```
load(file = "expr_df.Rdata")
```

	gene_id	N1	N2	N3	X1	X2	X3
1	ENSG00000223972	2	0	1	6	1	0
2	ENSG00000227232	107	88	87	62	81	58
3	ENSG00000278267	2	1	10	8	3	6
4	ENSG00000243485	0	1	0	1	0	0
5	ENSG00000284332	0	0	0	0	0	0
6	ENSG00000237613	0	0	0	0	0	0
7	ENSG00000268020	0	0	0	0	0	0
8	ENSG00000240361	0	0	0	0	0	0
9	ENSG00000186092	0	0	0	0	0	0
10	ENSG00000238009	0	1	0	0	0	0
11	ENSG00000239945	0	0	0	0	0	0
12	ENSG00000233750	2	3	5	3	2	0
13	ENSG00000268903	96	101	78	107	131	86
14	ENSG00000269981	70	85	95	101	82	70
15	ENSG00000239906	0	0	0	0	3	0
16	ENSG00000241860	7	6	5	6	3	3
17	ENSG00000222623	0	0	0	0	0	0
18	ENSG00000241599	0	0	0	0	0	0

第一列是ensemble的ID，后面是3V3的样本

构建metadata文件

> 使用DEseq分析时，需要制作一个分组指示信息

```
metadata <- data.frame(sample_id = colnames(expr_df)[-1])
sample <- rep(c("con","treat"),each=3)
metadata$sample <- relevel(factor(sample),"con")
```

	sample_id	sample
1	N1	con
2	N2	con
3	N3	con
4	X1	treat
5	X2	treat
6	X3	treat

构建dds对象

这一步由`DESeqDataSetFromMatrix`这个函数来完成，他需要输入我们的表达矩阵，制作好的metadata，还要制定分组的列，在这里是sample，最后一个tidy的意思是，我们第一列是基因ID，需要自动处理。

```
library(DESeq2)
## 第一列有名称，所以tidy=TRUE
dds <-DESeqDataSetFromMatrix(countData=expr_df,
                             colData=metadata,
                             design=~sample,
                             tidy=TRUE)
```

数据过滤

别看有这么多行，不是每一个就要都要表达的，如果一行基因在所有样本中的counts数小于等于1，我们就把他删掉，实际上，不做这一步，对差异分析的结果没有影响，可能会对GSEA的结果有影响。

```
dds <- dds[rowSums(counts(dds))>1,]
```

样本聚类

有时候，我们会把实验样本的的标签搞错，比如，明明是加药组，但是有一组没忘记加药，如果对他聚类，他会被划归为正常组，这个是需要我们知道的。

用vst来标化数据，实际上还有rlog方法，或者就是log2的方法，官网推荐<30个样本用rlog，大于30个样本用vst，速度更快，这里我们不要计较那么多了，就用vst，因为真实的TCGA数据，样本往往大于30个。

```
vsd <- vst(dds, blind = FALSE)
```

再用dist这个函数计算样本间的距离

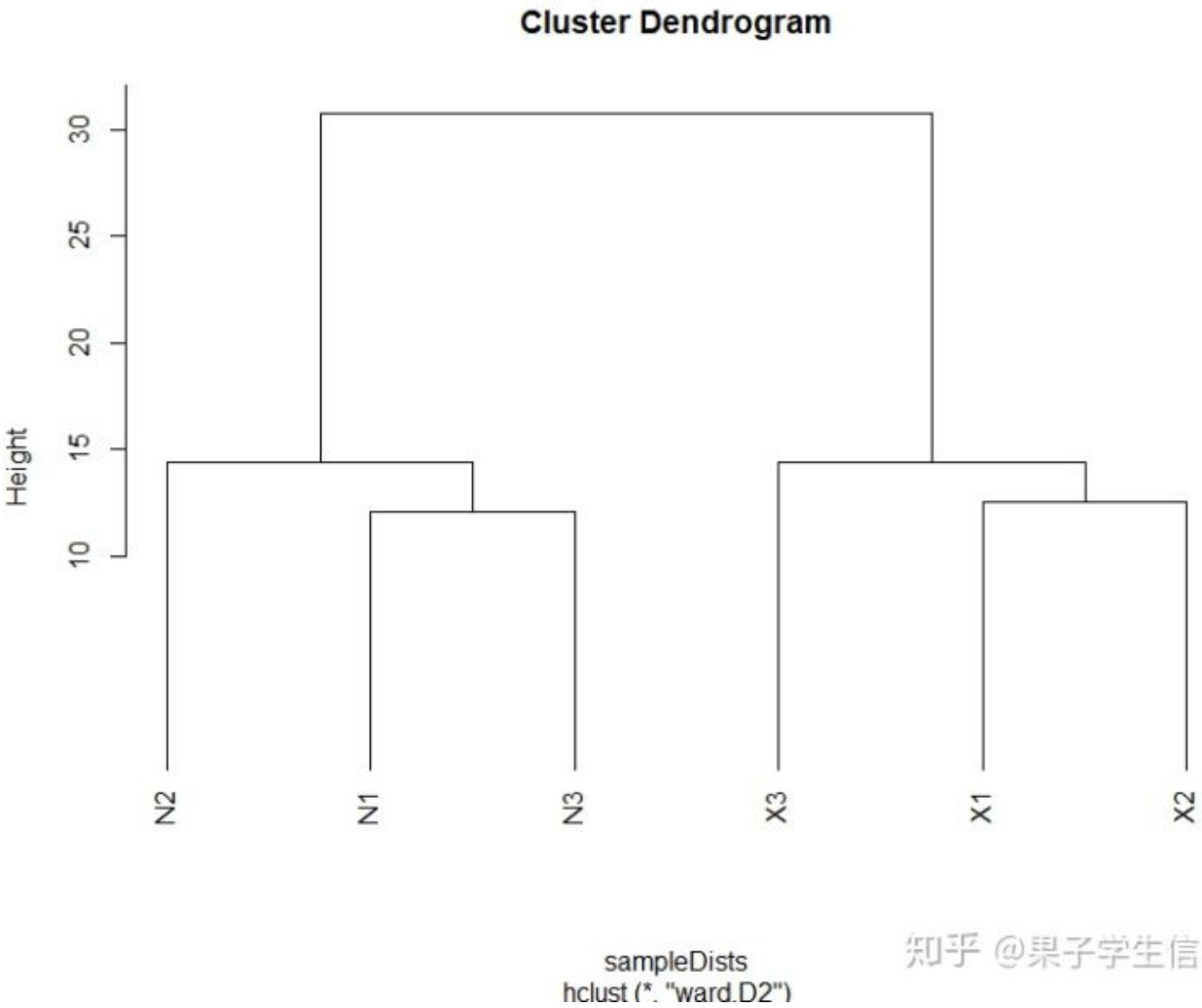
```
sampleDists <- dist(t(assay(vsd)))
```

