RNA-Seq analysis: practical session using Tuxedo suite

The "Tuxedo Suite" is mainly composed of Bowtie, Tophat, Cufflinks, CuffDiff. It has been developed in order to ease read mapping, discovery of splice junction and novel gene structure and differential expression analysis. In the practical session we will use this suite to analyse two samples obtained from study "SRP000698" available in the SRA database

Content

The SRP000698 dataset: Genome-wide analysis of allelic expression imbalance in human primary cells by high-throughput transcriptome resequencing

In this article the authors have used RNA-Seq technology to compare the transcriptome of actived and resting T-Cells. Using this technology they were also able to monitor allele-specific expression (ASE), that is, specific expression arising from maternally and paternally derived alleles. In this tutorial we will mainly concentrate on mapping read to the genome and compute gene expression levels with the Tuxedo suite.

Getting more informations about the experiment

- The SRA Sequence Read Archive (SRA) web site can be accessed here .
- The SRA Run browser (*Tools* section) can be used to search for a SRA object *Search >SRA Objects*

1. Get informations about the SRA study "SRP000698"
2. What is the study about ?
3. What platform was used ?
4. How many reads were produced ?
5. How many samples were analyzed ?
6. Get informations about experiments "SRX011549" and "SRX011550" (SRA Objects)
7. Which of these two samples is untreated or treated ?
8. How many runs were performed per samples ?
9. Is this experiment single-end or paire-end sequencing ? What are the sizes of the reads ?
10. How many reads are available per run on average ? Calculate it roughly.
11. Select one run. What is the sequence of the first read ?
12. what is the quality of this read ?

Obtaining the data

Analysis of the whole dataset would be time consuming and would require access to a computing server. To make the analysis feasible on a desktop computer, data were previously retrieved from SRA, fastq-transformed using SRA toolkit (fastq-dump) and mapped to the human genome (version hg19). A subset of reads that aligned onto chromosome 10 was extracted and will be used for this tutorial.

1. Open a terminal.
2. Change the current working directory to */filer/openspace/DEPOT*.
3. Create a new directory and give it your login as a name.
4. Go into this directory and create directories named *fastq*, *progs*, *index*, *tophat_results annotations*and *cufflinks_results*.
5. Go in the *fastq* directory and retrieve the datasets below.
6. Uncompress the datasets.
7. Look at the first lines of the *SRR027888.SRR027890_chr10_1.fastq* file to check the fastq format.

File name	Experiment	Description
SRR027888.SRR027890_chr10_1.fastq.gz	SRX011549	Right end read
SRR027888.SRR027890_chr10_2.fastq.gz	SRX011549	Left end read

SRR027889.SRR027891_chr10_1.fastq.gz SRX011550 Right end read
SRR027889.SRR027891_chr10_2.fastq.gz SRX011550 Left end read
View solution| Hide solution

Solution

```
## Changing working directory
WORKINGDIR=/filer/openspace/DEPOT/
cd ${WORKINGDIR}

## Creating sub-directories
mkdir -p ${USER}
ls -trl
cd ${USER}
mkdir -p fastq progs index tophat_results annotations cufflinks_results
cd fastq

## Retrieving data
wget ftp://tagc.univ-mrs.fr/public/Tagc/Denis/SRP000698/SRR027888.SRR027890_chr10_1.fastq.gz
wget ftp://tagc.univ-mrs.fr/public/Tagc/Denis/SRP000698/SRR027888.SRR027890_chr10_2.fastq.gz
wget ftp://tagc.univ-mrs.fr/public/Tagc/Denis/SRP000698/SRR027889.SRR027891_chr10_1.fastq.gz
wget ftp://tagc.univ-mrs.fr/public/Tagc/Denis/SRP000698/SRR027889.SRR027891_chr10_2.fastq.gz

## Uncompressing data
gunzip *gz

## Checking fastq file format
less SRR027888.SRR027890_chr10_1.fastq # q to quit
```

Quality control of high throughput sequencing data

FastQC aims to provide a simple way to do some quality control checks on raw sequence data coming from high throughput sequencing pipelines. It provides a modular set of analyses which you can use to give a quick impression of whether your data has any problems of which you should be aware before doing any further analysis.

