RNA-Seq Exercise

Table of Contents

1. Introduction

2. Alignment

3. Isoform expression and transcriptome assembly

4. Differential Expression

5. Functional Annotation of Differentially Expressed genes

1. Introduction

The goal of this hands-on session is to perform some basic tasks in the downstream analysis of RNA-seq data. We will start from RNA-seq data aligned to the zebrafish genome using Tophat. We will perform transcriptome reconstruction using Cufflinks and we will compare the gene expression between two different conditions in order to identify differentially expressed genes.

We will use a dataset derived from sequencing of mRNA from Danio rerio embryos in two different developmental stages. Sequencing was performed on the Illumina platform and generated 76bp paired-end sequence data using polyA selected RNA. Due to the time constraints of the practical we will only use a subset of the reads.

First, log in to the server. Create a folder "RNA-seq" and enter it

*$ mkdir RNA-seq*

*$ cd RNA-seq*

*$ ls*

Create directories with symbolic links to the folders that you want to use

*$ ln -s /data7/RNA-seq/data data*

*$ ln -s /data7/RNA-seq/genome genome*

*$ ln -s /data7/RNA-seq/annotation annotation*

*$ ln -s /data7/RNA-seq/cufflinks cufflinks*

*$ ln -s /data7/RNA-seq/cuffdiffcuffdiff*

*$ ln -s /data7/RNA-seq/tophat tophat*

*$ ls*

The data files are contained in the subdirectory called data and are the following:

• 2cells\_1.fastq and 2cells\_2.fastq: these files are based on RNA-seq data of a 2-cell zebrafish embryo, and

• 6h\_1.fastq and 6h\_2.fastq : these files are based on RNA-seq data of zebrafish embryos 6h post fertilization.

Check that the data folder contains the above-mentioned files by typing:

*$ ls data*

Note that all commands that are given in this tutorial should be run within the main folder RNA-seq.

$ head -n 8 data/2cells\_1.fastq

2. Alignment

There are numerous tools performing short read alignment and the choice of aligner should be carefully made according to the analysis goals/requirements. Here we will use Tophat, a widely used ultrafast aligner that performs spliced alignments. Tophat is based on Bowtie to perform alignments and uses an indexed genome for the alignment to keep its memory footprint small. For time purposes we have already generated the index for the zebrafish genome and placed it under the **genome subdirectory**.

*$ ls genome*

Tophat has a number of parameters in order to perform the alignment. To view them all type

*$ tophat --help*

**(Skip it)** In the terminal type:

*tophat --bowtie1*

 *--solexa-quals \*

 *-g 2 \*

 *--library-type fr-unstranded \*

 *-j annotation/Danio\_rerio.Zv9.66.spliceSites \*

 *-o <output dir> \*

 *genome/ZV9 \ (your genome)*

*<fastq file – left mate> \ (e.g. data/2cells\_1.fastq)*

*<fastq file – right mate> (e.g. data/2cells\_2.fastq)*

**Due to time constraints the alignment has already been performed for you and the results are stored under the tophat subdirectory.**

If you look in this subdirectory you will find another two folders one for each sample result, called ZV9\_2cells and ZV\_6h.

*$ ls tophat*

The alignment results are stored in the file tophat/<output dir>/accepted\_hits.bam in BAM (compressed binary version of the SAM format that stands for Sequence Alignment/Map). For more information regarding the SAM format please see: <http://samtools.sourceforge.net/SAM1.pdf>.

Tophat reports the alignments in a BAM file called accepted\_hits.bam. Among others it also creates a **junctions.bed** files that stores the coordinates of the splice junctions present in your dataset, as these have been extracted from the spliced alignments.

Now we will load the BAM file and the splice junctions onto IGV to visualize the alignments reported by Tophat.

We already know that in order to load a BAM file onto IGV we need to have this file sorted by genomic location and indexed. Here’s a reminder of the commands to perform these:

Sort the BAM file using samtools:

samtools sort [bam file to be sorted] [prefix of sorted bam output file]

Index the sorted file.

samtools index [sorted bam file]

**These commands have already been performed on the BAM files.**

*$ ls tophat/ZV9\_2cells*

Launch IGV. When it opens you have to load the genome of interest. On the top left of your screen choose from the drop down menu Zebrafish (Zv9). Then in order to load the desire files go to:File > Load from File. On the pop up window navigate to tophat>ZV9\_2cells folder and select the file **accepted\_hits.sorted.bam**. Once the file is loaded right-click on the name of the track on the left and choose Rename Track. Give the track a meaningful name. Follow the same steps in order to load the junctions.bed file from the same folder. Repeat it against to tophat>ZV9\_6h. Finally following the same process load the Ensembl annotation **Danio\_rerio.Zv9.66.gtf** stored under annotation folder. On the top middle box you can specify the region you want your browser to zoom. Type chr12:20,270,921-20,300,943. For more information please see: http://www.broadinstitute.org/igv/AlignmentData

Question:

Can you identify the splice junctions from the BAM file?

Are the junctions annotated for CBY1 consistent with the annotation?

3. Isoform expression and transcriptome assembly

There are a number of tools that perform reconstruction of the transcriptome and for this practice we are going to use Cufflinks. Cufflinks can do transcriptome assembly either de-novo or using a reference annotation.