用hclust来进行层次聚类

```
hc <- hclust(sampleDists, method = "ward.D2")
```

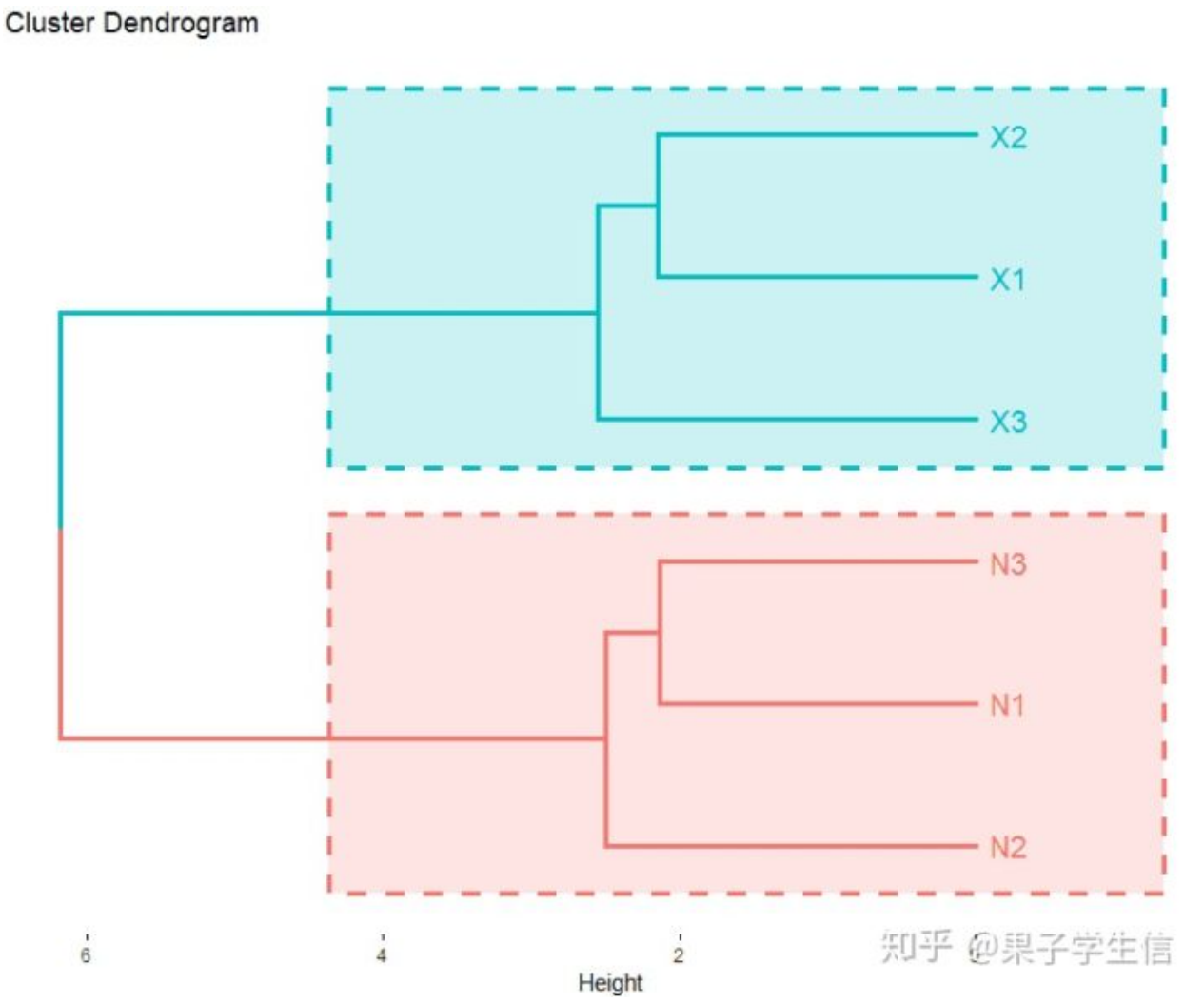
然后画图

```
plot(hc, hang = -1)
```




```
library(factoextra)
res <- hcut(sampleDists, k = 2, stand = TRUE)
# Visualize
fviz_dend(res,
          # 加边框
          rect = TRUE,
          # 边框颜色
          rect_border="cluster",
          # 边框线条类型
          rect_lty=2,
          # 边框线条粗细
          lwd=1.2,
          # 边框填充
          rect_fill = T,
          # 字体大小
          cex = 1,
```

```
          # 字体颜色
          color_labels_by_k=T,
          # 平行放置
          horiz=T)
```



Deseq2 计算

主程序是Deseq这个函数，里面顺序执行了一系列函数，每一步都可以单独运行。这一步，只有6个样本基本上就是10s以内，如果是1000个样本，小电脑跑不过去，跑过去也需要5个小时以上，很耗时间。

```
dds <- DESeq(dds)
```

```
> dds <- DESeq(dds)
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
```

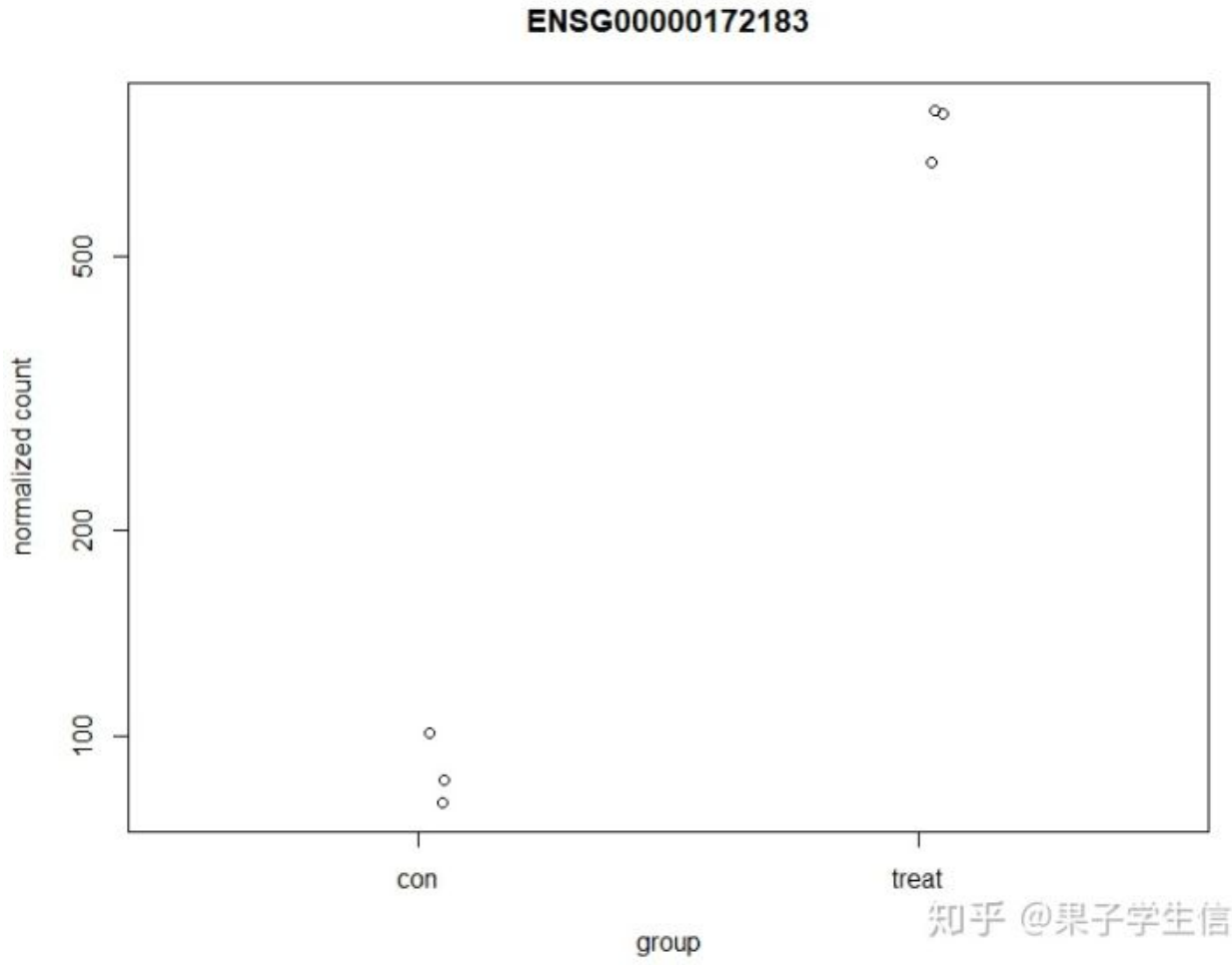

得到dds之后，我们可以通过counts这个函数得到能作图的标注化后的counts数据，他矫正了样本间测序的深度，使得样本间可以直接比较。

```
normalized_counts <- as.data.frame(counts(dds,
normalized=TRUE))
```

