# RNA-seq made easy: a “how to” manual from raw reads to counts, Wayne lab,UCLA

For part of the practical,we will use an example RNA-Seq data set from RNA samples of wolves from Yellowstone National Park (Charruau and Johnston.2016 Molecular Biology and Evolution). Details of the samples:

- RNA preservation: PAXgene blood RNA tubes
- RNA extraction: PAXgene blood RNA kit
- cDNA library preps: ScriptSeq Complete Gold Kit Blood (used for blood, includes globin mRNA and rRNA depletion; stranded)
- Sequencing: 100 bp paired-end reads on Illumina HiSeq 2000; 5-6 samples/lane (many more samples can be run per lane nowadays)

Loading the data for the workshop purpose

```
http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE80440
```

sample 471F_BL_SE.fastq

```
wget
ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByStudy/sra/SRP/SRP073/SRP07
3500/SRR3402485/SRR3402485.sra
```

These are FASTQ files that are specially compressed by the Short Read Archive and need to be opened using a special tool available in the SRA toolkit. SRA format needs to be converted into fastq to be used as input to Bowtie, Tophat etc. (fastq-dump)

*Resources: https://edwards.sdsu.edu/research/fastq-dump/*

# 1. QUALITY CONTROL: Read removal, trim adapters and low quality bp

"Once the sequencing is finished, the data becomes available for download as “fastq” text files, in which each short read takes up four lines. The first line (starting with an @) is a read identifier, the second is the DNA sequence, the third another identifier (same as line 1, but starting with a +(or sometimes only consisting of a +)) and the fourth is a Phred quality score symbol for each base in the read. The quality score is based on the ASCII character code used by computer keyboards (http://www.ascii□code.com/)(source: SFG)".

"Before we can use our data to answer any biological questions, we must remove poorly identified

bases as well as any adapter sequences from our reads. To evaluate the data set, it is also useful to know what the distribution of quality scores and nucleotides looks like. As the FASTQ files are too large to overview manually, we have to summarize the data and graph it, either by using command□line based software or web server applications (source: SFG)".

Should we remove the duplicates? Duplicates may correspond to biased pcr amplification of particular fragments.For highly expressed short genes, duplicates are expected even if there are no amplification bias Removing them might thus reduce the dynamic range of expression estimates so assess library and decide... If you remove them better to assess the duplicates at the level of paired end reads, not single end reads

*Resources:*
*SAM file description □ http://samtools.sourceforge.net/SAM1.pdf*
*http://sfg.stanford.edu/guide.html Simple Foold Guide to population genomics via RNA-seq*

## 1.1 FastQC

The first step to perform. You want to know the quality of your reads for downstream analyses.

```
fastqc 471F_BL_SE.fastq
```

*Resources:*
*http://www.bioinformatics.babraham.ac.uk/projects/fastqc/*
*https://www.youtube.com/watch?v=bz93ReOv87Y*

## 1.2 Illumina filter

The @ line in the fastq file contains many new fields, one of them indicates whether the read is filtered or not.
Reads that were filtered have 'Y' in the read-ID and are the LOW QUALITY reads. You most likely DO NOT want them!!!
Reads that were NOT filtered have 'N' in the read-ID and are the better-quality reads.

*Usage:*

```
fastq_illumina_filter  [--keep Y/N] [-NYhv] [-o OUTPUT] [INPUT]
```

*Example*:

```
fastq_illumina_filter  --keep N -v -o 471_illuminafilter.fq 471F_BL_SE.fastq
```

So using '-N' or '--keep N' is probably the option you want to use.

*Resource:*
*http://cancan.cshl.edu/labmembers/gordon/fastq_illumina_filter/*

## 1.3 Trim Galore

Trim Galore! is a wrapper script to automate quality and adapter trimming as well as quality control. In the first step, low-quality base calls are trimmed off from the 3' end of the reads before adapter removal. In the next step, its finds and removes adapter sequences from the 3’ end of reads. If no sequence was supplied it will use the first 13 bp of the standard Illumina paired-end adapters ('AGATCGGAAGAGC'), which recognises and removes adapters from most standard libraries.

