Quantification and differential expression analysis

Table of contents

Expected learning outcome

To understand the basics of RNA-Seq data, how to use RNA-Seq for different objectives and to familiarize yourself with some standard software packages for such analysis.

Getting started

The main goal of this activity is to go through a standard method to obtain gene expression values and perform differential gene expression analysis from an RNA-Seq experiment.

We will start performing gene quantification using the RSEM program and do differential gene expression analysis of the estimated counts using DESeq2. Ater that we do whole genome alignment and visualize the data using the TopHat2 spliced aligner.

All the data you need for this activity should be in the transcriptomics directory. Here we will use the *genome.quantification* and the *fastq.quantification* subdirectories.

DATASETS AND SOFTWARE

We will be using the following bioinformatics tools:

1. RSEM version v1.2.12 (http://www.biomedcentral.com/1471-2105/12/323)
2. Picard 1.2.7 (http://broadinstitute.github.io/picard/)
3. Bowtie2-2.1.0 (http://www.nature.com/nmeth/journal/v9/n4/full/nmeth.1923.html)
4. samtools 0.1.19 (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2723002/)
5. RStudio 0.98 (http://www.rstudio.org/)
6. IGV 2.3.40 (http://bib.oxfordjournals.org/content/14/2/178.abstract)
7. igvtools 2.3.40 (http://bib.oxfordjournals.org/content/14/2/178.abstract)

Data should be available in two directories called fastq.quantification (12 FASTQ files) and genome.quantification (files related to the mouse genome).

fastq.quantification (short read data):

- control_rep1.1.fq
- control_rep1.2.fq
- control_rep2.1.fq
- control_rep2.2.fq
- control_rep3.1.fq
- control_rep3.2.fq
- exper_rep1.1.fq
- exper_rep1.2.fq
- exper_rep2.1.fq
- exper_rep2.2.fq
- exper_rep3.1.fq
- exper_rep3.2.fq

and

genome.quantification:

- ucsc.gtf (reduced annotation file)
- ucsc_into_genesymbol.rsem (isoform ID to official gene symbol translation)
- mm10.fa (FASTA file of the reduced genome)

INTRODUCTION

Sample pooling has revolutionized sequencing. It is now possible to sequence tens of samples together. Different objectives require different sequencing depths. Doing differential gene expression analysis requires less sequencing depth than transcript reconstruction so when pooling samples it is critical to keep the objective of the experiment in mind. In this activity we will use subsets of experimentally generated datasets. The dataset used here was generated for differential gene expression analysis while the other dataset used for the optional exercise was generated for transcript annotation. For quantification we will use a set of data generated from the same strain as the genome reference mouse (C57BL/6J). The study concentrated on the effect of liver JNK signaling pathway in fatty acid metabolism (see Vernia et al, Cell Metabolism, 2014), the study generated triplicate libraries for 4 different genotypes (wildtype, Jnk1 KO, Jnk2 KO, and Jnk1, Jnk2 KO) in two different diets.

We selected three replicates from wild type (control) and three from the double knock-out strain under fat diet (exper), the two genotypes and condition having largest expression differences. We further created a reduced subset of the data that includes the most striking result of the paper. The idea is to find genes that are in the same pathway as the gene that were knock out. We will use a reduced genome consisting of the first 9.5 million bases of mouse chromosome 16 and the first 50.5

million bases of chromosome 7 for this workshop since, mapping and quantifying the reads on whole genome takes more time. We wanted to reduce the running time.

Exercise 1: Data inspection, prepare for genomic alignment

RSEM depends on an existing annotation and will only score transcripts that are present in the given annotation file.

The first step is to prepare the transcript set that we will quantify. We selected the UCSC genes which is a very comprehensive, albeit a bit noisy dataset. As with all the data in this activity we will only use the subset of the genes that map to the genome regions we are using.

