第九章　RNA-seq 实验与分析

基于下一代测序的 **RNA 测序** (RNA sequencing, RNA-seq) 能够揭示 RNA 的存在和数量，本身是一种非常强大的技术。而实验方法的不断改进和浩浩荡荡的的商业化过程又使得 RNA-seq" 飞入寻常百姓家"，不再复杂神秘。

近年来，由于某些热门领域的影响，掌握一定的测序数据分析技能已经成为了一种时尚，而如果能绘制一点图片就更好了。因此，培训班和教程层出不穷，它们是更好的学习资源。

但即使不期望在信息学上有所洞见，也不希图能够构建或读懂软件，只是学习一点实验原理和流程想必也是有好处的。抱着这样的想法，本章先介绍下一代测序、文库构建实验，然后完成一个完整的生物信息学分析流程和结果绘图。

本章的代码是课程助教提供的，有修改。

9.1　下一代测序

长核酸序列测定的首个实用方法由 Sanger 在 1975 年建立，称为“加减法”[1]；1977 年，Maxam 和 Gilbert 发明了化学法，其基本原理是用特异化学试剂[2]作用于 4 种碱基，使末端标记的 DNA 分子被切成不同长度的末端都是特异的碱基的片段，变性电泳后读出序列。同年，Sanger 等对于“加减法”作出重要改进，提出了“终止法”，即以双脱氧核苷三磷酸在随机位置终止反应，而后以变性凝胶电泳直接读出 DNA 序列的方法，这是第一代测序技术的基本原理。

下一代测序（Next Generation Sequencing, NGS）是相对于第一代测序技术而言的，尽管这一概念已经提出十几年，但它仍然是描述高通量测序的流行语。2006 年，Genome Analyzer（第一代 Solexa 测序仪）上市，随后主宰了高通量测序市场，次年，Illumina 收购了 Solexa。Illumina/Solexa 测序 (以下简称 Illumina 测序) 的基本原理是：桥式扩增（bridge amplification）和循环可逆终止（cyclic reversible termination, CRT）测序。Illumina 提供了形象的讲解视频，清晰易懂。

9.1.1　桥式扩增

测序载玻片上共价连接了很多正向和反向引物 (图 9.1 中红色和蓝色短片段)，密度很高。测序文库被变性后与这些固定的引物退火，开始第一轮延伸。第一个循环完毕后，即得到一

[1] 详见参考文献 [1]、[2]。

[2] 详见参考文献 [1]。

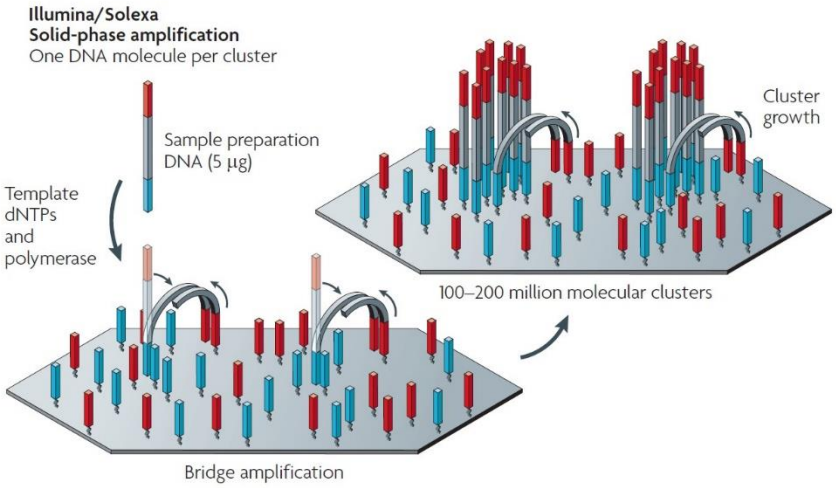

图 9.1: 桥式 PCR 示意图。引自参考文献[4]。

条固定在测序载玻片上的文库片段，该文库片段再经变性，其 3′ 端与被固定的引物退火-延伸（延伸产物形似桥），又得到第二条固定在载玻片上的文库片段。如此反复进行多轮桥式 PCR，得到很多片段簇（cluster），一个簇中包含很多序列相同的带相同接头的片段。

桥式 PCR 使测序模板被固定在了一个固相表面上，同时，不同的片段被分散在测序载玻片上，成很多空间上相互分隔的簇，这使高通量的检测成为可能。

9.1.2　循环可逆终止

循环可逆终止（CRT）包括 3 个步骤：核苷酸掺入、荧光成像、切除荧光染料基团和终止基团。

第一步：聚合酶在测序引物后添加 1 个核苷酸，由于核苷酸终止基团的存在，聚合酶无法添加第 2 个核苷酸。

第二步：洗去未被掺入的核苷酸，荧光成像，记录。

第三步：切除被掺入的核苷酸的荧光染料基团，移除终止基团解放 3′-OH，洗去残余试剂，准备下一次掺入。

如何做到循环可逆终止是一个关键问题，Illumina 测序使用了染料标记的修饰核苷酸（dye-labelled modified nucleotides）。如右图所示。红色基团代表 3’ OH 的终止基团 (3′-*O*-叠氮甲基，3′-*O*-azidomethyl，可使用还原剂 Tris(2-carboxyethyl)phosphine (TCEP) 来释放 3′-
OH)，dye 代表了荧光染料基团，蓝色部分代表一段 linker，黑色箭头指示了在第三步切割位置。

Illumina 测序在每次测序循环中使用 4 种染料分别标记 4 种核苷酸，因此每次循环将得到一张 4 色图像，如下图所示。

9.1.3　Illumina 测序平台

十几年来，Illumina 推出了多种型号的 NGS 测序平台，它们在原理上大同小异，但又各有所长。最早的测序平台是 Genome Analyzer (GA)，目前被广泛采用的测序平台包括：HiSeq、

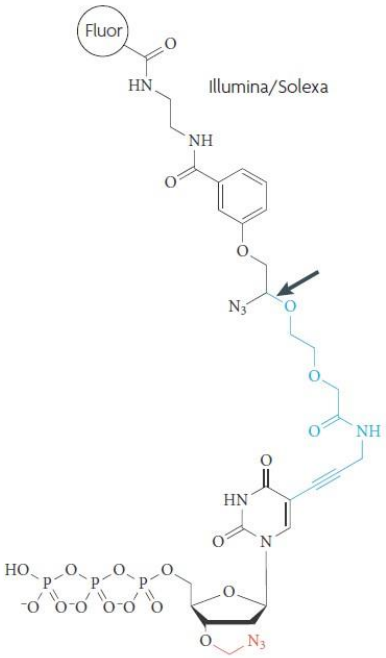

图 9.2: 染料标记的修饰核苷酸

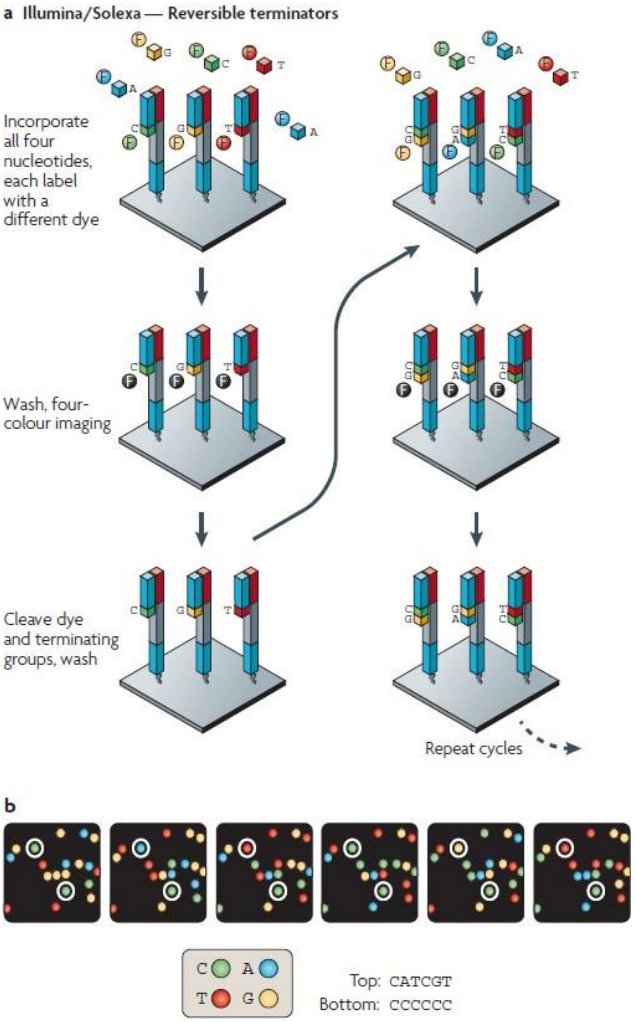

图 9.3: Illumina 采用的四色循环可逆终止方法。引自参考文献 [4]。

HiSeq X Ten、MiSeq、NextSeq 和NovaSeq。

其中，NovaSeq 发布于2017 年，有两个较大的改进：

- 可运行 S1、S2、S3 和 S4 四种不同类型的图案化流动槽，通量上面具有较大的灵活性，运行一次可产生 0.5-6 Tb 数据, 但是一条 lane 的数据量较大，增加了 pooling 和数据拆分的难度
- 采用改进的2色边合成变成测序 (SBS) 化学 (T=绿色, C=红色, A=绿色/红色, G=无信号) 和 3 代实时分析 (Real-Time Analysis) 系统，读写速度更快

具体的技术细节不是课程的核心内容，更加详细的介绍可以访问 Illumina 公司或测序服务提供商的网页。

9.2 RNA-seq

9.2.1 一般原理

RNA 测序（RNA sequencing, RNA-seq）出现于 2008 年，由多个研究组同时发表。时至今日，RNA-seq 已经发展出多种变体，无论方法如何变化，使用 NGS 的 RNA-seq 都需要用逆转录酶将 RNA 转换成DNA。

逆转录酶发挥作用需要引物，根据使用随机引物或 Oligo dT 引物，RNA-seq 分为两个分支。随机引物可能结合到任何 RNA 的任何位置，因此一般用与测序总 RNA；Oligo dT 结合到 mRNA 的 polyA 尾，一般用于测序 mRNA，即转录本。

在得到 cDNA 以后，下一个技术上的问题是，从样品中提取的 RNA 并不多，因此 cDNA 的量也较少，在测序之前，先要进行扩增。通用的扩增方法是聚合酶链式反应（PCR），然而，RNA 及其对应的 cDNA 序列多种多样，无法一一设计引物（事实上也不可能）。所以，需要在扩增之前为所有 RNA 添加统一的两种接头。

但此时又有新的问题，即 NGS 读长短，但 RNA 较长（m RNA 长度一般超过 1000bp），无法一次测得，因此需要被破碎。

综合考虑以上这些问题，技术开发者提出的方案是（mRNA 测序）：

1. 使用连有 Oligo dT 的磁珠富集带有 PolyA 尾的 mRNA
2. 使用超声打断 mRNA
3. 使用 Oligo dT 引物逆转录 mRNA，合成 cDNA 第一链
4. 合成 cDNA 第二链
5. 使用连接酶为 cDNA 两端加上不同的接头
6. PCR 扩增文库
7. 上机测序

很明显，这个流程有些问题。因为超声打断在逆转录之前，故只有 mRNA3‘端最接近 polyA 尾的一段能够被测序，因此，该流程只能对转录本进行大致的定量，却忽略选择性剪接等需要全长转录本信息才能了解的变异。

聪明的读者可以自己思考构建全长文库的方法。

技术本身更新换代很快，这是一件令人高兴又叹惋的事情。一旦新的替代品出现，旧的技术就会无人问津，成为一些文章 *Introduction* 部分论资排辈报出的菜名。研究老的方法，宛如考古。

目前，最广泛采用的文库构建解决方案是 Illumina Truseq RNA 建库方法，可以称为"事实上的"标准方法，其基本步骤如图 9.5 所示。

9.2.2　文库的结构

上述"标准方法"的接头连接步骤细节及构建的文库结构如图 9.4 所示。完整文库结构包括"P5-Index2-Rd1 SP-DNA Insert-Rd2 SP-Index1-P7"。

其中，Index1 与Index2 是文库的**标签**，为了多路复用（在一次上机，一个芯片中同时测多个文库）而设计，所有的文库都具有相同的结构，但由于其 Index 不同，因此数据下机后可按 Index 进行拆分。

