RNA-Seq: Quantification

BCB 5250 Introduction to Bioinformatics II

Spring 2020

Tae-Hyuk (Ted) Ahn
Department of Computer Science
Program of Bioinformatics and Computational Biology
Saint Louis University

SAINT LOUIS
UNIVERSITY
EST. 1818

RNA-Seq Data Analysis Protocol

- Quality Control
- Read mapping or alignment
- Quantification
- Differential gene expression analysis

Software for Quantification

Alignment-based

- HTSeq
- Cufflinks2
- StringTie
- RSEM

Alignment-free (Alignment-independent)

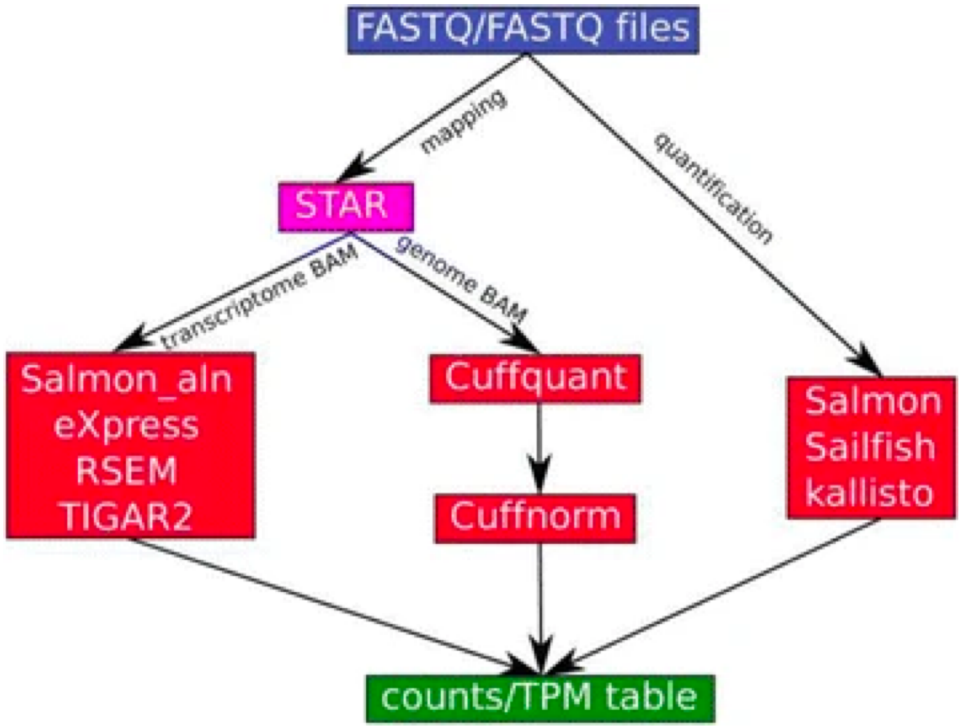
Pseudo-alignment

A classification based RNA-Seq

(Instead of using a genome as a reference, each read is classified as assigned a single transcript.)

- Sailfish
- Salmon
- Kallisto

Possible Workflow



- Workflow for transcript isoform quantification. Sequencing reads were either mapped by STAR aligner or directly fed into alignment-free methods, Salmon, Sailfish or Kallisto. The transcriptome BAM files were quantified by Salmon_aln, eXpress, RSEM or TIGAR2. The genome BAM files were quantified by Cuffquant and then Cuffnorm from the Cufflinks package. The results are summarized into counts and TPM tables for comparison

https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-017-4002-1

Research article | Open Access | Published: 07 August 2017

Evaluation and comparison of computational tools for RNA-seq isoform quantification

Chi Zhang, Baohong Zhang, Lih-Ling Lin & Shanrong Zhao

BMC Genomics **18**, Article number: 583 (2017) | Cite this article

18k Accesses | **30** Citations | **55** Altmetric | Metrics

Counting reads per genes with HTSeq

- Given a BAM file and a list of gene locations, counts how many reads map to each gene.
  - A gene is considered as the union of all its exons.
  - Reads can be counted also per exons.
- Locations need to be supplied in GTF file
  - Note that GTF and BAM must use the same chromosome naming
- Multimapping reads and ambiguous reads are not counted
- 3 modes to handle reads which overlap several genes
  - Union (default), Intersection-strict, Intersection-nonempty
- Attention: was your data made with stranded protocol?
  - You need to select the right counting mode!

