GEO数据库

简介：GEO数据库储存了大量的高通量基因表达数据，本文以鳞状肺癌细胞的基因表达图谱鉴定差异调控的基因为例，通过R语言来深入实现GEO数据库中转录组数据的挖掘和分析。这是本人首次做类似的分析，如果步骤和分析上有不足之处，还请谅解。

老师：赵虎

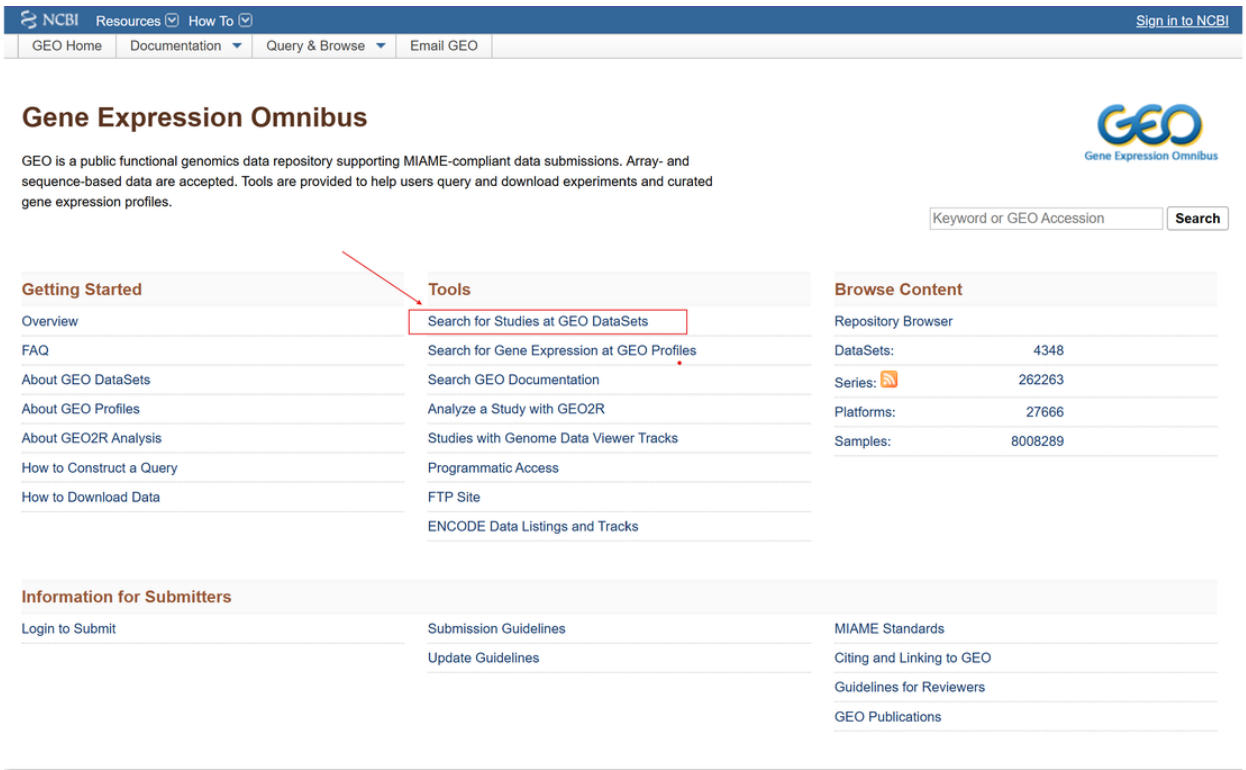
课题组：油菜机器智能

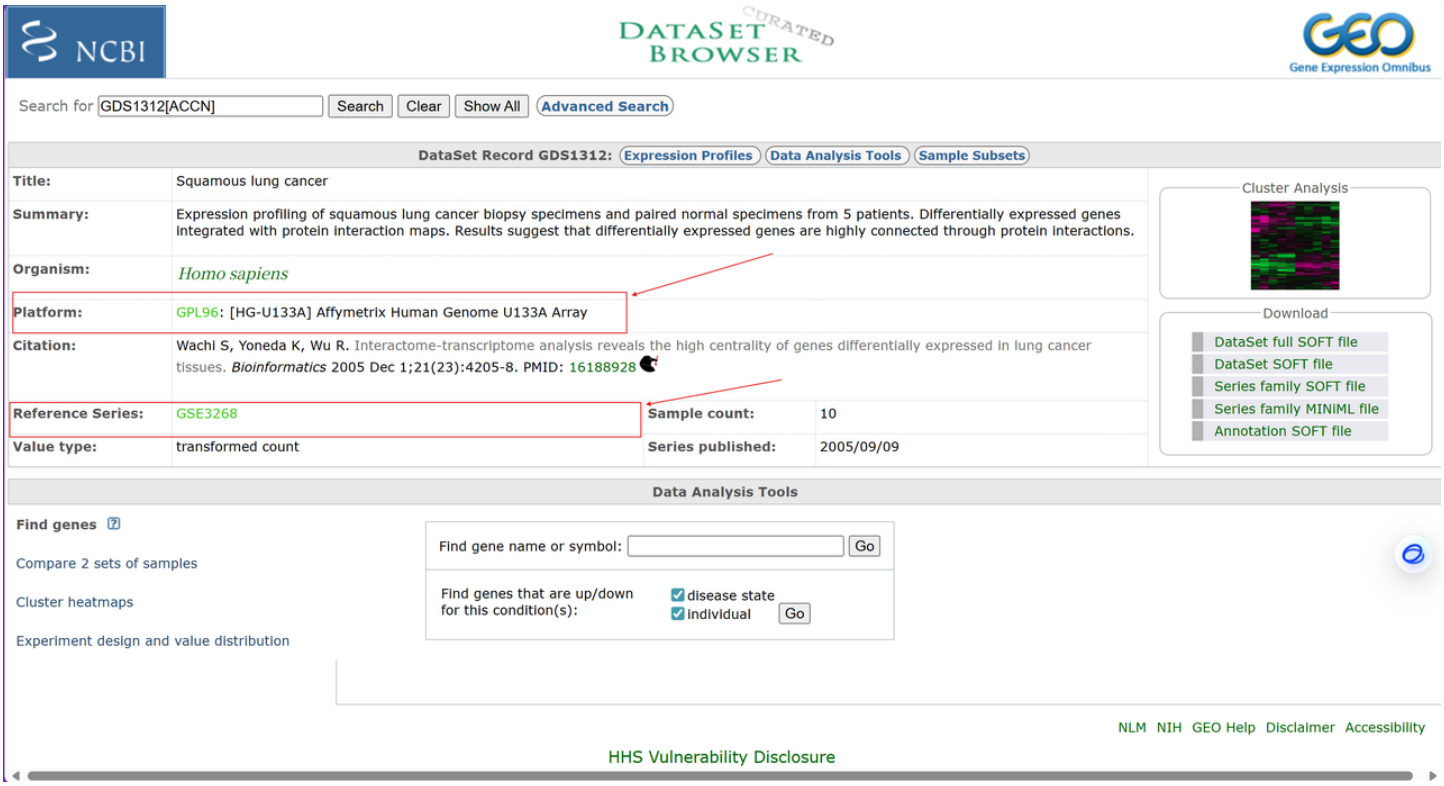
研究方向：调控元件的挖掘、GWAS

整理：魏雨城 华中农业大学 @不想下地的华农er

一、数据获取与预处理

1. 手动下载（如图）





> GPL96为探针与基因对应的注释库
>
> GSE3268为我们要的转录组数据
>
> 处理这两个数据需要用到perl语言，比较麻烦，我们主要还是用过R语言来实现数据的获取

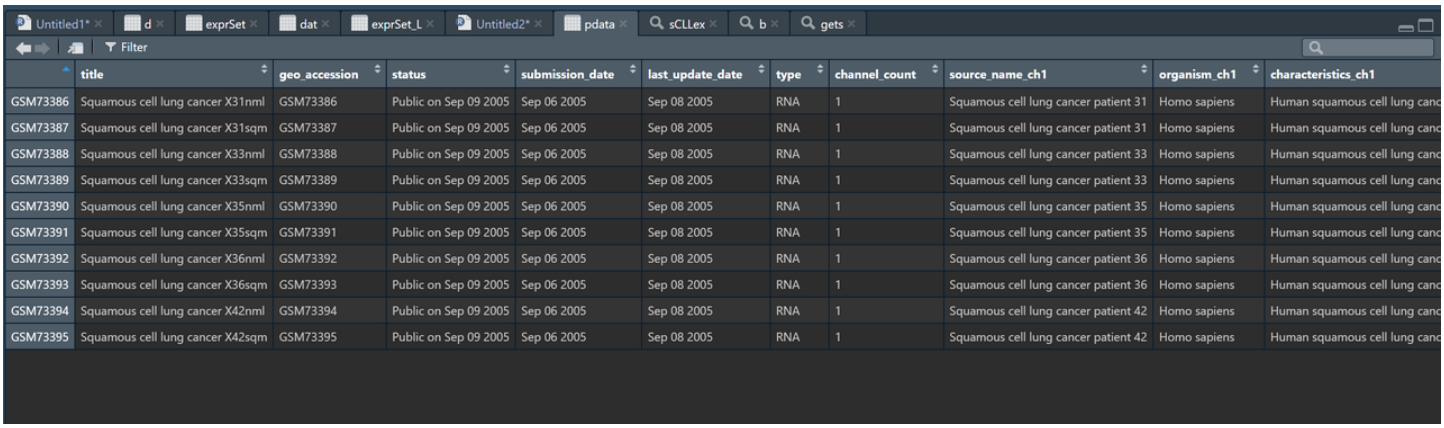
2. 利用 `GEOquery` 包来下载

代码块

```
1
2   # 第一部分：环境初始化与数据加载
3
4   install.packages("BiocManager")
5   BiocManager::install("GEOquery")
6   setwd('D:/计算生物学/GEO数据库')
7   if(F){
8     library(GEOquery)
9     #获取“GSE3268”号数据，运行此命令后在工作路径会有“GSE3268_series_matrix.txt.gz”新文
    件
10    gset <- getGEO('GSE3268',destdir = '.',AnnotGPL = F,getGPL = F)
11    save(gset,file = 'GSE3268.gset.Rdata')
12    #查看数据类型与结构
13  }
14  #加载数据
15  load('GSE3268.gset.Rdata')
16
17  #查看数据结构
18  class(gset)
19  str(gset)
```

```
20
21  b <- gset[[1]]  #提取信息
22  exprSet <- exprs(b)   # 提取表达矩阵
23
24  samples=sampleNames(b)  # 获取样本名称
25  ' [1] "GSM73386" "GSM73387" "GSM73388" "GSM73389" "GSM73390" "GSM73391"
    "GSM73392"
26   [8] "GSM73393" "GSM73394" "GSM73395"'
27
28  pdata=pData(b)  # 获取样本表型数据
29  group_list=as.character(pdata[,'description'])  # 提取分组信息（细胞类型）
30  ' [1] "Normal cells" "Tumor cells"  "Normal cells" "Tumor cells"  "Normal
    cells"
31   [6] "Tumor cells"  "Normal cells" "Tumor cells"  "Normal cells" "Tumor cells"
    '
32
33  # BiocManager::install('hgu133a2.db')
34  BiocManager::install("hgu133a2.db")  # 安装注释数据库（注释状态）
35  library(hgu133a2.db)         # 加载芯片注释包
36  ls("package:hgu133a2.db")    # 列出包中所有对象
37
38  ids <- toTable(hgu133a2SYMBOL)  #获取探针ID与基因符号的映射表
39  save(ids,exprSet,pdata,file = 'input.Rdata')  # 保存数据到文件
40  length(unique(ids$symbol))       #计算唯一基因符号的数量
41  #[1] 12643 ，探针数量有20201个，说明存在多个探针对应一个基因的情况
42
43  tail(sort(table(ids$symbol)))    #查看最多被注释的基因符号
44  #PDLIM5  FGFR2 PTGER3  CFLAR   TCF3    HFE
45  #     9     10     10     11     11     13
46  #HFE基因被注释的次数最多
47
48  table(sort(table(ids$symbol)))  # 统计各基因对应的探针数量分布
49  #   1    2    3    4    5    6    7    8    9   10   11   13
50  #8063 2708 1182  445  148   62   16    9    5    2    2    1
51  #有8063个基因对应一个探针，1个基因对应13个探针
52
53  plot(table(sort(table(ids$symbol)))) #将各基因对应的探针数量利用图表呈现
54
```

> `toTable()` 是 Bioconductor 注释包中的核心函数，主要功能是将复杂的注释对象转换为数据框格式，使其更易于操作和分析。
>
> `pData()` 是 Bioconductor 中用于处理 `ExpressionSet` 对象的函数，主要功能是提取样本的表型数据。



	title	geo_accession	status	submission_date	last_update_date	type	channel_count	source_name_ch1	organism_ch1	characteristics_ch1
GSM73386	Squamous cell lung cancer X31nml	GSM73386	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 31	Homo sapiens	Human squamous cell lung canc
GSM73387	Squamous cell lung cancer X31sqm	GSM73387	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 31	Homo sapiens	Human squamous cell lung canc
GSM73388	Squamous cell lung cancer X33nml	GSM73388	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 33	Homo sapiens	Human squamous cell lung canc
GSM73389	Squamous cell lung cancer X33sqm	GSM73389	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 33	Homo sapiens	Human squamous cell lung canc
GSM73390	Squamous cell lung cancer X35nml	GSM73390	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 35	Homo sapiens	Human squamous cell lung canc
GSM73391	Squamous cell lung cancer X35sqm	GSM73391	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 35	Homo sapiens	Human squamous cell lung canc
GSM73392	Squamous cell lung cancer X36nml	GSM73392	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 36	Homo sapiens	Human squamous cell lung canc
GSM73393	Squamous cell lung cancer X36sqm	GSM73393	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 36	Homo sapiens	Human squamous cell lung canc
GSM73394	Squamous cell lung cancer X42nml	GSM73394	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 42	Homo sapiens	Human squamous cell lung canc
GSM73395	Squamous cell lung cancer X42sqm	GSM73395	Public on Sep 09 2005	Sep 06 2005	Sep 08 2005	RNA	1	Squamous cell lung cancer patient 42	Homo sapiens	Human squamous cell lung canc

