WGCNA Section 3: Relating modules to external information and identifying important genes

Angela Yoder

2021-05-26

The goals of this program are:

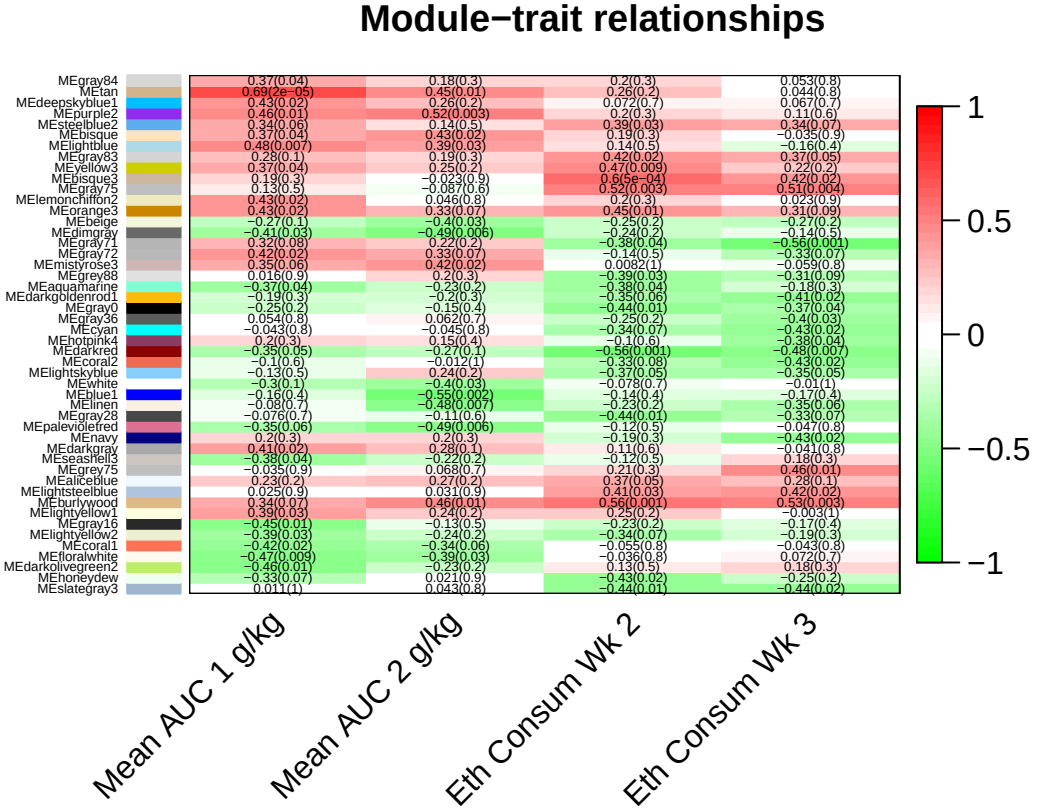
1. Show association between module eigengenes and clinical traits

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

sigpval = subset(moduleTraitPvalue, moduleTraitPvalue[,1] < 0.05 | moduleTraitPvalue[,2] < 0.05 | modul
names.sig = rownames(sigpval)

onlysigpval = subset(moduleTraitCor, rownames(moduleTraitCor) %in% names.sig)
deletegray = dim(onlysigpval)[1]-1
onlysigpval = onlysigpval[1:deletegray,]
sigpval = sigpval[1:deletegray,]


sigpval_auc1 = moduleTraitPvalue[moduleTraitPvalue[,1] < 0.05, 1]
sigpval_auc2 = moduleTraitPvalue[moduleTraitPvalue[,2] < 0.05, 2]
sigpval_et2 = moduleTraitPvalue[moduleTraitPvalue[,3] < 0.05, 3]
sigpval_et3 = moduleTraitPvalue[moduleTraitPvalue[,4] < 0.05, 4]

#onlysigpval_auc1 = subset(moduleTraitCor, rownames(moduleTraitCor) %in% names(sigpval_auc1))[,1]
#onlysigpval_auc2 = subset(moduleTraitCor, rownames(moduleTraitCor) %in% names(sigpval_auc2))[,2]
#onlysigpval_et2 = subset(moduleTraitCor, rownames(moduleTraitCor) %in% names(sigpval_et2))[,3]
#onlysigpval_et3 = subset(moduleTraitCor, rownames(moduleTraitCor) %in% names(sigpval_et3))[,4]

auc1_names = names(sigpval_auc1)
auc2_names = names(sigpval_auc2)
et2_names = names(sigpval_et2)
et3_names = names(sigpval_et3)

setwd("C:/Users/angel/OneDrive/Desktop/Saba Lab/Data/")

write.csv(auc1_names, "AUC1 Module Names.csv")
```

```
write.csv(auc2_names, "AUC2 Module Names.csv")
write.csv(et2_names, "EtOH Week 2.csv")
write.csv(et3_names, "EtOH Week 3.csv")
```

