Introduction to expression analysis
(RNA-seq)

RNA-seq Hands-on Exercise

Philippine Genome Centre
University of the Philippines

Originally prepared by
Myrto Kostadima, University of Cambridge
Maria Xenophontos, EMBL-EBI
(Revised and re-printed with permission)

Revised for PGC by
Nelzo C. Ereful
National Institute of Agricultural Botany
Cambridge, UK

General information

The following standard icons are used in the hands-on exercises to help you locate:

 Important Information

 General information / notes

 Follow the following steps

 Questions to be answered

 Warning – PLEASE take care and read carefully

 Optional Bonus exercise

 Optional Bonus exercise for a champion

Resources used

Tophat: http://ccb.jhu.edu/software/tophat/index.shtml

Cufflinks: http://cole-trapnell-lab.github.io/cufflinks/

Samtools: http://samtools.sourceforge.net/

BEDTools: https://github.com/arq5x/bedtools2

UCSC tools: http://hgdownload.cse.ucsc.edu/admin/exe/

IGV genome browser: http://www.broadinstitute.org/igv/

DAVID Functional Analysis: http://david.abcc.ncifcrf.gov/

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 by aligning RNA-seq data generated from *C. purpurea* genome using *Tophat*. We will perform transcriptome reconstruction using *Cufflinks* and we will compare the gene expression between dormant (control) and germinating stages of the fungal spore in order to identify differentially expressed genes. Have fun!

Prepare the environment

We will use the **trimmed (quality-base) dataset** you worked on this morning.

Open the Terminal.

First, go to the folder, where you saved the trimmed data.

```
cd ~
```

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

```
ls -l data
```

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. We have already seen how to index the genome (Lecture), therefore for time purposes we have already generated the index for the *Claviceps purpurea* genome and placed it under the `genome` subdirectory of **`/nfs/projects/training/RNASeq/`**.

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

```
tophat --help
```

The general format of the tophat command is (for paired-end):

```
tophat [options]* <index_base> <reads_1> <reads_2>
```

Where the last two arguments are the fastq files of the paired-end reads, and the argument before is the basename of the indexed genome. In our case, we will use **singe-end** reads.

Can you tell which quality encoding our fastq formatted reads are in?
Hint: Look at the first few reads of the raw (untrimmed) file `Claviceps.fastq` by typing:

```
head -n 20 Claviceps.fastq
```

And then compare the quality strings with the table found at: http://en.wikipedia.org/wiki/FASTQ_format#Encoding

Some other parameters that we are going to use to run *Tophat* are listed below:

- **`-g`**: maximum number of multihits allowed. Short reads are likely to map to more than one location in the genome even though these reads can have originated from only one of these regions. In RNA-seq we allow for a restricted number of multihits, and in this case we ask *Tophat* to report only reads that map at most onto 2 different loci
- **`-p:`** use these many threads to align reads (default is 1)
- **`--library-type`**: before performing any type of RNA-seq analysis you need to know a few things about the library preparation. Was it done using a strand-specific protocol or not? If yes, which strand? In our data the protocol was NOT strand specific
- **`-o`**: this specifies in which subdirectory *Tophat* should save the output files
- **`-G`** GTF file with known transcripts
- **`-j`**: there is an improved spliced alignment by providing *Tophat* with annotated splice junctions. Pre-existing genome annotation is an advantage when analysing RNA-seq data. This file contains the coordinates of annotated splice junctions from Ensembl. While this is highly recommended, we will not add this option in our case

TopHat default values are tuned for processing mammalian RNA-Seq reads. Setting the maximum intron size to 4 or 5 Kb is sufficient to discover most junctions while keeping the number of false positives low. In our case, since we are processing fungal genome we will adjust the following parameters:

- **`-i:`** minimum intron length, from 50 (default value) to 2
- **`-I:`** maximum intron length, from 500,000 (default value) to 1000

Before proceeding, we need the annotation of *C. purpurea* from Ensembl. The GTF file is already available in the `nfs` folder, you can just copy it by typing this command on your home directory:

```
cp \
/nfs/projects/training/RNASeq/Claviceps_purpurea_20_1.ASM34735v1.3
1.gtf ./
```

