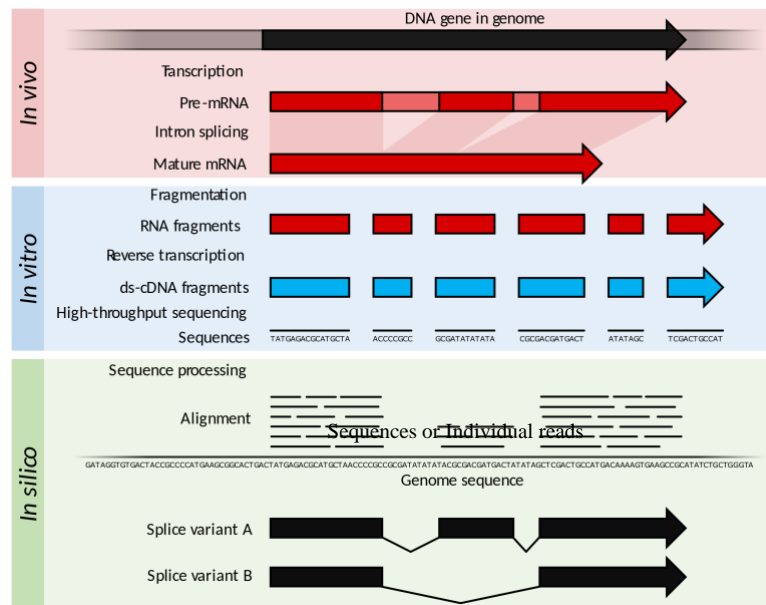


A brief introduction into differential expression analysis tools in the VEuPathDB Galaxy

RNA sequencing (RNA-Seq) uses next-generation sequencing (NGS) technology (e.g. Illumina, PacBio, etc.) to detect RNA. Application of the bioinformatics tools to the raw data allows to estimate RNA quantity and express it per gene and per transcript expression values. Using RNA-Seq, you can also look for novel transcripts and alternatively spliced genes, detect gene fusions or novel genes, etc.



Adapted from Wikipedia. <https://en.wikipedia.org/wiki/RNA-Seq>

Knowing what you want to know and what is the best way to find an answer is a key to a successful RNA-seq study. This includes thinking about technical and biological replicates, the number of replicates, sample pooling, etc. We will not cover best RNA-Seq practices in class, but in the Supplemental section, you will find suggested reads page that you may find useful.

During this workshop, we will work with public data. You will learn how to deploy RNA-Seq workflows for samples with replicates and analyse the results. To learn more about RNA-Seq and best practices, check out the Suggested Reading section.

Differential Expression Analysis is a comparative analysis of transcript abundance

- Assess **quantitative** differences between conditions
- Simplest form – Single-Factor Experiment comparing two conditions
 - e.g wild type vs mutant
- Multi-Factor Experiments comparing multiple conditions / samples
 - e.g Plants infected with fungal pathogen – samples grown at different temperature and with a protective chemical treatment or not.

Experimental design

- Replication – best practice is at least 3 biological replicates which allows for correction of within sample variation
- Biological question – does the design allow you to fully test your hypothesis, do you have the correct controls

Basic statistics that are needed to understand the results

P-value and False discovery rate (FDR) adjusted p values:

A p-value of 0.05 implies that 5% of all tests will result in false positives. An FDR adjusted p-value (or q-value) of 0.05 implies that 5% of significant tests will result in false positives. The latter will result in fewer false positives.

Fold Change:

Fold change is a measure describing how much a quantity changes going from an initial to a final value. For example, an initial value of 30 and a final value of 60 corresponds to a fold change of 2 (or equivalently, a change to 2 times), or in common terms, a one-fold increase.

Types of errors produced by Differential Expression Analysis tools

Type 1 – false POSITIVE:

In a type I error (or false-positive) the null hypothesis is really true (the gene is not differentially expressed) but the statistical test has led you to believe that it is false (there is a difference in expression). This type of error is potentially very dangerous, if a rejected hypothesis allows publication of a scientific finding, a type I error brings a “false discovery”, and the risk of publication of a potentially misleading scientific result.

Type 2 – false NEGATIVE:

In a type II error (or false-negative) the null hypothesis is really false (the gene is differentially expressed) but the test has not picked up this difference. This type of error is less dangerous than the type I but should still be avoided if possible.

Tools that are used in the pre-configured RNA-seq workflows within the VEuPathDB Galaxy Instance.