	N1	N2	N3	X1	X2	X3
ENSG00000223972	2.031542	0.000000e+00	1.031751	6.063759	9.651299e-01	0.000000
ENSG00000227232	108.687471	7.817833e+01	89.762350	62.658845	7.817552e+01	62.136629
ENSG00000278267	2.031542	8.883902e-01	10.317511	8.085012	2.895390e+00	6.427927
ENSG00000243485	0.000000	8.883902e-01	0.000000	1.010627	0.000000e+00	0.000000
ENSG00000233750	2.031542	2.665170e+00	5.158756	3.031880	1.930260e+00	0.000000
ENSG00000268903	97.513992	8.972741e+01	80.476589	108.137039	1.264320e+02	92.133623
ENSG00000269981	71.103953	7.551316e+01	98.016359	102.073280	7.914065e+01	74.992484
ENSG00000239906	0.000000	0.000000e+00	0.000000	0.000000	2.895390e+00	0.000000
ENSG00000241860	7.110395	5.330341e+00	5.158756	6.063759	2.895390e+00	3.213964
ENSG00000279457	235.658815	1.892271e+02	286.826818	211.220945	2.026773e+02	201.408385
ENSG00000228463	23.362727	2.576331e+01	15.476267	35.371929	3.957033e+01	20.355103
ENSG00000236679	0.000000	3.553561e+00	2.063502	2.021253	3.860520e+00	2.142642
ENSG00000237094	18.283874	1.599102e+01	25.793779	14.148771	9.651299e+00	11.784533
ENSG00000250575	1.015771	8.883902e-01	0.000000	1.010627	2.895390e+00	3.213964
ENSG00000230021	9.141937	3.553561e+00	10.317511	12.127518	1.158156e+01	7.499248

Deseq2内置了一个画图函数，可以方便地制定基因作图

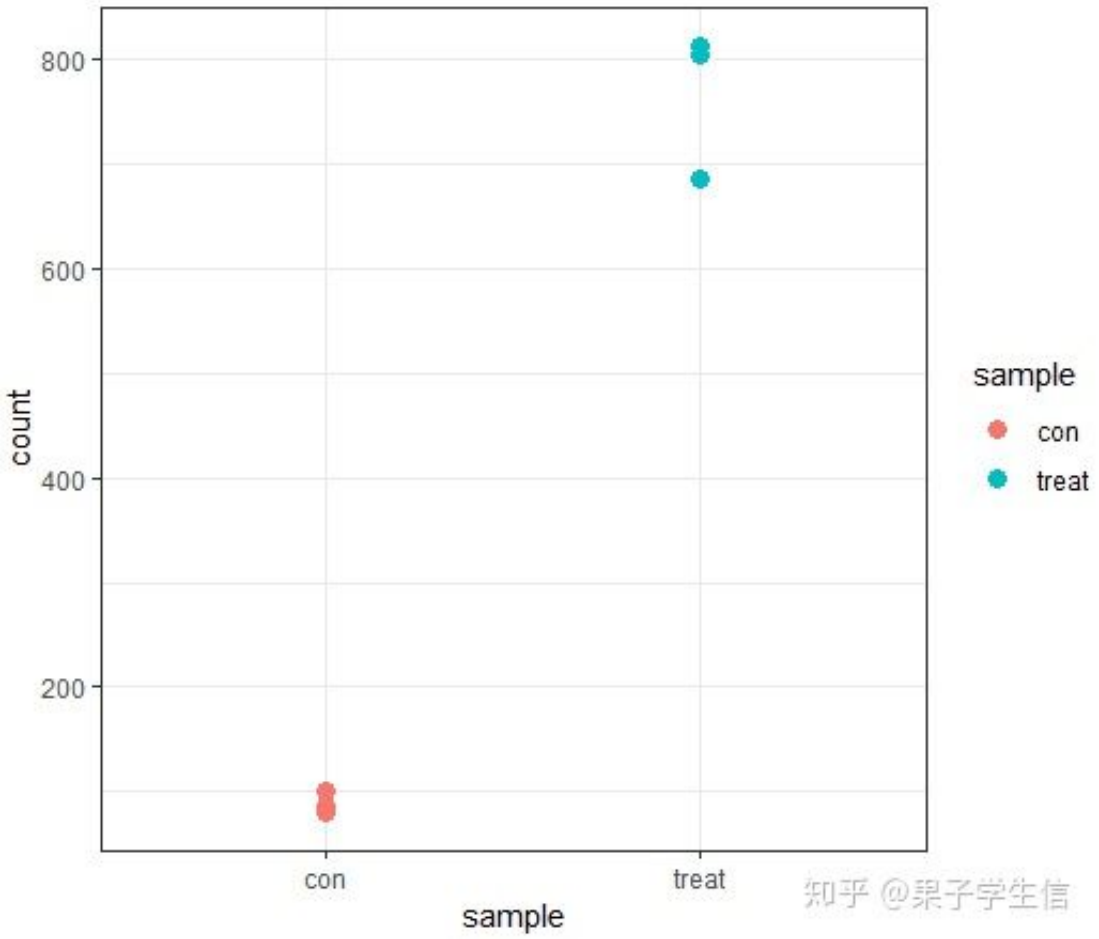
```
plotCounts(dds, gene = "ENSG00000172183",
intgroup=c("sample"))
```



这个功能本质上没有什么用，但是，可以提前确定实验的质量。比如，你的两组是敲减某个基因以及对照组，通过制定那个基因作图，就可以看出实验有没有成功，如果这个基因没有任何改变，也可以不用往下做了。回去重新做实验送样本吧。

我们说过，这种图自己看就可以，给老板看就算了，你得美化一下，那么他里面有个内置的参数returnData可以返回作图数据。 一旦返回数据，我们就可以用ggplot2自己简单画一下。

```
plotdata <- plotCounts(dds, gene = "ENSG00000172183",
intgroup=c("sample"),returnData = T)
library(ggplot2)
ggplot(plotdata,aes(x=sample,y=count,col=sample))+
  geom_point()+
  theme_bw()
```



