# RNA-seq 前处理

RNA-seq分析流程：

1. 质控：通过fastq文件查看测序质量，包括测序长度分布，测序准确性估计等；根据需要，去除低质量的数据；
2. 比对：将质控后的数据与参考基因组或者转录组数据比对，以确定它们在基因组上的位置；
3. 定量分析：根据比对后的结果对基因表达进行定量
4. 差异表达分析：比较不同条件或处理下的样本，以识别差异表达的基因。
5. 功能和通路分析：对差异表达的基因进行功能注释和通路分析，以了解它们在生物过程中的作用。

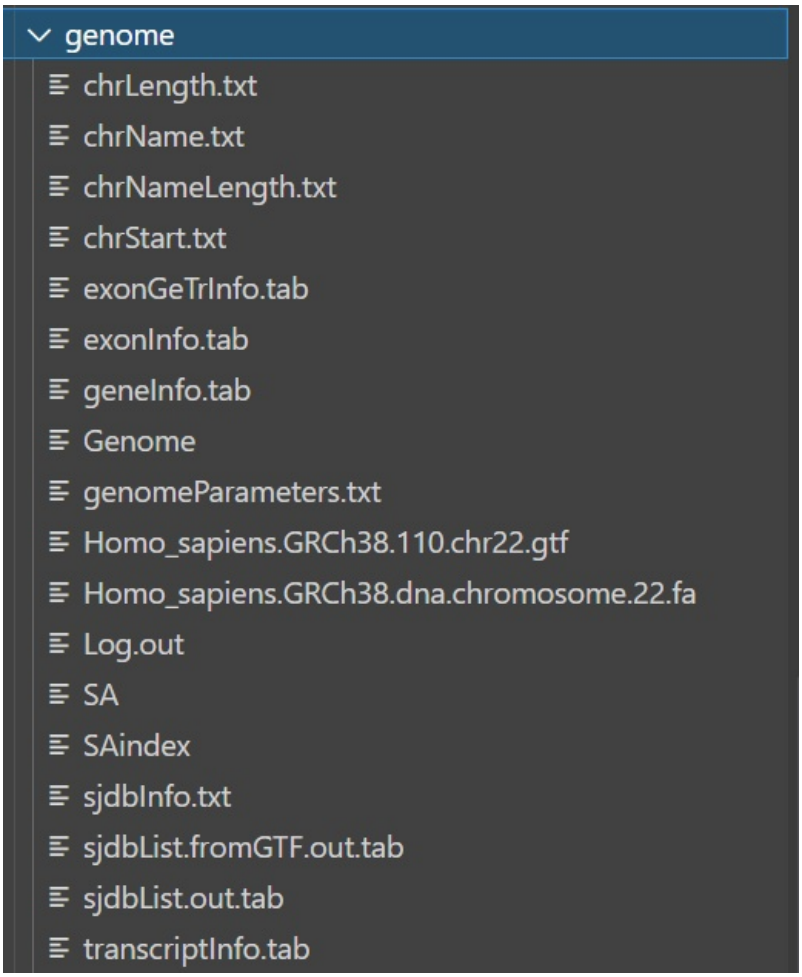
## *1, 2, 3 见 preprocess.sh*

### 代码截图

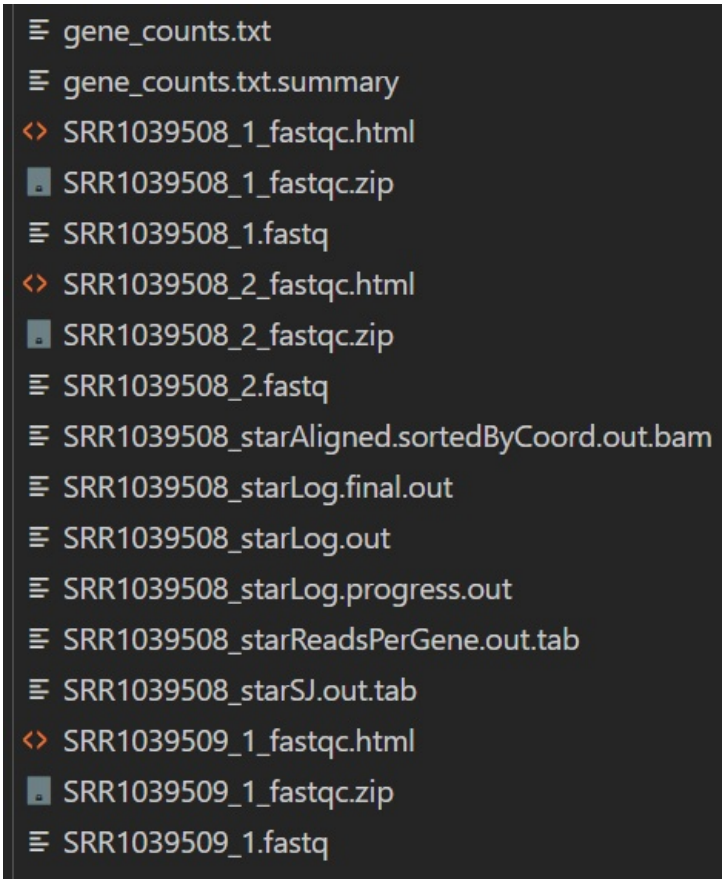
```
#!/bin/bash
# Usage: bash preprocess.sh


# read fastq files in fastq folder and output the row number, reads number and sequence length
# Define the directory
fastq_dir="$(dirname "$0")/data/fastq/"
log_file="$(dirname "$0")/data/fastq/data_overview.txt"

# Loop over each file in the directory
for file in "$fastq_dir"*.fastq
do
  # Extract the required information and print it
  echo $file >> $log_file
  awk '{if(NR==2) {printf "Seq length: %s\n", length($0)}}' $file >> $log_file
  echo "Row Number: $(awk 'END {print NR}' $file)" >> $log_file
  echo "Reads Number: $(awk 'END {print NR/4}' $file)" >> $log_file
done


# Loop over each unique pair of files in the directory
for file in "$fastq_dir"*_1.fastq
do
  # Get the base name without the _1.fastq suffix
  base_name="${file%_1.fastq}"

  # Define the names of the two files in the pair
  file1="${base_name}_1.fastq"
  file2="${base_name}_2.fastq"

  # Run the desired command on the pair of files
  fastqc $file1 $file2
done
```

```
# use STAR to align the reads to the genome
# Define the paths
genomeFastaFiles="$genomeDir/Homo_sapiens.GRCh38.dna.chromosome.22.fa"
sjdbGTFfile="$genomeDir/Homo_sapiens.GRCh38.110.chr22.gtf"
genomeDir="$(dirname "$0")/data/genome/"

# generate index
STAR --runMode genomeGenerate --genomeDir $genomeDir\
    --genomeFastaFiles $genomeFastaFiles --sjdbGTFfile $sjdbGTFfile --genomeSAindexNbases 11

# Run STAR on each file
for file in "$fastq_dir"*_1.fastq
do
  # Get the base name without the _1.fastq suffix
  base_name="${file%_1.fastq}"

  # Define the names of the two files in the pair
  file1="${base_name}_1.fastq"
  file2="${base_name}_2.fastq"
  STAR --runThreadN 12 --genomeDir $genomeDir --readFilesIn $file1 $file2\
  --outFileNamePrefix $base_name'_star' --outSAMtype BAM SortedByCoordinate --quantMode GeneCounts
done


# Generate quantitative output
# List BAM files in fastq_dir
bamfiles=$(ls $fastq_dir/*.bam)
# Pass BAM files as parameters to featureCounts
featureCounts -T 8 -a -p $genomeDir/Homo_sapiens.GRCh38.110.chr22.gtf \
              -o $fastq_dir/gene_counts.txt -g gene_id -s 1 $bamfiles
```