1. Quality Control and Trimming

FastQC - <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>

- Function: A quality control tool for high throughput sequence data
- Allows for quality control in raw sequence reads
- Can take in FastQ, BAM or SAM files
- Quick overview of problems
- Output: HTML report



FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and the corresponding quality scores. More:

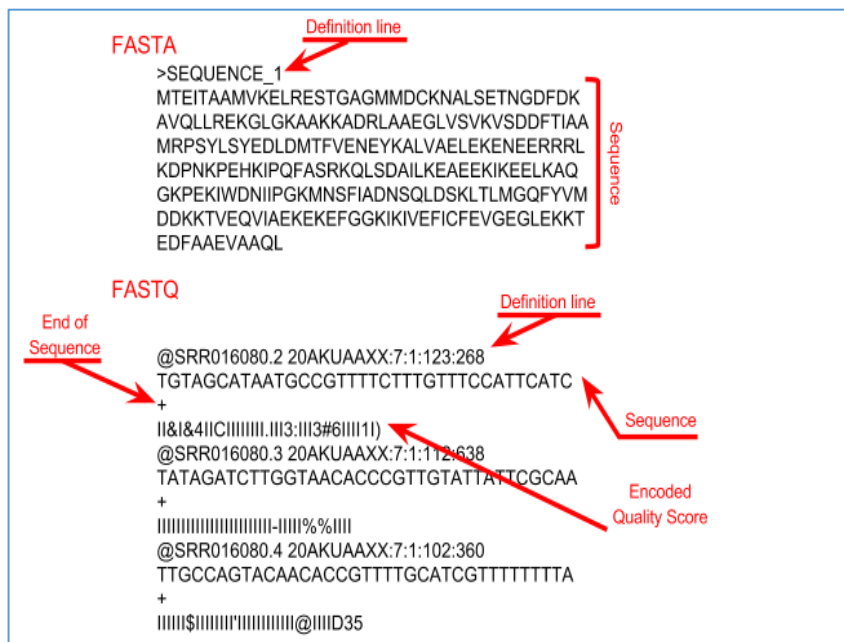
<https://pythonhosted.org/OBITools/fastq.html>

Structure of a FASTQ file

FASTQ file are text files (similar to FASTA) that include sequence quality information and details in addition to the sequence (ie. name, quality scores, sequencing machine ID, lane number etc.). FASTQ files are large and as a result not all sequencing repositories will store this format. However, tools are available to convert, for example, NCBI's SRA format to FASTQ. Sequence data is housed in three repositories that are synchronized on a regular basis.

- The sequence read archive at GenBank
- The European Nucleotide Archive at EMBL
- The DNA data bank of Japan

Illumina, Solexa and Sanger sequencing



Sickle - <https://github.com/ucdavis-bioinformatics/sickle>

- **Function:** Uses sliding windows and length thresholds to trim both 3' and 5' ends of reads using quality of FASTQ files
- Can take input from Illumina, Solexa and Sanger sequencing
- Single or paired-end reads
- **Output:** trimmed FASTQ files

Trimmomatic - <http://www.usadellab.org/cms/?page=trimmomatic>

- Function: Uses sliding window, length and quality thresholds to trim FASTQ data
- Illumina specific
- Adaptor trimming
- Phred-33 or Phred-64 quality
- FastQ files – compressed or uncompressed

2. Alignment tools

GSNAP - <http://research-pub.gene.com/gmap/>

- Function: Align single or paired-end reads to a reference sequence
- Can detect short- and long-distance splicing
- Permits SNP-tolerant alignment to a reference space of all possible combinations of major and minor alleles
- Can align reads from bisulfite-treated DNA for the study of methylation state
- Cannot map color-space data
- Output: SAM

TopHat2- <https://ccb.jhu.edu/software/tophat/index.shtml>

- Function: Spliced aligner tool (an effective method for finding multiexon genes for which a similar cDNA or protein sequence is available). Superseded by HISAT2
- Can align across fusion breaks
- Combines identification of novel splice sites and direct mapping to known transcripts
- Output:
 - junctions -- A UCSC BED track of junctions. 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
 - accepted_hits -- A list of read alignments in BAM format

💡 BED (Browser Extensible Data) format is a flexible way to define the data lines that are displayed in an annotation track. BED has three required fields (see below) and nine additional optional fields:

1. The name of chromosome or scaffold
2. The starting position of the feature in the chromosome or scaffold
3. The ending position of the feature in the chromosomes or scaffold

chr22	1000	5000	cloneA	960	+
chr22	2000	6000	cloneB	900	-
chromosome	chromStart	chromEnd	name	score	strand

More: <https://genome.ucsc.edu/FAQ/FAQformat.html>

Bowtie 2 - <http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>

- Function: Align sequence reads to long reference sequences
- Supports gapped, local, and paired-end alignment modes
- Output: Alignments in SAM format that can be used as inputs for other tools that use SAM (e.g. SAMtools, GATK)

💡 SAM (Sequence Alignment/Map) format is a TAB-delimited text format consisting of a header section (optional) and an alignment information section with short reads mapped against reference sequences. The alignments section contains read name, read sequence, read quality, alignment information, and custom tags. The read name includes the chromosome, start coordinate, alignment quality, and the match descriptor string. SAM is more human-readable and easier to process by conventional text-based processing programs such Python. More: <https://genome.ucsc.edu/FAQ/FAQformat.html>

HISAT2 - <http://ccb.jhu.edu/software/hisat2/index.shtml>

- Function: Spliced alignment program similar to TopHat2
- More than 50 times faster than TopHat2 with better alignment quality
- Output: Alignments in SAM format that can be used as inputs for other tools that use SAM (e.g. SAMtools, GATK)

3. Differential Expression Tools/Pipelines

A. *Cufflinks suite* -- <http://cole-trapnell-lab.github.io/cufflinks/>

Cufflinks

- Function: Uses aligned RNA-Seq reads (SAM or BAM) to assemble transcripts, estimate their abundances, and test for differential expression and regulation
- Input: a text file of SAM alignments
- Output: Three files - Transcripts and Genes (GTF), Transcripts only, and Genes only

💡 GTF (Gene transfer format) is a TAB-delimited file that holds information about gene structure and contains some additional features specific to gene information. More: <https://genome.ucsc.edu/FAQ/FAQformat.html>

💡 BAM (Sequence Alignment/Map) is the compressed binary version of a SAM file that is used to represent aligned sequences up to 128Mb. BAM is less human-readable. More: <https://genome.ucsc.edu/FAQ/FAQformat.html>

Cuffmerge

- Input: GTF file produced by Cufflinks and annotation file of a reference sequence (optional)
 - Combines multiple assemblies to form a master transcriptome
- Output: Merged transcripts files and a GTF file that contains an assembly that merges together the input assemblies

Cuffdiff

- Compares expression levels of genes and transcripts
- Can determine which genes are up- or down-regulated between two or more conditions
- Can determine which genes are differentially spliced or are undergoing other types of isoform-level regulation
- Input: A transcript GFF3 or GTF file produced by cufflinks, cuffcompare, or other source as well as two SAM files containing the fragment alignments for two or more samples
- Output: many files
 - Transcript FPKM expression tracking.
 - Gene FPKM expression tracking; tracks the summed FPKM of transcripts sharing each gene_id
 - Primary transcript FPKM tracking; tracks the summed FPKM of transcripts sharing each tss_id
 - Coding sequence FPKM tracking; tracks the summed FPKM of transcripts sharing each p_id, independent of tss_id
 - Transcript differential FPKM.
 - Gene differential FPKM. Tests difference in the summed FPKM of transcripts sharing each gene_id

- Primary transcript differential FPKM. Tests difference in the summed FPKM of transcripts sharing each tss_id
- Coding sequence differential FPKM. Tests difference in the summed FPKM of transcripts sharing each p_id independent of tss_id
- Differential splicing tests: this tab-delimited file lists, for each primary transcript, the amount of overloading detected among its isoforms, i.e. how much differential splicing exists between isoforms processed from a single primary transcript. Only primary transcripts from which two or more isoforms are spliced are listed in this file.
- Differential promoter tests: this tab-delimited file lists, for each gene, the amount of overloading detected among its primary transcripts, i.e. how much differential promoter use exists between samples. Only genes producing two or more distinct primary transcripts (i.e. multi-promoter genes) are listed here.
- Differential CDS tests: this tab-delimited file lists, for each gene, the amount of overloading detected among its coding sequences, i.e. how much differential CDS output exists between samples. Only genes producing two or more distinct CDS (i.e. multi-protein genes) are listed here.

