horvath

November 25, 2014

Contents

0 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
nGenes = ncol(datExpr)
nSamples = nrow(datExpr)
```

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.

5 Visualization of networks within R

5.a Visualizing the gene network

One way to visualize a weighted network is to plot its heatmap, Fig. 1. Each row and column of the heatmap correspond to a single gene. The heatmap can depict adjacencies or topological overlaps, with light colors denoting low adjacency (overlap) and darker colors higher adjacency (overlap). In addition, the gene dendrograms and module colors are plotted along the top and left side of the heatmap. The package provides a convenient function to create such network plots; Fig. 1 was created using the following code. This code can be executed only if the network was calculated using a single-block approach (that is, using the 1-step automatic or the step-by-step tutorials). If the networks were calculated using the block-wise approach, the user will need to modify this code to perform the visualization in each block separately. The modification is simple and we leave it as an exercise for the interested reader.

```
# Calculate topological overlap anew: this could be done more efficiently by saving the TOM
# calculated during module detection, but let us do it again here.
dissTOM = 1-TOMsimilarityFromExpr(datExpr, power = 6);
# Transform dissTOM with a power to make moderately strong connections more visible in the heatmap
plotTOM = dissTOM^7;
# Set diagonal to NA for a nicer plot
diag(plotTOM) = NA;
# Call the plot function
sizeGrWindow(9,9)
TOMplot(plotTOM, geneTree, moduleColors, main = "Network heatmap plot, all genes")
```

Note that the generating the heatmap plot may take a substantial amount of time. It is possible to restrict the number of genes to speed up the plotting; however, the gene dendrogram of a subset of genes will often look different from the gene dendrogram of all genes. In the following example we restrict the number of plotted genes to 400:

```
nSelect = 400
# For reproducibility, we set the random seed
set.seed(10);
select = sample(nGenes, size = nSelect);
selectTOM = dissTOM[select, select];
# There's no simple way of restricting a clustering tree to a subset of genes, so we must re-cluster.
selectTree = hclust(as.dist(selectTOM), method = "average")
selectColors = moduleColors[select];
# Open a graphical window
sizeGrWindow(9,9)
# Taking the dissimilarity to a power, say 10, makes the plot more informative by effectively changing
# the color palette; setting the diagonal to NA also improves the clarity of the plot
plotDiss = selectTOM^7;
diag(plotDiss) = NA;
TOMplot(plotDiss, selectTree, selectColors, main = "Network heatmap plot, selected genes")
```

5.b Visualizing the network of eigengenes

It is often interesting to study the relationships among the found modules. One can use the eigengenes as representative profiles and quantify module similarity by eigengene correlation. The package contains a convenient function `plotEigengeneNetworks` that generates a summary plot of the eigengene network. It is usually informative to add a clinical trait (or multiple traits) to the eigengenes to see how the traits fit into the eigengene network:

```
# Recalculate module eigengenes
MEs = moduleEigengenes(datExpr, moduleColors)$eigengenes
# Isolate weight from the clinical traits
weight = as.data.frame(datTraits$weight_g);
names(weight) = "weight"
# Add the weight to existing module eigengenes
```

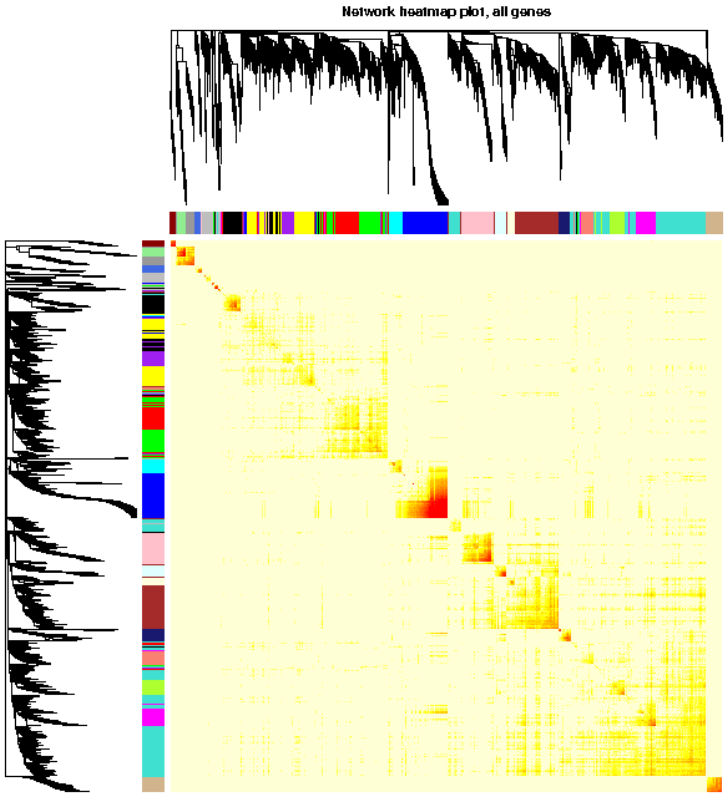

Figure 1: Visualizing the gene network using a heatmap plot. The heatmap depicts the Topological Overlap Matrix (TOM) among all genes in the analysis. Light color represents low overlap and progressively darker red color represents higher overlap. Blocks of darker colors along the diagonal are the modules. The gene dendrogram and module assignment are also shown along the left side and the top.