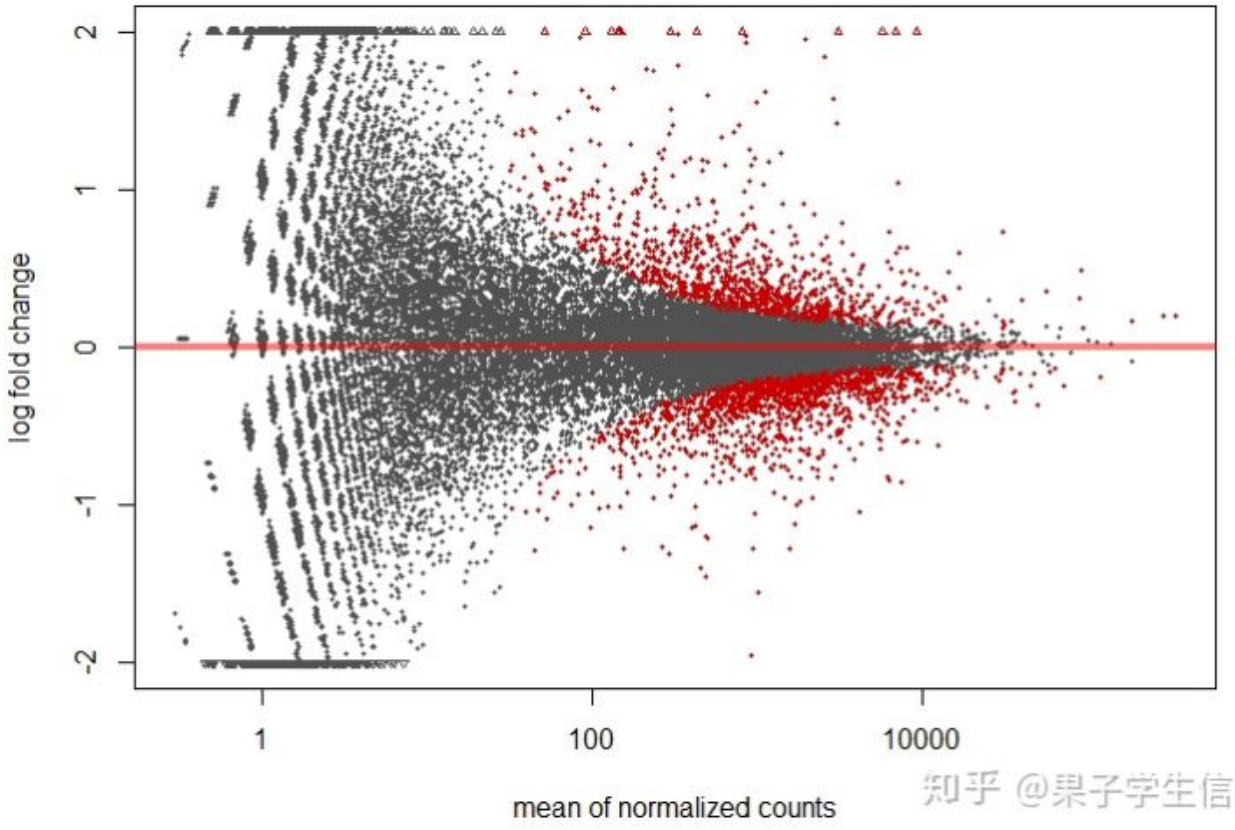
现在看起来平淡无奇，如果样本多，可以画出点图配boxplot，如果是配对样本，那么还可以画出配对的图。

有一点要强调一下，vst，以及log2的标准化，跟标准化的counts不是一个概念。前者是为了以后的聚类，热土，PCA分析，比如，我们计算样本间的距离就是用的vst 标化的数据，而标准化的counts是为了差异作图，你看纵坐标就会发现，他的数值一般很大。

LogFC的矫正

这一步，对于依赖logFC变化值的分析，很重要，比如GSEA分析。我们画一个MAplot图，看图说话

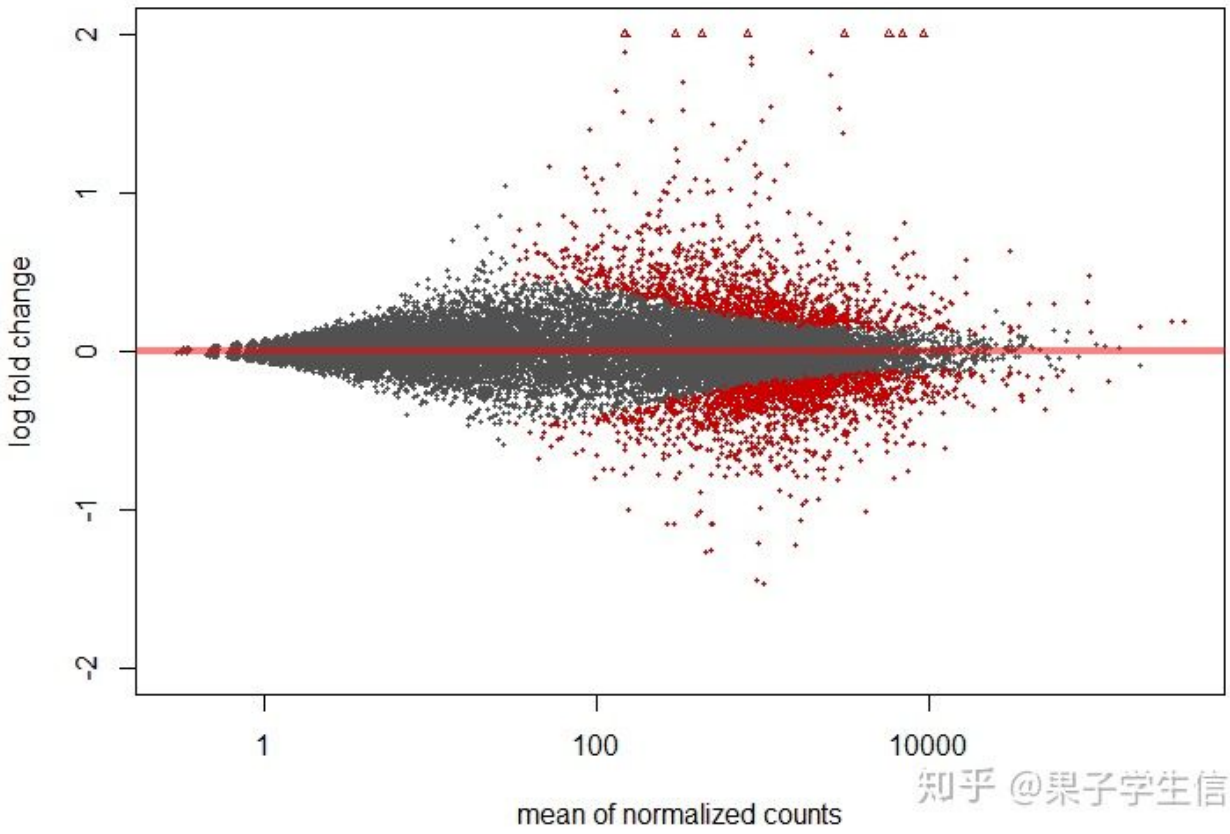
```
contrast <- c("sample", "treat", "con")
dd1 <- results(dds, contrast=contrast, alpha = 0.05)
plotMA(dd1, ylim=c(-2,2))
```



MA-plot上的点是每一个基因。横坐标是标准化后的counts平均值，纵坐标是log后的变化值。红色的点是p.adjust的值小于0.05的基因，由counts函数中的alpha 参数指定。 我们发现在左侧，有很多counts很小的基因，发生了很大的变化，但是没有明显意义。类似于从(1,3,9)变成了（20,12,3）他们的counts很小，波动性很大，对logFC产生了很大的影响。 GSEA分析中，排序就是按照logFC来进行的。按照这个结果往下做，GSEA那里，富集不到任何条目。

那就需要矫正。用的函数是lfcShrink，有很多参数，我们只演示一种

```
dd2 <- lfcShrink(dds, contrast=contrast, res=dd1)
plotMA(dd2, ylim=c(-2,2))
```



这样，原先那些波动性很大的基因，就被矫正了。而此时有LogFC以，红色的点为主。

差异分析的结果

这个结果实际上已经通过counts函数获得了，我们不在担心，处理组和对照组完全相反这种情况，因为他内置了参数设定比较组。比如，我们有5个处理组，我们不需要做5次Deseq，我们在results中指定即可。

用summary这个函数，可以看到差异分析的结果,高表达和低表达的比例。低丰度基因所占的比例。

```
summary(dd2, alpha = 0.05)
```

```
> summary(dd2, alpha = 0.05)

out of 27134 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 1117, 4.1%
LFC < 0 (down)     : 1447, 5.3%
outliers [1]       : 0, 0%
low counts [2]     : 14204, 52%
(mean count < 30)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```


再把差异分析的结果转化成data.frame的格式

