RNA-seq 期末报告

Group 2 刘曦瑞 吴仁杰 徐震洋 谢天辰

本流程使用的 Linux 软件均可通过 anaconda 安装，代码按使用顺序见下：

```
conda install -c bioconda sra-tools
conda install -c bioconda fastqc
conda install -c bioconda trimmomatic
conda install -c bioconda hisat2
conda install -c bioconda samtools
conda install -c bioconda stringtie
```

数据获取

测序数据

测序数据选自文章 *Arabidopsis* Duodecuple Mutant of PYL ABA Receptors Reveals PYL Repression of ABA-Independent SnRK2 Activity。通过 NCBI 数据库及文章后列出的 GEO 收录号 GSE114379 查找得到 SRA 序列号：

13	SRR7160928	SAMN09205655	5.27 G	1.84 Gb	SRX4080143	wildtype	GSM3140728	GSM3140728	300 mM mannitol for 24 hours
14	SRR7160929	SAMN09205654	4.80 G	1.70 Gb	SRX4080144	wildtype	GSM3140729	GSM3140729	300 mM mannitol for 24 hours
15	SRR7160930	SAMN09205653	5.03 G	1.73 Gb	SRX4080145	wildtype	GSM3140730	GSM3140730	300 mM mannitol for 24 hours
16	SRR7160931	SAMN09205652	4.58 G	1.60 Gb	SRX4080146	wildtype	GSM3140731	GSM3140731	1/2 MS for 24 hours
17	SRR7160932	SAMN09205651	5.00 G	1.72 Gb	SRX4080147	wildtype	GSM3140732	GSM3140732	1/2 MS for 24 hours
18	SRR7160933	SAMN09205650	4.71 G	1.64 Gb	SRX4080148	wildtype	GSM3140733	GSM3140733	1/2 MS for 24 hours

根据此序列号可以在安装 `sra-tools` 后通过 `prefetch` 直接下载测序数据。
由于数据量较大，且服务器上已经下载过了，因此实验流程中直接使用服务器上数据，并未实际执行以下步骤。

流程（以单个文件为例）：

```
prefetch SRR***
输出：SRR***.sra
```

获得 `.sra` 数据，再通过 fastq-dump 或 fasterq-dump 软件解压得到双端测序的 `.fastq` 文件：

```
fasterq-dump -e 4
    -p #显示进度条
    -3(--split-3) #分开两个read
    --gzip #输出压缩文件
    -o <outfile>
    SRR***.sra #之前下载的sra数据
输出：SRR***_1.fastq SRR***_2.fastq
```

基因组及注释数据

本次实验物种为拟南芥，参考基因组选自 TAIR www.arabidopsis.org/Genes 。下载的数据包括拟南芥基因组 gff3/gtf 注释文件与基因组序列 `.fas`。其中基因组序列 `.fas` 被用于 Hisat2 创建索引； gff3 注释文件被用于 Stringtie 转录本定量步骤。

```
#基因组注释文件，提供剪接位点与Stringtie所需注释信息
wget
https://www.arabidopsis.org/download_files/Genes/Araport11_genome_release/Araport11_GFF3_genes_transposons.May2022.gff.gz
wget
https://www.arabidopsis.org/download_files/Genes/Araport11_genome_release/Araport11_GTF_genes_transposons.May2022.gtf.gz

#依据该序列文件创建hisat2索引
wget
https://www.arabidopsis.org/download_files/Genes/TAIR10_genome_release/TAIR10_chromosome_files/TAIR10_chr_all.fas
```

去接头——Trimmomatic

实验流程中我们选取 Trimmomatic 作为质量控制与去接头软件。Illumina 测序数据常用 Trimmomatic 作为去接头软件。
Trimmomatic 安装时自带测序可能的接头/污染序列文件。如使用 conda 安装，该序列文件位于 `~/miniconda3/envs/ENV/share/trimmomatic-0.39-2/adapters/` 文件夹中，根据测序种类选取对应的文件：GAII 选择 TruSeq2 文件；HiSeq 与 MiSeq 选择 TruSeq3 文件。或者从 Illumina上直接查询接头。
使用时还需注意测序数据 `.fastq` 文件所使用的质量值体系，分为 phred33 与 phred64。目前常见的测序结果均为 phred33 格式。

本流程的脚本如下：

```
trimmomatic
    PE data/${input_file}_1.fastq data/${input_file}_2.fastq
    -phred33
    -baseout workdir/$input_file/$input_file
    -threads 4
    ILLUMINACLIP:./genome_annotation/TruSeq-PE.fa:2:30:10
    LEADING:5
    TRAILING:5
    SLIDINGWINDOW:5:20
    MINLEN:36
    AVGQUAL:20
```

其中各参数含义如下：

```
PE 指定为双端测序序列，后续两个参数分别为前导链与后随链
   如果为单端测序序列，使用SE
-phred33 指定phred33质量值体系
-baseout <path> 指定输出目录
-threads 4 指定线程数

下列参数为 trimmomatic特有的参数传递方式，具体形式为PARAM:value1:value2:…
    ILLUMINACLIP:./genome_annotation/TruSeq-PE.fa:2:30:10 切除Illumina测序接头，四个参数为：可能的污染序列文件、
    LEADING:5 切除read起始端质量低于5的碱基
    TRAILING:5 切除read末端质量低于5的碱基
    SLIDINGWINDOW:5:20 使用5个碱基的滑动窗口修剪，从5\'端开始扫描，当窗口内质量低于阈值20时，删除窗口内所有碱基
    MINLEN:36 舍弃长度低于36的read
    AVGQUAL:20 舍弃平均质量低于20的read
```

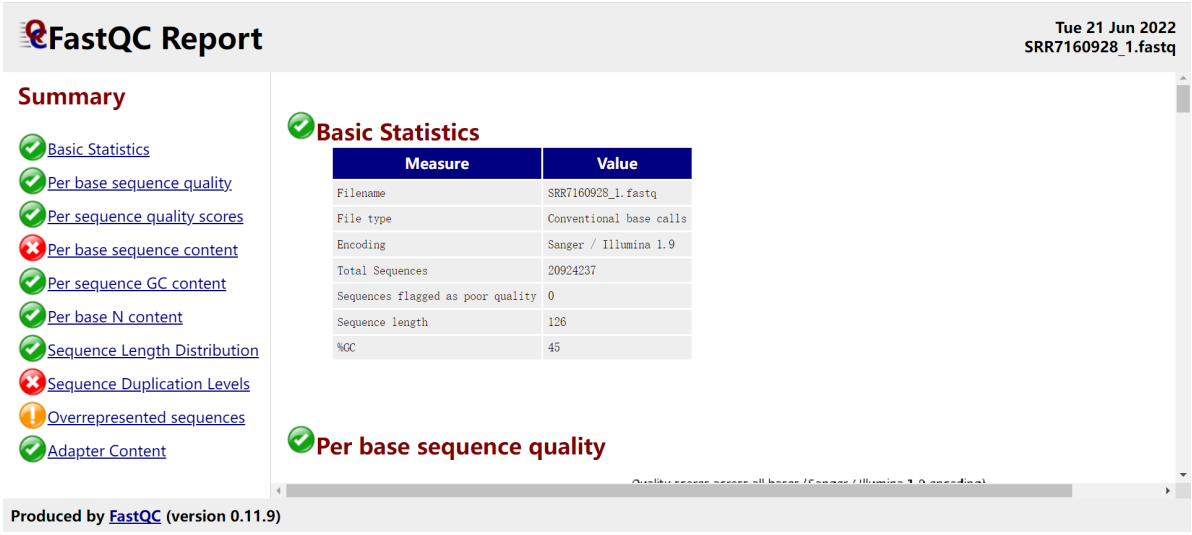
Trimmomatic 输出四个标准文件，后缀分别为 `1P` `1U` `2P` `2U`，这之中 `1U` 与 `2U` 为保留单端的读段，待用。`1P` 与 `2P` 分别为正向/反向双端均保留的读段，被用于进一步质量检验与后续处理。