P5 和 P7 是**流动槽结合接头**，位于文库的两端，因此也是 PCR（无论是为了扩增而进行的 PCR，还是为了定量而进行的 qPCR）使用的引物。上一节提到，NGS 需要先进行桥式扩增形成序列簇，序列簇正是靠这一段序列首次结合到固定在流动槽底部的接头上的。

Rd1 SP 和 Rd2 SP 分别是 Read 1 Sequencing Primer 和 Read 2 Sequencing Primer 的缩写，与真正的测序引物序列相同，因此测序引物可以结合在此序列的互补序列上，进行测序反应，最终会得到 Read1 和 Read2 的数据。

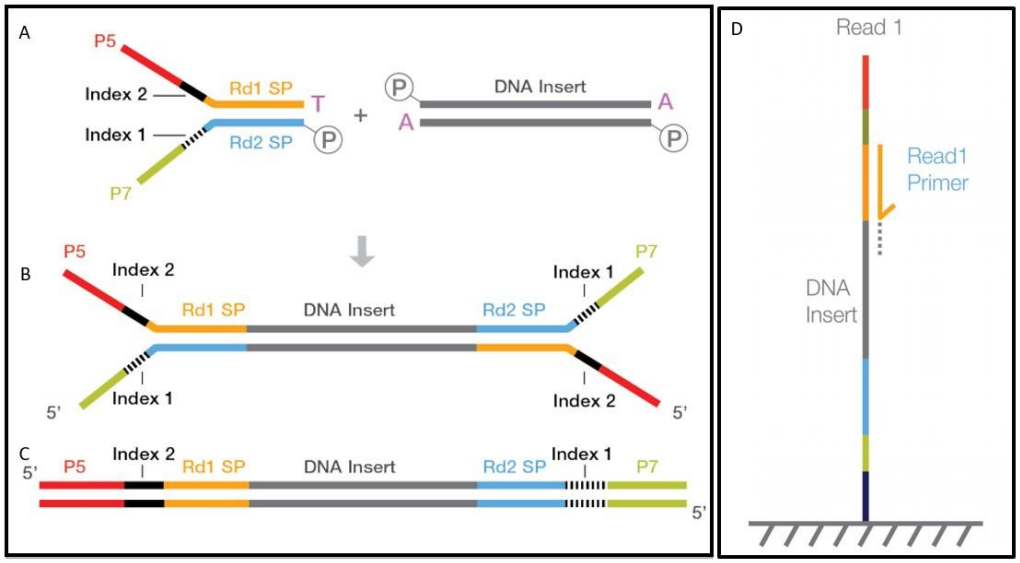

图 9.4: 接头连接和文库结构。接头连接依赖于末端加 A 和磷酸化构成的黏性末端。引自参考文献 [13]

9.2.3　方法变体

许多年过去，RNA-seq 已经发展出很多变体，这些变体往往服务于某种特定的需求，常常具有较为有趣的缩写。下面给出一些方法，有兴趣的读者可以自行了解。

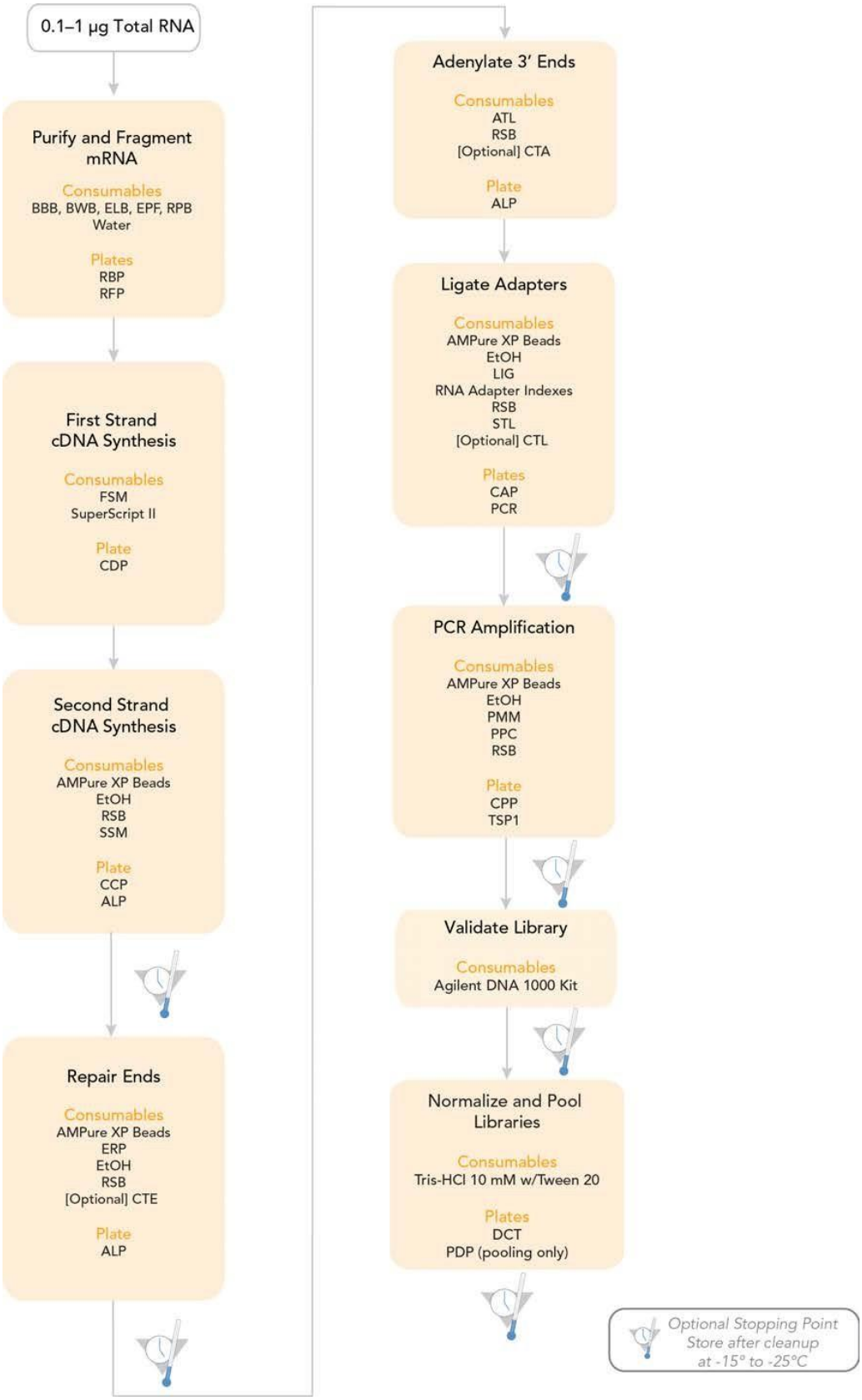

图 9.5: Illumina TruSeq RNA Library Preparation v2 基本流程。引自参考文献 [12]

针对 RNA-seq 本身进行优化，或使它在转录本检测方面具有更多功能的变体：TAIL-seq, PAL-seq, GRO-seq, PRO-seq, CAGE。

为研究 RNA-蛋白质相互作用而开发的变体方法：Ribo-seq, CLIP-seq, eCLIP, DLAF, CLASH。

为鉴定或研究RNA 上的化学修饰而开发的变体方法：MeRIP-seq, PSI-seq, Pseudo-seq, ICE。

为研究 RNA 结构而开发的变体方法：SHAPE-seq, PARS-seq, Cap-seq。

低起始量 RNA 建库方法：TRACE-seq, scRNA-seq, CEL-seq, Smart-seq, Smart-seq2, Smart-seq3, Drop-seq。

9.3 概述和准备工作

9.3.1 数据来源及实验设计

本分析示例数据来源于文献 Zhao, Y., Zhang, Z., Gao, J., Wang, P., Hu, T., Wang, Z., Hou, Y. J., Wan, Y., Liu, W., Xie, S., Lu, T., Xue, L., Liu, Y., Macho, A. P., Tao, W. A., Bressan, R. A., & Zhu, J. K. (2018). Arabidopsis Duodecuple Mutant of PYL ABA Receptors Reveals PYL Repression of ABA-Independent SnRK2 Activity. *Cell reports*, 23(11), 3340 - 3351.e5.。

这是一项植物生理学研究工作，我们重点关注其部分 RNA-seq 实验，在 SRA 中对应的实验编号为：SRP145580，该实验研究的是 *pyl112458* 和 *379101112* 突变体对脱落酸和渗透胁迫的基因表达反应。

该实验的整体设计是：野生型和突变体幼苗在处理前均垂直生长 9 天，分别进行：不处理、100 μM ABA 处理、在含 300mM 甘露醇的滤纸上处理 24 小时。转录组用 HiSeq2500 平台进行深度测序进行分析，每组进行三个重复。

几乎所有在野生型籽苗中响应 ABA 的基因表达反应在上述两种突变体中都减弱了。然而响应渗透胁迫的的转录本水平变化在突变体中却很少消失，大多数的响应仍然保持着，即使突变体中很多转录本的上调和下调幅度都减弱了。

在SRA Run Selector页面我们可以找到每一份数据对应的实验条件和基本信息(包括 SRR 号，GEO 登录号，碱基数，基因型，处理条件)，这里只节录分析时用到的 Run。

Run	Bases	Genotype	GEO_Accession	Treatment
SRR7160928	5.27 G	wildtype	GSM3140728	300 mM mannitol for 24 hours
SRR7160929	4.80 G	wildtype	GSM3140729	300 mM mannitol for 24 hours
SRR7160930	5.03 G	wildtype	GSM3140730	300 mM mannitol for 24 hours
SRR7160931	4.58 G	wildtype	GSM3140731	1/2 MS for 24 hours
SRR7160932	5.00 G	wildtype	GSM3140732	1/2 MS for 24 hours
SRR7160933	4.71 G	wildtype	GSM3140733	1/2 MS for 24 hours

上表中可见，6 份 RNA-seq 数据中，SRR7160928、SRR7160929，SRR7160930 是野生型幼苗渗透胁迫处理，SRR7160931、SRR7160932、SRR7160933 是野生型幼苗对照组。

据此，我们预期最后分析得到的差异表达基因应该富集在渗透胁迫反应功能上。

9.3.2　配置环境

为了不污染服务器环境，应该在虚拟环境中进行后来的操作。关于建立虚拟环境，有多种方案，此流程中使用 conda ，安装 conda 的方法请参考第一章。

执行下面的命令创建和激活一个名为 rnaseq 的环境，并安装需要用到的软件。

```
conda create -n rnaseq
conda activate rnaseq
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
```

9.3.3　下载参考文件和测序数据

新建 ./genome 目录，在其中下载参考文件：

```
# 参考基因组TAIR10
wget -b -c http://ftp.ensemblgenomes.org/pub/plants/release-54/fasta/
    arabidopsis_thaliana/dna/Arabidopsis_thaliana.TAIR10.dna.toplevel.
    fa.gz
# GTF文件
wget -b -c http://ftp.ensemblgenomes.org/pub/plants/release-54/gtf/
    arabidopsis_thaliana/Arabidopsis_thaliana.TAIR10.54.gtf.gz
# GFF3文件
wget -b -c http://ftp.ensemblgenomes.org/pub/plants/release-54/gff3/
    arabidopsis_thaliana/Arabidopsis_thaliana.TAIR10.54.gff3.gz

# 选项
# -c 设置wget自动断点续传
#  -b启动下载后转入后台执行
```

下载完毕后，使用 gunzip 命令解压缩。

接下来还要获得接头和引物序列，根据实际使用的引物、文献列出的引物或者测序平台提供的引物确定，这里已经有现成的文件 Truseq-PE.fa ：

```
>PrefixPE/1
AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT
>PrefixPE/2
CAAGCAGAAGACGGCATACGAGATCGGTCTCGGCATTCCTGCTGAACCGCTCTTCCGATCT
>PCR_Primer1
AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT
>PCR_Primer1_rc
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT
>PCR_Primer2
CAAGCAGAAGACGGCATACGAGATCGGTCTCGGCATTCCTGCTGAACCGCTCTTCCGATCT
>PCR_Primer2_rc
AGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGATCTCGTATGCCGTCTTCTGCTTG
>FlowCell1
TTTTTTTTTTAATGATACGGCGACCACCGAGATCTACAC
>FlowCell2
TTTTTTTTTTCAAGCAGAAGACGGCATACGA
```

