

WGCNA – Auto Network Construction for O2

```
1 rm(list=ls())
2
3 library(jpeg)
4 library(dplyr)
5 library(tidyr)
6 library(tibble)
7 library(stringr)
8 library(ggplot2)
9
10 library(foreach)
11 library(iterators)
12 library(parallel)
13 library(doParallel)
14
15 library(WGCNA)
16 # library(KEGGREST)
17 # library(biomaRt)
18
19 set.seed(1)
20
21 # Enable WGCNA threads to speed up calculations
22 enableWGCNAThreads()
23
24
```

```
25 #####
26 # Constants/Variables
27 #####
28 selected_genotype <- "02"
29
30 softPower <- 18
31 minModuleSize <- 5
32
33 # Eigengenes clustering tree cutting threshold
34 MEDissThres <- 0.0000001
35
36
37 #####
38 # Output folder
39 #####
40 output_path <- file.path("/home/ycth8/data/projects/05_30_2021_summer_WGCNA/Maize_proteomics_output/2021_06_10_02_auto_network_construction")
41
42 if(!dir.exists(output_path)){
43   dir.create(output_path, showWarnings=FALSE, recursive=TRUE)
44   if(!dir.exists(output_path)){
45     quit(status=1)
46   }
47 }
48
```

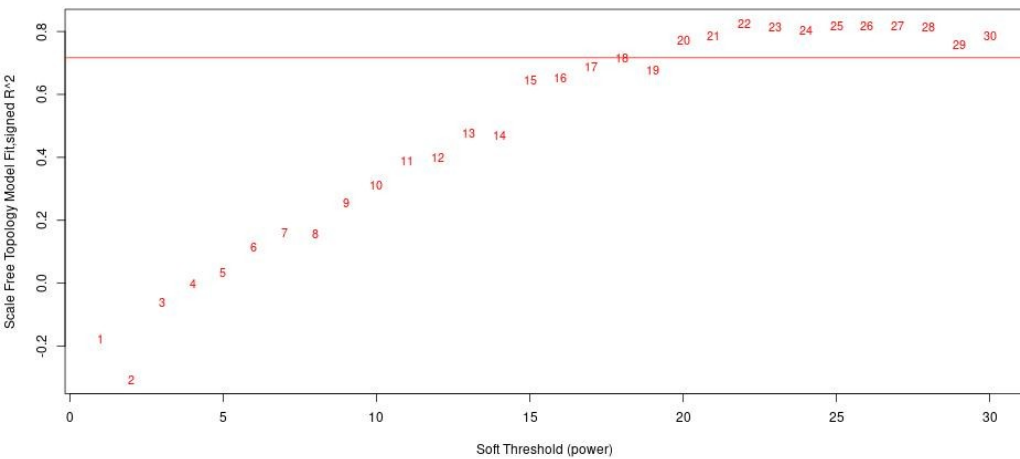
```
50 #####
51 # Read in input file
52 #####
53
54 folder_path = file.path("/home/ycth8/data/projects/05_30_2021_summer_WGCNA/Maize_proteomics_output")
55
56 datExpr = read.csv(
57   file = file.path(folder_path, "datExpr.csv"),
58   header = TRUE,
59   row.names = 1,
60   check.names = FALSE,
61   stringsAsFactors = FALSE
62 )
63
64 datExpr = datExpr[startsWith(rownames(datExpr), selected_genotype),]
