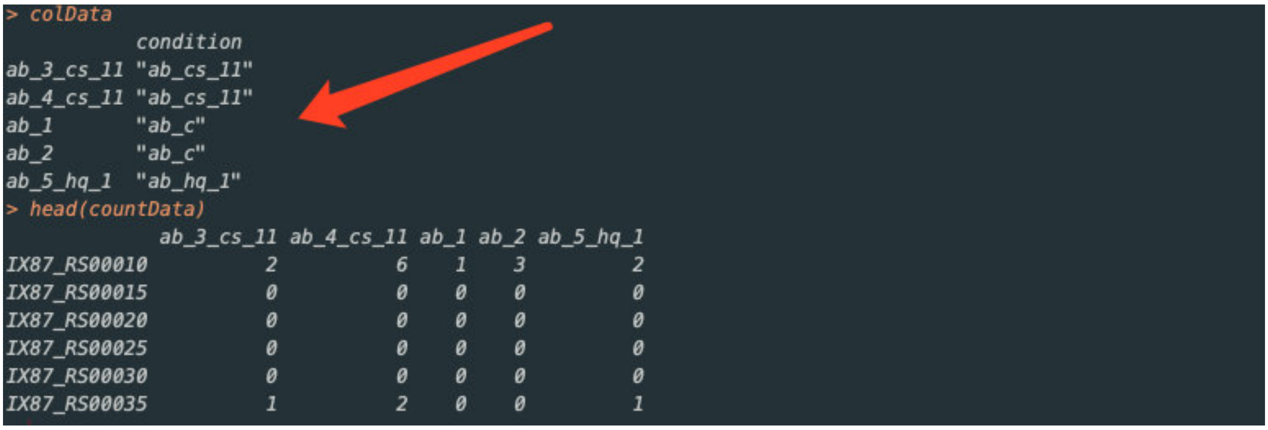
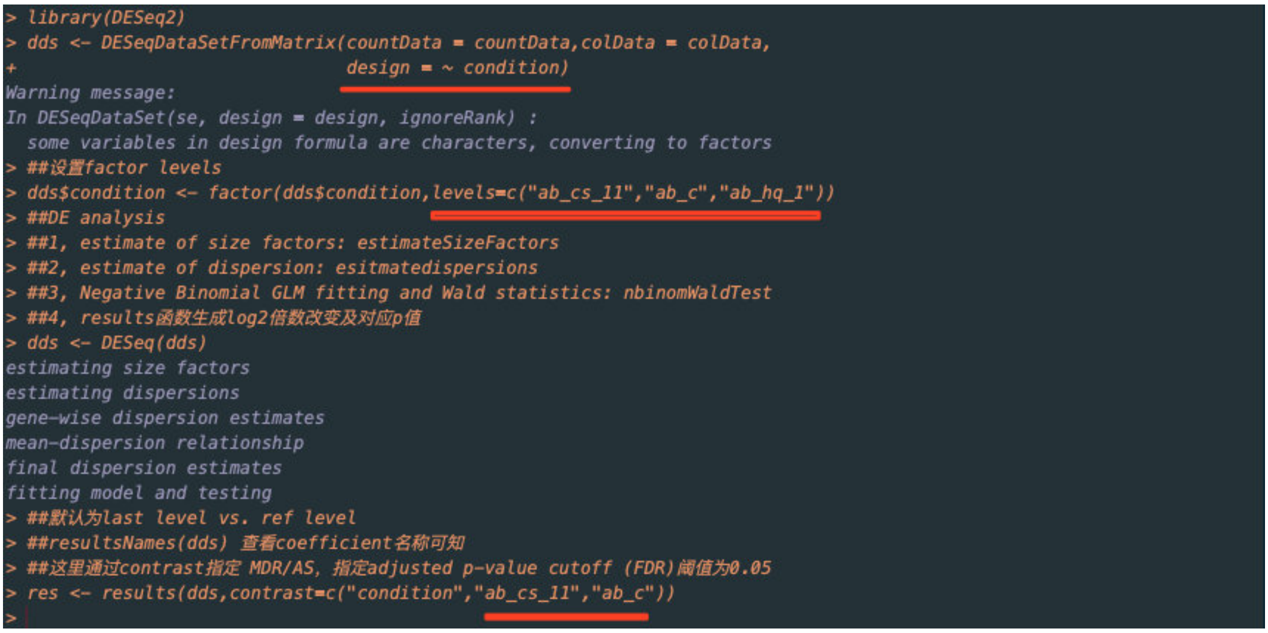
1. 参杂无关样本的差异分析

对'ab_cs_11'和'ab_c'两组测序数据做比较，使用DESeq2寻找差异基因，第一次分组时，将一个与本分析无关样本(分组为'ab_hg_1')加入`colData`和`countData`中分析，数据准备过程大同小异，这里仅展示相关步骤



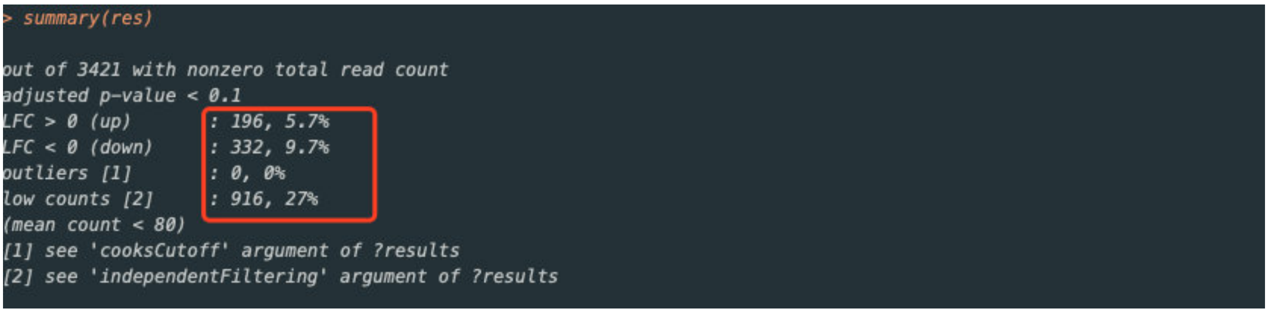
```
> colData
            condition
ab_3_cs_11 "ab_cs_11"
ab_4_cs_11 "ab_cs_11"
ab_1       "ab_c"
ab_2       "ab_c"
ab_5_hq_1  "ab_hq_1"
> head(countData)
              ab_3_cs_11 ab_4_cs_11 ab_1 ab_2 ab_5_hq_1
IX87_RS00010          2          6    1    3         2
IX87_RS00015          0          0    0    0         0
IX87_RS00020          0          0    0    0         0
IX87_RS00025          0          0    0    0         0
IX87_RS00030          0          0    0    0         0
IX87_RS00035          1          2    0    0         1
```

