4 Introduction to Next-Generation Sequencing

4.1 Introduction

The goal of this tutorial is to perform some preliminary tasks in the analysis of RNA-seq data. The data corresponds to zebrafish embryos at two different development stages. You will perform quality control, filter/trim low quality reads, and map the trimmed reads to the reference zebrafish genome assembly.

4.1.1 RNA-seq analysis tools

The tools required for the analysis are listed in Table 1. **You do not necessarily need to download or install anything.** `tools.tar.gz` contains all necessary binaries. If one tool might not work for you when using the binaries, please install the full version.

Tool	URL
FastQC	`http://www.bioinformatics.babraham.ac.uk/projects/fastqc/`
Trimmomatic	`http://www.usadellab.org/cms/?page=trimmomatic`
HISAT2	`http://daehwankimlab.github.io/hisat2/download/`
Samtools	`http://samtools.sourceforge.net/`

Table 1: RNA-seq analysis tools.

These are command-line tools and only run under **Unix-like operating systems**.

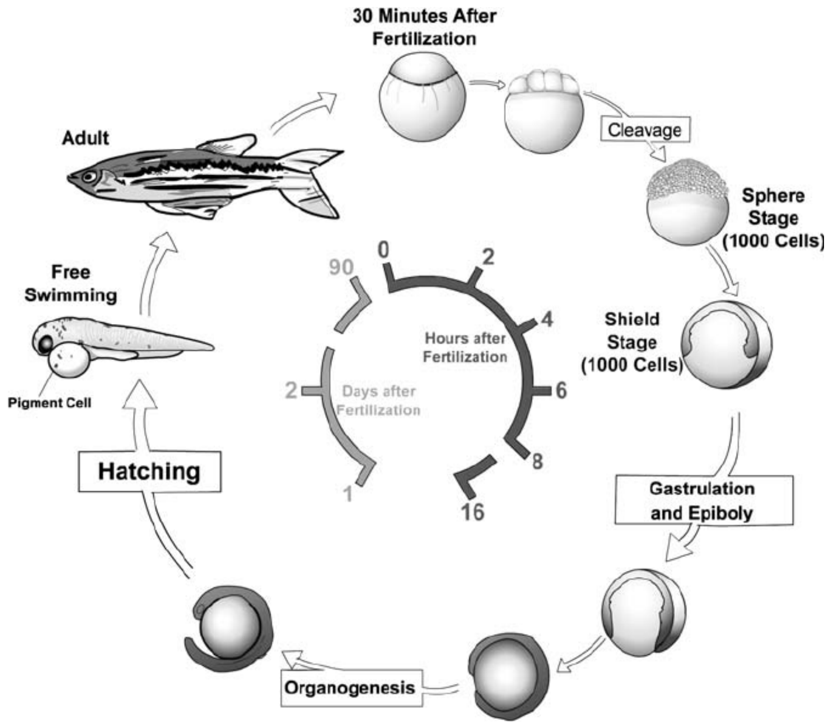
- Extract the files in `tools.tar.gz`:

```
tar -xzvf tools.tar.gz
```

  This will produce a folder called `tools`.

4.2 Dataset

To study the mechanisms regulating early development in zebrafish (*Danio rerio*), the Wellcome Trust Sanger Institute applied RNA-seq and generated comprehensive transcriptome profiles from a range of tissues and developmental stages. Sequencing was performed on the Illumina platform and generated paired-end sequence data using polyA-selected RNA. You will work with a subset of this dataset. The original data can be found at `http://www.ebi.ac.uk/ena/data/view/ERR022484` and `http://www.ebi.ac.uk/ena/data/view/ERR022485`.



Source: *D'Costa and Shepherd, Zebrafish, 2009*

Exercise 1

1. Extract the files in `data.tar.gz`:

   ```
   tar -xzvf data.tar.gz
   ```

   This will create a folder called `data`.

2. Extract the FASTQ files in the `data` folder:

   ```
   tar -xzvf 2cells.tar.gz
   tar -xzvf 6h.tar.gz
   ```

   - `2cells_1.fastq` and `2cells_2.fastq` contain the paired-end RNA-seq sequencing reads of a 2-cell zebrafish embryo.
     - `2cells_1.fastq` has the forward reads; and
     - `2cells_2.fastq` has the reverse reads.
   - `6h_1.fastq` and `6h_2.fastq` contain the paired-end RNA-seq sequencing reads of a zebrafish embryo 6 hours post-fertilization.
     - `6h_1.fastq` has the forward reads; and
     - `6h_2.fastq` has the reverse reads.

3. Use the `head` command to examine the beginning of the FASTQ files, e.g.:

   ```
   head 2cells_1.fastq
   ```

   - If necessary, use the `cd` ("change directory") command to change the current working directory.

