Introduction to Bulk RNAseq data analysis

QC of Aligned Reads

Checking the quality of the aligned data

Once we have aligned our reads, it is then possible to run some additional quality checks on our data. In the first instance we will just look at some basic metrics such as the percentage of aligned reads and the duplication rates. We can then use some more sophisticated methods to assess the integrity of our RNA and genomic locations to which our reads are aligning.

For the purposes of this training session we will work with a bam files that only contains reads aligned to chromosome 14. Running each tool on the full bam file would take too long for the practical. You should find a file called `SRR7657883.chr14.sorted.bam` under the directory `bam`, please use this rather than the bam file you created in the previous session.

1. A quick look at the alignment metrics with `samtools`

We previously used the `samtools` package to sort and index our bam file. `samtools` also includes some simple tools to generate alignment metrics. This can be very handy for a quick first look at our alignment quality. The tool we are going to use is called `flagstat`.

The **Usage** is:

```
samtools flagstat <in.bam>
```

Where `<in.bam>` is the bam file we wish to QC.

Exercise 1

1. Use the `samtools flagstat` command to generate alignment metrics for the bam file `bam/SRR7657883.chr14.sorted.bam`.

Q) What percentage of the reads have aligned to the genome?

2. More detailed metrics with Picard Tools

Picard Tools is a suite of tools for analysing and manipulating sequencing data. It is maintained by the Broad Institute and comprises 88 different tools for doing jobs such as generating QC metrics, modifying bam files in various ways, or converting files between different formats.

We are going to use three QC metrics tools to get a variety of different important metrics.

Picard is a java based programme and so to run a particular Picard tool the general format for the command is:

```
java -jar picard/picard.jar PicardToolName OPTION1=value1 OPTION2=value2 ...
```

- `picard.jar` is the **J**ava **AR**chive file that contains all the tools. It can be found in the *picard* directory under *Course_Materials*.
- `PicardToolName` is the name of the tools that you wish to use.

This is then followed by a series of options/arguments for that particular tool. Each tool has specific options and arguments and you can check these most easily by going to the Picard website, or by using the '–help" command:

```
java -jar picard/picard.jar PicardToolName --help
```

2.1 Duplication metrics

The first tool we are going to use is `MarkDuplicates`. This tool actually performs two tasks.

First, Picard reads through the bam file and finds any duplicate reads - these are reads with the same 5' position. For each group of duplicate reads, Picard selects a "primary" read based on the base quality scores and then marks all other reads in the group as "duplicates" by adding **1024** to the sam flag. For paired end reads, the entire fragment is considered, this essentially means both ends must be duplicated for the read pair to be marked as a duplicate.

Once the duplicate reads are marked, Picard then also generates a metrics file that contains information about the duplication rate.

Exercise 2.1

1. Run Picard's MarkDuplicates tool on the sorted bam file using the following command:

```
java -jar picard/picard.jar MarkDuplicates \
          INPUT=bam/SRR7657883.chr14.sorted.bam \
          OUTPUT=bam/SRR7657883.chr14.mkdup.bam \
          METRICS_FILE=bam/SRR7657883.chr14.mkdup_metrics.txt \
          CREATE_INDEX=true
```

Note: The \ at the end of each line tells the terminal that when you press `Enter`, you have not yet finished typing the command. You can if you wish, type the whole command on a single line, omitting the \. The command is written across multiple lines here just to make it easier to read.

Q. What is the duplication rate for this bam file? You'll need to look at the metrics file. The easiest way is to open in a spreadsheet. On the course machines we have LibreOffice Calc. You can find this in the launcher bar at the bottom of the desktop.

2.2 Alignment metrics

Next we will collect some detailed alignment metrics using the `CollectAlignmentSummaryMetrics` tool. In this case we need to provide an input bam, the name of the output metrics file and the fasta file with the genomic reference.

Exercise 2.2

1. Run Picard's `CollectAlignmentSummaryMetrics` tool on the chr14 sorted bam providing the following options.
   - INPUT - The sorted chr 14 only bam file
   - OUTPUT - `bam/SRR7657883.chr14.alignment_metrics.txt`
   - REFERENCE_SEQUENCE - `references/Mus_musculus.GRCm38.dna_sm.primary_assembly.fa`

2.3 Insert Size metrics

Next we will collect some metrics that relate to the *insert size*. This is the size of the fragment of RNA that each read pair has originated from. The pair of reads are just the ends of this fragment. We will use Picard's `CollectInsertSizeMetrics` tool. In this case we need to provide an input bam, the name of the output metrics file and a name for a pdf which will contain an plot showing the distribution of insert sizes in our library.

Exercise 2.3

1. Run Picard's `CollectInsertSizeMetrics` tool on the chr14 sorted bam providing the following options.
   - INPUT - The sorted chr 14 only bam file
   - OUTPUT - `bam/SRR7657883.chr14.insert_size.txt`
   - HISTOGRAM_FILE - `bam/SRR7657883.chr14.insert_size.pdf`
   - REFERENCE_SEQUENCE - `references/Mus_musculus.GRCm38.dna_sm.primary_assembly.fa`

   Open the PDF and look at the distribution fragment lengths (insert sizes) in the library. Considering this data is from paired 150 base reads, what are the implications of the fragment length distributions.

2.4 RNA alignment metrics

The `CollectRnaSeqMetrics` tool produces metrics describing the distribution of the reads across different genomic locations - intronic, exonic, intergenic, UTR - and the distribution of bases within the transcripts.

The `CollectRnaSeqMetrics` requires four pieces of information to run:

1. The input bam file
2. A file name for the output metrics file
3. A file containing gene annotations in a format called RefFlat that is defined here. This format can be generated from a gtf file. We've already generated this file for you at `references/Mus_musculus.GRCm38.102.txt`
4. A parameter for strand specificity. Strand specificity is factor of the library prep chemistry. There are three possible options:
   a) Forward stranded library prep, i.e. the reads are on the transcription strand - FIRST_READ_TRANSCRIPTION_ST
   b) Reverse stranded library prep, i.e. the reads are on the reverse strand - SECOND_READ_TRANSCRIPTION_STRA
   c) Unstranded library prep, i.e. the reads may be on either strand - NONE
   With your own data you would need to find this information out from whoever has prepared the library. In this case our library prep is unstranded, so we will use `NONE`.

Exercise 2.4

1. Run Picard's `CollectRnaSeqMetrics` tool on the sorted bam file providing the following options:
   - INPUT - The sorted bam file
   - OUTPUT - `bam/SRR7657883.chr14.RNA_metrics.txt`
   - REF_FLAT - the RefFlat reference file
   - STRAND - `NONE`

The results of this analysis are best viewed graphically, we will do this in the next exercise.

3. Visualising QC results with MultiQC

MultiQC is a tool for collating multiple QC results files into a single report. Its use is simple, you just run the command `multiqc` on the directory containing your metrics files. The general command is:

```
multiqc <directory containing metrics files>
```

We will add a couple of options to control the output directory and the name of the report.

Exercise 3.1

1. Run multiqc on the bam directory:

```
multiqc -n Alignment_QC_Report.html -o bam bam
```

   - `-n` - a name for the report
   - `-o` - the directory in which to place the report

2. Open the html report that was generated by multiqc and inspect the QC plots The easiest way to do this is type `xdg-open multiqc_report.html`, which will open the report in a web browser.

Exercise 3.2

In the `metrics` directory you should find Picard metrics for all of the bam files.

1. Run multiqc on the contents of the metrics directory.

2. Open the html report that was generated by multiqc and inspect the QC plots

Q. Are there any bam files that look problematic?