根据'condition'列设置`design`参数，同时设置比较组('ab_cs_11 vs ab_c')，并获得差异检出结果



```
> library(DESeq2)
> dds <- DESeqDataSetFromMatrix(countData = countData,colData = colData,
+                               design = ~ condition)
Warning message:
In DESeqDataSet(se, design = design, ignoreRank) :
  some variables in design formula are characters, converting to factors
> ##设置factor levels
> dds$condition <- factor(dds$condition,levels=c("ab_cs_11","ab_c","ab_hq_1"))
> ##DE analysis
> ##1, estimate of size factors: estimateSizeFactors
> ##2, estimate of dispersion: esitmatedispersions
> ##3, Negative Binomial GLM fitting and Wald statistics: nbinomWaldTest
> ##4, results函数生成log2倍数改变及对应p值
> dds <- DESeq(dds)
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
fitting model and testing
> ##默认为last level vs. ref level
> ##resultsNames(dds) 查看coefficient名称可知
> ##这里通过contrast指定 MDR/AS, 指定adjusted p-value cutoff (FDR)阈值为0.05
> res <- results(dds,contrast=c("condition","ab_cs_11","ab_c"))
>
```

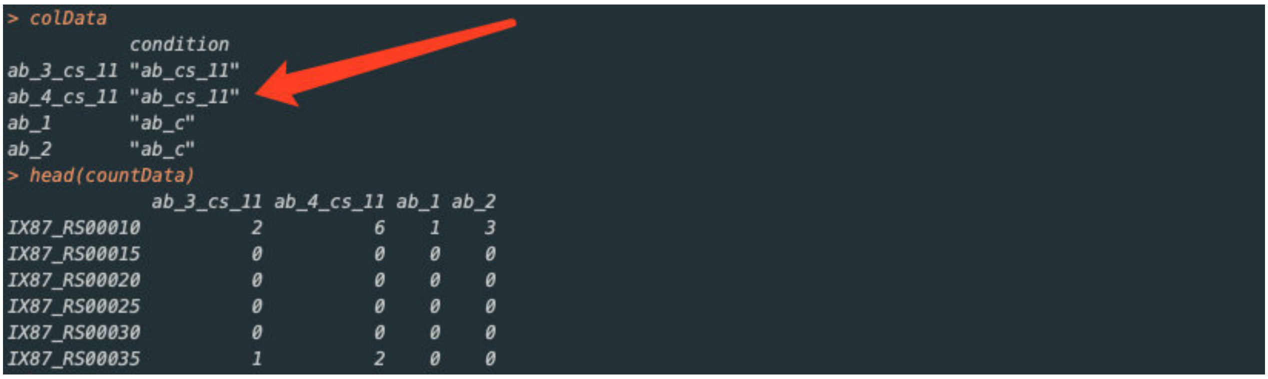
查看差异结果



```
> summary(res)

out of 3421 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 196, 5.7%
LFC < 0 (down)     : 332, 9.7%
outliers [1]       : 0, 0%
low counts [2]     : 916, 27%
(mean count < 80)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

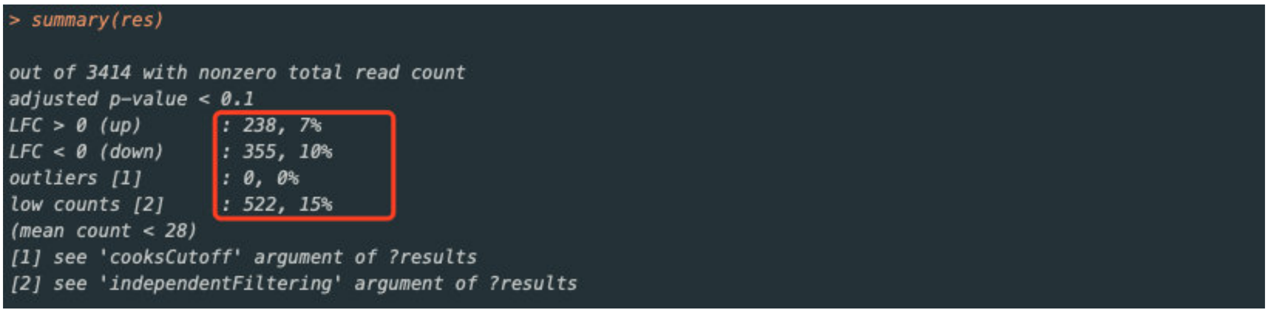
2. 去除无关样本后的分析

调整`colData`和`countData`参数，剔除无关样本('ab_hq_1')，其他分析步骤同上



```
> colData
           condition
ab_3_cs_11 "ab_cs_11"
ab_4_cs_11 "ab_cs_11"
ab_1       "ab_c"
ab_2       "ab_c"
> head(countData)
             ab_3_cs_11 ab_4_cs_11 ab_1 ab_2
IX87_RS00010          2          6    1    3
IX87_RS00015          0          0    0    0
IX87_RS00020          0          0    0    0
IX87_RS00025          0          0    0    0
IX87_RS00030          0          0    0    0
IX87_RS00035          1          2    0    0
```

根据'condition'列设置`design`参数，同时设置比较组('ab_cs_11 vs ab_c')，并获得差异检出结果；查看差异结果



```
> summary(res)

out of 3414 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)       : 238, 7%
LFC < 0 (down)     : 355, 10%
outliers [1]       : 0, 0%
low counts [2]     : 522, 15%
(mean count < 28)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

3. 比较两次`summary(res)`结果

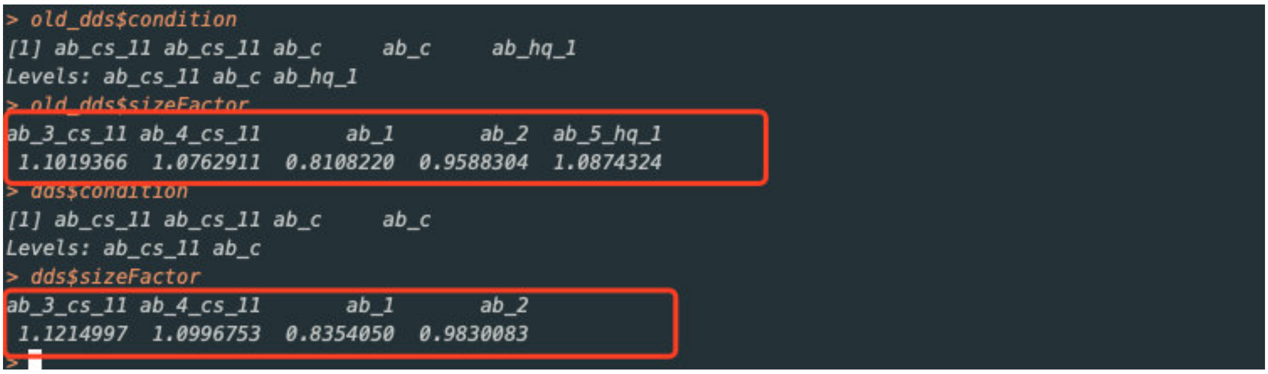
   第一次含有无关样本分析时: LFC > 0,196, 5.7% LFC < 0, 332, 9.7%

   第二次不含无关样本分析时: LFC > 0, 238, 7% LFC < 0, 355, 10%

可以发现同在`adjusted p-value < 0.1`时，使用DESeq2对相同两组样本检测了不同数目差异基因，难道比较组以外的样本的存在会影响比较组的差异检出???

