RNA-seq pipeline：fastqc + trim +hisat2

作者：卢建璋 李凯旋 徐舟通

1. 登陆并设置环境

Bash

```bash
#登陆
ssh jianl@10.105.100.153
#密码
111111
#设置成base环境
source ~/.bashrc
#切换环境
conda activate [environment] #常用环境为 tests twobittofa
```

2. Fastqc 质量检测

Bash

```bash
fastqc -o [output file pathway] -t 8 [input file pathway]
-t : 线程数量
```

结果输出：一个.html文件和一个压缩包。html文件可以下载到本地后使用默认浏览器打开，压缩包中含有质控检验结果的数字信息。

具体每个结果的意义详解：https://www.jianshu.com/p/134c45339805

3. 根据报告删除低质量reads（QC<x）

Bash

```
perl ./trim_and_filter_SE.pl （perl脚本的路径） -i [原fastq文件] -a 1 -b 100 -m 20 -q sanger -o [输出文件的位置以及文件头]

•SE为单端测序分析，PE的脚本为双端。
•-i为要检验的fastq文件
•-b是每一个read的长度
•-m是要求的最低QC
•-q是测序公司
•-o是输出文件的位置附加上文件头（title）
```

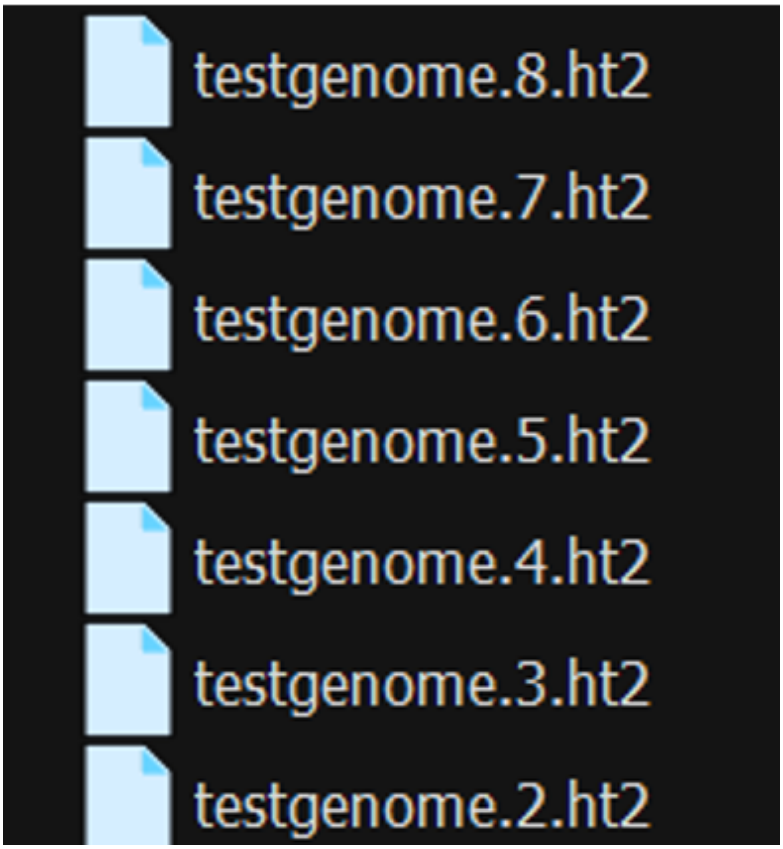
结果输出：title.trim_a_b.minQS_m.fastq

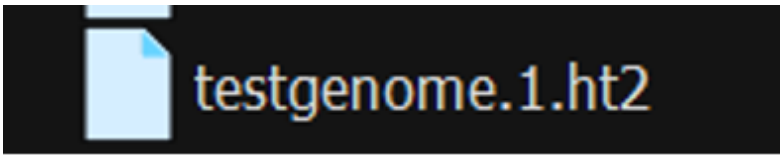
4.1 建立索引（hisat2）

Bash

```
hisat2-build -p 8 [参考序列位置(*.fa)] [输出文件前缀]
```

Hg19的索引已经建立完成，以后可以直接使用，这里的输出文件前缀为testgenome，可以根据自己的需要自定义文件开头名。





4.2 将reads比对到参考基因组上（hisat2）

Bash

```
hisat2 \
-q \
 -x [索引文件目录以及其前缀] \
 -U [输入文件位置 (*.fastq) ] \
 -S [输出文件位置及名字(*.sam) ] \
 -p 32
```