质量控制与检验——fastqc

`fastqc`是一个依赖Java环境的高通量序列数据的质量检测工具，用于快速了解数据是否存在问题。本流程的脚本如下：

```
fastqc -t 4 \
       -o ${workdir} \
       ${workdir}/${input_file}_1P \
       ${workdir}/${input_file}_2P \
# 输入文件为上一步trimmomatic去接头后得到的两个paired fastq文件
# -t 指定线程数
# -o 指定输出文件存放位置
```

运行上述命令后会生成一个*_fastqc.html_结果报告文件和一个*fastqc.zip压缩文件，压缩文件中除了有html的报告，还有报告结果图片（Images文件夹内）和所有数据点（fastqc_data.txt）。打开html报告可以看到直观的质检结果。左栏的Summary中结果分为绿色的PASS，黄色的WARN和红色的FAIL：



包括10项结果：

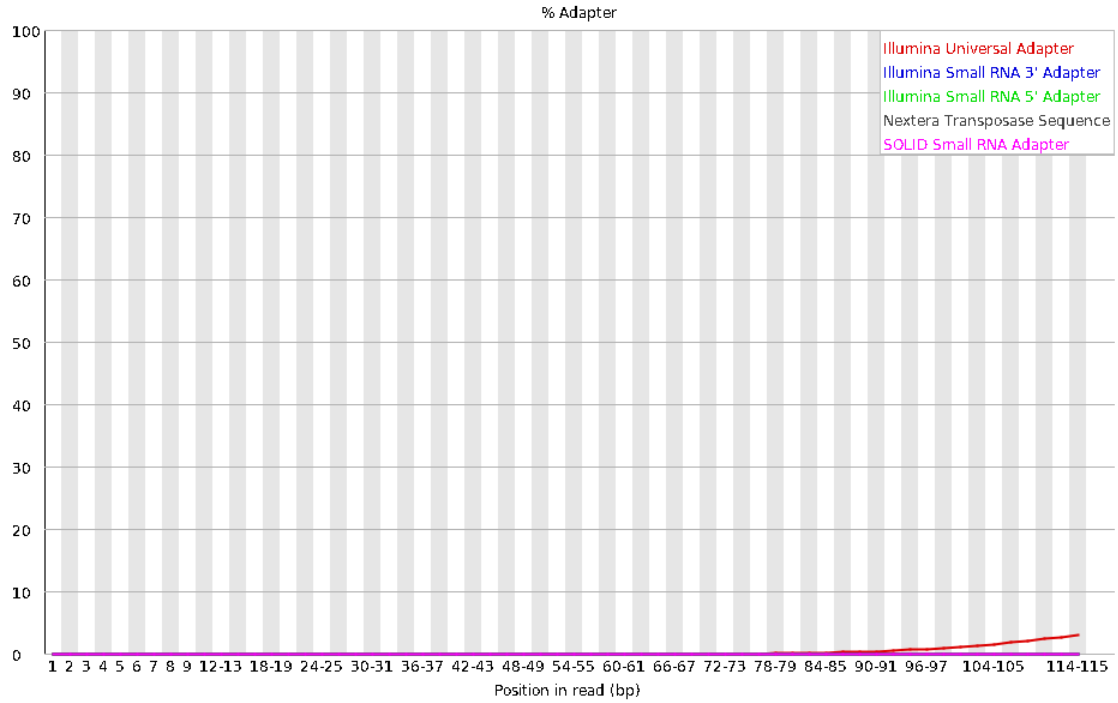
1. Basic Statistics基本信息：包括测序平台/类型、read数量、read长度、GC含量等
2. Per base sequence quality：所有reads在某个位置的测序质量统计结果
3. Per sequence quality scores：每条read的碱基质量均值的频率分布图
4. Per base sequence content：所有reads在某个位置的四种碱基含量
5. Per sequence GC content：统计reads的平均GC含量的分布
6. Per base N content：所有reads的某个位置N的比例
7. Sequence Length Distribution：统计reads长度的分布
8. Sequence Duplication Levels：统计序列的重复度，即文库中某条序列的拷贝数的分布
9. Overrepresented sequences：大量出现的序列列表
   统计接头序列的含量。一般测序仪自带软件会切去接头序列。
10. Adapter Content：统计所有reads再某个位置的接头含量

根据图中结果（SRR7160928_1.fastq的质检结果），发现存在Overrepresented sequences，且再Adapter Content中最后约30个碱基的位置曲线略微翘起：

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
ATCGGAAGAGCACACGTCTGAACTCCAGTCACTAGCTTATCTCGTATGCC	23029	0.11005897132593175	TruSeq Adapter, Index 10 (100% over 50bp)