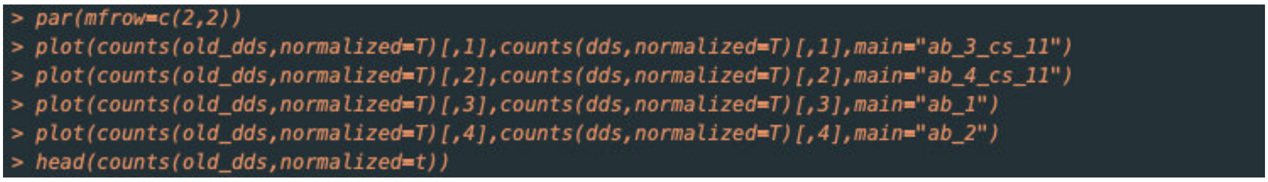
这里使用'old_res'/'old_dds'代表含无关样本比较结果; 'res'/'old_dds'代表不含无关样本比较结果

首先查看两种情况下的`sizeFactor`，可见无关样本缺失对`normalization`过程带来影响



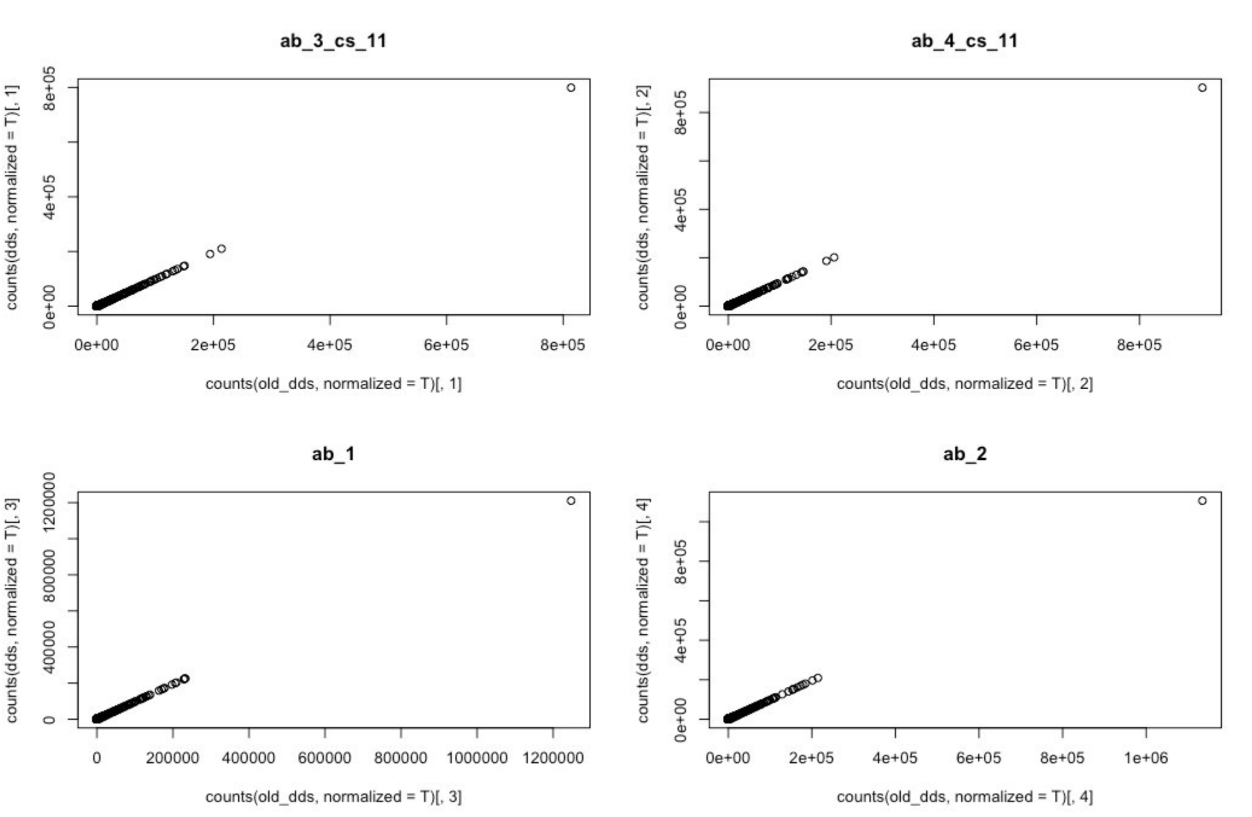
```
> old_dds$condition
[1] ab_cs_11 ab_cs_11 ab_c     ab_c     ab_hq_1
Levels: ab_cs_11 ab_c ab_hq_1
> old_dds$sizeFactor
ab_3_cs_11 ab_4_cs_11       ab_1       ab_2   ab_5_hq_1
 1.1019366  1.0762911  0.8108220  0.9588304  1.0874324
> dds$condition
[1] ab_cs_11 ab_cs_11 ab_c     ab_c
Levels: ab_cs_11 ab_c
> dds$sizeFactor
ab_3_cs_11 ab_4_cs_11       ab_1       ab_2
 1.1214997  1.0996753  0.8354050  0.9830083
>
```