1. Download the FastQC program and install it inside the progs directory.

```
## Changing working directory to 'progs'
cd ${WORKINGDIR}${USER}"/progs"

## Retrieving the FastQC program
wget http://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.10.1.zip

## Uncompressing the file
unzip fastqc_v0.10.1.zip

## make the fastqc file executable
chmod u+x FastQC/fastqc
```

2. Launch the FastQC program and check quality of *SRR027888.SRR027890_chr10_1.fastq* fastq file.

```
FastQC/fastqc
```

1. Carefully inspect all the statistics. What do you think of the overall quality of the dataset ?

Read trimming

Read trimming is a pre-processing step in which input read ends are cutted (most generally the right end). Here, reads were previously trimmed. However one should keep in mind that this step is crucial when working with bowtie/tophat. Indeed as bowtie

does not perform "hard-clipping" (that is clip sequence NOT present in the reference) it may be unable to align a large fraction of the dataset when poor quality ends are kept. Several software may be used to perform sequence trimming:

.

- FASTX-Toolkit
- sickle
- the ShortRead Bioconductor package

Mapping read with TopHat

TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons.

Indexing the reference

Here we will align reads to the chromosome 10 of the human genome (version hg19). We thus need to index the corresponding sequence to speed up the read mapping process. Sequence for chromosome 10 can be obtained from the ftp site of the UCSC.

```
## indexing the reference
cd ${WORKINGDIR}${USER}"/index/"
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/chromosomes/chr10.fa.gz
gunzip chr10.fa.gz
bowtie-build chr10.fa chr10.hs
```

1. Change directory to the *tophat_results* directory.
2. Create a new directory SRR027888.SRR027890 and go to this directory.
3. Use the `tophat` command to align left and right reads onto chromosome 10. Use default parameters except the following:
    1. Set *--library-type* argument to *fr-unstranded*.
    2. Set *max-multihits parameter* (*-g*) to 1 to eliminate multireads.
    3. Set *SRR027888.SRR027890_chr10_1.fastq* as the first RNA-Seq FASTQ file and *SRR027888.SRR027890_chr10_2.fastq* as the second RNA-Seq FASTQ file.
    4. Set *chr10.hs* as a reference genome.
    5. Set *Mean Inner Distance* (-r) between mate pairs to the correct value based on the library preparation protocol describe below.
    6. Set *Std. Dev for Distance between Mate Pairs* to 20 (you can use in R to visualize fragment length distribution under a normal assumption).
    7. Set the output directory (-o) to the current directory.
    8. Execute.

Illumina library construction

RNA was extracted using the RNeasy kit (QIAGEN, UK) following the manufacturer's instruction. Samples were subjected to additional DNase treatment using Turbo-DNase (Ambion, UK). RNA quantification and quality were assessed using Nanodrop (Nanodrop Technologies, USA) and RNA 6000 Agilent Bioanalyzer chip (Agilent Technologies, USA). A double poly(A) RNA isolation was performed on 10 µg of total RNA (Invitrogen). Poly(A) RNA was fragmented for exactly 5 min at 70°C in fragmentation buffer (Ambion) prior to random hexamer reverse transcription and second strand synthesis as previously described. Illumina GAII PE adapters were ligated to the DNA and the library generated according to the standard library generation protocol (Illumina, USA). A 300 bp size range was excised from the library on 2% agarose gel. The resultant library was subjected to 15 cycles of PCR (Phusion* DNA polymerase, Finnzymes, Finland). The library was quantified by Nanodrop (Nanodrop Technologies) and assayed for size using a DNA 7500 Agilent Bioanalyzer chip (Agilent).

View solution (fragment length distribution under a normal assumption)| Hide solution

Solution

```

## Start R and type
plot(100:300,dnorm(100:300,200,20),ty="l",ylab="density", xlab="Inner fragment
   length")
## quit R
q()
```

View solution (tophat parameters)| Hide solution

Solution

```
## Changing working directory to tophat_results
cd ${WORKINGDIR}${USER}"/tophat_results"

## Creating a directory for storing tophat results for SRR027888/SRR027890 runs.
mkdir SRR027888.SRR027890
cd SRR027888.SRR027890

## Launching tophat
tophat -g 1 -r 200 --mate-std-dev 30 -o ./  ../../index/chr10.hs
   ../../fastq/SRR027888.SRR027890_chr10_1.fastq
   ../../fastq/SRR027888.SRR027890_chr10_2.fastq

## listing the corresponding results
ls
```

Samtools: sorting and indexing the BAM file.

