转录组流程

step1: sra2fastq

下载SRA数据

新建一个名为`SRR_Acc_List.txt`的文档，将SRR号码保存在文档内，一个号码占据一行。文件可以在我的GitHub下载获取：https://raw.githubusercontent.com/jmzeng1314/GEO/master/airway_RNAseq/SRR_Acc_List.txt 当然了，你自己去GEO里面找到SRA再找到文件才是正路。

- prefetch下载数据

```
wkd=/home/jmzeng/project/airway/ #设置工作目录
source activate rna
cat SRR_Acc_List.txt | while read id; do (prefetch  ${id} );done
ps -ef | grep prefetch | awk '{print $2}' | while read id; do kill ${id}; done #在内地下载速度很慢，所以我杀掉这些下载进程
```

- 或者直接使用我已经下载好的sra数据

```
mkdir $wkd/raw
cd $wkd/raw
ls /public/project/RNA/airway/sra/*  | while read id; do ( fastq-dump --gzip --split-3 -O ./ ${id}  ); done ## 批量转换sra到fq格式。
source deactivate
```

- 得到的SRA数据如下

```
/public/project/RNA/airway/sra/
├── [1.6G]  SRR1039508.sra
├── [1.4G]  SRR1039509.sra
├── [1.6G]  SRR1039510.sra
├── [1.5G]  SRR1039511.sra
├── [2.0G]  SRR1039512.sra
├── [2.2G]  SRR1039513.sra
├── [3.0G]  SRR1039514.sra
├── [1.9G]  SRR1039515.sra
├── [2.1G]  SRR1039516.sra
├── [2.6G]  SRR1039517.sra
├── [2.3G]  SRR1039518.sra
├── [2.0G]  SRR1039519.sra
├── [2.1G]  SRR1039520.sra
├── [2.4G]  SRR1039521.sra
├── [2.0G]  SRR1039522.sra
└── [2.2G]  SRR1039523.sra
```

- sra格式转fastq格式

> 格式转还用到的软件是fastq-dump

```
for i in $wkd/*sra
do
        echo $i
        fastq-dump --split-3 --skip-technical --clip --gzip $i  ## 批量转换
done
```

- 得到fastq数据如下

> 原始数据是双端测序结果，fastq-dump配合--split-3参数，一个样本被拆分成两个fastq文件

```
├── [1.3G]  SRR1039508_1.fastq.gz
├── [1.3G]  SRR1039508_2.fastq.gz
├── [1.2G]  SRR1039509_1.fastq.gz
├── [1.2G]  SRR1039509_2.fastq.gz
├── [1.3G]  SRR1039510_1.fastq.gz
├── [1.3G]  SRR1039510_2.fastq.gz
├── [1.2G]  SRR1039511_1.fastq.gz
├── [1.2G]  SRR1039511_2.fastq.gz
├── [1.6G]  SRR1039512_1.fastq.gz
├── [1.6G]  SRR1039512_2.fastq.gz
├── [950M]  SRR1039513_1.fastq.gz
├── [952M]  SRR1039513_2.fastq.gz
├── [2.4G]  SRR1039514_1.fastq.gz
......
├── [1.5G]  SRR1039522_1.fastq.gz
├── [1.5G]  SRR1039522_2.fastq.gz
├── [1.8G]  SRR1039523_1.fastq.gz
└── [1.8G]  SRR1039523_2.fastq.gz
```

step2: check quality of sequence reads

> fastqc生成质控报告，multiqc将各个样本的质控报告整合为一个。

```
ls *gz | xargs fastqc -t 2
multiqc ./
```

- 得到结果如下

```
├── [4.0K]  multiqc_data
│   ├── [2.1M]  multiqc_data.json
│   ├── [6.8K]  multiqc_fastqc.txt
│   ├── [2.2K]  multiqc_general_stats.txt
│    ├── [ 16K]  multiqc.log
│    └── [3.4K]  multiqc_sources.txt
├── [1.5M]  multiqc_report.html
├── [236K]  SRR1039508_1_fastqc.html
├── [279K]  SRR1039508_1_fastqc.zip
├── [238K]  SRR1039508_2_fastqc.html
├── [286K]  SRR1039508_2_fastqc.zip
├── [236K]  SRR1039510_1_fastqc.html
├── [278K]  SRR1039510_1_fastqc.zip
├── [241K]  SRR1039510_2_fastqc.html
├── [292K]  SRR1039510_2_fastqc.zip
......
├── [220K]  SRR1039522_fastqc.zip
├── [234K]  SRR1039523_1_fastqc.html
├── [273K]  SRR1039523_1_fastqc.zip
├── [232K]  SRR1039523_2_fastqc.html
└── [274K]  SRR1039523_2_fastqc.zip
```

> 每个id_fastqc.html都是一个质量报告，multiqc_report.html是所有样本的整合报告

step3: filter the bad quality reads and remove adaptors.