65
```

```

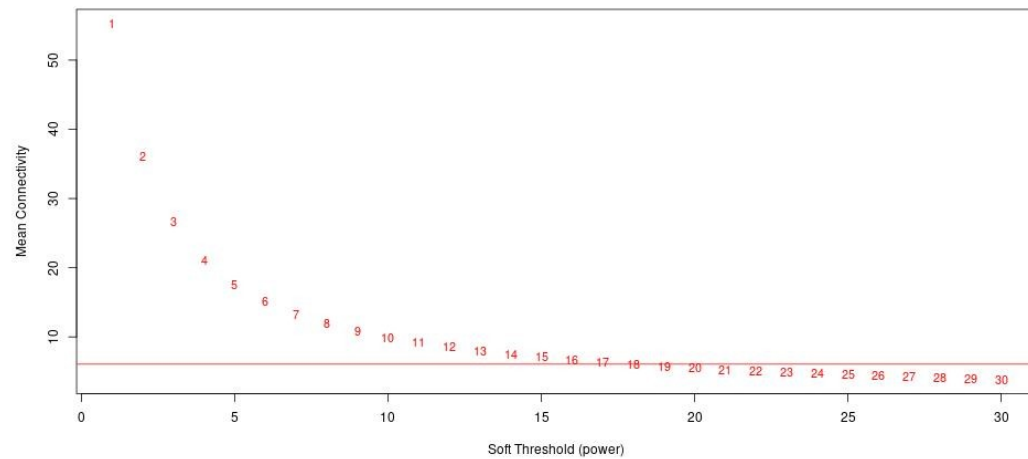
67 #####
68 # Choose a set of soft-thresholding powers
69 #####
70
71 powers = 1:30
72 # Call the network topology analysis function
73 sft = pickSoftThreshold(datExpr, powerVector = powers, verbose = 5)
74
75 # Plot tree
76 cat(rep("\n", 2))
77 jpeg(filename = file.path(output_path, "softThreshold.jpeg"), width = 1920, height = 480)
78 par(mfrow = c(1,2))
79 cex1 = 0.9
80
81 # Scale-free topology fit index as a function of the soft-thresholding power
82 plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
83      xlab="Soft Threshold (power)",ylab="Scale Free Topology Model Fit,signed R^2",type="n",
84      main = paste("Scale independence"))
85 text(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
86      labels=powers,cex=cex1,col="red")
87 # this line corresponds to using an R^2 cut-off of h
88 abline(h=sft$fitIndices[sft$fitIndices$Power == softPower, "SFT.R.sq"], col="red")
89
90 # Mean connectivity as a function of the soft-thresholding power
91 plot(sft$fitIndices[,1], sft$fitIndices[,5],
92      xlab="Soft Threshold (power)",ylab="Mean Connectivity", type="n",
93      main = paste("Mean connectivity"))
94 text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers, cex=cex1,col="red")
95 abline(h=sft$fitIndices[sft$fitIndices$Power == softPower, "mean.k."], col="red")
96 dev.off()
97
98
99 # Collect garbage
100 collectGarbage()

