RNA-seq分析流程构建（tophat-cufflinks/merge/diff, Subread -> featureCounts -> DESeq2, HISAT2 -> HTSeq -> DESeq2）

目标：利用python，对测序数据利用软件进行一系列的处理以及下游分析，输出可以作图的数据，利用matplotlib等做出合适的图。

前期准备：

RNA-seq数据下载（ncbi）

了解RNA-seq的一般分析流程

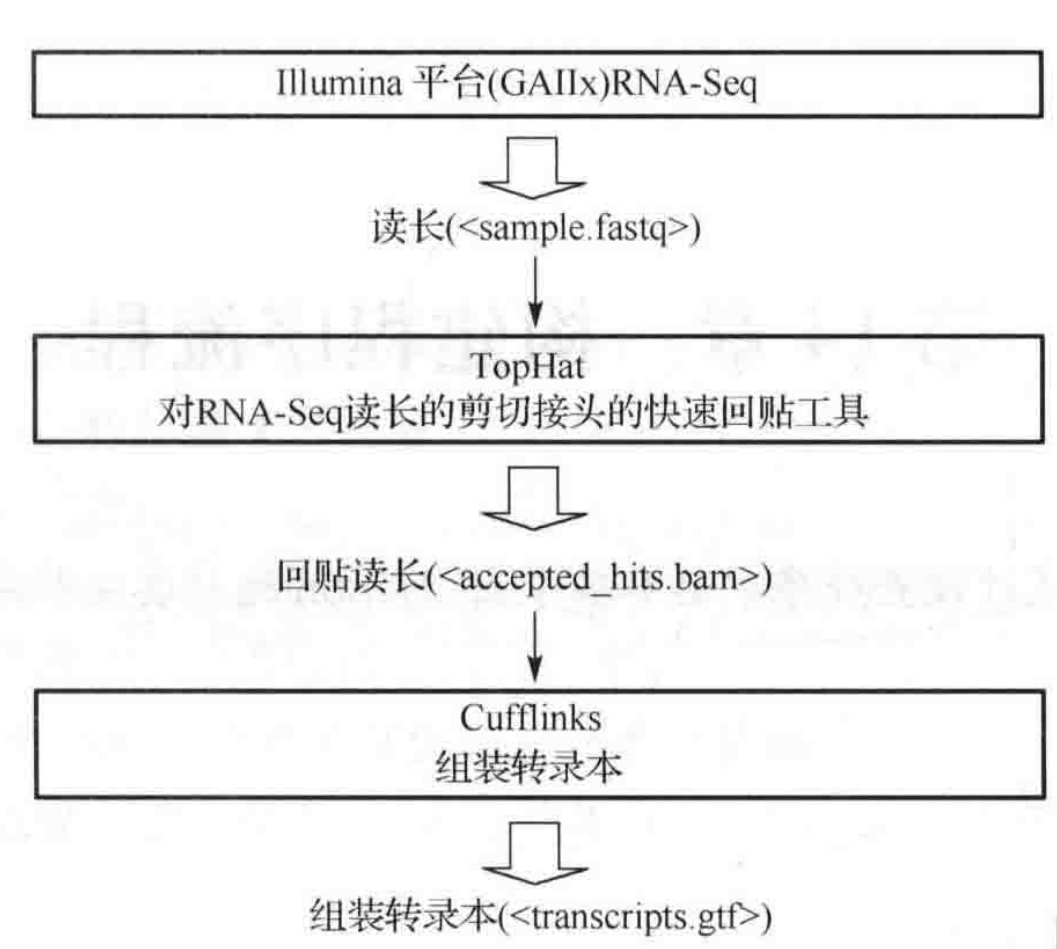
了解后续分析RNA-seq结果的R包

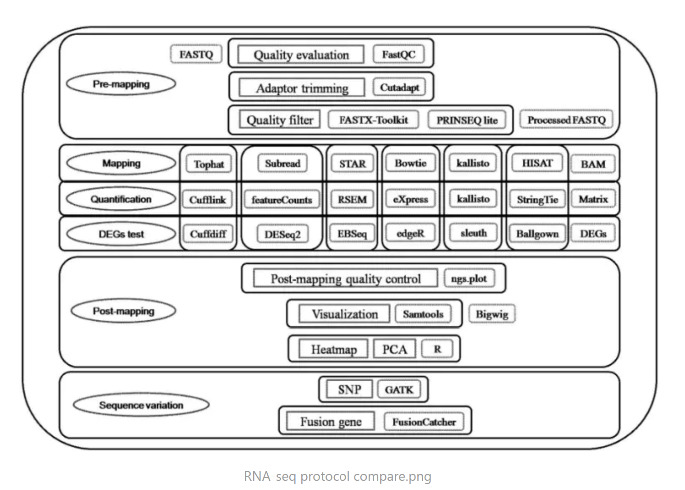
了解python的os模块（帮助运行linux命令）

了解python如何调用R包

将所有的流程都用一个python程序完成

RNA-seq分析的流程：



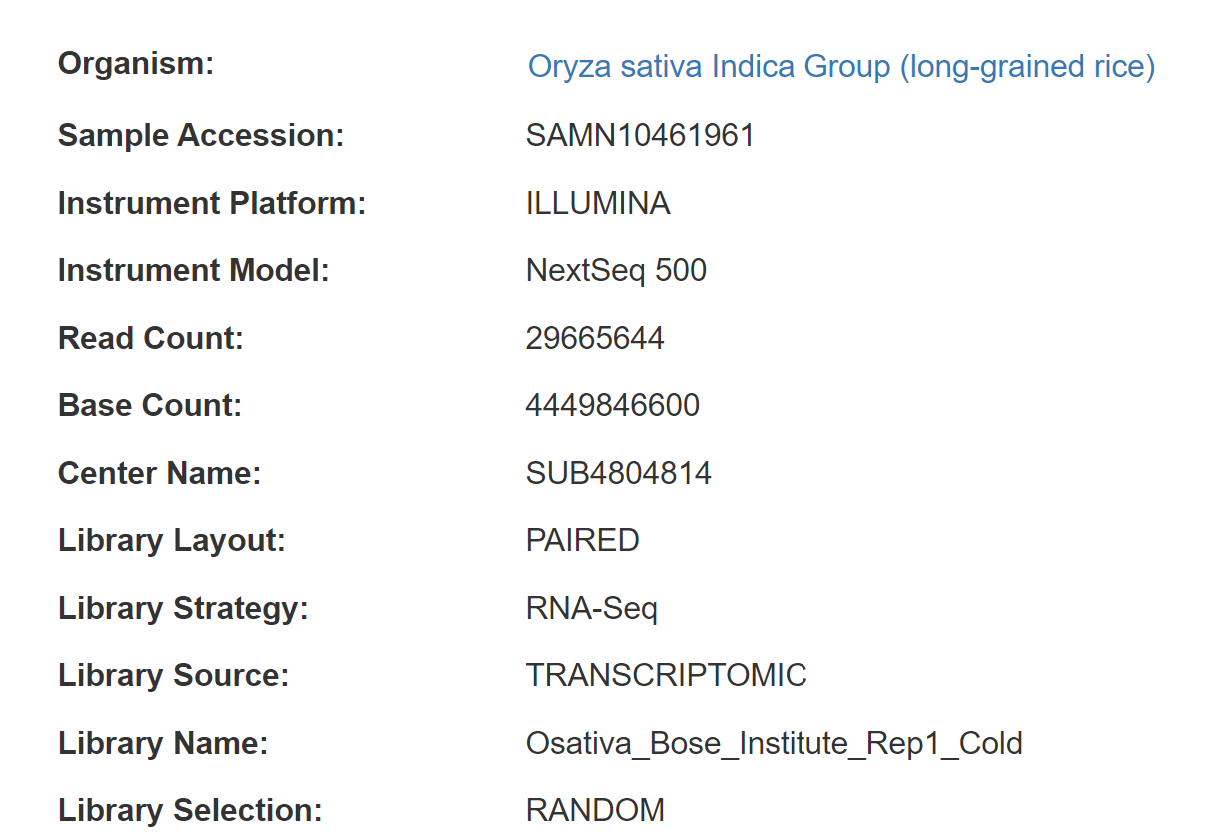


从NCBI-SRA和EBI-ENA上下载水稻RNA-seq的数据以供分析

有几种下载数据的办法Aspera、Sratoolsf、ftp，据说Asprea最快。NCBI上只能下载sra格式的数据，ENA上可以直接下载fastq格式的数据因此更为便利。

下载的数据的具体信息：





下载数据的具体方法：

ascp -QT -l 300m -P33001 -i ~/.aspera/connect/etc/asperaweb\_id\_dsa.openssh era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR823/002/SRR8237122/SRR8237122\_1.fastq.gz .

ascp -QT -l 300m -P33001 -i ~/.aspera/connect/etc/asperaweb\_id\_dsa.openssh era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR823/002/SRR8237122/SRR8237122\_2.fastq.gz .

