WGCNA-part 3: External data correlation

Rachel Richardson

January 29, 2019

Now that we've figured out what lipids we're interested in and and have our WGCNA matrix for our RNA samples, we can start correlating the data.

First, we need to retrieve the objects we saved earlier in parts one and two. If you saved other compoments, make sure these are loaded as well.

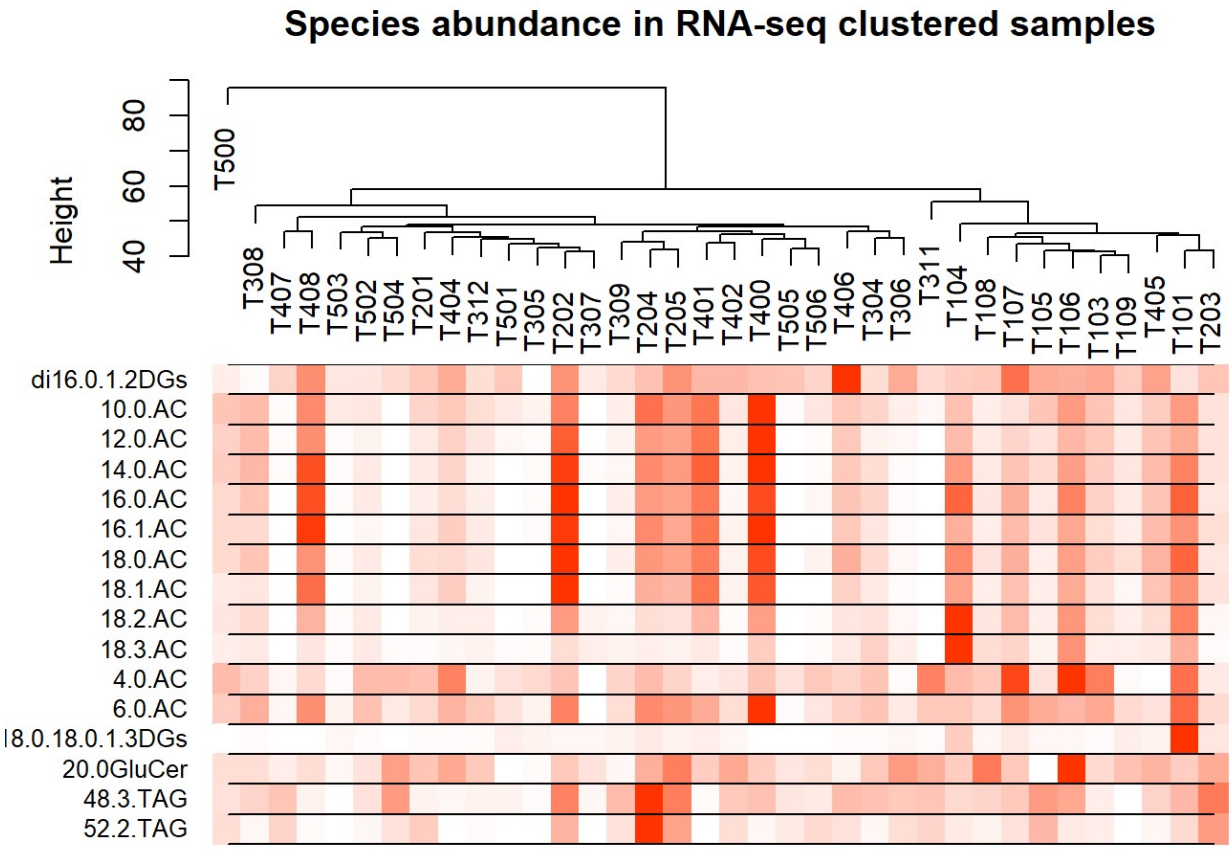
```
library(WGCNA)  #Don't forget to have the WGCNA library installed!
```

```
## ======================================================================
## *
## *  Package WGCNA 1.66 loaded.
## *
## ======================================================================
```

```
Mat.W <- readRDS(file = "Mat.W.rds")
dissTOM <- readRDS(file = "dissTOM.rds")
Goodsamps <- readRDS(file = "Goodsamps.rds")
```

Our first comparison will be looking at how our external data (lipid abundance) is concentrated across all of our subjects.

```
#Clustering of samples by RNA data, as in part 1
sampleTree1 <- hclust(dist(Goodsamps[122:length(Goodsamps)]), method = "average")

#Make a list of lipid species of interest from Mat.W
specieslist <- row.names(Mat.W)


# Shows abundance levels in coloration for each sample in selected lipid species
traitColors <-  numbers2colors(Goodsamps[,specieslist], signed = FALSE)
plotDendroAndColors(sampleTree1, traitColors, groupLabels = specieslist, main = "Speci
es abundance in RNA-seq clustered samples")
```



