**SOFTWARES REQUIRED**

A. [**FastQC**](http://www.bioinformatics.babraham.ac.uk/projects/fastqc/)

[**Use this software to do the quality control analysis of the raw sequences, decide the next step based on the results of this step**]

[MAIN](http://www.bioinformatics.babraham.ac.uk/projects/fastqc/) [MANUAL](http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/) [TUTORIAL](http://www.youtube.com/watch?v=bz93ReOv87Y)

B. [**FASTX-Toolkit**](http://hannonlab.cshl.edu/fastx_toolkit/)

[**Use this software to do the quality control processing of the raw sequences based on the results obtained from above**]

[MAIN](http://hannonlab.cshl.edu/fastx_toolkit/) [MANUAL](http://hannonlab.cshl.edu/fastx_toolkit/commandline.html) [TUTORIAL](http://hannonlab.cshl.edu/fastx_toolkit/galaxy.html)

C. [**BOWTIE**](http://bowtie-bio.sourceforge.net/index.shtml)

[**Bowtie is an ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: typically about 2.2 GB for the human genome (2.9 GB for paired-end)**]

[MAIN](http://bowtie-bio.sourceforge.net/index.shtml) [MANUAL](http://bowtie-bio.sourceforge.net/manual.shtml)

D. [**TOPHAT**](http://tophat.cbcb.umd.edu/)

[**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**]

[MAIN](http://tophat.cbcb.umd.edu/)[MANUAL](http://tophat.cbcb.umd.edu/manual.shtml)

E. [**CUFFLINK**](http://cufflinks.cbcb.umd.edu/)

[**Cufflinks assembles transcripts, estimates their abundances, and tests for differential expression and regulation in RNA-Seq samples. It accepts aligned RNA-Seq reads and assembles the alignments into a parsimonious set of transcripts. Cufflinks then estimates the relative abundances of these transcripts based on how many reads support each one, taking into account biases in library preparation protocols**]

[MAIN](http://cufflinks.cbcb.umd.edu/)[MANUAL](http://cufflinks.cbcb.umd.edu/manual.html)

F. [**SAMTOOLS**](http://samtools.sourceforge.net/)

[**SAM Tools provide various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing and generating alignments in a per-position format**]

[MAIN](http://samtools.sourceforge.net/)[MANUAL](http://samtools.sourceforge.net/samtools.shtml)

G. [**SRATOOLKIT**](http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=doc)

[**SRATOOLKIT IS USED TO CONVERT RAW SEQUENCES IN SRA FORMAT TO FASTQ**]

[MAIN](http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=show&f=software&m=software&s=software)[MANUAL](http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**A. INSTALLATION OF THE REQUIRED SOFTWARES**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[Type the following commands in HPC terminal or you can run the script **0.SoftwareInstaller.sh**]

1. BUILDING THE BIN FOLDER

$ mkdir bin

2. FOR FASTQC INSTALLATION [UPDATE THE FOLDERS WITH THE LATEST VERSION]

$ wget 'http://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc\_v0.10.1.zip'

$ unzip fastqc\_v0.10.1.zip

$ scp -r FastQC/\* $HOME/bin

$ chmod 755 bin /fastqc

$ rm -r fastqc\_v0.10.1.zip

$ rm -r FastQC

3. FOR FASTX TOOLKIT INSTALLATION [UPDATE THE FOLDERS WITH THE LATEST VERSION]

$ mkdir TEMP

$ cd TEMP

$ wget 'http://hannonlab.cshl.edu/fastx\_toolkit/fastx\_toolkit\_0.0.13\_binaries\_Linux\_2.6\_amd64.tar.bz2'

$ tar jxf fastx\_toolkit\_0.0.13\_binaries\_Linux\_2.6\_amd64.tar.bz2

$ scp -r $HOME/TEMP/bin/\* $HOME/bin

$ cd

$ rm -r TEMP

4. FOR BOWTIE1 INSTALLATION [UPDATE THE FOLDERS WITH THE LATEST VERSION]

$ wget 'http://sourceforge.net/projects/bowtie-bio/files/bowtie/0.12.9/bowtie-0.12.9-linux-x86\_64.zip/download'

$ unzip bowtie-0.12.9-linux-x86\_64.zip

$ cp bowtie-0.12.9/bowtie $HOME/bin

$ cp bowtie-0.12.9/bowtie-build $HOME/bin

$ cp bowtie-0.12.9/bowtie-inspect $HOME/bin

$ rm -r bowtie-0.12.9\*

5. FOR BOWTIE2 INSTALLATION [UPDATE THE FOLDERS WITH THE LATEST VERSION]

$ `wget 'http://sourceforge.net/projects/bowtie-bio/files/bowtie2/2.1.0/bowtie2-2.1.0-linux-x86\_64.zip/download'`