> pdata 样本表型数据示例

	GSM73386	GSM73387	GSM73388	GSM73389	GSM73390	GSM73391	GSM73392	GSM73393	GSM73394	GSM73395
1007_s_at	9.185	10.529	9.618	10.343	9.369	10.484	9.610	10.934	9.372	11.332
1053_at	6.282	6.803	6.389	6.717	6.402	6.616	6.395	6.600	6.340	7.067
117_at	6.508	6.514	6.480	6.427	6.587	6.557	6.658	6.486	6.799	6.436
121_at	8.945	8.898	9.004	9.017	9.145	9.017	9.032	8.791	8.719	8.725
1255_g_at	3.974	4.007	4.142	4.157	4.296	4.294	4.043	4.068	4.043	4.082
1294_at	8.272	7.508	8.542	7.994	8.342	7.674	8.213	7.267	8.649	7.667
1316_at	5.734	5.851	5.887	5.928	5.955	5.883	5.744	5.783	5.764	5.816
1320_at	5.328	4.861	5.157	4.877	4.964	4.944	4.934	4.844	4.908	4.780
1405_i_at	7.658	6.920	7.678	6.976	6.891	6.862	6.977	5.643	7.416	5.233
1431_at	4.059	4.133	3.981	4.066	4.000	4.041	3.974	3.966	4.060	3.945
1438_at	7.362	8.170	7.292	8.077	7.214	7.715	7.139	8.358	6.981	8.930
1487_at	7.799	7.723	7.855	7.784	7.805	7.772	7.722	7.448	7.696	7.771
1494_f_at	6.970	6.783	7.206	7.065	7.146	7.082	7.013	7.095	6.977	6.847
1598_g_at	10.951	10.202	10.778	10.353	10.644	10.047	10.778	9.548	11.145	9.637
160020_at	8.509	8.771	8.560	9.340	8.730	9.603	8.535	9.347	8.466	8.734
1729_at	8.010	8.153	8.043	7.907	7.929	8.125	7.997	7.838	8.218	7.643
1773_at	6.223	6.618	6.158	6.192	6.112	6.227	6.165	5.918	5.845	6.160
177_at	5.606	6.210	5.564	5.772	5.602	5.624	5.606	5.594	5.269	5.529
179_at	9.670	9.491	9.702	9.628	9.693	9.626	9.620	9.459	9.448	9.396
1861_at	6.646	6.624	6.519	6.606	6.267	6.811	6.606	6.703	6.288	6.765
200000_s_at	9.378	9.370	9.479	9.550	9.381	9.146	9.764	9.179	9.466	9.645
200001_at	11.032	10.849	11.052	10.729	10.942	11.113	11.125	10.456	10.855	10.372
200002_at	11.639	12.175	11.314	11.417	11.269	11.539	11.220	11.735	11.498	11.999
200003_s_at	12.390	12.369	12.398	12.946	12.556	12.884	12.442	13.185	12.342	12.411

基因探针

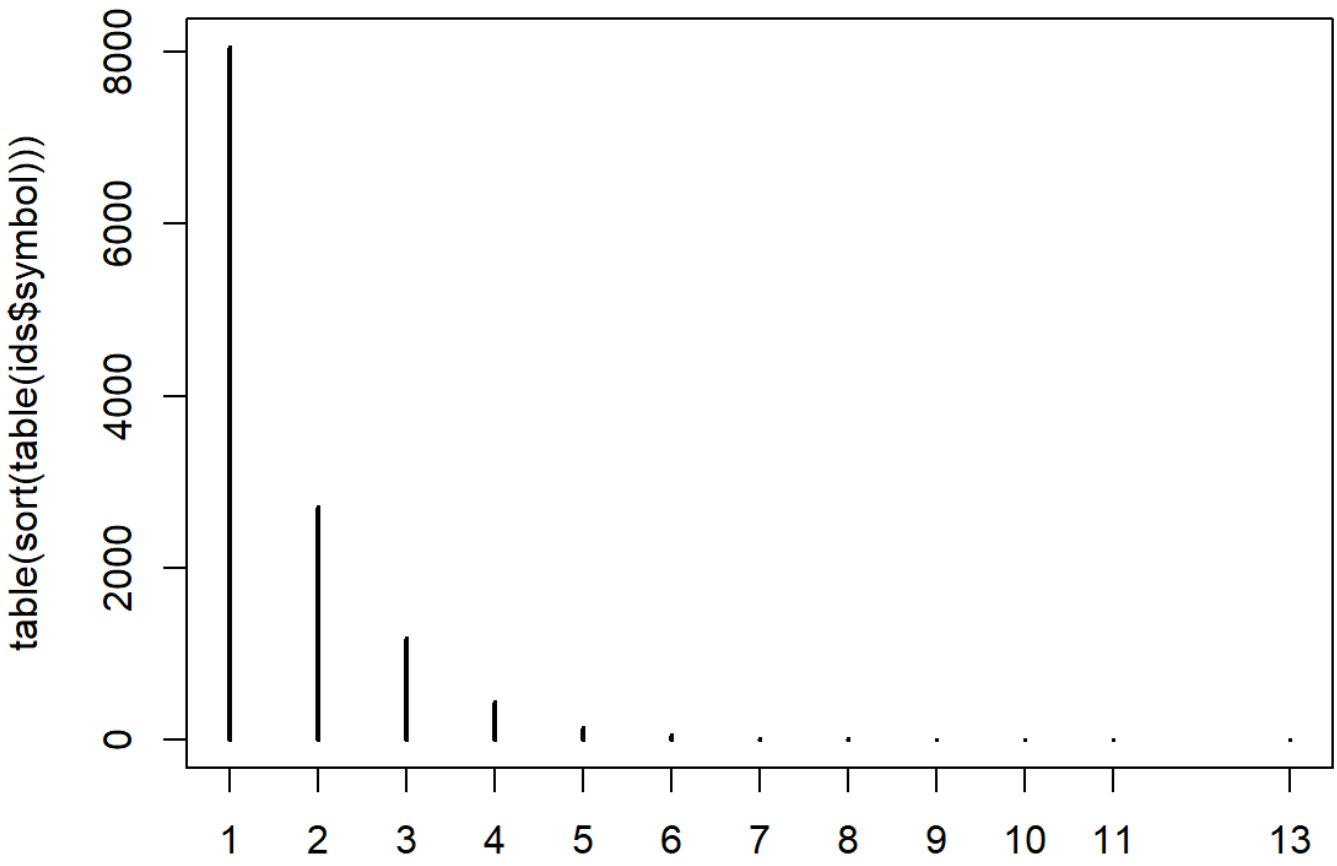
细胞编号

细胞表达量

> exprSet部分数据

	probe_id	symbol
1	1007_s_at	DDR1
2	1053_at	RFC2
3	117_at	HSPA6
4	121_at	PAX8
5	1255_g_at	GUCA1A
6	1294_at	UBA7
7	1316_at	THRA
8	1320_at	PTPN21
9	1405_i_at	CCL5
10	1431_at	CYP2E1
11	1438_at	EPHB3
12	1487_at	ESRRA
13	1494_f_at	CYP2A6
14	1598_g_at	GAS6

> ids的部分数据，第一列为探针信息，第二列为基因信息



> 基因探针图，存在不少重复的探针点位

二、探针过滤与基因注释整合

代码块

```r
# 第二部分：探针过滤与注释整合
table(rownames(exprSet) %in% ids$probe_id)  # 统计有注释的探针数量
 #FALSE  TRUE
 # 2082 20201  ；说明表达矩阵中2082个探针在"hgu133a2.db"注释库中没有注释，将其删去

dim(exprSet)  # 显示过滤前维度
#[1] 22283行    10列

exprSet=exprSet[rownames(exprSet) %in% ids$probe_id,]  # 保留有注释的探针
dim(exprSet)  # 显示过滤后维度
#[1] 20201行    10列 ；  相比过滤前少了2072行，说明没有注释信息的探针已被删去

ids=ids[match(rownames(exprSet),ids$probe_id),]  # 按表达矩阵顺序排列注释
head(ids)  # 查看注释表前几行
#   probe_id symbol
#1 1007_s_at   DDR1
#2   1053_at   RFC2
#3    117_at  HSPA6
#4    121_at   PAX8
#5 1255_g_at GUCA1A

exprSet[1:5,1:5]  # 再次查看表达矩阵
'''          GSM73386 GSM73387 GSM73388 GSM73389 GSM73390
1007_s_at    9.185   10.529    9.618   10.343    9.369
1053_at      6.282    6.803    6.389    6.717    6.402
117_at       6.508    6.514    6.480    6.427    6.587
121_at       8.945    8.898    9.004    9.017    9.145
1255_g_at    3.974    4.007    4.142    4.157    4.296 '''

```

> `match(x, table)` 函数：
>
> 1. 在 `table` 中查找 `x` 的每个元素
> 2. 返回 `x` 中每个元素在 `table` 中的位置索引
> 3. 如果元素不存在，返回 `NA`
>
> 通过match和索引操作，ids中的探针顺序和数量将和表达矩阵相同，这样我们才可以将基因与探针一一对应，进行相应的转换

```
> match(rownames(exprSet),ids$probe_id)
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
 [17]  NA  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
 [33]  NA  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46
 [49]  47  48  49  50  51  52  NA  53  54  55  NA  56  57  NA  58  59
 [65]  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75
 [81]  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91
 [97]  92  93  94  95  NA  96  97  98  99 100 101 102  NA 103 104 105
[113] 106 107 108  NA 109 110 111 112 113 114 115 116 117 118 119 120
[129] 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
[145] 137 138 139 140 141  NA  NA 142 143 144 145 146 147 148 149 150
[161] 151 152 153 154 155 156 157 158 159 160 161 162 163 164  NA 165
[177] 166 167 168 169  NA 170  NA 171 172 173 174 175 176 177 178 179
[193] 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195
```

三、基因水平表达矩阵构建

代码块

```
# 第三部分：基因水平表达矩阵构建（主用方法）
identical(ids$probe_id,rownames(exprSet))  # 检查探针ID是否一致
dat=exprSet  # 创建副本
dim(dat)
'[1] 20201    10'

ids$median=apply(dat,1,median) # 计算每个探针的中位数表达值
ids=ids[order(ids$symbol,ids$median,decreasing = T),] # 按基因符号和中位数降序排列
ids=ids[!duplicated(ids$symbol),] # 去除重复基因,由于上方已经降序排列，去除重复后保留最高表达探针
#print(!duplicated(ids$symbol))
'[1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
  [11]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
  [21]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
  [31]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
  [41]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE'


dat=dat[ids$probe_id,]  # 重构表达矩阵
rownames(dat)=ids$symbol # 行名设为基因符号
dat[1:4,1:4]  # 查看处理后的矩阵
'''
       GSM73386 GSM73387 GSM73388 GSM73389
ZZZ3      6.812    6.199    5.896    6.210
ZZEF1     7.552    7.060    7.488    7.234
ZYX       9.970    9.273    9.635    9.343
ZXDC      6.796    7.283    7.032    7.152
'''
```

```
28
29  dim(dat)  # 显示最终维度
30  '[1] 12643    10' #去除7558个重复基因
```

`identical(x, y)` 是R语言中的严格相等性检查函数，用于判断两个对象是否完全一致。

1. 内容检查：比较两个向量的每个元素是否完全相同
2. 顺序检查：验证元素的顺序是否一致
3. 类型检查：确认数据类型是否相同
4. 属性检查：比较名称、维度等属性

`duplicated(x)` 识别向量或数据框中的重复元素，返回一个逻辑向量指示哪些元素是重复的。

1. 首次出现：每个元素的第一次出现不被视为重复
2. 后续出现：相同元素的后续出现被标记为重复
3. 返回值：逻辑向量（TRUE = 重复，FALSE = 不重复）

	probe_id	symbol	median
11702	212893_at	ZZZ3	6.2395
11418	212601_at	ZZEF1	7.2660
14142	215706_x_at	ZYX	9.4840
16340	218639_s_at	ZXDC	7.1325
14360	216013_at	ZXDB	5.3910
3468	204026_s_at	ZWINT	7.0615
16056	218349_s_at	ZWILCH	5.1590
4237	204812_at	ZW10	7.3920
11186	212359_s_at	ZSWIM8	8.2510
15354	217592_at	ZSWIM1	6.0215
4591	205181_at	ZSCAN9	6.3455
17577	219904_at	ZSCAN5A	5.7805
17949	220292_at	ZSCAN32	7.0360
19466	222016_s_at	ZSCAN31	6.1230
12007	213218_at	ZSCAN26	6.2355
18158	220516_at	ZSCAN2	5.6650
16020	218312_s_at	ZSCAN18	7.8895

Ids 部分数据，从左往右依次为探针名称、与之对应的基因名和最高表达量

Filter

	GSM73386	GSM73387	GSM73388	GSM73389	GSM73390	GSM73391	GSM73392	GSM73393
ZZZ3	6.812	6.199	5.896	6.210	5.995	6.360	5.913	6.723
ZZEF1	7.552	7.060	7.488	7.234	7.298	6.743	7.641	6.886
ZYX	9.970	9.273	9.635	9.343	9.625	9.331	10.162	9.005
ZXDC	6.796	7.283	7.032	7.152	7.113	7.052	7.094	7.714
ZXDB	5.377	5.473	5.443	5.496	5.589	5.399	5.383	5.327
ZWINT	5.918	8.560	6.723	7.729	6.235	7.966	6.275	7.400
ZWILCH	4.861	5.493	4.730	5.449	4.790	5.409	4.909	5.534
ZW10	7.323	7.066	7.302	7.406	7.378	7.520	7.570	7.631
ZSWIM8	8.281	8.221	8.498	8.103	8.086	8.151	8.547	7.791
ZSWIM1	6.014	6.077	5.975	6.043	6.278	6.288	6.010	6.029
ZSCAN9	6.433	6.104	6.267	6.559	6.475	6.124	6.424	6.951
ZSCAN5A	5.888	5.596	5.669	5.578	5.729	5.978	5.703	5.983
ZSCAN32	7.057	7.025	7.047	7.280	7.115	7.010	6.815	7.052
ZSCAN31	6.143	7.365	5.948	6.103	5.873	5.699	6.233	8.002
ZSCAN26	6.087	6.234	6.245	6.255	6.215	6.032	6.368	6.425
ZSCAN2	5.677	5.455	5.462	5.611	5.735	5.726	5.653	5.595
ZSCAN18	8.233	7.741	8.306	6.861	7.714	7.981	8.446	7.510
ZSCAN16	4.411	4.802	4.781	4.768	4.579	4.687	4.677	5.065
ZSCAN12	4.188	4.108	4.199	4.088	4.136	4.201	4.110	4.246
ZRSR2	6.918	6.408	6.819	6.347	7.201	6.612	7.068	6.958