方法可行，只可惜下载速度还是很慢，而且60秒后会自动timeout（我猜测是这个网站怕用户长时间下载占用资源）。

还是用蠢办法从网页下载数据吧，至少不会timeout

使用Tophat进行比对回帖，具体的使用方法参考tophat使用手册

命令如下

bsub -n 16 -o bowtie2-build.log bowtie2-build rice.fa rice

#rice为输出的索引的前缀

bsub -n 16 -o tophat.log tophat -p 8 --solexa-quals -o SRR11308177 -G /gss1/seqlib/tair10/TAIR10\_GFF3\_genes\_transposons.gff /gss1/seqlib/tair10/TAIR10 /gss1/home/gaohao/RNA\_seq\_process/SRR11308177\_1.fastq /gss1/home/gaohao/RNA\_seq\_process/SRR11308177\_2.fastq

##

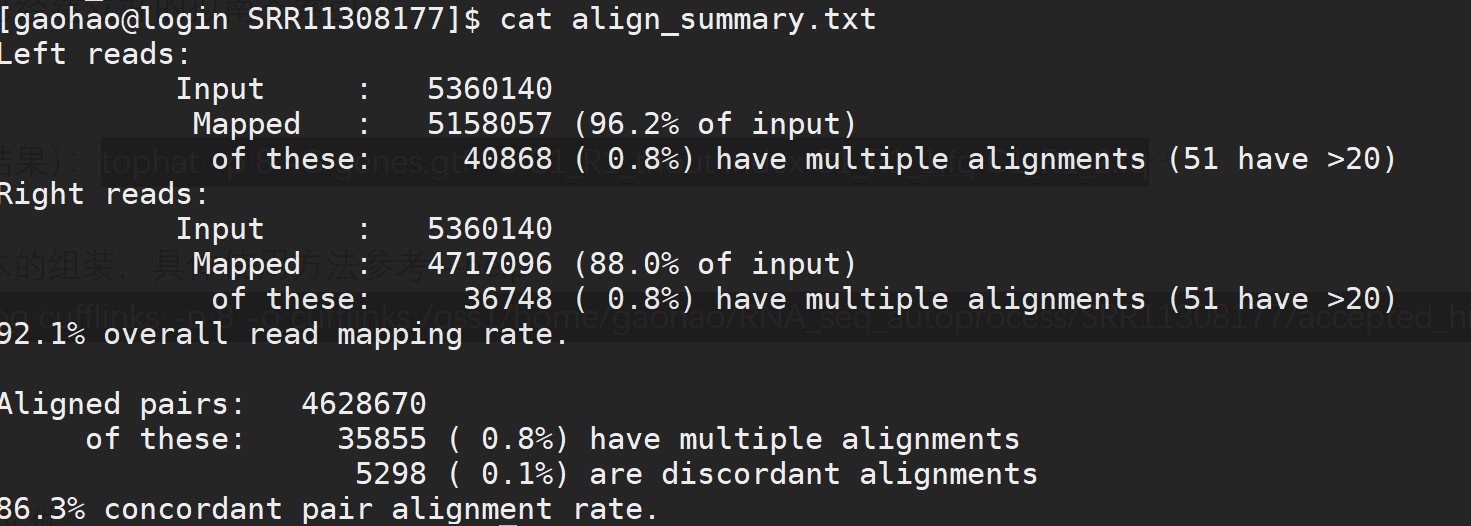
-p 8表示使用八个线程

-o SRR1916154.1表示文件输出到这个文件夹中，文件夹不需要提前创建

-G /gss1/home/gaohao/RNA\_seq\_process/rice\_bowtie2\_index/all.gff3 /gss1/home/gaohao/RNA\_seq\_process/rice\_bowtie2\_index/rice 后面跟随的是基因组注释文件即gff3/gtf/之前创建的索引文件

这里使用的是毛飞师兄已经建立好的拟南芥索引

简单的模板（双端测序结果）：tophat -p 8 -G genes.gtf -o C1\_R1\_thout index C1\_R1\_1.fq C1\_R1\_2.fq



使用cufflinks进行转录本的组装，具体使用方法参考—help

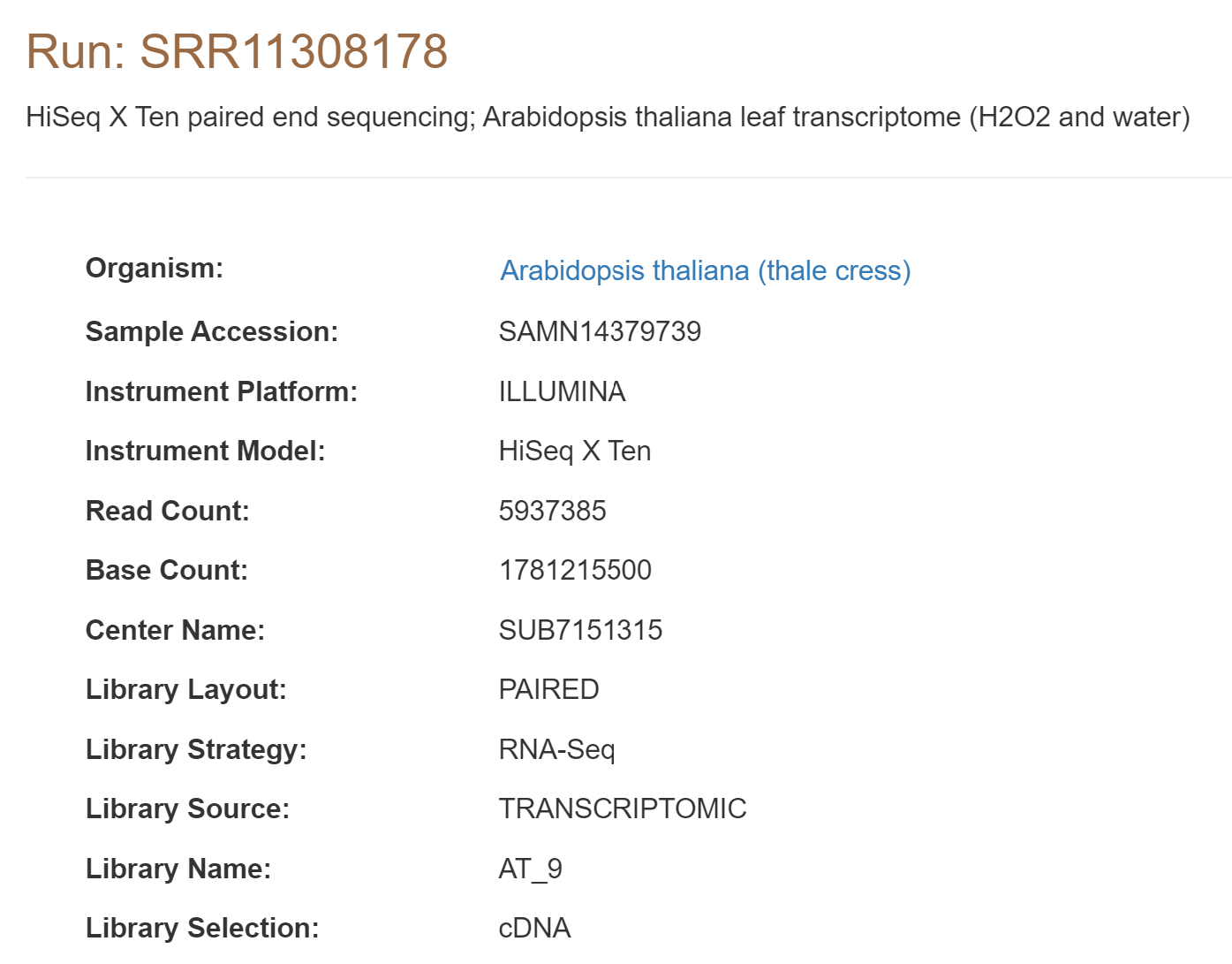
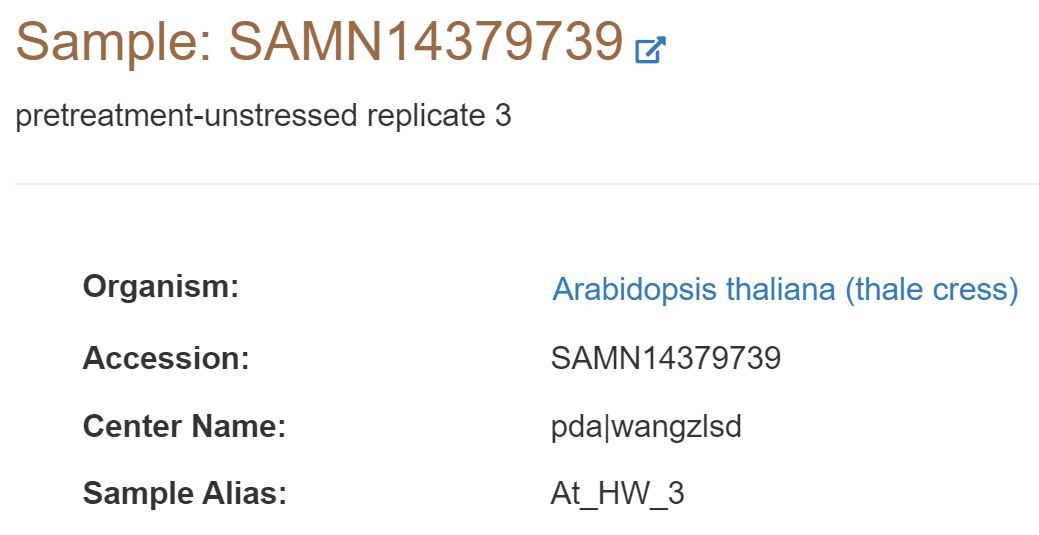
bsub -n 16 -o cufflinks.log cufflinks -p 8 -o cufflinks /gss1/home/gaohao/RNA\_seq\_process/SRR11308177/accepted\_hits.bam

