**“Linux 生物信息技术基础”课程小组总结报告**

组：G12 次：12 组长：马小凯 执笔：郑弘熙

1. **时间**

2023年 5 月 27 日

1. **方式**

线上腾讯会议

1. **主题**

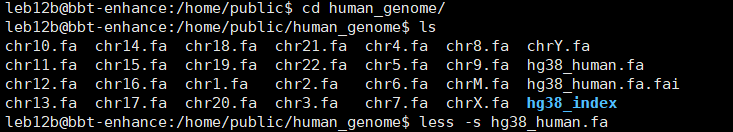
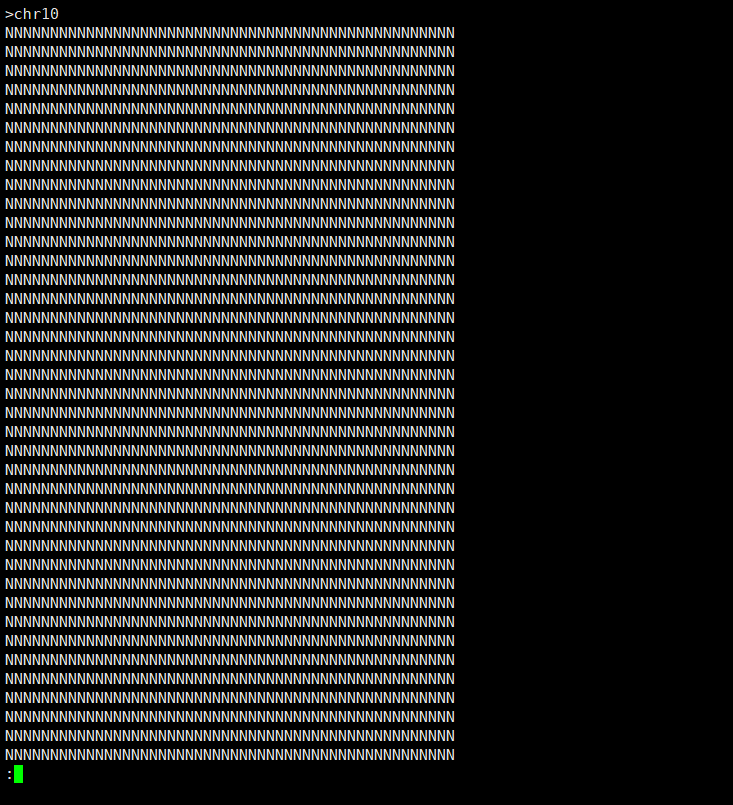
ChIP-seq

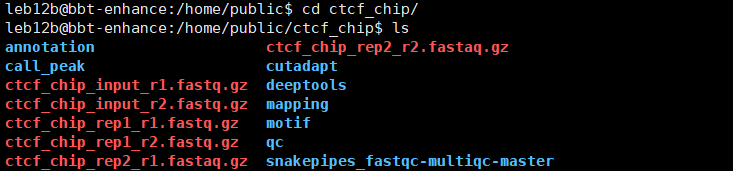
1. **内容**

**ChIP-seq分析流程——第一部分**

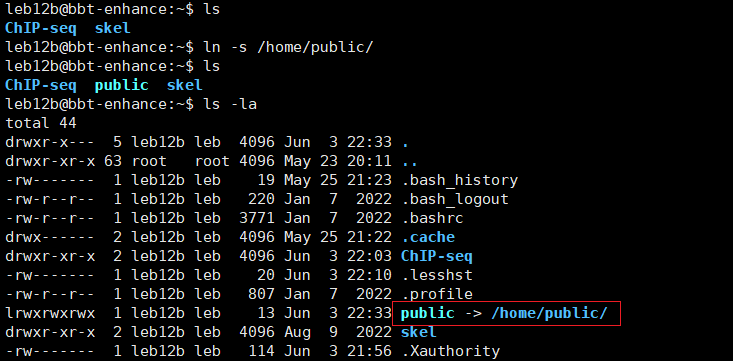
登录服务器后查看public文件夹内文件

查看human genome文件夹内文件。hg38\_human.fa为UCSC的人类参考基因组，其中展示了人类24条基本完整的染色体序列，即1-22，X，Y以及完整的线粒体序列，如下图所展示的N表示未检测出来的序列；其余的chr10.fa等是其中每条染色体的序列；hg38\_index则是索引文件，用于比对。