4. How many reads are there in each FASTQ file? Use the `wc` ("word count") to count the number of lines in a file:

   ```
   wc -l 2cells_1.fastq
   ```

5. How long are the reads in the FASTQ files?

4.3 Quality control

We will use the tool `FastQC` to verify read quality.

- Unzip `fastqc_v0.11.5.zip` in the `tools` folder:

  ```
  unzip fastqc_v0.11.5.zip
  ```

  This will create a folder called `FastQC`.

- Change to the `FastQC` folder and modify the permissions of the `fastqc` file to make it executable:

  ```
  cd ./tools/FastQC
  chmod +x fastqc
  ```

Exercise 2

FastQC a java program that provides a simple way to do some quality control checks on raw sequencing data. Examine their example reports for good and bad Illumina data.

Run `FastQC` on the forward reads of the 2-cell zebrafish embryo sample by typing the following command:

```
fastqc -o FastQC_output -t 2 --extract --nogroup 2cells_1.fastq
```

- `-o`: the output folder; ensure that output folder exists before running `FastQC`.
- `-t`: the number of threads to run in parallel.
- `--extract`: instructs `FastQC` to extract the compressed output files.

- `--nogroup`: instructs `FastQC` to show output data for each position in the read, instead of grouping neighboring positions (default).

The last argument required by `FastQC` is the FASTQ file.

- Note that you will need to include the complete paths to the tool and input file.

1. Run `FastQC` for **each** FASTQ file in the dataset. After running `FastQC` you should see two output files (a zip file and an html file) and a folder.
   - Use your favorite browser to open the html file.
   - Interpret the plots using with help of the `FastQC` manual (`"FastQC_Manual.pdf"`).

4.4 Trimming

`Trimmomatic` is a java program that can apply various filters to exclude low quality reads, trim low quality bases from the read ends, and remove adapter remnants.

- Unzip `Trimmomatic-0.36.zip` in the `tools` folder:

```
unzip Trimmomatic-0.36.zip
```

  This will create a folder called `Trimmomatic-0.36`.

- To ask for help, type:

```
java -jar trimmomatic-0.36.jar -h
Usage:
 PE [-threads <threads>] [-phred33|-phred64] [-trimlog<trimLogFile>]
[-basein <inputBase> | <inputFile1><inputFile2>]
[-baseout <outputBase> | <outputFile1P><outputFile1U> <outputFile2P> <outputFile2U>]
<trimmer1>...
 or:
 SE [-threads <threads>] [-phred33|-phred64] [-trimlog<trimLogFile>]
 <inputFile> <outputFile> <trimmer1>...
```

The arguments for paired-end reads are the names of the FASTQ files with the forward and reverse reads (`<inputFile1><inputFile2>`), the names of the output files for the paired and remaining unpaired reads after trimming the forward reads (`<outputFile1P><outputFile1U>`) and reverse reads (`<outputFile2P> <outputFile2U>`). In addition, you will have to specify the parameters for the trimming and filtering.

Exercise 3

1. Trim the reads in the FASTQ files corresponding to the 2-cell embryos using:

```
java -jar trimmomatic-0.36.jar PE -threads 2 -phred33
2cells_1.fastq 2cells_2.fastq 2cells_1.trim.fastq 2cells_1.trim.unpaired.fastq
2cells_2.trim.fastq 2cells_2.trim.unpaired.fastq LEADING:20 TRAILING:20 AVGQUAL:20 MINLEN:25
```

   The arguments are:

   - `PE`: reads are paired-end
   - `-threads`: number of threads
   - `-phred33`: quality encoding
   - `LEADING`: quality threshold for removing nucleotides from the 5' end
   - `TRAILING`: quality threshold for removing nucleotides from the 3' end
   - `AVGQUAL`: minimum average read quality (reads with an average quality lower than AVGQUAL will be discarded)
   - `MINLEN`: minimum read length (reads that are shorter than MINLEN after trimming will be discarded)

   - Note that you will need to include the complete paths to the tool and input files.

   - After `Trimmomatic` has run, you should see a report indicating the number of reads that passed the filters. How many read pairs did?

2. How many reads are there in the trimmed FASTQ files?

   - Verify that the trimmed FASTQ files for the forward and reverse reads contain the same number of reads.

3. Examine the header of the first read in the trimmed FASTQ files for the forward and reverse reads. The FASTQ files are sorted. Therefore, these will be the forward and reverse reads of a specific fragment, i.e., these two reads are paired. Note that the two reads in a pair have almost the same headers. What is the difference?

4. Count the number of the reads that have lost the other read of the pair during the trimming and filtering process. Unpaired reads will be ignored for the rest of the analysis. However, in principle, they could be used by treating them as single-end reads.

5. Run `FastQC` on the trimmed FASTQ files and compare the output reports against those for the *unprocessed* FASTQ files.

6. Repeat the same steps for the 6h-zebrafish embryos FASTQ files.

4.5 Genome Indexing

To efficiently map the reads to the reference zebrafish genome you first have to construct its GFM (Graph Ferragina Manzini)-index. It is based on an extension of Burrows-Wheeler transformation (BWT) for graphs.