```

Scale independence



Mean connectivity

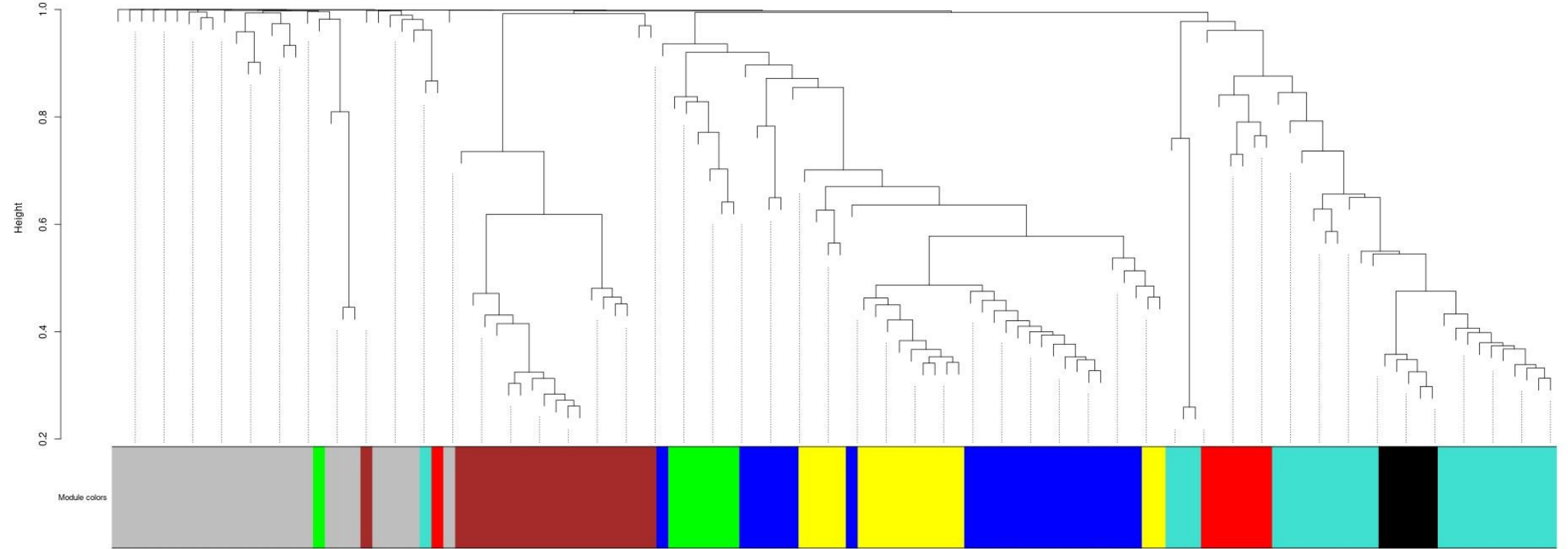


```
103 #####
104 # Create blockwise modules
105 #####
106
107 net = blockwiseModules(
108   datExpr,
109   power = softPower,
110   TOMType = "unsigned",
111   minModuleSize = minModuleSize,
112   reassignThreshold = 0,
113   mergeCutHeight = MEDissThres,
114   numericLabels = TRUE,
115   pamRespectsDendro = FALSE,
116   saveTOMs = TRUE,
117   saveTOMFileBase = file.path(output_path, "02MaizeTOM"),
118   verbose = 3
119 )
120
121 # The numbers on top are the color labels
122 print(table(net$colors))
123
```

```
125 #####
126 # Plot the dendrogram and colors underneath
127 #####
128
129 # Convert labels to colors for plotting
130 mergedColors = labels2colors(net$colors)
131
132 cat(rep("\n", 2))
133 jpeg(filename = file.path(output_path, "networkConstruction_merged_colors.jpeg"), width = 1920, height = 720)
134 plotDendroAndColors(
135   net$dendrograms[[1]],
136   mergedColors[net$blockGenes[[1]]],
137   "Module colors",
138   dendroLabels = FALSE,
139   hang = 0.03,
140   addGuide = TRUE,
141   guideHang = 0.05
142 )
143 dev.off()
...
```