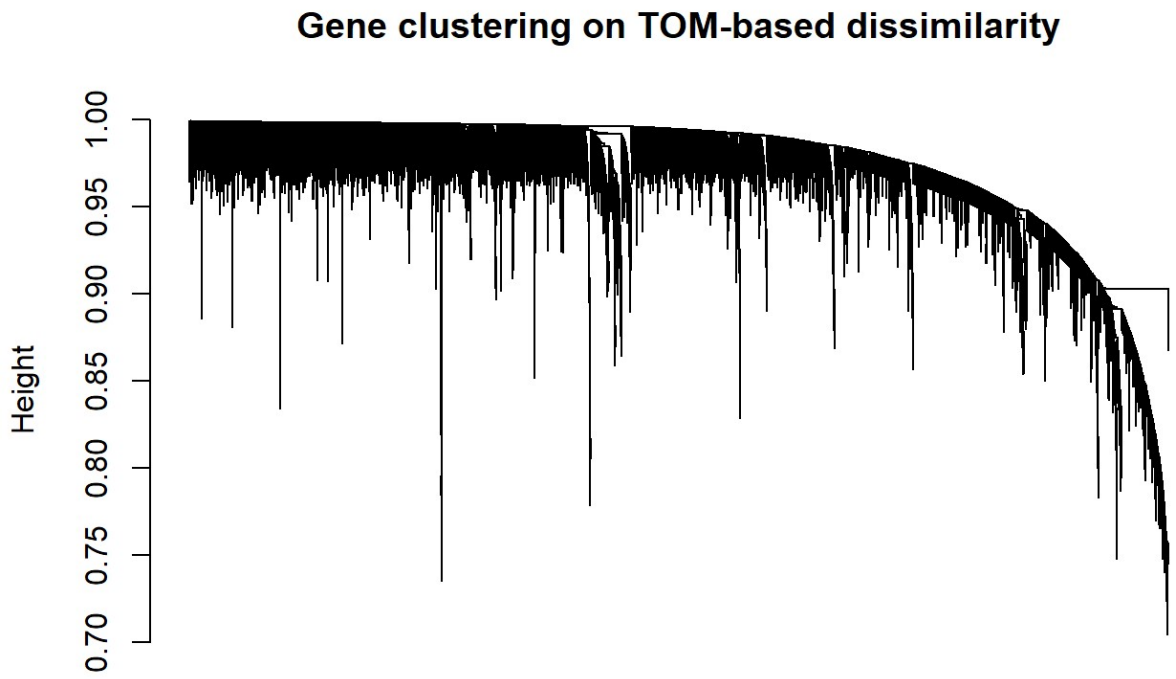

```
#Red = high, white = low
```

This step also helps inform other potential outliers in terms of external data across samples. Notice how the 1.3 DGs lipid is highly abundant in T101, which might be a point to investigate in terms of possible errors.

With no evidence suggesting that this number is an error, we continue moving forward in our analysis.

With the matrix dissTOM, we can make similar dendograms with genes and create modules for similarly expressing genes.

```
#Plot dissimilarities based on adjacency and topological overlap as a gene tree
geneTree = hclust(as.dist(dissTOM), method = "average")

plot(geneTree, xlab="", sub="", main = "Gene clustering on TOM-based dissimilarity", l
abels = FALSE, hang = 0.04)
```



```
minModuleSize = 20; #Larger modules and less modules, easier to explore with larger mi
nimum

# Module identification using dynamic tree cut:
dynamicMods = cutreeDynamic(dendro = geneTree, distM = dissTOM, deepSplit = 2, pamResp
ectsDendro = FALSE, minClusterSize = minModuleSize);
```

```
##  ..cutHeight not given, setting it to 0.998  ===>  99% of the (truncated) height ra
nge in dendro.
##  ..done.
```

```
table(dynamicMods)
```

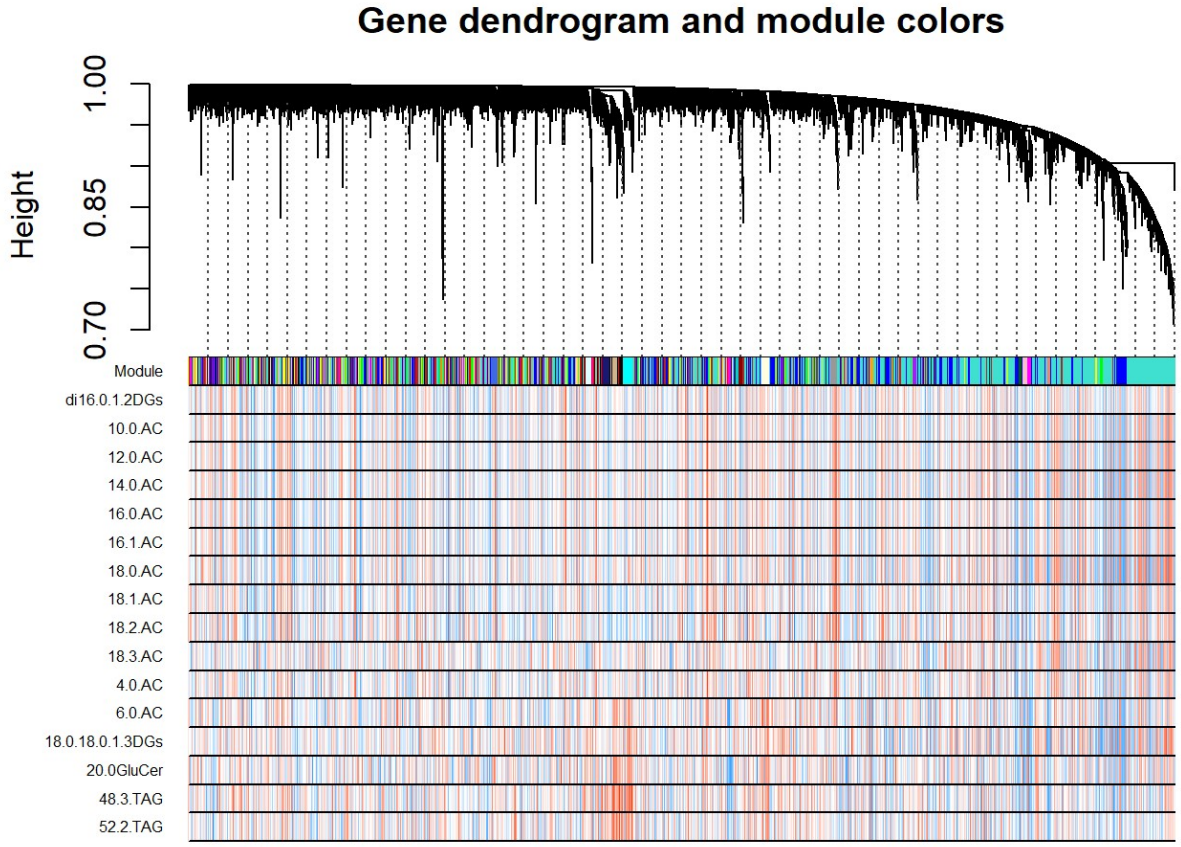
```
## dynamicMods
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
##   82 6422 3626 1773  830  570  565  363  342  313  267  266  219  199  178
##   15   16   17   18   19   20   21   22   23   24   25   26   27   28
##  163  133  125  117  115  107  100   82   80   71   56   50   45   41
```