index结果

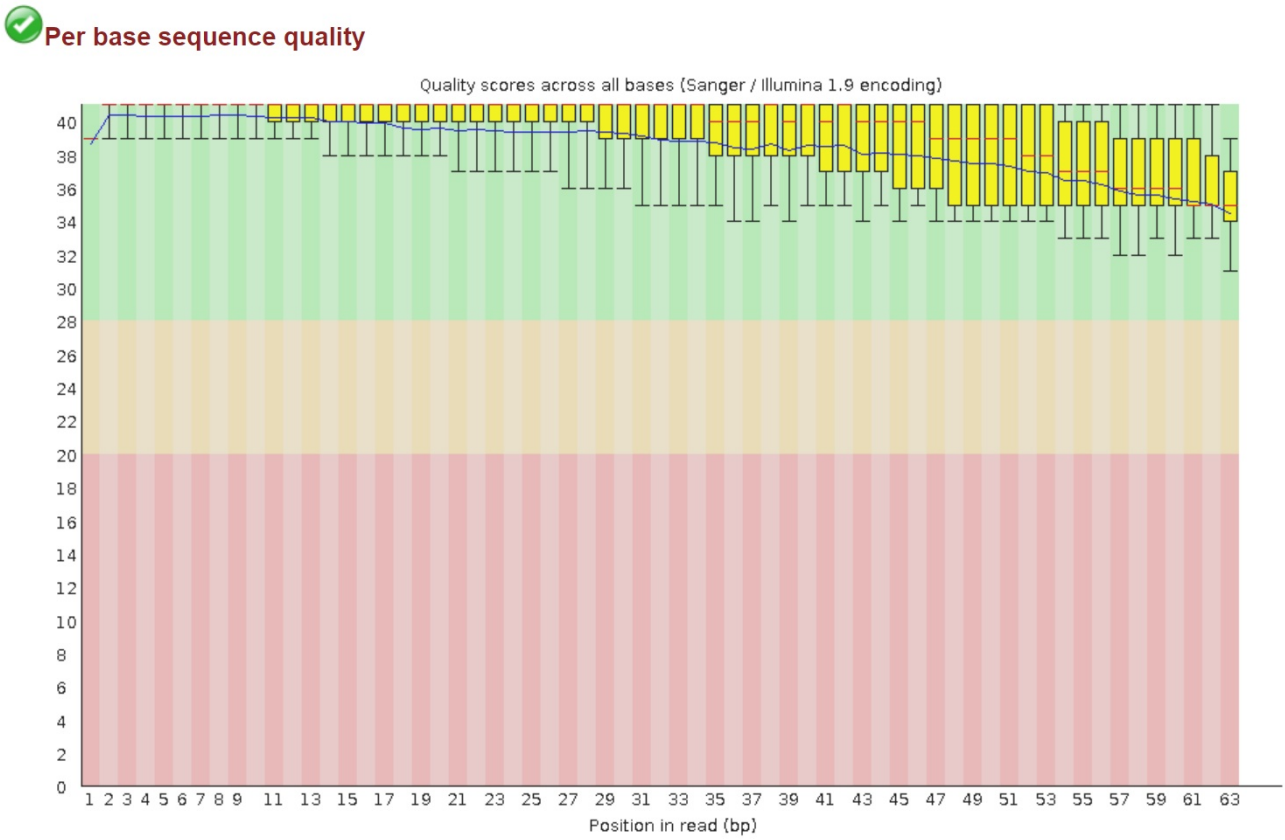



前处理结果




qc结果解读：

1. Per base sequence quality

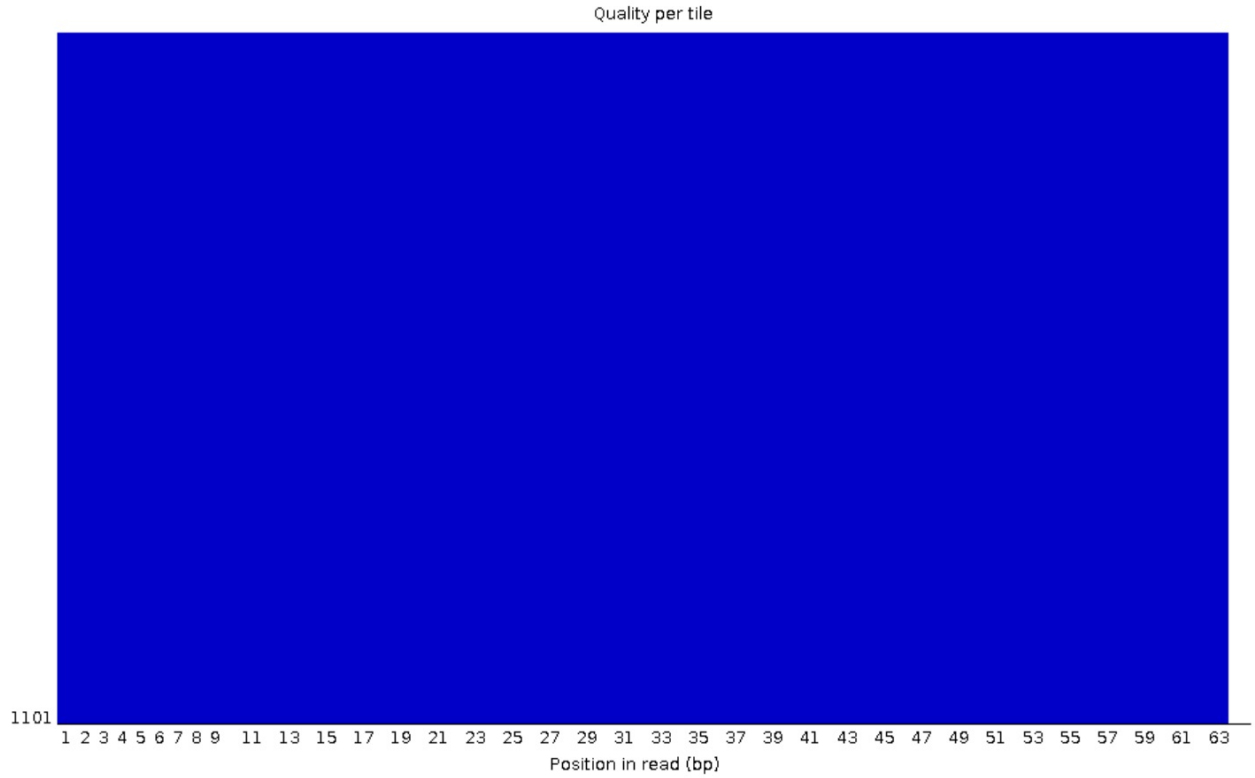



纵坐标为测序质量，根据测序质量划分成了3个区间，0-20之间，背景色为红色，测序质量非常糟糕；20-28之间，背景色为橘色，测序质量差；28以上，背景色为红色，测序质量良好。