其次查看`normalization`后counts分布是否存在差异

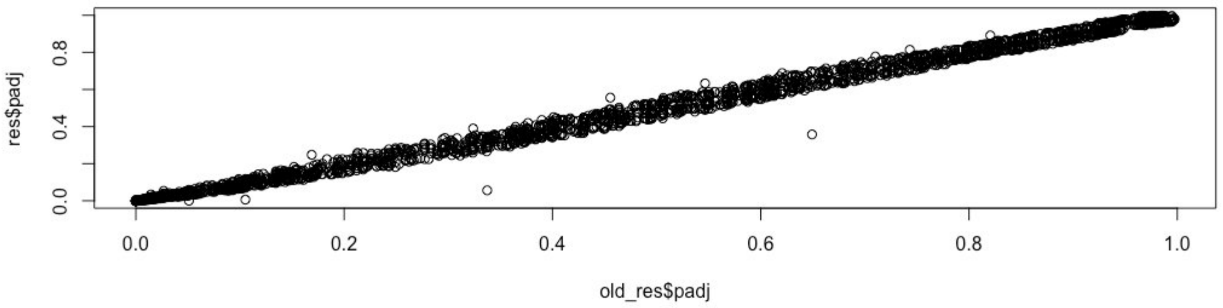


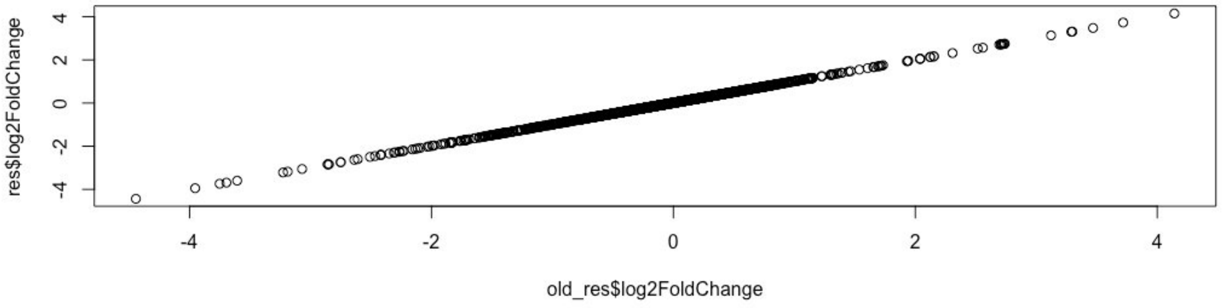
```
> par(mfrow=c(2,2))
> plot(counts(old_dds,normalized=T)[,1],counts(dds,normalized=T)[,1],main="ab_3_cs_11")
> plot(counts(old_dds,normalized=T)[,2],counts(dds,normalized=T)[,2],main="ab_4_cs_11")
> plot(counts(old_dds,normalized=T)[,3],counts(dds,normalized=T)[,3],main="ab_1")
> plot(counts(old_dds,normalized=T)[,4],counts(dds,normalized=T)[,4],main="ab_2")
> head(counts(old_dds,normalized=t))
```

如图, 未出现显著差异

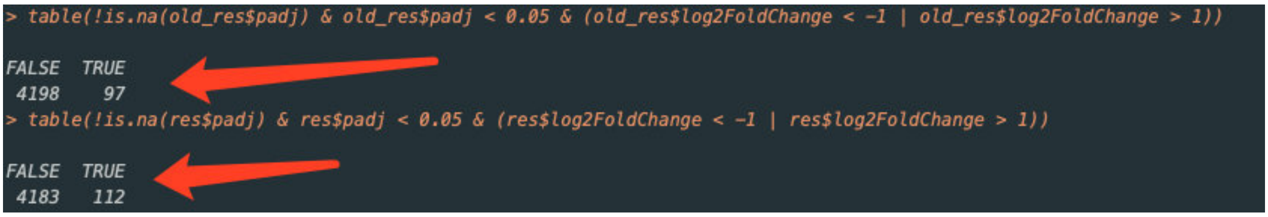


再次比较检查后的'adjusted p-value'和'log2foldchange'





趋势这么一致为何summary存在差异呢, 再次查看在'adjusted p-value < 0.05'时的分布情况

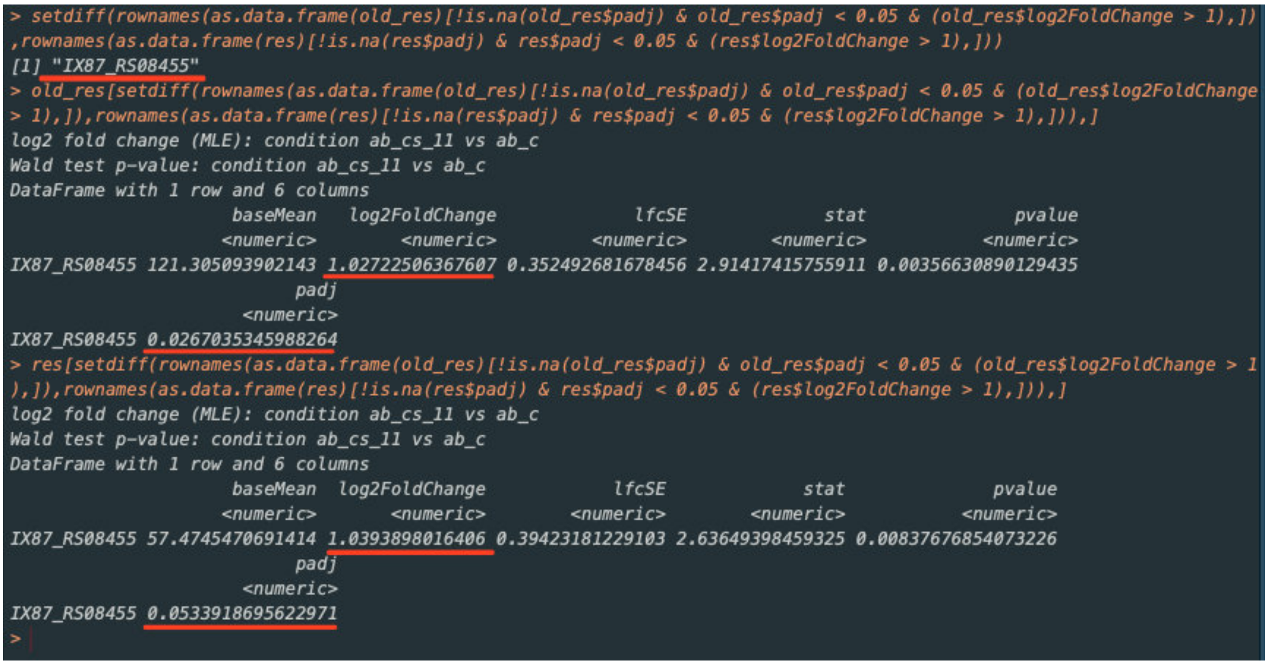


```
> table(!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange < -1 | old_res$log2FoldChange > 1))

FALSE  TRUE 
 4198    97 
> table(!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange < -1 | res$log2FoldChange > 1))

FALSE  TRUE 
 4183   112 
```