```
library(dplyr)
library(tibble)
res <- dd2 %>%
  data.frame() %>%
  rownames_to_column("gene_id")
```

	gene_id	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
1	ENSG00000223972	1.682030e+00	0.0418285544	0.06371061	0.678634284	4.973696e-01	NA
2	ENSG00000227232	7.993319e+01	-0.2914586649	0.16183791	-1.799867089	7.188163e-02	2.184911e-01
3	ENSG00000278267	5.107629e+00	0.0411319400	0.10308582	0.402010108	6.876766e-01	NA
4	ENSG00000243485	3.165028e-01	0.0004472123	0.03053290	0.014644081	9.883161e-01	NA
5	ENSG00000233750	2.469601e+00	-0.0566082563	0.08116804	-0.707585057	4.792030e-01	NA
6	ENSG00000268903	9.907011e+01	0.2045165108	0.15536372	1.315877678	1.882151e-01	4.119336e-01
7	ENSG00000269981	8.347332e+01	0.0456854659	0.16147255	0.282961090	7.772067e-01	8.960573e-01
8	ENSG00000239906	4.825650e-01	0.0261311993	0.03265422	0.705321812	4.806100e-01	NA
9	ENSG00000241860	4.962101e+00	-0.0636301442	0.11142475	-0.573390162	5.663806e-01	NA
10	ENSG00000279457	2.211699e+02	-0.1678148358	0.13353175	-1.256795678	2.088276e-01	4.400514e-01
11	ENSG00000228463	2.664994e+01	0.2173537327	0.16651233	1.304662218	1.920079e-01	NA
12	ENSG00000236679	2.273580e+00	0.0277247394	0.07828775	0.353782444	7.235019e-01	NA
13	ENSG00000237094	1.594221e+01	-0.2206089891	0.15601739	-1.419014575	1.558948e-01	NA
14	ENSG00000250575	1.504023e+00	0.0665268705	0.06661083	1.045132942	2.959615e-01	NA
15	ENSG00000230021	9.036889e+00	0.0852025553	0.13358294	0.640264347	5.220008e-01	NA
16	ENSG00000225972	3.106551e+02	0.6208761986	0.11360031	5.460080634	4.759184e-08	1.098862e-06
17	ENSG00000225630	1.723167e+03	0.1903890950	0.08326910	2.286402065	2.223075e-02	9.209983e-02
18	ENSG00000237973	7.303610e+03	0.2879909264	0.07103752	4.054028154	5.034315e-05	5.732337e-04
19	ENSG00000229344	8.772170e+02	0.4925253806	0.10271264	4.794215951	1.633123e-06	2.789469e-05

基因ID转换

以前我们从gtf文件转换的, 但是我们需要gtf文件来提取mRNA以及lncRNA，就顺手做了ID转换，而且，mRNA和lncRNA是分别做的Deseq2，这从原理上来讲，是有问题的，Deseq2矫正了测序的深度，而这个深度应该是所有基因算在一起的深度，不应该分开来算。 TGCA数据的标准化以及差异分析 用两个包来转换，得到ENTREZID用于后续分析，得到SYMBOL便于识别。

```
library(AnnotationDbi)
library(org.Hs.eg.db)
res$symbol <- mapIds(org.Hs.eg.db,
                     keys=res$gene_id,
                     column="SYMBOL",
                     keytype="ENSEMBL",
                     multiVals="first")
res$entrez <- mapIds(org.Hs.eg.db,
                     keys=res$gene_id,
                     column="ENTREZID",
                     keytype="ENSEMBL",
                     multiVals="first")
```

	gene_id	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol	entrez
7726	ENSG00000152495	3.531936e+02	-0.1849301203	0.10579443	-1.747775564	8.050290e-02	2.365312e-01	CAMK4	814
21860	ENSG00000004660	2.164335e+02	0.1761553748	0.12286493	1.433620238	1.516807e-01	3.594015e-01	CAMKK1	84254
17938	ENSG00000110931	8.141376e+02	0.1176713421	0.07705188	1.527170314	1.267187e-01	3.188930e-01	CAMKK2	10645
3091	ENSG00000143919	2.051232e+02	-0.0102056100	0.12798523	-0.079739815	9.364442e-01	9.711257e-01	CAMKMT	79823
7884	ENSG00000164615	8.447183e+02	0.1633018793	0.08610210	1.896563729	5.788554e-02	1.853723e-01	CAMLG	819
4922	ENSG00000164047	8.295953e-01	-0.0102884687	0.04370333	-0.237217418	8.124881e-01	NA	CAMP	820
14050	ENSG00000130559	8.065614e+02	-0.2137812335	0.08276799	-2.582752791	9.801553e-03	4.955541e-02	CAMSAP1	157922
2220	ENSG00000118200	1.836396e+03	0.1156190815	0.06378079	1.812755340	6.986956e-02	2.136740e-01	CAMSAP2	23271
24768	ENSG00000076826	1.626780e+00	0.0212852517	0.05919618	0.361549524	7.176887e-01	NA	CAMSAP3	57662
161	ENSG00000171735	6.466203e+02	-0.0925653843	0.08724598	-1.060954559	2.887106e-01	5.357387e-01	CAMTA1	23261
21895	ENSG00000108509	1.291030e+03	-0.0180221317	0.08572015	-0.210245544	8.334760e-01	9.251041e-01	CAMTA2	23125
17486	ENSG00000111530	3.466761e+03	0.0585510219	0.05501134	1.064343847	2.871730e-01	5.341120e-01	CAND1	55832
4632	ENSG00000144712	8.801969e+00	-0.1748394336	0.12567465	-1.419633484	1.557144e-01	NA	CAND2	23066
23236	ENSG00000171302	2.093754e+03	-0.2477611093	0.06033752	-4.106153189	4.023027e-05	4.728886e-04	CANT1	124583
8333	ENSG00000127022	3.320049e+04	-0.0295563019	0.05227227	-0.565429500	5.717817e-01	7.682778e-01	CANX	821
689	ENSG00000131236	3.763831e+03	-0.0943861817	0.05346865	-1.765249315	7.752187e-02	2.298986e-01	CAP1	10487
8520	ENSG00000112186	1.478337e+03	-0.0116969817	0.06247635	-0.187221713	8.514868e-01	9.[illegible]	CAP2	[illegible]
3392	ENSG00000042493	4.983678e+03	0.3200749172	0.05272519	6.070536471	1.274837e-09	3.806845e-08	CAPG	822

有了这个文件，里面有logFC，p值，还有基因名称，我们可以完成GO，KEGG，热图，火山图所有操作。这一部分内容参考 最有诚意的GEO数据库教程 这个帖子目前已经有接近4000次访问，这在我们这样一个小号是不容易的，靠的是真诚。

制作geneList

我们在那个帖子里面并没有讲GSEA分析，今天来展示一下。原理略过。我们这里还是用Y叔的神包clusterprofier，神包虽好，记得引用。

使用这个包做GSEA，要制作一个genelist，这个是一个向量，他的内容是排序后的logFC值，他的名称是ENTREZID，而这两个我们都是不缺的，在上一步得到的差异结果中。

```
library(dplyr)
gene_df <- res %>%
  dplyr::select(gene_id,log2FoldChange,symbol,entrez) %>%
  ## 去掉NA
  filter(entrez!="NA") %>%
  ## 去掉重复
  distinct(entrez,.keep_all = T)
```

	gene_id	log2FoldChange	symbol	entrez
1	ENSG00000223972	0.0418285544	DDX11L1	100287102
2	ENSG00000227232	-0.2914586649	WASH7P	653635
3	ENSG00000278267	0.0411319400	MIR6859-1	102466751
4	ENSG00000241860	-0.0636301442	LOC100996442	100996442
5	ENSG00000279457	-0.1678148358	WASH9P	102723897
6	ENSG00000230021	0.0852025553	LOC101928626	101928626
7	ENSG00000177757	0.0250741060	FAM87B	400728
8	ENSG00000228794	-0.0380230382	LINC01128	643837
9	ENSG00000225880	-0.0451629856	LINC00115	79854
10	ENSG00000230368	-0.0132923478	FAM41C	284593
11	ENSG00000223764	0.0593738730	LINC02593	100130417
12	ENSG00000187634	0.1803148251	SAMD11	148398
13	ENSG00000188976	-0.1356959902	NOC2L	26155

制作genelist的三部曲