如图，开始时测序质量较高，随着测序反应的进行，酶活性等因素降低，会导致测序质量变差，所以在结尾部分会观察到碱基质量降低的趋势。

2. Per tile quality

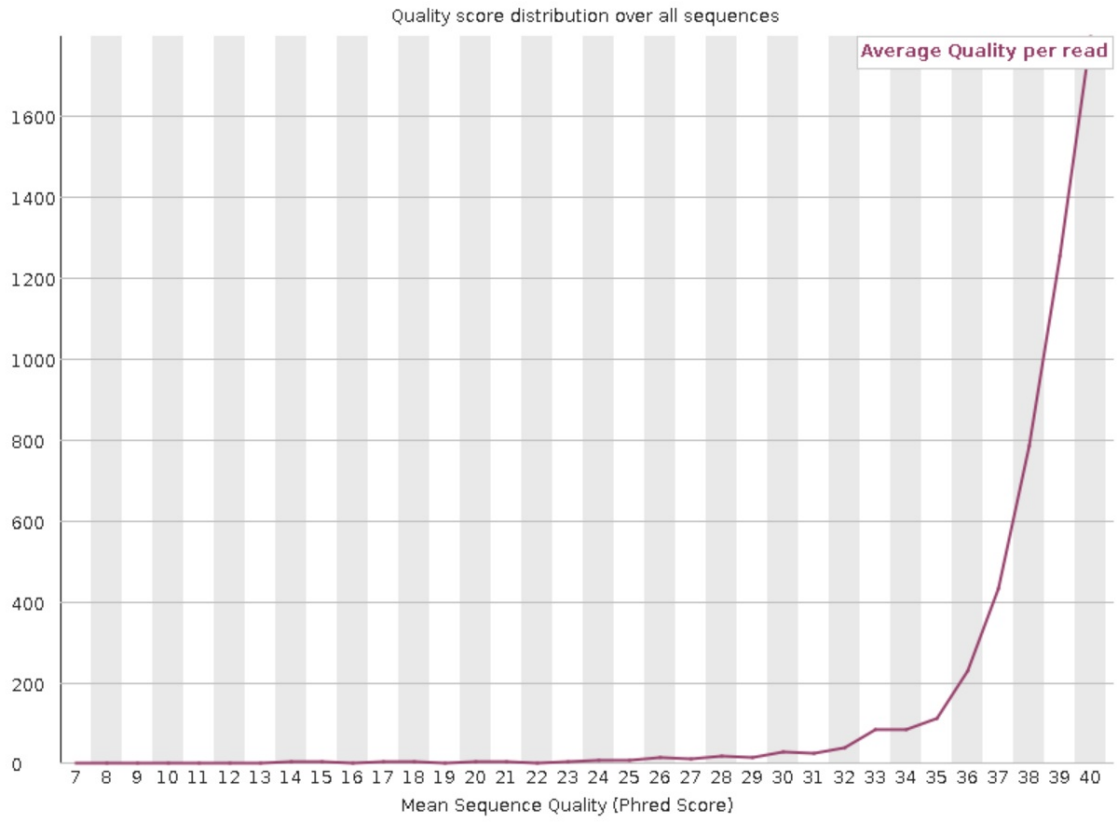
## Per tile sequence quality




热图的颜色从蓝色过滤到红色，蓝色表明该tile的测序质量好，红色表明该tile的测序质量差，一个良好的测序结果中，基本上全部是蓝色。

3. Per sequence quality scores

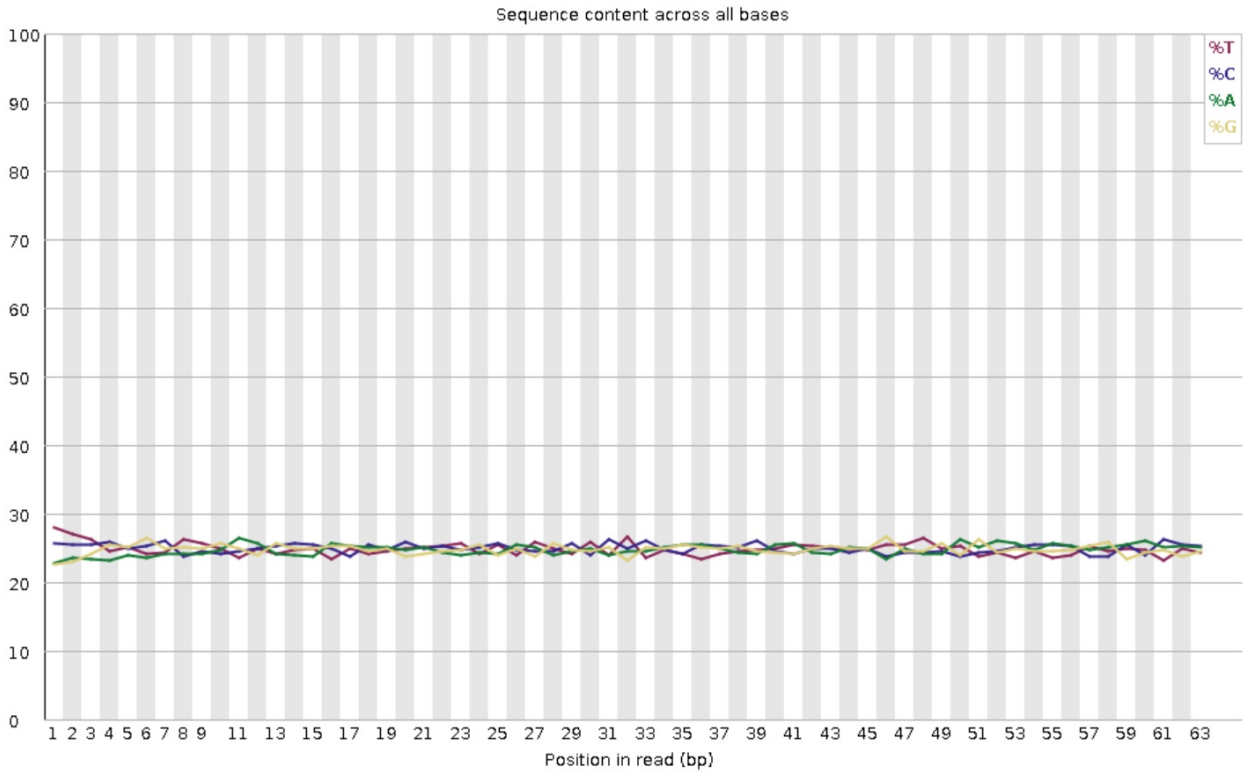