##

-p 线程

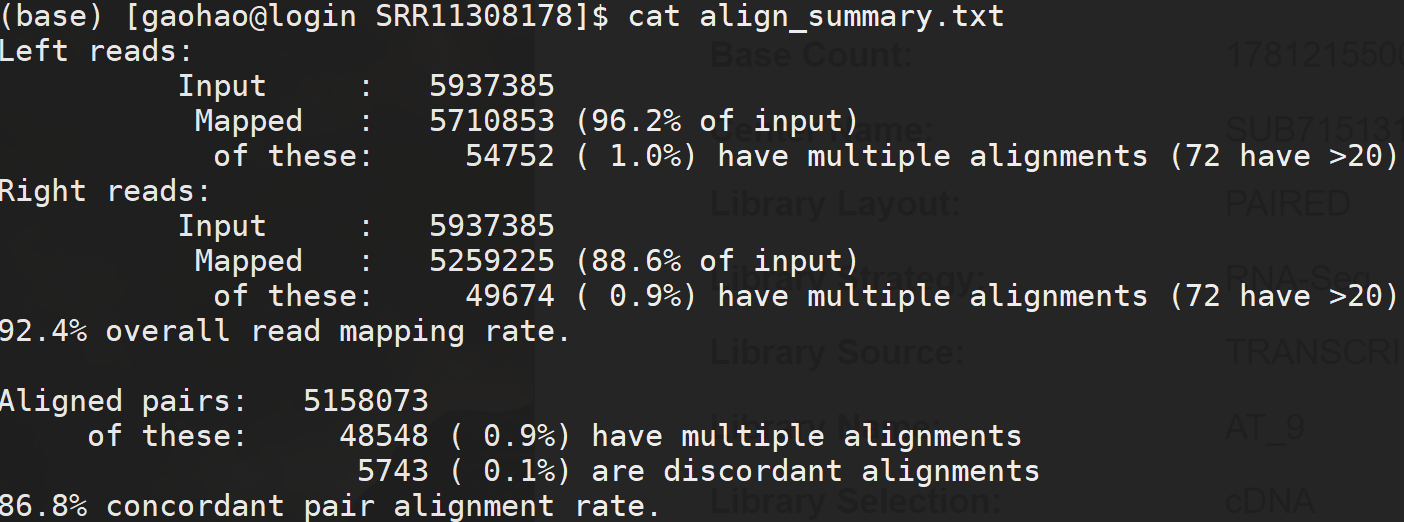
-o 输出文件的目录

后面跟的是输入的bam文件



处理方法类似：

bsub -n 16 -o tophat.log tophat -p 8 --solexa-quals -o SRR11308178 -G /gss1/seqlib/tair10/TAIR10\_GFF3\_genes\_transposons.gff /gss1/seqlib/tair10/TAIR10 /gss1/home/gaohao/RNA\_seq\_process/SRR11308178\_1.fastq /gss1/home/gaohao/RNA\_seq\_process/SRR11308178\_2.fastq



bsub -n 16 -o cufflinks.log cufflinks -p 8 -o cufflinks /gss1/home/gaohao/RNA\_seq\_process/SRR11308178/accepted\_hits.bam

cuffmerge进行转录本数据的融合

mkdir cuffmerge

cd cuffmerge

vim assemblies.txt

里面的内容为gtf文件的路径：

/gss1/home/gaohao/RNA\_seq\_process/SRR11308177/cufflinks/transcripts.gtf

/gss1/home/gaohao/RNA\_seq\_process/SRR11308178/cufflinks/transcripts.gtf

cuffmerge -g /gss1/seqlib/tair10/TAIR10\_GFF3\_genes\_transposons.gff -s /gss1/seqlib/tair10/TAIR10.fa -p 8 assemblies.txt

##

-g 后面跟参考基因组的注释文件

-s 后面跟参考基因组序列文件

-p 线程数

Cuffdiff寻找前后基因表达的差异

mkdir cuffdiff

cd cuffdiff

bsub -n 16 -o cuffdiff.log cuffdiff -o ./diff\_out/ -b /gss1/seqlib/tair10/TAIR10.fa -p 8 -L stressed,unstressed -u /gss1/home/gaohao/RNA\_seq\_process/cuffmerge/merged\_asm/merged.gtf /gss1/home/gaohao/RNA\_seq\_process/SRR11308177/accepted\_hits.bam /gss1/home/gaohao/RNA\_seq\_process/SRR11308178/accepted\_hits.bam

##

cuffdiff [options] <transcripts.gtf> <sample1\_hits.sam> <sample2\_hits.sam> [... sampleN\_hits.sam]

-o 输出文件夹

-b 使用偏差校正-需要参考fasta

-p 线程数目

-L 以逗号分隔的条件标签列表

-u 对muti-reads应用拯救方法

得到的gene\_exp.diff文件中各行的含义

Test\_id：测试id，如XLOC\_000001

Geng\_id：基因id，如XLOC\_000001

Gene：基因名，如AT1G01010可以用来GO分析

Locus：基因的具体位置，如Chr1:3630-5899

Sample\_1：样本一的label

Sample\_2：样本二的label

Status：测试的状态，可以是OK也可以使NOSET

Value\_1：样本一的值，代表了基因在样本一中的表达水平

Value\_2：基因在样本二中的表达水平

Log2（fold\_change）：log2（Value\_2/Value\_1），样本二相对于样本一的基因表达变化

Test\_stat：t测验的状态？

P\_value

Q\_value

Significant：前后变化是否显著

挑选差异表达的基因

awk '{if(($10<-2)&&($11<0.001))print $3"\t"$8"\t"$9"\t"$10}' gene\_exp.diff | grep -v 'inf' > down.txt

awk '{if(($10>2)&&($11<0.001))print $3"\t"$8"\t"$9"\t"$10}' gene\_exp.diff | grep -v 'inf' > up.txt

## 筛选出下调的基因（log2\_fold\_change < -2 & pvalue < 0.001），并输出基因名、表达值、和倍数变化

筛选出上调的基因（log2\_fold\_change > 2 & pvalue < 0.001）

grep -v ‘inf’输出不含inf的所有信息

需要注意的是前面的分析把stessed样本放在了前面把unstressed的样本放在了后面，与一般理解相反，不建议这么干。

下一步一般是画图，GO分析，通路分析之类，可以用专门的R包或者网站做，也可以自己写程序用python画。

把这些流程放在一个.py文件中一次性完成

利用python的os与sys模块，可以编写一个python包装器，调用unix（其他系统也可以）内的软件，具体怎么写，详见《python生物信息学数据管理》这本书