*Example*:

```
trim_galore -q 20 --fastqc -a AGATCGGAAGAGC --stringency 3 --length 25
471_illuminafilter.fq   # Use trim_galore to remove 3' base pairs with Pred score
<20 and adapter sequences (trim ends if 3 or more base pairs matched adapter
sequence). Removed reads if shorter than 25 base pairs (or if mate read is less
than 25 bp).
```

*Resources:*
*http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/trim_galore_User_Guide_v0.3.7.pdf*

## 1.4 FastQC

You will check again the quality of your reads. If your satisfied, go for downstream analyses otherwise play with the parameters to have good quality.

# 2. MAP READS AGAINST REFERENCE

The first step in the analysis process is to map the RNA-sequence (RNA-seq) reads against the reference genome, which provides the location from which the reads originated. In contrast to DNA-sequence alignment, RNA-seq mapping algorithms have two additional challenges. First, because genes in eukaryotic genomes contain introns, and because reads sequenced from mature mRNA transcripts do not include these introns, any RNA-seq alignment program must be able to handle gapped (or spliced) alignment with very large gaps. In mammalian genomes, introns span a very wide range of lengths, typically from 50 to 100,000 bases, which the alignment algorithm must accommodate. Second, the presence of processed pseudogenes, from which some or all introns have been removed, may cause many exon-spanning reads to map incorrectly. This is particularly acute for the human genome, which contains over 14,000 pseudogenes [2]. There several program that you can use to map your reads like STAR or Tophat2. In this practical, we use Tophat2. TopHat is a program that aligns RNA-Seq reads to a genome in order to identify exon-exon splice junctions. It is built on the ultrafast short read mapping program Bowtie. TopHat runs on Linux and OS X.

To use TopHat, you will need the following programs in your PATH:

- bowtie2 and bowtie2-align (or bowtie)
- bowtie2-inspect (or bowtie-inspect)
- bowtie2-build (or bowtie-build)
- samtools

Because TopHat outputs and handles alignments in BAM format, you will need to download and install the SAM tools. You may want to take a look at the Getting started guide for more detailed installation instructions, including installation of SAM tools and Boost. You will also need Python version 2.6 or higher. To begin, TopHat2 uses Bowtie2 to align reads to the genome. Alignment with Bowtie2 requires an indexed genome, which is represented in a collection of files suffixed with '.bt2'. We need to tell TopHat2 where to find this index when it requires it for alignment. The reference genome must first be "indexed" so that reads may be quickly aligned.

*Resources:*
* *https://www.broadinstitute.org/igv/GFF*
* *https://genomebiology.biomedcentral.com/articles/10.1186/gb-2013-14-4-r36*
* *https://ccb.jhu.edu/software/tophat/manual.shtml*
* *STAR (Alexander Dobin, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras STAR: ultrafast universal RNA-seq aligner. Bioinformatics (2013) 29 (1): 15-21*
* *http://ged.msu.edu/angus/tutorials-2013/rnaseq_tophat.html* *
*https://insidedna.me/tutorials/view/tophat2-analysis-of-rna-expression-is* *
*http://www.ngscourse.org/Course_Materials/alignment/tutorial/example.html*

## Step 1: Build indexes (take a while but you have to do it only once)

A good reference to find genome and annotation file is ENSEMBL. Check the latest version of ENSEMBL and download the genome as well as GTF file.

```
http://www.ensembl.org/info/data/ftp/index.html
```

*example:*

```
ftp://ftp.ensembl.org/pub/release-82/fasta/felis_catus/dna/
Download toplevel.fa.gz # # These files contains all sequence regions flagged as
toplevel in an Ensembl schema. This includes chromsomes, regions not assembled into
chromosomes and N padded haplotype/patch regions.
```

Create a new directory and gunzip the genome

```
cd /work/reference/Felis_Catus
mkdir Ensembl_release82_sept2015
cd Ensembl_release82_sept2015
gunzip Felis_catus.Felis_catus_6.2.dna.toplevel.fa.gz
```

From the directory containing the genome.fa file, run the "bowtie2-build" command.

```
 bowtie2-build -f Felis_catus.Felis_catus_6.2.dna.toplevel.fa FelisCatus62
```

This command will create 6 files with a *.bt2 file extension.