Adapter Content

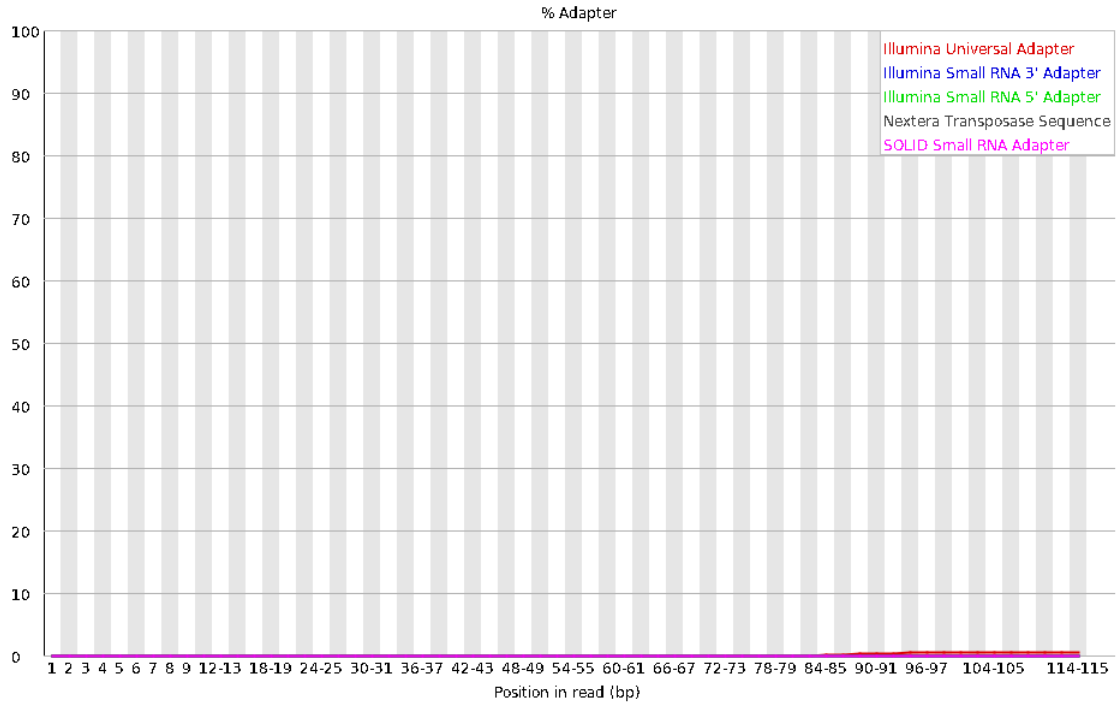


经过去接头，报告显示没有Overrepresented sequences，且Adapter Content曲线几乎为紧贴横轴的直线：

Overrepresented sequences

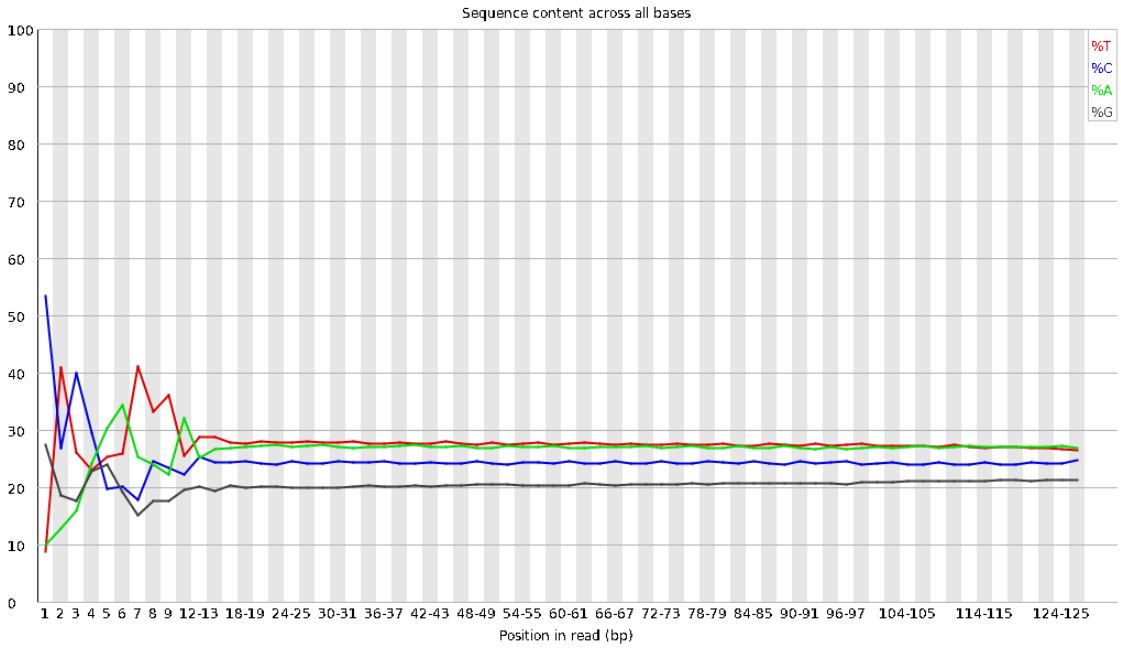
No overrepresented sequences

Adapter Content

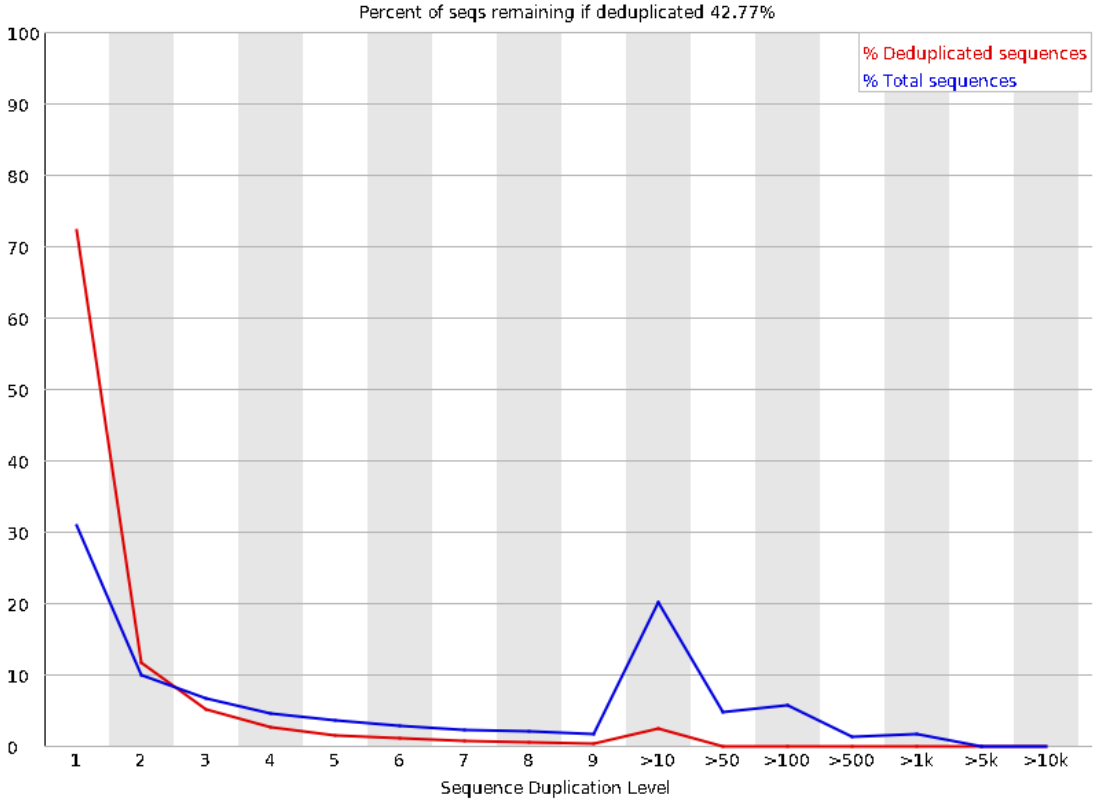


不过去接头后Per base sequence content与Sequence Duplication Levels仍报告FAIL：

Per base sequence content



Sequence Duplication Levels



这可能是建库过程存在偏差或文库遭到污染所造成的，并非trimmomatic去接头能改变的。

比对——Hisat2 & Samtools

本实验中，我们选取Hisat2进行序列比对，因其比对策略使之拥有更高的敏感性和更快的比对速度，同时方便后续使用stringtie继续处理。

在使用hisat2前，先对测序文件进行处理以得到外显子和剪接位点信息。使用hisat2提供的python脚本文件完成这一操作。

```
extract_exons.py Araport11_GTF_genes_transposons.Apr2022.gtf >
genome.exon.gtf
extract_splice_sites.py Araport11_GTF_genes_transposons.Apr2022.gtf >
splice_site.txt
```

随后即可使用hisat2建立索引。这一步需要用到之前生成的两个分别包含外显子和剪切位点信息的文件，以及在下载的数据包中包括的.fas基因组序列文件。

```
hisat2-build -p 16 --ss splice_site.txt --exon genome.exon.gtf
TAIR10_chr_all.fas ./genome
```

其中，-p设定线程数，-ss与--exon分别后跟外显子和剪切位点信息文件，TAIR10_chr_all.fas为下载的基因组序列文件，./genome为生成索引的前缀。

按照该命令运行，完成后生成了8个.ht(genome.*.ht)文件，即为索引。

完成建库之后，开始进行比对。

```
hisat2 -p 16 -x genome_annotation/genome --known-splicesite-infile
genome_annotation/splice_sites.txt --rna-strandness RF --dta -1 SRRnumber_1P