具体的代码如下：

#\_\*\_coding:utf8\_\*\_  
import os  
  
bowtie\_index = '/gss1/seqlib/tair10/TAIR10\_GFF3\_genes\_transposons.gff' + ' ' + '/gss1/seqlib/tair10/TAIR10'  
bowtie\_index\_fa = '/gss1/seqlib/tair10/TAIR10.fa'  
input\_fastq\_dir = '/gss1/home/gaohao/RNA\_seq\_process'  
fastq\_name = ['SRR11308177\_1.fastq','SRR11308177\_2.fastq','SRR11308178\_1.fastq','SRR11308178\_2.fastq']  
sample\_label = ['stressed','unstressed']  
current\_dir = '/gss1/home/gaohao/RNA\_seq\_autoprocess' #程序运行所在的目录  
  
gtf\_path = open('assemblies.txt','w')  
for x in range(1,int(len(fastq\_name)/2)+1):  
 tophat\_command = 'bsub -n 16 -K -o tophat.log tophat -p 8 --solexa-quals -o %s -G %s %s%s%s %s%s%s'%\  
 (fastq\_name[2\*x-1][:11],bowtie\_index,input\_fastq\_dir,os.sep,fastq\_name[2\*x-2],input\_fastq\_dir,os.sep,fastq\_name[2\*x-1])  
 #-K 提交作业，并且等待作业完成。当提交作业后，终端打印“waiting for dispath”。当作业完成后，终端打印“job is finished”。作业没有完成，不能提交新的作业  
 os.system(tophat\_command)  
  
 cufflinks\_command = 'bsub -n 16 -K -o cufflinks.log cufflinks -p 8 -o %s/%s/cufflinks %s%s%s%saccepted\_hits.bam'%\  
 (current\_dir,fastq\_name[2\*x-1][:11],current\_dir,os.sep,fastq\_name[2\*x-1][:11],os.sep)  
 #print(cufflinks\_command)  
 os.system(cufflinks\_command)  
  
 gtf\_path.write('%s%s%s/cufflinks/transcripts.gtf\n'%(current\_dir,os.sep,fastq\_name[2\*x-1][:11]))  
gtf\_path.close()  
  
cuffmerge\_command = 'bsub -n 16 -K -o cuffmerge.log cuffmerge -g %s -s %s -p 8 assemblies.txt'%(bowtie\_index.split(' ')[0],bowtie\_index\_fa)  
#print(cuffmerge\_command)  
os.system('mkdir cuffmerge')  
os.system('cp %s/assemblies.txt %s/cuffmerge/'%(current\_dir,current\_dir))  
os.chdir('%s/cuffmerge'%(current\_dir)) #切换目录要用os.chdir  
os.system(cuffmerge\_command)  
os.chdir(current\_dir)  
  
cuffdiff\_command = 'bsub -n 16 -K -o cuffdiff.log cuffdiff -o ./diff\_out/ -b %s -p 8 -L %s,%s -u %s/cuffmerge/merged\_asm/merged.gtf'%\  
 (bowtie\_index\_fa,sample\_label[0],sample\_label[1],current\_dir)  
for x in range(1,int(len(fastq\_name)/2)+1):  
 cuffdiff\_command = cuffdiff\_command + ' ' + '%s/%s/accepted\_hits.bam'%(current\_dir,fastq\_name[2\*x-1][:11])  
#print(cuffdiff\_command)  
os.system('mkdir cuffdiff')  
os.chdir('%s/cuffdiff'%(current\_dir))  
os.system(cuffdiff\_command)  
os.chdir('%s/cuffdiff/diff\_out'%(current\_dir))  
os.system('''awk '{if(($10<-2)&&($11<0.001))print $3"\t"$8"\t"$9"\t"$10}' gene\_exp.diff | grep -v 'inf' > down.txt''')  
os.system('''awk '{if(($10>2)&&($11<0.001))print $3"\t"$8"\t"$9"\t"$10}' gene\_exp.diff | grep -v 'inf' > up.txt''')

运行这一RNA\_seq.py可以运行tophat、cufflinks、cuffmerge、cuffdiff并挑选出基因表达水平上升与下降的基因。

新的分析流程Subread -> featureCounts -> DESeq2

SRR11308177 stressed\_replicate1

SRR11308185 stressed\_replicate2

ascp -QT -l 300m -P33001 -i ~/.aspera/connect/etc/asperaweb\_id\_dsa.openssh era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR113/085/SRR11308185/SRR11308185\_1.fastq.gz .

ascp -QT -l 300m -P33001 -i ~/.aspera/connect/etc/asperaweb\_id\_dsa.openssh era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR113/085/SRR11308185/SRR11308185\_2.fastq.gz .

SRR11308178 unstressed\_replicate1

SRR11308179 unstressed\_replicate2

ascp -QT -l 300m -P33001 -i ~/.aspera/connect/etc/asperaweb\_id\_dsa.openssh era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR113/079/SRR11308179/SRR11308179\_1.fastq.gz .

ascp -QT -l 300m -P33001 -i ~/.aspera/connect/etc/asperaweb\_id\_dsa.openssh era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR113/079/SRR11308179/SRR11308179\_2.fastq.gz .

这种下载方法突然又好使了，下载速度可以达到20Mb/s，确实比ftp要快多了，之前无法下载不知道是什么环节抽风了，反正前后的操作是一样的

subread-buildindex 建立索引

mkdir subread\_index

cd subread\_index

subread-buildindex -o TAIR10 TAIR10.fa

# TAIR10.fa 为之前一样的拟南芥参考基因组序列

# -o 输出索引的前缀

subjunc 进行比对

mkdir align

cd align

bsub -n 16 -o subjunc.log subjunc -T 5 -i /gss1/home/gaohao/RNA\_seq\_process1/subread\_index/TAIR10 -r /gss1/home/gaohao/RNA\_seq\_process/SRR11308177\_1.fastq -R /gss1/home/gaohao/RNA\_seq\_process/SRR11308177\_2.fastq -o SRR11308177

bsub -n 16 -o subjunc.log subjunc -T 5 -i /gss1/home/gaohao/RNA\_seq\_process1/subread\_index/TAIR10 -r /gss1/home/gaohao/RNA\_seq\_process/SRR11308185\_1.fastq -R /gss1/home/gaohao/RNA\_seq\_process/SRR11308185\_2.fastq -o SRR11308185

bsub -n 16 -o subjunc.log subjunc -T 5 -i /gss1/home/gaohao/RNA\_seq\_process1/subread\_index/TAIR10 -r /gss1/home/gaohao/RNA\_seq\_process/SRR11308178\_1.fastq -R /gss1/home/gaohao/RNA\_seq\_process/SRR11308178\_2.fastq -o SRR11308178

bsub -n 16 -o subjunc.log subjunc -T 5 -i /gss1/home/gaohao/RNA\_seq\_process1/subread\_index/TAIR10 -r /gss1/home/gaohao/RNA\_seq\_process/SRR11308179\_1.fastq -R /gss1/home/gaohao/RNA\_seq\_process/SRR11308179\_2.fastq -o SRR11308179

## 参数说明 (具体说明直接运行 subjunc )

-T 线程数

-i 之前建立的基因组索引的前缀

-r 接fastq文件

-R 双端测序的fastq文件

-o 输出名

samtools view file.bam |head

## 查看bam文件命令,比如查看前几行

featureCounts 进行定量

mkdir featureCounts

cd featureCounts

bsub -n 16 -o featureCounts.log featureCounts -p -T 6 -a /gss1/home/gaohao/RNA\_seq\_process1/subread\_index/TAIR10\_GFF3\_genes\_transposons.gtf -o stressed7785-unstressed7879\_fCount.out /gss1/home/gaohao/RNA\_seq\_process1/align/SRR11308177 /gss1/home/gaohao/RNA\_seq\_process1/align/SRR11308185 /gss1/home/gaohao/RNA\_seq\_process1/align/SRR11308178 /gss1/home/gaohao/RNA\_seq\_process1/align/SRR11308179

提取定量的信息

awk -F '\t' '{print $1,$7,$8,$9,$10}' OFS='\t' stressed7785-unstressed7879\_fCount.out > stressed\_unstressed\_matrix.out

## stressed7785-unstressed7879\_fCount.out数据中只有第一列和从第七列(即第一个bam文件)到最后一列信息才是我们所需要的，即我们这里有四个bam文件所以提取第7列到第10列

\t 表示以制表符分割开来

将矩阵导入R中，采用DESeq2进行差异分析(同一样本必须要有重复，这是DESeq2分析的要求)

countdata <- read.table("stressed\_unstressed\_matrix.out", header=TRUE, row.names=1) #导入数据

head(countdata) # 查看数据前几行

colnames(countdata) <- c("stressed\_rep1","stressed\_rep2","unstressed\_rep1","unstressed\_rep2") # 修改列名

dim(countdata) # 查看数据维度，即几行几列

condition <- factor(c("stressed","stressed","unstressed","unstressed")) # 赋值因子，即变量

library(DESeq2) # 这一步以后基本可以复制粘贴

coldata <- data.frame(row.names=colnames(countdata), condition) # 创建一个condition数据框

dds <- DESeqDataSetFromMatrix(countData=countdata, colData=coldata, design=~condition) ##构建dds矩阵(后面很多分析都是基于这个dds矩阵)

dds <- DESeq(dds)

resdata <- results(dds)

table(resdata$padj<0.05) # p<0.05的基因数(FALSE 15039 TRUE 3813)

res\_padj <- resdata[order(resdata$padj), ] ##按照padj列的值排序

write.table(resdata, file="diffexpr\_padj\_results.csv",sep = "\t",row.names = T) ## 将结果文件保存到本地