$ `unzip bowtie2-2.1.0-linux-x86\_64.zip`

$ `cp bowtie2-2.1.0/bowtie2\* $HOME/bin`

$ `rm -r bowtie2-2.1.0\*`

6. FOR TOPHAT INSTALLATION [UPDATE THE FOLDERS WITH THE LATEST VERSION]

$ wget 'http://tophat.cbcb.umd.edu/downloads/tophat-2.0.2.Linux\_x86\_64.tar.gz'

$ gunzip tophat-2.0.2.Linux\_x86\_64.tar.gz

$ tar -xvf tophat-2.0.2.Linux\_x86\_64.tar

$ cp tophat-2.0.2.Linux\_x86\_64/\* $HOME/bin

$ rm -r tophat-2.0.2.Linux\_x86\_64\*

7. FOR CUFFLINK INSTALLATION [UPDATE THE FOLDERS WITH THE LATEST VERSION]

$ wget 'http://cufflinks.cbcb.umd.edu/downloads/cufflinks-2.0.2.Linux\_x86\_64.tar.gz'

$ gunzip cufflinks-2.0.2.Linux\_x86\_64.tar.gz

$ tar -xvf cufflinks-2.0.2.Linux\_x86\_64.tar

$ cp cufflinks-2.0.2.Linux\_x86\_64/\* $HOME/bin

$ rm -r cufflinks-2.0.2.Linux\_x86\_64\*

8. FOR SAMTOOLS INSTALLATION [UPDATE THE FOLDERS WITH THE LATEST VERSION]

$ wget 'http://sourceforge.net/projects/samtools/files/samtools/0.1.18/samtools-0.1.18.tar.bz2/download'

$ tar jxf samtools-0.1.18.tar.bz2

$ cd samtools-0.1.18

$ make

$ cp samtools $HOME/bin

$ cd

$ rm -r samtools-0.1.18\*

9. FOR SRATOOLKIT INSTALLATION [UPDATE THE FOLDERS WITH THE LATEST VERSION]

$ wget 'http://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/2.3.3-3/sratoolkit.2.3.3-3-centos\_linux64.tar.gz'

$ tar xvf sratoolkit.2.3.3-3-centos\_linux64.tar.gz

$ mv sratoolkit.2.3.3-3-centos\_linux64 bin/

$ rm -r sratoolkit.2.3.3-3-centos\_linux64\*

10. LINKING THE FOLDER TO THE PATH

$ export PATH=$HOME/bin :$PATH

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**B. OBTAINING THE REFERENCE GENOME**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[Type the following commands in local terminal or you can run the script **0A.ReferenceGenomeFetcher.sh**]

Illumina has provided the RNA-Seq user community with a set of genome sequence indexes (including Bowtie indexes) as well as GTF transcript annotation files. These files can be used with TopHat and Cufflinks to quickly perform expression analysis and gene discovery. The annotation files are augmented with the tss\_id and p\_id GTF attributes that Cufflinks needs to perform differential splicing, CDS output, and promoter user analysis. We recommend that you download your Bowtie indexes and annotation files from this page. Go to [**iGenomes**](http://tophat.cbcb.umd.edu/igenomes.shtml)and choose the appropriate reference folder.

Obtain the data directly from the website. Do it on the local machine, as HPC does not allow downloading large files.

[Right click on the selected reference file and choose copy link address, wget with this address]

Type the following in your local machine home.

>> wget 'ftp://igenome:G3nom3s4u@ussd-ftp.illumina.com/Mus\_musculus/NCBI/build37.1/Mus\_musculus\_NCBI\_build37.1.tar.gz'

Copy the folder from the local machine to your HPC home

>> scp -r Mus\_musculus\_NCBI\_build37.1.tar.gz debjit\_ray@hpclogin1.wsu.edu:/home/yelab/debjit\_ray/

Remove the copy from your local machine

>> rm –r Mus\_musculus\_NCBI\_build37.1.tar.gz

Unzip the folder (This would take some time)

$ tar xvf Mus\_musculus\_Ensembl\_NCBIM37.tar.gz

This would create a folder called Mus musculus on your HPC home.

Later you can delete the Mus\_musculus\_Ensembl\_NCBIM37.tar.gz folder.