接着在满足'log2FoldChange' > 1/ < -1, 同时'pad < 0.05'条件下的差异基因分布

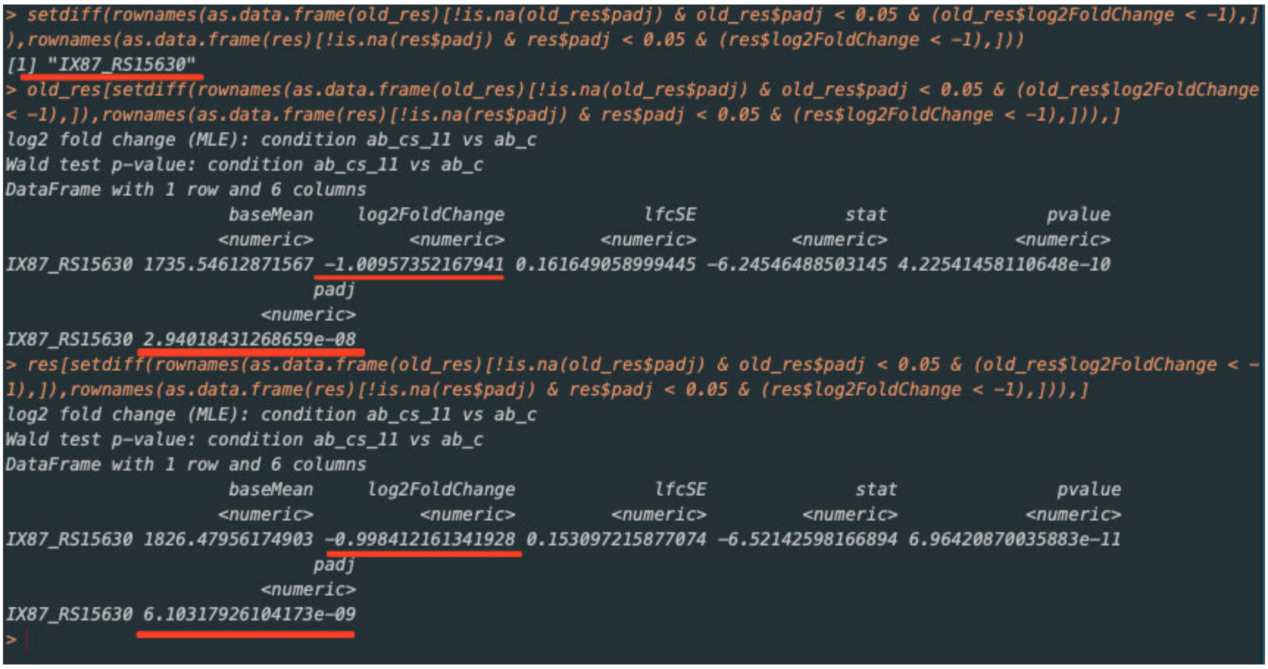
查看'old_res'独有差异基因

在'log2FoldChange >1'时, 'old_res'中独有的基因在'res'中情况, 可见'log2FoldChange'都差不多，就是res中对应'padj'值稍微大于0.05



```
> setdiff(rownames(as.data.frame(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange > 1),])
,rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange > 1),]))
[1] "IX87_RS08455"
> old_res[setdiff(rownames(as.data.frame(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange
> 1),]),rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange > 1),])),]
log2 fold change (MLE): condition ab_cs_11 vs ab_c 
Wald test p-value: condition ab_cs_11 vs ab_c 
DataFrame with 1 row and 6 columns
                     baseMean   log2FoldChange             lfcSE              stat                pvalue
                    <numeric>        <numeric>         <numeric>         <numeric>             <numeric>
IX87_RS08455 121.305093902143 1.02722506367607 0.352492681678456 2.91417415755911 0.00356630890129435
                           padj
                      <numeric>
IX87_RS08455 0.0267035345988264
> res[setdiff(rownames(as.data.frame(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange > 1
),]),rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange > 1),])),]
log2 fold change (MLE): condition ab_cs_11 vs ab_c 
Wald test p-value: condition ab_cs_11 vs ab_c 
DataFrame with 1 row and 6 columns
                     baseMean  log2FoldChange            lfcSE              stat                pvalue
                    <numeric>       <numeric>        <numeric>         <numeric>             <numeric>
IX87_RS08455 57.4745470691414 1.0393898016406 0.39423181229103 2.63649398459325 0.00837676854073226
                           padj
                      <numeric>
IX87_RS08455 0.0533918695622971
> 
```

而在'log2FoldChange < -1'时, 'old_res'独有的基因在'res'中情况, 可见'padj'均满足<0.05, 只是'res'中对应基因'log2FoldChange'稍微大于-1

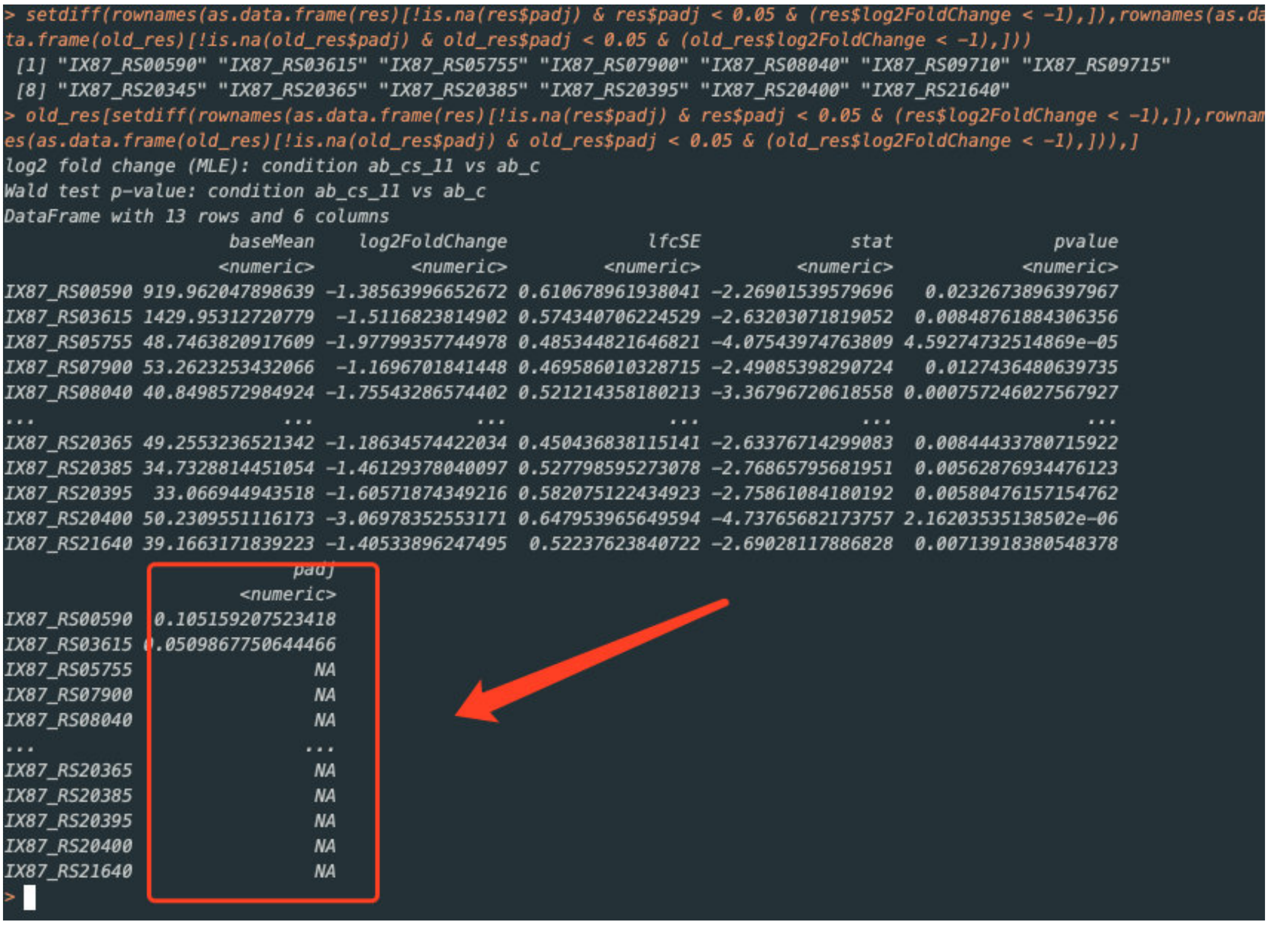


```
> setdiff(rownames(as.data.frame(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange < -1),]
),rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange < -1),]))
[1] "IX87_RS15630"
> old_res[setdiff(rownames(as.data.frame(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange
< -1),]),rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange < -1),])),]
log2 fold change (MLE): condition ab_cs_11 vs ab_c 
Wald test p-value: condition ab_cs_11 vs ab_c 
DataFrame with 1 row and 6 columns
                     baseMean    log2FoldChange             lfcSE              stat               pvalue
                    <numeric>         <numeric>         <numeric>         <numeric>            <numeric>
IX87_RS15630 1735.54612871567 -1.00957352167941 0.161649058999445 -6.24546488503145 4.22541458110648e-10
                          padj
                     <numeric>
IX87_RS15630 2.94018431268659e-08
> res[setdiff(rownames(as.data.frame(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange < -
1),]),rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange < -1),])),]
log2 fold change (MLE): condition ab_cs_11 vs ab_c 
Wald test p-value: condition ab_cs_11 vs ab_c 
DataFrame with 1 row and 6 columns
                     baseMean     log2FoldChange             lfcSE              stat               pvalue
                    <numeric>          <numeric>         <numeric>         <numeric>            <numeric>
IX87_RS15630 1826.47956174903 -0.998412161341928 0.153097215877074 -6.52142598166894 6.96420870035883e-11
                          padj
                     <numeric>
IX87_RS15630 6.10317926104173e-09
> 
```