At this point, We can plot the correlation of lipids to the gene dendogram, much like the subject dendogram.

```
# Convert numeric lables into colors for modules
dynamicColors = labels2colors(dynamicMods)
table(dynamicColors)
```

```
## dynamicColors
##         black          blue         brown          cyan     darkgreen
##           363          3626          1773           178            82
##      darkgrey    darkorange       darkred darkturquoise         green
##            71            50           100            80           570
##   greenyellow          grey        grey60     lightcyan    lightgreen
##           266            82           125           133           117
##   lightyellow       magenta  midnightblue        orange          pink
##           115           313           163            56           342
##        purple           red     royalblue        salmon       skyblue
##           267           565           107           199            41
##           tan     turquoise         white        yellow
##           219          6422            45           830
```

```
datColors=data.frame(dynamicColors)

#Make similar determinations to above using the different lipid species compared to th
e modules

for(column in 1:121){
  varcor <- Goodsamps[column]
  names(varcor) <- colnames(Goodsamps)[column]
  GS.varcor <- as.numeric(bicor(Goodsamps[122:length(Goodsamps)], varcor, use =
"p"))
  GS.weightColor <- numbers2colors(GS.varcor, signed = T)
  datColors <- cbind(datColors, GS.weightColor)
}

#Make an index of selected lipids that corresponded to datColors list
#Note that the index will be one less than the index in datColors with the inclusion o
f a mudule color row
specieslistindex <- which(!is.na(match(colnames(Goodsamps), specieslist)))

# Plot the dendrogram and colors underneath
plotDendroAndColors(geneTree, cbind(datColors[1], datColors[specieslistindex+1]), c("M
odule", specieslist), dendroLabels = FALSE, hang = 0.03, addGuide = TRUE, guideHang =
0.05, main = "Gene dendrogram and module colors", cex.colorLabels = 0.5)
```

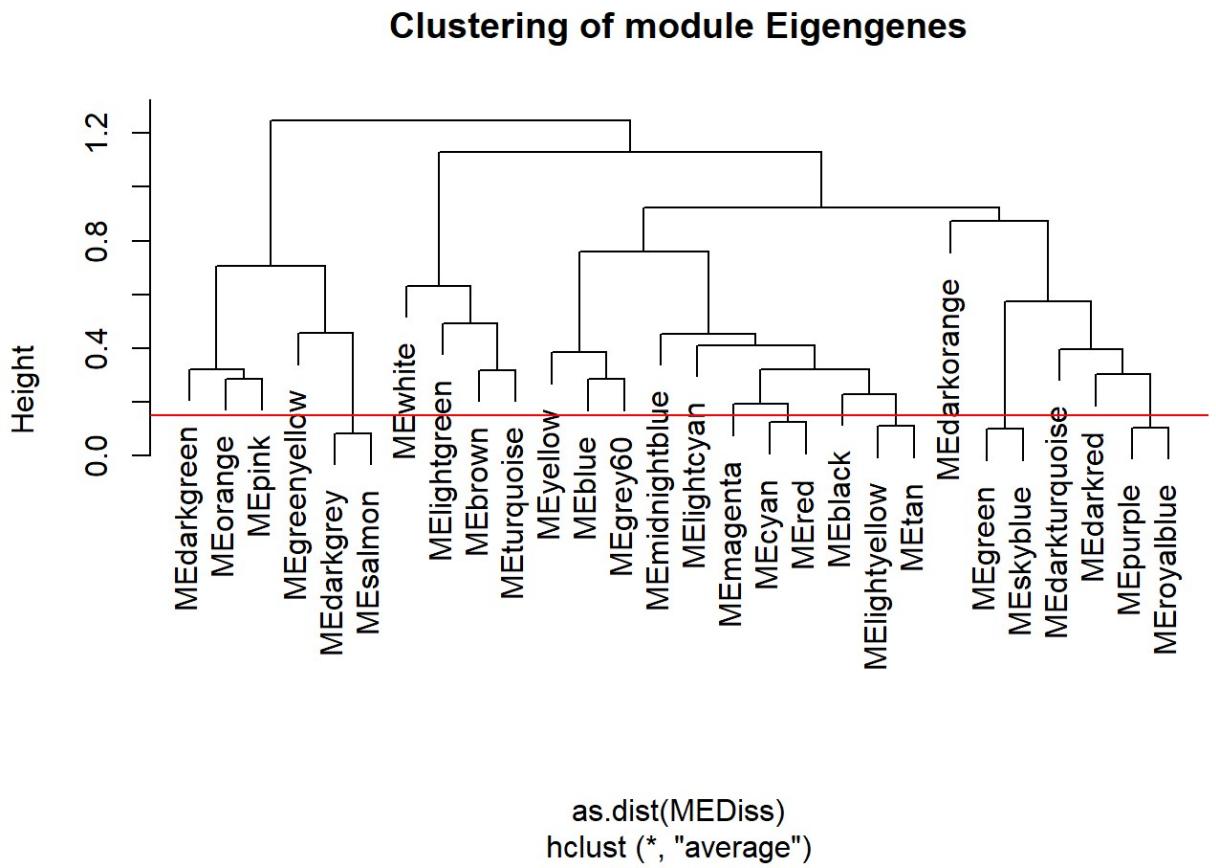



```
#Red = high correlation, blue/green = low, white = 0
```