> *Important:* The default for strandedness is *yes*. If your RNA-Seq data has not been made with a strand-specific protocol, this causes half of the reads to be lost. Hence, make sure to set the option `--stranded=no` unless you have strand-specific data!

htseq-count

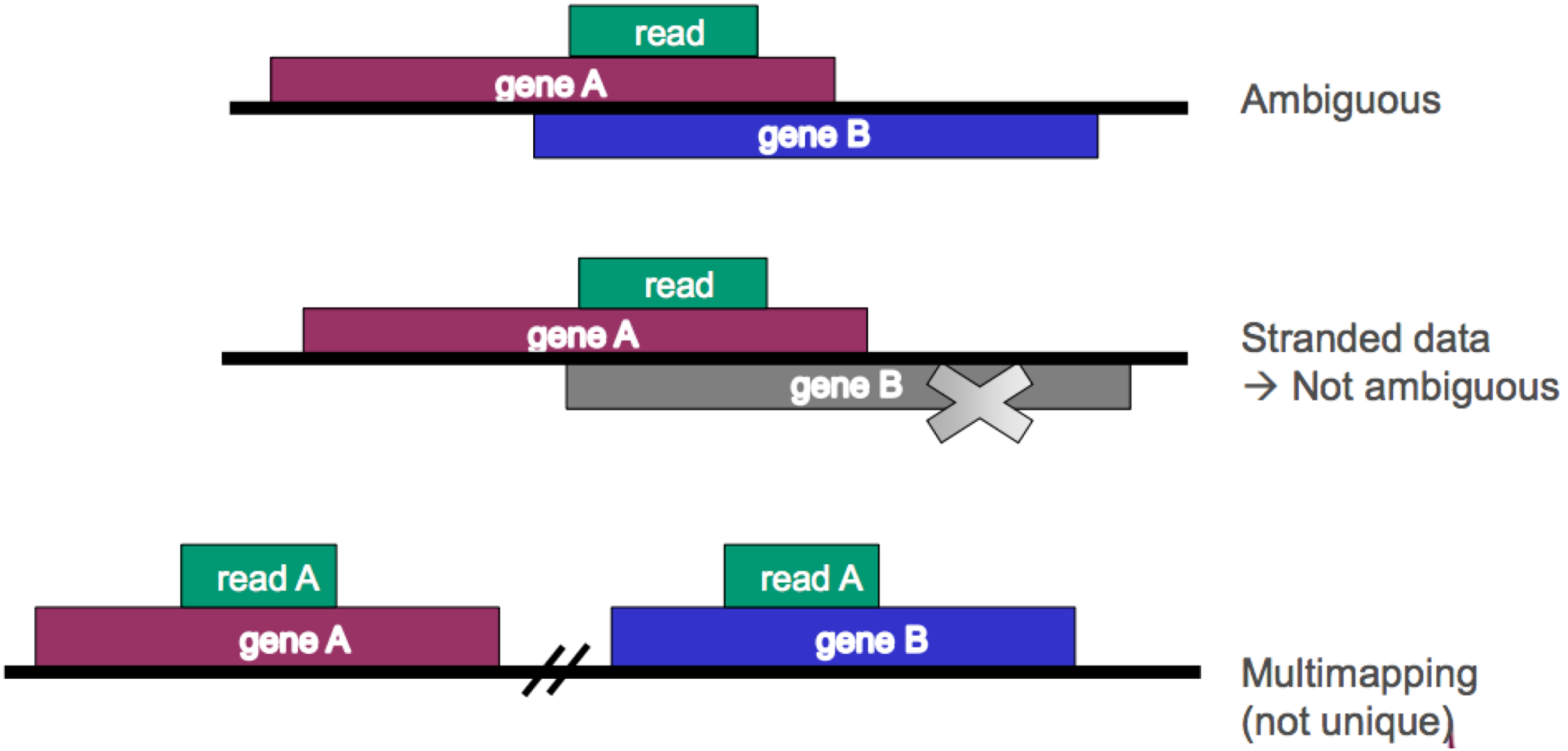
- The script *htseq-count* is a tool for RNA-Seq data analysis: Given a SAM/BAM file and a GTF or GFF file with gene models, it counts for each gene how many aligned reads overlap its exons.
- These counts can then be used for gene-level differential expression analyses using methods such as *DESeq2* or *edgeR*.

