**#1.试图将此版本作为最终版本，所以此版本比较全面以及精炼，并更多的增加使用时自动化过程；**

**#2. 标红为可修改项，标黄为重点的地方**

目录

[1.数据导入 1](#_Toc514787472)

[1.1导入表达矩阵 1](#_Toc514787473)

[1.2导入临床trait信息（可省略） 2](#_Toc514787474)

[2.生成module 3](#_Toc514787475)

[3.将modules和trait相关联 5](#_Toc514787476)

[4. VisANT 和Cytoscape 8](#_Toc514787477)

[4.1 VisANT 8](#_Toc514787478)

[4.2 Cytoscape 8](#_Toc514787479)

# 1.数据导入

## 1.1导入表达矩阵

library(WGCNA);

fpkm = read.csv("gene-fpkm.csv",header =T, row.name=1 );

###简单查看数据情况,可有可无

dim(fpkm);

names(fpkm);

### 表达矩阵行列转置

datExpr=as.data.frame(t(fpkm));

**###对样本进行聚类，如果有异常的样本，则异常的样本需要去掉**

sampleTree = hclust(dist(datExpr), method = "average");

sizeGrWindow(12,9)

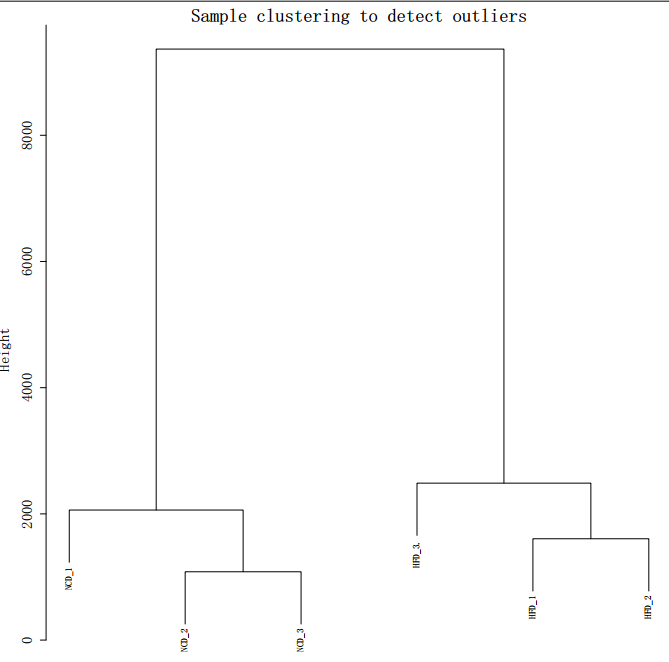
par(cex = 0.6);

par(mar = c(0,4,2,0))

plot(sampleTree, main = "Sample clustering to detect outliers", sub="", xlab="", cex.lab = 1.5,

cex.axis = 1.5, cex.main = 2)

图片：



###不好的样本可以手动在原文件里去掉，如果样本数太多，如大于100个，不容易看出哪个样本除了问题，则需要用代码去掉:

**# 自动去掉异常样本：画分割线，这个先需要根据图中异常值的大小话，尽量大一点**

cut<- 6000

abline(h = cut, col = "red");

# Determine cluster under the line

clust = cutreeStatic(sampleTree, cutHeight = cut , minSize = 10)

table(clust)

# clust 1 contains the samples we want to keep.

keepSamples = (clust==1)

datExpr = datExpr[keepSamples, ]

#生成新的聚类树并将聚类树保存成pdf文件

sampleTree = hclust(dist(datExpr), method = "average");

sizeGrWindow(12,9)

pdf("sampleClustering.pdf",family="GB1")

par(cex = 0.6);

par(mar = c(0,4,2,0))

plot(sampleTree, main = "Sample clustering to detect outliers", sub="", xlab="", cex.lab = 1.5,

cex.axis = 1.5, cex.main = 2)

dev.off()

## 1.2导入临床trait信息（可省略）

##对信息重新组建构架，将datExpr和trait相关联，如果没有trait信息也可以忽略此步骤

##导入trait信息，并根据样本名称和datExpr数据进行匹配：

allTraits = read.csv("trait.txt");

femaleSamples = rownames(datExpr);

traitRows = match(femaleSamples, allTraits$Sample\_geo\_accession); #修改样本名称

datTraits = allTraits[traitRows, 2];

rownames(datTraits) = allTraits[traitRows, 1];

collectGarbage();