However, if you want to get an experience on downloading files from Ensembl, here are the steps:

Open the link: http://goo.gl/zGK3BK

Hover your mouse over the *Claviceps* zipped file then right-click it. Then choose 'Copy link Location' if you are using Mozilla Firefox (this depends on your browser) then download this file on your home directory by typing in **'wget'**, then right-clicking your mouse:

```
wget Claviceps_purpurea_20_1.ASM34735v1.31.gtf.gz
```

Wait until the download is done. Unzip the file using this command:

```
gunzip Claviceps_purpurea_20_1.ASM34735v1.31.gtf.gz
```

Now we will proceed with the alignment of the single-end data.

Here is the command to align the quality-base-trimmed `reads` against the reference genome:

```
tophat -g 2 -i 20 -I 1000 \
  --library-type fr-unstranded \
  -o tophat_Claviceps \
  -G Claviceps_purpurea_20_1.ASM34735v1.31.gtf \
  /nfs/projects/training/RNASeq/genome/Claviceps.dna.genome\
  Claviceps.quality.fastq
```

Note: You will have to change the input fastq files and the output folder if you run similar files, e.g. change it to:

```
-o tophat_rep2
```

If you don't change the output folder, then these results will overwrite the previous dataset.

The alignment will take 15-20 minutes. In the meantime please move on with the practical and we will get back to the terminal once the alignment is done.

After the alignment, answer the following question:

How many reads aligned against the reference genome? ___

What is the percentage read mapping rate? ___

Hint: open the align_summary.txt of the Tophat output.

_____

We will first look at some of the files produced by *Tophat*. Look at the `tophat` output folder.

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.

After you generated your results, open the **accepted_hits.bam** file. This is a binary file so we will open it using **samtools**:

```
samtools view accepted_hits.bam | less -S
```

Identify each column and locate the CIGAR string.

IGV VISUALISATION (optional… or you can go back later for this part)

We use IGV to visualise the alignments reported by *Tophat*. If you do not have IGV installed on your machine, download it from https://www.broadinstitute.org/software/igv/download.

After downloading, launch IGV by double clicking the IGV icon on your Desktop. Ignore any warnings and when it opens you have to load the genome of interest. Import the *Claviceps purpurea* genome from the remote cluster to your local machine using WinSCP. Alternatively, you can go to Ensembl and download the fasta sequence to your local drive. Here is the link: http://goo.gl/0Ar5Vb

Load the genome by clicking 'Genomes' on the IGV window then 'Load Genome from File.'

```
Genomes > Load Genome from File
```

In the terminal, sort the BAM file using samtools:

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

Example:

```
samtools sort accepted_hit.bam accepted_hit.sorted
```

Index the sorted file.

```
samtools index [sorted bam file]
```

Example:

```
samtools index accepted_hit.sorted*
```

Then import both sorted bam and the index files on your local drive. Load the sorted bam file by clicking File > Load from File…

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.

Finally following the same process load the Ensembl annotation `*.gtf`.

On the top middle box you can specify the region you want your browser to zoom at . We recommend viewing: **CAGA01000164**

Right-click on the name of the Ensembl track and choose `Expanded`.

Close IGV.

Isoform expression and transcriptome assembly

There are a number of tools that perform reconstruction of the transcriptome and for this workshop we are going to use *Cufflinks*. *Cufflinks* can do transcriptome assembly either *ab initio* or using a reference annotation. It also quantifies the isoform expression in RPKMs or FPKMs.

FPKM stands for **F**ragments **P**er **K**ilobase of exon per **M**illion fragments mapped. What does RPKM stand for?

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

```
cufflinks --help
```

We aim 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 annotation, while in the latter case it will report novel isoforms as well.

The general format of the cufflinks command is:

```
cufflinks [options]* <aligned_reads.(sam/bam)>
```

Where the input is the aligned reads (either in SAM or BAM format).