SAM (Sequence Alignment/Map) format is a generic format for storing large nucleotide sequence alignments. The BAM files are compressed version of the SAM files. The samtools program provide various utilities for manipulating alignments in the SAM format, including sorting, merging and indexing (...).

In order to visualize the results in a genome browser, BAM file need to be sorted (according to chromosome and genomic coordinate). We must then index the subsequent file to speed up the queries when inspecting a particular region of the genome.

1. Use `samtools view` to visualize the content of the compressed BAM file (*accepted_hits.bam*).
2. Use `samtools sort` to sort the *accepted_hits.bam* file.
3. Index the sorted bam file.

View solution (tophat parameters)| Hide solution

Solution

```
## Viewing the alignment results
samtools view accepted_hits.bam | less # q to quit

## Sorting and indexing the bam file
samtools sort accepted_hits.bam accepted_hits.sorted
samtools index  accepted_hits.sorted.bam
```

Viewing the results with Integrated Genome Browser (IGV).

The Integrative Genomics Viewer (IGV) is a high-performance visualization tool for interactive exploration of large, integrated genomic datasets. It supports a wide variety of data types, including array-based and next-generation sequence data, and genomic annotations.

1. Create an IGV account here
2. Download IGV and launch it with 750 MB or 1.2 Gb depending of your machine
3. Select hg19 genome and chromosome 10
4. Use *File* > *Load from file* and browse to the *bam* file.
5. Zoom in to visualize reads mapped onto the genome (chr10).
6. Load the junctions.bed, insertions.bed and deletions.bed into IGV (*File* > *Load from file*).

Expression level estimate with cufflinks

Cufflinks perform transcript assembly and FPKM (RPKM) estimates for RNA-Seq data. Cufflinks can try to perform these tasks in several ways:

1. **Without any reference annotation**. Cufflinks will perform assembly based on the sole TopHat results. Infered gene structures and FPKM will be returned.
2. **With a reference as a guide (-g):** Cufflinks use the supplied reference annotation (GTF/GFF) to guide assembly of reference transcripts. Output will include all reference transcripts, novel genes and isoforms and their corresponding FPKM.
3. **with the reference only (-G)**. Cufflinks use the supplied reference annotation (GTF/GFF) to estimate FPKM values for known transcripts. Cufflinks will ignore alignments not structurally compatible with any reference transcript.

Getting annotation (gtf file)

1. GO to UCSC genome browser
2. From the upper menu select *Tables*
3. Select *human* (genome), *hg19* (assembly), *Gene and Genes prediction tracks* (group), *refGene* (Gene), *chr10* (position), GTF (output format), hg19.gtf (output file).
4. Click on *get output*
5. Copy the gtf file into the annotations folder.
   Use `less` to visualise the gtf fil content.

Computing FPKM

Here we will use RefSeq transcripts as references (that is we won't discover novel genes). First, we need a *gtf/gff* file indicating the locations of exonic regions.

Now that we have a GTF file we can ask cufflinks to compute coverage and FPKM based on the input gene structure.

1. Change directory to the *cufflinks_results* directory.
2. Create a new directory SRR027888.SRR027890 and go to this directory.
3. Use the `cufflinks` to compute FPKM of known genes. Use default parameters except the following:
   1. Set *--library-type* argument to *fr-unstranded*.
   2. Set *max-multihits parameter* (*-g*) to 1 to eliminate multireads.
   3. Set the hg19.gtf file as annotation source (-G).
   4. Set the *accepted_hits.bam* BAM file produced by TopHat as BAM input.
   5. Execute.
      Check the results in the isoforms.fpkm_tracking file.