We will be working in the transcriptomics directory:

```
cd ~/workshop_data/transcriptomics/genome.quantification
```

RSEM provides a program to generate indices for transcriptome alignment which is a one time process for each transcriptome. To generate our indices use

```
rsem-prepare-reference --gtf ucsc.gtf --transcript-to-gene-map
ucsc_into_genesymbol.rsem --bowtie2 mm10.fa mm10.rsem
```

Which should extract transcript sequences and generate bowtie2 index files against these sequence. You can check that the program ran successfully by ensuring that the following files were created:

```
mm10.rsem.1.bt2

mm10.rsem.2.bt2

mm10.rsem.3.bt2

mm10.rsem.4.bt2

mm10.rsem.chrlist

mm10.rsem.grp

mm10.rsem.idx.fa

mm10.rsem.rev.1.bt2

mm10.rsem.rev.2.bt2
```

```
mm10.rsem.seq

mm10.rsem.ti

mm10.rsem.transcripts.fa
```

Please note that the GTF file format is not fully standardized so many times you need to modify the gene_id and transcript_id labels. Especially, when you downlaod this file from UCSC, trancript_id and gene_id's are the same and usually they are transcript_ids. So, if you want to quantify gene and isoforms, you need to replace transcript_id's in gene_id field with official gene symbol. This has already been taken care of in the ucsc.gtf file for this workshop.

Question: What do you think each file that is created with rsem-prepare-reference is?

Exercise 2: Quantify with the RSEM program

2.1 Calculate expression

We will assume that your current directory is transcriptomics and that you have four subdirectories within:

```
bin

data

fastq.quantification

genome.quantification
```

We also have fastq.reconstruction and genome.reconstruction directories but we are not going to use them in this part. You can use them in optional part about transcript reconstruction, if you have time after you finish this part.

We'll add one more subdirectory to hold the quantification results:

```
cd ~/workshop_data/transcriptomics

mkdir rsem
```

We are now ready to align and then attempt to perform read assignment and counting for each isoform in the file provided above. You must process each one of the 6 libraries: (Here "\" end of each line denotes that the command will continue in the next line to increase the readibility of the commands, sometimes we are writing one line command in multiple lines. You can remove all "\" and run the command in one line, if you prefer.)

```
rsem-calculate-expression --paired-end -p 2 --bowtie2 \
 --output-genome-bam fastq.quantification/control_rep1.1.fq \
fastq.quantification/control_rep1.2.fq genome.quantification/mm10.rsem \
rsem/ctrl1.rsem
```

And similarly for each of the other 5 libraries

```
rsem-calculate-expression –paired-end -p 2 –bowtie2 \ –output-genome-bam
fastq.quantification/control_rep2.1.fq \
fastq.quantification/control_rep2.2.fq genome.quantification/mm10.rsem \
rsem/ctrl2.rsem
rsem-calculate-expression --paired-end -p 2 --bowtie2 \
--output-genome-bam fastq.quantification/control_rep3.1.fq \
fastq.quantification/control_rep3.2.fq genome.quantification/mm10.rsem \
rsem/ctrl3.rsem
rsem-calculate-expression --paired-end -p 2 --bowtie2 \
--output-genome-bam fastq.quantification/exper_rep1.1.fq \
fastq.quantification/exper_rep1.2.fq genome.quantification/mm10.rsem \
rsem/expr1.rsem
rsem-calculate-expression --paired-end -p 2 --bowtie2 \
--output-genome-bam fastq.quantification/exper_rep2.1.fq \
fastq.quantification/exper_rep2.2.fq genome.quantification/mm10.rsem \
rsem/expr2.rsem
```

```
rsem-calculate-expression --paired-end -p 2 --bowtie2 \

--output-genome-bam fastq.quantification/exper_rep3.1.fq \

fastq.quantification/exper_rep3.2.fq genome.quantification/mm10.rsem \

rsem/expr3.rsem
```