-2 SRRnumber_2P > hisat_out_sam
```

其中，-p设定线程数，因为该步骤极为耗时，因此建议设大；-x声明索引前缀，--known-splicesite-infile在比对时提供已知的剪切位点信息；--rna-strandness确定链特异性，此处参数设定为RF，则在sam文件中会通过"+""-"代表该reads所在转录本与基因组序列的关系；--dta使输出结果适用于stringtie处理；-1，-2为输入的两个双端测序文件。

比对完成后，hisat2会输出日志文件，给出数据的整体比对率、唯一比对率与多重比对等信息。

hisat2输出的sam文件需要使用samtools排序处理以方便下一步流程。

```
(samtools sort -@ 16 -l 9 -o bam -T tmp hisat_out_sam > hisat_out_bam) >
hisat_map_info 2>&1
```

各参数中，-@设定线程数；-l则是对输出文件的压缩等级进行设定，此处设定为最高的9；-o设定输出为bam文件；-T设定排序过程中临时文件的前缀；在将排序结果输出至hisat_out_bam后，再通过2>&1将标准输出与错误一起输出至hisat_map_info中，方便对整个流程的检查。

输出的bam文件进一步用于Stringtie的转录本定量处理。

转录本组装与定量——Stringtie

本实验从众多的组装工具中选择了stringtie作为最终选择的比对工具，其特点有：

1. 使用流神经网络算法；
2. 组装效果更好，有更高的灵敏度和准确度；
3. 运行速度远快于cufflinks等，适合本次实验。
   以下是实验脚本用到的程序，功能为记录组装转录本的信息，并进行转录本定量。

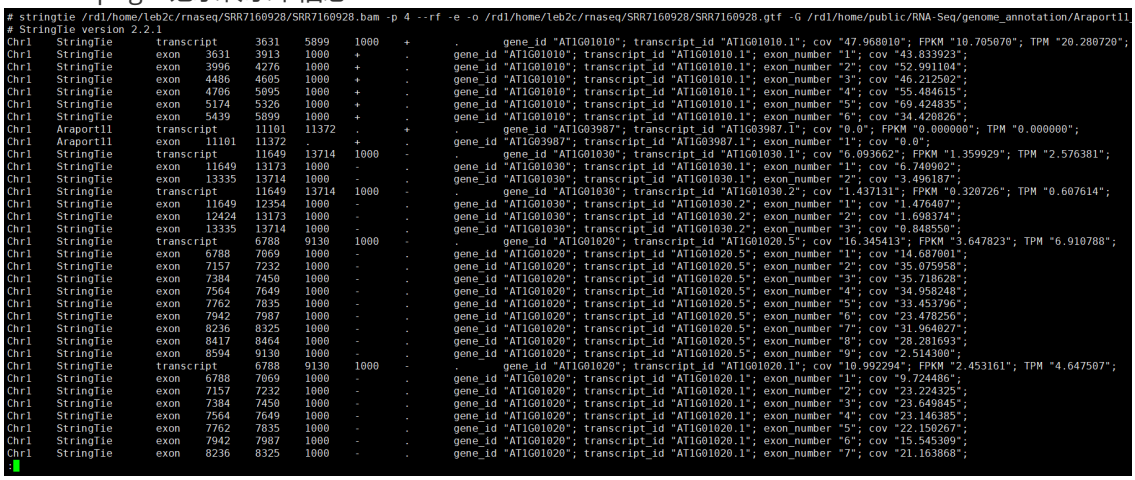
```
stringtie ${workdir}/${input_file}.bam -p 4 --rf -e \
          -o ${workdir}/${input_file}.gtf \
          -G
${pubR}/genome_annotation/Araport11_GTF_genes_transposons.Apr2022.gtf \
          -b ${workdir} 2> ${workdir}/stringLog.txt
```

其中各参数的含义如下：

```
输入：hisat和samtool处理完生成的bam文件
    -p 4 线程数
    --rf 选择链的建库方式：fr-firststrand
    -e 限制read的处理，仅处理与-G给出的参考转录本匹配的部分
    -o 输出的gtf文件命名
    -G 注释文件
    -b 输出*.ctab的路径，用于下游差异表达分析
```