接下来下载测序数据文件：

```
SRR7160928_1.fastq SRR7160929_1.fastq SRR7160930_1.fastq SRR7160931_1.
    fastq SRR7160932_1.fastq SRR7160933_1.fastq
SRR7160928_2.fastq SRR7160929_2.fastq SRR7160930_2.fastq SRR7160931_2.
    fastq SRR7160932_2.fastq SRR7160933_2.fastq
```

根据 SRR 编号，指定下载路径，用 `wget` 下载到 `./data` 目录下，此处使用循环下载所有的文件：

执行以下脚本：

```
#! /bin/bash
mkdir data
for i in {28..33}
do
    access="SRR71609${i}"
    wget -c -b -O ./data/${access}.sra "https://sra-pub-run-odp.s3.
        amazonaws.com/sra/${access}/${access}"
done

# 选项
# -O 指定下载文件所在的文件夹和文件名
```

注意到，下载的文件是 SRA 格式（Sequence Read Archive），我们需要将其转换为 `fastq` 格式，并拆分为 Read1 与 Read2。这里需要用到 SRA Toolkit，建议手动安装并配置路径，也可以使用 `conda` 安装。随后，执行下面的脚本转换格式并拆分数据：

```
#! /bin/bash
for i in {28..33}
do
    access="SRR71609${i}"
    fastq-dump --split-3 -O . data/${access}.sra &
done

# fastq-dump 是sra-tools中的程序，可用于下载或拆分SRA数据，这里只用它来拆分数据
# --split-3 将双端测序分为两份，放在不同的文件，但是对于一方有而一方没有的reads会单独放在一个文件夹里
# -O指定输出的文件夹
# data/${access}.sra 要拆分的文件
```

此时，文件目录为：

```
.
    data
        SRR7160928_1.fastq
        SRR7160928_2.fastq
        SRR7160928.sra
        SRR7160929_1.fastq
        SRR7160929_2.fastq
        SRR7160929.sra
        SRR7160930_1.fastq
        SRR7160930_2.fastq
        SRR7160930.sra
        SRR7160931_1.fastq
        SRR7160931_2.fastq
        SRR7160931.sra
        SRR7160932_1.fastq
        SRR7160932_2.fastq
        SRR7160932.sra
        SRR7160933_1.fastq
        SRR7160933_2.fastq
        SRR7160933.sra
```

```
21      genome
22          Arabidopsis_thaliana.TAIR10.54.gff3
23          Arabidopsis_thaliana.TAIR10.54.gtf
24          Arabidopsis_thaliana.TAIR10.dna.toplevel.fa
25          Truseq-PE.fa
```

得到这些文件以后，就可以开始分析了。

9.4 RNA-seq 分析 I

RNA-seq 的分析，一般分为所谓“上游”和“下游”，前者指在 Linux 终端进行的，使用不同的工具从测序数据产生表达计数矩阵的分析过程；后者指拿到表达计数矩阵后，用 R 或 Python 等脚本语言进行绘图的分析过程。这里只是一提，区分这些概念没有什么意义，只需全部完成一遍即可。

9.4.1 序列修剪

首先创建工作目录：`mkdir workdir`，并在其中为每个实验组创建子目录。目录结构如下：

```
1 workdir/
2     SRR7160928
3     SRR7160929
4     SRR7160930
5     SRR7160931
6     SRR7160932
7     SRR7160933
```

第一步是序列修剪，因为测序平台的不同，测序的接头的序列也会不同，用户需要自己指定所使用的接头种类（可能污染的接头种类）。

常用的去接头软件包括 `trimmomatic` 和 `cutadapt`，这里使用前者，并以一个样品结果的处理为例详细解读。

`trimmomatic` 有两种修剪，第一种，如果一个序列来源于技术而非样品，显然是应该去除的 (这种修剪较为复杂，分为简单模式和回文模式，后者适合于检出“接头通读”式 reads)；第二种，如果一个序列质量太差，则即使它来源于样品，也是应该去除的 (通过滑动窗口过滤和最大信息过滤)。

```
1 trimmomatic PE -threads 9 data/SRR7160928_1.fastq data/SRR7160928_2.
      fastq -baseout workdir/SRR7160928/SRR7160928 ILLUMINACLIP:./
      genome_annotation/Truseq-PE.fa:2:30:10 LEADING:5 TRAILING:5
      SLIDINGWINDOW:5:20 MINLEN:36 AVGQUAL:20
```

```
2
3  # 选项
4  #  PE 指定双端测序模式
5  # -threads 8 指定线程数
6  # data/SRR7160928_1.fastq data/SRR7160928_2.fastq 指定两个输入文件
7  #  -baseout  workdir/SRR7160928/SRR7160928  指定输出文件的存储路径
       和基本文件名
8  #   ILLUMINACLIP:./genome_annotation/Truseq-PE.fa:2:30:10   下文详细解释
9  #  LEADING:5  设定碱基质量阈值，从reads起始端开始切除质量低于此的碱基
10 #  TRAILING:5  设定碱基质量阈值，从reads末端开始切除质量低于此的碱基
11 #  SLIDINGWINDOW:5:20  滑动窗口剪切，第一个值设定窗口大小，第二个值设定平
       均质量阈值，这里检测整个reads，如某个窗口中平均碱基质量低于20，则将这
       个窗口整体切除
12 # MINLEN:36 设定reads长度阈值，如果剪切后reads长度低于此则丢弃整条reads
13 #  AVGQUAL:20  设定reads平均碱基质量阈值，如果剪切后reads平均碱基质量低于
       此则丢弃整条reads
```

ILLUMINACLIP 选项中，第一个参数 ./genome_annotation/Truseq-PE.fa 代表要过滤的接头和引物序列，因为这些序列并不是来自于生物样品，而是实验过程中引入的。

类似于一些序列比对软件，trimmomatic 先将接头和引物序列片段切成种子，与 reads 进行比对；得到匹配良好的词对后（如 BLAST 中的 HSP），再进行全长比对。

第二个参数代表种子搜索允许的错配碱基数，这里设置为 2。

第三个参数针对 PE 的回文修剪模式[3]下，需要 R1 和 R2 之间至少多少比对分值，才会进行接头切除，这里取 30。

第四个参数是切除接头序列的最低比对分值，这里设置为 10。为了对每个样品都进行一样的处理，可使用循环：

```
1 #! /bin/bash
2 for i in {28..33}
3 do
4     access="SRR71609${i}"
5     trimmomatic PE -threads 8 data/${access}_1.fastq data/${access}_2.
          fastq -baseout workdir/${access}/${access} ILLUMINACLIP:./
          genome/Truseq-PE.fa:2:30:10 LEADING:5 TRAILING:5 SLIDINGWINDOW
          :5:20 MINLEN:36 AVGQUAL:20 &
6 done
```

trimmomatic 将输出一些运行信息，如下所示，稍作解释：

[3]请阅读参考文献 [1][2][3]

```
# 命令
TrimmomaticPE: Started with arguments:
-threads 56 data/SRR7160928_1.fastq data/SRR7160928_2.fastq -baseout workdir/SRR7160928/SRR7160928 ILLUMINACLIP:./genome_annotation/Truseq-PE.fa:2:30:10 LEADING:5 TRAILING:5 SLIDINGWINDOW:5:20 MINLEN:36 AVGQUAL:20
# 输出文件，其中P代表Paired，即两端都保留的reads，U代表Unpaired，即仅有一段保留者
Using templated Output files: workdir/SRR7160928/SRR7160928_1P workdir/SRR7160928/SRR7160928_1U workdir/SRR7160928/SRR7160928_2P workdir/SRR7160928/SRR7160928_2U
# 欲切除的引物和接头
Using PrefixPair: 'AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT' and 'CAAGCAGAAGACGGCATACGAGATCGGTCTCGGCATTCCTGCTGAACCGCTCTTCCGATCT'
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT'
Using Long Clipping Sequence: 'AGATCGGAAGAGCGGTTCAGCAGGAATGCCGAGACCGATCTCGTATGCCGTCTTCTGCTTG'
Using Long Clipping Sequence: 'TTTTTTTTTTAATGATACGGCGACCACCGAGATCTACAC'
Using Long Clipping Sequence: 'TTTTTTTTTTCAAGCAGAAGACGGCATACGA'
Using Long Clipping Sequence: 'CAAGCAGAAGACGGCATACGAGATCGGTCTCGGCATTCCTGCTGAACCGCTCTTCCGATCT'
Using Long Clipping Sequence: 'AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT'
ILLUMINACLIP: Using 1 prefix pairs, 6 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
#  trimmomatic会自动识别碱基质量编码方式，因此无需在命令中指定
Quality encoding detected as phred33
# 给出简单的统计结果，两端保留的，前向保留的，反向保留的，丢弃的reads的条目和占比
Input Read Pairs: 20924237 Both Surviving: 18744203 (89.58%) Forward Only Surviving: 1832502 (8.76%) Reverse Only Surviving: 128862 (0.62%) Dropped: 218670 (1.05%)
TrimmomaticPE: Completed successfully
```

此时，workdir 目录的结构为：

```
workdir/
```

```
2     SRR7160928
3        SRR7160928_1P
4        SRR7160928_1U
5        SRR7160928_2P
6        SRR7160928_2U
7     SRR7160929
8        SRR7160929_1P
9        SRR7160929_1U
10       SRR7160929_2P
11       SRR7160929_2U
12    SRR7160930
13       SRR7160930_1P
14       SRR7160930_1U
15       SRR7160930_2P
16       SRR7160930_2U
17    SRR7160931
18       SRR7160931_1P
19       SRR7160931_1U
20       SRR7160931_2P
21       SRR7160931_2U
22    SRR7160932
23       SRR7160932_1P
24       SRR7160932_1U
25       SRR7160932_2P
26       SRR7160932_2U
27    SRR7160933
28       SRR7160933_1P
29       SRR7160933_1U
30       SRR7160933_2P
31       SRR7160933_2U
```

trimmomatic 输出的文件中，后缀为 1P 的是正向配对 reads，后缀为 1U 的是正向未配对 reads，后缀为 2P 的是反向配对 reads，后缀为 2U 的是反向未配对 reads。

9.4.2 质量控制

9.4.2.1 fastq 文件结构

当前数据文件是以 fastq 文件格式保存的，其中的 q 即代表 Quality，它比 fasta 更有用的地方就在于整合了碱基质量的信息，这也是质量控制的基础。因此，了解一点 fastq 文

件的结构是很有必要的。

fastq 文件中，一个序列由 4 行构成，我们以 SRR7160928_1.fastq 的最后 4 行为例：

```
1 @SRR7160928.20924237 20924237 length=126
2 CTGAATCACAGGGATAGTGTGAAGATCGACCAAAGTGAATTTATCAGAAGCCAAATACTTGGACTCAC
      CAAGCCTGTGTTCGTAAACATCGAGGACCTTGGCTAGCTTAGCCTCTTCTTCTTCAAC
3 +SRR7160928.20924237 20924237 length=126
4 BBBB/B/F/FFFFFFFFFFFFFFFF<FF/<//FF<FFFFFFFFFFFFFF/<FFFFFFFFF/BBBF<<
      FF/<<<FFBBFFFF<FFFFFF/BFFB//FFFF/BF/FFFF//FFBBBFFFB<FFFF
```

- 第一行以 @ 开头，之后为序列的标识符以及描述信息（与 FASTA 格式的描述行类似）
- 第二行为序列信息
- 第三行以 + 开头，之后可以再次加上序列的标识及描述信息（可选）
- 第四行为质量得分信息，与第二行的序列相对应，长度必须与第二行相同

这里面最复杂的是碱基质量得分信息，它有两种编码方式：phred33 和 phred64 （碱基质量得分是由 phred 程序开发者定义的）。相比于这些字符编码，更加直观的碱基质量表示是**碱基错误率**（P_{error}）（因为 CCD 捕捉的信号荧光信号并不永远清晰易分辨），错误率可以从信号本身的特征计算出来。

碱基错误率 (P_{error}) 与碱基质量得分 (Q) 的关系是：

$$Q = -10\log_{10} P_{error}$$

如果 $P_{error} = 1$，则 $Q = 0$；如果 $P_{error} = 0figure.001$，则 $Q = 30$。

出于节约存储空间的考虑，需要将可能的碱基质量得分对应于单个符号，而非像 30 这样的两位数字。

美国信息交换标准代码（American Standard Code for Information Interchange, ASCII）就是一个简便的方案，ASCII 是一套字符编码方案，1963 年被作为通过电话线传送文本信息的标准而发明，其单个字符由 7 个 bit 表示，因此总共可以表示 $2^7 = 128$ 个字符，编号为 0-127。修订后的 ASCII 表如下：