#如果trait本身样本和表达矩阵对应好，则可以直接简便的运行以下两行：

allTraits = read.csv("h\_or\_l\_trait.csv",row.names = 1,header=TRUE);

datTraits<- allTraits

###通过样本系统树查看临床信息和样本的关联

sampleTree2 = hclust(dist(datExpr), method = "average")

# Convert traits to a color representation: white means low, red means high, grey means missing entry

traitColors = numbers2colors(datTraits, signed = FALSE);

# Plot the sample dendrogram and the colors underneath.

plotDendroAndColors(sampleTree2, traitColors,

groupLabels = names(datTraits),

main = "Sample dendrogram and trait heatmap")

####因为样本数比较多，所以R里显示不出聚类图，但是PDF是可用的；上面的##numbers2colors因为trait里有字符，所以不能成功转变，所以上图无法画出

# 2.生成module

**##对基因做module划分，这里是一步法相对简单很多（还有另外两种方法，比较麻烦，暂时不写）**

###先做一个软件阈值检测，求一个最好的power值，这个必须得做

powers = c(c(1:10), seq(from = 12, to=20, by=2))

sft = pickSoftThreshold(datExpr, powerVector = powers, verbose = 5)

sizeGrWindow(9, 5)

par(mfrow = c(1,2));

cex1 = 0.9;

plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])\*sft$fitIndices[,2],

xlab="Soft Threshold (power)",ylab="Scale Free Topology Model Fit,signed R^2",type="n",

main = paste("Scale independence"));

text(sft$fitIndices[,1], -sign(sft$fitIndices[,3])\*sft$fitIndices[,2],

labels=powers,cex=cex1,col="red");

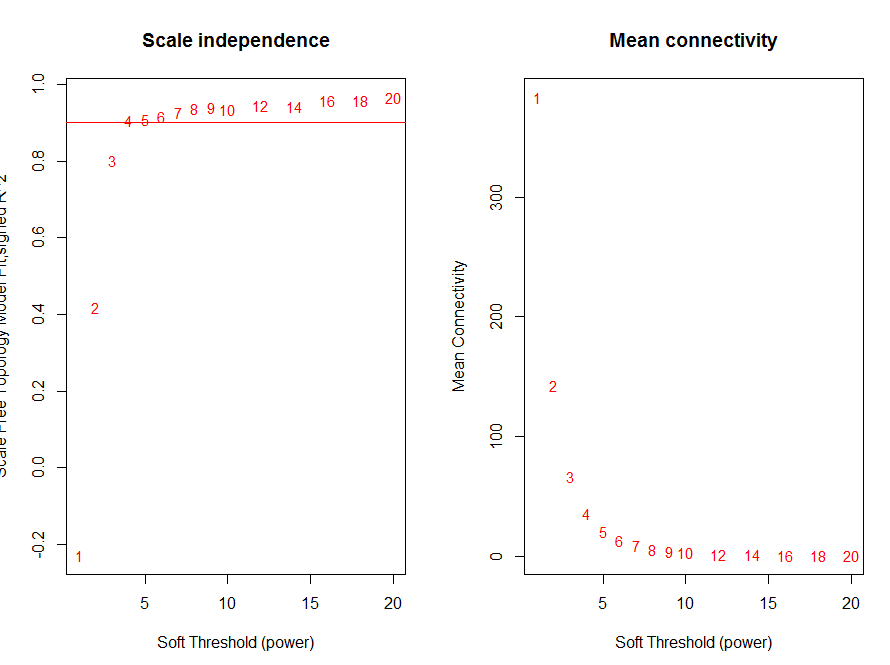
abline(h=0.90,col="red")

plot(sft$fitIndices[,1], sft$fitIndices[,5],

xlab="Soft Threshold (power)",ylab="Mean Connectivity", type="n",

main = paste("Mean connectivity"))

text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers, cex=cex1,col="red")