Some of available parameters of *Cufflinks* that we are going to use to run *Cufflinks* are listed below:

- **-o**: output directory
- **-G**: tells *Cufflinks* to use the supplied annotation strictly in order to estimate isoform annotation
- **-g** tells *Cufflinks* to assemble the transcriptome using the supplied annotation as a guide and allowing for novel transcripts
- **-b**: instructs *Cufflinks* to run a bias detection and correction algorithm which can significantly improve accuracy of transcript abundance estimates. To do this *Cufflinks* requires a multi-fasta file with the genomic sequences against which we have aligned the reads
- **-u**: tells *Cufflinks* to do an initial estimation procedure to more accurately weight reads mapping to multiple locations in the genome (multi-hits)
- **--library-type**: see *Tophat* parameters.
- **-p**: see *Tophat* parameters.

In the terminal type:

```
cufflinks -o cufflinks_output \
    -g ./Claviceps_purpurea_20_1.ASM34735v1.31.gtf\
    -b
    /nfs/projects/training/RNASeq/genome/Claviceps_purpurea
    _20_1.ASM34735v1.31.dna.genome.fa \
    -u \
    --library-type fr-unstranded \
```

tophat_Claviceps/accepted_hits.bam

Note: we will use the output of the previous Tophat run as the input file: **tophat_Claviceps/accepted_hits.bam**

After running the command, take a look at the output folders that have been created. The results from *Cufflinks* are stored in 4 different files named:

`genes.fpkm_tracking, isoforms.fpkm_tracking, skipped.gtf transcripts.gtf.`

Here's a short description of these files:

- `genes.fpkm_tracking`: contains the estimated gene-level expression values.
- `isoforms.fpkm_tracking`: contains the estimated isoform-level expression values.
- `transcripts.gtf`: This GTF file contains *Cufflinks*' assembled isoforms.

The complete documentation can be found at:

http://cole-trapnell-lab.github.io/cufflinks/cufflinks/#cufflinks-output-files
For further reading on Cufflinks, read:

http://homer.salk.edu/homer/basicTutorial/rnaseqCufflinks.html

Note that the above command performs guided transcriptome assembly (transcriptome assembly that reports novel transcripts as well).

Given the previous command, how would you run *Cufflinks* if you are to supply annotation strictly in order to estimate isoform? Do not run this command, just write it below:

Don't forget to change the output folder. Otherwise the second command will overwrite the results of the previous run.

Optional:

Go back to the *IGV* browser and load the file `transcripts.gtf` which is located in the subdirectory `cufflinks/`. Rename the track into something meaningful.

This file contains the transcripts that *Cufflinks* assembled based on the alignment of our reads onto the genome.

In the search box, type **CAGA01000164** in order for the browser to zoom in to the gene of interest. Compare between the already annotated transcripts and the ones assembled by *Cufflinks*. Do you observe any difference?

Look for variants between the reference and the assembled genomes.

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; dormant (Control) and germinating spore (W).

The general format of the `cuffdiff` command is:

```
cuffdiff [options]* <transcripts.gtf>

        <sample1_replicate1.sam[,...,sample1_replicateM]>

        <sample2_replicate1.sam[,...,sample2_replicateM.sam ]>
```

Where: the input includes a `transcripts.gtf` file, which is an annotation file of the genome of interest, and the aligned reads (either in SAM or BAM format) for the conditions.

Some of the *Cufflinks* options that we will use to run the program are:

- **`-o`**: output directory,
- **`-L`**: labels for the different conditions,
- **`-T`**: tells *Cuffdiff* that the reads are from a time series experiment (in our case, this is not a time-series data so we will omit it)
- **`-b`**, **`-p`**, **`-u`**, **`--library-type`**: same as above in *Cufflinks*.

Performing the alignment for the second sample (Germinating/W) would take another 20min. Therefore, we have performed this for you. The aligned reads are stored at

`/nfs/projects/training/RNASeq/`

with the file name **`W_accepted_hits.bam`**

You can either provide the path in the command line or copy (do not move) the file then paste it in your home directory.