```
## 1.获取基因logFC
geneList <- gene_df$log2FoldChange
## 2.命名
names(geneList) = gene_df$entrez
## 3.排序很重要
geneList = sort(geneList, decreasing = TRUE)
```

看一下这个genelist，增加感性的理解

```
head(geneList)
```

```
> head(geneList)
    8638     9636     3434     3433     3437     3669
3.207248 3.078077 2.910439 2.838995 2.765290 2.681219
```

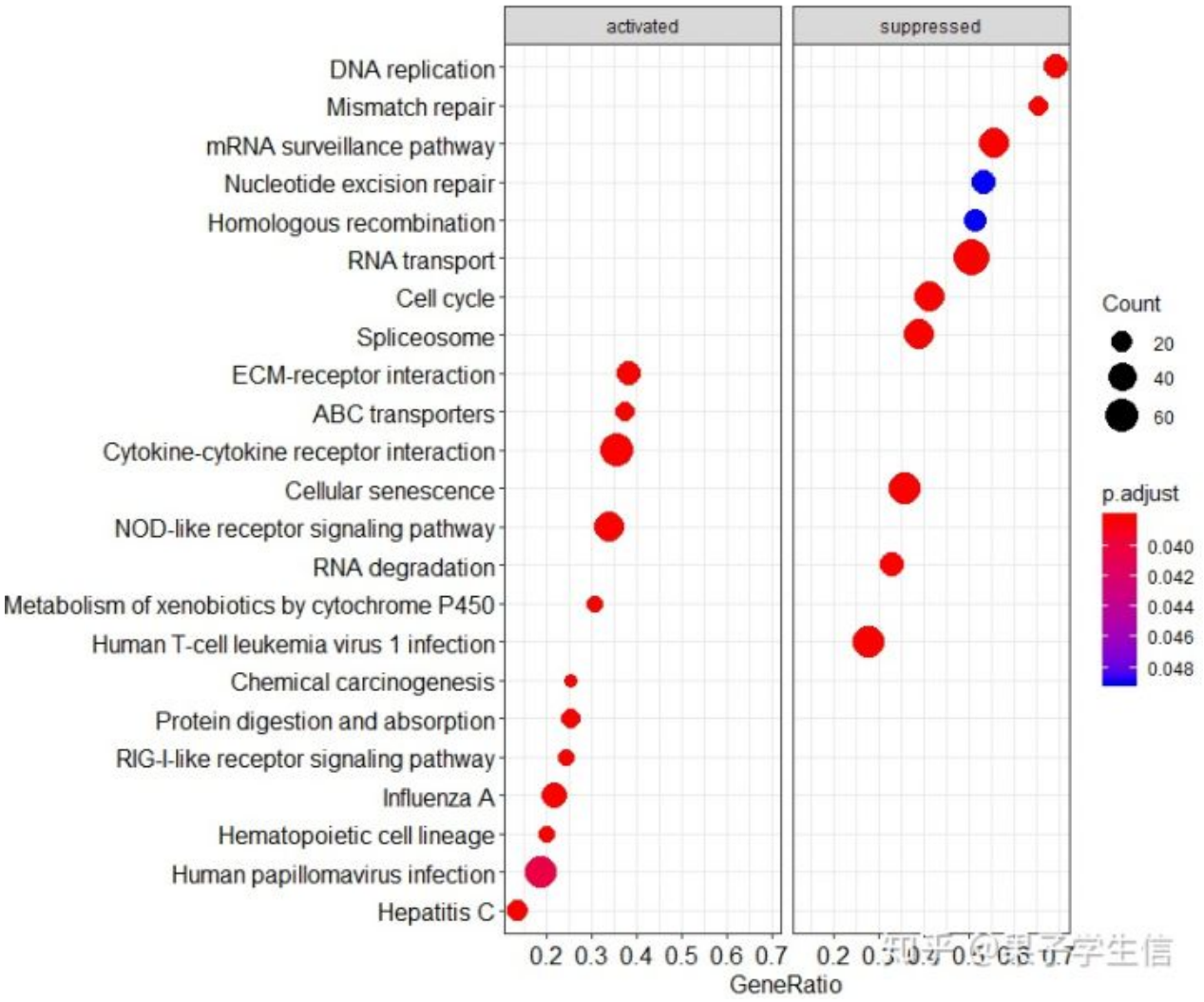
GSEA分析

完成KEGG的GSEA分析

```
library(clusterProfiler)
## 没有富集到任何数据
gseaKEGG <- gseKEGG(geneList     = geneList,
                    organism     = 'hsa',
                    nPerm        = 1000,
                    minGSSize    = 20,
                    pvalueCutoff = 0.05,
                    verbose      = FALSE)
```

作图展示富集分布图