To ensure that you have run all commands successfully, you should check that your rsem directory contains the following result files:

```
ctrl1.rsem.genes.results

ctrl1.rsem.genome.bam

ctrl1.rsem.genome.sorted.bam

ctrl1.rsem.genome.sorted.bam.bai

ctrl1.rsem.isoforms.results

ctrl1.rsem.stat

ctrl1.rsem.transcript.bam

ctrl1.rsem.transcript.sorted.bam

ctrl1.rsem.transcript.sorted.bam.bai

ctrl2.rsem.genes.results

ctrl2.rsem.genome.bam

ctrl2.rsem.genome.sorted.bam

ctrl2.rsem.genome.sorted.bam.bai

ctrl2.rsem.isoforms.results

ctrl2.rsem.stat

ctrl2.rsem.transcript.bam
```

```
ctrl2.rsem.transcript.sorted.bam

ctrl2.rsem.transcript.sorted.bam.bai

ctrl3.rsem.genes.results

ctrl3.rsem.genome.bam

ctrl3.rsem.genome.sorted.bam

ctrl3.rsem.genome.sorted.bam.bai

ctrl3.rsem.isoforms.results

ctrl3.rsem.stat

ctrl3.rsem.transcript.bam

ctrl3.rsem.transcript.sorted.bam

ctrl3.rsem.transcript.sorted.bam.bai

expr1.rsem.genes.results

expr1.rsem.genome.bam

expr1.rsem.genome.sorted.bam

expr1.rsem.genome.sorted.bam.bai

expr1.rsem.isoforms.results

expr1.rsem.stat

expr1.rsem.transcript.bam

expr1.rsem.transcript.sorted.bam

expr1.rsem.transcript.sorted.bam.bai

expr2.rsem.genes.results

expr2.rsem.genome.bam
```

```
expr2.rsem.genome.sorted.bam

expr2.rsem.genome.sorted.bam.bai

expr2.rsem.isoforms.results

expr2.rsem.stat

expr2.rsem.transcript.bam

expr2.rsem.transcript.sorted.bam

expr2.rsem.transcript.sorted.bam.bai

expr3.rsem.genes.results

expr3.rsem.genome.bam

expr3.rsem.genome.sorted.bam

expr3.rsem.genome.sorted.bam.bai

expr3.rsem.isoforms.results

expr3.rsem.stat

expr3.rsem.transcript.bam

expr3.rsem.transcript.sorted.bam

expr3.rsem.transcript.sorted.bam.bai
```

2.3 Create consolidated table

In the bin directory we provide a simple script to take all the independent RSEM output and combine it into a single table, which is needed for inspection and ready for differential gene expression analysis.

To find out what the script does you may type the following command in the transcriptomics directory

```
cd ~/workshop_data/transcriptomics

perl bin/rsem.to.table.pl -help
```

To quit the help page, just press q.

We will generate two tables using two measures for gene expression level. These are expected_count and tpm values. ‘expected count’ is the number of expected RNA-seq fragments assigned to the transcript given maximum-likelihood transcript abundance estimates. ‘tpm’ is the number of transcripts per million.
We are not going to use tpm values in this workshop but, using rsem.to.table.pl script you can create this table to use these normalized counts in your further analysis.

```
perl bin/rsem.to.table.pl -out rsem.gene.summary.count.txt -indir rsem -
gene_iso genes -quantType expected_count

perl bin/rsem.to.table.pl -out rsem.gene.summary.tpm.txt -indir rsem -
gene_iso genes -quantType tpm
```

Question: Can you find genes that look affected by the experiment?

Hint: To be able to look the data, you can use the following command;

```
column -t rsem/expr1.rsem.genes.results | less -S
```

2.4 Visualize the raw data: Make a IGV genome with the transcriptome

We need to create an artificial genome composed of the transcripts we used to annotate. Launch your IGV browser from the terminal in x2goclient:

```
igv.sh
```

Then, from the top menu, select

genome -> create .genome file …

A pop up box should ask for information on the genome. Give the genome a name such as for example mm10ReducedTranscriptome. The name is important as it will be used later, if you choose to name this genome differently take note and make sure you keep using that name in what follows. We will refer to this reduced transcriptopme as “mm10ReducedTranscriptome” and use the same for description.