This is where the avoidance of multiple testing is needed. We will create false genes, “eigengenes,” to correlate the lipids to that are representative of the trends in each module. However, some modules are very similar to other modules, so not all of them are nessesarily informative independently.

We calculate eigengenes and merge highly similar modules (merging correlated modules based on their eigengens >0.85)

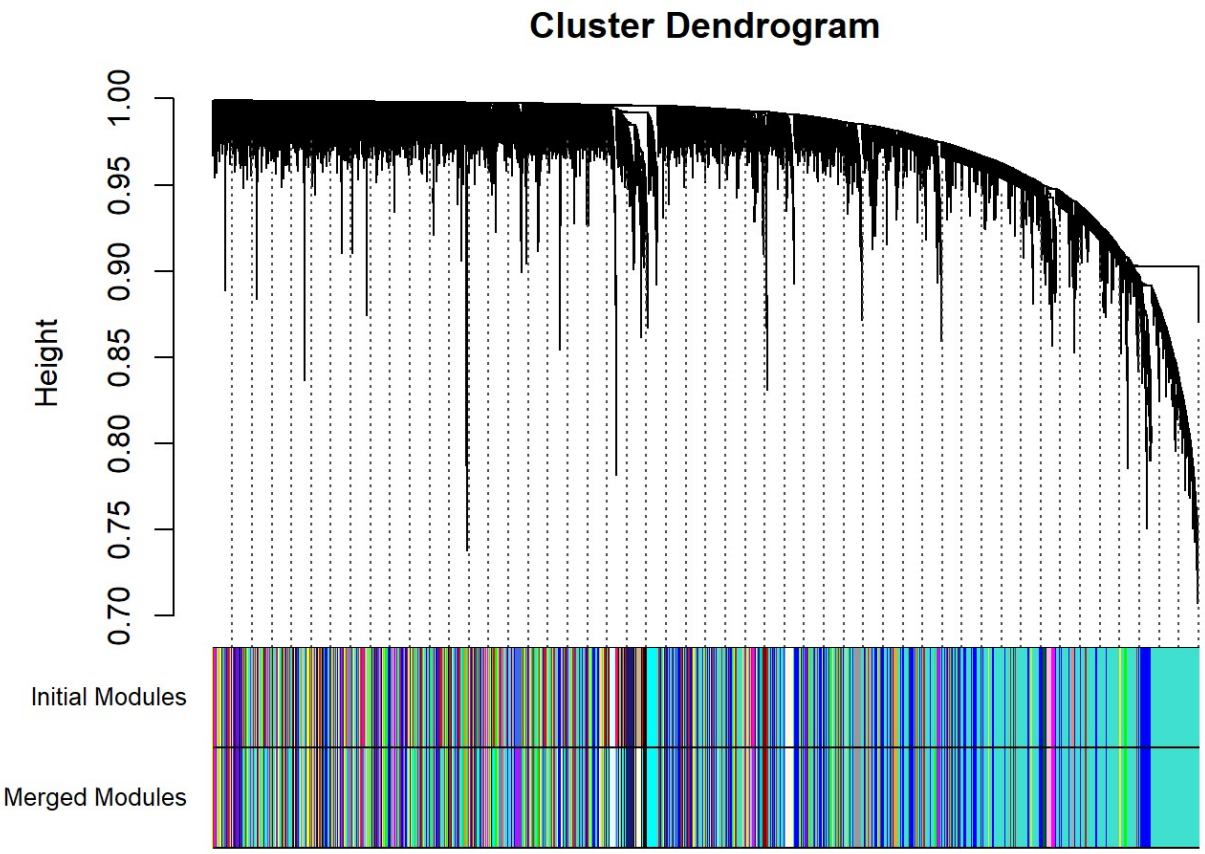
```
#Calculate initial Eigengenes
#Grey = unassigned genes to a module
#Recall that our soft power = 6
MEList <-  moduleEigengenes(Goodsamps[122:length(Goodsamps)], colors = dynamicColors,
softPower = 6, excludeGrey = TRUE)

MEs <- MEList$eigengenes

#Check similarity of Eigengenes and merge
#Cluster similar Eigengenes based on dissimilarity
MEDiss <- 1-cor(MEs)
METree <- hclust(as.dist(MEDiss), method = "average")


#Merge eigengenes with 0.85 correlation
plot(METree, main = "Clustering of module Eigengenes")
MEDissThres = 0.15
abline(h=MEDissThres, col = "red")
```




```
merge <- mergeCloseModules(Goodsamps[122:length(Goodsamps)], dynamicColors, cutHeight
= MEDissThres, verbose = 3)
```

