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

3. Relating modules to external information and identifying important genes

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
```

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.

3 Relating modules to external clinical traits

3.a Quantifying module–trait associations

In this analysis we would like to identify modules that are significantly associated with the measured clinical traits. Since we already have a summary profile (*eigengene*) for each module, we simply correlate eigengenes with external traits and look for the most significant associations:

```
# Define numbers of genes and samples
nGenes = ncol(datExpr);
nSamples = nrow(datExpr);
# Recalculate MEs with color labels
MEs0 = moduleEigengenes(datExpr, moduleColors)$eigengenes
MEs = orderMEs(MEs0)
moduleTraitCor = cor(MEs, datTraits, use = "p");
moduleTraitPvalue = corPvalueStudent(moduleTraitCor, nSamples);
```

Since we have a moderately large number of modules and traits, a suitable graphical representation will help in reading the table. We color code each association by the correlation value:

```
sizeGrWindow(10,6)
# Will display correlations and their p-values
textMatrix = paste(signif(moduleTraitCor, 2), "\n(",
                        signif(moduleTraitPvalue, 1), ")", sep = "");
dim(textMatrix) = dim(moduleTraitCor)
par(mar = c(6, 8.5, 3, 3));
# Display the correlation values within a heatmap plot
labeledHeatmap(Matrix = moduleTraitCor,
             xLabels = names(datTraits),
             yLabels = names(MEs),
             ySymbols = names(MEs),
             colorLabels = FALSE,
             colors = greenWhiteRed(50),
             textMatrix = textMatrix,
             setStdMargins = FALSE,
             cex.text = 0.5,
             zlim = c(-1,1),
             main = paste("Module-trait relationships"))
```

The resulting color-coded table is shown in Fig. 1.
The analysis identifies the several significant module–trait associations. We will concentrate on weight as the trait of interest.

3.b Gene relationship to trait and important modules: Gene Significance and Module Membership

We quantify associations of individual genes with our trait of interest (weight) by defining Gene Significance GS as (the absolute value of) the correlation between the gene and the trait. For each module, we also define a quantitative measure of module membership MM as the correlation of the module eigengene and the gene expression profile. This allows us to quantify the similarity of all genes on the array to every module.

```
# Define variable weight containing the weight column of datTrait
weight = as.data.frame(datTraits$weight_g);
names(weight) = "weight"
# names (colors) of the modules
modNames = substring(names(MEs), 3)