```
library(ggplot2)
dotplot(gseaKEGG,showCategory=12,split=".sign")+facet_grid(
~.sign)
```



这时候，我们看到有一些通路是被激活的，有一些通路是被抑制的。比如Cell cycle是被抑制的，我们可以选取单个通路来作图。

把富集的结果转换成data.frame,找到Cell cycle的通路ID是hsa04110

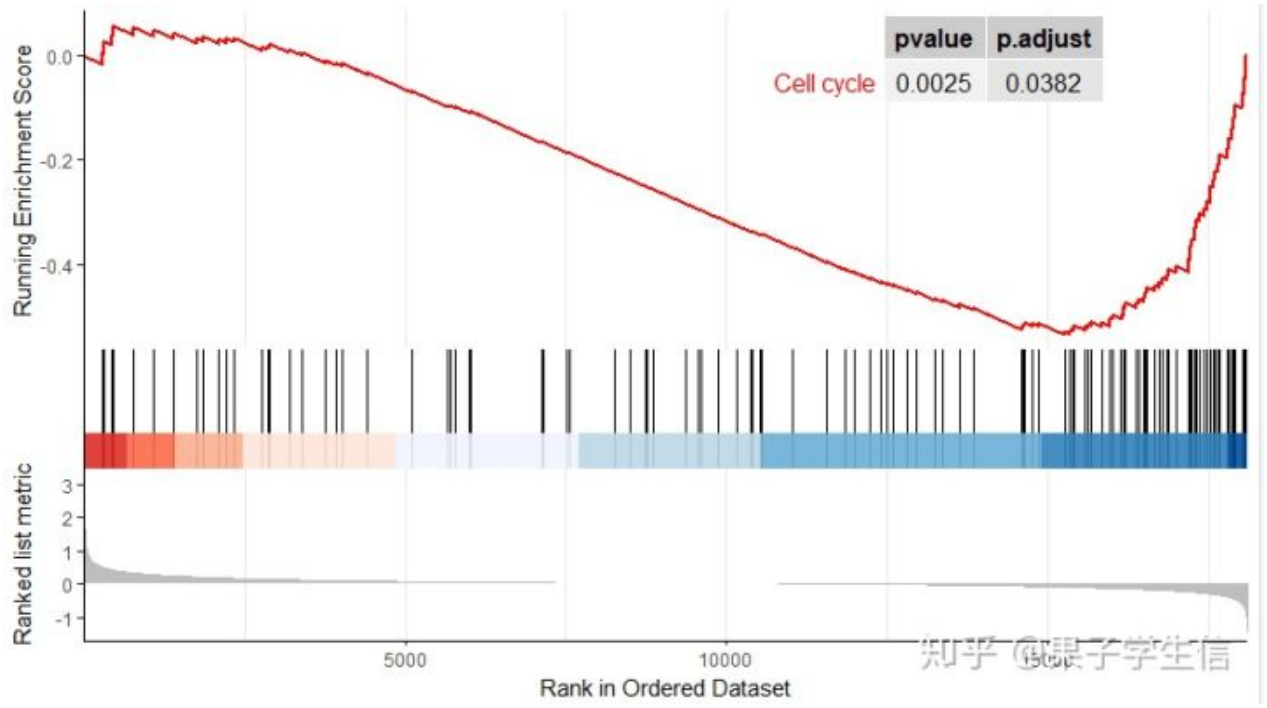
```
gseaKEGG_results <- gseaKEGG@result
```

Filter

	ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalu
hsa05160	hsa05160	Hepatitis C	131	0.4660235	1.679999	0.001620746	0.03817204	0.030
hsa05164	hsa05164	Influenza A	141	0.4870129	1.771464	0.001631321	0.03817204	0.030
hsa04060	hsa04060	Cytokine-cytokine receptor interaction	173	0.5236478	1.939748	0.001636661	0.03817204	0.030
hsa04621	hsa04621	NOD-like receptor signaling pathway	140	0.4377442	1.586410	0.001647446	0.03817204	0.030
hsa04512	hsa04512	ECM-receptor interaction	73	0.6139433	2.033929	0.001712329	0.03817204	0.030
hsa04622	hsa04622	RIG-I-like receptor signaling pathway	53	0.5839897	1.831332	0.001748252	0.03817204	0.030
hsa05204	hsa05204	Chemical carcinogenesis	47	0.5856296	1.808573	0.001751313	0.03817204	0.030
hsa04974	hsa04974	Protein digestion and absorption	63	0.6257804	2.022398	0.001757469	0.03817204	0.030
hsa00980	hsa00980	Metabolism of xenobiotics by cytochrome P450	42	0.6553960	1.983997	0.001763668	0.03817204	0.030
hsa04640	hsa04640	Hematopoietic cell lineage	64	0.5883498	1.900435	0.001763668	0.03817204	0.030
hsa02010	hsa02010	ABC transporters	40	0.6529914	1.939234	0.001838235	0.03817204	0.030
hsa03430	hsa03430	Mismatch repair	23	-0.7329885	-2.037972	0.002127660	0.03817204	0.030
hsa03030	hsa03030	DNA replication	36	-0.7568684	-2.306723	0.002212389	0.03817204	0.030
hsa03018	hsa03018	RNA degradation	76	-0.4920448	-1.758150	0.002439024	0.03817204	0.030
hsa03015	hsa03015	mRNA surveillance pathway	81	-0.5628821	-2.042957	0.002450980	0.03817204	0.030
hsa04110	hsa04110	Cell cycle	121	-0.5368014	-2.086810	0.002500000	0.03817204	0.030
hsa03013	hsa03013	RNA transport	154	-0.4612240	-1.837450	0.002583979	0.03817204	0.030
hsa04218	hsa04218	Cellular senescence	146	-0.4215640	-1.659832	0.002583979	0.03817204	0.030

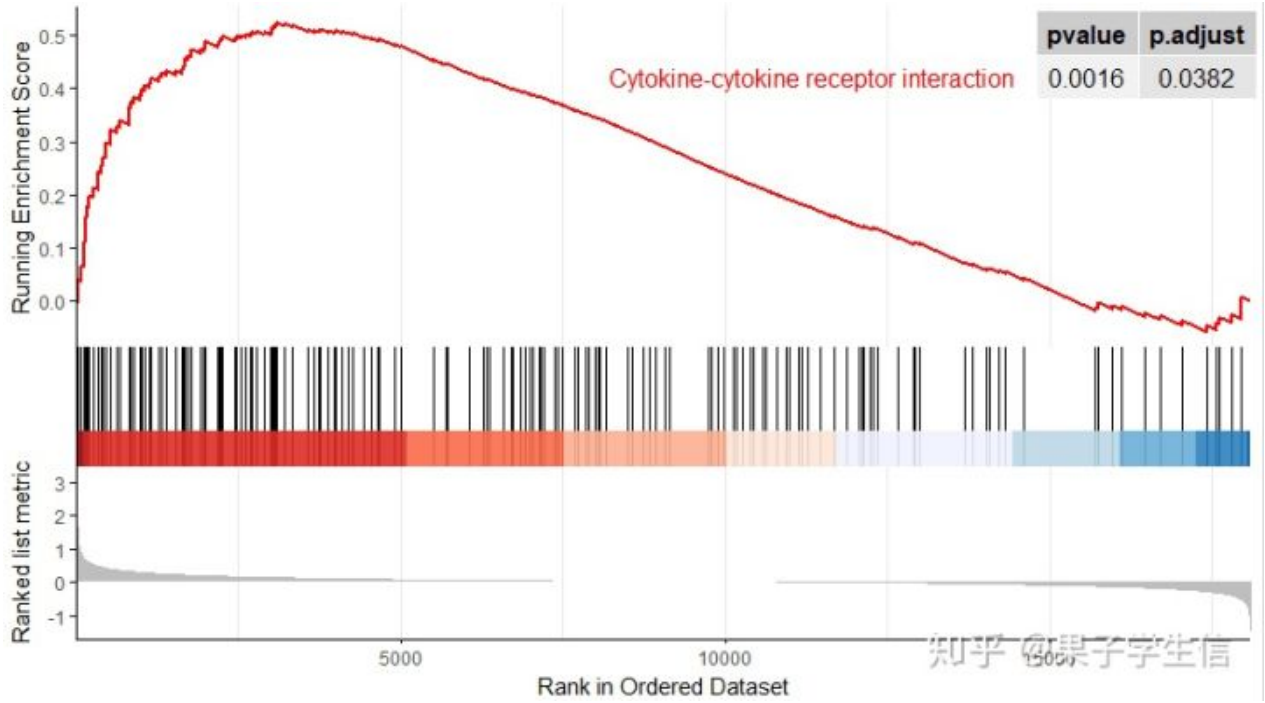
使用gseaplot2把他画出来