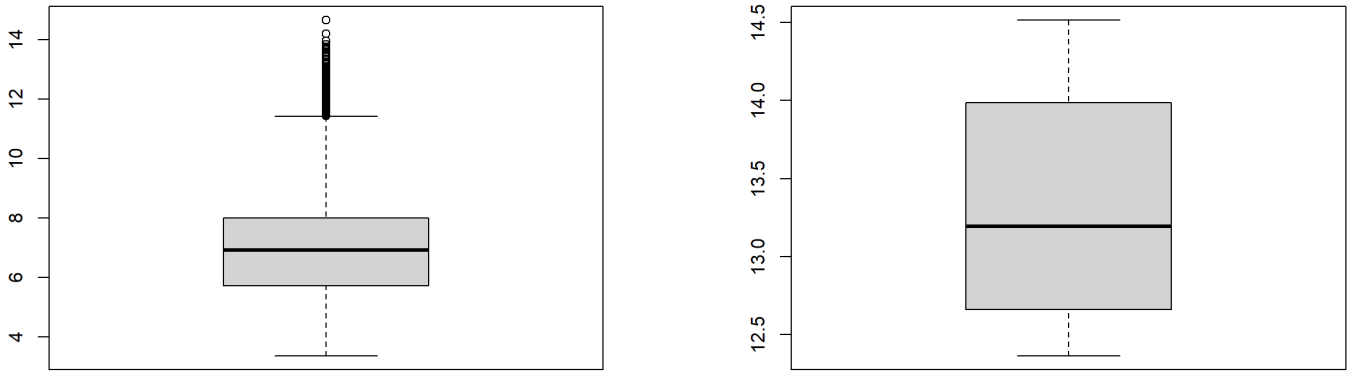
> dat重构后部分数据

四、基础表达分析

代码块

```
# 第四部分：基础表达分析
exprSet=dat  # 更新表达矩阵
exprSet['GAPDH',]  # 查看管家基因GAPDH表达，“如果管家基因表达异常的话，说明样本存在问题”
'GSM73386 GSM73387 GSM73388 GSM73389 GSM73390 GSM73391
  12.663   13.654   12.733   13.847   12.677   14.517
GSM73392 GSM73393 GSM73394 GSM73395
  12.634   13.988   12.364   14.067 '

boxplot(exprSet[,1])  # 绘制第一个样本的箱线图
boxplot(exprSet['GAPDH',])  # 绘制GAPDH的箱线图
exprSet['ACTB',]  # 查看管家基因ACTB表达
'GSM73386 GSM73387 GSM73388 GSM73389 GSM73390 GSM73391
  13.746   13.458   13.824   13.438   13.719   13.836
GSM73392 GSM73393 GSM73394 GSM73395
  13.844   13.562   13.731   13.748 '
```

> exprSet函数



> 左图为第一个样本GSM73386的表达量箱线图，由图中可以看出，有许多异常点
>
> 右图为管家基因GAPDH的表达量的箱线图

五、数据可视化

代码块

```
# 第五部分：数据可视化
library(reshape2)  # 加载数据重塑包
exprSet_L=melt(exprSet)  # 宽表转长表
colnames(exprSet_L)=c('probe','sample','value')  # 设置列名

exprSet_L$group=rep(group_list,each=nrow(exprSet))  # 添加分组信息
#group_list: 每一个细胞的类型储存于group_list中
' [1] "Normal cells" "Tumor cells"  "Normal cells" "Tumor cells"  "Normal cells"
 [6] "Tumor cells"  "Normal cells" "Tumor cells"  "Normal cells" "Tumor cells"
'
#nrow(exprSet)为每一个细胞样本对应的基因数量，其值为 12643;
#each=nrow(exprSet)将group_list中的每一个数据重复12643次


head(exprSet_L)  # 查看长格式数据
'  probe   sample value        group
1  ZZZ3 GSM73386 6.812 Normal cells
```

```
2 ZZEF1 GSM73386 7.552 Normal cells
3   ZYX GSM73386 9.970 Normal cells
4  ZXDC GSM73386 6.796 Normal cells
5  ZXDB GSM73386 5.377 Normal cells
6 ZWINT GSM73386 5.918 Normal cells '

### ggplot2可视化
library(ggplot2)
# 箱线图
p=ggplot(exprSet_L,aes(x=sample,y=value,fill=group))+geom_boxplot()
print(p)
# 小提琴图
p=ggplot(exprSet_L,aes(x=sample,y=value,fill=group))+geom_violin()
print(p)
# 分面直方图
p=ggplot(exprSet_L,aes(value,fill=group))+geom_histogram(bins=200)+facet_wrap(~sample,nrow=4)
print(p)
# 分面密度图
p=ggplot(exprSet_L,aes(value,col=group))+geom_density()+facet_wrap(~sample,nrow=4)
print(p)
# 合并密度图
p=ggplot(exprSet_L,aes(value,col=group))+geom_density()
print(p)
# 带均值点的箱线图
p=ggplot(exprSet_L,aes(x=sample,y=value,fill=group))+geom_boxplot()
p=p+stat_summary(fun.y="mean",geom="point",shape=23,size=3,fill="red")
p=p+theme_set(theme_set(theme_bw(base_size=20)))
p=p+theme(text=element_text(face='bold'),axis.text.x=element_text(angle=30,hjust=1),axis.title=element_blank())
print(p)
```

`melt()` 是数据重塑函数，用于将宽格式数据转换为长格式数据，这是数据可视化（特别是ggplot2）的标准输入格式。

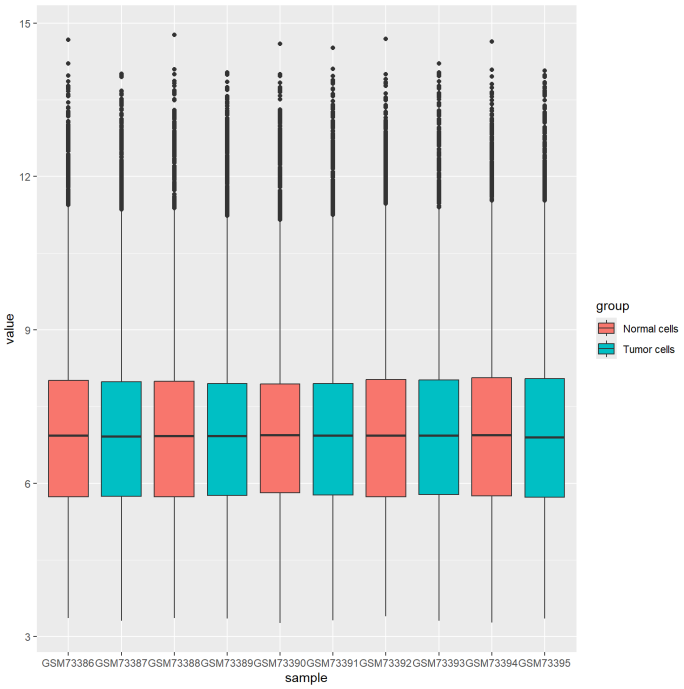
`rep()` 用于重复向量元素，创建特定模式的序列。

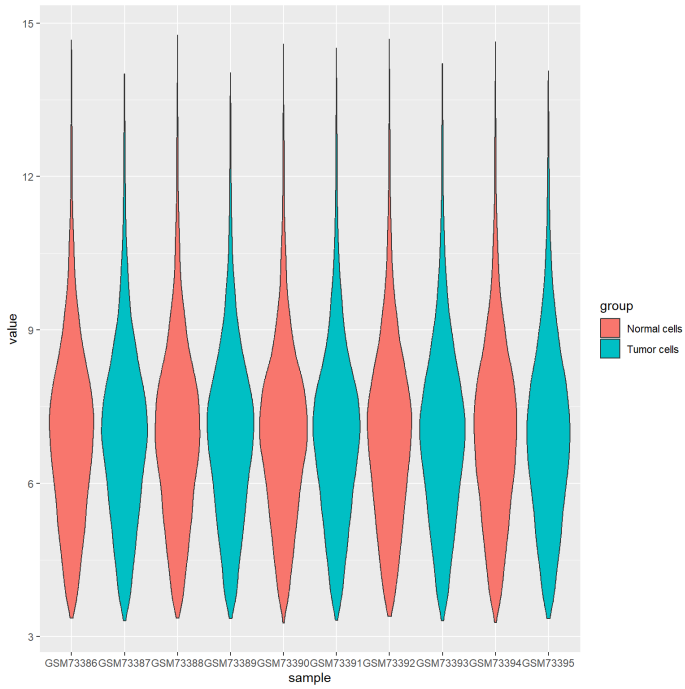
- `group_list`：样本分组向量（如 `c("Control", "Treatment", "Control")`）
- `each = nrow(exprSet)`：每个分组重复基因数量次

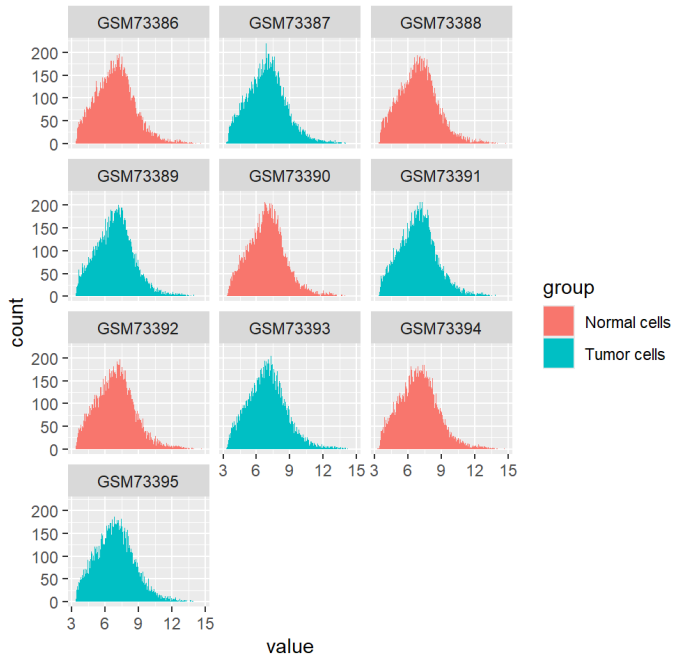
	probe	sample	value	group
1	ZZZ3	GSM73386	6.812	Normal cells
2	ZZEF1	GSM73386	7.552	Normal cells
3	ZYX	GSM73386	9.970	Normal cells
4	ZXDC	GSM73386	6.796	Normal cells
5	ZXDB	GSM73386	5.377	Normal cells
6	ZWINT	GSM73386	5.918	Normal cells
7	ZWILCH	GSM73386	4.861	Normal cells
8	ZW10	GSM73386	7.323	Normal cells
9	ZSWIM8	GSM73386	8.281	Normal cells
10	ZSWIM1	GSM73386	6.014	Normal cells
11	ZSCAN9	GSM73386	6.433	Normal cells
12	ZSCAN5A	GSM73386	5.888	Normal cells
13	ZSCAN32	GSM73386	7.057	Normal cells
14	ZSCAN31	GSM73386	6.143	Normal cells