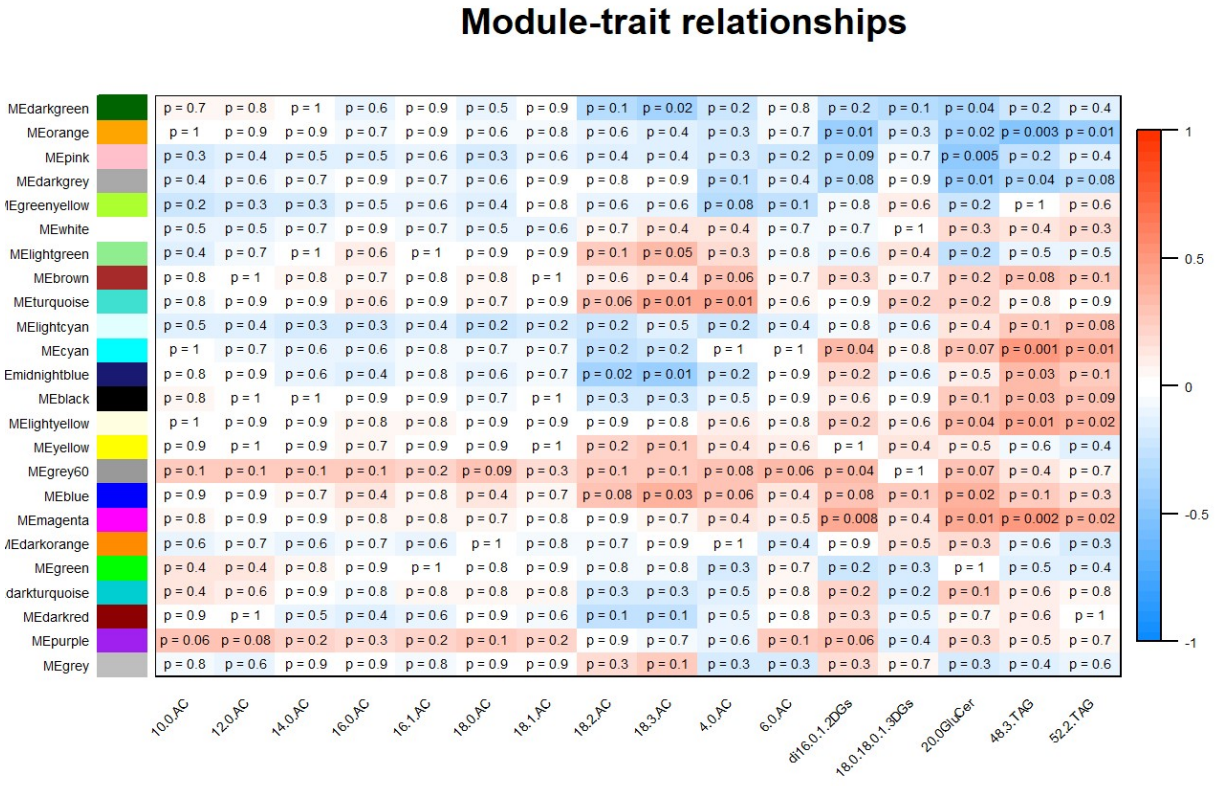
```
##  mergeCloseModules: Merging modules whose distance is less than 0.15
##    multiSetMEs: Calculating module MEs.
##      Working on set 1 ...
##      moduleEigengenes: Calculating 29 module eigengenes in given set.
##    multiSetMEs: Calculating module MEs.
##      Working on set 1 ...
##      moduleEigengenes: Calculating 24 module eigengenes in given set.
##    Calculating new MEs...
##    multiSetMEs: Calculating module MEs.
##      Working on set 1 ...
##      moduleEigengenes: Calculating 24 module eigengenes in given set.
```

```
mergedColors <- merge$colors
mergedMEs <- merge$newMEs


#Observe the change in modules across genes
plotDendroAndColors(geneTree, cbind(dynamicColors, mergedColors), c("Initial Module
s", "Merged Modules"), dendroLabels = FALSE, hang = 0.03, addGuide = TRUE, guideHang
= 0.05)
```



Our eigengenes are now ready for direct comparison with our lipids. We used a labeled heatplot to visualize.

```
#Reorder eigengenes with similar eigengenes next to each other
MEs <-  orderMEs(mergedMEs)

#Correlate modules to the lipids
#Note the use of cor vs. bicor; see disclaimer
modulelipidCor <-  cor(MEs, Goodsamps[1:121],use = "p")
modulelipidPvalue <- corPvalueStudent(modulelipidCor, 36)

#Text for the labeled heat map
textMatrix = paste("p = ", signif(modulelipidPvalue, 1), sep = "")
dim(textMatrix) = dim(modulelipidCor)

#Heatmap; note small text sizes, red= more correlation, blue = inverse correlation
labeledHeatmap(Matrix = modulelipidCor[,specieslistindex],
               xLabels = names(Goodsamps[specieslistindex]),
               yLabels = names(MEs),
               ySymbols = names(MEs),
               colorLabels = FALSE,
               colors = blueWhiteRed(50),
               textMatrix = textMatrix[,specieslistindex],
               setStdMargins = FALSE,
               cex.text = 0.45,
               zlim = c(-1,1),
               main = "Module-trait relationships",
               cex.lab = 0.45,
               yColorWidth = 0.05,
               #colors.lab.y = "white",  #change if you'd prefer module labels as text
               textAdj = 0.5
               )
```



This is our basis of picking modules of interest for further analysis. (Gene ontology, KEGG pathways, etc). In order to figure out which modules are significant outside of the relation to the external data, a few more stats can be computed.