```
#sigpval2 = signif(sigpval, 1)

#sigpval = as.numeric(sigpval)
#sizeGrWindow(10,6)
# Will display correlations and their p-values
textMatrix = paste(signif(onlysigpval, 2), "(",
  signif(sigpval, 1), ")", sep = "");
textMatrix = matrix(data = textMatrix, nrow = dim(onlysigpval)[1], ncol = dim(onlysigpval)[2])
#dim(textMatrix) = dim(onlysigpval)
par(mar = c(6, 8.5, 3, 3));
# Display the correlation values within a heatmap plot
labeledHeatmap(Matrix = onlysigpval,
  cex.lab.y = 0.4,
  xLabels = c("Mean AUC 1 g/kg", "Mean AUC 2 g/kg", "Eth Consum Wk 2", "Eth Consum Wk 3"),
  yLabels = rownames(onlysigpval),
  ySymbols = rownames(onlysigpval),
  colorLabels = FALSE,
  colors = greenWhiteRed(50),
  textMatrix = textMatrix,
  setStdMargins = FALSE,
  cex.text = 0.4,
  zlim = c(-1,1),
  main = paste("Module-trait relationships"))
```

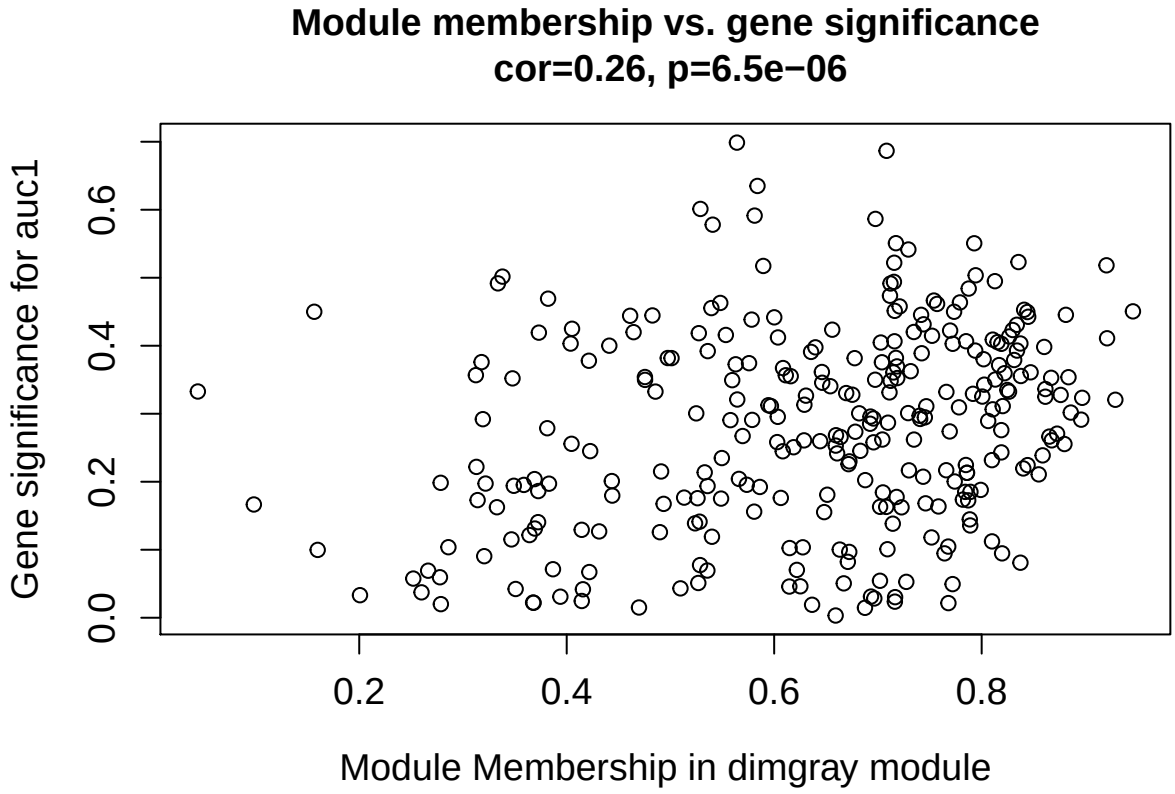
```
## Warning in greenWhiteRed(50): WGCNA::greenWhiteRed: this palette is not suitable for people
## with green-red color blindness (the most common kind of color blindness).
## Consider using the function blueWhiteRed instead.
```