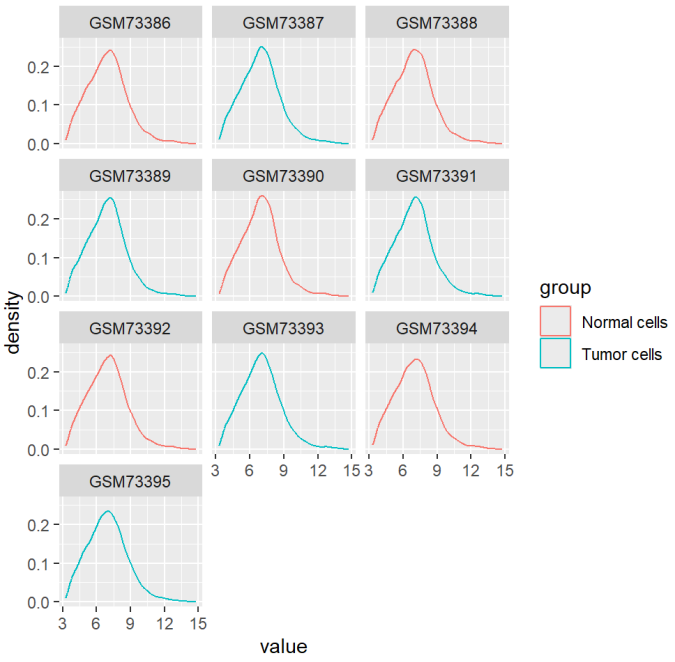
基因 细胞编号 表达量 细胞类型

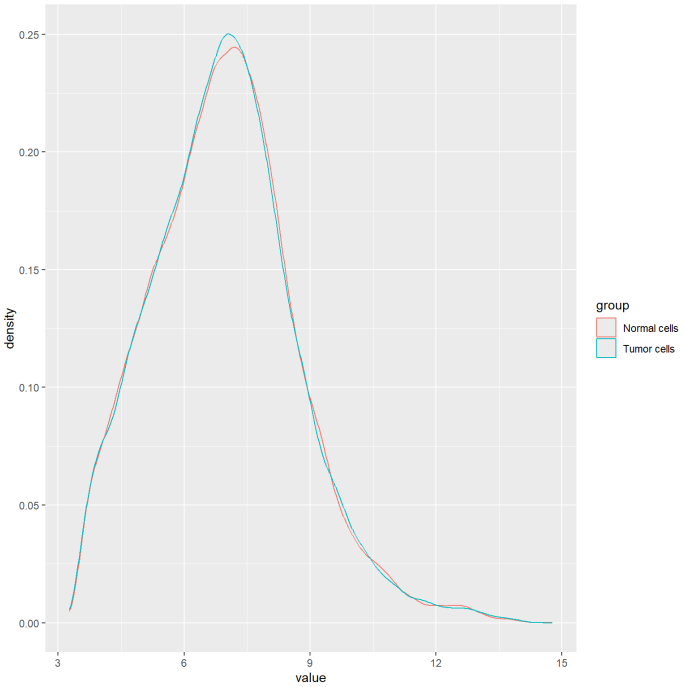
> exprSet_L数据表展示（用melt函数转换为长格式）

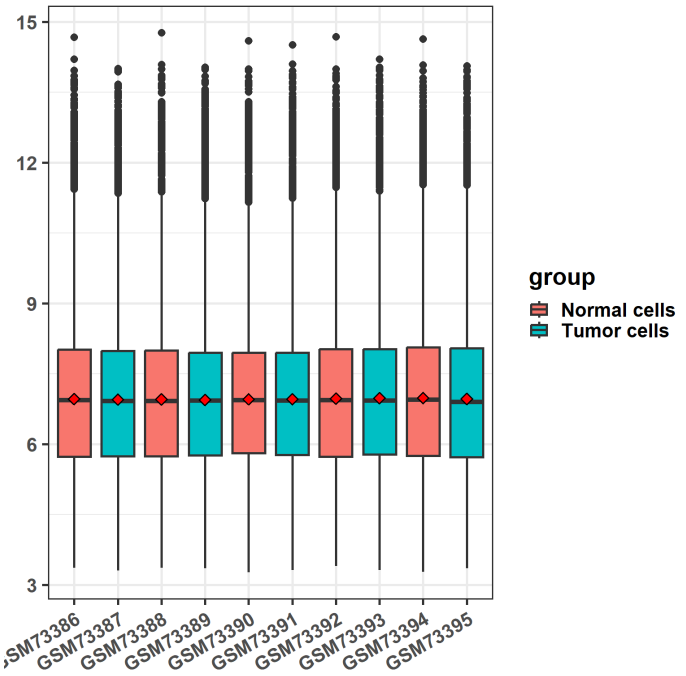












六、基因表达变异性分析

代码块

```
# 第六部分：基因表达变异性分析
## 计算各种统计量并取前50
g_mean <- tail(sort(apply(exprSet,1,mean)),50) #按照基因从小到大的表达量进行排序
'TMSB10   RPS24   RPS27    PPIA   RPLP1   RPS12   RPS11   RPS4X
12.6906 12.6930 12.7165 12.7224 12.7312 12.7321 12.7790 12.7833
  RPS19    RPL3   RPL7A     B2M   RPL30   HLA-A    RPL7   RPS15
12.7849 12.8003 12.8258 12.8267 12.8505 12.8615 12.8619 12.8703
  RPS20  RPL27A    CFL1   RPL39   HLA-C     UBB   EEF1G  SFTPA2
12.8708 12.9011 12.9307 12.9378 12.9400 12.9494 12.9718 12.9935
```

```
  RPL31   RPS17   RPL32  TUBA1C   RPL13   HUWE1   RPL10   RPS16
13.0026 13.0280 13.0550 13.0920 13.1383 13.1498 13.1601 13.1630 '


g_median <- tail(sort(apply(exprSet,1,median)),50)
g_max <- tail(sort(apply(exprSet,1,max)),50)
g_min <- tail(sort(apply(exprSet,1,min)),50)
g_sd <- tail(sort(apply(exprSet,1,sd)),50)
g_var <- tail(sort(apply(exprSet,1,var)),50)
g_mad <- tail(sort(apply(exprSet,1,mad)),50)  # 中位数绝对偏差
g_mad
names(g_mad)  # 查看高变异基因名

## 热图绘制
install.packages('pheatmap')
library(pheatmap)
# 1. 计算每个基因的MAD（中位数绝对偏差），使用中位数偏差代表基因变异度
gene_mad <- apply(exprSet, 1, mad)
# 2. 按MAD排序并取最大的50个
sorted_mad <- sort(gene_mad, decreasing = TRUE)
top_genes <- names(head(sorted_mad, 50))

choose_matrix=exprSet[top_genes,]  # 提取这些基因的表达矩阵
#print(choose_matrix)
'         GSM73386 GSM73387 GSM73388 GSM73389 GSM73390 GSM73391
KRT6A       4.506   10.912    4.588   11.773    4.607   11.206
SPP1        7.612   11.959    5.287   11.098    6.540   10.734
KRT6B       6.833   11.152    6.724   11.133    6.535   11.722
FGG        10.011    4.942    6.109    5.195    7.357    5.368
AGER       11.273    5.480   11.548    6.535   10.649    7.174'

choose_matrix=t(scale(t(choose_matrix)))  # 基因水平标准化,'z-score标准化'
#scale(t(choose_matrix))对每一列基因在不同样本间的表达量进行标准化
#print(choose_matrix)
'           GSM73386   GSM73387   GSM73388    GSM73389   GSM73390
KRT6A    -0.99664910  0.9002355 -0.9723680  1.15518672 -0.9667419
SPP1     -0.34741155  1.2384312 -1.1956021  0.92432705 -0.7384912
KRT6B    -0.91432918  0.8293072 -0.9583339  0.82163668 -1.0346357
FGG       0.96131294 -1.1901707 -0.6948498 -1.08278751 -0.1651494
AGER      0.95445196 -1.3645264  1.0645364 -0.94220256  0.7046604'


pheatmap(choose_matrix)  # 绘制热图

## UpSetR可视化
install.packages('UpSetR')
library(UpSetR)
```

```
57
58  # 创建一个包含所有可能基因的独特列表
59  g_all <- unique(c(names(g_mean),names(g_median),names(g_max),names(g_min),
60                    names(g_sd),names(g_var),names(g_mad)))
61  #整合来自不同筛选标准的基因，为后续交集分析做准备。
62  '[1] "TMSB10"    "RPS24"     "RPS27"     "PPIA"      "RPLP1"
63    [6] "RPS12"     "RPS11"     "RPS4X"     "RPS19"     "RPL3"
64   [11] "RPL7A"     "B2M"       "RPL30"     "HLA-A"     "RPL7"
65   [16] "RPS15"     "RPS20"     "RPL27A"    "CFL1"      "RPL39"
66   [21] "HLA-C"     "UBB"       "EEF1G"     "SFTPA2"    "RPL31"    '
67
68
69  # 构建一个二进制矩阵，表示每个基因是否出现在各个统计方法的结果中。
70  dat=data.frame(g_all=g_all,
71                 #使用 ifelse()函数：如果基因在该方法的列表中，值为1，否则为0。
72                 g_mean=ifelse(g_all %in% names(g_mean),1,0),
73                 g_median=ifelse(g_all %in% names(g_median),1,0),
74                 g_max=ifelse(g_all %in% names(g_max),1,0),
75                 g_min=ifelse(g_all %in% names(g_min),1,0),
76                 g_sd=ifelse(g_all %in% names(g_sd),1,0),
77                 g_var=ifelse(g_all %in% names(g_var),1,0),
78                 g_mad=ifelse(g_all %in% names(g_mad),1,0))
79  #print(dat)
80  '        g_all g_mean g_median g_max g_min g_sd g_var g_mad
81  1      TMSB10      1        1     0     0    0     0     0
82  2       RPS24      1        1     0     1    0     0     0
83  3       RPS27      1        1     0     1    0     0     0
84  4        PPIA      1        0     0     1    0     0     0
85  5       RPLP1      1        1     0     1    0     0     0'
86
87  upset(dat,nsets=7)  # 绘制UpSet图，sets = 7：指定显示7个集合（对应7种统计方法）。
88
89
```

`tail()` 用于查看数据对象的最后部分，是数据探索和结果验证的重要工具。

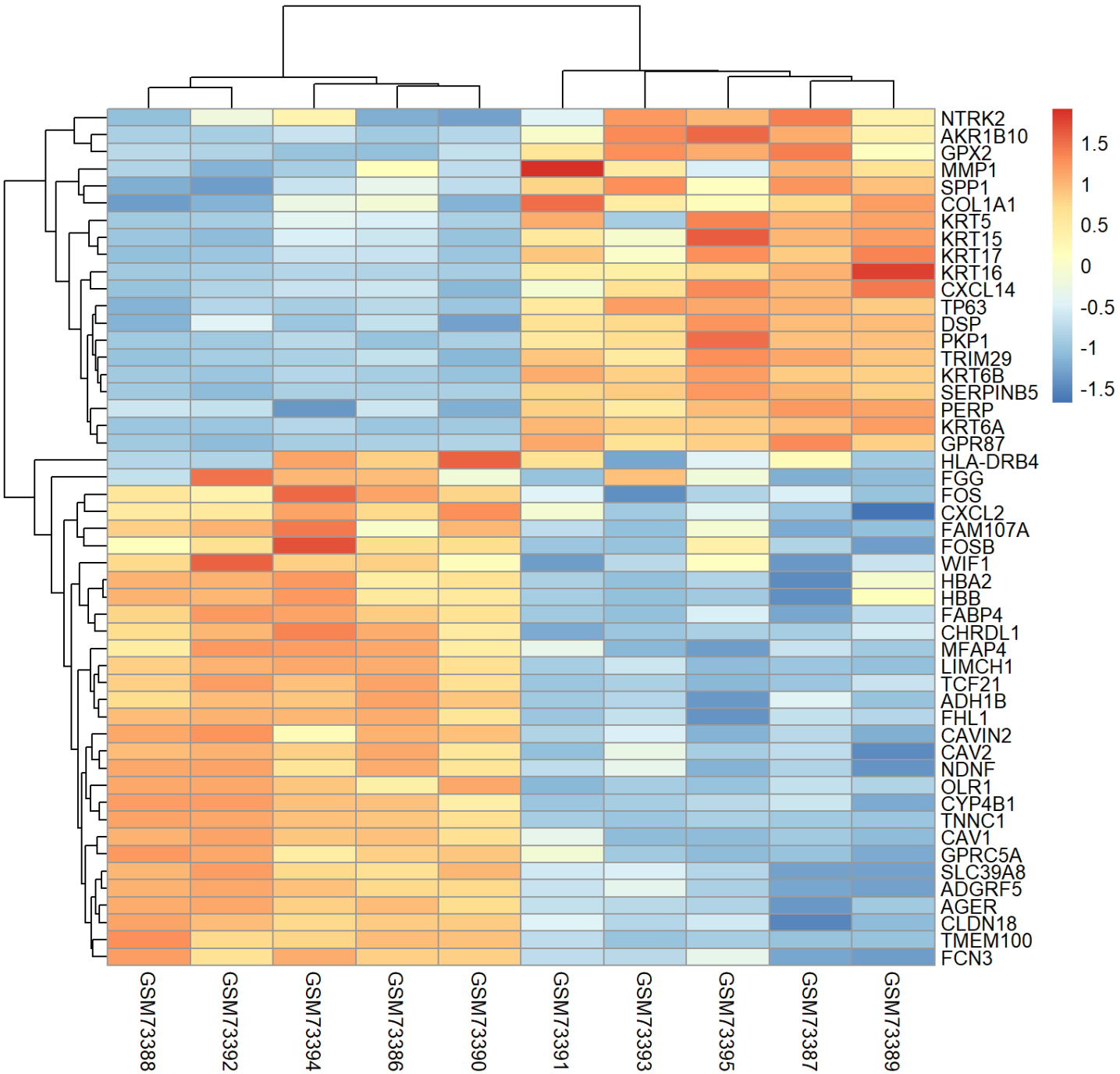
- `x`：数据对象（向量、矩阵、数据框）
- `n`：要显示的元素数量（默认为6）

`names()` 用于获取或设置向量、列表等对象的名称属性，是数据注释的关键工具。

- 提取向量中所有元素的名称
- 返回基因符号组成的字符向量

`scale ()` 对数据进行“z-score标准化”

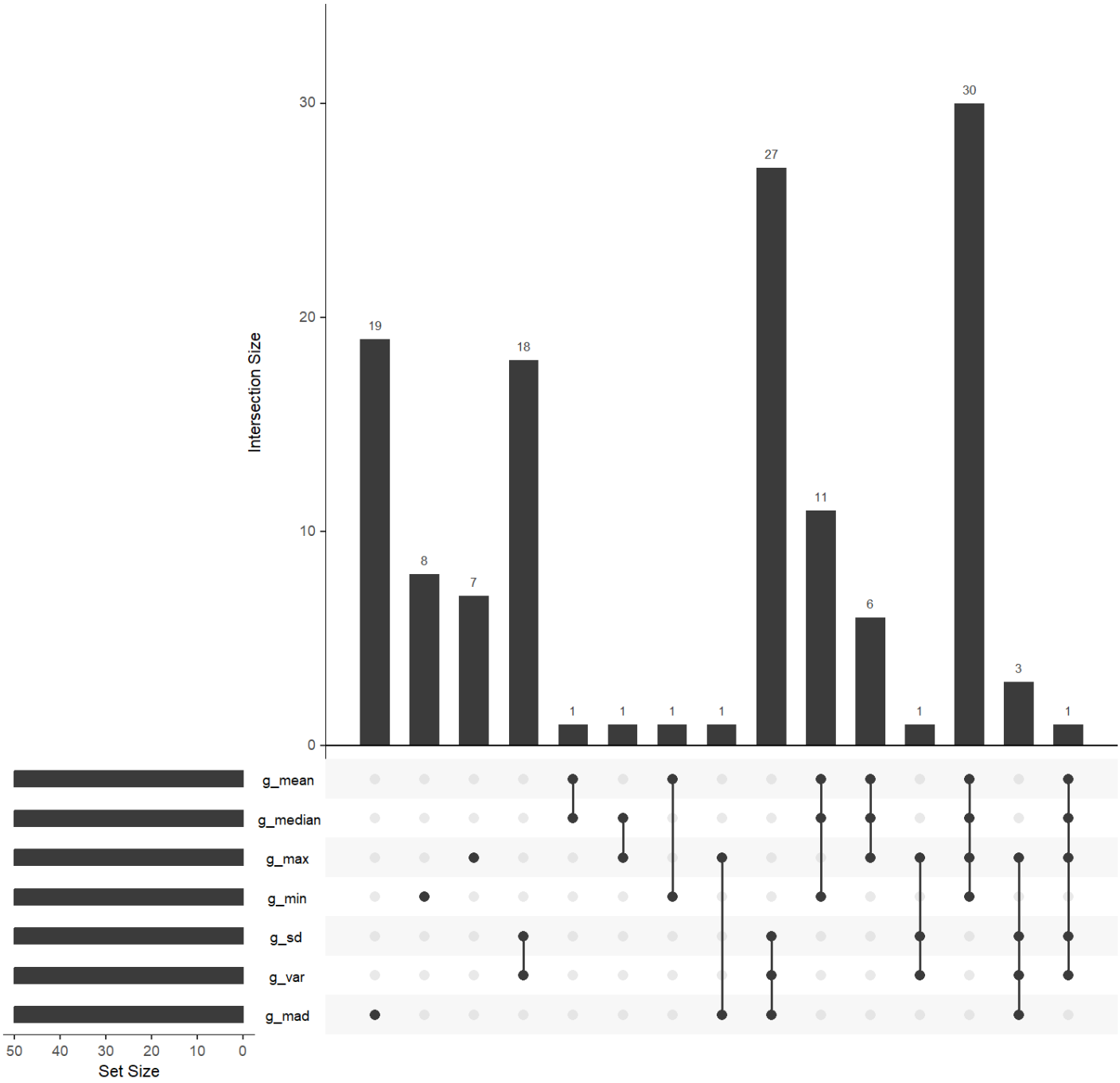
- 每列减去该列的均值
- 除以该列的标准差



- Y轴：50个基因
- X轴：10个样本
- 颜色尺度：蓝色-1.5 到红色 + 1.5
  - 红色区域:识别在特定样本组中一致高表达的基因,这些可能是样本的特征性标志物或者潜在的驱动基因
  - 蓝色区域：识别在特定样本组中一致低表达的基因，这可能指示功能缺失或者抑制状态
- 聚类分析：两侧树状图显示基因和样本的聚类关系

    - GSM73388 GSM73392 GSM73394 GSM73386 GSM73390 在聚类为一簇，与实际情况“normal cell“ 结果吻合，
    - GSM73391 GSM73393 GSM73395 GSM73387 GSM73389 聚类为一族，与实际情况”cancer cell“结果吻合