实验组为rep，其中进行了两次重复，分别为rep1和rep2，而input则是DNA-蛋白质复合物在抗体富集之前的样品直接测序得到的结果，接近全基因组测序，这里用作生物学对照，但是此处**更应该用IgG或多种IgG富集后的样品作为对照**。

与public文件夹建立软链接，方便调用其中的内容！

**#代码：**ln -s 源文件（夹）地址 目标文件夹地址

1. **对原始测序数据进行质控和过滤**
   1. **使用fastqc对测序reads进行质量检测**

**#代码说明：**

# fastqc [options] fastq1 [fastq2 ...] > output.log 2>&1 &

# -t 指定使用的线程数

# -o 指定输出文件夹

# > output.log 输出至output.log文件中，将标准输出重定位至log文件中

# 2>&1 将标准错误和标准输出合并到一个文件中

# & 最后的&使其后台运行

**#代码**

#fastqc -t 24 -o results/qc/ public/ctcf\_chip/ctcf\_chip\_input\_r1.fastq.gz public/ctcf\_chip/ctcf\_chip\_input\_r2.fastq.gz public/ctcf\_chip/ctcf\_chip\_rep1\_r1.fastq.gz public/ctcf\_chip/ctcf\_chip\_rep1\_r2.fastq.gz public/ctcf\_chip/ctcf\_chip\_rep2\_r1.fastaq.gz public/ctcf\_chip/ctcf\_chip\_rep2\_r2.fastaq.gz > results/qc/fastqc.log 2>&1 &

过程中可以使用cat results/qc/fastqc.log查看运行情况。使用ps -ef查看进程，使用ps -ef | grep fastqc查看含有fastqc名字的进程。kill PID根据PID进程号杀死进程，-9强制杀死

结果如下展示

使用multiqc可以整合fastqc结果，生成一份整体的分析报告。

**#代码：**multiqc . 对当前文件夹下文件进行整合

* 1. **使用cutadapt进行过滤，去接头**

**#代码：**cutadapt -j 2 --time 1 -e 0.1 -O 3 --quality-cutoff 25 -m 55

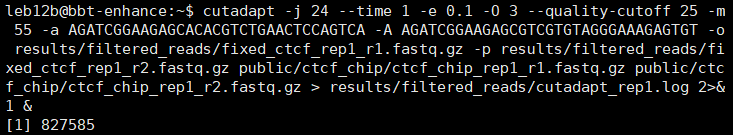
-a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA

-A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT

-o filtered\_reads/fixed\_ctcf\_rep1\_read1.fastq.gz

-p filtered\_reads/fixed\_ctcf\_rep1\_read2.fastq.gz

ctcf\_rep1\_read1.fastq.gz ctcf\_rep1\_read2.fastq.gz

> filtered\_reads/cutadapt.log 2>&1 &

**#代码说明**

# -j 2 设置线程数

# --times 1 只去处一次接头，因为接头只出现一次

# -e 0.1 去除的序列与adaptor相比，missmatch率低于该值，则认为是adaptor，一般设置为0.1

# -O 3 当与adaptor overlap的碱基数大于等于该值时，才进行去除

# --quality-cutoff 25 小于等于该qulity的碱基去除

# -m 55 trim之后低于该值的一对reads丢弃，一般要大于40

# -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA read1 3'的adaptor

# -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT read2 3'的adaptor

# -o fix.fastq/test\_R1\_cutadapt.temp.fq.gz read1的输出文件

# -p fix.fastq/test\_R2\_cutadapt.temp.fq.gz read2的输出文件

# Read1输入文件

# Read2输入文件

``````

cutadapt结果的解读<https://cutadapt.readthedocs.io/en/stable/guide.html#reporting>

1. **将过滤后结果比对至基因组上（Mapping）**

主流方法：bowtie，bowtie2，bwa，hisat2，tophat2，star，hisat3n

* 1. **Builing index**

**#代码**

# bowtie2-build -f ../hg38\_human.fa hg38\_human\_index > build\_index.log 2>&1 &

**#代码说明**

# bowtie2-build [options]\* <reference\_in> <bt2\_base>

# -f 基因组为fasta格式

**BWT算法构建index：https://chengjunwen.github.io/2016/11/08/bwt/**

* 1. **Mapping**
     1. **比对过程**

ChIP-Seq的Mapping相对简单，原因如下

1. 不像RNA的Mapping, 不需要考虑exon和intron
2. 不用考虑正负链信息的问题
3. 不像Bisulfite-Seq, 需要比对三元碱基

使用默认参数即可

**#代码**

# bowtie2 -x /index/hg38\_human\_index

-1 ../filtered\_reads/fixed\_ctcf\_rep1\_r1.fastq.gz

-2 ../filtered\_reads/fixed\_ctcf\_rep1\_r2.fastq.gz

-p 4

-S ctcf\_chip\_mapped\_rep1.sam > bowtie2\_mapping\_rep1.log 2>&1 &

**#代码说明**

# -x index file

# -1 read1 fastq file

# -2 read2 fastq file

# -p threads

# -S out SAM file

* + 1. **BWT算法**

**BWT算法比对：**[**https://www.jianshu.com/p/f94b2631681c**](https://www.jianshu.com/p/f94b2631681c)

**Bowtie2 official site：<https://bowtie-bio.sourceforge.net/bowtie2/index.shtml>**

（1） 启发式算法

（2） bowtie2支持gap，但是seed序列不能有gap

* + 1. **Bowtie2比对策略**

**（1）Paired-end mode:** 一对reads应该按照他们正确的方向（反向平行）、以及合适的距离（使用`-X`, `-I`参数可以限制该距离）比对到基因组上，这种比对被称为和谐的比对（cordant alignments），其他的被称为不和谐的比对（discordant alignments）；通常，bowtie2两种比对结果都会输出，除非使用`--no-discordant`参数

**（2）Mixed mode：**如果bowtie2找不到一对reads的成对比对，就会以非配对的方式比对这一对中的两条reads，使用`--no-mixed`参数可以关闭这个功能

**（3）End to End mode:**与loacl mode相对，reads的两端必须比对到基因组，除非用其他trim参数将reads两端的碱基修剪掉

**报告的比对结果：默认搜索多个比对结果，但是仅报告比对结果最好的**

**SAM文件的解读：**[**http://samtools.github.io/hts-specs/SAMv1.pdf**](http://samtools.github.io/hts-specs/SAMv1.pdf)

**SAM文件各列的含义**

1：序列名称

2：SAM FLAG：对SAM过滤的重要标准

3：比对到的参考序列明（染色体名）

4：比对到的第一个位置（没有比对上则记录为0）

5：比对质量，MAPQ（255表示质量不可用）

6：比对的具体方式/CIGAR

7：mate比对到的染色体，（没有比对上或没有mate记录为\*，完全比对记录为=）

8：mate比对的第一个位置（0表示不可用）

9 ：文库插入片段的长度（无mate则为0）

10：read的序列

11：read的测序质量

12：附加的标记

* + 1. **比对结果的排序、过滤、去重、索引**
       1. **SAM文件排序，生成BAM文件**

**#代码**

# samtools sort -O BAM -o ctcf\_chip\_mapped\_rep1.bam -@ 4 -m 4G ctcf\_chip\_mapped.sam

**#代码说明**

# 按照染色体坐标排序

# 如果需要构建BAM文件的index文件，必须按照染色体坐标排序

* + - 1. **BAM文件过滤**

**#代码**

# samtools view -bS -q 30 -@ 4 -m 4G -f 1 -f 2 -o ctcf\_chip\_mapped\_rep1\_filtered.bam ctcf\_chip\_mapped\_rep1.bam

**Samtools Flag的解释：**[**https://broadinstitute.github.io/picard/explain-flags.html**](https://broadinstitute.github.io/picard/explain-flags.html)