表中，0-31 号字符和 127 号字符控制码，用于控制电传打字机，无法在屏幕上显示。剩余的字符是可打印字符，包括大写字母、小写字母、数字和基本标点。

像上面所说的，Q 的取值一般在 0-几十这个范围活动，如果直接对应，很多质量值都会对应到控制码，无法显示。因此，必须为 Q 加上某个数值，使得可能的取值都能对应上可打印字符。phred33 方案选择 $Q = Q + 33$，而 phred64 方案选择 $Q = Q + 64$。

如上面例子中的碱基质量值 F ，查 ASCII 表知，F 编号为 70，又由于该文件采用 phred33方案，则原始 Q 值为 $70 - 33 = 37$。则对应的 $P = 10^{-10} = 10^{-3.7} \approx 0.0002$。碱基错误率仅仅为 0.02%，其质量是非常好的。

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL (null)	32	SPACE	64	@	96	`
1	SOH (start of heading)	33	!	65	A	97	a
2	STX (start of text)	34	"	66	B	98	b
3	ETX (end of text)	35	#	67	C	99	c
4	EOT (end of transmission)	36	$	68	D	100	d
5	ENQ (enquiry)	37	%	69	E	101	e
6	ACK (acknowledge)	38	&	70	F	102	f
7	BEL (bell)	39	'	71	G	103	g
8	BS (backspace)	40	(	72	H	104	h
9	TAB (horizontal tab)	41	)	73	I	105	i
10	LF (NL line feed, new line)	42	*	74	J	106	j
11	VT (vertical tab)	43	+	75	K	107	k
12	FF (NP form feed, new page)	44	,	76	L	108	l
13	CR (carriage return)	45	-	77	M	109	m
14	SO (shift out)	46	.	78	N	110	n
15	SI (shift in)	47	/	79	O	111	o
16	DLE (data link escape)	48	0	80	P	112	p
17	DC1 (device control 1)	49	1	81	Q	113	q
18	DC2 (device control 2)	50	2	82	R	114	r
19	DC3 (device control 3)	51	3	83	S	115	s
20	DC4 (device control 4)	52	4	84	T	116	t
21	NAK (negative acknowledge)	53	5	85	U	117	u
22	SYN (synchronous idle)	54	6	86	V	118	v
23	ETB (end of trans. block)	55	7	87	W	119	w
24	CAN (cancel)	56	8	88	X	120	x
25	EM (end of medium)	57	9	89	Y	121	y
26	SUB (substitute)	58	:	90	Z	122	z
27	ESC (escape)	59	;	91	[	123	{
28	FS (file separator)	60	<	92	\	124	|
29	GS (group separator)	61	=	93	]	125	}
30	RS (record separator)	62	>	94	^	126	~
31	US (unit separator)	63	?	95	_	127	DEL

图 9.6: ASCII 表

9.4.2.2 fastqc 实践

进行质量控制的脚本如下：

```
#! /bin/bash
for i in {28..33}
 do
    input_file="SRR71609${i}"
    fastqc -t 4 workdir/${input_file}/${input_file}_1P
       workdir/${ input_file}/${input_file}_2P -o
       workdir/${input_file}/
```

参数较为简单，-t 指定线程数，-o 指定输出目录，还需要给程序提供一个实验组的 fastq 文件（即 read1 和 read2）。

fastqc 的结果以 HTML 文件的形式呈现，下载到本地使用浏览器查看；如果想要自行绘图，则需要在 fastqc.zip 中寻找所需的数据段。

9.4.2.3 fastqc 结果

下面以 SRR7160928_1P_fastqc.html 为例，解读 fastqc 结果 (打开 HTML 文件)，报告分为 10 个部分，在左侧有导航栏和图标提示（绿色为通过，黄色为警告，红色为未通过）：

1. 基本统计量 (BasicStatistics)

如图 9.7，该部分给出被质控数据的基本信息和统计值。包括：

- 文件名
- 文件类型
- 质量值编码方式
- 总序列数：处理的序列总数的计数
- 被标记的低质量序列的计数
- 序列长度：提供集合中最短和最长序列的长度，如果所有序列的长度相同，则只报告一个值
- %GC：所有序列中所有碱基的总体 GC 百分

比该部分永远不会发出警告或报告不通过。

2. 每个碱基序列质量 (Per base sequence quality)

如图 9.8，该部分显示 FASTQ 文件中每个位置的所有碱基的质量值范围，x 轴代表碱基在 read 中的位置，y 轴代表碱基的质量分数，分数越高质量越好。FastQC 将自动检测数据使用的碱基质量编码方法，无需用户指定。

图中，红线是碱基质量中位数，黄色框代表四分位数范围（25-75%），上下横线分别代表 10% 和 90% 点，蓝色曲线代表平均质量。

底纹中，红色区域代表质量差，橙色区域代表质量尚可，绿色区域代表质量很好。

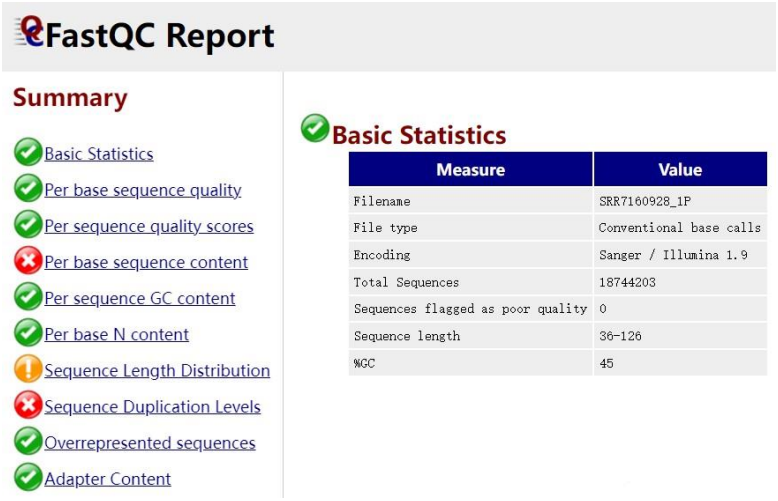

图 9.7: FastQC 报告 (1)-基本统计。

一般而言，测序的质量会随着在 read 上位置推后而下降。本例测序数据质量很好。蓝线一直位于绿色部分。

如果任何碱基质量的下四分位数小于 10，或任何碱基质量的中位数小于 25，将发出警告。如果任何碱基质量的下四分位数小于 5 或 任何碱基质量的中位数小于 20，将不予通过。

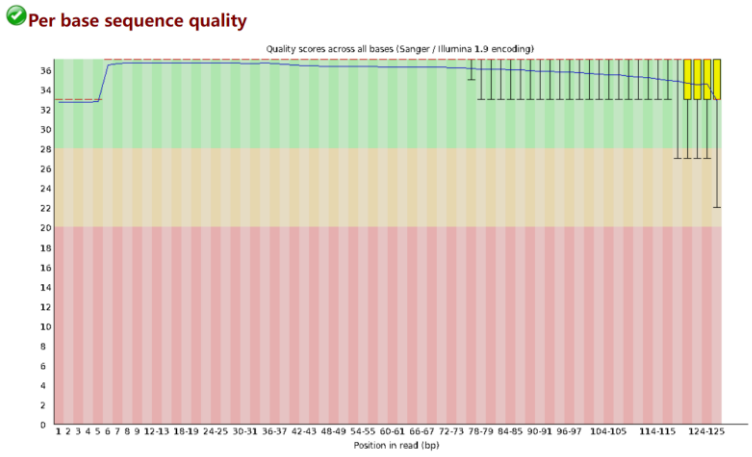

图 9.8: FastQC 报告 (2)-每碱基序列质量。

3. 每条序列质量分数 (Per sequence quality scores)

如图 9.9，该部分显示 FASTQ 文件中序列的质量分布。横轴代表每一条序列中碱基的平均质量分数，纵轴代表有多少碱基获得了该分数。该图还会标出所有序列的平均质量分数的平均值。

如果该值低于 27——这相当于 0.2% 的错误率，则会发出警告；如果低于 20——这相当于 1% 的错误率，则不予通过。

本例中该值高达 35-36，测序数据的整体质量是很好的。

4. 每个碱基序列组成 (Per base sequence content)

如图 9.10，该部分给出 reads 中每个位置的碱基组成，四种颜色分别代表四种碱基的百分比。

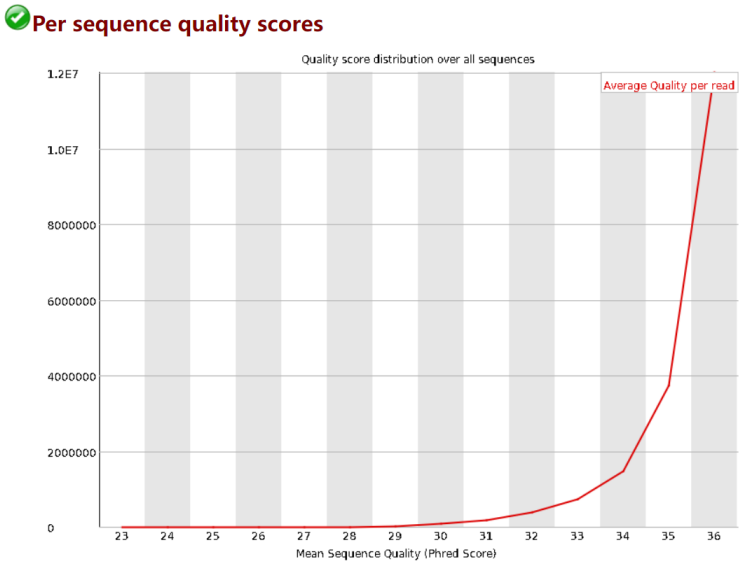

图 9.9: FastQC 报告 (3)-每条序列的质量分数。

如果一个文库内容是随机的，不同碱基的占比应该从头到尾都是相同的。真实情况可能有些波动，但绝不应该产生巨大差距。

本例中，reads 的开头部分各种碱基的比例差别较大。这或许是技术造成，某些类型的文库总是会产生有偏差的序列组成，通常在读取开始时。使用随机六聚体引发的文库（包括几乎所有的 RNA-Seq 文库）和使用转座酶片段化的文库读取开始位置会有内在偏差。

如果 A 和 T 或 G 和 C 之间的差异在任何位置大于 10%，此模块会发出警告；如果大于 20 %，则不予通过。

本例中，数据未通过这个检验，这或许是因为片段化过程引入的偏差（bias），这无伤大雅。

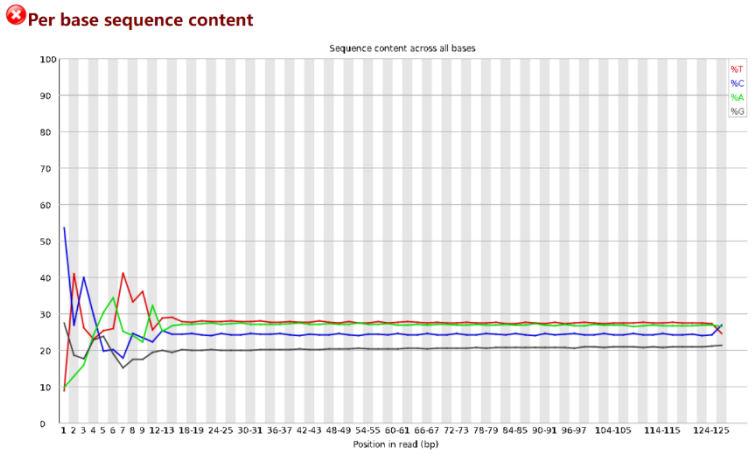

图 9.10: FastQC 报告 (4)-每碱基序列含量。

5. 每条序列 GC 含量 (Per sequence GC content)

如图 9.11，该部分描述所有序列的 GC 含量情况。x 轴代表 GC 百分比，纵轴代表有多少序列的 GC 含量在该百分比。

在正常的随机文库中，用户会期望看到大致正态分布的 GC 含量，其中中心峰对应于基础基因组的整体 GC 含量。由于我们不知道基因组的 GC 含量，因此理论曲线 GC 含量是根据观察到的数据计算的，并用于构建参考分布。