$ rm -r Mus\_musculus\_Ensembl\_NCBIM37.tar.gz

The Ensemble/NCBIM37/Annotation directory contains another directory called ‘Genes’, which contains a file called ‘genes.gtf’. For the time being, create a link to this file in your working directory (to simplify the commands needed during the protocol).

From your working directory, type:

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Annotation/Genes/genes.gtf

Create links to the 6 **Bowtie1** index included with the iGenome package:

[Avoid this Step when using **Bowtie2**]

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.1.ebwt

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.2.ebwt

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.3.ebwt

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.4.ebwt

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.rev.1.ebwt

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.rev.2.ebwt

Similarly, create links to the 6 **Bowtie2** index included with the iGenome package:

[Avoid this Step when using **Bowtie1**]

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.1.bt2

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.2.bt2

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.3.bt2

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.4.bt2

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.rev.1.bt2

$ ln -s ./Mus\_musculus/Ensembl/NCBIM37/Sequence/BowtieIndex/genome.rev.2.bt2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**C. OBTAINING THE RAW DATA**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Illumina raw sequence data can be mainly of two types:

1. DATA in SRA format
2. DATA in fastq format
3. **DATA in SRA** **format**:

Convert the SRA to fastq using the sratoolkit:

$ bin/./sratoolkit.2.3.3-3-centos\_linux64/bin/fastq-dump --split-3 FILENAME.sra

NOTE: When we run SRA toolkit's "fastq-dump" utility on paired-end sequencing SRA files, sometimes we get only one files where all the mate-pairs are stored in one file rather than two or three files. The solution for the problem is to always run fastq-dump with "--split-3" option. If the experiment is single-end sequencing, only one fastq file will be generated. If it is paired-end sequencing, there may be two or three fastq files. Two files (with suffix "\_1" and "\_2") are matched mate-pair read file where as the third one (without any suffix) contains all the reads that do not have any mate-pairs (or SRA couldn't resolve mate-pairs for them).

1. **DATA in fastq format**:

FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. Both the sequence letter and quality score are encoded with a single ASCII character for brevity. It was originally developed at the Wellcome Trust Sanger Institute to bundle a FASTA sequence and its quality data, but has recently become the de facto standard for storing the output of high throughput sequencing instruments such as the Illumina Genome Analyzer.

Fastq files can be directly used for the next steps.

**Save the raw data in a folder called “Sequencing” on the HPC home.**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**D. QUALITY CONTROL ANALYSIS FOR RAW DATA**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[Type the following commands on HPC or you can run the script **0B.QAPreprocessing.sh**]

Modern high throughput sequencers can generate tens of millions of sequences in a single run. Before analyzing this sequence to draw biological conclusions you should always perform some simple quality control checks to ensure that the raw data looks good and there are no problems or biases in your data, which may affect how you can usefully use it.

Most sequencers will generate a QC report as part of their analysis pipeline, but this is usually focused on identifying problems, which were generated by the sequencer itself. FastQC aims to provide a QC report, which can spot problems, which originate either in the sequencer or in the starting library material.

FastQC can be run in one of two modes. It can either run as a stand alone interactive application for the immediate analysis of small numbers of FastQ files, or it can be run in a non-interactive mode where it would be suitable for integrating into a larger analysis pipeline for the systematic processing of large numbers of files.

$ bin/FastQC/./fastqc [FILE1].fastq

$ bin/FastQC/./fastqc [FILE2].fastq

This creates a folder containing the reports for the different module tests in the same folder as the sequences. For visualizing the reports we need to download them on the local computers, as the HPC does not have any html viewer.

Use [FUGU](http://rsug.itd.umich.edu/software/fugu/) on your local computer to link to the folder and right click and select the option **Download with secure copy**. The next step should be decided based on the results obtained from these.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**E. QUALITY CONTROL BASED PREPROCESSING**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FASTX-Toolkit available tools

**FASTQ-to-FASTA converter**: Convert FASTQ files to FASTA files.

**FASTQ Information**: Chart Quality Statistics and Nucleotide Distribution

**FASTQ/A Collapser**: Collapsing identical sequences in a FASTQ/A file into a single sequence (while maintaining reads counts)

**FASTQ/A Trimmer**: Shortening reads in a FASTQ or FASTQ files (removing barcodes or noise).

**FASTQ/A Renamer**: Renames the sequence identifiers in FASTQ/A file.

**FASTQ/A Clipper**: Removing sequencing adapters / linkers

**FASTQ/A Reverse-Complement**: Producing the Reverse-complement of each sequence in a FASTQ/FASTA file.

**FASTQ/A Barcode splitter**: Splitting a FASTQ/FASTA files containning multiple samples

**FASTA Formatter**: changes the width of sequences line in a FASTA file

**FASTA Nucleotide Changer**: Converts FASTA sequences from/to RNA/DNA

**FASTQ Quality Filter**: Filters sequences based on quality

**FASTQ Quality Trimmer**: Trims (cuts) sequences based on quality

**FASTQ Masker**: Masks nucleotides with 'N' (or other character) based on quality

Choose the actions based on the quality report you obtained earlier from the fastqc.