**###此处根据图上的数据显示，4和红色横线相交，所以最好的power值为4，默认最好的值存放在sft$powerEstimate中，所以可以直接调用；**

#用得到的power值求module

net = blockwiseModules(datExpr, power = sft$powerEstimate,

TOMType = "unsigned", minModuleSize = 30,

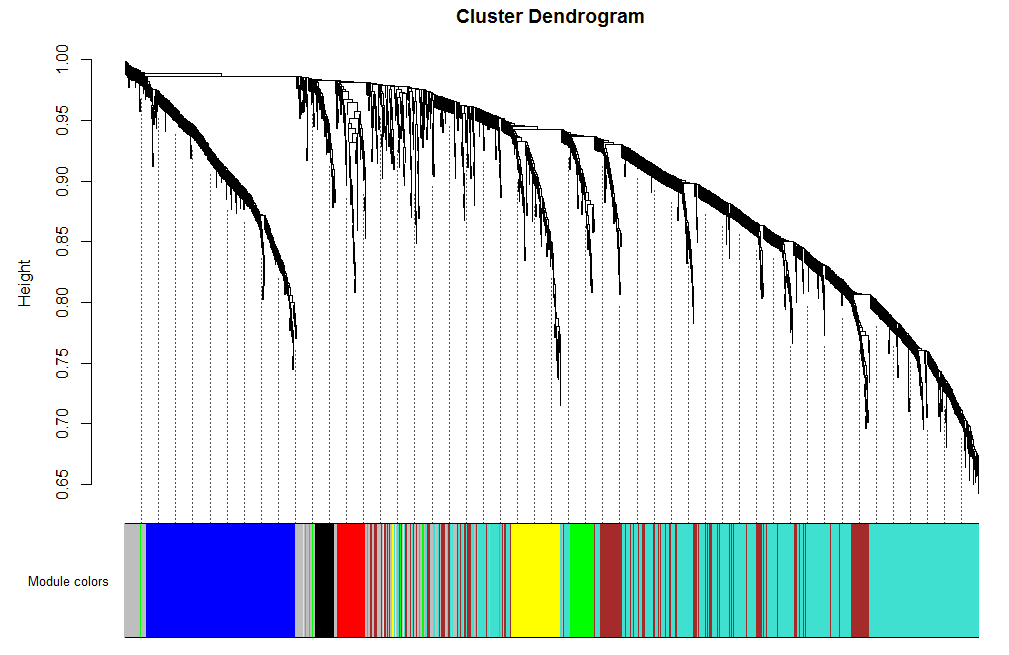
reassignThreshold = 0, mergeCutHeight = 0.25,

numericLabels = TRUE, pamRespectsDendro = FALSE,maxBlockSize = 18000,

saveTOMs = TRUE,

saveTOMFileBase = "femaleMouseTOM",

verbose = 3)



**###这里有两个重要的输出：**

moduleLabels = net$colors

moduleColors = labels2colors(net$colors)

MEs = net$MEs;

geneTree = net$dendrograms[[1]];

###输出color和gene对应表格：

probes = names(datExpr);

info = data.frame(Probe = probes, ModuleLabel = moduleLabels,ModuleColor = labels2colors(moduleLabels));

write.csv(info, file = "color-gene-POCD.csv",row.names = FALSE, quote = FALSE);

###输出color和gene的相关系数：

geneModuleMembership = as.data.frame(cor(datExpr, MEs, use = "p"));

write.csv(geneModuleMembership, file = "color-gene-POCD-相关系数.csv",row.names = TRUE, quote = FALSE);

###画图表示各个module，文章中可能会放：

# open a graphics window

sizeGrWindow(12, 9)

# Convert labels to colors for plotting

mergedColors = labels2colors(net$colors)

# Plot the dendrogram and the module colors underneath

plotDendroAndColors(net$dendrograms[[1]], mergedColors[net$blockGenes[[1]]],

"Module colors",

dendroLabels = FALSE, hang = 0.03,

addGuide = TRUE, guideHang = 0.05)

# 3.将modules和trait相关联

**#此章节主要是求module和trait的相关性，以及求gene和trait的相关性**

#3.1将modules和trait进行定量计算(trait里不能有字符信息,必须是数字，不同的值可以用1,2,3等数字来表示)