如果实际分布与正态分布的偏差总和超过 15%，则会发出警告; 如果超过 30% 的读数，则不予通过。

本例中，实际 GC 含量分布与正态分布符合得较好。

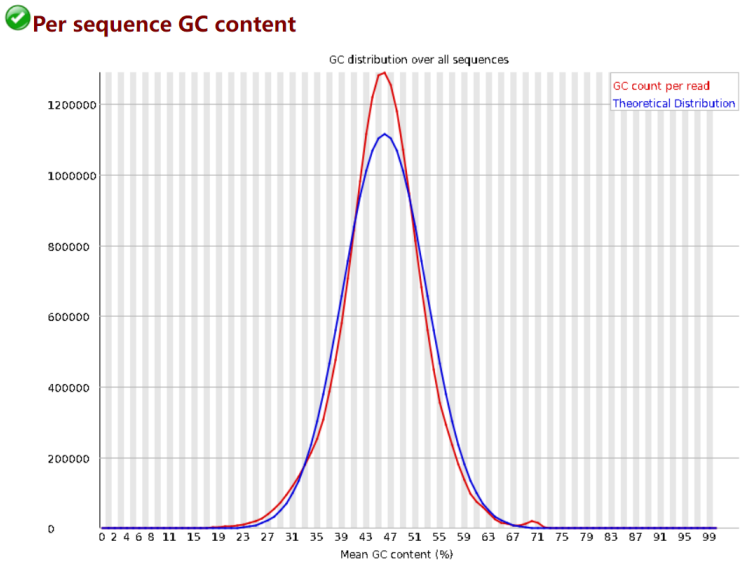

图 9.11: FastQC 报告 (5)-每条序列 GC 含量。

6. 每个碱基位置的 N 含量 (Per base N content)

如图 9.12，该图表示在每个位置 N 碱基的百分比。当测序仪无法就某个位置作出明确判断时，就会将其记为 N。

如果任何位置显示 N 含量 >5%，此模块会发出警告；如果 >20%，将不予通过。大量的 N 代表着该数据整体质量低，在本例中，极少有（或无）N 出现。

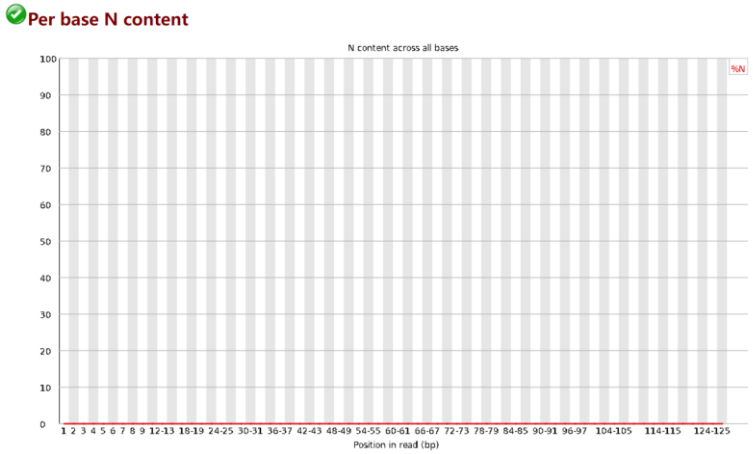

图 9.12: FastQC 报告 (6)-每条序列 N 含量。

7. 序列长度分布 (Sequence Length Distribution)

如图 9.13，显示序列长度的分布。该部分的警告或不通过不必要很担心，有些测序平台所有的 reads 都有相同的长度，有些平台并不，后者会被发给警告，但无需为此担忧。

当任何序列的长度为 0 时，不予通过。

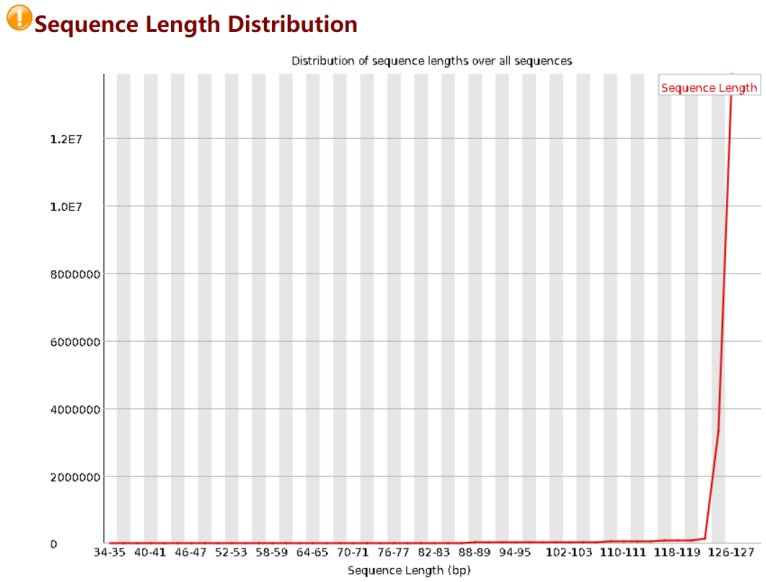

图 9.13: FastQC 报告 (7)-序列长度分布。

8. 序列重复水平 (Sequence Duplication Levels)

如图 9.14，该图显示具有不同重复水平的序列的相对数量。注意这是一个折线图，且越到右边，x 轴划定分组的范围就越大。

图中的蓝色线展示全序列集合中重复水平的分布，红色线展示去重后的序列集的重复水平的分布。重点关注蓝色线。

如果非唯一序列占总数的 20% 以上，此模块将发出警告；如果超过 50%，将不予通过。很遗憾，本例数据没有通过此项质控。读者可以自己寻找原因，推测可能是过度 PCR 或者过度测序。

9. 过度代表的序列 (Overrepresented sequence)

如图 9.15，本例中没有过度代表的序列。

正常的文库应该是多样的，如果发现单个序列在数据中的代表性过高，要么意味着它具有高度的生物学意义，要么表明文库被污染。

如果发现任何序列占总数的 0.1% 以上，此模块将发出警告；如果超过 1%，将不予通过。

10. 接头含量 (Adapter Content)

如图 9.15，该图显示每个位置看到每个适配器序列的文库比例的累积百分比计数。一旦在 reads 中看到一个序列，它就被视为一直存在到 reads 结束，因此百分比只会随着 reads 长度的增加而增加。

如果任何序列存在于超过 5% 的 reads 中，此模块将发出警告；如存在于超过 10 % 的 reads 中，此模块将不予通过。

本例中的数据中很少有接头序列。

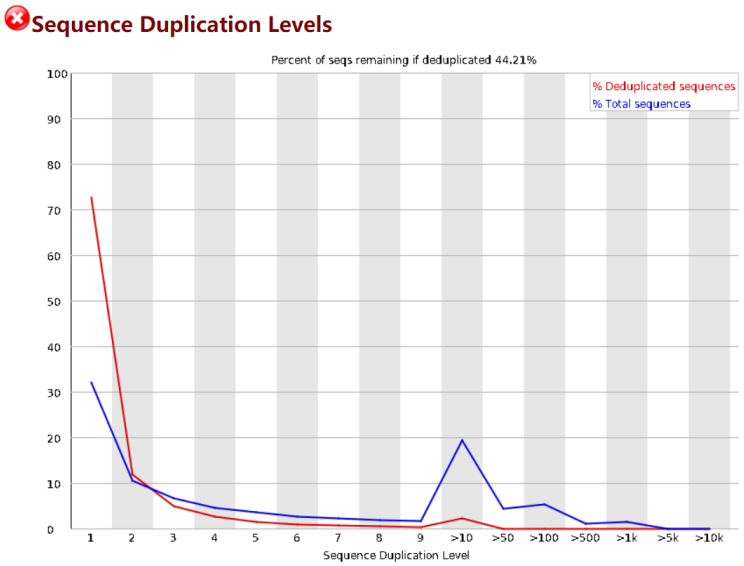

图 9.14: FastQC 报告 (8)-序列重复水平。

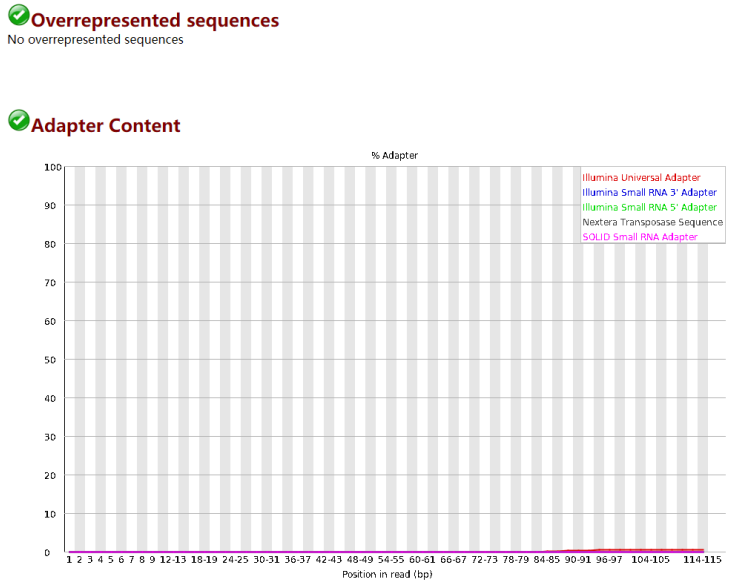

图 9.15: FastQC 报告 (9)(10)-过度代表的序列和接头内容。

至此，FastQC 报告内容已经完毕。读者或许有一种感觉，这份测序数据的质量未免**太好了**。然而，应该注意到，这是经过 `trimmomatic` 处理之后的质控结果! 因此，并不能说明原始测序数据的质量，读者可以对原始测序数据也进行一次质控，并作类似的解读。

9.4.3 比对和转换

如果质控通过，数据就可以进入比对分析；但如果没有通过，可以采用更严格的质量过滤处理一遍，重新质控，也可以重新测序，还可以忽略质控警告，直接进行下一步分析。

9.4.3.1 比对软件的介绍

第三章介绍了序列比对的基本方法，这里结合高通量测序的读段比对，做一点补充。据参考文献 [14] 统计，从 1988 年到发表前 (2021 年)，已经至少有107种比对工具问世。这里摘取一部分：

表 9.1: 几种常用比对工具

工具名称	出版年份	用途	索引方法	配对比对方法
FASTA	1988	DNA	Hashing	SW and NW
BLAST	1990	DNA	Hashing	Non-DP Heuristic
Gapped BLAST	1997	DNA	Hashing	SW
BLAT	2002	DNA	Hashing	Non-DP heuristic
BWT-SW	2008	DNA	BWT	SW
BWA	2009	DNA	BWT-FM	Semi-Global
Bowtie	2009	DNA	BWT-FM	HD
TopHat	2009	RNA-Seq	BWT-FM	HD
BWA-SW	2010	DNA	BWT-FM	SW
Bowtie2	2012	DNA	BWT-FM	SW & NW
BWA-MEM	2013	DNA	BWT-FM	SW & NW
STAR	2013	RNA-Seq	Suffix array	SW
TopHat2	2013	RNA-Seq	BWT-FM	SW & NW
HISAT	2015	RNA-Seq	BWT-FM	Non-DP Heuristic
BWA-MEM2	2019	DNA	BWT-FM	SW
HISAT2	2019	DNA	BWT-FM	Non-DP Heuristic

如要介绍所有软件的原理和实现，既不必要，也不可能，因此，这里推荐有兴趣的读者自行查找相关文献或软件文档。

本章我们使用的比对工具是 HISAT2，下一章 ChIP-seq 项目笔记中使用的比对工具是 Bowtie2。

HISAT2，发表于 2019 年（参考文献 [15]），是一个快速、灵敏的比对程序，可将 NGS reads（DNA 或 RNA）比对到人类基因组或其他参考基因组。基于图的 BWT 的扩展，HISAT2 首次实现了一个图 FM 索引（GFM）。除了使用一个代表人类基因组的全局 GFM 索引之外，HISAT2 还使用很多加起来可覆盖整个基因组的小的 GFM 索引。这些小的索引（称为局部索引）与多种比对策略一起，实现了测序 reads 的快速精确比对。新的索引策略被称为层次性图 FM 索引（Hierarchical Graph FM index，HGFM）。

Bowtie2，发表于 2012 年（参考文献 [16]），是一个将测序 reads 快速比对到长参考序列的程序，它的一大优点是省内存。Bowtie2 特别擅长于将大约 50-100nt 的 reads 比对到相对较长的基因组上。它使用 FM 索引来索引基因组以维持较少的内存开销，对于人类基因组，其一般所需内存为 3.2G。Bowtie2 支持有空位的、局部的以及端到端比对模式。[4]