Genes that will be of interest will likely be a combanation of highly related to external data of interest and main drivers of the module trend.

```
#Shorten eigengene names to just colors
modNames <- substring(names(MEs), 3)

#Gene module membership is determined by correlation between the eigengene trend and t
he trends for individual genes. The genes with the highest correlation drive the group
ing pattern in that particular module and may have the most direct interaction with si
gnificantly correlated external data.

geneModuleMembership <- as.data.frame(cor(Goodsamps[122:length(Goodsamps)], MEs, use
= "p"))
MPvalue <- as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership), 36))
names(geneModuleMembership) <- paste("MM", modNames, sep = "")


#Gene lipid significance indicates how strongly each gene relates to the external dat
a in question.
weight <-  as.data.frame(Goodsamps[specieslistindex]) #External data

genelipidSignificance <- as.data.frame(cor(Goodsamps[122:length(Goodsamps)], weight, u
se = "p"))
GSPvalue <- as.data.frame(corPvalueStudent(as.matrix(genelipidSignificance), 36))
names(genelipidSignificance) <- paste("GS.", names(weight), sep = "")
names(GSPvalue) <- paste("p.GS.", names(weight), sep = "")
```

This data contains the basis of genes in each module.

For analysis, we can call all the genes in any specified module(s). We can use these sublists for further analysis as needed.

```
weightmodule <- c("midnightblue") #Modules of interest, can enter multiple

genedata<- data.frame()


for (weightcol in 1:length(weightmodule)){
    module <- weightmodule[weightcol]
    column <- match(module, modNames)
    moduleGenes <- mergedColors==module

    temp <- data.frame(names(Goodsamps[122:length(Goodsamps)][moduleGenes]), #Genes
                           rep(module, length(Goodsamps[122:length(Goodsamps)][moduleG
enes])), #Module
                           geneModuleMembership[moduleGenes,column], #Module Membershi
p
                           genelipidSignificance[moduleGenes,1]) #Lipid Significance
    names(temp) <- c("Gene", "Module", "Module Membership", "Lipid significance")

    genedata <- rbind(genedata,temp)

}

summary(genedata) #All genes in all modules
```

```
##                  Gene              Module    Module Membership
##  ENSMMUG00000000102:  1   midnightblue:163   Min.   :-0.8517
##  ENSMMUG00000000265:  1                      1st Qu.: 0.3994
##  ENSMMUG00000000378:  1                      Median : 0.7040
##  ENSMMUG00000000512:  1                      Mean   : 0.4469
##  ENSMMUG00000000896:  1                      3rd Qu.: 0.7741
##  ENSMMUG00000001357:  1                      Max.   : 0.9473
##  (Other)           :157
##  Lipid significance
##  Min.   :-0.280358
##  1st Qu.:-0.084261
##  Median :-0.003704
##  Mean   : 0.004749
##  3rd Qu.: 0.082244
##  Max.   : 0.325279
##
```

```
#Remove negative membership genes (Not representative of module eigengene that the lip
ids correlate to, which we've selected our modules for )
MMpos <- genedata[genedata$`Module Membership`>0,]

summary(MMpos)
```

```
##                  Gene              Module    Module Membership
##  ENSMMUG00000000102:  1   midnightblue:133   Min.   :0.04324
##  ENSMMUG00000000265:  1                      1st Qu.:0.65060
##  ENSMMUG00000000378:  1                      Median :0.73242
##  ENSMMUG00000000512:  1                      Mean   :0.70054
##  ENSMMUG00000000896:  1                      3rd Qu.:0.79319
##  ENSMMUG00000001357:  1                      Max.   :0.94732
##  (Other)           :127
##  Lipid significance
##  Min.   :-0.22046
##  1st Qu.:-0.06273
##  Median : 0.01453
##  Mean   : 0.01689
##  3rd Qu.: 0.08822
##  Max.   : 0.32528
##
```

```
###If the interest is to find driving genes:

#Select for genes that correlate with lipids
onlypos <- MMpos[MMpos$`Lipid significance`>0,]
onlypos$product <- unlist(onlypos[3])*unlist(onlypos[4])  #product of significance an
d membership, rudimentry way to find maximum in both features.
onlypos <- onlypos[order(onlypos$product, decreasing = TRUE),]

#Select for genes that inversely correlate with lipids
onlyneg <- MMpos[MMpos$`Lipid significance`<0,]
onlyneg$product <- unlist(onlyneg[3])*unlist(onlyneg[4])
onlyneg <- onlyneg[order(onlyneg$product),]
```