*Resources:*
* *http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#the-bowtie2-build-indexer*
* *https://ccb.jhu.edu/software/tophat/manual.shtml*
* *SAMtools: http://samtools.sourceforge.net/ : a collection of programs to manipulate .sam and .bam files.*

# Step 2: Alignment with Tophat2

*https://genomebiology.biomedcentral.com/articles/10.1186/gb-2013-14-4-r36*

## 2.1 Work with a subset of samples

You have millions of reads, lots of sample. In order to optimize your time, it is highly recommended to work with a subset of samples first (~ 2000000 reads). Once you have chosen your best parameters then you can run the script for all the samples.

*example*:

```
head -n .. yoursamples > subset.fq # don't forget that a fastq file has 4 lines..
```

## 2.2 Optimize your parameters

It is important to consider the species that we will use to map the reads. For instance, if you want to map the reads of bobcats against the cat, the two species diverged around 7.9 Mya!

You can play with several options such as
* --read-mismatches
* --read-gap-length
* --read-edit-dist

- How to control the alignment of reads in terms of number of mismatches, gap length etc?*
- -N (mismatches) : if you have too much mismatches then the risk is to map everywhere! If we increase the mismatch then we increase the edit distance as well as max insertion and deletion.

- For instance, if you want read alignments with at most 2 base mismatches and no gaps then you can specify:
  --read-mismatches 2 --read-gap-length 0 --read-edit-dist 2
- Or if you want read alignments with total length of indels (alignment gaps) of at most 3bp and at most 2 base mismatches you can use these options:
  --read-mismatches 2 --read-gap-length 3 --read-edit-dist 3

An edit distance is a generalization of the concept of number of mismatches (in point of fact, it's a common distance metric for string comparisons). The general idea is that the edit distance is the number of changes to string A required to produce string B. If the only difference between the two is mismatches (e.g. you have an A in one and a T at the same place in another),then the edit distance and number of mismatches are the same. If you have an insertion or deletion between the two strings,then the number of mismatches will be less than the edit distance, as the former lacks any conception of what an insertion or deletion is. Since having insertions/deletions is relatively common when dealing with sequencing data, the concept of an edit distance is rather more useful than the number of mismatches. There are also several other parameters that you might want to consider for the mapping. Check the manuals for these parameters.

### 2.3 Play with the parameters and compare the results in excel between N (mismatches), the uniq reads (aligned 1 time only) and the % overall alignment (summary of tophat or left-mapped bowtie in your directory) then make a graph comparing the % of mapped reads with the number of mismatch.

It is not always straightforward to choose the best parameter!! You use th will use the one that will give you the highest number of reads with high % of unique reads, if you have the same results for different parameters then take the most conservative.

*Example*:

```
tophat2 --read-mismatches 3 --read-gap-length 3 --read-edit-dist 3 --no-convert-bam
--mate-inner-dist 60 --mate-std-dev 35 --min-anchor-length 8 --max-multihits 5
--library-type fr-secondstrand --b2-very-sensitive --b2-N 1 --transcriptome-index
canfam3.1.74.trans canfam3.1.74 file1.fq file2.fq
```

### 2.4 Output files

You will have several output files. For the downstream analyses you will use :

- accepted_hits.bam— This BAM-format file details the mapping of each read to the genome.

You will sort your accepted_hits.bam with samtools

*example:*

```
samtools sort accepted_hits.bam accepted_hits_sorted
```

- The sort command sorts a BAM file based on its position in the reference, as determined by its alignment.
- coordinate: the file sorts by chromosome and start position
- queryname: the file sorts by the read IDs
- unsorted: this option is only supported when the Group Order in the SAM/BAM header is "none"
- Default: coordinate

## 2.5 Quality of the mapping

Once you mapped your reads against the reference genome, you will check for the quality of the mapping.

### 2.5.1 IGV (Resources:https://www.broadinstitute.org/igv/)

IGV requires that the alignment file, whether BAM or SAM, is sorted and indexed by coordinates. Indexing produces a secondary file with either a BAI or SAI extension, respectively. The resulting file can be associated with the alignment track by file naming convention, or loaded independently as a separate track with the index query parameter.