## Per sequence quality scores




碱基平均质量越高的reads数越多，说明测序质量越好。上图说明大部分reads序列平均质量在Q40以上，测序质量良好。

4. Per base sequence content

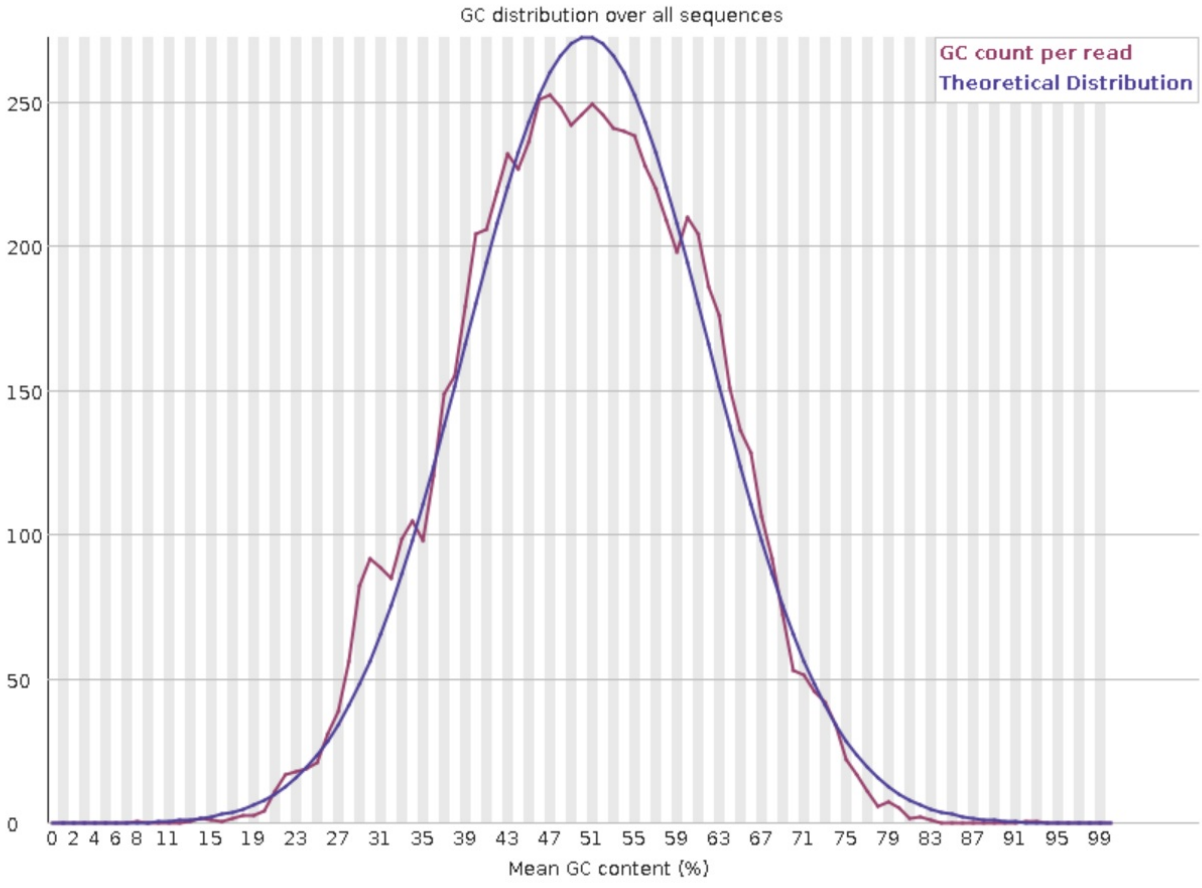
## Per base sequence content




理想情况下，各个碱基的比例并不会随着测序反应的进行发生变化，所以每个碱基对应的线相互平行，且对于碱基随机分布的文库，A和T碱基数量相等，G和C碱基数量相等。

5. Per sequence gc content

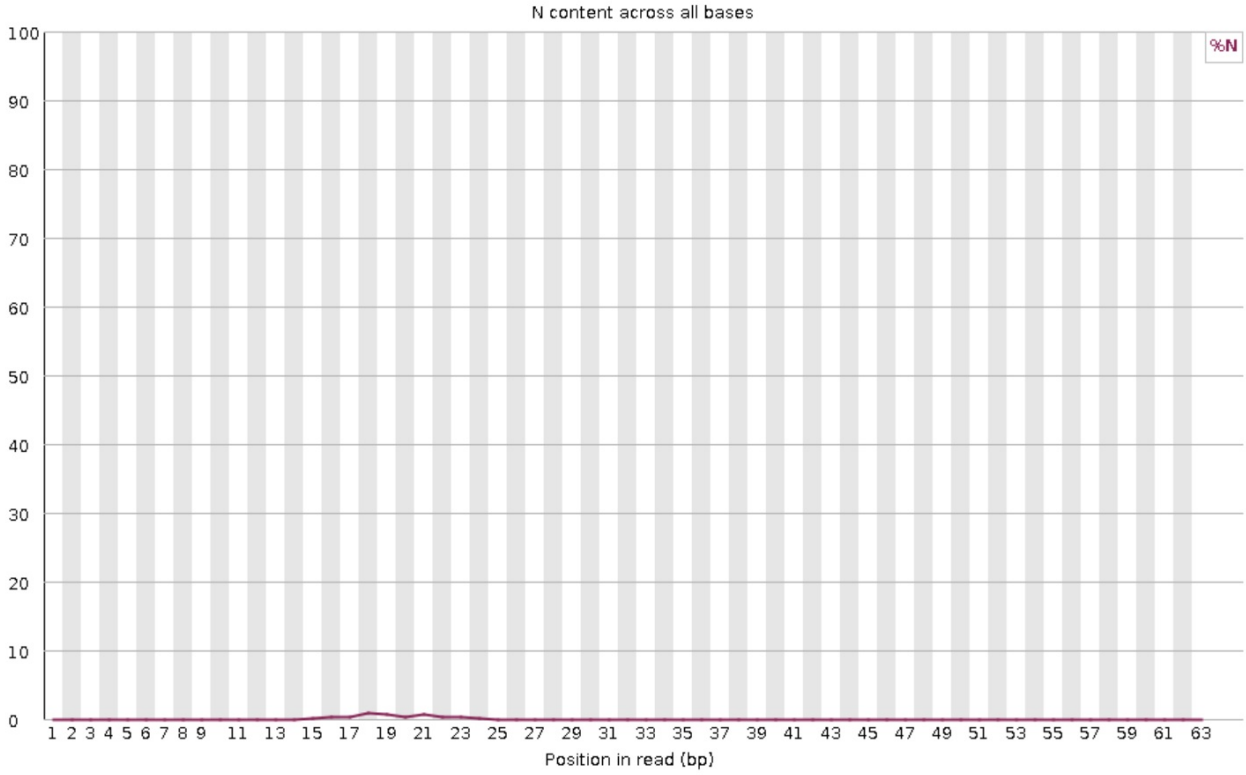
## Per sequence GC content




理想情况下，序列的GC含量分布是符合正态分布的，该数据基本符合正态分布质量良好。