# 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);

**###将定量的结果画图**

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),

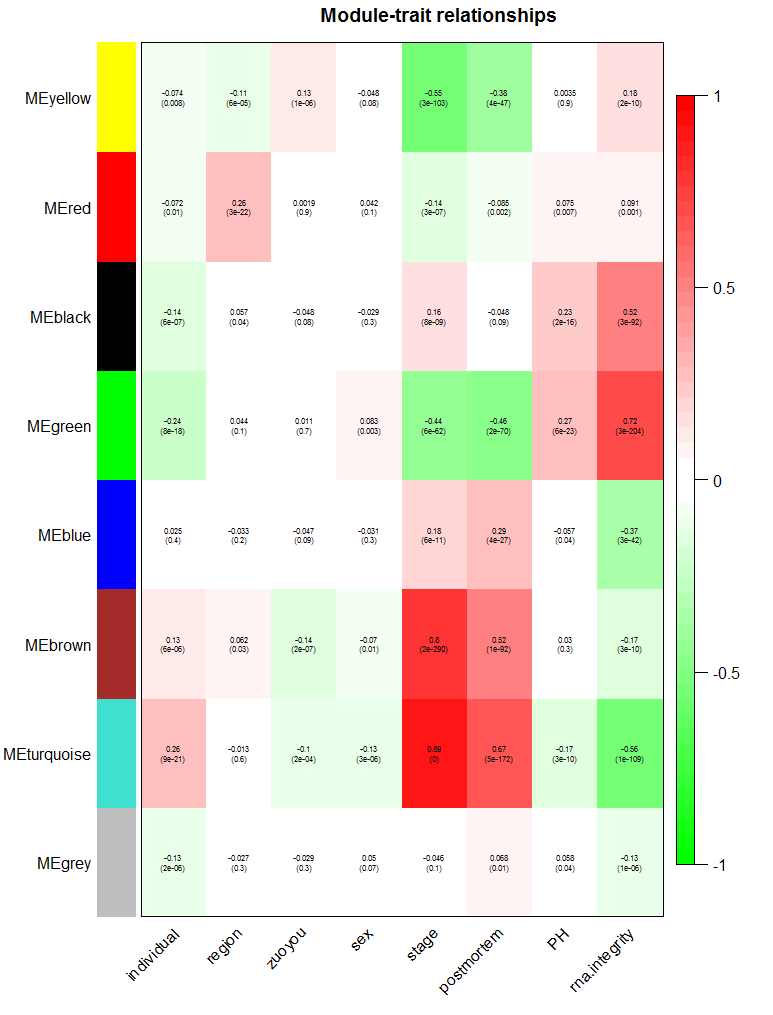
textMatrix = textMatrix,

setStdMargins = FALSE,

cex.text = 0.5,

zlim = c(-1,1),

main = paste("Module-trait relationships"))



#手动保存成图片

###输出基因和trait的关系矩阵

geneTraitSignificanceall = as.data.frame(cor(datExpr, datTraits, use = "p"))

write.csv(geneTraitSignificanceall, file = " trait-gene-相关系数.csv",row.names = TRUE, quote = FALSE);

###3.b将gene和trait关联分析：这里选取stage(感觉用不到)

# Define variable weight containing the weight column of datTrait

stage = as.data.frame(datTraits$stage);

names(stage) = "stage"

# names (colors) of the modules

modNames = substring(names(MEs), 3)

##gene和module相关性

geneModuleMembership = as.data.frame(cor(datExpr, 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="");

#gene和weight(具体的重量形状)相关性

geneTraitSignificance = as.data.frame(cor(datExpr, stage, use = "p"));

GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance), nSamples));

names(geneTraitSignificance) = paste("GS.", names(stage), sep="");

names(GSPvalue) = paste("p.GS.", names(stage), sep="");

###3.c针对weight（trait）和brown（module）两个具体的作散点图

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)

###3.c针对stage（trait）和brown（module）两个具体的作散点图

module = "turquoise"

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",

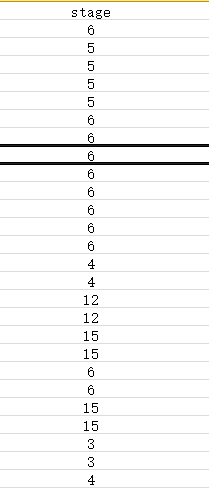
main = paste("Module membership vs. gene significance\n"),

cex.main = 1.2, cex.lab = 1.2, cex.axis = 1.2, col = module)