输出文件：

1. i2t.ctab、e2t.ctab、i_data.ctab、e_data.ctab、t_data.ctab共五个*.ctab类型的文件
2. transcript.gtf 记录转录本信息



```
# stringtie /rd1/home/leb2c/rnaseq/SRR7160928/SRR7160928.bam -p 4 --rf -e -o /rd1/home/leb2c/rnaseq/SRR7160928/SRR7160928.gtf -G /rd1/home/public/RNA-Seq/genome_annotation/Araport11
# StringTie version 2.2.1
Chr1    StringTie       transcript      3631    5899    1000    +       .       gene_id "AT1G01010"; transcript_id "AT1G01010.1"; cov "47.968010"; FPKM "10.705070"; TPM "20.280720";
Chr1    StringTie       exon    3631    3913    1000    +       .       gene_id "AT1G01010"; transcript_id "AT1G01010.1"; exon_number "1"; cov "43.833923";
Chr1    StringTie       exon    3996    4276    1000    +       .       gene_id "AT1G01010"; transcript_id "AT1G01010.1"; exon_number "2"; cov "52.991104";
Chr1    StringTie       exon    4486    4605    1000    +       .       gene_id "AT1G01010"; transcript_id "AT1G01010.1"; exon_number "3"; cov "46.212502";
Chr1    StringTie       exon    4706    5095    1000    +       .       gene_id "AT1G01010"; transcript_id "AT1G01010.1"; exon_number "4"; cov "55.484615";
Chr1    StringTie       exon    5174    5326    1000    +       .       gene_id "AT1G01010"; transcript_id "AT1G01010.1"; exon_number "5"; cov "69.424835";
Chr1    StringTie       exon    5439    5899    1000    +       .       gene_id "AT1G01010"; transcript_id "AT1G01010.1"; exon_number "6"; cov "34.420826";
Chr1    Araport11       transcript      11101   11372   .       +       .       gene_id "AT1G03987"; transcript_id "AT1G03987.1"; cov "0.0"; FPKM "0.000000"; TPM "0.000000";
Chr1    Araport11       exon    11101   11372   .       +       .       gene_id "AT1G03987"; transcript_id "AT1G03987.1"; exon_number "1"; cov "0.0";
Chr1    StringTie       transcript      11649   13714   1000    -       .       gene_id "AT1G01030"; transcript_id "AT1G01030.1"; cov "6.093662"; FPKM "1.359929"; TPM "2.576381";
Chr1    StringTie       exon    11649   13173   1000    -       .       gene_id "AT1G01030"; transcript_id "AT1G01030.1"; exon_number "1"; cov "6.740902";
Chr1    StringTie       exon    13335   13714   1000    -       .       gene_id "AT1G01030"; transcript_id "AT1G01030.1"; exon_number "2"; cov "3.496187";
Chr1    StringTie       transcript      11649   13714   1000    -       .       gene_id "AT1G01030"; transcript_id "AT1G01030.2"; cov "1.437131"; FPKM "0.320726"; TPM "0.607614";
Chr1    StringTie       exon    11649   12354   1000    -       .       gene_id "AT1G01030"; transcript_id "AT1G01030.2"; exon_number "1"; cov "1.476407";
Chr1    StringTie       exon    12424   13173   1000    -       .       gene_id "AT1G01030"; transcript_id "AT1G01030.2"; exon_number "2"; cov "1.698374";
Chr1    StringTie       exon    13335   13714   1000    -       .       gene_id "AT1G01030"; transcript_id "AT1G01030.2"; exon_number "3"; cov "0.848550";
Chr1    StringTie       transcript      6788    9130    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; cov "16.345413"; FPKM "3.647823"; TPM "6.910788";
Chr1    StringTie       exon    6788    7069    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "1"; cov "14.687001";
Chr1    StringTie       exon    7157    7232    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "2"; cov "35.075958";
Chr1    StringTie       exon    7384    7450    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "3"; cov "35.718628";
Chr1    StringTie       exon    7564    7649    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "4"; cov "34.958248";
Chr1    StringTie       exon    7762    7835    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "5"; cov "33.453796";
Chr1    StringTie       exon    7942    7987    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "6"; cov "23.478256";
Chr1    StringTie       exon    8236    8325    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "7"; cov "31.964027";
Chr1    StringTie       exon    8417    8464    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "8"; cov "28.281693";
Chr1    StringTie       exon    8594    9130    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.5"; exon_number "9"; cov "2.514300";
Chr1    StringTie       transcript      6788    9130    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.1"; cov "10.992294"; FPKM "2.453161"; TPM "4.647507";
Chr1    StringTie       exon    6788    7069    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.1"; exon_number "1"; cov "9.724486";
Chr1    StringTie       exon    7157    7232    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.1"; exon_number "2"; cov "23.224325";
Chr1    StringTie       exon    7384    7450    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.1"; exon_number "3"; cov "23.649845";
Chr1    StringTie       exon    7564    7649    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.1"; exon_number "4"; cov "23.146385";
Chr1    StringTie       exon    7762    7835    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.1"; exon_number "5"; cov "22.150267";
Chr1    StringTie       exon    7942    7987    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.1"; exon_number "6"; cov "15.545309";
Chr1    StringTie       exon    8236    8325    1000    -       .       gene_id "AT1G01020"; transcript_id "AT1G01020.1"; exon_number "7"; cov "21.163868";
:
```

其中前8列为常规的gtf格式文件信息，从左到右依次为：
序列名、注释来源、类型、起始位置、终止位置、质量分数、正/负链、到下个密码子的步进。
第9列为属性，除了有`gene_id`，`transcript_id`等，还有覆盖度`cov`和衡量表达量的`FPKM`和`TPM`。例如`TPM`，先将覆盖基因的reads数除去基因长度，再除去样本深度总和，这样表达量不仅可以在不同长度基因之间进行比较，还可以在不同样本之间进行比较。

下游分析--deseq2

在进行deseq2前，现需要将stringtie输出的转录本信息读入r中。为此，首先要做的是将各gtf文件合成到csv内。

```
nano samplelist.txt:
  osmotic1 ./SRR7160928/transcript.gtf
  osmotic2 ./SRR7160929/transcript.gtf
  osmotic3 ./SRR7160930/transcript.gtf
  control1 ./SRR7160931/transcript.gtf
  control2 ./SRR7160932/transcript.gtf
  control3 ./SRR7160933/transcript.gtf

wget http://ccb.jhu.edu/software/stringtie/dl/prepDE.py3
python prepDE.py3 -i samplelist.txt
```

第一步将转录本信息文件路径保存至samplelist.txt文件中，第二步下载stringtie所提供的py脚本并运行。输出的两个文件transcript_count_matrix.csv和gene_count_matrix.csv即可用于下游分析中。

随后在r中进行读入，并对读入数据进行处理构建样品信息colData。此处按照之前stringtie处理，将样品分为osmotic和control两组。

```
countData <- as.matrix(read.csv("gene_count_matrix.csv",
row.names="gene_id"))
condition <- factor(c("control", "control", "control", "osmotic", "osmotic",
"osmotic"))
colData <- data.frame(row.names=colnames(countData), condition)
```

读入完毕后开始从表达矩阵countData与样品矩阵colData构建DESeqDataSet对象，并对低丰度数据进行过滤。

```
dds <- DESeqDataSetFromMatrix(countData = countData, colData = colData,
design = ~ condition)
keep <- rowSums(counts(dds)) >= 10
dds <- dds[keep,]
```

完成dds构建后，就需要对该对象进行处理进行计算以得到差异基因数据。

```
dds <- DESeq(dds, fitType = 'mean')
res <- results(dds, contrast = c('condition', 'control', 'osmotic'))
res1 <- data.frame(res, stringsAsFactors = FALSE, check.names = FALSE)
```