6. Per base N content

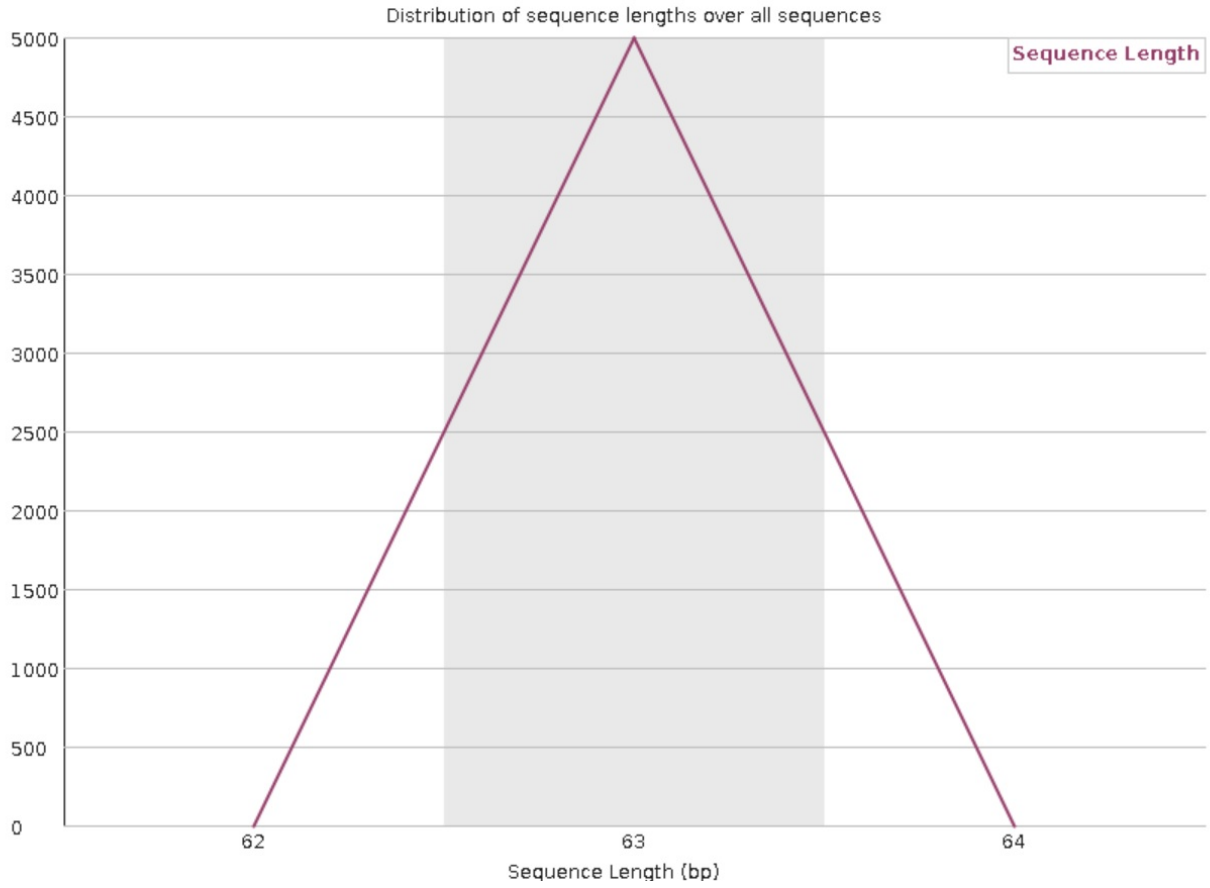
## Per base N content




当测序仪无法识别具体是哪种碱基时，就会给出N, 该数据N比例小，数据质量好。

7. sequence length distribution

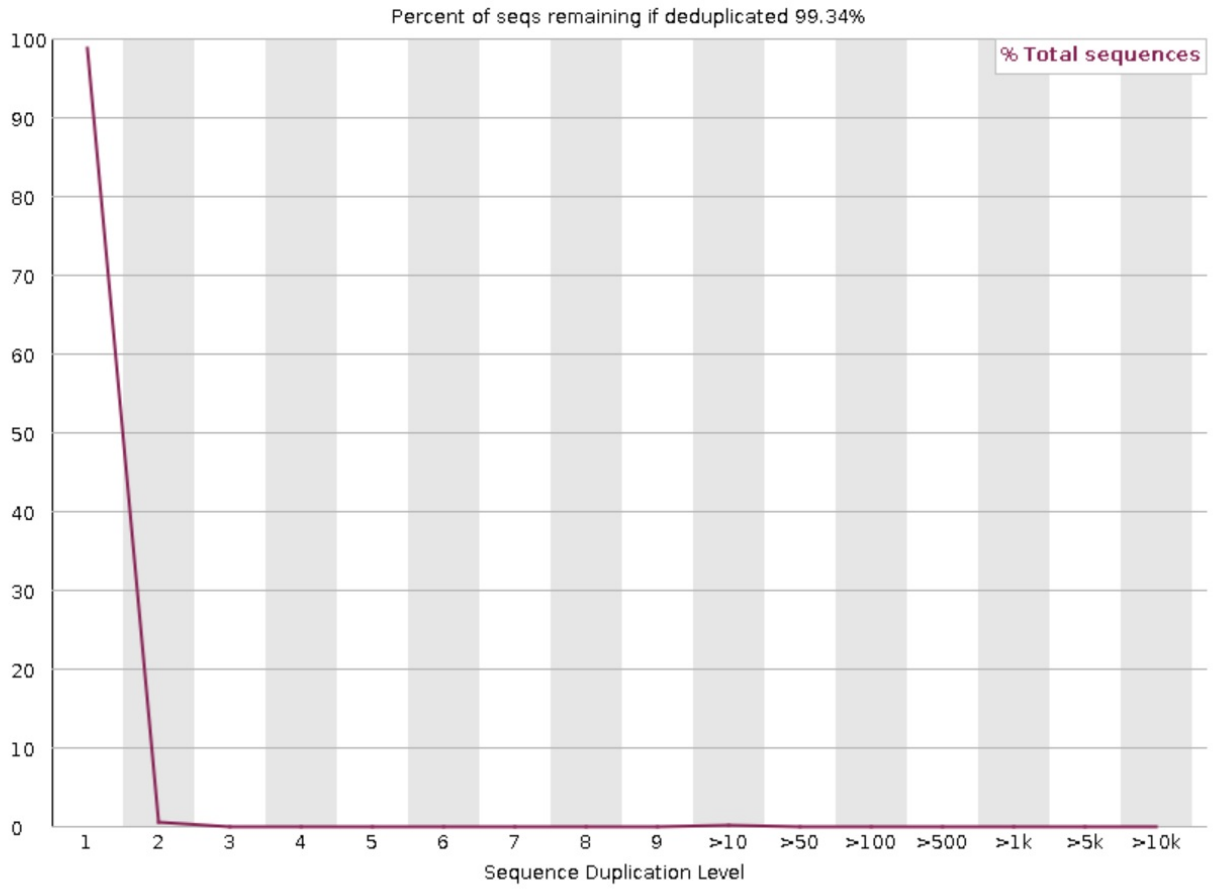
## Sequence Length Distribution




序列长度分布，该分布表明序列长度均为63.

8. Duplicate sequences