To run cuffdiff type on the terminal:

```
cuffdiff -o cuffdiff/ -L Control,Germinating \
     -b /path/to/genome/*.dna.fa -u \

     --library-type fr-unstranded \

     ./Claviceps_purpurea_20_1.ASM34735v1.31.gtf \

     tophat/Claviceps/accepted_hits.bam \

/nfs/projects/training/RNASeq/W_accepted_hits.bam
```

Note that we are comparing two samples each with one replicate. Ideally, to improve statistical power, two or more biological replicates (for RNA-seq) should be considered for differential expression analysis. You will see:

`Warning: No conditions are replicated…`

Ignore this warning for now.

The alignment will take roughly 25 minutes. In the meantime have your snack … or your Water Closet (WC) break.

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 20 cuffdiff/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 ./gene_exp.diff \

     > ./gene_exp_qval.sorted.diff
```

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

```
less -S cuffdiff/gene_exp_qval.sorted.diff
```

Do you see any genes differentially expressed between the two treatments at q-value < 0.05?

Long-term assignment: Extract the fasta sequences of these genes and search their annotation and/or possible function(s).

Outline how you would proceed.

The succeeding parts will be done tomorrow.

Extracting fasta sequence using gffread

After doing the last cufflinks assembly, we will extract fasta sequences which include transcripts that were identified *de novo*. Using the output of the last cufflinks command (with the **–g** option), do the following.

General format:
gffread -w out.fa -g /path/to/genome.fa transcripts.gtf

**gffread -w Claviceps.new.output.fa **
**-g genome/Claviceps_purpurea_20_1.ASM34735v1.31.dna.genome.fa **
cufflinks/transcripts.gtf

Ignore any warnings for now.
Where: cufflinks/transcript.gtf is the transcript file you generated using cufflinks (with -g option)

Remapping

The extracted fasta sequence will serve as the new reference sequence. We will remap the reads using this new reference data file.

First, you need to index your reference file. The reference file is the file you generated using gffread command (above).

What is the command to index reference sequence using bowtie? Do this in the terminal.

_____

What is the purpose of indexing a reference file?

_____

Remap your reads using the generated gffread fasta file using the following commands, taking note of the basename of the index file. Please use your quality-based-trimmed, adapter-clipped fastq file.

```
bowtie2 -x Claviceps.new.output \
-U Claviceps.quality.fastq -a --rdg 6,5 --rfg 6,5 \
--score-min L,-.6,-.4 | samtools view -bS \
-o alignments.bam -
```

Identify aligned reads using the following commands:

```
samtools view -F 4 alignments.bam | less -S
```

Locate the CIGAR column and note the strings used. How many reads aligned to the reference sequence? Hint: bowtie gives a summary statistics after alignment. Obtain the alignment percentage. Alternatively, use the following command:

```
samtools view -F 4 alignments.bam | wc -l
```

Now, identify the unaligned reads using the following commands:

```
samtools view -f 4 alignments.bam | less -S
```

Locate the CIGAR column and note the strings used. How many reads did not align to the reference file?

Do you notice any difference in the CIGAR strings used between the aligned and the unaligned reads? Determine the percentage of aligned and unaligned reads?

Aligned: _____

Unaligned: _____

Compare these values against the ones you obtained in Tophat section.

% improvement in the alignment rate: _____

What do you think is/are the purpose of re-mapping?

_____

What will you do with the un-aligned reads after re-mapping?

_____

CONGRATULATIONS! You've made it to the end of the practical.

We hope you enjoyed it!

Don't hesitate to ask any questions and feel free to contact us any time (email address: nelzo.ereful@niab.com).

References:

1. Trapnell, C., Pachter, L. & Salzberg, S. L. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics* **25**, 1105–1111 (2009).
2. Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* **28**, 511–515 (2010).
3. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S. L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **10**, R25 (2009).
4. Roberts, A., Pimentel, H., Trapnell, C. & Pachter, L. Identification of novel transcripts in annotated genomes using RNA-Seq. *Bioinformatics* **27**, 2325–2329 (2011).
5. Roberts, A., Trapnell, C., Donaghey, J., Rinn, J. L. & Pachter, L. Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biol.* **12**, R22 (2011).

Note: These reference materials are attributed to the original authors.

Sequencing reads in this exercise are used with permission.

Thank you and we hope you enjoyed the exercise.