```
setwd("C:/Users/angel/OneDrive/Desktop/Saba Lab/Data/")
write.csv(onlysigpval, "Chosen Modules.csv")
```

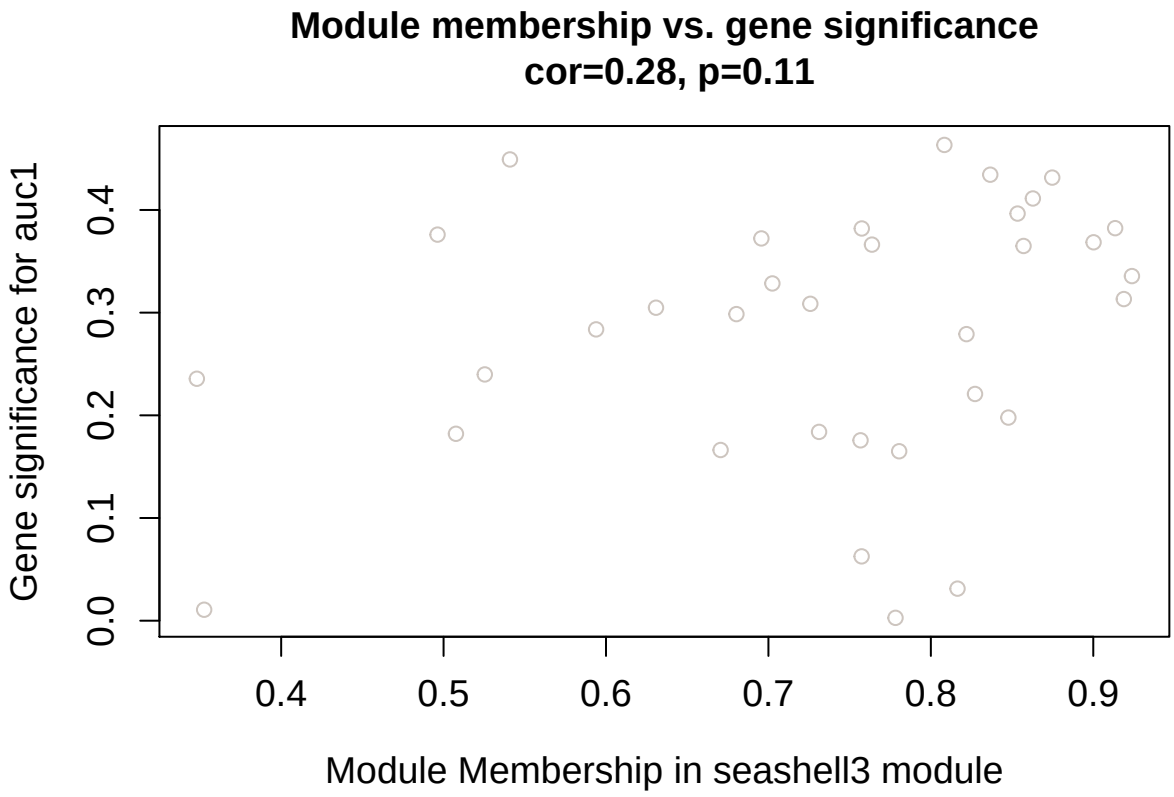
```
## AUC1

# Define variable weight containing the weight column of datTrait
auc1 = as.data.frame(phen$meanauc1gkg);
names(auc1) = "auc1"
# names (colors) of the modules
modNames = substring(names(MEs), 3)
geneModuleMembership = as.data.frame(cor(protein_brain, MEs, use = "p"));
MMPvalue = as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership), nSamples));
names(geneModuleMembership) = paste("MM", modNames, sep="");
names(MMPvalue) = paste("p.MM", modNames, sep="");
geneTraitSignificance = as.data.frame(cor(protein_brain, auc1, use = "p"));
GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance), nSamples));
names(geneTraitSignificance) = paste("GS.", names(auc1), sep="");
names(GSPvalue) = paste("p.GS.", names(auc1), sep="")

#best = numeric(2)
#colorAUC = rownames(moduleTraitCor)
#minAUC1 = grep(min(moduleTraitCor[,1]), moduleTraitCor[,1])
#best[1] = colorAUC[minAUC1]
#maxAUC1 = grep(max(moduleTraitCor[,1]), moduleTraitCor[,1])
#best[2] = colorAUC[maxAUC1]
```