geneModuleMembership = as.data.frame(cor(datExpr, MEs, use = "p"));
MMPvalue = as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership), nSamples));
```

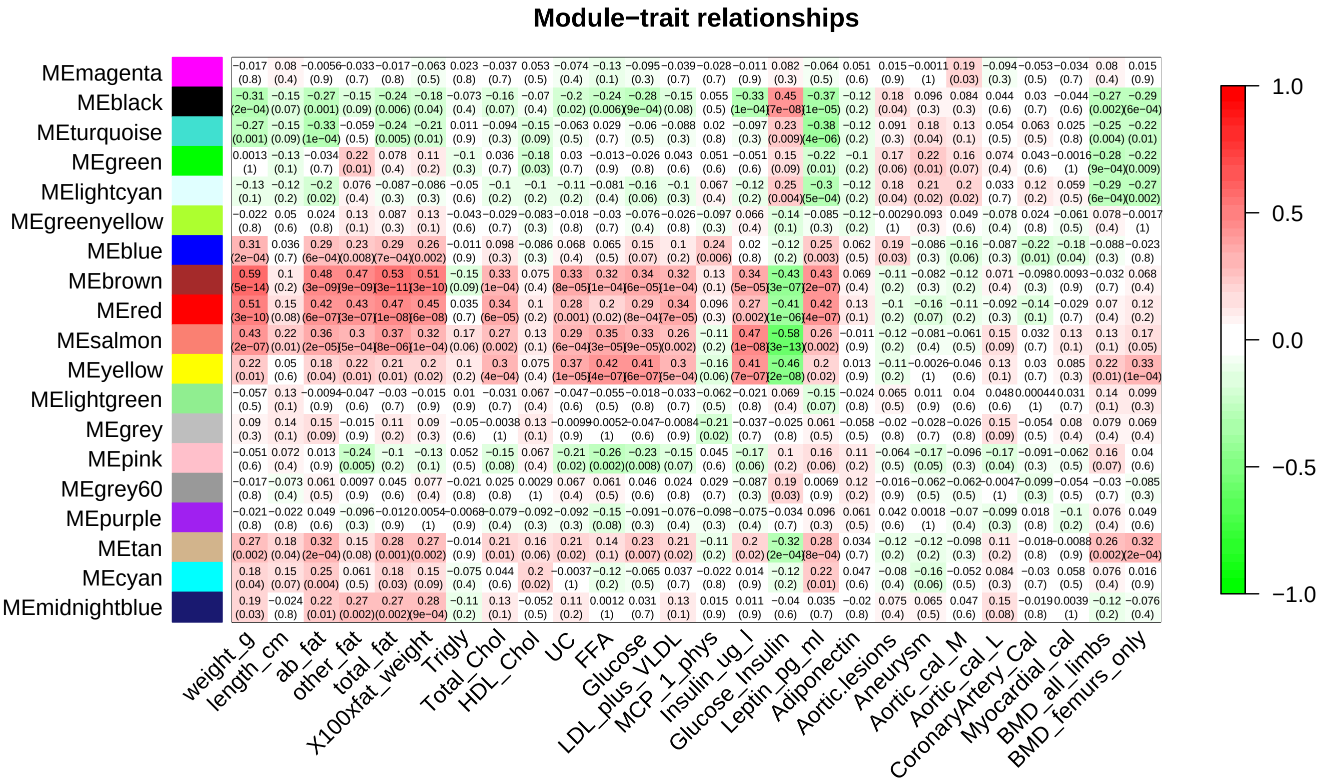

Figure 1: Module-trait associations. Each row corresponds to a module eigengene, column to a trait. Each cell contains the corresponding correlation and p-value. The table is color-coded by correlation according to the color legend.

```
names(geneModuleMembership) = paste("MM", modNames, sep="");
names(MMPvalue) = paste("p.MM", modNames, sep="");

geneTraitSignificance = as.data.frame(cor(datExpr, weight, use = "p"));
GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance), nSamples));

names(geneTraitSignificance) = paste("GS.", names(weight), sep="");
names(GSPvalue) = paste("p.GS.", names(weight), sep="");
```

3.c Intramodular analysis: identifying genes with high GS and MM

Using the GS and MM measures, we can identify genes that have a high significance for weight as well as high module membership in interesting modules. As an example, we look at the brown module that has the highest association with weight. We plot a scatterplot of Gene Significance vs. Module Membership in the brown module:

```
module = "brown"
column = match(module, modNames);
moduleGenes = moduleColors==module;