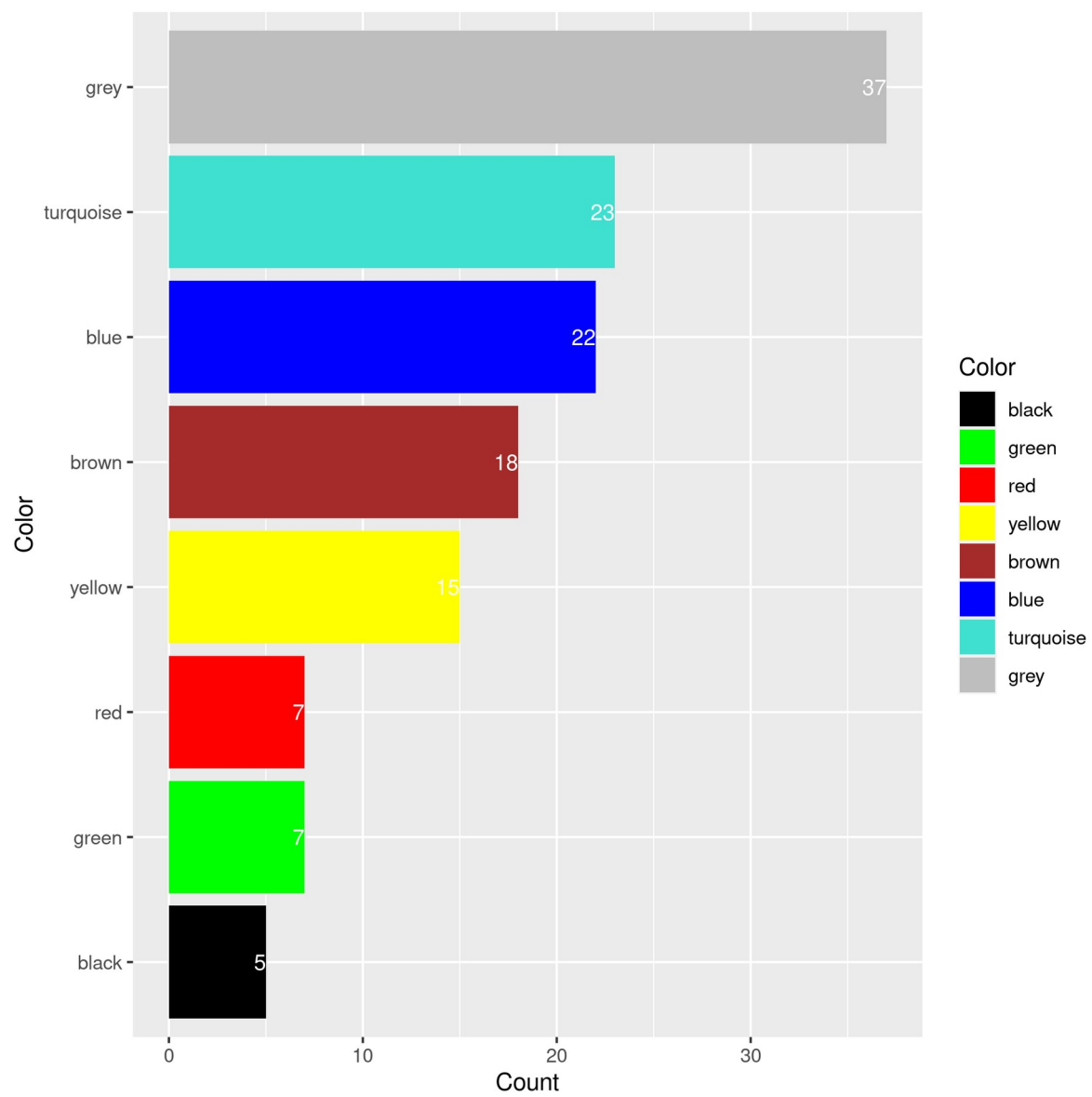


Cluster Dendrogram



```
169 #####
170 # Save genes and colors
171 #####
172 genes_colors_df <- data.frame(
173   "Gene" = colnames(datExpr),
174   "Color" = mergedColors,
175   stringsAsFactors = FALSE
176 )
177
178 write.csv(
179   x = genes_colors_df,
180   file = file.path(output_path, "genes_colors_df.csv"),
181   na = "",
182   quote = FALSE,
183   row.names = FALSE
184 )
```

```
187 genes_colors_summary_df <- genes_colors_df %>%
188   group_by(Color) %>%
189   summarize(Count = n()) %>%
190   arrange(Count) %>%
191   as.data.frame(stringsAsFactors = FALSE)
192
193 genes_colors_summary_df$Color <- factor(genes_colors_summary_df$Color, levels = unique(genes_colors_summary_df$Color))
194
195 p <- ggplot(data=genes_colors_summary_df, aes(x = Color, y=Count)) +
196   geom_bar(mapping = aes(fill = Color), stat="identity") +
197   geom_text(aes(label=Count), hjust=1, color="white", size=3.5) +
198   coord_flip() +
199   scale_fill_manual(values = levels(genes_colors_summary_df$Color))
200
201 ggsave(
202   filename = "genes_colors_summary.png",
203   plot = p,
204   path = output_path
205 )
```



```
146 #####
147 # Save important variables as RData
148 #####
149
150 moduleLabels = net$colors
151 moduleColors = labels2colors(net$colors)
152 MEs = net$MEs;
153 geneTree = net$dendrograms[[1]];
154 save(
155   MEs, moduleLabels, moduleColors, geneTree,
156   file = file.path(output_path, "O2-networkConstruction-auto.RData")
157 )
```

---