You will map the RNA-seq reads using `HISAT2`. `HISAT2` is the follow-up tool of `TOPHAT2` which was one of the first tools specially designed to handle splice junctions, which is necessary for mapping RNA-seq reads to a reference genome.

- Unzip `hisat2-2.2.1-Linux_x86_64.zip` in the `tools` folder:

  ```
  unzip hisat2-2.2.1-Linux_x86_64.zip
  ```

  This will create a folder called `hisat2-2.2.1`.

- You can construct the GFM-index of the reference genome using the `hisat2-build` tool. Change to the folder containing the file `hisat2-build` and change its permissions to make it executable:

  ```
  cd ./tools/hisat2-2.2.1
  chmod +x hisat2-build
  ```

  `hisat2-build` requires several arguments:

  ```
  ./hisat2-build --help
  ```

  The general usage is:

  ```
  Usage: hisat2-build [options]* <reference_in> <ht2_index_base>
  ```

  – `reference_in`: the file containing the reference assembly, in FASTA format.

  – `ht2_index_base`: a prefix to name the resulting index files.

The zebrafish genome can be downloaded from the UCSC Genome Browser (`http://hgdownload.cse.ucsc.edu/goldenPath/danRer10/bigZips/danRer10.fa.gz`). Nevertheless, you will find a FASTA-formatted file containing chromosome 12 together with the FASTQ files (`danRer10.chr12.fa.gz`).

Exercise 4

1. Extract the reference genome in the `data` folder:

   ```
   gunzip danRer10.chr12.fa.gz
   ```

2. Now, index it:

   ```
   ./hisat2-build danRer10.chr12.fa danRer10.chr12
   ```

   - Note that you will need to include the complete path to the tool and input file.

   The arguments are:

- `danRer10.chr12.fa`: reference assembly in FASTA file
- `danRer10.chr12`: the prefix for all output files written by `hisat2-build`

3. Once `hisat2-build` is finished, look at the output files. The files having extension `.ht2` and prefix `danRer10.chr12` contain the index of the reference assembly (in this case, only of chromosome 12) in a format that `HISAT2` can use.

4.6 Read mapping using `hisat2`

- Change the permissions of the file `"hisat2"` in `"hisat2-2.2.1"` to make it executable:

```
cd complete_path_to_tools/tools/hisat2-2.2.1/
chmod +x hisat2
```

The general format of the `hisat2` command is:

```
./hisat2 --help
./hisat2 [options]* -q -x <index_base> -1 <reads_1> -2 <reads_2> -S <output.sam>
```

-q indicates we will be working with FASTQ files -x indicates the indexed genome -1 and -2 are the FASTQ files with the forward and reverse trimmed reads. -S defines the output file, in this case always in .sam format `index_base` is the prefix of the indexed genome (here, `"danRer10.chr12"`).

Exercise 5

1. Map the trimmed reads in the 2-cell zebrafish embryo sequencing library:

   ```
   ./hisat2 -q -x danRer10.chr12 -1 2cells_1.trim.fastq -2 2cells_2.trim.fastq -S 2cells.sam
   ```

   The alignment summary will be directly printed to STDOUT. How many reads aligned concordantly exactly one time?

   - Note that you will need to include the complete paths to the tool and input files.

   This is going to be the longest step of the pipeline, requiring a few minutes.

2. While the job is running, examine the files that `HISAT2` is writing.
3. Map the remaining libraries.

4.6.1 Examining the Read Mappings with Samtools

When the mapping is finished, you find the most important statistics directly at the STDOUT.

To convert a SAM-formatted file into a BAM-formatted file you can use `SAMtools`.

- From the `tools` folder:

  ```
  tar xvjf samtools-bcftools-htslib-1.0_x64-linux.tar.bz2
  ```

- From the resulting `samtools-bcftools-htslib-1.0_x64-linux` folder:

  ```
  chmod u+x bin/samtools
  ```

Exercise 6

1. Convert the SAM to BAM files:

   ```
   samtools view -bS 2cells.sam > 2cells.bam
   ```

2. Examine the contents of the BAM files:

   ```
   samtools view 2cells.bam | head
   ```

3. Examine the BAM files in the web version of Integrative Genomics Viewer (IGV) https://igv.org/app/, therefore the BAM file needs to be sorted and indexed:

```
samtools sort -o 2cells_sorted.bam 2cells.bam
samtools index 2cells_sorted.bam
```

What do you see in IGV?

Exercise 7

Creating a Bash script will allow you to define a series of actions which the computer will then perform without us having to enter the commands yourself.

1. Open an empty plain-text file and write:

```
#! /usr/bin/env bash
```

2. Copy all your commands (not the ones for extracting the tools!).
3. Save the file as `my_rnaseq_script.sh` and make it executable.