B. HTSeq-count followed by DESeq2

HTSeq-count - http://htseq.readthedocs.io/en/release_0.9.1/count.html

- Function: Determines counts (how many reads map to a genomic feature – gene, exon etc.)
- Input: Aligned Sam/BAM and annotation file (GFF)
- Output: A table with counts for each feature, followed by the special counters, which count reads that were not counted for any feature for various reasons:
 - *no_feature*: reads which could not be assigned to any feature (set S as described above was empty).
 - *ambiguous*: reads which could have been assigned to more than one feature and hence were not counted for any of these (set S had more than one element).
 - *too_low_aQual*: reads which were not counted due to the -a option, see below
 - *not_aligned*: reads in the SAM file without alignment
 - *alignment_not_unique*: reads with more than one reported alignment. These reads are recognized from the NH optional SAM field tag. (If the aligner does not set this field, multiply aligned reads will be counted multiple times.)

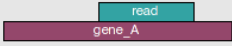
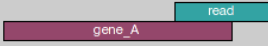

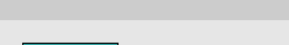

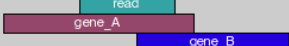
	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous	gene_A	gene_A
	ambiguous	ambiguous	ambiguous

Figure 1: HTSeq-count options for the overlap resolution modes

DESeq2 - <https://bioconductor.org/packages/release/bioc/html/DESeq2.html>

- Function: Determines differentially expressed features from count tables (input)
- Comparison of expression between features
- Takes count data as input
- Can deal with pairwise or multiple comparisons
- Requires replicates
- Based on negative binomial models
- Uses shrinking estimations for dispersion and fold change
- Initial gene-wise dispersion is estimated using maximum likelihood
- A smooth curve is fitted to the data to capture the trend of estimates based on average expression strength
- This is then used as a guide for the second round of estimations that shrink the noisy gene-wise estimates towards the mean
- In previous studies, this method has been shown to be more conservative than the other two methods
- Output: a TAB file

4. VEuPathDB Galaxy tools

VEUPATHDB APPLICATIONS

VEuPathDB Export Tools

Bigwig Files to VEuPathDB

Export one or more bigwig files to VEuPathDB where they can be viewed as tracks in the Genome Browser.

RNA-Seq to VEuPathDB Export an RNA-Seq result to VEuPathDB

VEuPathDB OrthoMCL Tools

OrthoMCL Clean FASTA file

Clean a FASTA file for use in OrthoMCL. The input is a pre-existing proteome FASTA file. The output is a proteome FASTA file that conforms to OrthoMCL requirements

OrthoMCL Reformat Blast

Reformat an NCBI BLAST tabular output file into the tabular format required by OrthoMCL

OrthoMCL Map Proteome to Groups Use BLAST results to map a proteome to OrthoMCL groups.

VEuPathDB RNA-Seq Tools

HTSeqCountToFPKM compute FPKM from per-gene read counts and reference genome

HTSeqCountToTPM compute TPM from per-gene read counts and reference genome

VEuPathDB Galaxy offers several tools to help you analyse dataset using VEuPathDB tools or transfer your results from VEuPathDB Galaxy into private my workspace in VEuPathDB sites where you can continue to analyse and enrich data.

FPKM and TPM are both metrics that are designed to normalise for sequencing depth and gene length. To calculate them, the same operations are carried out but in a different order. For example:

For FPKM:

- Count up all the reads in a sample and divide by 1,000,000 (per million scaling factor)
- For each gene, divide the read counts by the per million scaling factor (reads per million, RPM)
- Divide the RPM value by the gene length in kb

For TPM:

- For each gene, divide the read counts by the length of the gene in kb (reads per kilobase, RPK)
- Sum the RPK values for all the genes in the genome and divide by 1,000,000 (per million scaling factor)
- For each gene, divide the RPK value by the per million scaling factor

Although this seems like a small change, the effect is quite profound. If you are using TPM, the sum of TPM values for all genes should be 1,000,000.

A TPM value of 35 means that if all genes were the same length, then for every 1,000,000 fragments sequenced 35 would map to this gene.

With FPKM, the sum of the FPKM values isn't always the same. This makes FPKM harder to interpret when

comparing samples from different experiments because the denominator might not always be the same (i.e., an FPKM of 35 doesn't always mean 35/1,000,000).

💡 Learn more about FPKM (fragments per kilobase of exon model per million reads mapped) and TPM (Transcripts Per Kilobase Million): <https://www.rna-seqblog.com/rpkm-fpkm-and-tpm-clearly-explained/>