Click on *browse* and navigate to your *genome.quantification* directory and select the *mm10.rsem.transcripts.fa* file and save the transcripts genome file in the same genome.quantification directory.

This also offers a good example of how to use IGV when working with an assembly or non-published genome sequence. In this cases, you create a genome file in the same way we created the transcript genome.

Open a new terminal to create tdf files.

IGV is highly versatile and can display a large number of file formats. In this exercise we will try two formats, sorted and indexed bam file and a TDF file. the tdf is a reduced/lighter version of the bam that only contains coverage information.
The TDF file will be generated from the bam file and will be faster to work with in IGV since the converage is precalculated. The bam file is calculates the coverage interactively, but has the advantage of displaying the individual reads and sequencing errors and SNPs.
To compare these two file formats, you can either create a TDF file or open bam file. To create tdf file:

```
igvtools count -w 5
rsem/ctrl1.rsem.transcript.sorted.bam  rsem/ctrl1.rsem.transcript.sorted.b
am.tdf \

genome.quantification/mm10.rsem.transcripts.fa
```

Go back to IGV and choose the mm10ReducedTranscriptome genome in the drop down menu at the top left corner. Open the bam file (rsem/ctrl1.rsem.transcript.sorted.bam) and tdf file you just generated.

Have a closer look at one of the genes called fgf21 by typing its ucsc_id (uc009gwe.1) into the search window on IGV and press go. Explore the different bam files you generated and look at the gene expression of the double knockout mice (expr files) versus the wildtype (ctrl files).

Can you see any difference in the expression patterns?

We will come back to the results of exercise 1 later where we will compare the results from RSEM and from Tophat (exercise 4).
We now have the files needed to identify differentially expressed genes.

Exercise 3: Differential gene expression analysis with DESeq

If you reach to this level, you can either skip to excersice 4 or wait for the rest of the class.
You can upload the count matrix to DEBrowser to do downstream analysis.

The DBrowser documantaion and installation instructions are accessible in the linke below;

https://debrowser.readthedocs.io/en/master/

Exercise 4: Genome alignment of RNA-seq reads

RSEM depends on an existing annotation and will only scores transcripts that are present in the given annotation file.
If we want to do de-novo RNASeq quantification. We cannot use RSEM. We need to map the reads to whole genome. In this way, the reads mapped to undefined regions in annotation file can be quantified.

We will compare the alignments produced by RSEM (exercise 2) and tophat (this exercise) and the difference will become clear.

The *fastq.quantification* folder contains a relative small set of illumina sequencing reads. We will examine this set by first directly mapping to the reduced mouse genome using tophat.

Make sure you are in the transcriptomics directory for this activity. *genome.quantification*, *genome.reconstruction*, *fastq.quantification* and *fastq.reconstruction* should be subdirectories. Check this before you proceed.

To avoid cluttering the workspace we will direct the output of each exercise to its own directory. In this case for example:

```
cd ~/workshop_data/transcriptomics

mkdir tophat
```

First build the index files using bowtie2 (for details of indexing, see exercise 1).

```
bowtie2-build genome.quantification/mm10.fa genome.quantification/mm10
```

The following file should be generated in genome.quantification directory:

```
mm10.1.bt2

mm10.2.bt2

mm10.3.bt2

mm10.4.bt2

mm10.rev.1.bt2

mm10.rev.2.bt2
```

Then align each of the libraries to the genome. The *fastq.quantification* subdirectory contains six different libraries, three for a control experiment from wild type mouse liver and from mouse that are deficient in two different proteins. Each genotype was sequenced in triplicates using paired-end 50 base paired reads.

