RNA-seq analysis in R

Read alignment with HISAT2 - exercise solutions

1 Indexing the genome for Hisat2

Exercise 1

1. Check your current working directory and if necessary navigate to the `Course_Materials/` directory using the command `cd` (change directory).

`pwd` - to check **p**resent **w**working **d**irectory, then if necessary:

```
cd ~/Course_Materials
```

2. Use `ls` to list the contents of the directory.

```
ls
```

3. Use `ls references` to list the contents of the `references` directory.

```
ls references
```

4. We need a directory for hisat2 to write the index files in. Make a directory (`mkdir`) inside the `references` directory called `hisat2_index_chr14`:

```
mkdir references/hisat2_index_chr14
```

5. To create the hisat2 index run the following command:

```
hisat2-build -p 7 \
    references/Mus_musculus.GRCm38.dna_sm.chr14.fa \
    references/hisat2_index_chr14/mmu.GRCm38
```

Questions:
a) Why do we use `-p 7`? Take a look at `hisat2-build` help.

```
hisat2-build --help
```

```
HISAT2 version 2.2.1 by Daehwan Kim (infphilo@gmail.com, http://www.ccb.jhu.edu/people/infphilo
Usage: hisat2-build [options]* <reference_in> <ht2_index_base>
    reference_in            comma-separated list of files with ref sequences
    hisat2_index_base       write ht2 data to files with this dir/basename
Options:
    -c                      reference sequences given on cmd line (as
                            <reference_in>)
    --large-index           force generated index to be 'large', even if ref
                            has fewer than 4 billion nucleotides
    -a/--noauto             disable automatic -p/--bmax/--dcv memory-fitting
    -p <int>                number of threads
    --bmax <int>            max bucket sz for blockwise suffix-array builder
    ...
    ...
```

The `-p` flag is used to instruct hisat2 about how many threads (processors) it should use when running an operation. Using multiple processors in parallel speeds up the analysis. In our case, the machines we are using have 8 processors and so we tell hisat2 to use 7 of these which leaves one free.

b) How many files are created?

```
ls references/hisat2_index_chr14/
```

```
mmu.GRCm38.1.ht2
mmu.GRCm38.2.ht2
mmu.GRCm38.3.ht2
mmu.GRCm38.4.ht2
mmu.GRCm38.5.ht2
mmu.GRCm38.6.ht2
mmu.GRCm38.7.ht2
mmu.GRCm38.8.ht2
```

Hisat2 always creates 8 index files that start with our base name end with `.X.ht2`. So in this case we have `mmu.GRCm38.1.ht2` to `mmu.GRCm38.8.ht2`.

2 Align with Hisat2

Exercise 2

Use HISAT2 to align the fastq file.

We wil be writing the output to the `bam` directory. You will notice there is already a file in there called `SRR7657883.chr14.sorted.bam`. This contains reads aligned to chromosome 14. We will be using this file in some exercises later in the course, try not to write over it or delete it.

Use the following parameters:

- Index (the full genome this time) - `references/hisat2_index/mmu.GRCm38`
- Fastq file #1 mate (read 1) - `fastq/SRR7657883.sra_1.fastq.gz`
- Fastq file #2 mate (read 2) - `fastq/SRR7657883.sra_2.fastq.gz`
- Output file - `bam/SRR7657883.sam`
- Set the number of threads (number of processors to use) to 7 - check the help page to find the appropriate flag

```
hisat2 --help
```

```
HISAT2 version 2.2.1 by Daehwan Kim (infphilo@gmail.com, www.ccb.jhu.edu/people/infphilo)
Usage:
  hisat2 [options]* -x <ht2-idx> {-1 <m1> -2 <m2> | -U <r>} [-S <sam>]
...
...
...

Options (defaults in parentheses):

 Input:
  -q                 query input files are FASTQ .fq/.fastq (default)
  --qseq             query input files are in Illumina's qseq format
```

```
  -f                query input files are (multi-)FASTA .fa/.mfa
  -r                query input files are raw one-sequence-per-line
...
...
...
... **almost right at the bottom** ...
...
Performance:
  -o/--offrate <int> override offrate of index; must be >= index's offrate
  -p/--threads <int> number of alignment threads to launch (1)
...
...
```

- Add the `-t` flag so that HISAT2 will print a time log to the console

```
hisat2 -x references/hisat2_index/mmu.GRCm38 \
       -1 fastq/SRR7657883.sra_1.fastq.gz \
       -2 fastq/SRR7657883.sra_2.fastq.gz \
       -S bam/SRR7657883.sam \
       -p 7 \
       -t
```

⇒ *bam/SRR7657883.sam*

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.

3 Convert the SAM output to BAM

Exercise 3

1. Transform your aligned SAM file in to a BAM file called `SRR7657883.bam`. Use the option `-@ 7` to use 7 cores. This vastly speeds up the compression.

```
samtools view -b -@ 7 bam/SRR7657883.sam > bam/SRR7657883.bam
```

⇒ *bam/SRR7657883.bam*

2. Sort the BAM file to a create a bam files called `SRR7657883.sorted.bam`. Again use the `-@ 7` options to use 7 cores.

```
samtools sort -@ 7 bam/SRR7657883.bam > bam/SRR7657883.sorted.bam
```

⇒ *bam/SRR7657883.sorted.bam*

3. Index the sorted BAM file

```
samtools index bam/SRR7657883.sorted.bam
```

⇒ *bam/SRR7657883.sorted.bam.bai*