9.4.3.2 构建索引

比对之前先要为基因组构建索引，使用 `hisat2-build` 程序。某些基因组在网络上已有构建完成的索引可供下载，但拟南芥基因组不属于此类，因此需要自己构建。不同于 Bowtie2 的索引只有基因组序列信息，HISAT2 的索引包含序列信息和转录组信息。转录组信息需要自己从 GTF 注释文件中提取，HISAT2 提供了两个程序 `hisat2_extract_splice_sites.py`、`hisat2_extract_exons.py` 可以完成这项工作：

```
hisat2_extract_splice_sites.py genome/Arabidopsis_thaliana.TAIR10.54.gtf > genome/splice_sites.txt
hisat2_extract_exons.py genome/Arabidopsis_thaliana.TAIR10.54.gtf > genome/exons.txt
```

构建索引：

```
hisat2-build -p 50 --ss genome/splice_sites.txt --exon genome/exons.txt genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa genome/genome > genome/hisat2-build.log 2>&1 &

# 选项
# -p 使用的线程数
# --ss  指定一个剪接位点的列表，注意，必须是hisat2自己的格式
# --exon  指定一个外显子列表，注意，必须是hisat2自己的格式
#  注意，--ss选项和--exon选项必须同时使用，或同时不用
#  genome/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa  基因组FASTA文件
# genome/genome 输出文件的位置和前缀
# genome/hisat2-build.log 构建索引日志
```

生成的文件如下：

[4] 这些话无用，只是鹦鹉学舌，弄清楚原理需要大量时间，讲清楚原理就更难了。

```
genome
    Arabidopsis_thaliana.TAIR10.54.gff3
    Arabidopsis_thaliana.TAIR10.54.gtf
    Arabidopsis_thaliana.TAIR10.dna.toplevel.fa
    exons.txt
    genome.1.ht2 # 索引文件都以ht2结束，共8个
    genome.2.ht2
    genome.3.ht2
    genome.4.ht2
    genome.5.ht2
    genome.6.ht2
    genome.7.ht2
    genome.8.ht2
    splice_sites.txt
    Truseq-PE.fa
```

索引构建完成以后，即可开始比对。

9.4.3.3 比对

执行下面的脚本:

```
#! /bin/bash
for i in {28..33}
do
    access="SRR71609${i}"
    hisat2 -p 50 --rna-strandness RF --known-splicesite-infile genome/
        splice_sites.txt      -x      genome/genome      --dta      -1
        workdir/${access}/${              access}_1P              -2
        workdir/${access}/${access}_2P > workdir/${access
        }/${access}.sam
done

# 选项
# -p线程数
# --rna-strandness 指定链特异性信息，默认是没有链特异性的，对于单端测序，选择F或R，F代表读段对应于转录本，R代表读段是转录本的反向互补序列；对于双端测序，指定FR或RF，使用这个选项让每个比对获得一个XS属性标签，+标签表示该读段属于一个在基因组+链上的转录本，-标签表示该读段属于一个在基因组-链上的转录本
# --known-splicesite-infile 指定已知剪接位点信息，在本例这一选项是不需
```

```
    要的，因为在构建索引的过程中已经考虑了剪接位点，而这里指定的剪接位点
    与之前指定的剪接位点相同
12  # -x 指定索引文件的位置和前缀
13  # --dta 即--downstream-transcriptome-assembly，如果下游需要进行转录本组
    装，就应该增加这个选项，使用此选项将报告为转录本组装程序而额外剪裁过
    的比对，hisat2需要更长的锚定长度才能从从头发现剪接位点，到这于短锚的
    比对减少，帮助提高转录组组装程序的计算和内存使用率
14  # -1指定read1文件
15  # -2指定read2文件
16  # workdir/${access}/${access}.sam  指定输出文件的位置和文件名
```

SAM 文件一般较大，在存储空间有限的情况下应该使用管道符 | 直接传递给 samtools 。HISAT2 将产生一些打印在屏幕上的输出结果，节录如下：

```
1  16389493 reads; of these: # 总reads数目
2     16389493 (100.00%) were paired; of these: # 可配对的reads，因为是用
         trimmomatic输出的_1P和_2P做的比对，当然是100%
3     337606 (2.06%) aligned concordantly 0 times # 没有比对上的reads
4     15292177 (93.30%) aligned concordantly exactly 1 time # 唯一比对的
         reads
5     759710 (4.64%) aligned concordantly >1 times # 不唯一比对的reads
6     ----
7     337606 pairs aligned concordantly 0 times; of these: # 没有合
         理比对的
8        137938 (40.86%) aligned discordantly 1 time # 只有不合理的比对的
9     ----
10    199668 pairs aligned 0 times concordantly or discordantly; of these
         :
11       399336 mates make up the pairs; of these:
12       199796 (50.03%) aligned 0 times
13       179862 (45.04%) aligned exactly 1 time # 作为单端比对，可以
            唯一比对
14       19678 (4.93%) aligned >1 times # 不唯一比对
15 99.39% overall alignment rate # 总体比对率=唯一比对+不唯一比对+
    不能比对中可以单端比对的
```

本次数据的比对结果普遍很好。

9.4.3.4 SAM 文件排序和转换

SAM 格式，全称 Sequence Alignment/Map，是一个用于存储比对到参考序列上的生物序列的文本格式。BAM 格式，全称 Binary Alignment/Map，是无损失压缩的二进制 SAM 文件。SAM，BAM，及配套分析工具 SAMtools 是中国科学家李恒主导开发的 (参考文献[19])。

SAM 文件由文件头 (header) 和比对区 (alignment section) 组成。文件头部分以 `@` 开头，截取 `SRR7160928.sam` 的文件头如下：

```
1 @HD    VN:1.0 SO:unsorted
2 @SQ    SN:1 LN:30427671
3 @SQ    SN:2 LN:19698289
4 @SQ    SN:3 LN:23459830
5 @SQ    SN:4 LN:18585056
6 @SQ    SN:5 LN:26975502
7 @SQ    SN:Mt LN:366924
8 @SQ    SN:Pt LN:154478
9 @PG    ID:hisat2     PN:hisat2     VN:2.2.1      CL:"/home/lishuai/
    Software/miniconda3/envs/jejune/bin/hisat2-align-s --wrapper basic
    -0 -p 50 --rna-strandness RF --known-splicesite-infile genome/
    splice_sites.gtf -x genome/genome --dta -1 workdir/SRR7160928/
    SRR7160928_1P -2 workdir/SRR7160928/SRR7160928_2P"
```

比对区由很多行构成，每行包括 11 个固定的字段，以及几个可选字段。这 11 个强制性字段包括：

列	字段	类型	描述
1	QNAME	String	read 名字
2	FLAG	Int	比对 FLAG 数字之和
3	RNAME	String	参考序列名，比对到参考序列的染色体号
4	POS	Int	read 比对到参考序列上，第一个碱基所在的位置
5	MAPQ	Int	比对的质量分数，越高说明比对越唯一
6	CIGAR	String	CIGAR 值
7	RNEXT	String	mate 或者下一个序列参考序列名（染色体号）
8	PNEXT	Int	mate 或者下一个序列的位置
9	TLEN	Int	模板长度观测值
10	SEQ	String	片段序列
11	QUAL	String	Phred33 方法表示的碱基质量

截取 `SRR7160928.sam` 比对区的一个条目如下：

```
SRR7160928.3  99     3      16908211     60     100M1D26M     =
    16908271     186
    AGCCATTCCCGATGCCGCGTTTGCCTGTTATCATATCAATACACATAACAAAACACTGAGAAGAT
    GCACATCTCTTCAAAAACACATTTTCTAAAACAAAATCACAACTTAACACAGAAATTGGAT
    BBBBBFFFFFFFFFFFFFFFFFFFFFFF<FFFFFFFF<FFFFF/FFF<BFFFFFFFFF<FF<
    FFFFFFBBFF//<FFBB/<<BBFFBFF/<FFFFFF<FBFFFFFFFFFFFBFFBFFFFFF<//< AS:
    i:-8 XN:i:0 XM:i:0 XO:i:1 XG:i:1 NM:i:1 MD:Z:100^G26 YS:i:-8 YT:Z:
    CP XS:A:- NH:i:1

# 固定字段
# SRR7160928.3: read名字，在一个SAM文件中可能出现多个同read名的条目，这是因为一条read可能被比对到多个位置
#  99:  以bit形式表示的FLAGs，这里99=64+32+2+1分别代表"这条序列是read1"、"配对序列反向比对"、"配对序列正负链完美比对"、"read是成对序列中的一条"
# 3: 该read被比对到3号染色体
# 16908211: 该read比对上的起始位点是16908211
# 60: 该read的比对质量，计算方法为-10*log10(Pr{mapping position is wrong})，对最近的整数取整
# 100M1D26M:  CIGAR字符串，100个匹配、1个Deletion、26个匹配
# =: mate比对到的参考序列名，也是3，标记为=
# 16908271: mate比对上的起始位点是16908271
# 186: 成对reads比对起点到比对终点的距离，或者说是比对的长度或模板序列的长度
# AGCCATTCCCGATGCCGCGTTTGCCTGTTATCATATCAATACACATAACAAAACACTGAGAAGAT
    GCACATCTCTTCAAAAACACATTTTCTAAAACAAAATCACAACTTAACACAGAAATTGGAT
    read的序列
# BBBBBFFFFFFFFFFFFFFFFFFFFFFF<FFFFFFFF<FFFFF/FFF<BFFFFFFFFF<FF<
    FFFFFFBBFF//<FFBB/<<BBFFBFF/<FFFFFF<FBFFFFFFFFFFFBFFBFFFFFF<//<:
    Phred33编码表示的碱基质量

# 可选字段
# 可选字段遵循统一的格式：TAG:TYPE:VALUE
# AS:i:-8 比对分数，带符号整数型，得分为-8
# XN:i:0 在参考序列上模糊碱基的个数
# XM:i:0 错配的个数
# XO:i:1 gap open的个数
# XG:i:1 gap 延伸的个数
#  NM:i:1  到参考序列的编辑距离，包括模糊的碱基，但不包括剪切，带符号整数
```

```
    型，编辑距离为1，因为仅有1个deletion；关于编辑距离可参考第3章
24  # MD:Z:100^G26 错配位置字符串，可能是所有的可打印字符，100个碱基后出
    现一个gap
25  # YS:i:-8 成对序列比对得分
26  # YT:Z:CP 比对的类型，CP 表示合理的比对
27  # XS:A:- 第二好的匹配的得分，这里没有
28  # NH:i:1  在本记录中已经报告的包含此查询序列的比对数目
```

SAM 格式是很复杂的，如果想要尝试挖掘更多的比对信息，SAM 文件是很好的入手之处。但本章的介绍就到此为止，聊为一点引入。SAM 文件以可直接读取的方式保存比对结果，方便人类阅读，但不利于计算机处理，因此需要转换成二进制的 BAM 文件，执行如下脚本转换：

```
1 #! /bin/bash
2 for i in {28..33}
3 do
4     access="SRR71609${i}"
5     samtools sort -l 5 -O BAM -T workdir/${access}/tmp -o
          workdir/${ access}/${access}.bam
          workdir/${access}/${access}.sam -@ 50 -m 1G >
          workdir/${access}/${access}.sort.log 2>&1
6 done
7 # 选项
8 # -l 指定压缩水平，0为不压缩，1为最低压缩，9为最高压缩，较高的压缩可以缩
    减文件大小但会拖慢处理速度
10 # -O 指定输出文件格式
11 # -T 指定临时文件存放位置和前缀
12 # -o 指定输出文件
13 # -@ 线程数
14 # -m 为每个线程分配的内存大小
```

注意，转换后的 BAM 文件是不能用 `less`、`cat` 等命令直接读取的。

9.4.4 转录本组装

获得 BAM 文件后，即可进行转录本的组装。不过，为什么要组装 (assemble)?