View solution (tophat parameters)| Hide solution

Solution

```
01.
02. ## Changing working directory to cufflinks_results
03. cd ${WORKINGDIR}${USER}"/cufflinks_results"
04. mkdir SRR027888.SRR027890
05. cd SRR027888.SRR027890
06.
07. ## startingcufflinks
08. cufflinks --library-type fr-unstranded -p 10 -G ../../annotations/hg19.gtf
       ../../tophat_results/SRR027888.SRR027890/accepted_hits.bam
09. less genes.fpkm_tracking
```

Now perform the same analysis (read mapping and FPKM computation) for the SRR027889.SRR027891 run.
View solution (tophat parameters)| Hide solution

Solution

```
01.
02.
03. ## Changing working directory to tophat_results
04. cd ${WORKINGDIR}${USER}"/tophat_results"
05.
06. ## Creating a directory for storing tophat results for SRR027889.SRR027891 runs.
07. mkdir SRR027889.SRR027891
08. cd SRR027889.SRR027891
09.
10. ## Launching tophat
11. tophat -g 1 -r 200 --mate-std-dev 30 -o ./  ../../index/chr10.hs
       ../../fastq/SRR027889.SRR027891_chr10_1.fastq
       ../../fastq/SRR027889.SRR027891_chr10_2.fastq
12.
```

```
## Changing working directory to cufflinks_results
cd ${WORKINGDIR}${USER}"/cufflinks_results"
mkdir SRR027889.SRR027891
cd SRR027889.SRR027891

## startingcufflinks
cufflinks --library-type fr-unstranded -p 10 -G ../../annotations/hg19.gtf
   ../../tophat_results/SRR027889.SRR027891/accepted_hits.bam
```

Comparing expression levels in R

- Retrieve transcript id to gene symbol mapping from UCSC

```

cd ${WORKINGDIR}${USER}"/annotations"
wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/refGene.txt.gz
gunzip refGene.txt.gz
cut -f2,13 refGene.txt  | sort | uniq > transcript2Gene.txt
head transcript2Gene.txt
```

- Change directory to cufflinks_results and start R.

```

cd ${WORKINGDIR}${USER}"/cufflinks_results"
R
```

- Use the R code below to compare gene expression levels.

```
## First we read FPKM tracking.
sample_1 <- read.table("SRR027888.SRR027890/isoforms.fpkm_tracking" ,s
   head=T,row=1)
colnames(sample_1)
#fix(sample_1)
sample_2 <- read.table("SRR027889.SRR027891/isoforms.fpkm_tracking" ,sep="\t",
   head=T,row=1)
colnames(sample_2)
#fix(sample_2)


## creating an expression matrix
transcript.name <- union(rownames(sample_1),rownames(sample_2))
exprs.mat <- cbind(sample_1[transcript.name ,]$FPKM,sample_2[transcript.name
   ,]$FPKM)
rownames(exprs.mat) <- transcript.name


## Selecting genes with FPKM above 0 in both sample
aboveZero <- exprs.mat > 2
sum.aboveZero <- apply(aboveZero,1,sum)
exprs.mat <- exprs.mat[sum.aboveZero >= 1, ]

## Values are log2 transformed
##  (a pseudo-count is added in case one of the sample is equal or close to zero)
exprs.mat <- log2(exprs.mat +1)

## Checking distribution of FPKM values
hist(exprs.mat, main="Distribution of FPKM values")
boxplot(exprs.mat,col=c("red","gray"), pch=16)


## Getting gene symbols
transcript2Gene <-
   read.table("../annotations/transcript2Gene.txt",sep="\t",head=F, row=1)


## Scatter plot comparing expression levels in sample 1 and 2
par(xaxs='i',yaxs='i')
```

```
plot(exprs.mat[,1],exprs.mat[,2],pch=20, panel.first=grid(col="darkgray"))
identify(exprs.mat[,1], exprs.mat[,2],lab=transcript2Gene[rownames(exprs.mat),])
```

Discovering novel genes

If you have some time left, use cufflinks using the -g argument to search for unknown gene structures. Using bedtools try to identify novel genes that are at least 10kb away from known any genes.

References

1. **Genome-wide analysis of allelic expression imbalance in human primary cells by high-throughput transcriptome resequencing.** Heap GA, Yang JH, Downes K, Healy BC, Hunt KA, Bockett N, Franke L, Dubois PC, Mein CA, Dobson RJ, Albert TJ, Rodesch MJ, Clayton DG, Todd JA, van Heel DA, Plagnol V. Hum Mol Genet. 2010 Jan 1;19(1):122-34.