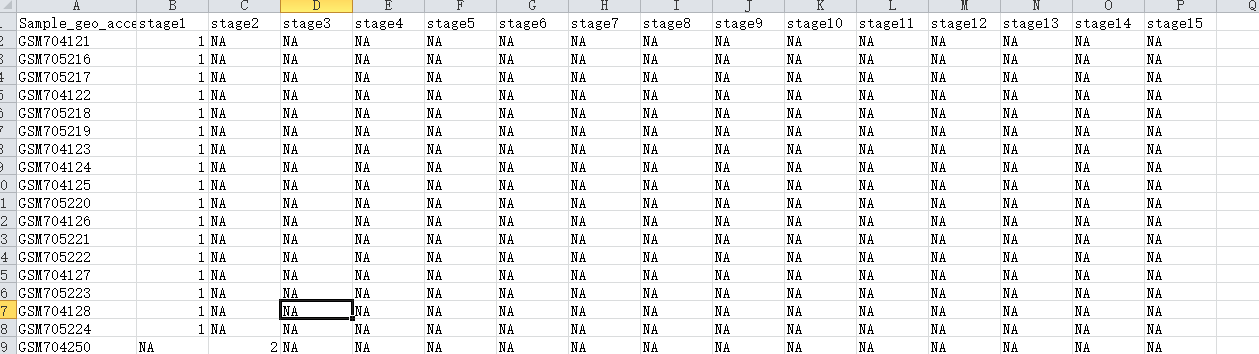
**#案例：针对单个trait进行拆解，进行详细的module相关性计算**

首先将trait进行拆解，分割不同变量：

初始：



调整：



#代码部分和之前的相同；

allTraits1 = read.csv("stage-trait.csv");

femaleSamples = rownames(datExpr);

traitRows1 = match(femaleSamples, allTraits1$Sample\_geo\_accession);

datTraits1 = allTraits1[traitRows1, -1];

rownames(datTraits1) = allTraits1[traitRows1, 1];

collectGarbage();

###重新绘图计算

####3.a将modules和trait进行定量计算(trait里不能有字符信息)

# 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)

moduleTraitCor1 = cor(MEs, datTraits1, use = "p");

moduleTraitPvalue1 = corPvalueStudent(moduleTraitCor1, nSamples);

###将定量的结果画图

sizeGrWindow(10,6)

# Will display correlations and their p-values

textMatrix1 = paste(signif(moduleTraitCor1, 2), "\n(",

signif(moduleTraitPvalue1, 1), ")", sep = "");

dim(textMatrix1) = dim(moduleTraitCor1)

par(mar = c(6, 8.5, 3, 3));

# Display the correlation values within a heatmap plot

labeledHeatmap(Matrix = moduleTraitCor1,

xLabels = names(datTraits1),

yLabels = names(MEs),

ySymbols = names(MEs),

colorLabels = FALSE,

colors = greenWhiteRed(50),

textMatrix = textMatrix1,

setStdMargins = FALSE,

cex.text = 0.5,

zlim = c(-1,1),

main = paste("Module-stage relationships"))

# 4. VisANT 和Cytoscape

## 4.1 VisANT

暂略

## 4.2 Cytoscape

##从module中分析gene间的相关性

# Recalculate topological overlap if needed

TOM = TOMsimilarityFromExpr(datExpr, power = 6);#应该适合前面的power值一样

# 选择要分析的基因，可以以module的形式选择

modules = c("brown", "red");#也可以选择所有基因

# Select module probes

probes = names(datExpr)

inModule = is.finite(match(moduleColors, modules));

modProbes = probes[inModule];

# Select the corresponding Topological Overlap

modTOM = TOM[inModule, inModule];

dimnames(modTOM) = list(modProbes, modProbes)

cyt = exportNetworkToCytoscape(modTOM,

edgeFile = paste("CytoscapeInput-edges-", paste(modules, collapse="-"), ".txt", sep=""),

nodeFile = paste("CytoscapeInput-nodes-", paste(modules, collapse="-"), ".txt", sep=""),

weighted = TRUE,

threshold = 0.02,

nodeNames = modProbes,

altNodeNames = modProbes,

nodeAttr = moduleColors[inModule]);

#如果基因太多，导入到cytoscape时可能会非常慢

从小的gene list中寻找gene的相关性图

genematrix<- read.csv("test.csv",row.names = 1,head=T)

genematrix1=as.data.frame(t(genematrix))

genecor<- cor(genematrix1[,1:length(genematrix1[1,])])

cyt = exportNetworkToCytoscape(genecor,edgeFile = paste("33CytoscapeInput-edges-", paste(modules, collapse="-"), ".txt", sep=""),nodeFile = paste("33CytoscapeInput-nodes-", paste(modules, collapse="-"), ".txt", sep=""),weighted = TRUE,threshold = 0.80);