- 生物学意义解读
    - 鳞状细胞特征
        - 多个角蛋白基因(KRT系列)表达提示上皮细胞来源鳞状细胞癌
    - 细胞连接与粘附：
        - DSP(桥粒斑蛋白)、PKP1(plakophilin-1)在鳞状肺癌细胞表达很高
        - 可能反映细胞连接状态的变化，
    - 肿瘤微环境：
        - CXCL14, CXCL2是趋化因子，参与免疫细胞招募
    - 潜在治疗靶点：
        - NTRK2是已知的治疗靶点



> `g_mean`：前50个基于平均表达量筛选的基因
>
> `g_median`：前50个基于中位数表达量筛选的基因
>
> `g_max`：前50个基于最大表达量筛选的基因
>
> `g_min`：前50个基于最小表达量筛选的基因
>
> `g_sd`：前50个基于标准差筛选的基因（高变异基因）
>
> `g_var`：前50个基于方差筛选的基因
>
> `g_mad`：前50个基于中位数绝对偏差筛选的基因（高稳健变异基因）

- 上方柱状图：表示不同集合交集的基因数量。从图中可以看出最大交集有30个，最小0个，柱状图的高度值反映交集的规模
- 下方点线图：点代表每一个统计方法，连线代表这些方法的交集关系

- 下左方条形图：条形的长度代表每种统计方法所包含的样本数量
- 轴标签
  - Y轴：交集大小
  - X轴：表示单交集的排序
- 生物学意义分析
  - 大型交集：可能为高置信度基因，包括管家基因，关键调控因子和潜在生物标志物
  - 小型交集：
    - 高变异基因（仅被g_sd、g_var、g_mad选中），可能与样本异质性相关。
    - 极端表达基因（仅被g_max或g_min选中），可能在特定样本组中表达异常。
  - 空交集：某些方法组合没有共同基因，表明这些方法筛选标准差异大，可能捕获不同生物学信号。

七、样本聚类与降维

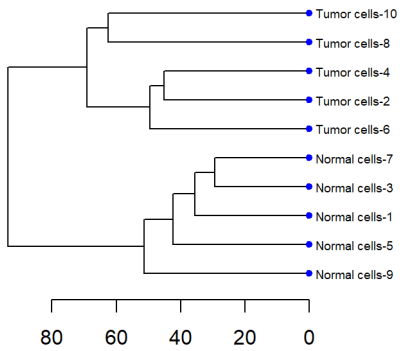
代码块

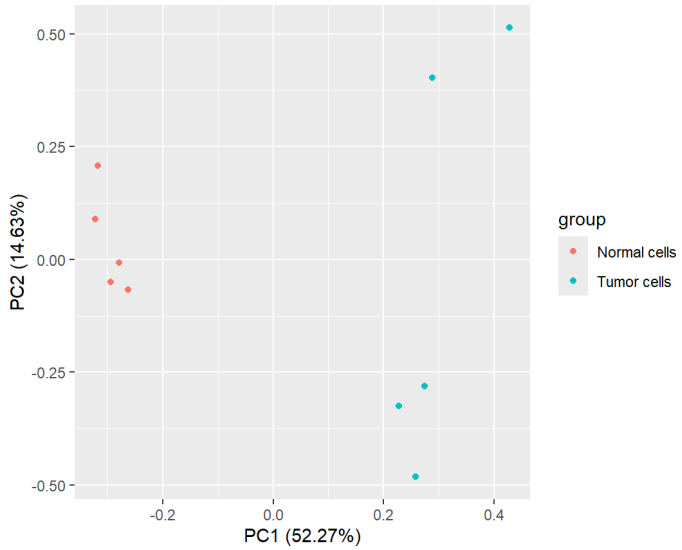
```
                    # 第七部分：样本聚类与降维

group_list=as.character(pdata[,2])  # 重新获取分组信息
' "GSM73386" "GSM73387" "GSM73388" "GSM73389" "GSM73390" "GSM73391"
  "GSM73392" "GSM73393" "GSM73394" "GSM73395"'
dim(exprSet)
'12643    10'
exprSet[1:5,1:5]
'      GSM73386 GSM73387 GSM73388 GSM73389 GSM73390
ZZZ3     6.812    6.199    5.896    6.210    5.995
ZZEF1    7.552    7.060    7.488    7.234    7.298
ZYX      9.970    9.273    9.635    9.343    9.625
ZXDC     6.796    7.283    7.032    7.152    7.113
ZXDB     5.377    5.473    5.443    5.496    5.58'


## 层次聚类
colnames(exprSet)=paste(group_list,1:10,sep='')  # 重命名列
#paste(group_list, 1:10, sep=''): 将分组信息与序号结合

nodePar <- list(lab.cex=0.6,pch=c(NA,19),cex=0.7,col="blue")  # 节点参数
#lab.cex=0.6: 标签大小
#pch=c(NA,19): 节点符号（NA表示无符号，19为实心圆）
#col="blue": 节点颜色
```

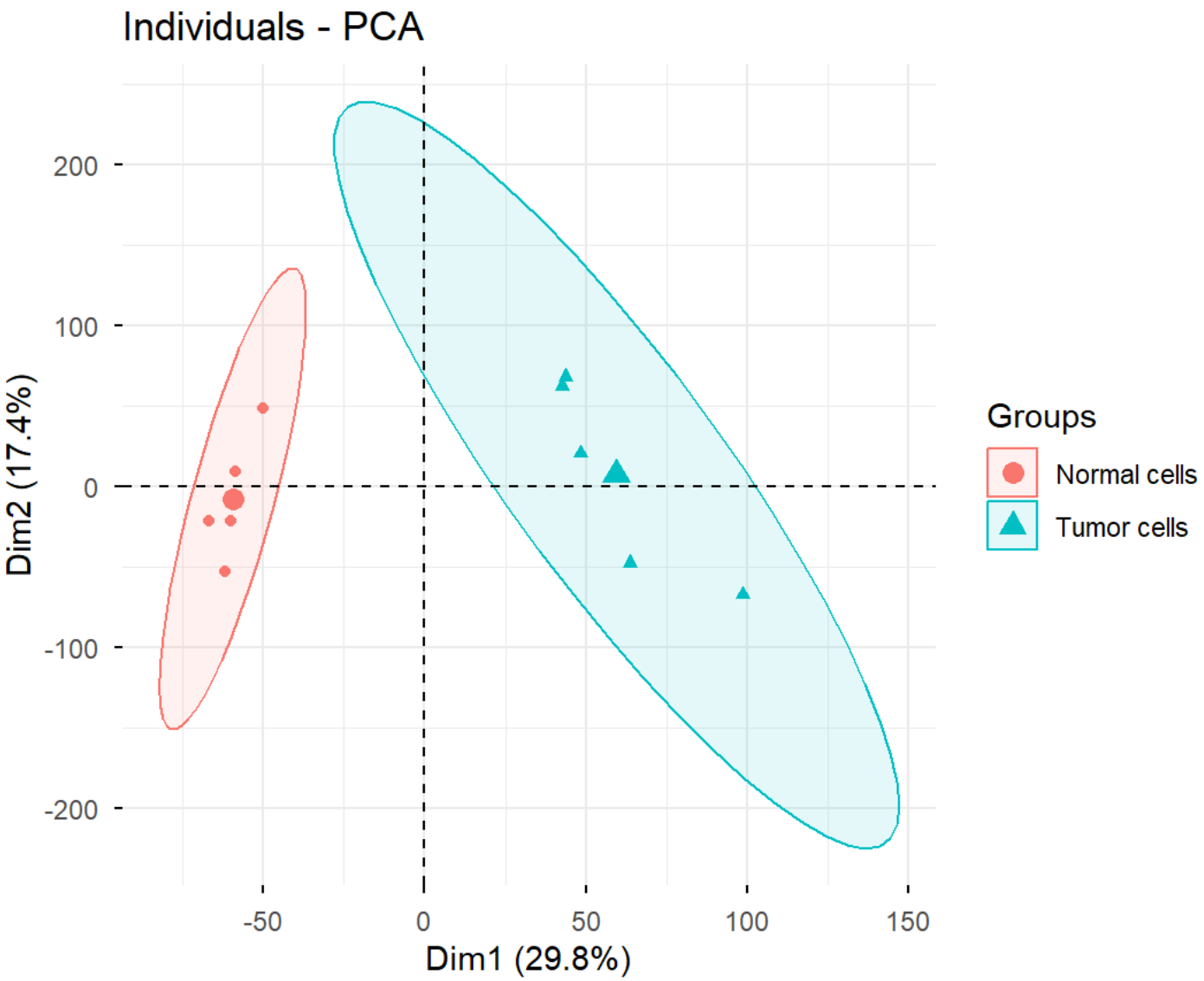
```r

# 样本聚类,dist(t(exprSet))：计算样本间的欧氏距离，转置t()使样本为行，基因为列
hc=hclust(dist(t(exprSet)))

par(mar=c(5,5,5,10))  # 调整边距

 # 绘制水平树状图
plot(as.dendrogram(hc),nodePar=nodePar,horiz=TRUE)
#as.dendrogram()：将聚类结果转为树状图对象
#horiz=TRUE：创建水平树状图


## PCA分析
install.packages('ggfortify')
install.packages('FactoMineR')
install.packages('factoextra')

library(ggfortify)
df=as.data.frame(t(exprSet))  # 转置矩阵，样本为行，基因为列

df$group=group_list  # 添加分组
autoplot(prcomp(df[,1:(ncol(df)-1)]),data=df,colour='group')  # 自动绘图
#autoplot()：自动绘制PCA结果；prcomp()：执行主成分分析，排除分组列；colour='group'按组着色

library("FactoMineR")
library("factoextra")
df=as.data.frame(t(exprSet)) # 转置矩阵，样本为行，基因为列
dat.pca <- PCA(df,graph=FALSE)  # 执行PCA

fviz_pca_ind(dat.pca,  # 专门绘制样本PCA结果
             geom.ind="point",    #样本显示为点
             col.ind=as.factor(group_list), #按因子着色
             addEllipses=TRUE,    #添加分组椭圆
             legend.title="Groups") #分组标签

```

	ZZZ3	ZZEF1	ZYX	ZXDC	ZXDB	ZWINT	ZWILCH	ZW10	ZSWIM8
GSM733861	6.812	7.552	9.970	6.796	5.377	5.918	4.861	7.323	8.281
GSM733872	6.199	7.060	9.273	7.283	5.473	8.560	5.493	7.066	8.221
GSM733883	5.896	7.488	9.635	7.032	5.443	6.723	4.730	7.302	8.498
GSM733894	6.210	7.234	9.343	7.152	5.496	7.729	5.449	7.406	8.103
GSM733905	5.995	7.298	9.625	7.113	5.589	6.235	4.790	7.378	8.086
GSM733916	6.360	6.743	9.331	7.052	5.399	7.966	5.409	7.520	8.151
GSM733927	5.913	7.641	10.162	7.094	5.383	6.275	4.909	7.570	8.547
GSM733938	6.723	6.886	9.005	7.714	5.327	7.400	5.534	7.631	7.791
GSM733949	6.269	7.718	10.340	7.251	5.353	6.040	4.540	7.143	8.857
GSM7339510	7.576	6.676	9.000	7.875	5.231	8.491	5.781	7.912	8.292

> df数据表展示







- Dim1 (第一主成分)：解释29.8%的总体变异，Dim2 (第二主成分)：解释17.4%的总体变异，累计解释度47.2%（29.8% + 17.4%）
- 图中每一个点代表一个样本
- 靠得近的样本具有相似的基因表达图谱