## 筛选差异基因

subset(resdata,pvalue < 0.001) -> diff ## 先筛选pvalue < 0.001的行

subset(diff,log2FoldChange < -2) -> down ## 筛选出下调的基因（后面的相对于前面的）

subset(diff,log2FoldChange > 2) -> up ## 筛选出上调的基因

write.table(up, file="up.csv",sep = "\t",row.names = T) #结果文件保存到本地

write.table(down, file="down.csv",sep = "\t",row.names = T)

利用ggplot2绘制火山图

rm(list=ls()) #删除所有的变量

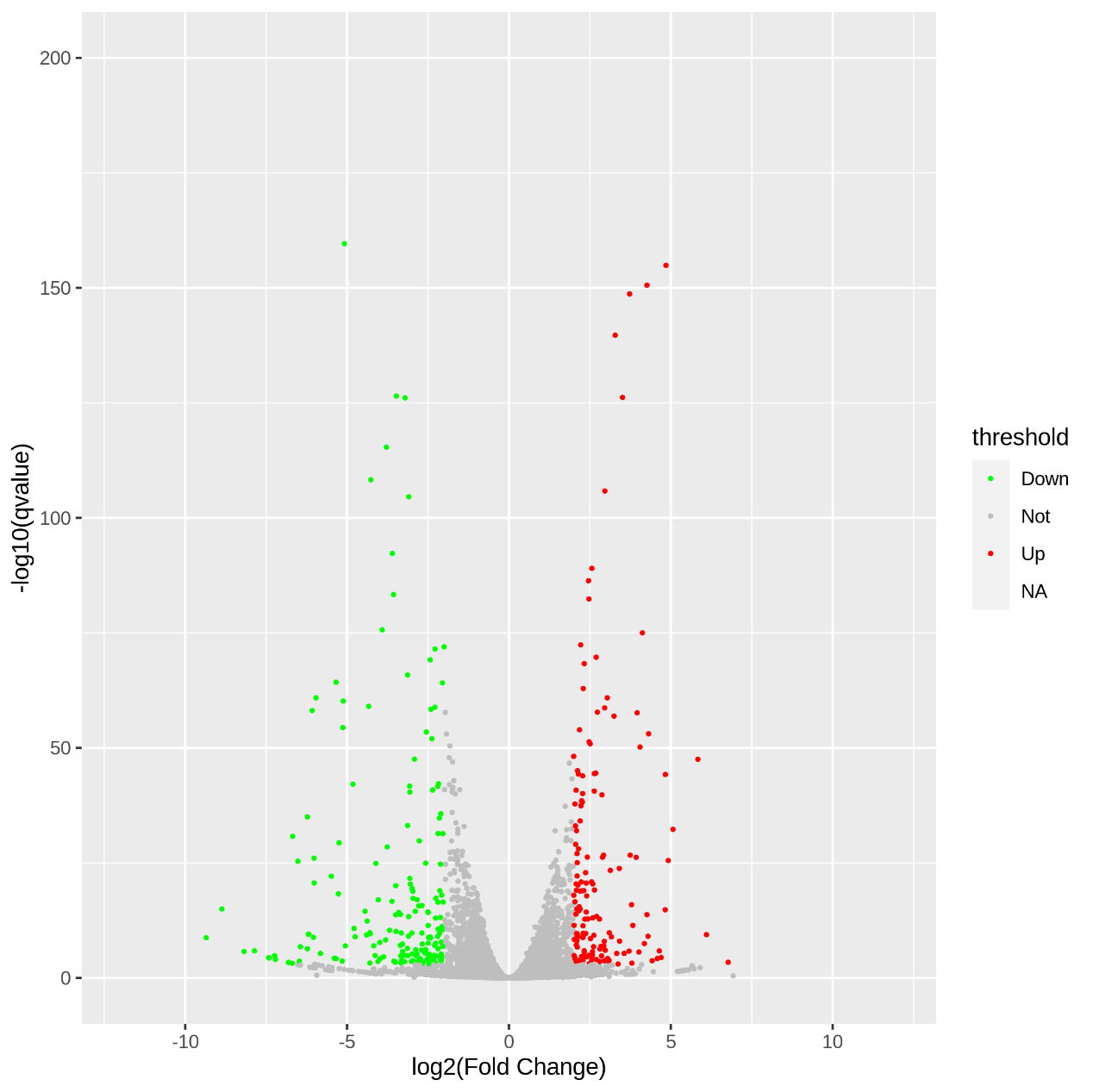
resdata <- read.csv(file.choose(),header = T , sep = "\t") ## 加载DESeq2中生成的resdata文件

threshold <- as.factor(ifelse(resdata$padj < 0.001 & abs(resdata$log2FoldChange) >= 2 , ifelse(resdata$log2FoldChange >= 2 ,'Up','Down'),'Not')) #如果p值显著，当倍数变化的绝对值大于2时，标记为up或者down，否则为not，如果大于2则为up，否则为down

library(ggplot2)

ggplot(resdata,aes(x=log2FoldChange,y=-log10(padj),colour=threshold)) + xlab("log2(Fold Change)")+ylab("-log10(qvalue)") + geom\_point(size = 0.5,alpha=1) + ylim(0,200) + xlim(-12,12) + scale\_color\_manual(values=c("green","grey", "red"))

ggsave("volcano.png") # path设定图形储存路径。ggsave() 可以生成以下格式： .eps, .pdf, .svg, .wmf,.png, .jpg, .bmp, and .tiff.



GO富集及clusterProfiler画图

<http://systemsbiology.cau.edu.cn/chromstates_tt/GO_analysis.php>

BiocManager::install("clusterProfiler")

BiocManager::install("org.At.tair.db") # 拟南芥的GO注释信息

library(clusterProfiler)

library(org.At.tair.db)

keytypes(org.At.tair.db) # 查看提供哪些id类型

getwd() #获得工作路径

genes <- read.delim('down1.txt', header = TRUE, stringsAsFactors = FALSE)[[1]] #读取基因列表文件中的基因名称，文件里面就是DESeq2输出的表达水平显著下降的基因名列表

#GO富集分析

#对于加载的注释库的使用，以上述为例，就直接在 OrgDb 中指定人（org.Hs.eg.db）或绵羊（sheep）

enrich.go <- enrichGO(gene = genes, #基因列表文件中的基因名称

OrgDb = 'org.At.tair.db', #指定物种的基因数据库

keyType = 'TAIR', #指定给定的基因名称类型，例如这里以 TAIR 为例

ont = 'ALL', #可选 BP、MF、CC，也可以指定 ALL 同时计算 3 者

pAdjustMethod = 'fdr', #指定 p 值校正方法

pvalueCutoff = 0.05, #指定 p 值阈值，不显著的值将不显示在结果中

qvalueCutoff = 0.2, #指定 q 值阈值，不显著的值将不显示在结果中

readable = FALSE)

#例如上述指定 ALL 同时计算 BP、MF、CC，这里将它们作个拆分后输出

BP <- enrich.go[enrich.go$ONTOLOGY=='BP', ]

CC <- enrich.go[enrich.go$ONTOLOGY=='CC', ]

MF <- enrich.go[enrich.go$ONTOLOGY=='MF', ]

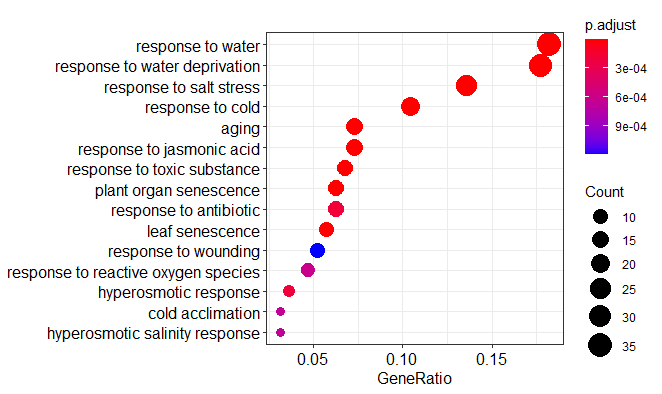
write.table(as.data.frame(BP), 'go.BP.txt', sep = '\t', row.names = FALSE, quote = FALSE)

write.table(as.data.frame(CC), 'go.CC.txt', sep = '\t', row.names = FALSE, quote = FALSE)