## Sequence Duplication Levels




基因组覆盖度越高，测序得到的序列重复比例会越低；在文库构建过程中，如果某些片段PCR扩增的比例大于随机扩增的比例，会导致重复序列比例高。

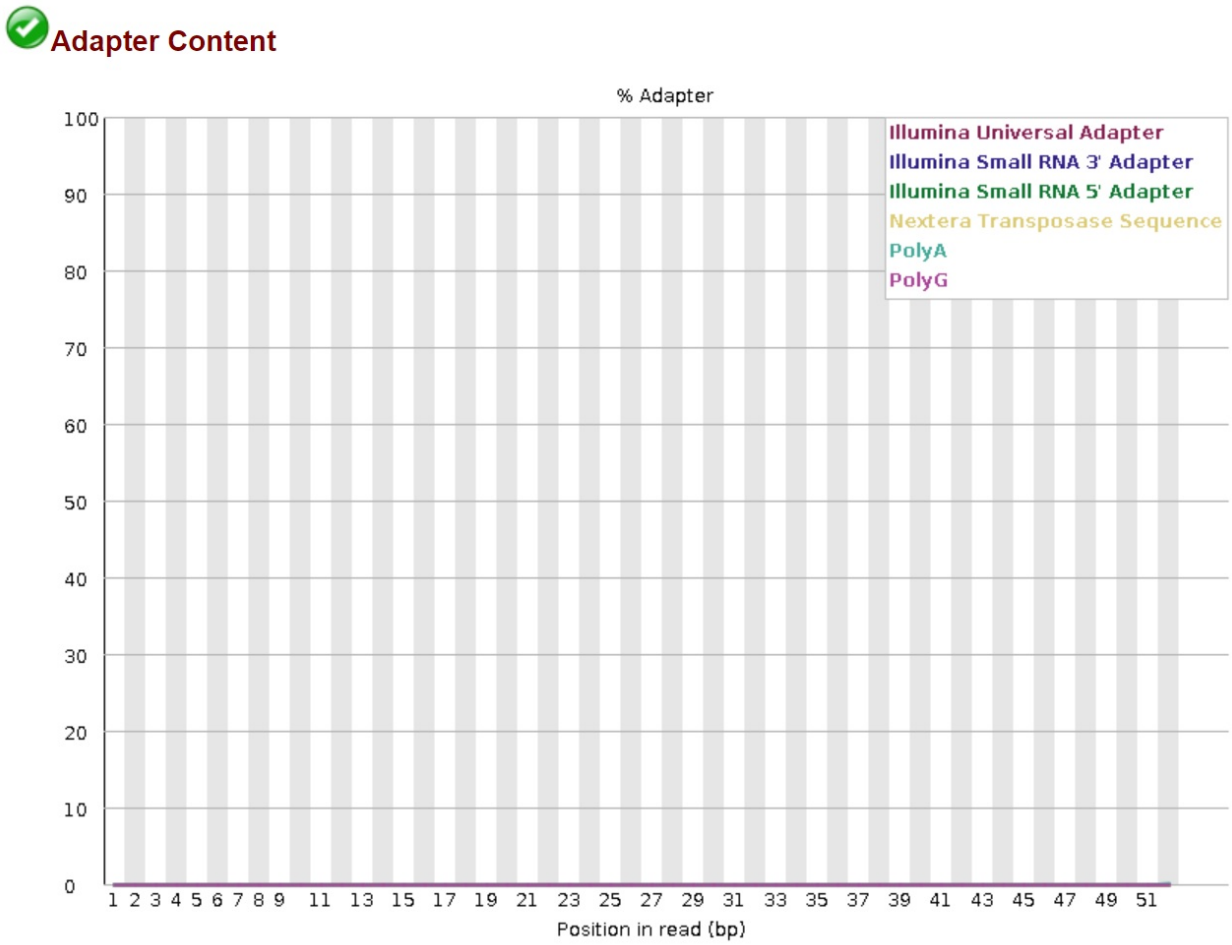
9. overrepresented sequences

## Overrepresented sequences

| Sequence | Count | Percentage | Possible Source |
|---|---|---|---|
| ACACGTCTGAACTCCAGTCACCGATGTATCTCGTATGCCGTCTTCTGCTT | 15 | 0.3 | TruSeq Adapter, Index 2 (100% over 50bp) |