八、差异表达分析

代码块

```
# 第九部分：差异表达分析
## t检验方法
dat = exprSet
group_list=as.factor(group_list)  # 转为因子
#print(group_list)
' [1] Normal cells Tumor cells  Normal cells Tumor cells  Normal cells
 [6] Tumor cells  Normal cells Tumor cells  Normal cells Tumor cells
Levels: Normal cells Tumor cells'
```

```
group1 = which(group_list == levels(group_list)[1])  # “Normal cells”索引
#print(group_list == levels(group_list)[1])
'TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE'
#which(group_list == levels(group_list)[1])
'1 3 5 7 9'

group2 = which(group_list == levels(group_list)[2])  # “Tumor cells”索引
'FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE'
#which(group_list == levels(group_list)[2])
'2  4  6  8 10'

dat1 = dat[, group1]  # “Normal cells”数据,1 3 5 7 9 列
#head(dat1)
'      Normal cells-1 Normal cells-3 Normal cells-5 Normal cells-7 Normal
cells-9
ZZZ3           6.812          5.896          5.995          5.913
6.269
ZZEF1          7.552          7.488          7.298          7.641
7.718
ZYX            9.970          9.635          9.625         10.162
10.340
ZXDC           6.796          7.032          7.113          7.094
7.251
ZXDB           5.377          5.443          5.589          5.383
5.353
ZWINT          5.918          6.723          6.235          6.275
6.040'
dat2 = dat[, group2]  # “Tumor cells”数据,2 4 6 8 10 列
#head(dat2)
'     Tumor cells-2 Tumor cells-4 Tumor cells-6 Tumor cells-8 Tumor cells-10
ZZZ3          6.199         6.210         6.360         6.723          7.576
ZZEF1         7.060         7.234         6.743         6.886          6.676
ZYX           9.273         9.343         9.331         9.005          9.000
ZXDC          7.283         7.152         7.052         7.714          7.875
ZXDB          5.473         5.496         5.399         5.327          5.231
ZWINT         8.560         7.729         7.966         7.400          8.49'

dat = cbind(dat1, dat2)  # 合并数据
#head(dat)
'      dat1  dat2
ZZZ3  6.812 6.199
ZZEF1 7.552 7.060
ZYX   9.970 9.273
ZXDC  6.796 7.283
ZXDB  5.377 5.473
ZWINT 5.918 8.560'
```

```
49
50  # 对每个基因进行t检验
51  pvals = apply(exprSet,1,function(x){t.test(as.numeric(x)~group_list)$p.value})
52  #head(pvals)
53  '      ZZZ3        ZZEF1          ZYX         ZXDC         ZXDB        ZWINT
54  0.2029566944 0.0015490661 0.0031297584 0.0933966496 0.5178616790 0.0002962808 '
55
56  p.adj = p.adjust(pvals,method="BH")  # p值校正
57  #head(p.adj)
58  '     ZZZ3      ZZEF1        ZYX       ZXDC       ZXDB      ZWINT
59  0.40105994 0.02583752 0.03705024 0.25076230 0.69730636 0.01166940'
60
61  avg_1 = rowMeans(dat1)  #  计算正常组平均表达
62  #head(avg_1)
63  '  ZZZ3  ZZEF1    ZYX   ZXDC   ZXDB  ZWINT
64  6.1770 7.5394 9.9464 7.0572 5.4290 6.2382 '
65
66  avg_2 = rowMeans(dat2)  # 计算肿瘤组平均表达
67  #head(avg_2)
68  '  ZZZ3  ZZEF1    ZYX   ZXDC   ZXDB  ZWINT
69  6.6136 6.9198 9.1904 7.4152 5.3852 8.0292 '
70
71  log2FC = avg_2-avg_1  # # 计算log2折叠变化（ Tumor-Normal）
72  #head(log2FC)
73  '   ZZZ3   ZZEF1     ZYX    ZXDC    ZXDB   ZWINT
74   0.4366 -0.6196 -0.7560  0.3580 -0.0438  1.7910'
75
76  DEG_t.test = cbind(avg_1,avg_2,log2FC,pvals,p.adj)  # 组合结果
77  DEG_t.test=DEG_t.test[order(DEG_t.test[,4]),]  # 按p值排序
78  DEG_t.test=as.data.frame(DEG_t.test)  # 转为数据框
79  head(DEG_t.test)  # 查看结果
80  '        avg_1   avg_2  log2FC        pvals        p.adj
81  KRT6A  4.6850 11.0586  6.3736 7.533152e-08 0.0009524164
82  LIMCH1 9.7152  6.1536 -3.5616 2.940358e-07 0.0011772651
83  SNRPC  8.6006  9.1016  0.5010 4.141796e-07 0.0011772651
84  DAPK1  9.3148  7.9608 -1.3540 4.806734e-07 0.0011772651
85  TCF21  7.6272  4.9880 -2.6392 6.080540e-07 0.0011772651
86  KRT6B  6.7652 11.4304  4.6652 6.488116e-07 0.001177265'
87
88
89  ## limma差异分析
90  suppressMessages(library(limma)) #专门为基因表达数据设计的差异分析R包，尤其擅长处理小
    样本数据
91
92  #如果列名中带有空格等字符，运行以下代码
93  if(F){
```

```r
  normal_samples <- c("Normal cells-1","Normal cells-3","Normal cells-5","Normal cells-7","Normal cells-9")
  group_list <- ifelse(colnames(exprSet) %in% normal_samples, "Normal", "Tumor")
  print(group_list)
}

design <- model.matrix(~0+factor(group_list))  # 设计矩阵
#factor(group_list)：将分组向量转为因子（两个水平：Normal和Tumor）
#~0+：指定无截距模型，为每个组创建独立列
colnames(design)=levels(factor(group_list))
rownames(design)=colnames(exprSet)
#head(design)
'               Normal Tumor
Normal cells-1      1     0
Tumor cells-2       0     1
Normal cells-3      1     0
Tumor cells-4       0     1
Normal cells-5      1     0
Tumor cells-6       0     1'

contrast.matrix<-makeContrasts(paste0(unique(group_list), #limma中更常见 Normal-Tumor,与DEG_t.test中刚好相反
                                      collapse="-"),levels=design)  # 对比矩阵
#head(contrast.matrix)
'        Contrasts
Levels   Normal-Tumor
  Normal            1
  Tumor            -1'

fit <- lmFit(exprSet,design)  # 线性模型拟合
fit2 <- contrasts.fit(fit,contrast.matrix)  # 对比拟合，计算每个基因的表达量与分组的关系
fit2 <- eBayes(fit2)  # 贝叶斯方法调整方差估计

tempOutput = topTable(fit2,coef=1,n=Inf)  # 提取结果
#coef=1,指定第一个对比 n=inf 返回所有基因结果

nrDEG = na.omit(tempOutput)  # 处理缺失值，删除任何包含NA值的行
head(nrDEG)
'           logFC AveExpr         t      P.Value    adj.P.Val        B
KRT6A    -6.3736  7.8718 -30.20437 1.388246e-11 1.755159e-07 15.79025
KRT6B    -4.6652  9.0978 -25.29914 8.787077e-11 5.554751e-07 14.51813
SERPINB5 -4.0920  7.4352 -23.57876 1.824758e-10 5.924541e-07 13.97051
TNNC1     3.3818  6.5783  23.51776 1.874410e-10 5.924541e-07 13.94994
TMEM100   4.0222  7.1447  19.15770 1.556108e-09 3.934775e-06 12.23637
LIMCH1    3.5616  7.9344  17.14135 4.861140e-09 1.024323e-05 11.24635'
```

```
137
```

> `makecontrast`：**作用**：定义组间比较关系
>
> - `unique(group_list)`：获取唯一分组（["Normal","Tumor"]）
> - `paste0(...,collapse="-")`：创建"Normal-Tumor"字符串
> - `makeContrasts`：生成对比矩阵，指定比较Normal组vs Tumor组
> - `lmFit`：计算每个基因的表达量与分组的关系
>   - 对表达数据进行线性拟合
>   - 输出：包含系数、残差等信息的拟合对象
> - `contrasts.fit`：计算组间差异（Normal vs Tumor）
>   - 输出：更新后的拟合对象，包含对比结果
> - `eBayes` 使用经验贝叶斯方法调整方差估计
>   - 特别适用于小样本数据，提高差异检测的稳健性
> - `topTable`：提取差异表达基因表
>   - `coef=1`：指定第一个对比（本例只有"Normal-Tumor"）
>   - `n=Inf`：返回所有基因的结果（不限制数量）
> - `na.omit`：删除任何包含NA值的行

- 结果解读

```
> head(DEG_t.test)   # 查看结果
        avg_1   avg_2  log2FC        pvals        p.adj
KRT6A  4.6850 11.0586  6.3736 7.533152e-08 0.0009524164
LIMCH1 9.7152  6.1536 -3.5616 2.940358e-07 0.0011772651
SNRPC  8.6006  9.1016  0.5010 4.141796e-07 0.0011772651
DAPK1  9.3148  7.9608 -1.3540 4.806734e-07 0.0011772651
TCF21  7.6272  4.9880 -2.6392 6.080540e-07 0.0011772651
KRT6B  6.7652 11.4304  4.6652 6.488116e-07 0.0011772651
```

> **log2FC > 0**：表示肿瘤组表达高于正常组（在肿瘤中上调）
>
> **log2FC < 0**：表示肿瘤组表达低于正常组（在肿瘤中下调）
>
> **p.adj < 0.05**：表示差异表达具有统计学显著性（经过多重检验校正）

```
> head(nrDEG)
           logFC AveExpr         t      P.Value    adj.P.Val        B
KRT6A    -6.3736  7.8718 -30.20437 1.388246e-11 1.755159e-07 15.79025
KRT6B    -4.6652  9.0978 -25.29914 8.787077e-11 5.554751e-07 14.51813
SERPINB5 -4.0920  7.4352 -23.57876 1.824758e-10 5.924541e-07 13.97051
TNNC1     3.3818  6.5783  23.51776 1.874410e-10 5.924541e-07 13.94994
TMEM100   4.0222  7.1447  19.15770 1.556108e-09 3.934775e-06 12.23637
LIMCH1    3.5616  7.9344  17.14135 4.861140e-09 1.024323e-05 11.24635
```

- logFC：对数折叠变化
  - logFC < 0 负值：在Tumor组表达上调（Normal < Tumor）
  - logFC > 0 正值：在Tumor组表达下调（Normal > Tumor）
  - **与上述结果刚刚相反，需注意！！！**
- AveExpr：所有样本的平均表达量
- t：t统计量，值越大差异越显著
- `P.Value`：原始p值
- `adj.P.Val`：校正后的p值（FDR）
- `B`：对数几率（log-odds），值越大差异越显著

- 生物学意义

1. **KRT6A**：
   - logFC = -6.37（在肿瘤中显著下调）
   - 角蛋白基因，参与细胞骨架形成
   - 在多种癌症中表达异常，与肿瘤侵袭相关
2. **KRT6B**：
   - logFC = -4.67（在肿瘤中显著下调）
   - 同属角蛋白家族，功能类似KRT6A
3. **SERPINB5**：
   - logFC = -4.09（在肿瘤中下调）
   - 编码maspin蛋白，具有肿瘤抑制功能

> 所有显著基因的logFC均为负值，表明这些基因在肿瘤中表达降低，这与KRT基因在肿瘤中常下调的报道一致，
>
> 下调基因可能参与维持正常细胞结构，在肿瘤中下调可能与细胞去分化相关

- 注意事项

  logFC方向：对比是"Normal-Tumor"，所以logFC = Normal - Tumor ,负logFC表示肿瘤中表达更高

显著性阈值：通常使用adj.P.Val < 0.05和|logFC| > 1

结果验证：通过实验方法(qPCR)验证关键基因

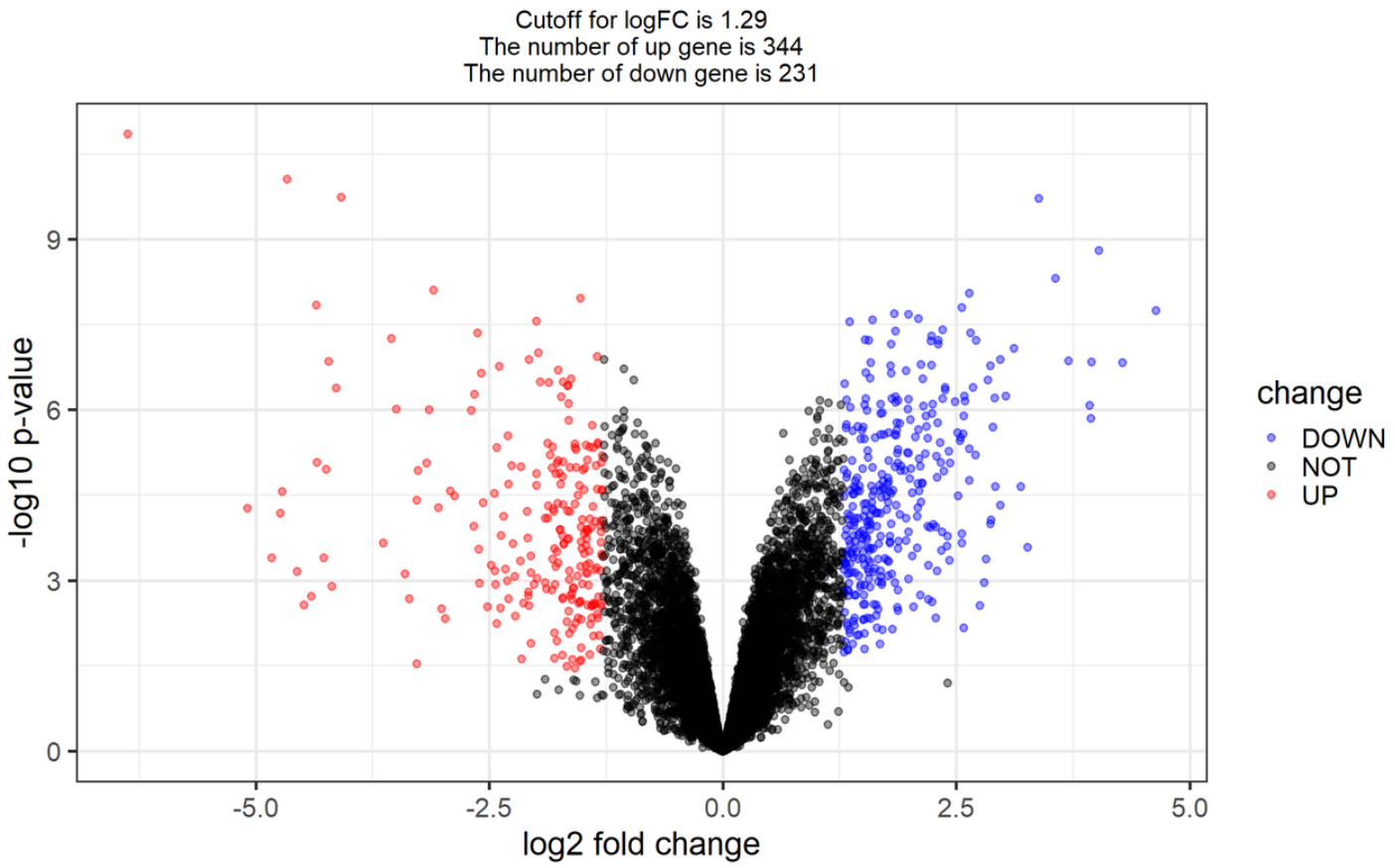
九、结果可视化与验证

代码块

```
# 第九部分：结果可视化与验证
## 火山图
DEG=nrDEG

logFC_cutoff <- with(DEG,mean(abs(logFC))+2*sd(abs(logFC)))  # 动态阈值
#基于logFC绝对值的均值和标准差计算（均值 + 2倍标准差）logFC_cutoff = 1.290401

DEG$change = as.factor(ifelse(DEG$P.Value<0.05 & abs(DEG$logFC)>logFC_cutoff,
                              ifelse(-(DEG$logFC) >
logFC_cutoff,'UP','DOWN'),'NOT'))
#head(DEG)
'           logFC AveExpr         t      P.Value    adj.P.Val        B change
KRT6A    -6.3736  7.8718 -30.20437 1.388246e-11 1.755159e-07 15.79025     UP
KRT6B    -4.6652  9.0978 -25.29914 8.787077e-11 5.554751e-07 14.51813     UP
SERPINB5 -4.0920  7.4352 -23.57876 1.824758e-10 5.924541e-07 13.97051     UP
TNNC1     3.3818  6.5783  23.51776 1.874410e-10 5.924541e-07 13.94994   DOWN
TMEM100   4.0222  7.1447  19.15770 1.556108e-09 3.934775e-06 12.23637   DOWN
LIMCH1    3.5616  7.9344  17.14135 4.861140e-09 1.024323e-05 11.24635   DOWN'
#UP：在肿瘤细胞中显著上调（p值 < 0.05 且 logFC > 阈值）
#DOWN：在肿瘤细胞中显著下调（p值 < 0.05 且 logFC < -阈值）
#NOT：不显著

#this_tile：图标题，显示logFC阈值、上调基因数和下调基因数。
this_tile <- paste0('Cutoff for logFC is ',round(logFC_cutoff,3),
                    '\nThe number of up gene is ',nrow(DEG[DEG$change=='UP',]),
                    '\nThe number of down gene is
',nrow(DEG[DEG$change=='DOWN',]))

#绘制火山图，logFC为x轴，-log10(P.Value)为纵轴，change为分类变量
g = ggplot(data=DEG,aes(x=logFC,y=-log10(P.Value),color=change))+
  geom_vline(xintercept = c(-logFC_cutoff, logFC_cutoff), # 在x轴这两个位置画竖线
             color = "black", # 线条颜色为黑色
             linetype = "dashed", # 线型为虚线
             size = 0.5) + # 线的粗细
  geom_point(alpha=0.4,size=1.75)+
  theme_set(theme_set(theme_bw(base_size=20)))+
  xlab("log2 fold change")+ylab("-log10 p-value")+
  ggtitle(this_tile)+theme(plot.title=element_text(size=15,hjust=0.5))+
```