查看'res'独有差异基因

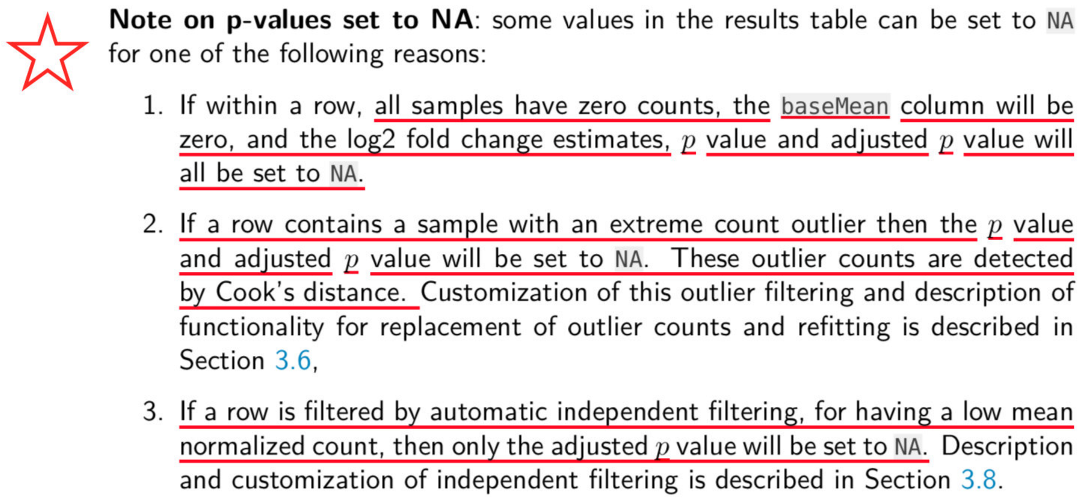
在'log2FoldChange >1'时, 'res'中独有的基因在‘res’中情况, 可见存在两种情况, 'log2FoldChange'小于或padj为NA

在'log2FoldChange < -1'时, 'res'中独有的基因在‘res’中情况, 可见存在两种情况, 'log2FoldChange'小于或padj为NA(图片太大, 仅展示‘res'中独有基因在'old_res'照中的情况)



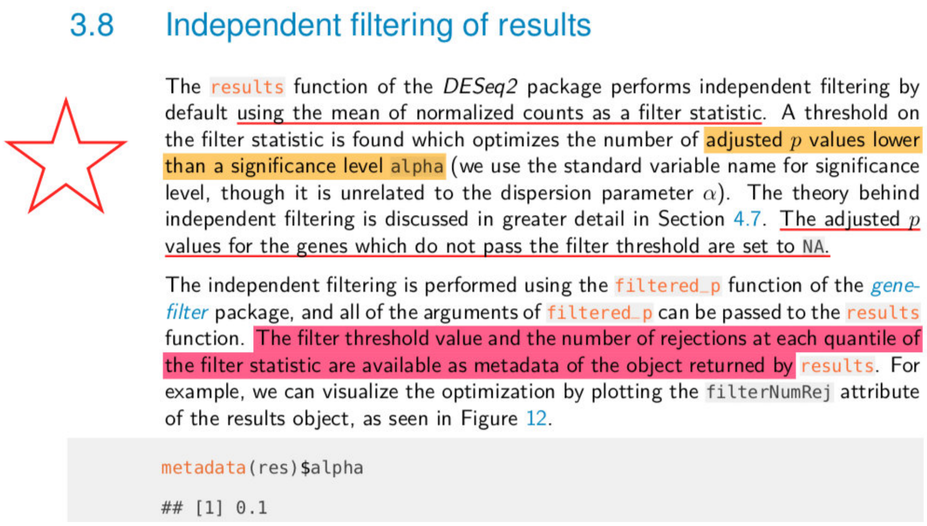
```
> setdiff(rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange < -1),]),rownames(as.da
ta.frame(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange < -1),]))
 [1] "IX87_RS00590" "IX87_RS03615" "IX87_RS05755" "IX87_RS07900" "IX87_RS08040" "IX87_RS09710" "IX87_RS09715"
 [8] "IX87_RS20345" "IX87_RS20365" "IX87_RS20385" "IX87_RS20395" "IX87_RS20400" "IX87_RS21640"
> old_res[setdiff(rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange < -1),]),rownam
es(as.data.frame(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange < -1),])),]
log2 fold change (MLE): condition ab_cs_11 vs ab_c
Wald test p-value: condition ab_cs_11 vs ab_c
DataFrame with 13 rows and 6 columns
                     baseMean    log2FoldChange             lfcSE              stat               pvalue
                    <numeric>         <numeric>         <numeric>         <numeric>            <numeric>
IX87_RS00590 919.962047898639 -1.38563996652672 0.610678961938041 -2.26901539579696   0.0232673896397967
IX87_RS03615 1429.95312720779  -1.5116823814902 0.574340706224529 -2.63203071819052  0.00848761884306356
IX87_RS05755 48.7463820917609 -1.97799357744978 0.485344821646821 -4.07543974763809 4.59274732514869e-05
IX87_RS07900 53.2623253432066  -1.1696701841448 0.469586010328715 -2.49085398290724   0.0127436480639735
IX87_RS08040 40.8498572984924 -1.75543286574402 0.521214358180213 -3.36796720618558 0.000757246027567927
...                       ...               ...               ...               ...                  ...
IX87_RS20365 49.2553236521342 -1.18634574422034 0.450436838115141 -2.63376714299083  0.00844433780715922
IX87_RS20385 34.7328814451054 -1.46129378040097 0.527798595273078 -2.76865795681951  0.00562876934476123
IX87_RS20395  33.066944943518 -1.60571874349216 0.582075122434923 -2.75861084180192  0.00580476157154762
IX87_RS20400 50.2309551116173 -3.06978352553171 0.647953965649594 -4.73765682173757 2.16203535138502e-06
IX87_RS21640 39.1663171839223 -1.40533896247495  0.52237623840722 -2.69028117886828  0.00713918380548378
                          padj
                     <numeric>
IX87_RS00590 0.105159207523418
IX87_RS03615 0.0509867750644466
IX87_RS05755                NA
IX87_RS07900                NA
IX87_RS08040                NA
...                        ...
IX87_RS20365                NA
IX87_RS20385                NA
IX87_RS20395                NA
IX87_RS20400                NA
IX87_RS21640                NA
> 
```