因为 NGS 获得的小片段不能直接用于后续的生物学分析过程，需要将这些小片段拼接称较长的片段。本次分析使用 StringTie 进行。

StringTie 发表于 2015 年（参考文献 [21]）是一款专为 RNA-seq 设计的转录本组装与定量软件，与后续的分析软件如 DESeq2、EdgeR 等有良好的兼容性。关于它的详细介绍，请阅读参考文献 [7]。

执行下面的脚本：

```
#! /bin/bash
for i in {28..33}
do
    access="SRR71609${i}"
    stringtie workdir/${access}/${access}.bam --rf -p 50 -o
       workdir/${ access}/transcript.gtf -G
       genome/Arabidopsis_thaliana.TAIR10
       .54.gtf -e -b workdir/${access} > workdir/${access}/stringtie.
       log 2>&1
done

# 选项
# -rf 指定链特异性建库方式：fr-firststrand
#  -p 组装转录本的线程数
#  -o 设置StringTie组装转录本的输出GTF文件的路径和文件名
#  -G 使用参考注释基因文件指导组装过程，格式GTF/GFF3
# -e 限制reads比对的处理，仅估计和输出与用-G选项给出的参考转录本匹配的组
   装转录本，使用该选项会跳过处理与参考转录本不匹配的组装转录本，将大大
   提升处理速度
# -b 指定.ctab 文件的输出路径
#  workdir/${access}/${access}.bam 指定输入的BAM文件，注意该输入文件必须
   按其基因组位置排序，HISAT2的输出文件则需经过samtools sort生成的bam文
   件才可当做输入文件
```

StringTie 的主要输出文件包括（以 SRR7160928 为例）：

```
SRR7160928
    e2t.ctab # 所有.ctab文件都是用于下游Ballgown软件差异表达分析的输
    入文件
    e_data.ctab
    i2t.ctab
    i_data.ctab
    stringtie.log # 组装日志
    t_data.ctab
    transcript.gtf # 包含已组装转录本的GTF文件
```

9.4.5 生成计数矩阵

在 Linux 终端使用各种软件进行的操作已经接近尾声，接下来我们需要从 StringTie 产生的 GTF 文件中生成计数矩阵，以供基因差异表达分析使用。

StringTie 软件包提供了一个 Python 脚本 `prepDE.py`，可从 `stringtie` 输出的 GTF 文件中方便地提取 reads 计数映射到特定基因组区域的矩阵，供 `DESeq2` 使用。

执行下面的命令：

```
prepDE.py -i workdir -l 126

# 选项
# -i 指定输入文件，需要给一个包含所有样本子目录的目录，在这里我们直接给 workdir/目录
# -l 平均reads长度，可从fastqc结果中获知
```

该命令将产生两个文件：`gene_count_matrix.csv` 和 `transcript_count_matrix.csv` 。使用 FileZilla 或 Xftp 将它们下载到本地，RNA-seq 分析第一阶段至此结束，接下来的分析可在个人电脑中完成。

9.5 RNA-seq 分析 II

Linux-based Essential Bioinformatics(LEB) 是这门课程的缩写，既然如此，为什么不把这一部分分析也在 Linux 平台上完成呢？

因为在 Linux 上安装和编译 R 包太容易出错，且 R 软件对于 Windows 平台还算友好。

9.5.1 R 与 Bioconductor 简介

R 是科学计算中重要的开源软件，软件包中涉及多种算法和应用，而生物相关的软件都收集在Bioconductor平台下。CRAN(综合性 R 存档网络，Comprehensive R Archive Network) 是由 R 基金支持的 R 语言的中心软件仓库。要在计算机上安装 R，可访问 CRAN 下载 R 软件安装包。

Bioconductor 项目旨在开发、支持和传播免费的开源软件，促进对生物数据的严格的和可重复的分析。

为编写 R 程序，我们需要一个好用的集成开发环境 (IDE)，这里推荐RStudio，可安装 RStudio Desktop Open Source Edition。软件的安装和配置可参考互联网教程。顺便一提，安装完毕后记得到 RStudio>Tools>Global Options>Packages>Management 中，将 Primary CRAN repository 改为 China (Beijing) [https] - TUNA Team, Tsinghua University，这将显著提升下载包的速度。

一切准备就绪后，即可进行分析。

9.5.2 用 DESeq2 进行差异表达分析

差异基因表达 (Differential Gene Expression) 分析是 RNA-seq 分析的重要一环，顾名思义，该分析能找出在对照组和实验组中具有差异表达的基因。这些基因表达水平在实验条件下的上调或下调，如果足够显著的话，是很可能有生物学意义的。

DESeq2 是分析差异表达的常用 R 包，用于高维计数数据的归一化、可视化和差异分析。它利用经验贝叶斯技术来估计对数倍数变化和离散程度的先验估计值，并计算这些量的后验估计值。

安装和载入 `DESeq2`：

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager") # 检查BiocManager是否安装，
        如无，则安装

BiocManager::install("DESeq2")  # 从bioconductor安装DESeq2
library("DESeq2") #载入DESeq2包
```

在R 控制台输入`vignette("DESeq2")` 可以查看DESeq2 的使用说明 *Analyzing RNA-seq data with DESeq2*，十分值得一读。`DESeq2` 的基本使用方法如下（引自 DESeq2 使用说明），先留下一个印象，之后会结合本次分析的例子详细解释：

```
dds <- DESeqDataSetFromMatrix(countData = cts,
                              colData = coldata,
                              design= ~ batch + condition)
dds <- DESeq(dds)
resultsNames(dds)
res <- results(dds, name="condition_trt_vs_untrt")
res <- lfcShrink(dds, coef="condition_trt_vs_untrt", type="apeglm")
```

DESeq2 预期输入的计数数据是一个整数构成的矩阵。该矩阵的第 i 行第 j 列记录了在第 j 个样品数据中有多少条 reads 可以被分配到第 i 个基因头上。DESeq2 将在内部校正文库大小，因此输入的数据无需也不应该事先归一化。

我们不妨检查一下 `gene_count_matrix.csv` 文件的内容，可使用 RStudio 的内置工具打开（不要使用 Excel）。截取前几行：

```
gene_id,SRR7160928,SRR7160929,SRR7160930,SRR7160931,SRR7160932,
    SRR7160933
AT1G01010|NAC001,643,445,523,285,335,293
AT1G03987|AT1G03987,0,0,0,0,0,0
AT1G01030|NGA3,115,122,123,106,100,116
AT1G01020|ARV1,451,421,440,390,454,477
AT1G01060|LHY,673,643,654,928,1044,853
```

```
AT1G01070|UMAMIT28,614,546,529,207,204,179
AT1G01080|AT1G01080,1002,819,843,1878,2073,2119
AT1G03997|AT1G03997,0,0,0,0,0,0
```

第 1 列是基因编号，第 2-7 列分别对应 SRR7160928 到 SRR7160933 共 6 个样品，数据都是整数，应该符合 DESeq2 的要求。

DESeq2 中重要的数据结构是DESeqDataSet(在代码中常以`dds` 表示该数据结构的实例)，构建 `dds` 需要一个计数矩阵 `cts` ，一个关于样品信息的表格 `coldata` 和设计公式。

DESeqDataSet 对象必须具有关联的设计公式（design formula）。设计公式表达了建模中用到的变量，应该以波浪线 (~) 开头，后面是以加号 + 连接的多个变量。设计公式用来估计离散程度和估计模型的log2 倍数的变化。

首先构建`dds`对象：

```
# 在之前同一环境中执行
# gene_count_matrix.csv 需放在当前工作目录下
countData <- as.matrix(read.csv("gene_count_matrix.csv", row.names="gene_id")) # 载入csv文件，指定第一列为列名而非数据，countData内容如图9.16
condition <- factor(c(rep("Osmotic",3),rep("Control",3))) # 将字符串向量转换成因子，colData数据框内容如下一个代码框中所示
colData <- data.frame(row.names=colnames(countData), condition) # 以countData的列名为行名，以因子为第二列内容
dds <- DESeqDataSetFromMatrix(countData = countData, # 计数矩阵
                              colData = colData, # 样品信息，已经实现样品名(SRR71609..)和处理条件因子("Osmotic","Control")的对应
                              design = ~ condition) # 仅仅研究实验条件对表达的影响
```

```
> colData # colData内容
           condition
SRR7160928 Osmotic
SRR7160929 Osmotic
SRR7160930 Osmotic
SRR7160931 Control
SRR7160932 Control
SRR7160933 Control
```

生成`dds`对象后，即可进行差异表达分析。差异表达分析的标准步骤都被封装在了函数`DESeq`中。可以在R 控制台输入`?DESeq`查看帮助。执行下面的命令进行分析：

	SRR7160928	SRR7160929	SRR7160930	SRR7160931	SRR7160932	SRR7160933
AT1G01010|NAC001	643	445	523	285	335	293
AT1G03987|AT1G03987	0	0	0	0	0	0
AT1G01030|NGA3	115	122	123	106	100	116
AT1G01020|ARV1	451	421	440	390	454	477
AT1G01060|LHY	673	643	654	928	1044	853
AT1G01070|UMAMIT28	614	546	529	207	204	179
AT1G01080|AT1G01080	1002	819	843	1878	2073	2119
AT1G03997|AT1G03997	0	0	0	0	0	0
AT1G01040|DCL1	1683	1615	1603	1662	1882	1744
AT1G03993|AT1G03993	1	1	1	1	1	0
AT1G01046|ath-MIR838	42	34	43	43	52	51
AT1G01050|PPa1	2293	2459	2362	2292	2494	2371
AT1G01090|PDH-E1 ALPHA	3884	3496	3702	4019	4499	4082
AT1G01110|IQD18	80	71	75	130	108	95
AT1G01120|KCS1	2217	1712	1913	2583	2829	2544
AT1G01100|AT1G01100	4722	3955	4381	5020	5159	4861
AT1G01180|AT1G01180	416	409	424	451	469	458
AT1G01200|RABA3	35	28	41	76	65	59
AT1G01190|CYP78A8	156	172	246	155	204	180
AT1G01130|AT1G01130	48	47	37	33	29	51
AT1G01140|CIPK9	2925	2605	2481	1791	1998	1615

图 9.16: countData 矩阵的内容（截取）。

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

而后用 `results` 对处理条件对比构建结果表：

```
res <- results(dds, contrast = c("condition","Osmotic","Control"))
```

不妨看看 `res` 对象（在 R 控制台输入 `res` 后回车即可），这是一个 32833 行 6 列的表格。其中的信息包括 $\log_2$ 倍数改变（log 2 Fold Change, LFC），p 值和校正后的 p 值。在 `results`函数没有特殊参数，$\log_2$ 倍数改变和 Wald 测验 p 值是针对设计公式的最后一项而言的；如果该项是一个因子，那么就会在作该因子的最后一个水平与参考水平之间的比较。

本例指定参数 `contrast = c("condition","Osmotic","Control")` ，即指定 $\log_2$ 倍数改变 $= \log_2 \frac{Osmotic}{Control}$。Wald test 的 p 值也是针对 Osmotic vs Control 而言的。

```
> res
log2 fold change (MLE): condition Osmotic vs Control
Wald test p-value: condition Osmotic vs Control
DataFrame with 32833 rows and 6 columns
                     baseMean log2FoldChange lfcSE      stat
                       pvalue     padj
                    <numeric>   <numeric> <numeric> <numeric> <numeric>
                       <numeric>
AT1G01010|NAC001  414.226    0.6991045 0.1378251 5.072404 3.92821e
```

```
      -07 1.19671e-06
8  AT1G03987|AT1G03987   0.000           NA       NA        NA         NA
            NA
9  AT1G01030|NGA3     113.442    0.0418131 0.2049195 0.204047 8.38317e
      -01 8.79056e-01
10 AT1G01020|ARV1     438.232   -0.1280266 0.1179763 -1.085189 2.77838e
      -01 3.57643e-01
11 AT1G01060|LHY      802.978   -0.6359010 0.0917137 -6.933542 4.10432e
      -12 1.76098e-11
12 ...                    ...          ...      ...       ...        ...
           ...
13 ATCG00010          0.000000          NA       NA        NA         NA
            NA
14 ATCG00060          0.501972     0.796135  3.05109 0.2609347  0.794143
           NA
15 ATCG00230          0.337939    -0.118069  3.91989 -0.0301205  0.975971
           NA
16 ATCG00260          0.177490    -1.079850  4.08047 -0.2646386  0.791288
           NA
17 ATCG01030          0.337939    -0.118069  3.91989 -0.0301205  0.975971
           NA
```

我们可以按校正后 p 值从小到大对 res 表格进行排序，而后写入文件中。

