Course Practical 2 : Short Read Alignment and Quality Control

Shamith Samarajiiwa, Dora Bihary

18th September 2017

Contents

---

1 Short Read Alignment and Quality Control

Introduction to the dataset used in this part of the course

I'll be using ChIP-seq and RNA-seq datasets to demonstrate how to align ChIP-seq and RNA-seq data to the `GRCh38` reference genome. The data-set for this practical is a publicly available dataset downloaded from the `NCBI GEO` repository with the accession `GSE15780`. [https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE15780] It's a study "Crosstalk between c-Jun and TAp73alpha/beta contributes to the apoptosis-survival balance" PMID:21459846 by Koeppel et al. This study explores genome wide binding of transcription factors (`TP53` and `TP73` splice variants) and gene expression in a human cancer cell line.

1.1 This practical consists of 5 sections:

1. How to download public data-sets from repositories
2. `BWA` alignment of ChIP-seq to `GRCh38` reference
3. `SAM Tools` tutorial
4. `Bowtie2` alignmnet of ChIP-seq to `GRCh38` reference
5. `STAR` alignment of RNA-seq data

2 Downloading fastq files from public sequence repositories

We downloaded the dataset (`fastq files`) from the Sequence Read Archive using the `SRA-toolkit`. There are multiple ways of doing this.

1. Browse the **SRA dababase** and download the data.

2. Use **SRA toolkit**. You need to install and configure this on your computer first. Detailed instructions are here.
3. Use the Bioconductor package **SRAdb** to search and download the sra or fastq files.

The files you need are in **/home/participant/Course_Materials/Introduction/SS_DB/Raw_Data/**. These are Large files, so do not run this bit of R code below! It's there just to show you how to download the files from the Sequence Read Archive.

```
print("Don't run me!!")

#setup SRAdb
# This will download a very large (~30 Gb) file!

library(SRAdb)

sqlfile <- 'SRAmetadb.sqlite'

if(!file.exists('SRAmetadb.sqlite')) sqlfile <<- getSRAdbFile()

#establish a connection to the database
sra_con <- dbConnect(SQLite(),sqlfile)

# get SRR runs
# You need to give it the Experiment ID (SRX) of the dataset you would like to download

rs = listSRAfile(c("SRX016980"), sra_con, fileType = 'sra' )


# download the SRR file
getSRAfile(c("SRR036615"), sra_con, fileType='sra')

# convert to fastq using SRA Toolkit (you needs to install the SRA-toolkit on your computer)
[](https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/)

system ("fastq-dump SRR036615.lite.sra")

# or get the fastq file directly from EBI using ftp
getFASTQinfo(c("SRR036615"), sra_con, srcType = 'ftp' )
getSRAfile(c("SRR036615"), sra_con, fileType = 'fastq' )
```

Reference genomes can be downloaded from UCSC, Ensembl or NCBI genome resources. Here's the UCSC genome browser url for the human reference GRCh38:

[http://hgdownload.soe.ucsc.edu/goldenPath/hg38/chromosomes/]

Download instructions are on that page.

Whole genomic alignments can be time consuming and not realistic to do in the short time we have. Therefore, we downloaded and preprocessed a single chromosome (chr3)from the above dataset to save time. The preprocessing step included aligning to a *GRCh38* genome with a *sponge-database* (which removes artefacts and non-chromosomal sequences) and then regenerating the chr3 fastq file. Because we've done this for you there is no need for you to use a sponge database again when you do the tutorial below.

3 Sequence alignment with BWA

```
# Using BWA to align a fastq ChIP-seq sample to the GRCh38 reference genome

cd /home/participant/Course_Materials/Introduction/SS_DB/Alignment/BWA
```

create a BWA hg38chr3 index. We will use the bwa index command.

-p option give the index name (you can call this whatever you want, we named it hg38chr3bwaidx) -a option chooses one of the alignment algorithm within bwa

Normally you would use a complete genome build fasta (hg38.fa) file to build a bwa index. In this case we're using chromosome3 hg38_chr3.fa (again to save time).

```
bwa index -p hg38chr3bwaidx \
-a bwtsw /home/participant/Course_Materials/Introduction/SS_DB/Reference/BWA/hg38_chr3.fa
```

Align to hg38 and generate the single end alignment SAM file

```
# uses mem algorithm, -M  This option leaves the best (longest) alignment for a read as is
# but marks additional alignments for the read as secondary
# -t = number of processor cores

bwa mem -M -t 4 hg38chr3bwaidx \
/home/participant/Course_Materials/Introduction/SS_DB/RawData/ChIPseq/tp53_r2.fastq.gz \
> tp53_r2.fastq.sam
```