```
library(enrichplot)
pathway.id = "hsa04110"
gseaplot2(gseaKEGG,
          color = "red",
          geneSetID = pathway.id,
          pvalue_table = T)
```



也可以画出一个激活的

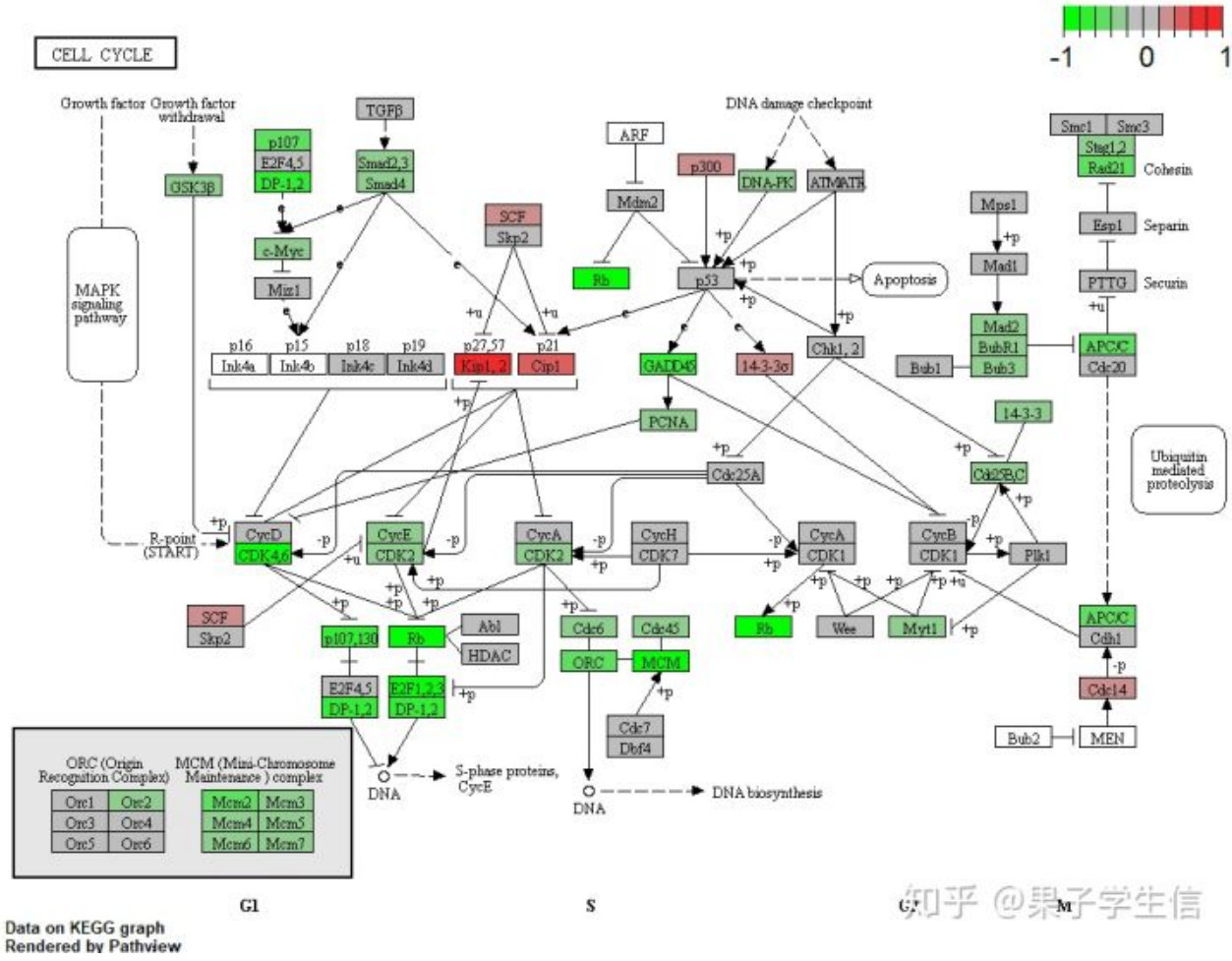
```
pathway.id = "hsa04060"
gseaplot2(gseaKEGG,
          color = "red",
          geneSetID = pathway.id,
          pvalue_table = T)
```



pathview 展示

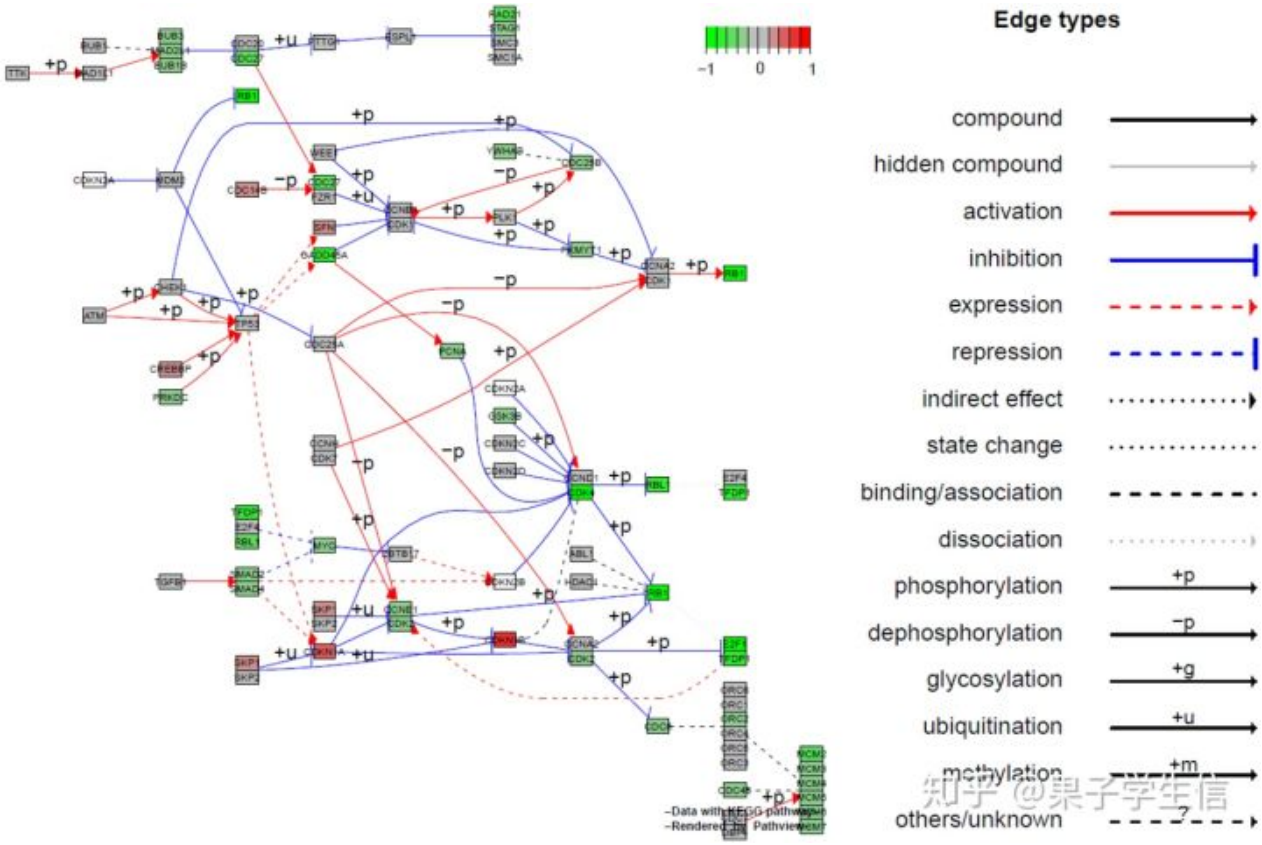
我们现在知道cell cycle是被抑制的，如果还想看一下这个通路里面的基因是如何变化的，应该怎么办呢，pathview 可以帮到我们。

```
library(pathview)
pathway.id = "hsa04110"
pv.out <- pathview(gene.data  = geneList,
                   pathway.id = pathway.id,
                   species    = "hsa")
```



改变一下参数，可以得到另外一种构图

```
library(pathview)
pathway.id = "hsa04110"
pv.out <- pathview(gene.data  = geneList,
                   pathway.id = pathway.id,
                   species    = "hsa",
                   kegg.native = F)
```



一眼看过去，都是绿的，说明这个通路确实是被抑制了，还可以在图上缕一缕，哪些是核心分子，一般说来，越往上游越核心。

总结

写到这里，GEO的分析，TCGA的基本分析，RNA-seq的基本分析都写完了。这几个帖子可以把大部分的培训班给搞定。里面的内容，只要会一点R语言，就可以重复。说到底，R语言的培训，最应该培训的是基本技能。

但是你有没有发现，虽然图做的这么好看，我们总觉得还欠缺了些什么。这也是目前生物信息分析和实验结合的痛点。

> 我如果开题，你这一通分析，还是没有让我确定要研究的核心基因。

而这个，是一篇帖子，或者普通培训班不能给予的。

> 如果未来的生信培训要出点什么彩，这个一定是个方向。

生信技术是通用的，优点就是可被重复，可以被写成教程，但是挖掘能做实验可发文章的核心基因，目前并没有系统教程，类似于玄学。这需要我们长年累月地积累，才能有点感觉，而我觉得这是科研人员做生信分析的核心技能。在这方面，我也是个初学者。

> 否则，我们就是个能做实验会做分析的技术人员，谈不上一个独立的科研人。