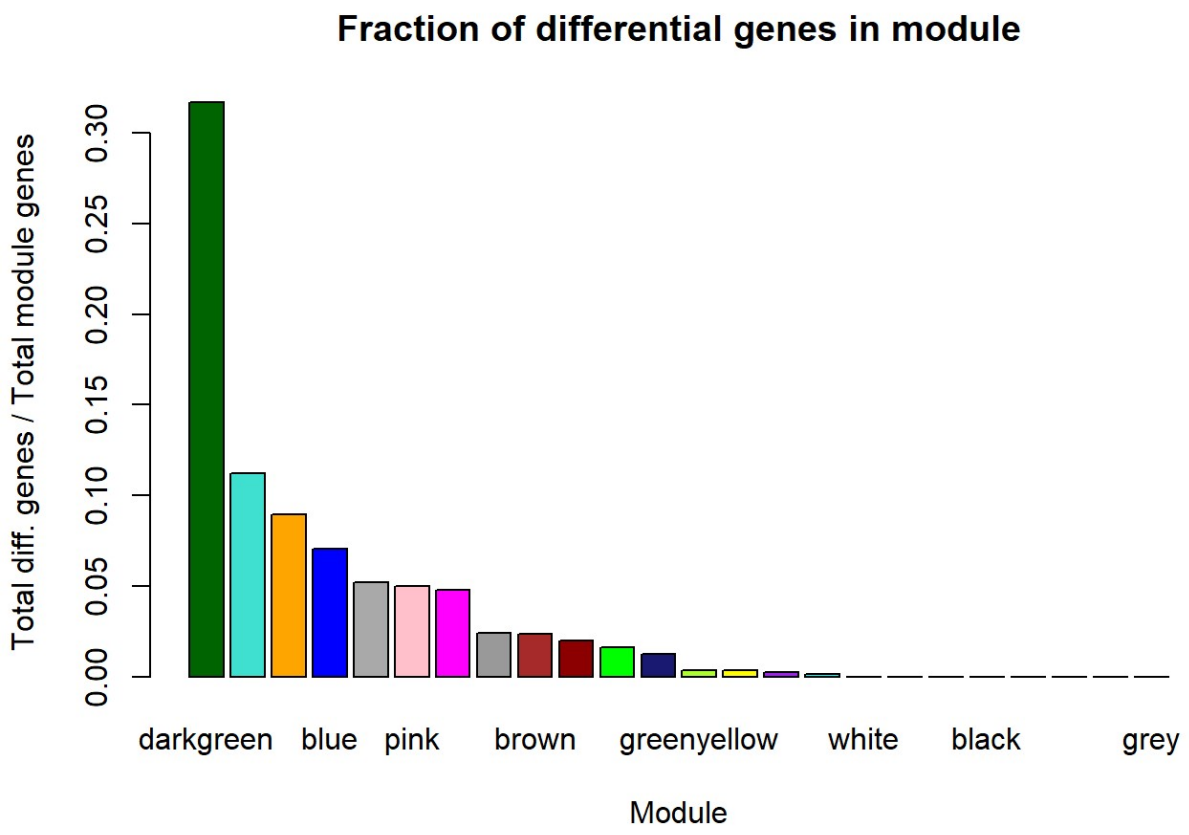
While the above code demonstrates how to pull out genes from modules, it isn't always intuitive which modules are interesting enough to look at. One way might be to look for where your differentially expressed genes are sorted.

We can determine which modules are enriched for differentially expressed genes.

```
diff <- read.csv("Maternaldiet.diff.csv", row.names = 1, header = TRUE) #Differentiall
y expressed genes, processed prior to WGCNA analysis.

#indexes of differentally expressed genes in our Goodsamps data
diffindexes <- which(colnames(Goodsamps) %in% diff$mmu_names)

#Holder for our data
difftable <- data.frame(c("Module", "Number of diff. genes", "Total genes", "Fractio
n"))

for (item in modNames){   #For each module
  module <- item
  moduleGenes <- mergedColors==module   #Pull indexes of all module genes
  diffmod <- diff$mmu_names %in% colnames(Goodsamps[122:length(Goodsamps)][moduleGene
s])  # and determine which diff. genes are in the module.
  difftable <- data.frame(difftable,
                          c(module, #Record module
                            length(diff$mmu_names[diffmod]), #Add the total number dif
fgenes to the dataframe
                            sum(moduleGenes),  #And the total genes in the module
                            length(diff$mmu_names[diffmod])/sum(moduleGenes))) #And th
e proportion
}

colnames(difftable) <- unlist(difftable[1,])
row.names(difftable) <- unlist(difftable[,1])
difftable <- difftable[2:4,2:length(difftable)]
difftable <- data.frame(t(difftable))
difftable <- difftable[order(difftable$Fraction, decreasing = TRUE),]

colnames(difftable) <- c( "Number of diff. genes", "Total genes", "Fraction")
head(difftable)
```

```
##           Number of diff. genes Total genes           Fraction
## darkgreen                    26          82  0.317073170731707
## turquoise                   722        6422  0.112426035502959
## orange                        5          56 0.0892857142857143
## blue                        255        3626 0.0703254274682846
## darkgrey                     14         270 0.0518518518518519
## pink                         17         342 0.0497076023391813
```

```
plotdiff <- type.convert(difftable[,3])
plotdiff <- plotdiff[order(plotdiff, decreasing = TRUE)]
barplot(unlist(plotdiff), names.arg = row.names(difftable), col = row.names(difftabl
e), main = "Fraction of differential genes in module", xlab = "Module", ylab = "Total
diff. genes / Total module genes")
```



So what is going on with the dark green module? We're not really sure yet! Overall, dark green does not show over representation.

We can check it out other modules too by using KEGG pathway overrepresentation analysis.