```
gee = ceiling(runif(2, min = 0, max = length(auc1_names)))
best = auc1_names[gee]
best = sapply(best, FUN = gsub, pattern = "ME", replacement = "")


module = best[1]
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 auc1",
main = paste("Module membership vs. gene significance\n"),
cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = "black")
```



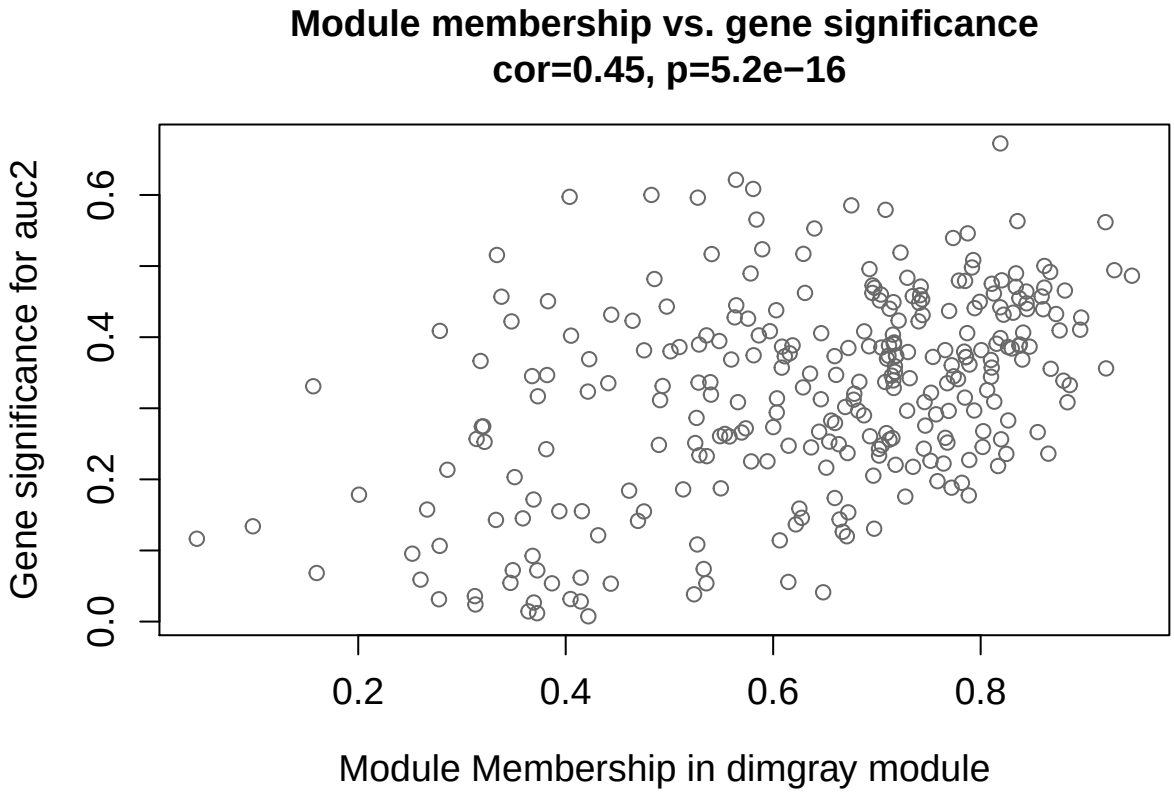