Usage for adapter sequence trimmer:

$ `fastx\_clipper -v -i [FILENAME].fastq -Q 33 -a [ADAPTER SEQUENCE] -o [FILENAME]\_trimmed.fastq`

**The resulting fatq files are to be used for the later downstream processes.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**F. PROCESSING THE SEQUENCING DATA** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Type the following commands on HPC or you can run the script **0C.Processing.sh**]

-------------------------------------------------------------------------------------------------------------------------------------

**# STEP 1**

-------------------------------------------------------------------------------------------------------------------------------------

***CODE***: \*\* TOPHAT \*\*

***USAGE***: tophat [options]\* <index\_base> <reads1\_1[,...,readsN\_1]> [reads1\_2,...readsN\_2]

***EXAMPLE USING BOWTIE1*** [Avoid this Step when using **Bowtie2**]:

$ tophat --bowtie1 -p 8 -G genes.gtf -o Sequencing/OUT/[FILE1]\_thout genome Sequencing/[FILE1].fastq

$ tophat --bowtie1 -p 8 -G genes.gtf -o Sequencing/OUT/[FILE2]\_thout genome Sequencing/ [FILE2].fastq

***EXAMPLE USING BOWTIE2*** [Avoid this Step when using **Bowtie1**]:

$ tophat -p 8 -G genes.gtf -o Sequencing/OUT/[FILE1]\_thout genome Sequencing/[FILE1].fastq

$ tophat -p 8 -G genes.gtf -o Sequencing/OUT/[FILE2]\_thout genome Sequencing/ [FILE2].fastq

-p means: Use this many threads to align reads. The default is 1. Here we use 8.

-G means: Supply TopHat with a set of gene model annotations and/or known transcripts, as a GTF 2.2 or GFF3 formatted file. If this option is provided, TopHat will first extract the transcript sequences and use Bowtie to align reads to this virtual transcriptome first. Only the reads that do not fully map to the transcriptome will then be mapped on the genome. The reads that did map on the transcriptome will be converted to genomic mappings (spliced as needed) and merged with the novel mappings and junctions in the final tophat output.

-o means:: Sets the name of the directory in which TOPHAT will write all of its output. The default is "./".

***OUTPUT FOLDER***:

Sequencing/OUT/[FILE1]\_thout

Sequencing/OUT/[FILE2]\_\_thout

***OUTPUT FILES*** (in each folder):

accepted\_hits.bam (THIS IS THE MAIN FILE USED LATER)

junctions.bed

insertions.bed

deletions.bed

***NOTES***:

a. accepted\_hits.bam: A list of read alignments in SAM format. SAM is a compact short read alignment format that is increasingly being adopted. The formal specification is http://samtools.sourceforge.net/SAM1.pdf.

b. junctions.bed: A UCSC BED track(http://genome.ucsc.edu/FAQ/FAQformat.html#format1) of junctions reported by TopHat. Each junction consists of two connected BED blocks, where each block is as long as the maximal overhang of any read spanning the junction. The score is the number of alignments spanning the junction.

c. insertions.bed and deletions.bed: UCSC BED tracks of insertions and deletions reported by TopHat.

Inserion.bed - chromLeft refers to the last genomic base before the insertion.

Deletions.bed - chromLeft refers to the first genomic base of the deletion.

-------------------------------------------------------------------------------------------------------------------------------------

**# STEP 2**

-------------------------------------------------------------------------------------------------------------------------------------

***CODE***: \*\* CUFFLINK \*\*

***USAGE***: cufflinks [options]\* <aligned\_reads.(sam/bam)>

***EXAMPLE***:

$ cufflinks -p 8 -o Sequencing/OUT/[FILE1]\_clout Sequencing/OUT/[FILE1]\_thout/accepted\_hits.bam

$ cufflinks -p 8 -o Sequencing/OUT/[FILE2]\_clout Sequencing/OUT/[FILE2]\_thout/accepted\_hits.bam

-p means:: Use this many threads to align reads. The default is 1. Here we use 8.