-q: 输入的文件为fastq或者其压缩形式；

-f：代表输入的文件为fasta或者其压缩形式。

-x: 后面加索引文件目录以及其前缀。

-U: 指的单端测序数据，若为双端测序，则此参数为-1和-2。

-S: 指定输出的sam文件名。

结果分析：

a. 单端：

- 结果显示（重要信息，务必记录或截图保存！！！）

```
95054259 reads; of these:
  95054259 (100.00%) were unpaired; of these:
    9943537 (10.46%) aligned 0 times
    75953573 (79.91%) aligned exactly 1 time
    9157149 (9.63%) aligned >1 times
89.54% overall alignment rate
```

- 单端需要查看确认的结果：

a. 整体比对率 (Overall alignment rate)

b. 恰好比对上一次的个数 (exactly 1 time)。

c. 一般不看Aligned > 1，因为Aligned大于一次意味着这段可以比对到基因组的多个地方。

b. 双端：

- 结果显示（重要信息，务必记录或截图保存！！！）

```
95054259 reads; of these:
  95054259 (100.00%) were paired; of these:
    13846051 (14.57%) aligned concordantly 0 times
    75149946 (79.06%) aligned concordantly exactly 1 time
    6058262 (6.37%) aligned concordantly >1 times
    ----
    13846051 pairs aligned concordantly 0 times; of these:
      426068 (3.08%) aligned discordantly 1 time
    ----
    13419983 pairs aligned 0 times concordantly or discordantly; of these:
      26839966 mates make up the pairs; of these:
        16561219 (61.70%) aligned 0 times
        8853787 (32.99%) aligned exactly 1 time
        1424960 (5.31%) aligned >1 times
91.29% overall alignment rate
```

- 双端需要查看的结果

a. Aligned concordantly表示read1，read2**同时合理匹配**的次数 (重点看正好为1的次数)

b. 在Aligned concordantly=0当中：

1. Aligned discordantly=1表示read1和read2**能都比对上但是不合理**（read之间长度过大/方向错误）

2. 剩余的即为两个reads**只有一个可以比对或者都不能比对**，此时将两个reads看成2个**单端测序**。其中aligned 0 time的就是**完全无法比对**的序列。

5. sam文件转化成bam文件

Bash

```
samtools view -bS *.sam > *.bam
```

6. 将bam/sam文件排序(SortSam.jar)

Bash

```
java -Xms2g -jar /public/workspace/jianl/Software/picard-tools-1.96/SortSam.jar INPUT=[输入的*.bam文件路径] OUTPUT=[输出的排序好的*.bam文件路径] SORT_ORDER=coordinate TMP_DIR=./tmp
```

由于sort的中间过程产生大量临时文件，容易占用服务器根目录下的TEMP文件夹空间，所以需要在当前工作路径下手动建一个临时文件夹tmp

7. 将排序好的bam/sam文件去重(MarkDuplicates.jar)

Bash

```
java -Xms2g -jar /public/workspace/jianl/Software/picard-tools-1.96/MarkDuplicates.jar INPUT= [排序好的bam文件] OUTPUT= [输出的bam文件] REMOVE_DUPLICATES=TRUE METRICS_FILE=DeDUPLICATE.txt
```

此步骤会得到一个去重后的bam文件外加一个显示重复信息的txt文件

结果分析

找到显示重复信息的txt文件

```
## METRICS CLASS  net.sf.picard.sam.DuplicationMetrics
LIBRARY UNPAIRED_READS_EXAMINED READ_PAIRS_EXAMINED UNMAPPED_READS  UNPAIRED_READ_DUPLICATES  READ_PAIR_DUPLICATES  READ_PAIR_OPTICAL_DUPLICATES  PERCENT_DUPLICATION ESTIMATED_LIBRARY_SIZE
Unknown Library 9698249 81924525  16561219  7688173 28411740  14910 0.371724  88868968
```

(UNPAIRED_READS_EXAMINED * 1) + (READ_PAIRS_EXAMINED * 2) + (UNMAPPED_READS * 1) = reads * 2

- UNMAPPED_READS就是Aligned 0 times的read个数
- UNPAIRED_READS_EXAMINED=**the number of read which is paired to an unmapped mate**
- READ_PAIRS_EXAMINED=**the number of mapped read pairs**

8. 转化文件类型和可视化 (bam到bw)

a. 获得bam文件的索引文件：samtools index [*.bam文件]

b. 可以得到一个 *.bam.bai文件

Bash

```
bamCoverage --bam [*.bam文件的位置]-o [*.bw文件的位置和名字] -p 32 --normalizeUsing RPKM

-p 线程
--normalizeUsing 表示使用的标准化方法
```