*Example:*

```
samtools index accepted_hits_sorted.bam #generate a .bam.bai that can be used for
the IGV view
```

### 2.5.2 Qualimap (http://qualimap.bioinfo.cipf.es/)

*Example:*

```
qualimap bamqc -bam accepted_hits_sorted.bam #(html file)
```

### 2.5.3 Use 'samtools flagstat' to get a basic summary of an alignment

*Example:*

```
samtools flagstat accepted_hits_sorted.bam
```

### 2.5.4 Another way to check the quality

*Example:*

```
samstat accepted_hits_sorted.bam` # (html file)
```

### 2.6 SORT & FILTER to keep uniquely mapped reads

If read has a pair, only keep pairs that mapped to the same chromosome.
* $5= colonne 5 = quality of the mapping in the sam file, if =50 -->> uniq reads
* $7 =colonne 7 = Ref. name of the mate/next read..
* Here single end reads so $7=="*")

There are different ways to do it

1)

```
samtools sort accepted_hits.bam sorted
samtools view -h sorted.bam > sorted.sam
samtools view -h -q 50 sorted.sam > uniq.sam
```

2)

```
samtools view -H accepted.sam > head.sam
awk '(=="50"&&=="="||=="50"&&=="*") {print


  htseq-count --mode=union --stranded=yes --idattr=gene_id -
  ~/references/Canis_familiaris.CanFam3.1.74.gtf > counts_file.txt


;}' accepted.sam > select.sam
cat head.sam select.sam > final.sam
samtools view -bS final.sam >final.bam
samtools flagstat final.bam
```

# 3. COUNT READS AND NORMALIZATION

## 3.1 First step : counts the read mapped to each gene with Htseq

Read the counts with htseq counts (http://www-huber.embl.de/users/anders/HTSeq/doc/count.html). It is important to make sure that the gene annotation uses the same version and source of reference genome in the reads mapping step above. Take everything from Ensembl! (not the genome from genbank and the gtf from Ensembl for instance)

*Example:*

```
myFiles = list.files(path="D:/POSTDOC_LA/cahierdelabo/Bobcat/ANALYSES-2016/htseq",
pattern="*.txt")
data <- lapply(myFiles, read.table, sep="\t", header=FALSE)
names(data) <- myFiles
for(i in myFiles)
```

```
data[[i]]$Source = i
do.call(rbind, data)
write.table(data, file="htseq.txt",sep="\t")
```

ATTENTION: For libraries prepared using TruSeq Stranded RNA Kits (dUTP second strand marking) you should set htseq-count -stranded=reverse => dUTP-based libraries convey strand with read#2, so htseq-count --stranded=reverse will produce sense counts for such datasets. Using --stranded=yes would then yield anti-sense counts. Htseq-count was unfortunately designed at a time when non-dUTP methods were still popular, so its default stranded option is opposite of what's now common. For TruSeq Stranded libraries it is --fr-firstrand for TopHat and --stranded=reverse for htseq-count but in any case check with a subset of samples to see which one works!

*Example of results with HTSEQ counts:*

ENSFCAG00000000001 129
ENSFCAG00000000006 73
ENSFCAG00000000007 60
ENSFCAG00000000015 19
ENSFCAG00000000021 5
ENSFCAG00000000022 24
ENSFCAG00000000023 1
ENSFCAG00000000024 0
ENSFCAG00000000027 0
ENSFCAG00000000028 3

in R you can do a loop to concatenate all the htseq counts

setwd("D:/POSTDOC_LA/cahierdelabo/Bobcat/ANALYSES-2016/htseq")

```
myFiles = list.files(path="D:/POSTDOC_LA/cahierdelabo/Bobcat/ANALYSES-2016/htseq",
pattern="*.txt") data <- lapply(myFiles, read.table, sep="\t", header=FALSE)
names(data) <- myFiles for(i in myFiles) data[[i]]$Source = i do.call(rbind, data)
write.table(data, file="htseq.txt",sep="\t")
```