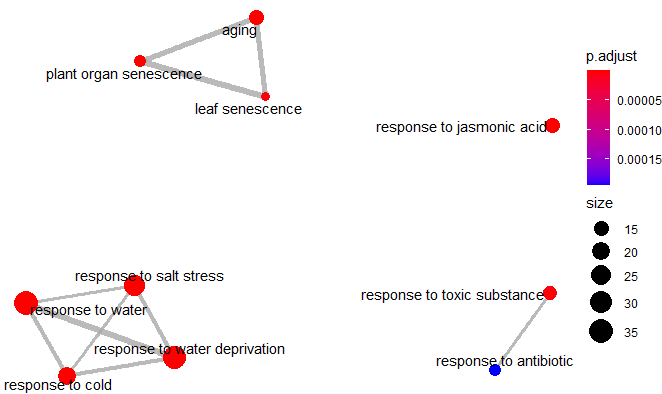
write.table(as.data.frame(MF), 'go.MF.txt', sep = '\t', row.names = FALSE, quote = FALSE)

下面是GO分析之后用clusterProfiler接着画的图（点状图、GOterms关系网络图、条形图、GO树状图）就用down的enrich.go为例

dotplot(enrich.go, showCategory=15)



emapplot(enrich.go,showCategory=10)



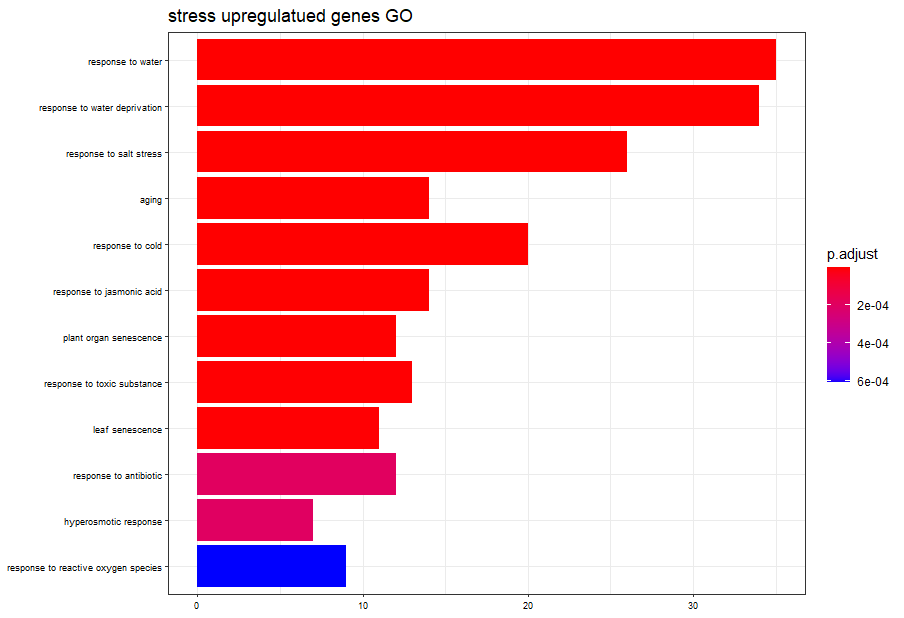
#对于富集到的GO terms之间的基因重叠关系进行展示

#每个节点是一个富集到的GO term，默认画top30个富集到的GO terms

#节点大小对应该GO terms下富集到的基因个数，节点的颜色对应p.adjust的值，红色小蓝色大

#如果两个GO terms的差异基因存在重叠，说明这两个节点存在overlap关系，用线条连接起来

barplot(enrich.go,showCategory=12,font.size=7,title="stress upregulatued genes GO")



library(topGO)

enrich.go.BP <- enrichGO(gene = genes, #基因列表文件中的基因名称

OrgDb = 'org.At.tair.db', #指定物种的基因数据库

keyType = 'TAIR', #指定给定的基因名称类型，例如这里以 TAIR 为例

ont = 'BP', #可选 BP、MF、CC，也可以指定 ALL 同时计算 3 者

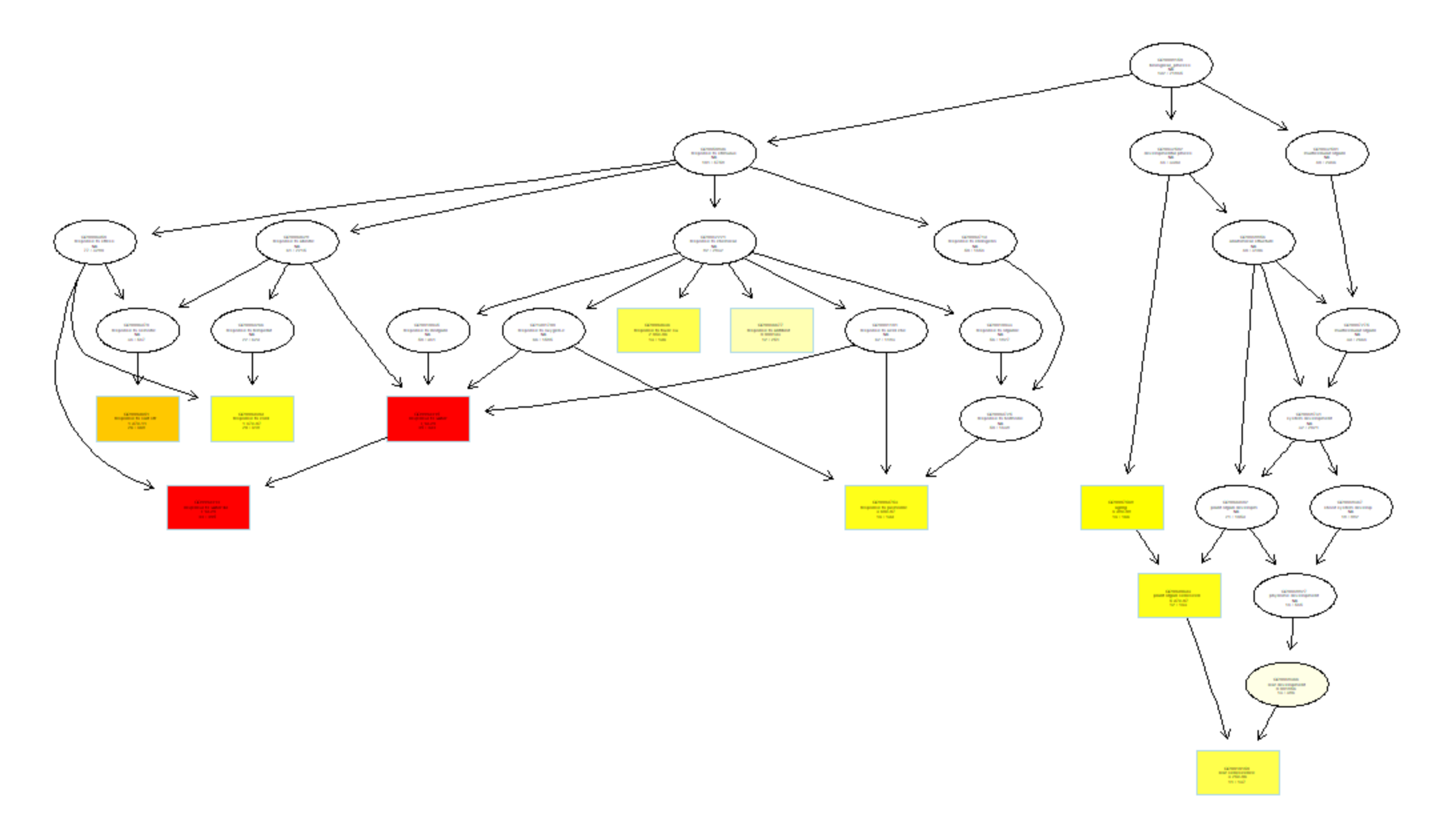
pAdjustMethod = 'fdr', #指定 p 值校正方法

pvalueCutoff = 0.05, #指定 p 值阈值，不显著的值将不显示在结果中

qvalueCutoff = 0.2, #指定 q 值阈值，不显著的值将不显示在结果中

readable = FALSE)

plotGOgraph(enrich.go.BP)