- 运行如下代码，得到名为config的文件，包含两列数据

```
mkdir $wkd/clean
cd $wkd/clean
ls /home/jmzeng/project/airway/raw/*_1.fastq.gz >1
ls /home/jmzeng/project/airway/raw/*_2.fastq.gz >2
paste 1 2  > config
```

- 打开文件 qc.sh，并且写入如下内容

> trim_galore，用于去除低质量和接头数据

```
source activate rna
bin_trim_galore=trim_galore
dir='/home/jmzeng/project/airway/clean'
cat $1 |while read id
do
        arr=(${id})
        fq1=${arr[0]}
        fq2=${arr[1]}
 $bin_trim_galore -q 25 --phred33 --length 36 --stringency 3 --paired -o $dir  $fq1 $fq2
done
source deactivate
```

- 运行qc.sh

```
bash qc.sh config #config是传递进去的参数
```

- 结果显示如下

```
├── [2.9K]  SRR1039508_1.fastq.gz_trimming_report.txt
├── [1.2G]  SRR1039508_1_val_1.fq.gz
├── [3.1K]  SRR1039508_2.fastq.gz_trimming_report.txt
├── [1.2G]  SRR1039508_2_val_2.fq.gz
├── [2.9K]  SRR1039509_1.fastq.gz_trimming_report.txt
......
├── [2.9K]  SRR1039522_1.fastq.gz_trimming_report.txt
├── [1.4G]  SRR1039522_1_val_1.fq.gz
├── [3.1K]  SRR1039522_2.fastq.gz_trimming_report.txt
├── [1.4G]  SRR1039522_2_val_2.fq.gz
├── [2.9K]  SRR1039523_1.fastq.gz_trimming_report.txt
├── [1.7G]  SRR1039523_1_val_1.fq.gz
├── [3.1K]  SRR1039523_2.fastq.gz_trimming_report.txt
└── [1.7G]  SRR1039523_2_val_2.fq.gz
```

step4: alignment

> star, hisat2, bowtie2, tophat, bwa, subread都是可以用于比到的软件

- 先运行一个样本，测试一下

```
mkdir $wkd/test
cd $wkd/test
source activate rna
ls $wkd/clean/*gz |while read id;do (zcat ${id}|head -1000> $(basename ${id} ".gz"));done
id=SRR1039508
hisat2 -p 10 -x /public/reference/index/hisat/hg38/genome -1 ${id}_1_val_1.fq   -2 ${id}_2_val_2.fq  -S ${id}.hisat.sam
subjunc -T 5  -i /public/reference/index/subread/hg38 -r ${id}_1_val_1.fq -R ${id}_2_val_2.fq -o ${id}.subjunc.sam
bowtie2 -p 10 -x /public/reference/index/bowtie/hg38  -1 ${id}_1_val_1.fq   -2 ${id}_2_val_2.fq  -S ${id}.bowtie.sam
bwa mem -t 5 -M  /public/reference/index/bwa/hg38 ${id}_1_val_1.fq   ${id}_2_val_2.fq > ${id}.bwa.sam
```

- 批量比对代码

```
cd $wkd/clean
ls *gz|cut -d"_" -f 1 |sort -u |while read id;do
ls -lh ${id}_1_val_1.fq.gz   ${id}_2_val_2.fq.gz
hisat2 -p 10 -x /public/reference/index/hisat/hg38/genome -1 ${id}_1_val_1.fq.gz   -2 ${id}_2_val_2.fq.gz  -S ${id}.hisat.sam
subjunc -T 5  -i /public/reference/index/subread/hg38 -r ${id}_1_val_1.fq.gz -R ${id}_2_val_2.fq.gz -o ${id}.subjunc.sam
bowtie2 -p 10 -x /public/reference/index/bowtie/hg38  -1 ${id}_1_val_1.fq.gz   -2 ${id}_2_val_2.fq.gz  -S ${id}.bowtie.sam
bwa mem -t 5 -M  /public/reference/index/bwa/hg38 ${id}_1_val_1.fq.gz   ${id}_2_val_2.fq.gz > ${id}.bwa.sam
done
```