htseq-count

The three overlap resolution modes of htseq-count work as follows. For each position *i* in the read, a set *S(i)* is defined as the set of all features overlapping position *i*. Then, consider the set *S*, which is (with *i* running through all position within the read or a read pair)

- the union of all the sets *S(i)* for mode union. This mode is recommended for most use cases.
- the intersection of all the sets *S(i)* for mode intersection-strict.
- the intersection of all non-empty sets *S(i)* for mode intersection-nonempty.

Not unique or ambiguous?



https://chipster.csc.fi/material/rna/RNAseqDataAnalysis2019.pdf

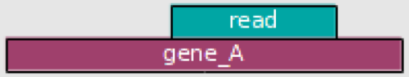
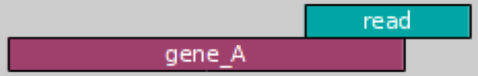
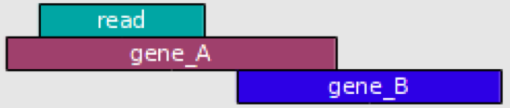
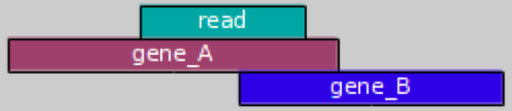
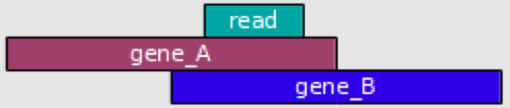
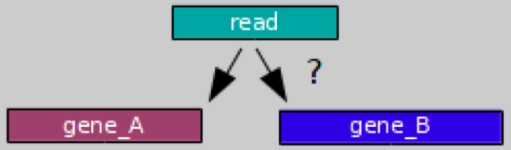
- https://htseq.readthedocs.io/en/release_0.11.1/count.html

If *S* contains precisely one feature, the read (or read pair) is counted for this feature. If *S* is empty, the read (or read pair) is counted as `no_feature`. If *S* contains more than one feature, `htseq-count` behaves differently based on the `--nonunique` option:

- `--nonunique none` (default): the read (or read pair) is counted as `ambiguous` and not counted for any features. Also, if the read (or read pair) aligns to more than one location in the reference, it is scored as `alignment_not_unique`.
- `--nonunique all`: the read (or read pair) is counted as `ambiguous` and is also counted in all features to which it was assigned. Also, if the read (or read pair) aligns to more than one location in the reference, it is scored as `alignment_not_unique` and also separately for each location.

Notice that when using `--nonunique all` the sum of all counts will not be equal to the number of reads (or read pairs), because those with multiple alignments or overlaps get scored multiple times.

The following figure illustrates the effect of these three modes and the `--nonunique` option:

	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 (both genes with --nonunique all)	gene_A	gene_A
	ambiguous (both genes with --nonunique all)		
	alignment_not_unique (both genes with --nonunique all)		

htseq-count

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4287950/

We distribute two stand-alone scripts with HTSeq, which can be used from the shell command line, without any Python knowledge, and also illustrate potential applications of the HTSeq framework. The script *htseq-qa* is a simple tool for initial quality assessment of sequencing runs. It produces plots that summarize the nucleotide compositions of the positions in the read and the base-call qualities.