```
# AUC1 .2

module = best[2]
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 auc1",
main = paste("Module membership vs. gene significance\n"),
cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
```

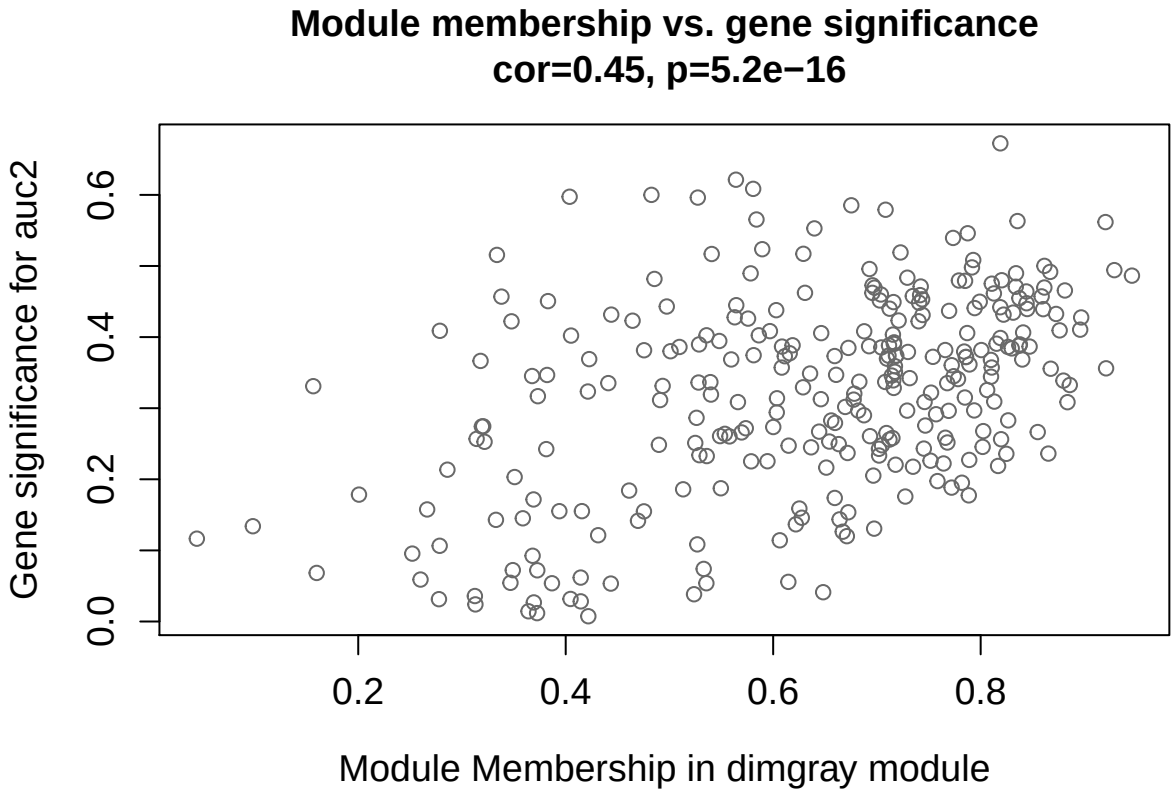



```
## AUC2

# Define variable weight containing the weight column of datTrait
auc2 = as.data.frame(phen$meanauc2gkg);
names(auc2) = "auc2"
# names (colors) of the modules
modNames = substring(names(MEs), 3)
geneModuleMembership = as.data.frame(cor(protein_brain, MEs, use = "p"));
MMPvalue = as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership), nSamples));
names(geneModuleMembership) = paste("MM", modNames, sep="");
names(MMPvalue) = paste("p.MM", modNames, sep="");
geneTraitSignificance = as.data.frame(cor(protein_brain, auc2, use = "p"));
GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance), nSamples));
names(geneTraitSignificance) = paste("GS.", names(auc1), sep="");
names(GSPvalue) = paste("p.GS.", names(auc2), sep="")

rand_module = ceiling(runif(2, min = 0, max = length(auc2_names)))
best = auc2_names[rand_module]
best = sapply(best, FUN = gsub, pattern = "ME", replacement = "")
```