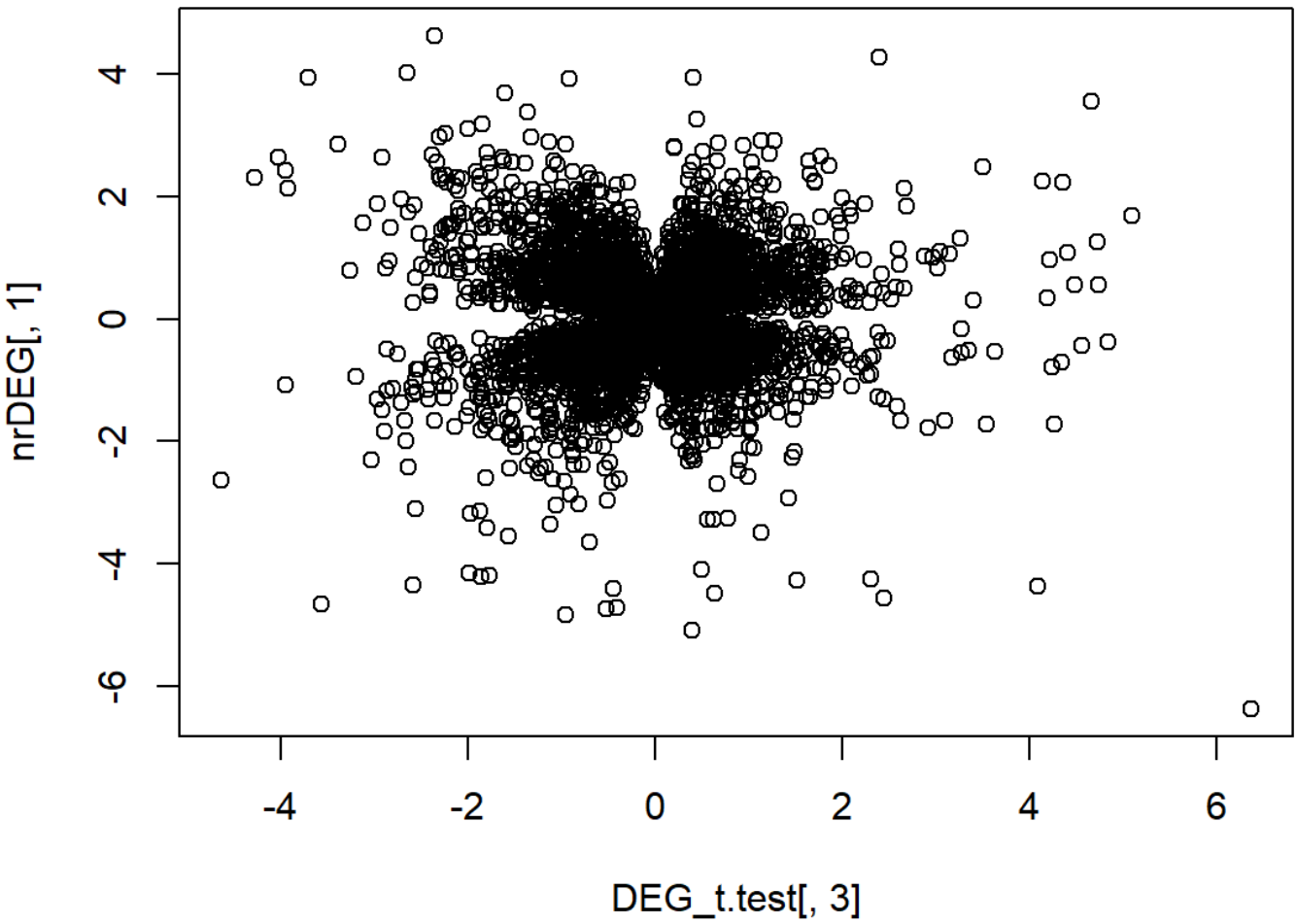
```
  scale_colour_manual(values = c('NOT' = 'black', 'UP' = 'red', 'DOWN' = 'blue'))
  #蓝色代表下调，红色代表上调，黑色代表不显著
print(g)

## 比较两种差异分析方法（t检验和limma）的结果一致性。
#DEG_t.test：来自t检验的结果数据框
#nrDEG：来自limma的结果数据框
plot(DEG_t.test[,3],nrDEG[,1])  # 比较logFC
plot(DEG_t.test[,4],nrDEG[,4])  # 比较p值
plot(-log10(DEG_t.test[,4]),-log10(nrDEG[,4]))  # 比较-log10(p值)

## 特定基因验证
#检查特定基因的表达值，验证数据质量和差异分析结果。
exprSet['GAPDH',]  # 管家基因
'GSM73386 GSM73387 GSM73388 GSM73389 GSM73390 GSM73391
  12.663   13.654   12.733   13.847   12.677   14.517
GSM73392 GSM73393 GSM73394 GSM73395
  12.634   13.988   12.364   14.067 '
exprSet['ACTB',]   # 管家基因
'GSM73386 GSM73387 GSM73388 GSM73389 GSM73390 GSM73391
  13.746   13.458   13.824   13.438   13.719   13.836
GSM73392 GSM73393 GSM73394 GSM73395
  13.844   13.562   13.731   13.748 '
exprSet[ 'DLEU1',]  # 差异基因
'GSM73386 GSM73387 GSM73388 GSM73389 GSM73390 GSM73391
   6.258    6.298    6.232    6.682    6.280    6.906
GSM73392 GSM73393 GSM73394 GSM73395
   6.251    6.724    6.053    6.884 '

## 基因表达箱线图
library(ggpubr)
my_comparisons <- list(c("Normal","Tumor"))  # 比较组

# 创建数据框
dat=data.frame(group=group_list,
               sampleID=names(exprSet['DLEU1',]),
               values=as.numeric(exprSet['DLEU1',]))
#head(dat)
'    group       sampleID values
1  Normal Normal cells-1  6.258
2   Tumor  Tumor cells-2  6.298
3  Normal Normal cells-3  6.232
4   Tumor  Tumor cells-4  6.682
5  Normal Normal cells-5  6.280'

# 绘制箱线图并添加t检验
```

```
ggboxplot(dat,x="group",y="values",color="group",add="jitter")+
  stat_compare_means(comparisons=my_comparisons,method="t.test")  # 添加t检验

## 差异基因热图
library(pheatmap)
choose_gene=head(rownames(nrDEG),25)  # 从limma结果中取p值最小的25个基因（最显著）。
#head(choose_gene)
'[1] "KRT6A"    "KRT6B"    "SERPINB5" "TNNC1"    "TMEM100"
 [6] "LIMCH1"   "GPR87"    "TCF21"    "RACGAP1"  "TP63"
[11] "EPAS1"    "AGER"     "GPD1L"    "A2M"      "ENTREP1"
[16] "DPEP2"    "MIF"      "DAPK1"    "PAPSS2"   "FOXF1"
[21] "RGCC"     "MCM2"     "GATA6"    "TRIM29"   "CELF2'
choose_matrix=exprSet[choose_gene,]
choose_matrix=t(scale(t(choose_matrix)))  # 基因水平标准化
#head(choose_matrix)
'         Normal cells-1 Tumor cells-2 Normal cells-3 Tumor cells-4 Normal
cells-5
KRT6A        -0.9966491     0.9002355     -0.9723680     1.1551867
-0.9667419
KRT6B        -0.9143292     0.8293072     -0.9583339     0.8216367
-1.0346357
SERPINB5     -0.8778658     1.0427530     -0.9666087     0.8133087
-0.8898208
TNNC1         0.8876207    -0.9380493      1.1163860    -0.9942665
0.6755535
TMEM100       0.9717928    -1.0050348      1.2646389    -1.0106217
0.9149928
LIMCH1        1.0731679    -0.9928743      0.7969619    -1.0489540
0.6701272
          Tumor cells-6 Normal cells-7 Tumor cells-8 Normal cells-9 Tumor cells-
10
KRT6A        0.9872920     -0.9747369     0.8182128     -0.8077305
0.8572994
KRT6B        1.0594236     -0.9777121     0.8264812     -0.8234938
1.1716558
SERPINB5     0.7921575     -1.0774225     0.8583699     -0.8921198
1.1972485
TNNC1       -0.9052095      1.1141596    -0.9725589      0.9121114
-0.8957472
TMEM100     -0.7256904      0.7417992    -1.0292447      0.7883566
-0.9109888
LIMCH1      -0.9105890      1.0286186    -0.6160391      1.0978011
-1.0982204'
pheatmap(choose_matrix)  # 绘制热图,红色表示高表达，蓝色表示低表达


```

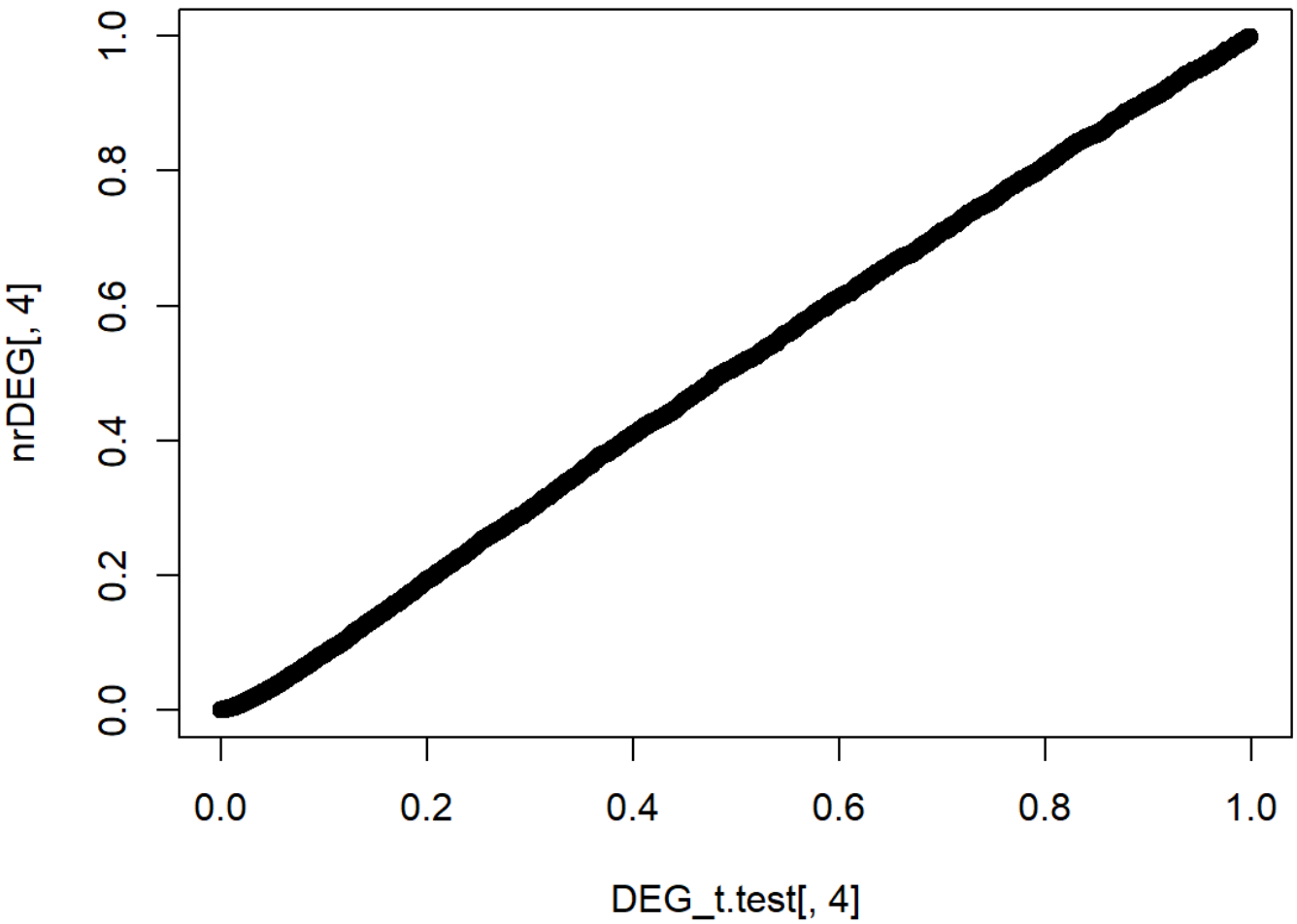


- 横轴:
  - **含义**：表示基因表达量变化的**方向和幅度**。
  - **零点**：图中央的竖线（0点）。一个基因的点越靠**左**，表示它在实验组（如肿瘤）中的表达量相对于对照组（如正常组织）**越高**（上调）。越靠**右**，则表达量**越低**（下调） **我这里与正常火山图相反！！！请大家注意**
- 纵轴
  - **含义**：表示基因表达量变化的**统计显著性**。
  - **数值越大**（点越高），代表P值越小，差异越显著，结果越可靠。通常将 `-log10(0.05) ≈ 1.3` 作为一条参考线，点高于这个高度一般认为具有统计学意义。
- 颜色
  - **红色点 (UP)**：**显著上调**的基因。满足： `P值 < 0.05` **且** `logFC < -1.29` 。共有344个。
  - **蓝色点 (DOWN)**：**显著下调**的基因。满足： `P值 < 0.05` **且** `logFC > 1.29` 。共有231个
  - **灰色点 (NOT)**：**没有显著变化**的基因。不满足上述条件。这些基因占大多数。
- **阈值线 (Cutoff)**
  - 图中有两条**无形的竖线**，分别位于 `x = 1.29` 和 `x = -1.29` 的位置。只有突破了这两条线且P值显著的基因，才会被标记为红色或蓝色
- 生物学意义