利用DESeq2的分析结果画热图、相关系数图、散点图

热图

rm(list=ls()) ## 清除当前环境变量

resdata <- read.csv(file.choose(),header = T , sep = "\t") ## 导入数据diffexpr\_padj\_results.csv

countdata <- read.table("stressed\_unstressed\_matrix.out", header=TRUE, row.names=1)

colnames(countdata) <- c("stressed\_rep1","stressed\_rep2","unstressed\_rep1","unstressed\_rep2") #修改列名

total\_data<-cbind(resdata,countdata) #将两个表横向拼接在一起

subset(total\_data,padj < 0.001 & abs(total\_data$log2FoldChange) >= 2) -> diff\_gene ## 筛选出差异基因

diff\_gene[,7:ncol(diff\_gene)] -> heatmap\_data ## 提取counts value所在列，ncol(diff\_gene) ## 表示该数据有多少列

as.matrix(heatmap\_data) -> x ## 转换数据格式

library(pheatmap) # 加载包

pheatmap(x,scale="row",cellwidth=80,cellheight=1, ## 设置热图每个格子的宽高

cluster\_col = F,cluster\_row= F, ## 按行还是按列聚类，一般按行，即基因

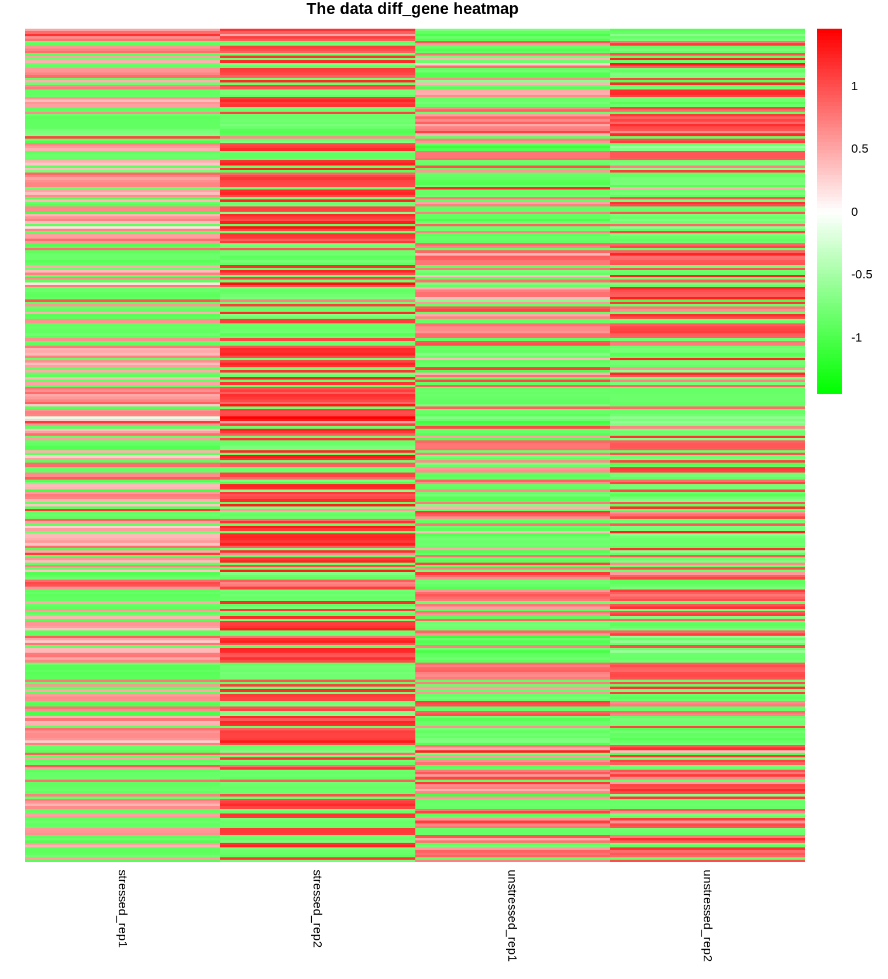
main="The data diff\_gene heatmap ", #标题

fontsize=5,treeheight\_row = 2,show\_rownames= F, ## 是否显示行名(基因名)

cutree\_row=1,display\_numbers = FALSE, ## 是否显示数值

color = colorRampPalette(c("green","white","red"))(100), ## 设置颜色

clustering\_distance\_rows = "correlation", filename="heatmap.png")



相关系数图

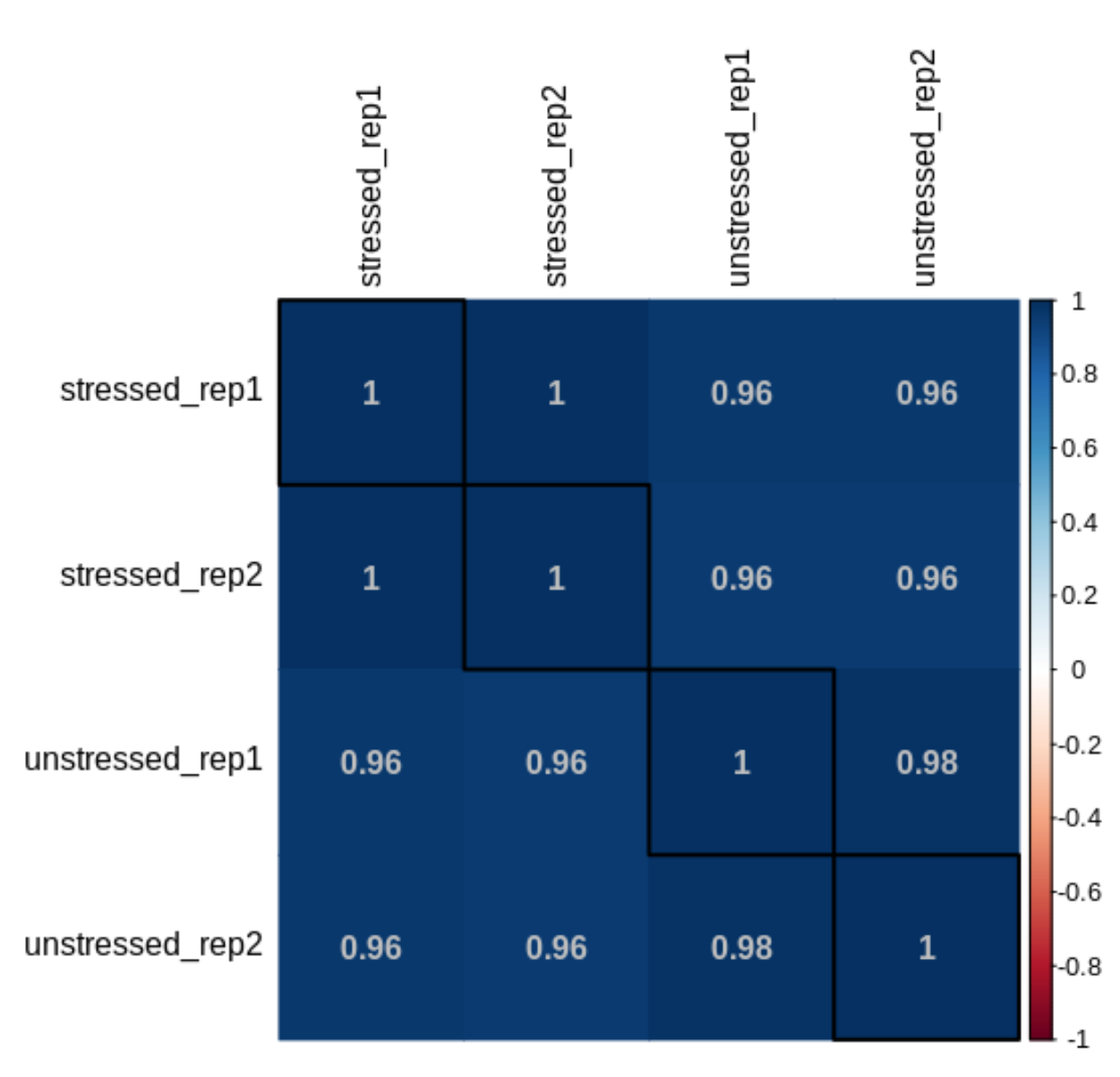
library(corrplot)

total\_data[,7:ncol(total\_data)] -> heatmap\_data ## 提取counts value所在的列

cor(heatmap\_data) -> cor\_data ## 计算相关系数值

corrplot(cor\_data,type = "upper",tl.col="black") ## 绘制相关系数图

corrplot(cor\_data, cor\_data,method = "color",order = "hclust",tl.col="black",addrect=4,addCoef.col = "grey")