```
module = best[1]
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 auc2",
main = paste("Module membership vs. gene significance\n"),
cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
```

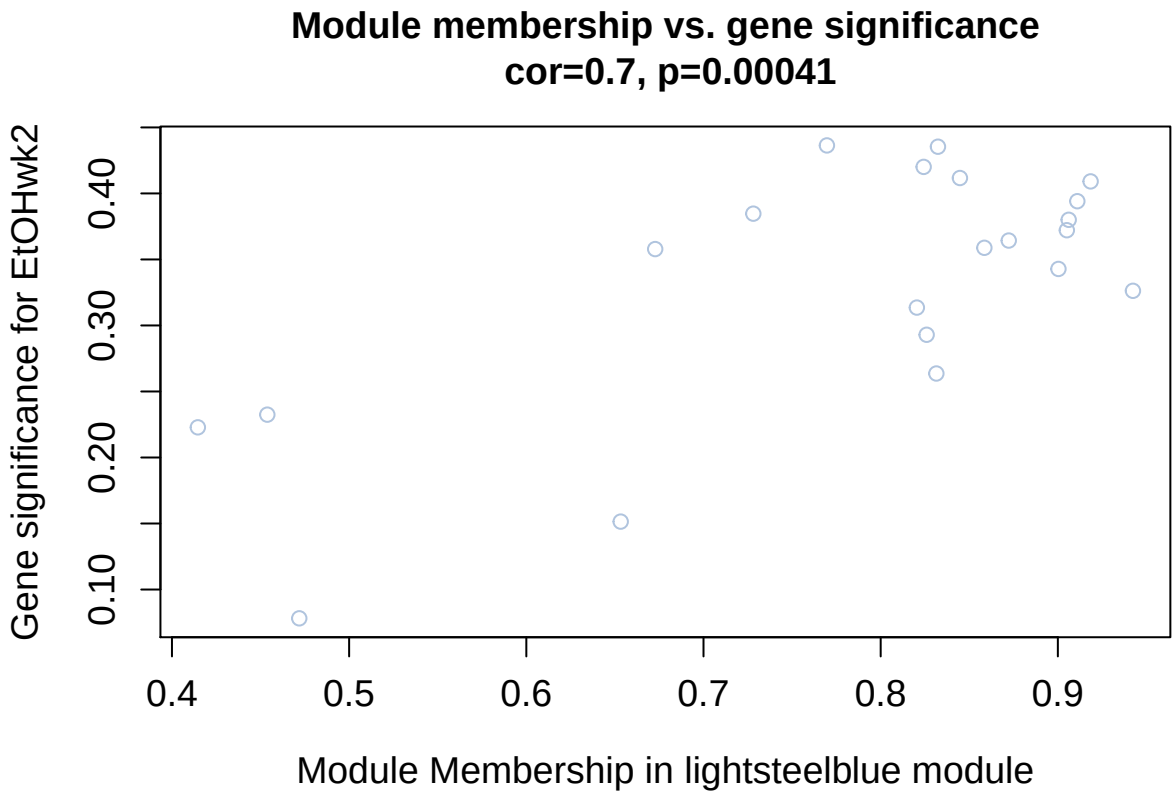



```
# AUC2 .2

module = best[2]
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 auc2",
main = paste("Module membership vs. gene significance\n"),
cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
```

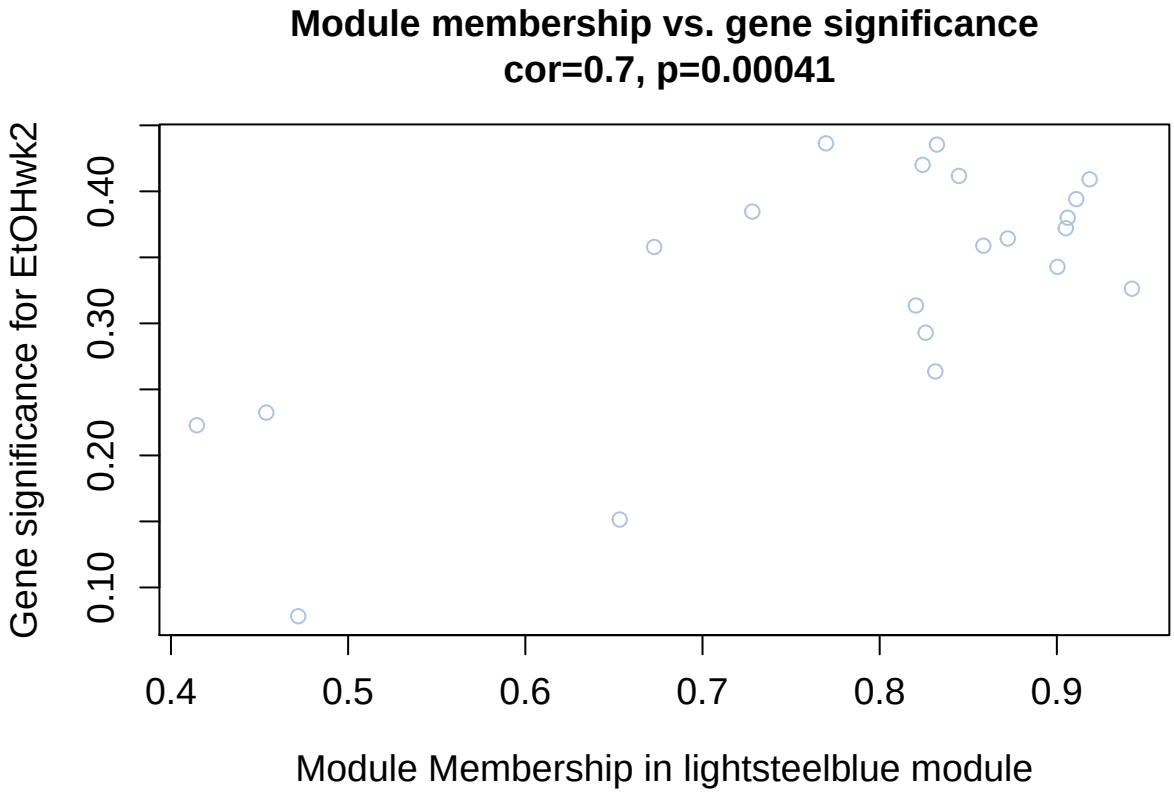