得到的表格结果包括6列：baseMean、log2FC、lfcSE、stat、pvalue、padj

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
AT1G01010|2200934,UniProt=Q0WV96	692.686145	0.70148733	0.17166693	4.08632757	4.382549e-05	1.354180e-04
AT1G01020	587.053469	-0.23112948	0.15590748	-1.48247839	1.382130e-01	2.109152e-01
AT1G01020|2200939,UniProt=Q5MK24	144.671314	0.29522492	0.52433599	0.56304531	5.734040e-01	6.728923e-01
AT1G01030|2200949,UniProt=Q9MAN1	167.811904	-0.17850682	0.25888040	-0.68953395	4.904873e-01	5.962455e-01
AT1G01030	21.197704	2.07414228	3.05188155	0.67962739	4.967404e-01	6.020717e-01
AT1G01040	2837.603071	-0.22554451	0.05064305	-4.45361197	8.443764e-06	2.867581e-05
AT1G01046	73.646704	-0.41108690	0.20097587	-2.04545398	4.081012e-02	7.370339e-02
AT1G01050	3973.202146	-0.12457482	0.07947727	-1.56742692	1.170149e-01	1.832167e-01
AT1G01060	882.654993	-0.77681220	0.36083208	-2.15283575	3.133159e-02	5.825471e-02
AT1G01060|2200969,UniProt=Q6R0H1	367.168611	-0.20555480	0.68185968	-0.30146203	7.630622e-01	8.311748e-01
AT1G01060|1005715162,UniProt=Q6R0H1	91.068214	-1.06896138	0.62188583	-1.71890294	8.563205e-02	1.404969e-01
AT1G01070	620.618895	1.40217902	0.11406988	12.29228065	9.965425e-35	1.834197e-33
AT1G01080	1132.137978	-1.53966523	0.30604735	-5.03080721	4.884191e-07	1.897399e-06
AT1G01080|2200974,UniProt=Q8W592	943.435061	-0.80579152	0.53830684	-1.49690000	1.344193e-01	2.060629e-01

1. baseMean表示所有样本经过归一化系数矫正的read counts（counts/sizeFactor）的均值
2. log2Foldchange表示该基因的表达发生了多大的变化，对差异表达的倍数取以2为底的对数
3. lfcSE(logfoldchange Standard Error)是对于log2Foldchange估计的标准误差估计
4. stat由log2Foldchange除以标准差所得
5. pvalue为原始的p值
6. padj为校正后的均值，由于重复独立试验次数很大时小概率事件更可能会出现，为避免假阳性需要对p值进行校正

然后找出$padj < 0.05$且$|log2Foldchange| > 1$（即表达量变化了两倍以上）的基因，在原始表达量矩阵中筛选出这些基因所在行

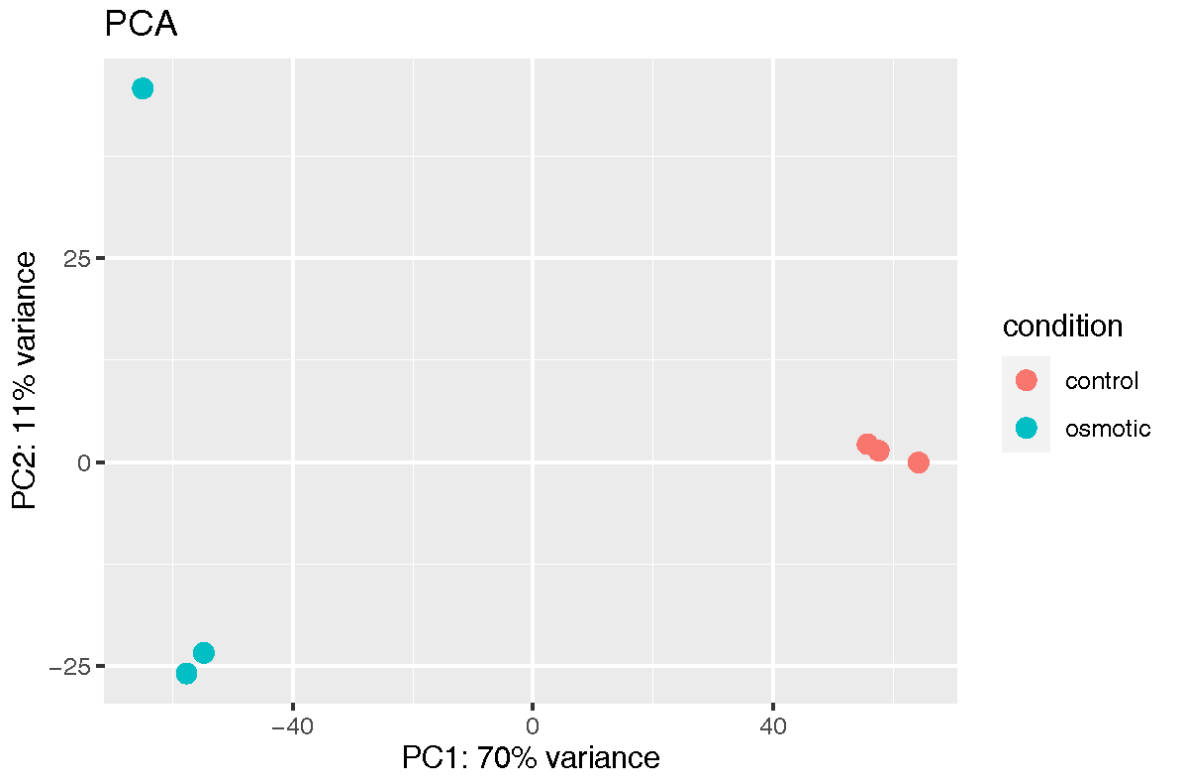
```
res1_up<- res1[which(res1$log2FoldChange >= 1 & res1$padj < 0.05),]
res1_down<- res1[which(res1$log2FoldChange <= -1 & res1$padj < 0.05),]
res1_total <- rbind(res1_up,res1_down)
df <-
as.matrix(countData[intersect(rownames(countData),rownames(res1_total)),])
```

PCA 分析

上游流程生成的原始表达矩阵通过 R 导入，通过 `DESeq2` 包内的 `vst` 函数进行方差标准化，再用 plotPCA 函数取差异表达量最高的10000个基因（实验证明该数值不影响结果）画出数据中实验组与对照组方差最大的两个维度，再通过 `ggplot` 包表示出解释方差的百分比。这一部分的全部代码如下：

```
vsd <- vst(dds)
pcaData <- plotPCA(vsd, intgroup = 'condition',
                   ntop = 10000, returnData = T)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(x = PC1, y = PC2, color = condition)) +
  geom_point(size =3) +
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance")) +
  ggtitle("PCA")
```