过表达序列为adapter

10. adapter content




这部分内容给出序列中包含的adapter 序列的情况，该图片表明序列中几乎没有包含图例中的adapter序列。

## 4. 基因表达定量

In [ ]:
```
# Read the featureCounts TXT file
expression_matrix <- read.table("./data/fastq/gene_counts.txt", row.names = 1, header=T, sep = "\t",)
# rename the headers from ..X.fastq.. to X
colnames(expression_matrix) <- gsub("..data.fastq..", "", colnames(expression_matrix))
colnames(expression_matrix) <- gsub("_starAligned.sortedByCoord.out.bam", "", colnames(expression_matrix))

# drop the rows of all zero
expression_matrix <- expression_matrix[1:nrow(expression_matrix),6:ncol(expression_matrix)]
expression_matrix <- expression_matrix[rowSums(expression_matrix) != 0,]

gene_ids <- rownames(expression_matrix)
```

In [ ]:
```
library(clusterProfiler)
library(org.Hs.eg.db)
# transform gene ids to symbols
gene_symbols <- bitr(gene_ids, fromType = "ENSEMBL", toType = "SYMBOL", OrgDb = org.Hs.eg.db)
```

In [ ]:
```
# alternative way to get gene symbols
# library(biomaRt)

# ensembl <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")

# genes <- getBM(attributes = c('ensembl_gene_id', 'external_gene_name'),
#                filters = 'ensembl_gene_id',
#                values = gene_ids,
#                mart = ensembl)
```

In [ ]:
```
# Assign new row names
genes_subset <- subset(gene_symbols, SYMBOL != "")
gene_symbols <- gene_symbols$SYMBOL[match(gene_ids, gene_symbols$ENSEMBL)]
```

In [ ]:
```
# transform geneid in expression matrix to gene symbol and sum the expression of the same gene and set the grou
expression_matrix <- aggregate(expression_matrix, by=list(gene_symbols), FUN=sum)
colnames(expression_matrix)[1] <- "gene_symbol"

# set gene symbol as row name and remove the first column
rownames(expression_matrix) <- expression_matrix$gene_symbol
expression_matrix <- expression_matrix[,-1]
```

```
# remove the first row
# expression_matrix <- expression_matrix[-1,]
```

In [ ]:
```
expression_matrix[1:5,1:ncol(expression_matrix),]
```

A data.frame: 5 × 4

| | SRR1039508 | SRR1039509 | SRR1039512 | SRR1039513 |
|---|---|---|---|---|
| | <int> | <int> | <int> | <int> |
| **ACO2** | 4 | 2 | 0 | 0 |
| **APOBEC3C** | 0 | 0 | 0 | 2 |
| **APOL6** | 0 | 0 | 2 | 2 |
| **ATF4** | 0 | 2 | 0 | 0 |
| **ATP5MGL** | 0 | 0 | 2 | 0 |

In [ ]:
```
# transform counts in expression_matrix to TPM
# TPM = (count/sum of counts in a sample) * 1,000,000

# Convert expression_matrix to numeric matrix
expression_matrix <- as.matrix(expression_matrix)

# Calculate the sum of counts in each sample
sum_counts <- colSums(expression_matrix)

# Calculate the scaling factor for each sample
scaling_factor <- sum_counts / sum(sum_counts) * 1e6

# Transform counts to TPM
expression_matrix_tpm <- t(t(expression_matrix) / scaling_factor)
expression_matrix_tpm <- data.frame(expression_matrix_tpm)

# Print the transformed expression matrix
expression_matrix_tpm[1:5, 1:ncol(expression_matrix_tpm)]
```

A data.frame: 5 × 4

| | SRR1039508 | SRR1039509 | SRR1039512 | SRR1039513 |
|---|---|---|---|---|
| | <dbl> | <dbl> | <dbl> | <dbl> |
| **ACO2** | 1.253465e-05 | 9.240876e-06 | 0.000000e+00 | 0.000000e+00 |
| **APOBEC3C** | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 8.167742e-06 |
| **APOL6** | 0.000000e+00 | 0.000000e+00 | 9.107914e-06 | 8.167742e-06 |
| **ATF4** | 0.000000e+00 | 9.240876e-06 | 0.000000e+00 | 0.000000e+00 |
| **ATP5MGL** | 0.000000e+00 | 0.000000e+00 | 9.107914e-06 | 0.000000e+00 |

In [ ]:
```
write.csv(expression_matrix, file = "./data/gene_counts.csv")
write.csv(expression_matrix_tpm, file = "./data/gene_tpm.csv")
```

# RNA-seq 处理后分析

## 差异表达分析

In [ ]:
```
library(DESeq2)
library(pheatmap)
```

In [ ]:
```
# load data
data <- read.csv("./data/raw_count.csv", row.names = 1)

# overview of the data
cat('row=',nrow(data), ' col=',ncol(data))
data[1:6, 1:4]
```

```
row= 56636  col= 4
```

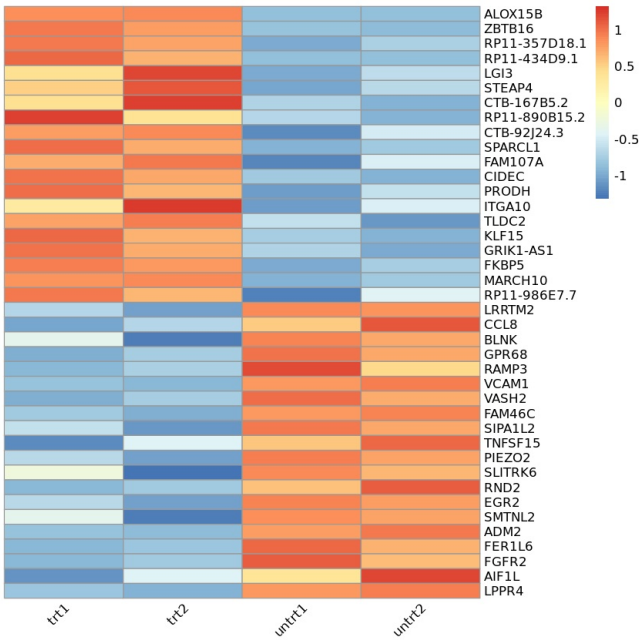
A data.frame: 6 × 4

| | trt1 | trt2 | untrt1 | untrt2 |
|---|---|---|---|---|
| | <int> | <int> | <int> | <int> |
| **A1BG** | 18 | 6 | 18 | 23 |
| **A1BG-AS1** | 115 | 105 | 90 | 110 |
| **A1CF** | 0 | 0 | 1 | 0 |
| **A2M** | 17398 | 30450 | 22673 | 37152 |
| **A2M-AS1** | 60 | 20 | 94 | 44 |
| **A2ML1** | 0 | 0 | 0 | 4 |