这里是演示多个比对工具，但事实上，对RNA-seq数据来说，不要使用bwa和bowtie这样的软件，它需要的是能进行跨越内含子比对的工具。

- sam文件转bam

```
ls *.sam|while read id ;do (samtools sort -O bam -@ 5  -o
$(basename ${id} ".sam").bam   ${id});done
rm *.sam
```

- 为bam文件建立索引

```
ls *.bam |xargs -i samtools index {}
```

- reads的比对情况统计

```
ls *.bam |xargs -i samtools flagstat -@ 2  {}  >
ls *.bam |while read id ;do ( samtools flagstat -@ 1 $id >
 $(basename ${id} ".bam").flagstat  );done
source deactivate
```

- 最终结果显示如下

```
├── [1.8G]  SRR1039508.bowite2.bam
├── [2.9M]  SRR1039508.bowite2.bam.bai
├── [ 444]  SRR1039508.bowite2.flagstat
├── [ 10G]  SRR1039508.bowite2.sam
├── [1.7G]  SRR1039509.bowite2.bam
......
├── [2.0G]  SRR1039521.bowite2.bam
├── [2.9M]  SRR1039521.bowite2.bam.bai
├── [ 444]  SRR1039521.bowite2.flagstat
├── [ 10G]  SRR1039521.bowite2.sam
├── [2.3G]  SRR1039522.bowite2.bam
├── [3.0M]  SRR1039522.bowite2.bam.bai
├── [ 444]  SRR1039522.bowite2.flagstat
├── [ 12G]  SRR1039522.bowite2.sam
├── [2.5G]  SRR1039523.bowite2.bam
├── [3.0M]  SRR1039523.bowite2.bam.bai
├── [ 444]  SRR1039523.bowite2.flagstat
└── [ 14G]  SRR1039523.bowite2.sam
```

step5: counts

```
mkdir $wkd/align
cd $wkd/align
source activate rna
# 如果一个个样本单独计数，输出多个文件使用代码是：
for fn in {508..523}
do
featureCounts -T 5 -p -t exon -g gene_id  -a /public/reference/gtf/gencode/gencode.v25.annotation.gtf.gz -o $fn.counts.txt SRR1039$fn.bam
done
# 如果是批量样本的bam进行计数，使用代码是：
mkdir $wkd/align
cd $wkd/align
source activate rna
gtf="/public/reference/gtf/gencode/gencode.v25.annotation.gtf.gz"
featureCounts -T 5 -p -t exon -g gene_id  -a $gtf -o all.id.txt  *.bam  1>counts.id.log 2>&1 &
# 这样得到的  all.id.txt  文件就是表达矩阵啦，但是，这个 featureCounts有非常多的参数可以调整。
source deactivate
```

- 得到的文件如下

```
      1 # Program:featureCounts v1.6.1;
Command:"featureCounts" "-T" "5" "-p" "-t" "exon" "-g"
"gene_id" "-a" "/public/reference/gtf/gencode/ge
      2 Geneid  Chr     Start   End     Strand  Length
/home/llwu/RNA/airway/2.align/bowite2/SRR1039523.bowite2.ba
m
      3 ENSG00000223972.5
chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1
 11869;12010;12179;12613;12613;12975;13221;13221;13453
12227;1
      4 ENSG00000227232.5
chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1
 14404;15005;15796;16607;16858;17233;17606;17915;18268;2
      5 ENSG00000278267.1       chr1    17369   17436   -
    68      9
      6 ENSG00000243485.4
chr1;chr1;chr1;chr1;chr1;chr1
29554;30267;30366;30564;30976;30976
30039;30667;30503;30667;31097;31109
      7 ENSG00000237613.2       chr1;chr1;chr1;chr1;chr1
     34554;35245;35277;35721;35721
35174;35481;35481;36073;36081   -;-;-;-;-
      8 ENSG00000268020.3       chr1    52473   53312   +
    840     0
      9 ENSG00000240361.1       chr1    62948   63887   +
    940     0
     10 ENSG00000186092.4       chr1    69091   70008   +
    918     0
```

上面的文件，请务必仔细了解。

作者：生信技能树 链接：https://www.jianshu.com/p/a84cd44bac67 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。