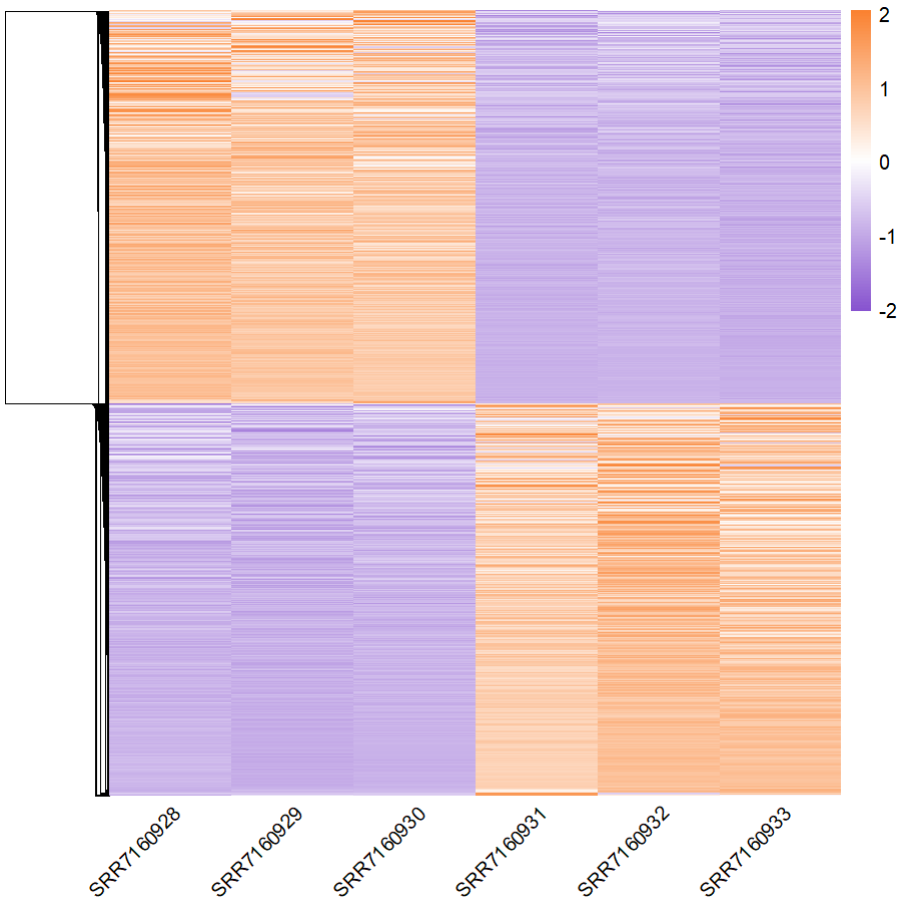
绘制结果见下，方差最大的维度将实验组与对照组分开，说明甘露醇处理实验组与对照组的 RNA 表达有明显差异：



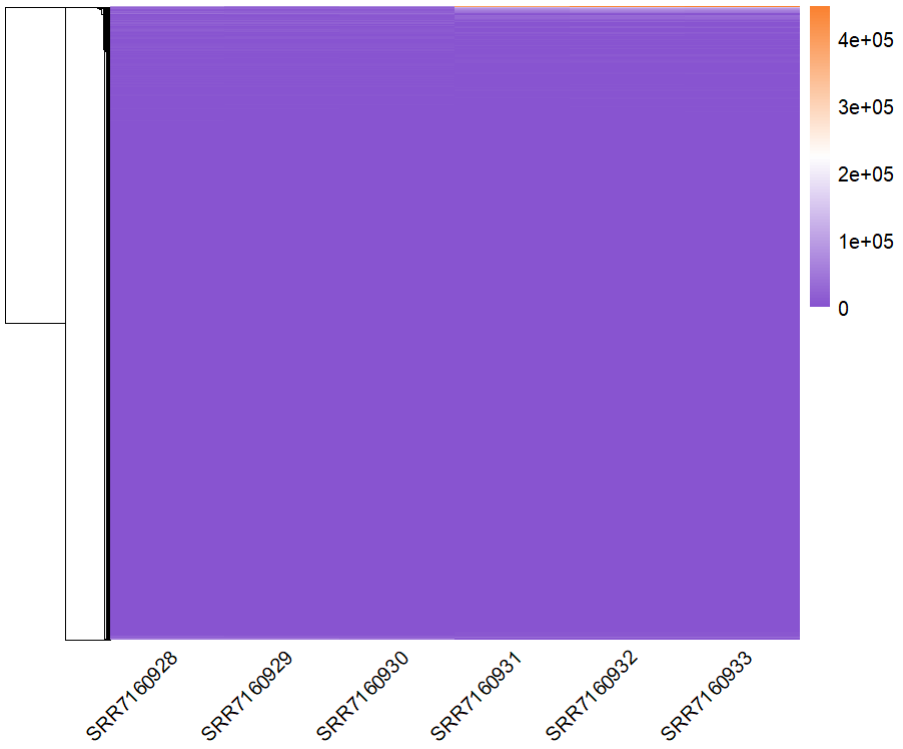
基因差异表达可视化——pheatmap

用上一步筛选后的表达量矩阵，通过pheatmap生成差异表达热图。代码和结果如下：

```
pheatmap(df,
         show_rownames = F,
         show_colnames = T, # 仅展示样本名
         cluster_cols = F,
         cluster_rows = T, # 将差异表达的基因聚在一起
         height=10, # 输出图片的高度
         scale = "row", # 对行进行归一化
         angle_col = 45, # 调整字的角度
         color = colorRampPalette(c("#8854d0", "#ffffff","#fa8231"))(100), # 颜色设置
         clustering_distance_rows = 'euclidean',
         clustering_method = 'single',
)
```