The script *htseq-count* is a tool for RNA-Seq data analysis: Given a SAM/BAM file and a GTF or GFF file with gene models, it counts for each gene how many aligned reads overlap its exons. These counts can then be used for gene-level differential expression analyses using methods such as *DESeq2* (Love *et al.*, 2014) or *edgeR* (Robinson *et al.*, 2010). As the script is designed specifically for *differential* expression analysis, only reads mapping unambiguously to a single gene are counted, whereas reads aligned to multiple positions or overlapping with more than one gene are discarded. To see why this is desirable, consider two genes with some sequence similarity, one of which is differentially expressed while the other one is not. A read that maps to both genes equally well should be discarded, because if it were counted for both genes, the extra reads from the differentially expressed gene may cause the other gene to be wrongly called differentially expressed, too. Another design choice made with the downstream application of differential expression testing in mind is to count fragments, not reads, in case of paired-end data. This is because the two mates originating from the same fragment provide only evidence for one cDNA fragment and should hence be counted only once.

As the *htseq-count* script has found widespread use over the past 3 years, we note that we recently replaced it with an overhauled version, which now allows processing paired-end data without the need to sort the SAM/BAM file by read name first. See the documentation for a list of all changes to the original version.

run htseq-count

- One example to run:
  - Go to the star_align directory
  - Run as below:

$ htseq-count -f bam ERR188044_chrX_Aligned.sortedByCoord.out.bam ../genes/chrX.gtf > ERR188044_chrX_htseq.out

```
NR_125730   0
NR_126059   72
NR_126093   0
NR_126094   46
NR_126350   0
NR_126557   1
NR_126564   0
NR_130172   0
NR_130179   0
NR_130469   0
NR_130638   0
NR_130730   2
NR_130733   39
NR_130768   0
NR_130770   4
NR_131204   0
NR_131236   3
NR_131237   0
NR_131238   0
NR_131250   0
NR_131982   4
NR_132337   0
NR_132647   0
NR_132964   0
NR_133641   0
__no_feature    1448685
__ambiguous 446972
__too_low_aQual 0
__not_aligned   24119
__alignment_not_unique  30798
```

run htseq-count

Make your own script to run htseq-count with all BAM files

run htseq-count

Make your own script to run htseq-count with all BAM files

- Go to the chrX_data
- Prepare run_htseq.sh as below:

```
#!/bin/bash
dir=/faculty/ahnt/Course/bcb5250/2020S/labs/rna_seq/chrX_data
files=$dir/star_align/*.bam
gff_file=$dir/genes/chrX.gtf
for entry in $files
do
    echo $entry
    basename=$(echo $entry|egrep -o "ERR[^_]+")
    echo $basename
    htseq-count -f bam $entry $gff_file > "$dir"/htseq_count/"$basename"_chrX_htseq_count.txt
done
```

Lab: run featureCounts

http://bioinf.wehi.edu.au/featureCounts/

featureCounts: a ultrafast and accurate read summarization program

featureCounts is a highly efficient general-purpose read summarization program that counts mapped reads for genomic features such as genes, exons, promoter, gene bodies, genomic bins and chromosomal locations. It can be used to count both RNA-seq and genomic DNA-seq reads. It is available in the SourceForge Subread package or the Bioconductor Rsubread package.

Input and output

featureCounts takes as input SAM/BAM files and an annotation file including chromosomal coordinates of features. It outputs numbers of reads assigned to features (or meta-features). It also outputs stat info for the overall summrization results, including number of successfully assigned reads and number of reads that failed to be assigned due to various reasons (these reasons are included in the stat info).