散点图(这里也是表现的相关性)

library(ggplot2)

ggplot(resdata,aes(x= total\_data$ stressed\_rep1,y= total\_data$ stressed\_rep2)) +

geom\_point() + xlim(0,80000) + ylim(0,80000) +

geom\_abline() + theme\_bw() +

xlab("stressed\_rep1") +ylab("stressed\_rep2") +

theme(panel.border = element\_blank(),panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),axis.line = element\_line(colour = "black"))

ggplot(resdata,aes(x= total\_data$ unstressed\_rep1,y= total\_data$ unstressed\_rep2)) +

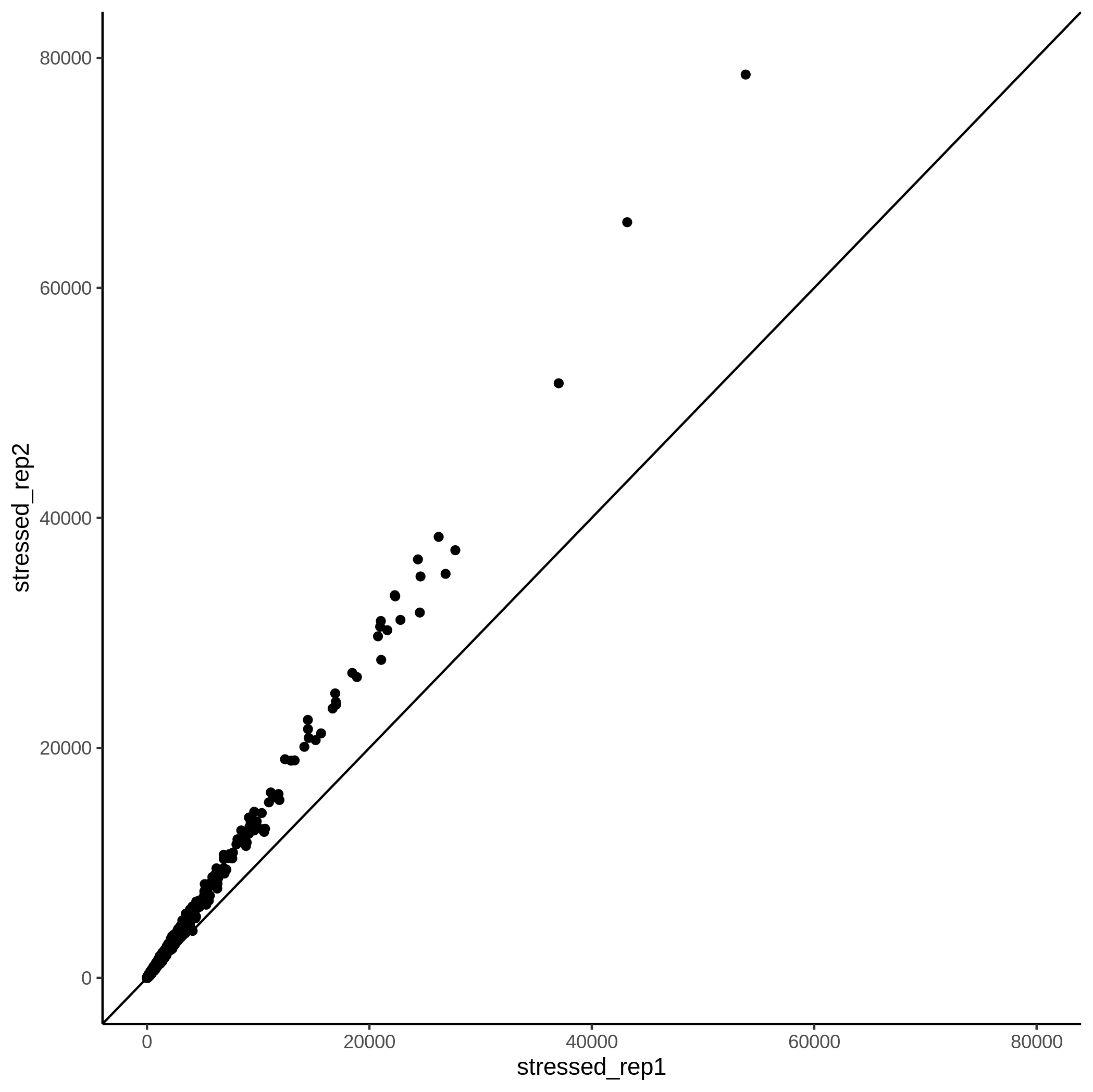
geom\_point() + xlim(0,80000) + ylim(0,80000) +

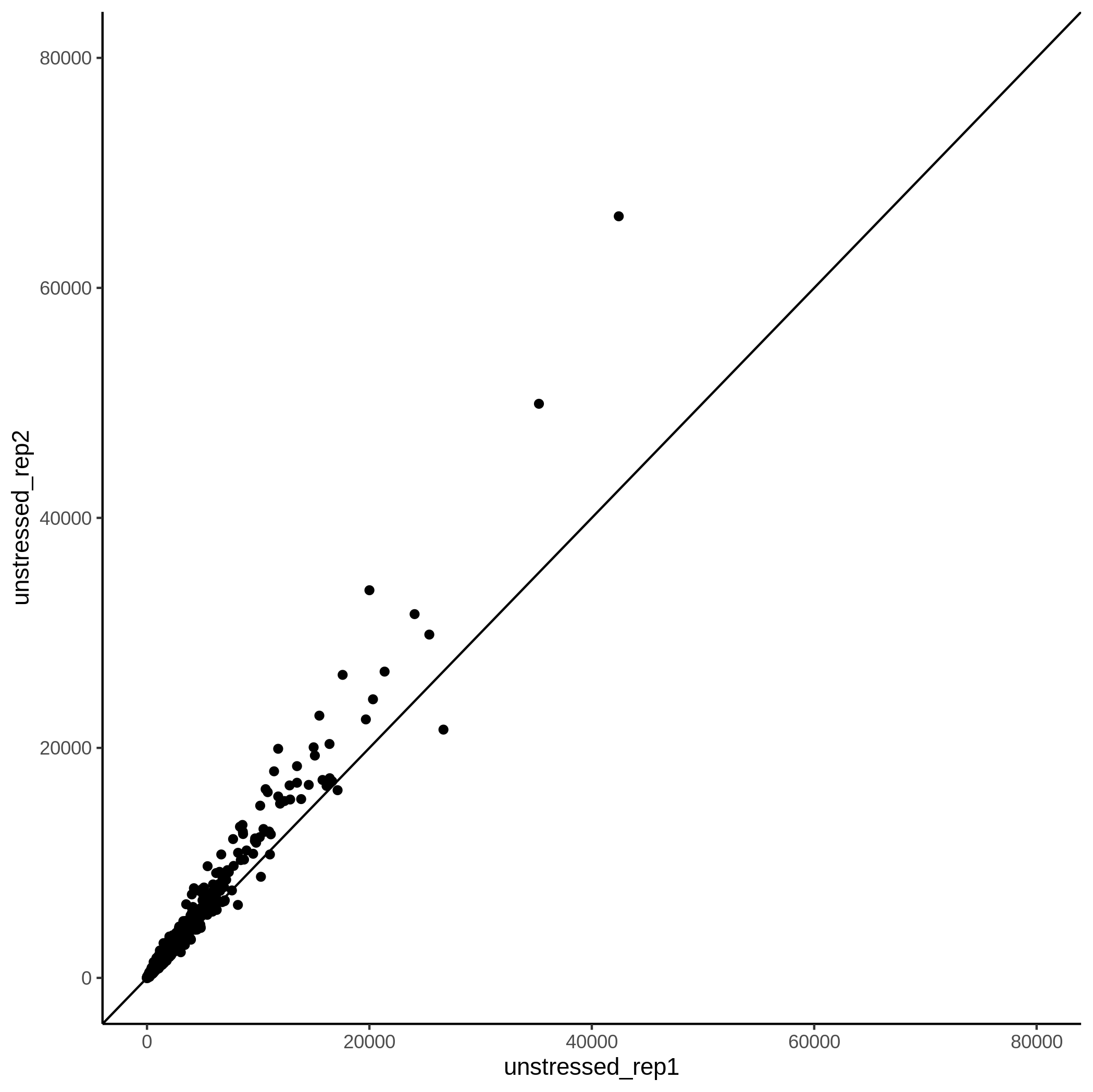
geom\_abline() + theme\_bw() +

xlab("unstressed\_rep1") + ylab("unstressed\_rep2") +

theme(panel.border = element\_blank(),panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),axis.line = element\_line(colour = "black"))





如何在python里调用R包，虽然比较鸡肋，但还是看一下（参见《python生物信息学数据管理》第十三章）