```
1  resOrdered <- res[order(res$padj), ] # 对res根据padj进行排序
2  write.table(x = as.data.frame(resOrdered), file = "./
      Osmotic_Control_all", quote = F, sep = "\t", row.names = TRUE, col.
      names = TRUE) #将排序后的结果写入文件
3
4  # 选项
5  # x = as.data.frame(resOrdered) 要写入文件的对象，这里进行强制类
      型转换成数据框
6  # file = paste0(".","Osmotic","_","Control","_all")
7  # quote = F 字符或因子不会被双引号包围
8  # sep = "\t" 指定字段分隔符，x中每一行都会被这个符号分开，这里选择的
      是制表符
9  # row.names = TRUE 存在行名列
10 # col.names = TRUE 存在列名行
11 # file = "./Osmotic_Control_all" 指定导出的文件路径及其路径
```

此后，对 resOrdered 中的条目作一些筛选，选择其中变化较为显著的。这里的关键问题是：什么样的变化“显著”？本例中采取的标准是要求：校正后 p 值非空（NA），且，校正后 $p < 0.05$，且，$\log_2$ 倍数变化的绝对值 >= 1（这一位置至少发生了 2 倍以上的上调或下调）。代码如下：

```
resSig <- resOrdered[! is.na(resOrdered$padj) & resOrdered$padj < 0.05
    & abs(resOrdered$log2FoldChange)>=1, ] # 筛选显著变化的基因存入
    resSig中
write.table(x = as.data.frame(resSig), file = "./Osmotic_Control",
    quote = F, sep = "\t", row.names = TRUE,col.names = TRUE) #
    写入文件中
```

此时，当前工作目录下将得到两个文件 Osmotic_Control_all 和 Osmotic_Control 。后者是在渗透处理后表达水平显著变化的基因，我们可以用此表格进行富集分析。

查看 Osmotic_Control 文件的前几行：

```
    baseMean  log2FoldChange lfcSE stat   pvalue padj
AT1G16850|AT1G16850  1715.99374481034   5.49670350388015
    0.139677664645622    39.352773529153 0    0
AT1G17020|SRG1 3114.9192729861 4.06382582468953  0.0788026125440306
    51.5696839672528     0      0
AT1G54570|PES1 6540.59765028929   3.65917578065995
    0.0904564945713295   40.4523279174223     0      0
AT1G54575|AT1G54575  3949.00276631203   4.4835805051715
    0.0889960098312112 50.3795677320254     0      0
AT1G58360|AAP1 7140.71482107642   2.6659669727083 0.0629809894512548
    42.3297092652326     0      0
```

该表格的内容很丰富，但我们只需要其中的基因名，不需要 baseMean、log2FoldChange、lfcSE、stat、pvalue 或 padj 等信息，因此需要稍做处理，在 Linux 中会比较方便：

```
sed -n '2,$p' ./Osmotic_Control | cut -f 1 > ./DE_genes

# 解释
# sed -n '2,$p' ./Osmotic_Control 打印该文件第2到最后一行的内容
# | 管道符，sed命令处理完的行传输给cut命令
# cut -f 1 提取该行的第一个字段
# > ./DE_genes文件 处理完毕的结果重定向到./DE_genes文件
```

查看 DE_genes 文件前几行：

```
AT1G16850|AT1G16850
```

```
AT1G17020|SRG1
AT1G54570|PES1
AT1G54575|AT1G54575
AT1G58360|AAP1
AT1G64110|DAA1
AT1G75040|PR5
AT2G19900|NADP-ME1
AT2G22990|SNG1
AT2G37870|AT2G37870
AT2G38530|LTP2
```

提取基因名成功。

9.5.3 GO 富集分析

在第七章中我们曾简单介绍 InterPro 数据库中的 GO 条目，可以回顾。

GO 是基因本体（Gene Ontology）的缩写，它的数据库是世界上最大的基因功能信息知识库，是大规模分子生物学和遗传学实验的计算分析的基础。其知识是人类可读的，也是机器可读的。GO 协作团队的目标是建立一个全面的生物系统计算模型，纵向囊括从分子到有机体水平，横向跨越生命之树的多样。

基因本体是一种系统地对物种基因及其产物属性进行注释的方法和过程。它的目标是：

- 维护和发展有限的基因及其产物属性描述的词汇
- 注释基因及其产物，同化和传播注释数据
- 提供方便的工具访问数据
- 实现在实验数据的基础上，使用 GO 进行解析，例如基因富集组分分析

GO 主要包括三个分支：细胞组分、分子功能和生物过程。

不应该被"Ontology" 这个词所迷惑，它既不是哲学意义上的本体论，也不是信息学意义上的本体。在这里，Ontology 指代具有层次的术语表，提供一种规范的注释方式。[5]

本次分析使用 `clusterProfiler` 进行 GO 富集。首先进行准备工作。使用 clusterProfiler 这个包进行 GO 分析时，通常并不需要转换 ID，所有的 ID 类型都可以由 `OrgDb` 文件提供。

Bioconductor 维护了常见物种的 `OrgDb` 文件，可以通过安装包使用。

```
# 安装BiocManager，如果本没有安装的话
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

# 安装包
```

[5]读者可以看一看：*The Ontology of the Gene Ontology*，这是一篇有点意思的文章，可以消除 GO 的神秘感，或者说，对它进行“祛魅”。

```
BiocManager::install("AnnotationHub")
BiocManager::install("org.At.tair.db")
BiocManager::install("clusterProfiler")
BiocManager::install("dplyr") # 如果安装不顺利，可尝试添加force=TRUE参数
BiocManager::install("ggplot2")
# 载入包
library("AnnotationHub")
library("org.At.tair.db")
library("clusterProfiler")
library("dplyr")
library("ggplot2")
```

读入表达水平显著改变的基因名，并用 enrichGO 进行富集分析，随后作图。富集结果如图所示 9.17，可见差异表达基因被富集在呼吸电子传递链、电子传递链、细胞呼吸、氧化有机化合物供能、ATP 合成偶联的电子传递、氧化磷酸化、无氧呼吸、ATP 代谢过程、前体代谢物和能量生成方面。

对此可以进行一些解释：渗透胁迫可能导致细胞损伤，因此可能激活非正常的呼吸链或无氧呼吸，导致线粒体膜破坏进而引发电子传递链断裂，相关基因表达上调；渗透胁迫可能导致细胞内盐离子浓度偏高，导致需要额外的能量将多余的盐离子排出细胞，引发氧化有机化合物功能、氧化磷酸化、ATP 代谢的变化；渗透胁迫还可能促使植物在细胞内积聚代谢中间体以抵抗渗透压力，这导致了前体代谢物的积累。

总而言之，正如我的一位植物专业室友[6]所说的，“解释都可以解释，但真正研究还要仔细甄别”。

```
gene_all_tair <- read.table(file="./DE_genes") # 读入文件
gene_all_tair <- as.list(gene_all_tair)$V1 # 取其第一列作为列表

go_all <- enrichGO ( gene_all_tair, OrgDb=org.At.tair.db , ont = "BP" , pvalueCutoff = 0.05 , keyType="TAIR")

# enrichGO选项
# gene_all_tair 输入的基因列表
# OrgDb=org.At.tair.db 指定转换基因名所用的数据库
# ont = "BP" 在所有三个水平进行富集
# pvalueCutoff = 0.05 指定pvalue截断值
# keyType="TAIR" 标明输入的数据

```

[6]Zhang TY

```
13 #pdf("./DE_genes_GO.pdf",,width=10,height=10,compress=F)#如要
     生成pdf文件，取消本行注释
14 barplot(go_all,showCategory=20) + scale_fill_gradient( low = "#ffc080"
     ,high="#55a0fb",space="Lab",aesthetics="fill",guide=
     guide_colorbar(reverse= TRUE)) #  如要生成pdf文件，取消本行注释
15 #dev.off()
```

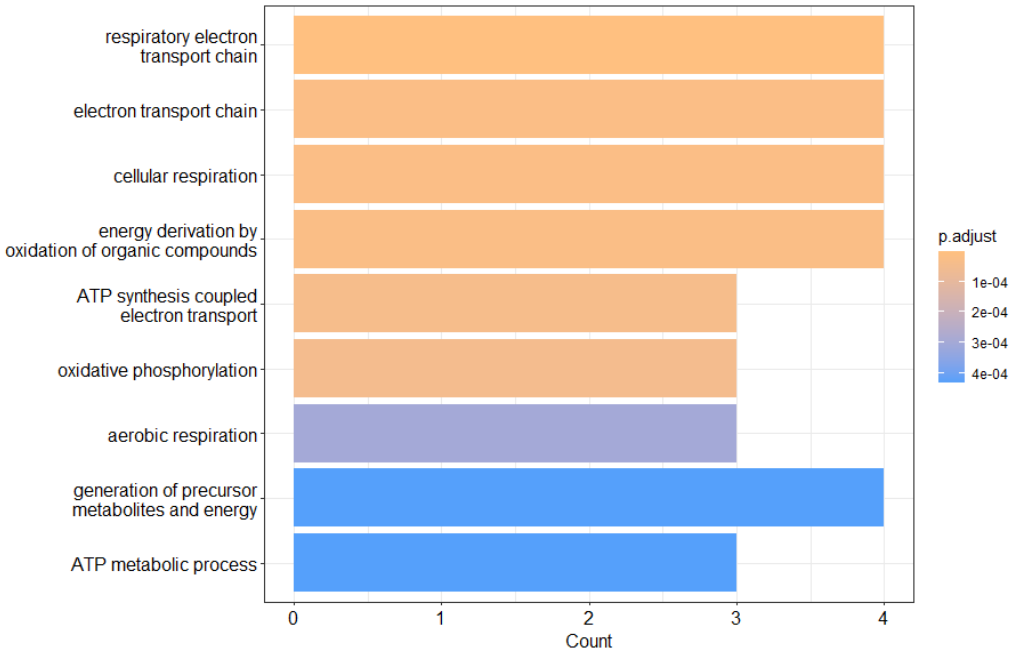

图 9.17: GO 富集分析气泡图-生物过程层次。这张图做了一些美化。

至此，全部 RNA-seq 分析已经结束。这只是一个简单演示，其结果是否正确，我没有什么把握，但本章的代码均已运行成功，或许对于读者有所帮助。

如果读者想要绘制热图、火山图，或者进行 KEGG 通路富集，可以阅读参考文献 [18] 的系列文章。参考文献 [18] 是本班同学的期末项目成果，相当详尽，是本章的内容所难以达到的，我自请避贤路。

参考文献

[1] NGS 数据过滤之Trimmomatic 详细说明

[2] TrimmomaticManual: V0.32

[3] Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics (Oxford, England)*, 30(15), 2114 – 2120.

[4] SAMtools Documentation

[5] FastQC Documentation

[6] NGS 文库制备-Illumina 技术页

[7] StringTie Manual

[8] Sultan, M., Schulz, M. H., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., ... & Yaspo, M. L. (2008). A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, 321(5891), 956-960.

[9] Kukurba, K. R., & Montgomery, S. B. (2015). RNA sequencing and analysis. *Cold Spring Harbor Protocols*, 2015(11), pdb-top084970.

[10] mRNA 文库构建

[11] Illumina TruSeq RNA Library Prep Kit v2 Data Sheet

[12] TruSeq Sample Preparation Best Practices and Troubleshooting Guide

[13] How short inserts affect sequencing performance

[14] Alser, M., Rotman, J., Deshpande, D., Taraszka, K., Shi, H., Baykal, P. I., Yang, H. T., Xue, V., Knyazev, S., Singer, B. D., Balliu, B., Koslicki, D., Skums, P., Zelikovsky, A., Alkan, C., Mutlu, O., & Mangul, S. (2021). Technology dictates algorithms: recent developments in read alignment. *Genome biology*, 22(1), 249.

[15] HISAT2 主页

[16] Kim, D., Paggi, J. M., Park, C., Bennett, C., & Salzberg, S. L. (2019). Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature biotechnology*, 37(8), 907 – 915.

[17] Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4), 357 – 359.

[18] 李扬. 等. RNA-seq 数据分析原理及流程详解.

[19] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., & 1000 Genome Project Data Processing Subgroup (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*, 25(16), 2078 – 2079.

[20] SAM 格式文档

[21] Pertea, M., Pertea, G. M., Antonescu, C. M., Chang, T. C., Mendell, J. T., & Salzberg, S. L. (2015). StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature biotechnology*, 33(3), 290 – 295.

[22] Bioconductor Project

[23] DESeq2 Manual

[24] 罗静初. Linux 生物信息技术基础. 课堂练习.