sizeGrWindow(7, 7);
par(mfrow = c(1,1));
verboseScatterplot(abs(geneModuleMembership[moduleGenes, column]),
                 abs(geneTraitSignificance[moduleGenes, 1]),
                 xlab = paste("Module Membership in", module, "module"),
                 ylab = "Gene significance for body weight",
                 main = paste("Module membership vs. gene significance\n"),
                 cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
```

The plot is shown in Fig. 2. Clearly, GS and MM are highly correlated, illustrating that genes highly significantly associated with a trait are often also the most important (central) elements of modules associated with the trait. The reader is encouraged to try this code with other significance trait/module correlation (for example, the magenta, midnightblue, and red modules with weight).

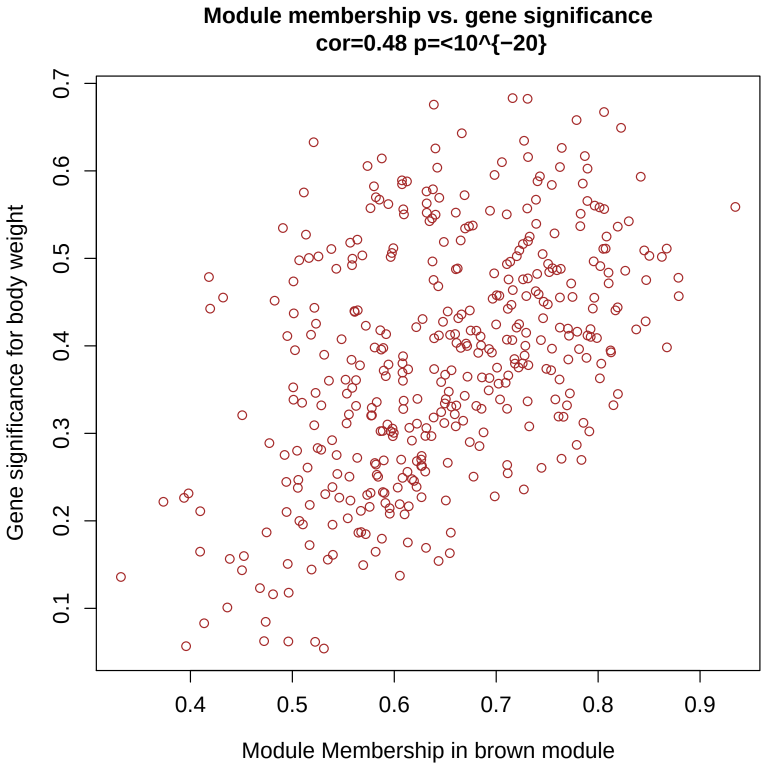

Figure 2: A scatterplot of Gene Significance (GS) for weight vs. Module Membership (MM) in the brown module. There is a highly significant correlation between GS and MM in this module.

3.d Summary output of network analysis results

We have found modules with high association with our trait of interest, and have identified their central players by the Module Membership measure. We now merge this statistical information with gene annotation and write out a file that summarizes the most important results and can be inspected in standard spreadsheet software such as MS Excel or Open Office Calc. Our expression data are only annotated by probe ID names: the command

```
names(datExpr)
```

will return all probe IDs included in the analysis. Similarly,

```
names(datExpr)[moduleColors=="brown"]
```

will return probe IDs belonging to the brown module. To facilitate interpretation of the results, we use a probe annotation file provided by the manufacturer of the expression arrays to connect probe IDs to gene names and universally recognized identification numbers (Entrez codes).

```
annot = read.csv(file = "GeneAnnotation.csv");
dim(annot)
names(annot)
probes = names(datExpr)
probes2annot = match(probes, annot$substanceBXH)
# The following is the number or probes without annotation:
sum(is.na(probes2annot))
# Should return 0.
```

We now create a data frame holding the following information for all probes: probe ID, gene symbol, Locus Link ID (Entrez code), module color, gene significance for weight, and module membership and p-values in all modules. The modules will be ordered by their significance for weight, with the most significant ones to the left.

```
# Create the starting data frame
geneInfo0 = data.frame(substanceBXH = probes,
                      geneSymbol = annot$gene_symbol[probes2annot],
                      LocusLinkID = annot$LocusLinkID[probes2annot],
                      moduleColor = moduleColors,
                      geneTraitSignificance,
                      GSPvalue)
# Order modules by their significance for weight
modOrder = order(-abs(cor(MEs, weight, use = "p")));
# Add module membership information in the chosen order
for (mod in 1:ncol(geneModuleMembership))
{
  oldNames = names(geneInfo0)
  geneInfo0 = data.frame(geneInfo0, geneModuleMembership[, modOrder[mod]],
                         MMPvalue[, modOrder[mod]]);
  names(geneInfo0) = c(oldNames, paste("MM.", modNames[modOrder[mod]], sep=""),
                       paste("p.MM.", modNames[modOrder[mod]], sep=""))
}
# Order the genes in the geneInfo variable first by module color, then by geneTraitSignificance
geneOrder = order(geneInfo0$moduleColor, -abs(geneInfo0$GS.weight));
geneInfo = geneInfo0[geneOrder, ]
```

This data frame can be written into a text-format spreadsheet, for example by

```
write.csv(geneInfo, file = "geneInfo.csv")
```

The reader is encouraged to open and view the file in a spreadsheet software, or inspect it directly within R using the command `fix(geneInfo)`.