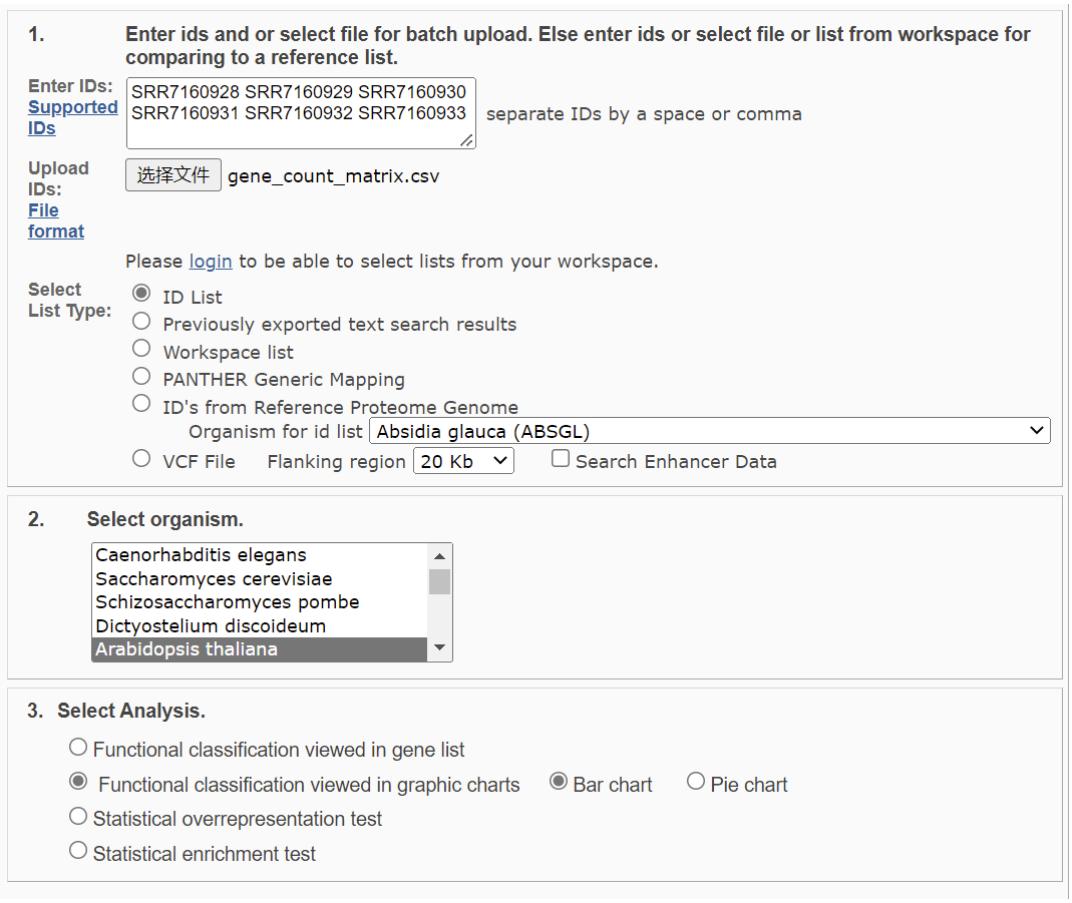


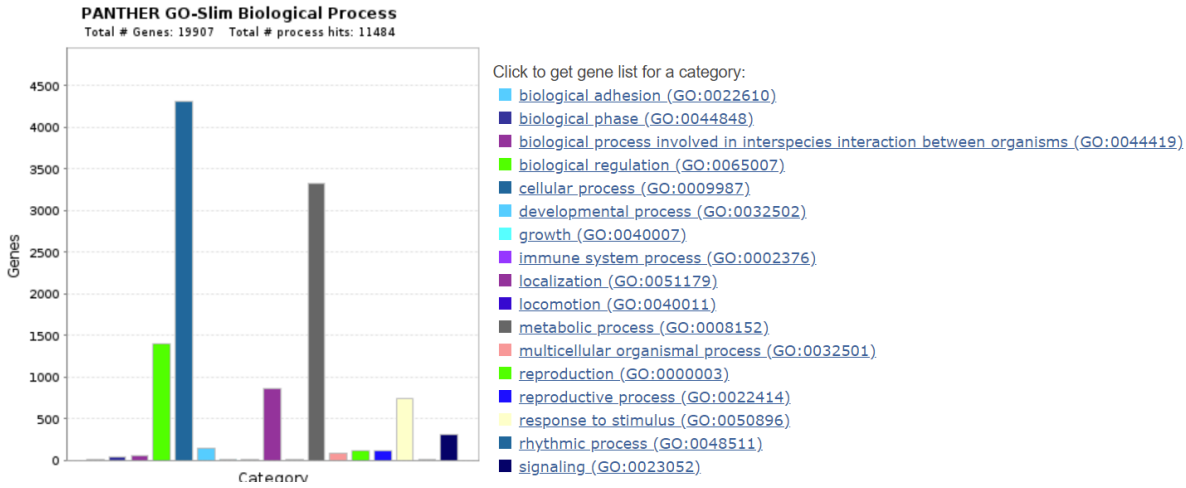
其中每一列为不同的样本，每一行为不同的基因。为了使一行中的不同列（即同一基因在不同样本中的表达量）看起来差异更显著，需要按行进行归一化处理（即`scale = "row"`参数），否则会因为一些基因表达量过高而导致整张图看起来都是低表达。例如将参数改为`scale = none`就会出现下图：



富集分析-Go

我们选择网页版Go进行富集分析，填写的数据见下图。输入序列ID以及DEseq的结果之后，网页自动输出Go分析的结果，其中包括以下几个方面：分子功能、生物学过程、细胞组成、蛋白质种类、作用路径。其中生物学过程以截图的形式进行展示，其余结果见报告内 pic 文件夹。





附录：脚本

00-pre.sh

```
#外显子序列
~/miniconda3/envs/RNAseq/bin/extract_splice_sites.py /rd1/home/public/RNA-Seq/genome_annotation/Araport11_GTF_genes_transposons.Apr2022.gtf > splice_site.txt

#构建hisat2基因组索引
hisat2-build -p 8 genome_annotations/TAIR10_chr_all.fas mygenome/genome

cat smlist | while read line
do
input_file=`echo "$line" |cut -f1`
mkdir -p workdir/$input_file
bash 01-hisat_stringtie.sh $input_file > ./log/${input_file}.log 2>&1
done
```

01-hisat_stringtie.sh

```
input_file=$1

trimmomatic
    PE data/${input_file}_1.fastq data/${input_file}_2.fastq \
    -threads 4 -baseout workdir/$input_file/$input_file \
    ILLUMINACLIP:./genome_annotation/TruSeq-PE.fa:2:30:10 \
    LEADING:5 TRAILING:5 SLIDINGWINDOW:5:20 MINLEN:36 AVGQUAL:20

fastqc
    workdir/$input_file/${input_file}_1P \
    workdir/$input_file/${input_file}_2P \
    -o workdir/$input_file/ -q -t 4

(hisat2
    -p 4 --rna-strandness RF \
    --known-splicesite-infile genome_annotation/splice_sites.txt \
    -x genome_annotation/genome --dta \
    -1 workdir/$input_file/${input_file}_1P \
    -2 workdir/$input_file/${input_file}_2P \
| samtools sort \
    -l 9 -O bam -T workdir/$input_file/tmp -@ 4 \
```

```
> workdir/$input_file/hisat_out_bam) \
> workdir/$input_file/hisat_map_info 2>&1

stringtie
    workdir/$input_file/hisat_out_bam --rf -p 4 \
    -o workdir/$input_file/transcript.gtf \
    -G  genome_annotation/Araport11_GFF3_genes_transposons.May2022.gff \
    -e -b workdir/$input_file
```

```
prepDE.py \
    -i smlist \ #包含所有SRR序列号与.gtf输出的地址
    -l 126 #read平均长度
```