```
## EtOHwk2

# Define variable weight containing the weight column of datTrait
EtOHwk2 = as.data.frame(phen$EtOHing_wk2);
names(EtOHwk2) = "EtOHwk2"
# names (colors) of the modules
modNames = substring(names(MEs), 3)
geneModuleMembership = as.data.frame(cor(protein_brain, MEs, use = "p"));
MMPvalue = as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership), nSamples));
names(geneModuleMembership) = paste("MM", modNames, sep="");
names(MMPvalue) = paste("p.MM", modNames, sep="");
geneTraitSignificance = as.data.frame(cor(protein_brain, EtOHwk2, use = "p"));
GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance), nSamples));
names(geneTraitSignificance) = paste("GS.", names(EtOHwk2), sep="");
names(GSPvalue) = paste("p.GS.", names(EtOHwk2), sep="")

rand_module = ceiling(runif(2, min = 0, max = length(et2_names)))
best = et2_names[rand_module]
best = sapply(best, FUN = gsub, pattern = "ME", replacement = "")

module = best[1]
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 EtOHwk2",
main = paste("Module membership vs. gene significance\n"),
cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
```

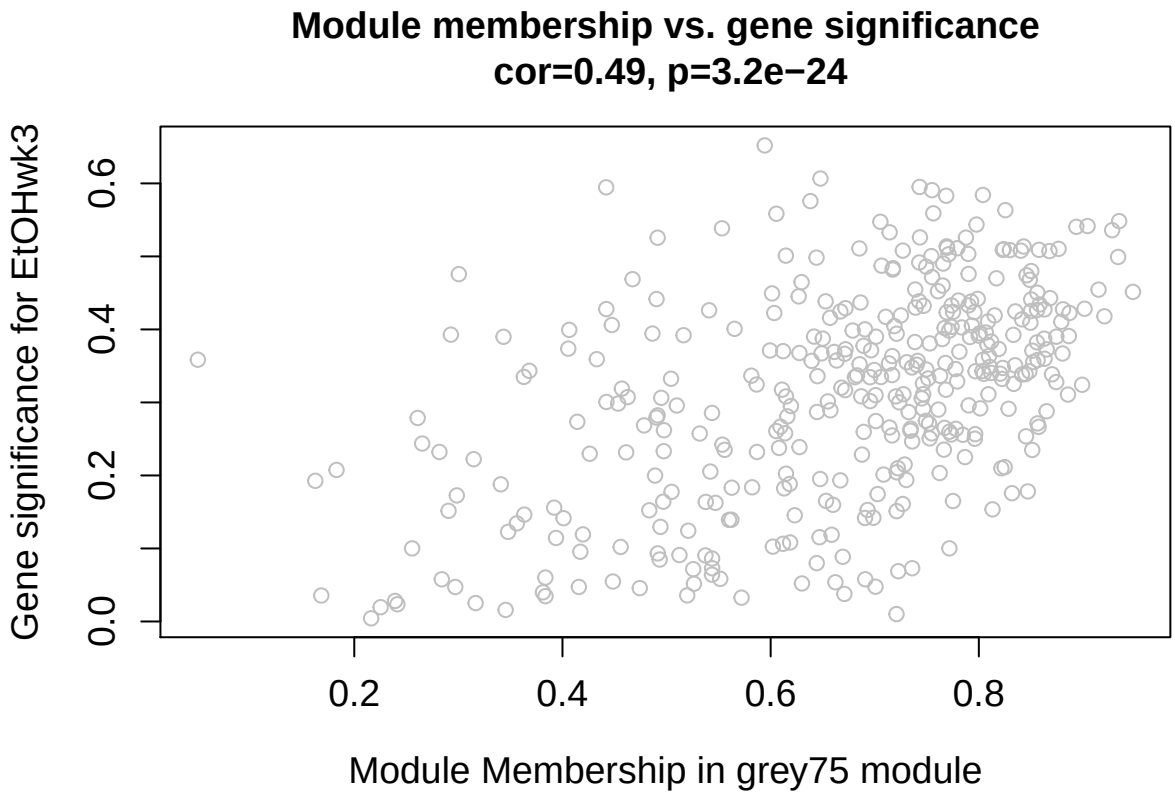