因此, 构建'colData'和'countData'的不同影响固定阈值下检出差异基因的不同, 存在有三情况, 前两种是由于我们所选择的硬性阈值导致的, 这个可以理解, 在做'normalization'时, 数据结构的不同将导致数据微小的偏差; 最后一种是由于‘padj‘为‘NA'导致, 查看'padj'为‘NA‘的软件解释:



Note on p-values set to NA: some values in the results table can be set to NA for one of the following reasons:

1. If within a row, all samples have zero counts, the baseMean column will be zero, and the log2 fold change estimates, p value and adjusted p value will all be set to NA.
2. If a row contains a sample with an extreme count outlier then the p value and adjusted p value will be set to NA. These outlier counts are detected by Cook's distance. Customization of this outlier filtering and description of functionality for replacement of outlier counts and refitting is described in Section 3.6,
3. If a row is filtered by automatic independent filtering, for having a low mean normalized count, then only the adjusted p value will be set to NA. Description and customization of independent filtering is described in Section 3.8.



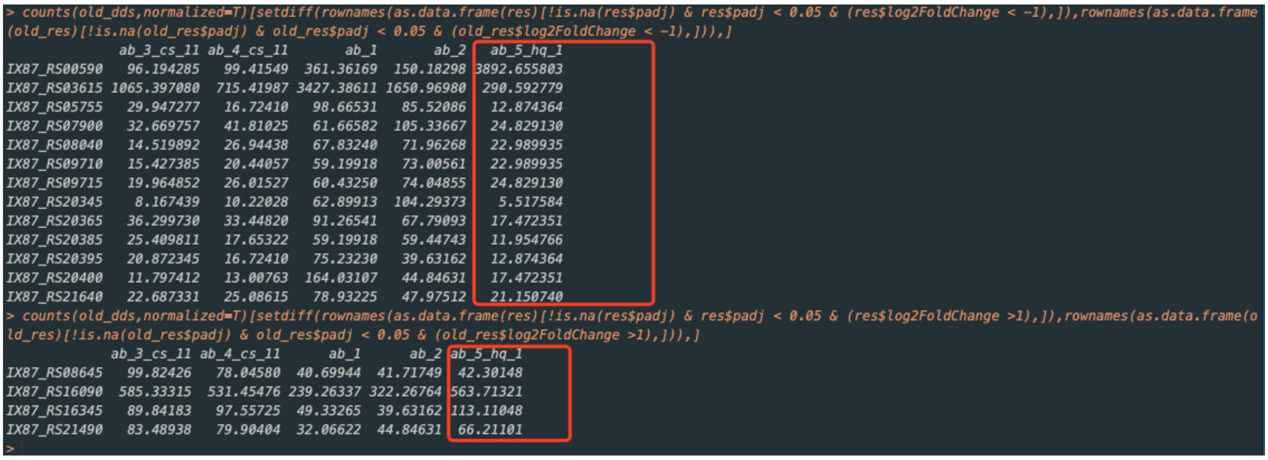
3.8 Independent filtering of results

The results function of the *DESeq2* package performs independent filtering by default using the mean of normalized counts as a filter statistic. A threshold on the filter statistic is found which optimizes the number of adjusted p values lower than a significance level alpha (we use the standard variable name for significance level, though it is unrelated to the dispersion parameter α). The theory behind independent filtering is discussed in greater detail in Section 4.7. The adjusted p values for the genes which do not pass the filter threshold are set to NA.

The independent filtering is performed using the filtered_p function of the *genefilter* package, and all of the arguments of filtered_p can be passed to the results function. The filter threshold value and the number of rejections at each quantile of the filter statistic are available as metadata of the object returned by results. For example, we can visualize the optimization by plotting the filterNumRej attribute of the results object, as seen in Figure 12.

```
metadata(res)$alpha

## [1] 0.1
```

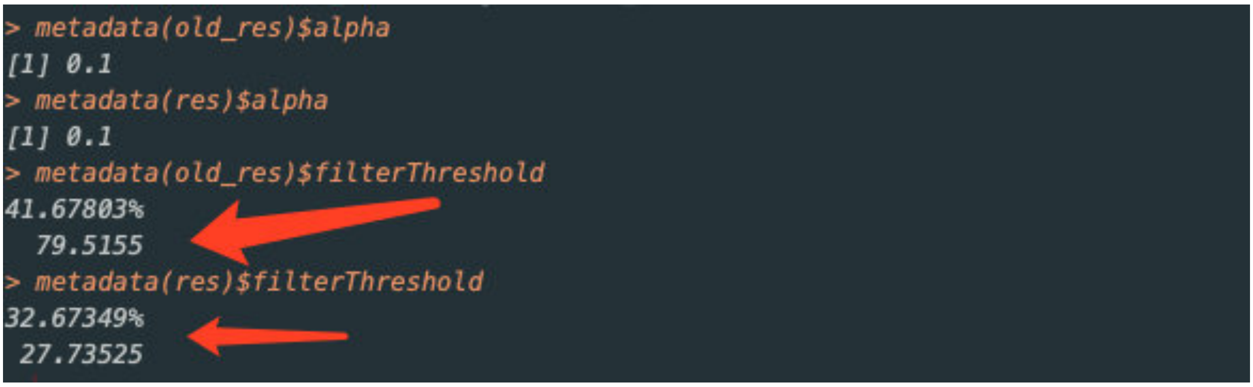
那么, 这里带来的'NA'应该是'ab_hg_1'样本导致的, 且是由于第三种情况, 查看:



```
> counts(old_dds,normalized=T)[setdiff(rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange < -1),]),rownames(as.data.frame
(old_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange < -1),])),]
             ab_3_cs_11 ab_4_cs_11        ab_1        ab_2   ab_5_hq_1
IX87_RS00590   96.194285   99.41549  361.36169  150.18298 3892.655803
IX87_RS03615 1065.397080  715.41987 3427.38611 1650.96980  290.592779
IX87_RS05755   29.947277   16.72410   98.66531   85.52086   12.874364
IX87_RS07900   32.669757   41.81025   61.66582  105.33667   24.829130
IX87_RS08040   14.519892   26.94438   67.83240   71.96268   22.989935
IX87_RS09710   15.427385   20.44057   59.19918   73.00561   22.989935
IX87_RS09715   19.964852   26.01527   60.43250   74.04855   24.829130
IX87_RS20345    8.167439   10.22028   62.89913  104.29373    5.517584
IX87_RS20365   36.299730   33.44820   91.26541   67.79093   17.472351
IX87_RS20385   25.409811   17.65322   59.19918   59.44743   11.954766
IX87_RS20395   20.872345   16.72410   75.23230   39.63162   12.874364
IX87_RS20400   11.797412   13.00763  164.03107   44.84631   17.472351
IX87_RS21640   22.687331   25.08615   78.93225   47.97512   21.150740
> counts(old_dds,normalized=T)[setdiff(rownames(as.data.frame(res)[!is.na(res$padj) & res$padj < 0.05 & (res$log2FoldChange >1),]),rownames(as.data.frame(o
ld_res)[!is.na(old_res$padj) & old_res$padj < 0.05 & (old_res$log2FoldChange >1),])),]
            ab_3_cs_11 ab_4_cs_11      ab_1      ab_2 ab_5_hq_1
IX87_RS08645   99.82426   78.04580  40.69944  41.71749  42.30148
IX87_RS16090  585.33315  531.45476 239.26337 322.26764 563.71321
IX87_RS16345   89.84183   97.55725  49.33265  39.63162 113.11048
IX87_RS21490   83.48938   79.90404  32.06622  44.84631  66.21101
>
```