-o means:: Sets the name of the directory in which Cufflinks will write all of its output. The default is "./".

***OUTPUT FOLDER***:

Sequencing/OUT/[FILE1]\_clout

Sequencing/OUT/[FILE2]\_clout

***OUTPUT FILES*** (in each folder):

transcripts.gtf (THIS IS THE MAIN FILE USED LATER)

isoforms.fpkm\_tracking

genes.fpkm\_tracking

***NOTES***:

a. transcripts.gtf: This GTF file contains Cufflinks' assembled isoforms. The first 7 columns are standard GTF, and the last column contains attributes, some of which are also standardized ("gene\_id", and "transcript\_id"). There one GTF record per row, and each record represents either a transcript or an exon within a transcript.

b. isoforms.fpkm\_tracking: This file contains the estimated isoform-level expression values in the generic FPKM Tracking Format. Note, however that as there is only one sample, the "q" format is not used.

c. genes.fpkm\_tracking: This file contains the estimated gene-level expression values in the generic FPKM Tracking Format. Note, however that as there is only one sample, the "q" format is not used.

-------------------------------------------------------------------------------------------------------------------------------------

**# STEP 3**

-------------------------------------------------------------------------------------------------------------------------------------

Create a file called **assemblies.txt** that lists the assembly file for each sample. The file should contain the following lines:

./Sequencing/OUT/[FILE1]\_clout/transcripts.gtf

./Sequencing/OUT/[FILE2]\_\_clout/transcripts.gtf

-------------------------------------------------------------------------------------------------------------------------------------

**# STEP 4**

-------------------------------------------------------------------------------------------------------------------------------------

***CODE***: \*\*CUFFMERGE\*\*

***USAGE***: cuffmerge [options]\* <assembly\_GTF\_list.txt>

***EXAMPLE***:

$ cuffmerge -g genes.gtf –s Mus\_musculus/Ensembl/NCBIM37/Sequence/WholeGenomeFasta/genome.fa -p 8 assemblies.txt

-g means: An optional "reference" annotation GTF. The input assemblies are merged together with the reference GTF and included in the final output.

-s means: This argument should point to the genomic DNA sequences for the reference. If a directory, it should contain one fasta file per contig. If a multifasta file, all contigs should be present. The merge script will pass this option to cuffcompare, which will use the sequences to assist in classifying transfrags and excluding artifacts (e.g. repeats). For example, Cufflinks transcripts consisting mostly of lower-case bases are classified as repeats. Note that <seq\_dir> must contain one fasta file per reference chromosome, and each file must be named after the chromosome, and have a .fa or .fasta extension.

-p means: Use this many threads to align reads. The default is 1. Here we use 8.

***OUTPUT FOLDER***:

merged\_asm

***MAIN FILES***:

merged.gtf (cuffmerge produces a GTF file that contains an assembly that merges together the input assemblies)

-------------------------------------------------------------------------------------------------------------------------------------

**# STEP 5**

-------------------------------------------------------------------------------------------------------------------------------------

***CODE***: \*\*CUFFDIFF\*\*

***USAGE***: cuffdiff [options]\* <transcripts.gtf> <sample1\_replicate1.bam[,...,sample1\_replicateM]> <sample2\_replicate1.bam[,...,sample2\_replicateM.bam]>]

***EXAMPLE***:

$ cuffdiff -o diff\_out -b Mus\_musculus/Ensembl/NCBIM37/Sequence/WholeGenomeFasta/genome.fa -p 8 -L WT,MUT -u merged\_asm/merged.gtf Sequencing/OUT/SRR527218\_thout/accepted\_hits.bam Sequencing/OUT/SRR527219\_thout/accepted\_hits.bam

**OUTPUT FOLDER**:

diff\_out

***OUTPUT FILES***:

bias\_params.info

cds\_exp.diff

cds.count\_tracking

cds.diff

cds.fpkm\_tracking

cds.read\_group\_tracking

gene\_exp.diff

genes.count\_tracking

genes.fpkm\_tracking

genes.read\_group\_tracking

isoform\_exp.diff

isoforms.count\_tracking

isoforms.fpkm\_tracking

isoforms.read\_group\_tracking

promoters.diff

read\_groups.info

run.info

splicing.diff

tss\_group\_exp.diff

tss\_groups.count\_tracking

tss\_groups.fpkm\_tracking

tss\_groups.read\_group\_tracking

var\_model.info

**Download this folder to your local machine for further analysis**.