In [ ]:
```r
# create a coldata dataframe that links the sample names to the condition
countData <- as.matrix(data)
colData <- data.frame(condition = factor(c("trt", "trt", "untrt", "untrt")))

# create a DESeqDataSet object
dds <- DESeqDataSetFromMatrix(countData = countData, colData = colData, design = ~ condition)

dds = dds[rowSums(counts(dds))>1,]
# DESeq2 analysis
dds <- DESeq(dds)
res <- results(dds, contrast = c("condition", "trt", "untrt"))
```

In [ ]:
```r
# for high expression genes
sig_up <- subset(res, padj < 0.01 & log2FoldChange > 1)
sig_up_top20 <- head(sig_up[order(sig_up$log2FoldChange, decreasing = TRUE),], 20)
sig_up_top50 <- head(sig_up[order(sig_up$log2FoldChange, decreasing = TRUE),], 50)

# for low expression genes
sig_down <- subset(res, padj < 0.01 & log2FoldChange < -1)
sig_down_top20 <- head(sig_down[order(sig_down$log2FoldChange, decreasing = FALSE),], 20)

# combine top 20 up and down genes
top_genes <- c(rownames(sig_up_top20), rownames(sig_down_top20))

# extract count data for top genes and normalization
heatmap_data <- countData[top_genes, ]
heatmap_data <- log2(heatmap_data + 1)
```

In [ ]:
```r
# plot heatmap
pheatmap(heatmap_data, cluster_rows = FALSE, cluster_cols = FALSE, scale = "row", angle_col = 45,)
```




## GO biological process 富集

In [ ]:
```r
library(clusterProfiler)
library(enrichplot)
library(org.Hs.eg.db)
library(GOplot)
```

In [ ]:
```r
genes_symbol <- c(rownames(sig_up))
```

```
genes_entrez <- bitr(genes_symbol, fromType = "SYMBOL", toType = "ENTREZID", OrgDb = org.Hs.eg.db)
```

In [ ]:
```
ego <- enrichGO(gene         = genes_entrez[['ENTREZID']],
                OrgDb        = org.Hs.eg.db,
                keyType      = "ENTREZID",
                ont          = "BP",  # Biological Process
                pAdjustMethod = "BH",
                pvalueCutoff = 0.01,
                qvalueCutoff = 0.01)
```

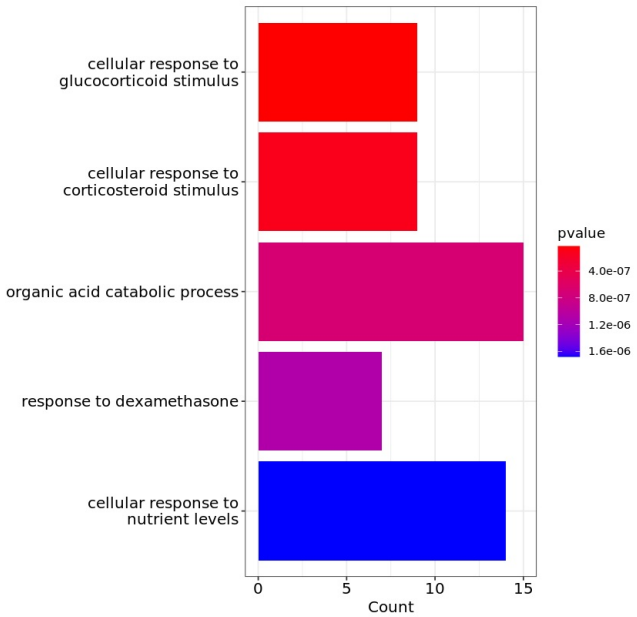
In [ ]:
```
# table view
ego_top5 <- head(ego, 5)
ego_top5
```

A data.frame: 5 × 9

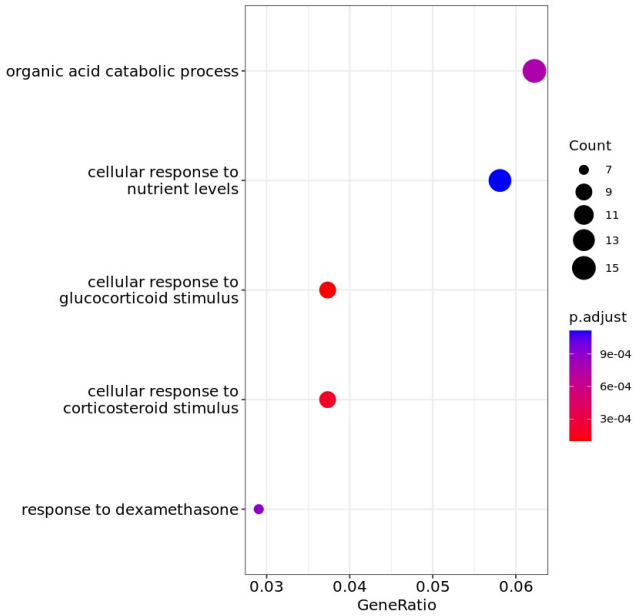
| | ID | Description | GeneRatio | BgRatio | pvalue | p.adjust | qvalue | |
|---|---|---|---|---|---|---|---|---|
| | <chr> | <chr> | <chr> | <chr> | <dbl> | <dbl> | <dbl> | |
| **GO:0071385** | GO:0071385 | cellular response to glucocorticoid stimulus | 9/241 | 55/18903 | 2.925768e-08 | 9.716476e-05 | 8.158274e-05 | |
| **GO:0071384** | GO:0071384 | cellular response to corticosteroid stimulus | 9/241 | 64/18903 | 1.147052e-07 | 1.904679e-04 | 1.599231e-04 | |
| **GO:0016054** | GO:0016054 | organic acid catabolic process | 15/241 | 247/18903 | 6.927150e-07 | 7.668355e-04 | 6.438604e-04 | 10157/18/35/13787 |
| **GO:0071548** | GO:0071548 | response to dexamethasone | 7/241 | 43/18903 | 1.093799e-06 | 9.081266e-04 | 7.624930e-04 | |
| **GO:0031669** | GO:0031669 | cellular response to nutrient levels | 14/241 | 231/18903 | 1.679412e-06 | 1.115465e-03 | 9.365815e-04 | 467/11170/2308/23 |

In [ ]:
```
# visualization
library(ggplot2)
library(gridExtra)
library(patchwork)
```

In [ ]:
```
barplot(ego, showCategory = 5, color = "pvalue")
```




In [ ]:
```
dotplot(ego, showCategory = 5)
```




In [ ]:
```
cnetplot(ego, showCategory = 5, circular = TRUE)
```