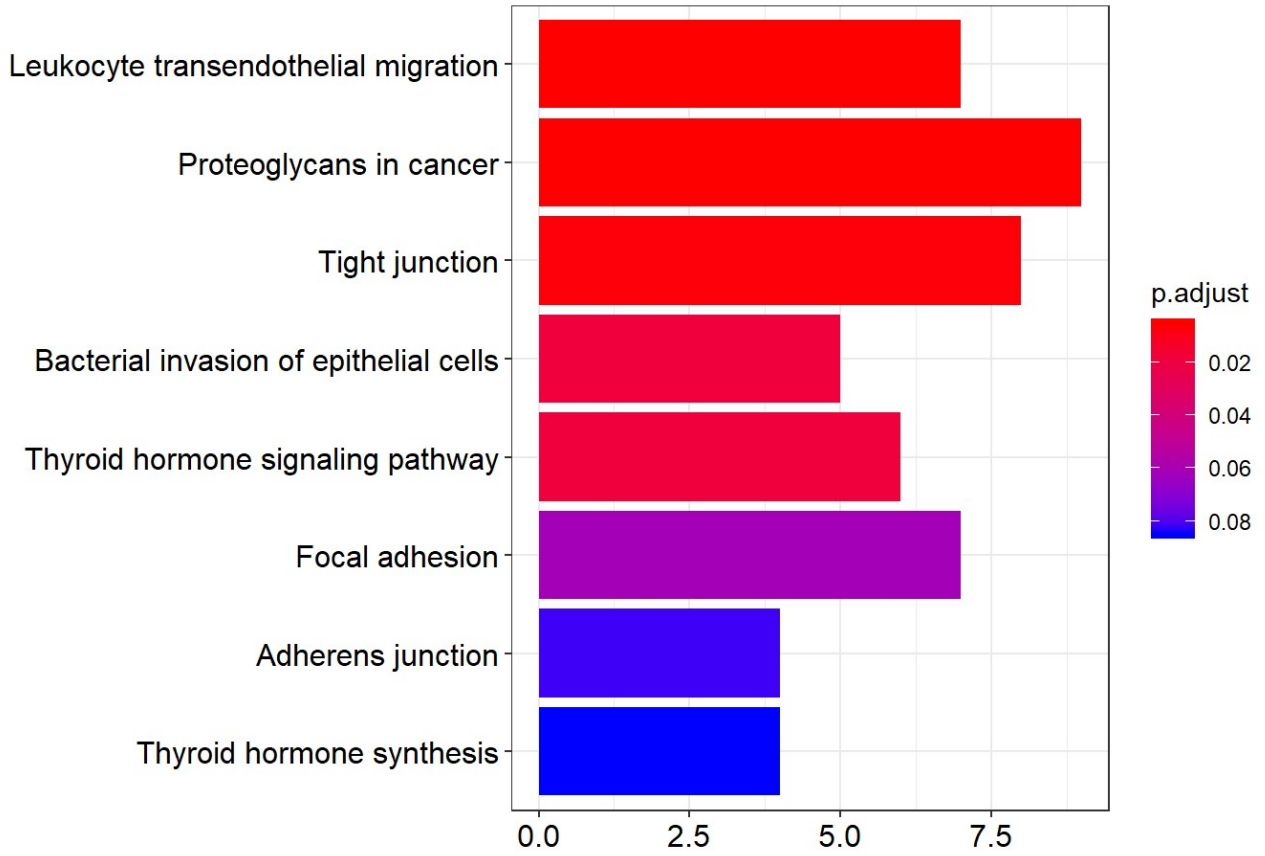
```
library(clusterProfiler) #For overrepresentation analysis
library(org.Mmu.eg.db) #Rhesus monkey database

gene.df <- bitr(genedata$Gene, fromType = "ENSEMBL",      #Convert genes in module fro
m ensembl to entrez format
          toType = "ENTREZID",
          OrgDb = org.Mmu.eg.db)  #Compare with rhesus monkey pathways, closest relati
ve in the KEGG pathways


search_kegg_organism('mcc', by='kegg_code') #Check kegg code is correct
```

```
##   kegg_code scientific_name   common_name
## 7       mcc  Macaca mulatta rhesus monkey
```

```
kk <- enrichKEGG(gene = gene.df$ENTREZID, organism = 'mcc', pvalueCutoff = 1)  #Check
for over enrichment!


head(kk)
```

```
##                ID                            Description GeneRatio
## mcc04670 mcc04670   Leukocyte transendothelial migration      7/67
## mcc05205 mcc05205                Proteoglycans in cancer      9/67
## mcc04530 mcc04530                          Tight junction      8/67
## mcc05100 mcc05100 Bacterial invasion of epithelial cells      5/67
## mcc04919 mcc04919      Thyroid hormone signaling pathway      6/67
## mcc04510 mcc04510                          Focal adhesion      7/67
##           BgRatio       pvalue    p.adjust      qvalue
## mcc04670 114/7632 5.725009e-05 0.005672289 0.004827480
## mcc05205 202/7632 6.034350e-05 0.005672289 0.004827480
## mcc04530 170/7632 1.094678e-04 0.006859984 0.005838284
## mcc05100  74/7632 4.523089e-04 0.019099775 0.016255127
## mcc04919 116/7632 5.079727e-04 0.019099775 0.016255127
## mcc04510 204/7632 1.965746e-03 0.061593385 0.052419902
##                                                                  geneID
## mcc04670            719824/100427824/574285/711712/704212/718302/713687
## mcc05205 703669/698293/574285/706980/711712/574315/698444/574320/713687
## mcc04530     719824/100427824/574285/711712/709643/574320/709836/713687
## mcc05100                             574285/706980/574315/700536/713687
## mcc04919                     710368/574285/696791/574320/713687/719119
## mcc04510               698293/574285/706980/574315/698444/574320/713687
##          Count
## mcc04670     7
## mcc05205     9
## mcc04530     8
## mcc05100     5
## mcc04919     6
## mcc04510     7
```

```
#png("AC.pathway.png",  height = 600, width = 800)
barplot(kk)
```




```
#dev.off()
```

clusterProfiler Yu G, Wang L, Han Y, He Q (2012). “clusterProfiler: an R package for comparing biological themes among gene clusters.” OMICS: A Journal of Integrative Biology, 16(5), 284-287. doi: 10.1089/omi.2011.0118.

Future directions: continue module exploration! An ANOVA across eigengenes has been suggested for notation of significantly different modules to investigate.

END of the WGCNA pipeline.

Disclaimer: There points in this pipeline that have not been fully examined for use. Further investigation is required to determine best results with the following parameters:

Correlation conducted against normalized external data?

bicor vs.cor correlation with external data (deviating from tutorial)

height cutting of dendrogram