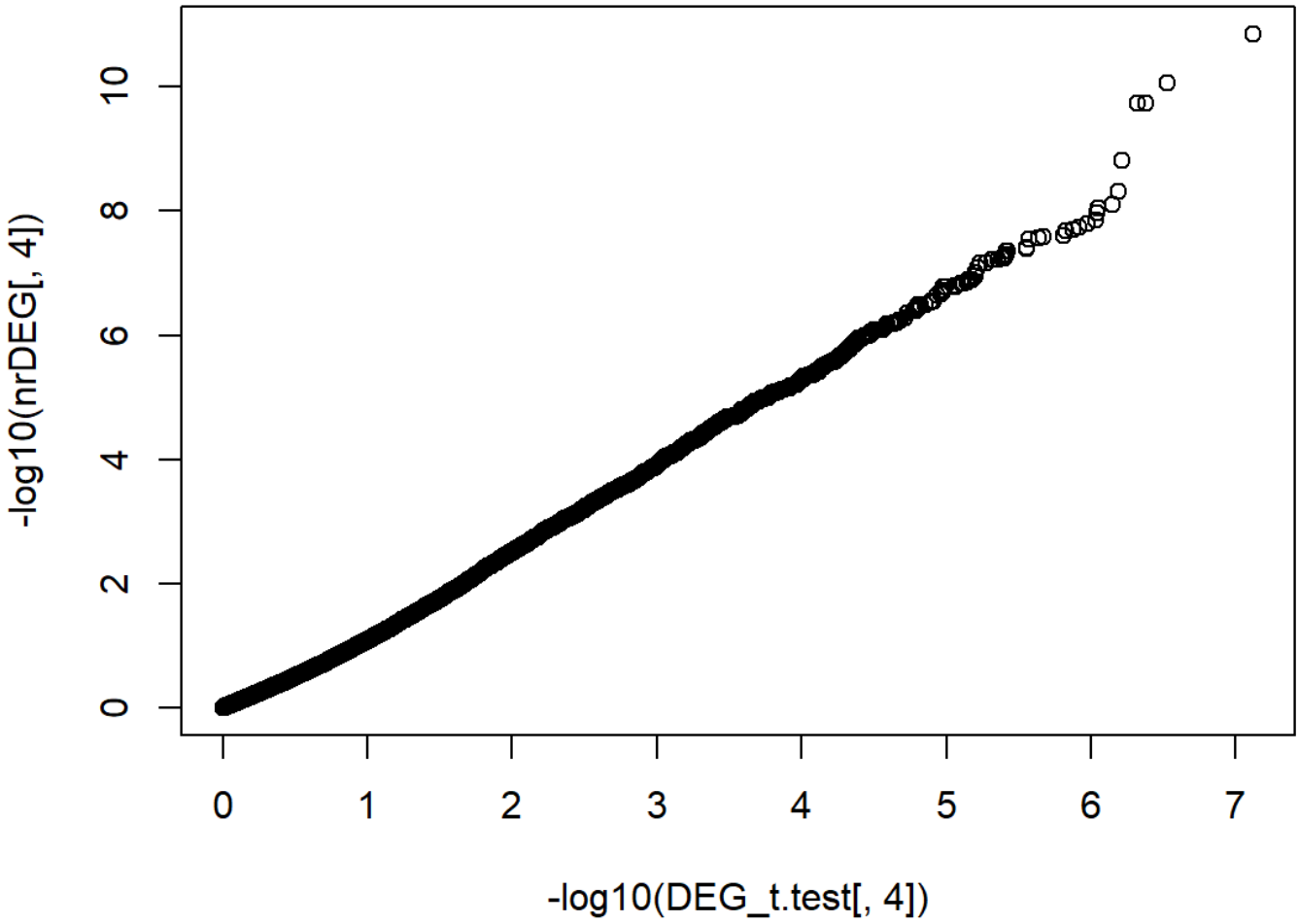
- 红点和蓝点非常多，说明实验组和对照组之间的基因表达谱存在巨大差异
- 图的最左端有一些非常高的红点。这些基因不仅上调幅度非常大（`logFC`值很高），而且统计显著性极强（`-log10(P值)`很高），是最值得关注的候选基因，可能是驱动疾病发生的关键基因。同样，最左端的一些非常高的蓝点是显著下调的关键基因
- 筛选关键基因：根据 `logFC` 和 `P值` 排序，挑出变化最显著的前几十个基因，作为后续研究的重点目标
- 功能富集分析 (GO/KEGG Analysis)：将所有的红色基因（上调解）和蓝色基因（下调集）分别拿去进行功能分析



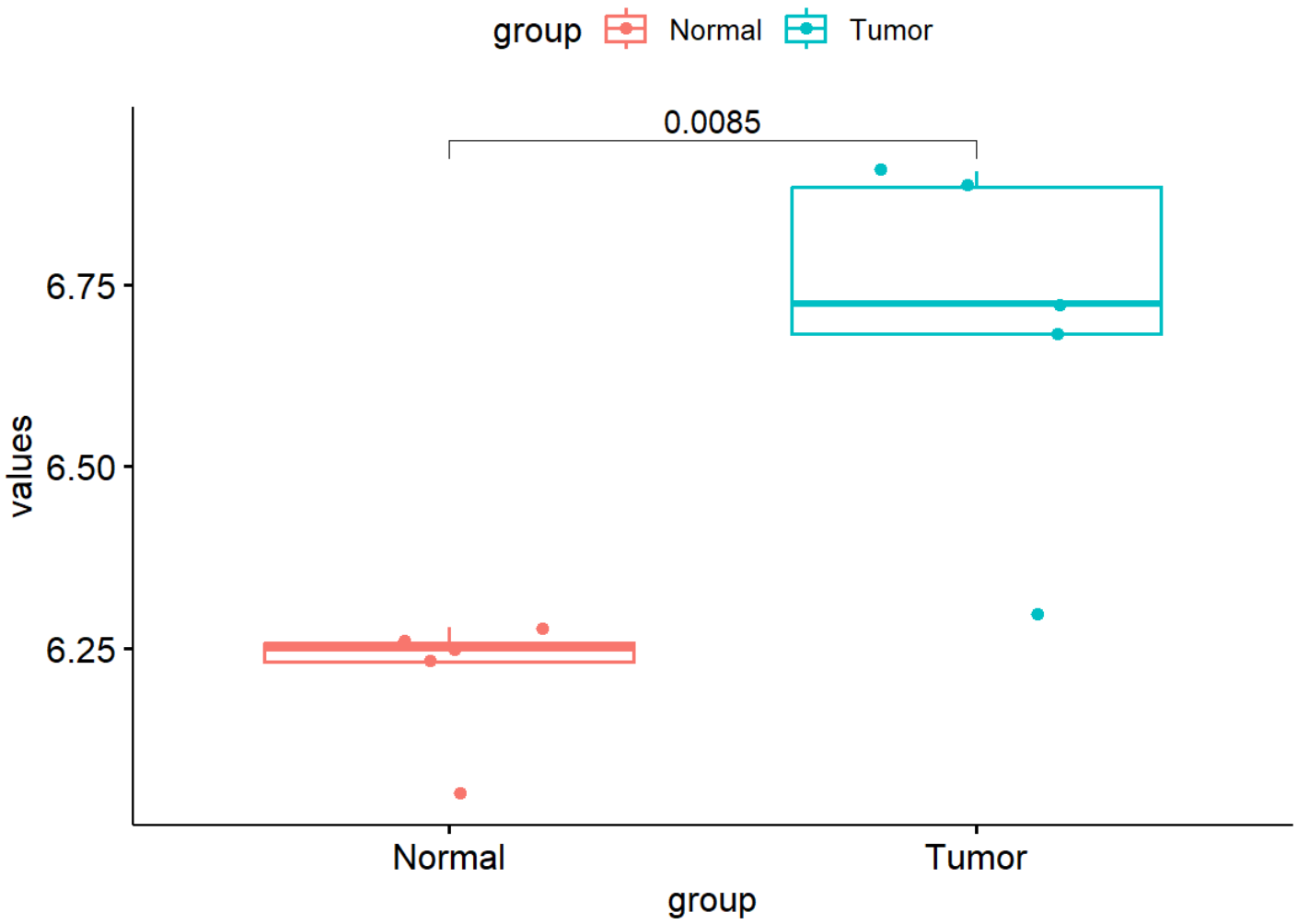
- 横轴是t检验的logFC值（`DEG_t.test[, 3]`），纵轴是limma的logFC值（`nrDEG[, 1]`）
- 数据点密集分布在中心区域，但也有一些分散点。整体上，点沿对角线分布，表明两种方法的logFC值有正相关关系。
- 然而，部分点偏离对角线，说明在某些基因上，两种方法计算的logFC值存在幅度或符号上的差异。这可能是因为t检验直接基于组间均值差，而limma使用了线性模型和方差收缩，更稳健于小样本数据。



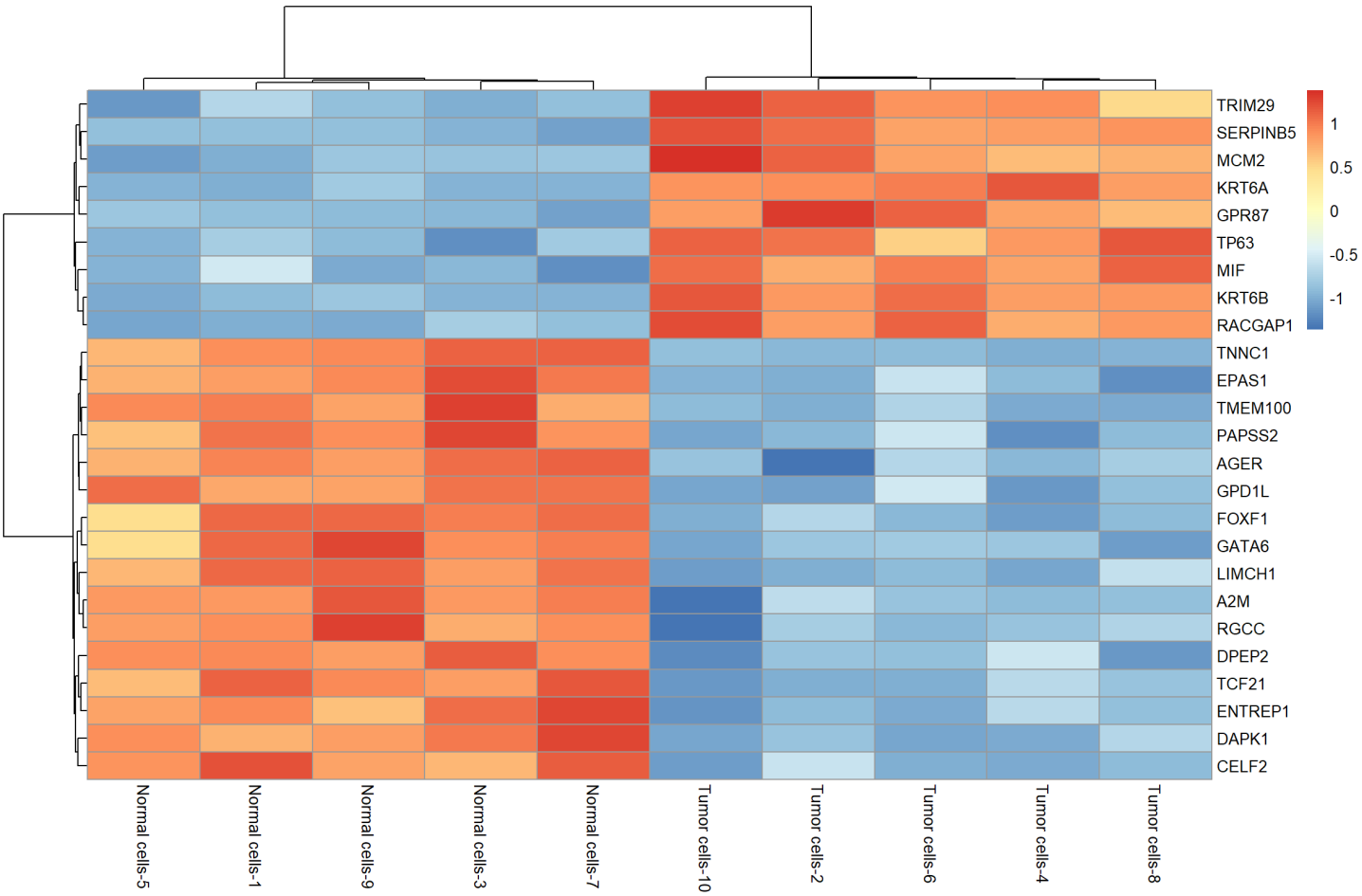
- 横轴是t检验的p值（`DEG_t.test[, 4]`），纵轴是limma的p值（`nrDEG[, 4]`）。
- 数据点沿对角线密集分布，再次确认p值的强正相关。limma的p值可能经过经验贝叶斯调整，更保守（即p值略大），但整体趋势一致。



- 横轴是t检验p值的负对数（`-log10(DEG_t.test[, 4])`），纵轴是limma p值的负对数（`-log10(nrDEG[, 4])`）。
- 数据点从左下角向右上角延伸，呈现明显的正相关趋势。这表明两种方法在识别显著基因方面高度一致：p值小的基因在两种方法中都显著，p值大的基因都不显著。



- **Tumor组 (蓝绿色)**：基因表达量的中位数**更高**，集中在 **6.75** 左右。
- **Normal组 (红粉色)**：基因表达量的中位数**较低**，集中在 **6.25** 附近。
- 结论：DLEU1基因在肿瘤组织中的表达显著高于正常组织。
- **P值 = 0.0085，**说明我们**有非常充分的统计学证据**拒绝“两组表达无差异”的原假设。也就是说，Normal组和Tumor组之间DLEU1表达水平的差异**不是由随机误差造成的**，而是具有统计学意义的真实差异。
- **生物学意义：DLEU1** 不是一个普通的基因，它是一个**长链非编码RNA (lncRNA)**。根据已有的研究，**DLEU1被广泛认为是一个原癌基因**，其在多种癌症（如慢性淋巴细胞白血病、肝癌、胃癌等）中高表达，并通过调控细胞周期、抑制凋亡等机制来促进肿瘤细胞的增殖和存活。**DLEU1在肿瘤细胞中的高表达**这与已知的生物学知识和文献报道是完全一致的，这是一个显著上调的基因



- **Y轴（纵向）**：是您从limma结果中筛选出的**前25个最显著差异基因**（p值最小的基因）。
- **X轴（横向）**：是所有分析的**样本**。
- **颜色**：
  - **红色**：表示**高表达**（高于该基因在所有样本中的平均表达水平）。
  - **蓝色**：表示**低表达**（低于该基因在所有样本中的平均表达水平）。
- **树状图（Dendrogram）**：
  - **顶部（样本聚类）**：显示了样本之间的相似性。靠得近的样本，其整体基因表达谱也更相似。
  - **左侧（基因聚类）**：显示了基因之间的相似性。靠得近的基因，其在样本中的表达变化模式也更相似（即共表达）。
- **生物学意义**
  - 样本聚类显示出明显的分组趋势：这意味着这25个基因的表达模式如此不同，以至于光靠这25个基因就能很好地区分肿瘤和正常组织
  - 左侧的基因聚类树状图将基因分成了几个大的簇。**通常，功能相似的基因会聚集在一起。**
  - **右上角（红块）**：这里聚集的是一组在**肿瘤样本中特异性高表达**的基因。根据您之前的limma结果，这几乎肯定包括了 **KRT6A, KRT6B** 等基因。这些是潜在的**原癌基因**。
  - **左下角（蓝块）**：这里聚集的是一组在**肿瘤样本中特异性低表达**的基因。这些是潜在的**抑癌基因**。

- 那些在肿瘤样本中呈现**最亮红色**（最高表达）的基因，是推动肿瘤发生发展的最核心候选基因。**KRT6A**和**KRT6B**就是典型的例子，它们是著名的癌基因，在多种上皮源性肿瘤中高表达。
- 同样，那些在肿瘤样本中呈现**最深蓝色**（最低表达）的基因，可能是功能缺失的抑癌基因。
- 下一步可进行功能富集分析和实验验证，如果包含患者的临床数据，则可以开展生存分析