```
# EtOHwk2 .2

module = best[2]
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 EtOHwk2",
main = paste("Module membership vs. gene significance\n"),
cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
```

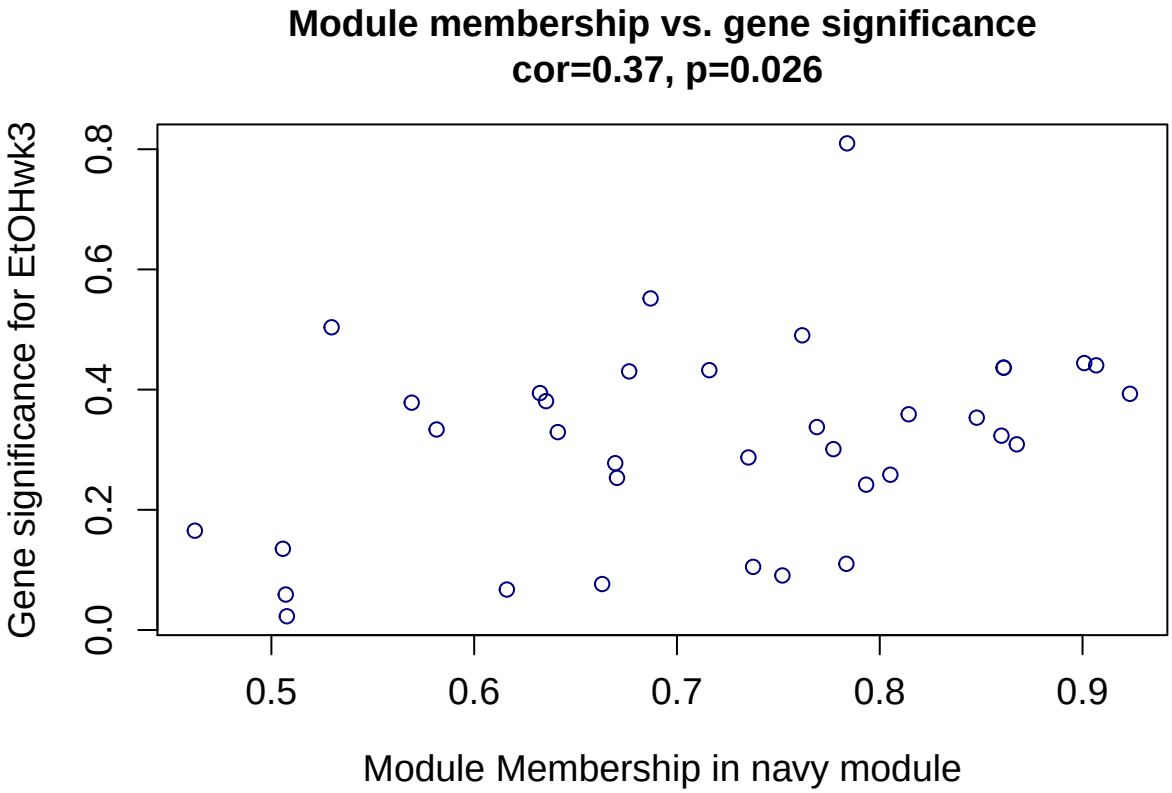



```
## EtOHwk3

# Define variable weight containing the weight column of datTrait
EtOHwk3 = as.data.frame(phen$EtOHing_wk3);
names(EtOHwk3) = "EtOHwk3"
# names (colors) of the modules
modNames = substring(names(MEs), 3)
geneModuleMembership = as.data.frame(cor(protein_brain, MEs, use = "p"));
MMPvalue = as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership), nSamples));
names(geneModuleMembership) = paste("MM", modNames, sep="");
names(MMPvalue) = paste("p.MM", modNames, sep="");
geneTraitSignificance = as.data.frame(cor(protein_brain, EtOHwk3, use = "p"));
GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance), nSamples));
names(geneTraitSignificance) = paste("GS.", names(EtOHwk3), sep="");
names(GSPvalue) = paste("p.GS.", names(EtOHwk3), sep="")

rand_module = ceiling(runif(2, min = 0, max = length(et3_names)))
best = et3_names[rand_module]
best = sapply(best, FUN = gsub, pattern = "ME", replacement = "")

module = best[1]
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 EtOHwk3",
main = paste("Module membership vs. gene significance\n"),
cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
```




```
# EtOHwk2 .2

module = best[2]
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 EtOHwk3",
main = paste("Module membership vs. gene significance\n"),
cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)
```


Module membership vs. gene significance
cor=0.37, p=0.026
Gene significance for EtOHwk3
0.0
0.2
0.4
0.6
0.8
0.5
0.6
0.7
0.8
0.9
Module Membership in navy module