Cufflinks has a number of parameters in order to perform transcriptome assembly and quantification. To view them all type

*$ cufflinks --help*

It is common to reconstruct the transcriptome for both samples by using the **Ensembl annotation** both **strictly** and **as a guide.** In the first case Cufflinks will only report isoforms that are included in the Ensembl annotation, while in the latter case it will report **novel** isoforms as well. In today’s class, we chose the second one as practice. The annotation from Ensembl of Danio rerio is stored under the folder annotation in a file called Danio\_rerio.Zv9.66.gtf.

Performing the guided transcriptome analysis for the 2cells and 6h data sets would take 15-20min each. Therefore, we have pre-computed these for you and have the results under subdirectories:

cufflinks/ZV9\_2cells and cufflinks/ZV9\_6h.

**(Skip it)** In the terminal type:

*$ cufflinks -o cufflinks/ZV9\_2cells\_gff \ (output dir)*

*-g annotation/Danio\_rerio.Zv9.66.gtf \ (ref annotation)*

*-b genome/Danio\_rerio.Zv9.66.dna.fa \ (ref genome)*

*-u \*

*--library-type fr-unstranded \*

*tophat/ZV9\_2cells/accepted\_hits.bam (your mapped bam file)*

Take a look at the output folders (cufflinks/ZV9\_2cells and cufflinks/ZV9\_6h) that have been created. The results from Cufflinks are stored in the files named: **genes.fpkm\_tracking**, **isoforms.fpkm\_tracking**, and **transcripts.gtf**. The complete documentation can be found at:

[http://cole-trapnell-lab.github.io/cufflinks/cufflinks/index.html#cufflinks-output-files](http://cole-trapnell-lab.github.io/cufflinks/cufflinks/index.html%2523cufflinks-output-files)

Go back to the IGV browser and load the file **transcripts.gtf** which is located in the subdirectory cufflinks/ZV9\_2cells/ and cufflinks/ZV9\_6h/. These files contains the transcripts that Cufflinks assembled based on the alignment of our reads onto the genome.

Question:

In the search box type ENSDART00000082297 in order for the browser to zoom in to the gene of interest. Compare between the Ensembl annotation transcripts and the ones assembled by Cufflinks. Do you observe any difference?

4. Differential Expression

One of the stand-alone tools that perform differential expression analysis is Cuffdiff. We use this tool to compare between two conditions; for example different conditions could be control and disease, or wild-type and mutant, or various developmental stages. In our case we want to identify genes that are differentially expressed between two developmental stages; a 2 cell embryo and 6h post fertilization.

*$ cuffdiff --help*

*$ cuffdiff -o cuffdiff\_new/ \*

*-L ZV9\_2cells,ZV9\_6h \*

*-T \*

*-b genome/Danio\_rerio.Zv9.66.dna.fa \*

*-u \*

*--library-type fr-unstranded \*

*annotation/Danio\_rerio.Zv9.66.gtf \*

*tophat/ZV9\_2cells/accepted\_hits.bam \*

*tophat/ZV9\_6h/accepted\_hits.bam*

We are interested in the differential expression at the gene level. The results are reported by Cuffdiff in the file cuffdiff/gene\_exp.diff. Look at the first few lines of the file using the following command:

$ head -n 10 cuffdiff\_new/gene\_exp.diff

We would like to see which are the most significantly differentially expressed genes. Therefore we will sort the above file according to the q value (corrected p value for multiple testing). The result will be stored in a different file called gene\_exp\_qval.sorted.diff.

$ sort -t$'\t' -g -k 13 cuffdiff\_new/gene\_exp.diff > cuffdiff\_new/gene\_exp\_qval.sorted.diff

Look again at the first few lines of the sorted file by typing:

$ head -n 20 cuffdiff\_new/gene\_exp\_qval.sorted.diff

Copy the Ensembl identifier of one of these genes. Now go back to the IGV browser and paste it in the search box. Look at the raw aligned data for the two datasets.

Question:

Do you see any difference in the gene coverage between the two conditions that would justify that this gene has been called as differentially expressed?

How to sort the differential expression by log2(fold\_change)?

5. Functional Annotation of Differentially Expressed genes

After you have performed the differential expression analysis you are interested in identifying if there is any functionality enrichment for your differentially expressed genes. We have already performed differential expression analysis genome wide for you and the results are stored within the cuffdiff folder in the file ‘diffExprs\_Genes\_qval.01.txt’. This file contains only significantly differential expressed genes using a cutoff of 0.01 for the qvalue. From this file we extract only the first column, which contains the Ensembl gene identifiers of the differentially expressed genes, and show the top 100 based on their q-value.

$ cut -f 1 cuffdiff/diffExprs\_Genes\_qval.01.txt | head -n 100

Open a web browser and go to the following URL: <http://david.abcc.ncifcrf.gov/> . On the left side click on *Functional Annotation*. Then click on the *Upload* tab. Under the section, paste your gene list to the text area. Under Step 2 select *ENSEMBL\_GENE\_ID* from the drop-down menu. Finally select *Gene list* and then press *Submit List*. Click on Gene Ontology and then click on the CHART button of the *GOTERM\_BP\_FAT* item.

Question:

Do these categories make sense given the samples we’re studying?