To first explore the data visually in IGV, we'll use the TopHat2 aligner to map these reads to our reduced genome:

```
tophat2 --library-type fr-firststrand --segment-length 20 \
```

```
-G genome.quantification/ucsc.gtf -o tophat/th.quant.ctrl1 \

genome.quantification/mm10 fastq.quantification/control_rep1.1.fq \

fastq.quantification/control_rep1.2.fq
```

Here the parameters we used in tophat explained below;
–library-type: The default is unstranded (fr-unstranded). If either fr-firststrand or fr-secondstrand is specified.
–segment-length: Each read is cut up into segments, each at least this long. These segments are mapped independently. The default is 25.
-G: Supply TopHat with a set of gene model annotations and/or known transcripts, as a GTF 2.2 or GFF3 formatted file.
-o: output directory
Please visit tophat manual to learn the rest of the parameters using the following link; http://ccb.jhu.edu/software/tophat/manual.shtml

And using this command as a template, align the other libraries

Tophat always reports its alignment in a file named “accepted_hits.bam”. To make things clear we’ll move these files onto a clean directory. Move the files by, for example, doing

```
cd ~/workshop_data/transcriptomics

mv tophat/th.quant.ctrl1/accepted_hits.bam tophat/th.quant.ctrl1.bam

mv tophat/th.quant.ctrl2/accepted_hits.bam tophat/th.quant.ctrl2.bam

mv tophat/th.quant.ctrl3/accepted_hits.bam tophat/th.quant.ctrl3.bam

mv tophat/th.quant.expr1/accepted_hits.bam tophat/th.quant.expr1.bam

mv tophat/th.quant.expr2/accepted_hits.bam tophat/th.quant.expr2.bam

mv tophat/th.quant.expr3/accepted_hits.bam tophat/th.quant.expr3.bam
```

To visualize the alignments we generate indexes (for rapid data access) and compressed read density plots:

```
picard-tools BuildBamIndex I=tophat/th.quant.ctrl1.bam
O=tophat/th.quant.ctrl1.bam.bai
```

and with all the other libraries:

Finally create read density files to be able to look at all libraries together

```
igvtools count -w 5 tophat/th.quant.ctrl1.bam
tophat/th.quant.ctrl1.bam.tdf genome.quantification/mm10.fa
```

NOTE: You need to refer to the genome file (the last argument above) using the same name you

Ideally, being mouse data we would use the mouse (mm10) genome provided by IGV. Data would be concentrated to the regions of chromosomes 7 and 16 we aligned to. However, the mm10 genome version may not be available from the install directory.
In this case it is better to make a new “genome” instead of downloading the full mm10 genome. This exercise is a little bit different than the one in exercise 2. In exercise 2 we visualized only transcripts. In that case, the introns were spliced out and we could visualize only exonic regions.
However, since tophat aligns whole genome we see intronic and intergenic alignments too.

To create the reduced genome for IGV:

Launch your IGV browser from the terminal in x2goclient. (Tip: Run igv.sh).

In the top menu select

genome –> create .genome file

You may enter any name and description of your liking, but you need to use this name in the next set of instructions. We will name it mm10.reduced

For the FASTA file use the genome.quantification/mm10.fa file

and for the Gene file use the genome.quantification/ucsc.gtf file

Please load some of the files you prepared. (ie: tophat/th.quant.expr1.baman d tophat/ctrl1.quant.expr1.bam or tophat/th.quant.expr1.bam.tdf and tophat/ctrl1.quant.expr1.bam.tdf)

Some of the genes are good examples of differentially expressed genes. For example the whole region around the key *Fgf21* gene is upregulated in experiment vs controls, while the gene *Crebbp* is downregulated in experiments vs controls. To point your browser to either gene just type or copy the name of the gene in the location box at the top.

You can also compare the results with rsem genomic alignments. Make sure you are loading genome alignments not transcript in rsem results, since both alignment results exist in the rsem directory.
If you want to compare tdf files, you can use the command below.

```
igvtools count -w 5 rsem/ctrl1.rsem.genome.sorted.bam
rsem/ctrl1.rsem.genome.sorted.bam.tdf genome.quantification/mm10.fa

igvtools count -w 5 rsem/expr1.rsem.genome.sorted.bam
rsem/expr1.rsem.genome.sorted.bam.tdf genome.quantification/mm10.fa
```

Question: What is different in tophat and rsem alignments?