4 A SAM Tools tutorial

Generate BAM file. This example shows how to use the Samtools program.

```
# look at the first 10 lines of your SAM file
head tp53_r2.fastq.sam

#convert to BAM
samtools view -bT ~/Course_Materials/Introduction/SS_DB/Reference/BWA/hg38_chr3.fa tp53_r2.fastq.sam > tp5
```

Sort BAM file

```
samtools sort -T sorted tp53_r2.fastq.bam -o tp53_r2.fastq_sorted.bam
```

Generate BAM index

```
samtools index tp53_r2.fastq_sorted.bam tp53_r2.fastq_sorted.bai
```

Convert BAM to SAM

```
samtools view -h tp53_r2.fastq_sorted.bam > tp53_r2.fastq_sorted_anotherCopy.sam
```

Filter unmapped reads in BAM

```
samtools view -h -F 4 tp53_r2.fastq_sorted.bam > tp53_r2.fastq_sorted_onlymapped.bam
```

If you need help decoding SAM flags

If we want all reads mapping within the genomic coordinates chr3:200000-500000

```
samtools view tp53_r2.fastq_sorted.bam chr3:200000-500000 >tp53_r2.fastq_sorted_200-500k.bam
```

Simple statistics using SAM Tools flagstat

```
#index the bam file first
samtools flagstat tp53_r2.fastq_sorted.bam
```

Create a `fastq` file from a `BAM` file

```
samtools bam2fq tp53_r2.fastq_sorted.bam > tp53_r2.new_allreads.fastq
```

How to do this using `bedtools`

```
bedtools bamtofastq -i input.bam -fq output.fastq

#paired-end reads:
samtools sort -n input.bam -o input_sorted.bam  # sort by read name (-n)

bedtools bamtofastq -i input_sorted.bam -fq output_r1.fastq -fq2 output_r2.fastq
```

Run `SAMStat` to asses BAM QC

```
samstat tp53_r2.fastq_sorted.bam
```

Generate a tdf (tile data format) file for viewing in `IGV` browser.

```
igvtools count -z 5 -w 25 -e 250 \
tp53_r2.fastq_sorted.bam tp53_r2.fastq_sorted.tdf hg38
```

5 Sequence alignment with bowtie2

```
cd /home/participant/Course_Materials/Introduction/SS_DB/Alignment/Bowtie
```

To get a list of options

```
bowtie2 -h
```

First step is to build a database (index)

```
`bowtie2-build -f genome.fa dbname`

bowtie2-build -f /home/participant/Course_Materials/Introduction/SS_DB/Reference/Bowtie/hg38_chr3.fa \
hg38_chr3
```

Align to chr3

```
bowtie2 -x hg38_chr3 \
-U /home/participant/Course_Materials/Introduction/SS_DB/RawData/ChIPseq/tp53_r2.fastq.gz \
-S tp53_r2.sam
```

```
samtools view -Sb tp53_r2.sam > tp53_r2.bam
```

```
samtools sort tp53_r2.bam > tp53_r2_sorted.bam
samtools index tp53_r2_sorted.bam
```

6 Transcriptome alignment with STAR

Generate genome indices:

```
cd /home/participant/Course_Materials/Introduction/SS_DB/Alignment/STAR

STAR --runThreadN 4 --runMode genomeGenerate \
--genomeDir /home/participant/Course_Materials/Introduction/SS_DB/Reference/STAR/ \
--genomeFastaFiles /home/participant/Course_Materials/Introduction/SS_DB/Reference/STAR/hg38_chr3.fa
```

Download an annotation GTF file and unzip it

```
# if your reference genome is from Ensembl get GTF file from Ensembl else get from UCSC table
# browser
#wget ftp://ftp.ensembl.org/pub/release-90/gtf/homo_sapiens/Homo_sapiens.GRCh38.90.gtf.gz

#chmod 755 Homo_sapiens.GRCh38.90.gtf.gz
#gunzip Homo_sapiens.GRCh38.90.gtf.gz

#Get gtf from ucsc table browser and name it hg38.gtf
# Instructor will demonstrate this
```

```
STAR --runThreadN 4 --genomeDir \
/home/participant/Course_Materials/Introduction/SS_DB/Reference/STAR/ \
--readFilesIn \
/home/participant/Course_Materials/Introduction/SS_DB/RawData/RNAseq/tp53_rep1_trimmed.fastq.gz \
--readFilesCommand zcat --outFileNamePrefix RNA  --outSAMtype BAM SortedByCoordinate \
--sjdbGTFfile /home/participant/Course_Materials/Introduction/SS_DB/Alignment/STAR/hg38.gtf \
--sjdbOverhang 100 --twopassMode Basic --outWigType bedGraph --outWigStrand Stranded
```

While STAR is running, the status messages will be appearing on the screen and the progress of the mapping job can be checked in the Log.progress.out

This should give you basic skills for doing next generation sequence alignment, and this is also the end of this practical!