```
MET = orderMEs(cbind(MEs, weight))
# Plot the relationships among the eigengenes and the trait
sizeGrWindow(5,7.5);
par(cex = 0.9)
plotEigengeneNetworks(MET, "", marDendro = c(0,4,1,2), marHeatmap = c(3,4,1,2), cex.lab = 0.8, xLabelsAngle
= 90)
```

The function produces a dendrogram of the eigengenes and trait(s), and a heatmap of their relationships. To split the dendrogram and heatmap plots, we can use the following code

```
# Plot the dendrogram
sizeGrWindow(6,6);
par(cex = 1.0)
plotEigengeneNetworks(MET, "Eigengene dendrogram", marDendro = c(0,4,2,0),
                      plotHeatmaps = FALSE)
# Plot the heatmap matrix (note: this plot will overwrite the dendrogram plot)
par(cex = 1.0)
plotEigengeneNetworks(MET, "Eigengene adjacency heatmap", marHeatmap = c(3,4,2,2),
                      plotDendrograms = FALSE, xLabelsAngle = 90)
```

Fig. 2 shows the output of the above code. The eigengene dendrogram and heatmap identify groups of correlated eigengenes termed *meta-modules*. For example, the dendrogram indicates that red, brown and bluw modules are highly related; their mutual correlations are stronger than their correlations with weight. On the other hand, the salmon module, which is also significantly correlated with weight, is not part of the same meta-module as the red,

brown and blue modules, at least if meta-modules are defined as tight custers of modules (for example, modules with a correlation of eigengenes of at least 0.5).

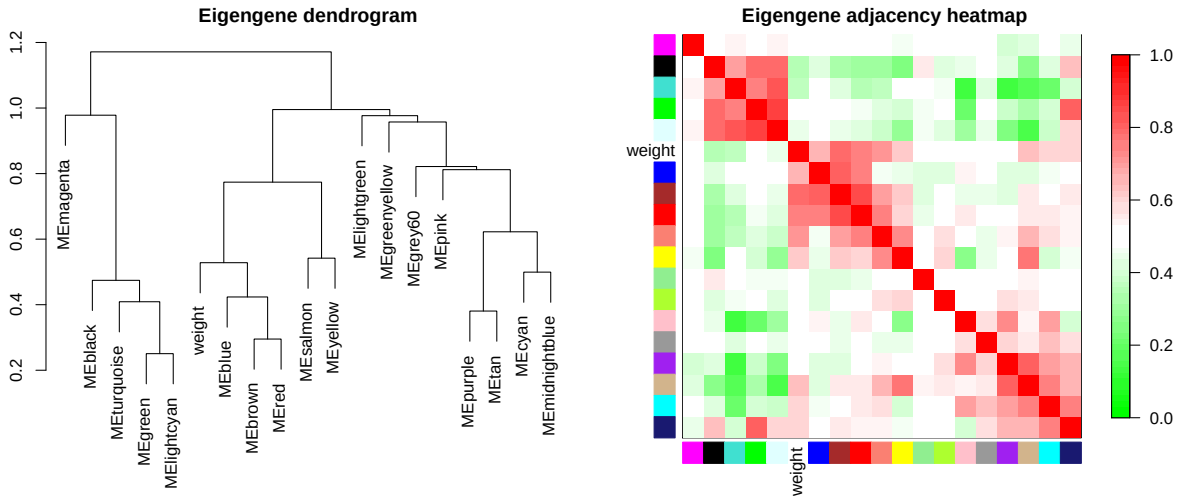

Figure 2: Visualization of the eigengene network representing the relationships among the modules and the clinical trait weight. Panel (a) shows a hierarchical clustering dendrogram of the eigengenes in which the dissimilarity of eigengenes E_I, E_J is given by $1 - \mathrm{cor}(E_I, E_J)$. The heatmap in panel (b) shows the eigengene adjacency $A_{IJ} = (1 + \mathrm{cor}(E_I, E_J))/2$.