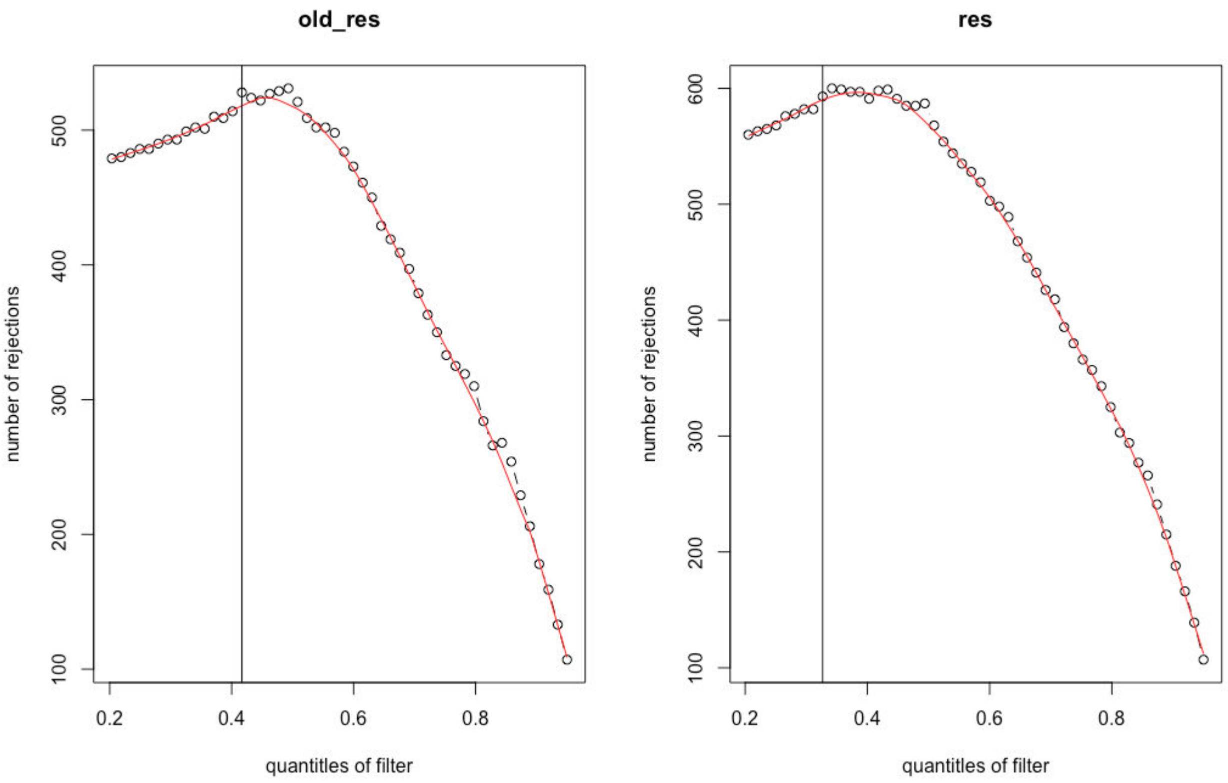
又根据软件解释其'independent filtering'是采用'genefilter'包的'filtered_p'函数

对应查看, 缺失存在差异:



```
> metadata(old_res)$alpha
[1] 0.1
> metadata(res)$alpha
[1] 0.1
> metadata(old_res)$filterThreshold
41.67803%
  79.5155
> metadata(res)$filterThreshold
32.67349%
 27.73525
```

根据软件代码(3.8 Independent filtering of results)解释绘图:



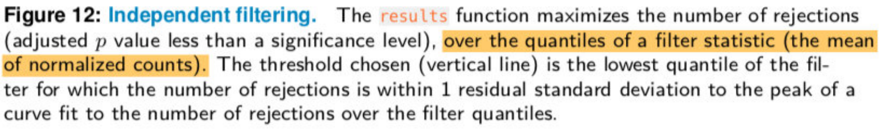
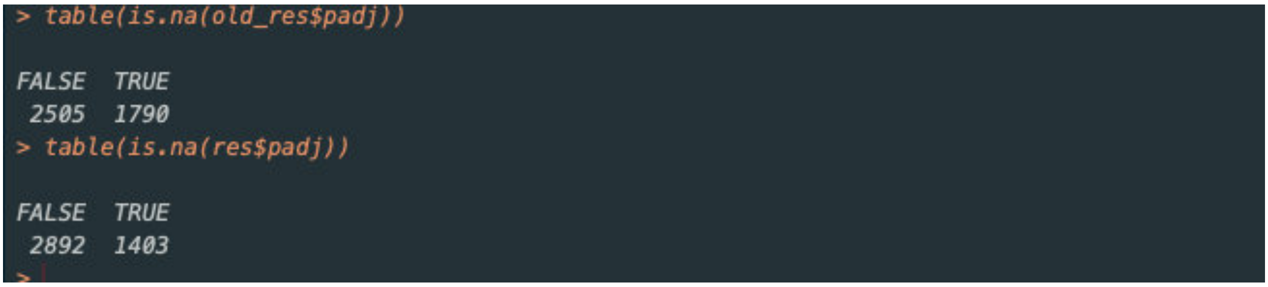


Figure 12: **Independent filtering.** The `results` function maximizes the number of rejections (adjusted p value less than a significance level), over the quantiles of a filter statistic (the mean of normalized counts). The threshold chosen (vertical line) is the lowest quantile of the filter for which the number of rejections is within 1 residual standard deviation to the peak of a curve fit to the number of rejections over the filter quantiles.

根据其解释尝试理解, “Independent filtering by default using the mean of normalized counts as a filter statistic. A threshold on the filter statistic is found which optimizes the number of adjusted p values lower than a significance leve alpha", 这里两次检出的'alpha'均为'0.1'.

那么个人理解就是, 'ab_hq_1'样本的存在改变了其‘independent filtering‘的阈值所导致的'NA', 可以是由其'ab_hq_1'样本本身, 也可以是其他4个样本所致.



```
> table(is.na(old_res$padj))

FALSE  TRUE 
 2505  1790 
> table(is.na(res$padj))

FALSE  TRUE 
 2892  1403 
> 
```

因此, 避免无关样本可以增加检出敏感度.