The annotation file should be in either GTF format or a simplified annotation format (SAF) as shown below (columns are tab-delimited):

```
GeneID  Chr     Start   End     Strand
497097  chr1    3204563 3207049 -
497097  chr1    3411783 3411982 -
497097  chr1    3660633 3661579 -
...
```

Lab: run featureCounts

Download featureCount or add publicly available path into your .bashrc

export PATH=/public/ahnt/courses/bcb5250/rna_seq_lab/software/subread-2.0.0-Linux-x86_64/bin:$PATH

Usage: featureCounts [options] -a <annotation_file> -o <output_file> input_file1 [input_file2] ...

Example run with ERR188044_chrX_Aligned.sortedByCoord.out.bam

```
$ featureCounts -p -T 10 -a genes/chrX.gtf -o
feature_count/ERR188044_chrX_featureCount.txt
star_align/ERR188044_chrX_Aligned.sortedByCoord.out.bam
```

Lab: run featureCounts

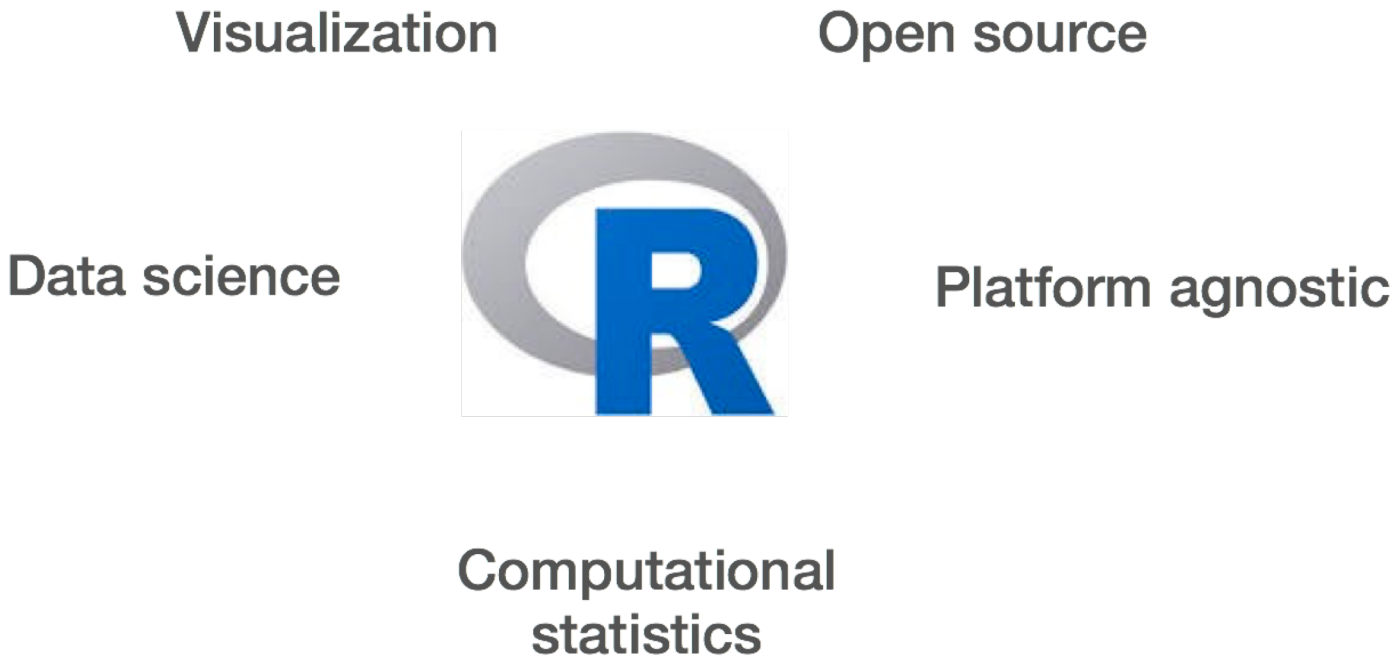
- A nice feature of featureCounts is that we can list multiple BAM files
- Use a special wildcard characters ?, * in Linux.

```
$ featureCounts -p -T 10 -a genes/chrX.gtf -o feature_count/all_featureCounts.txt
star_align/ERR*bam
```

htseq-count vs featureCounts

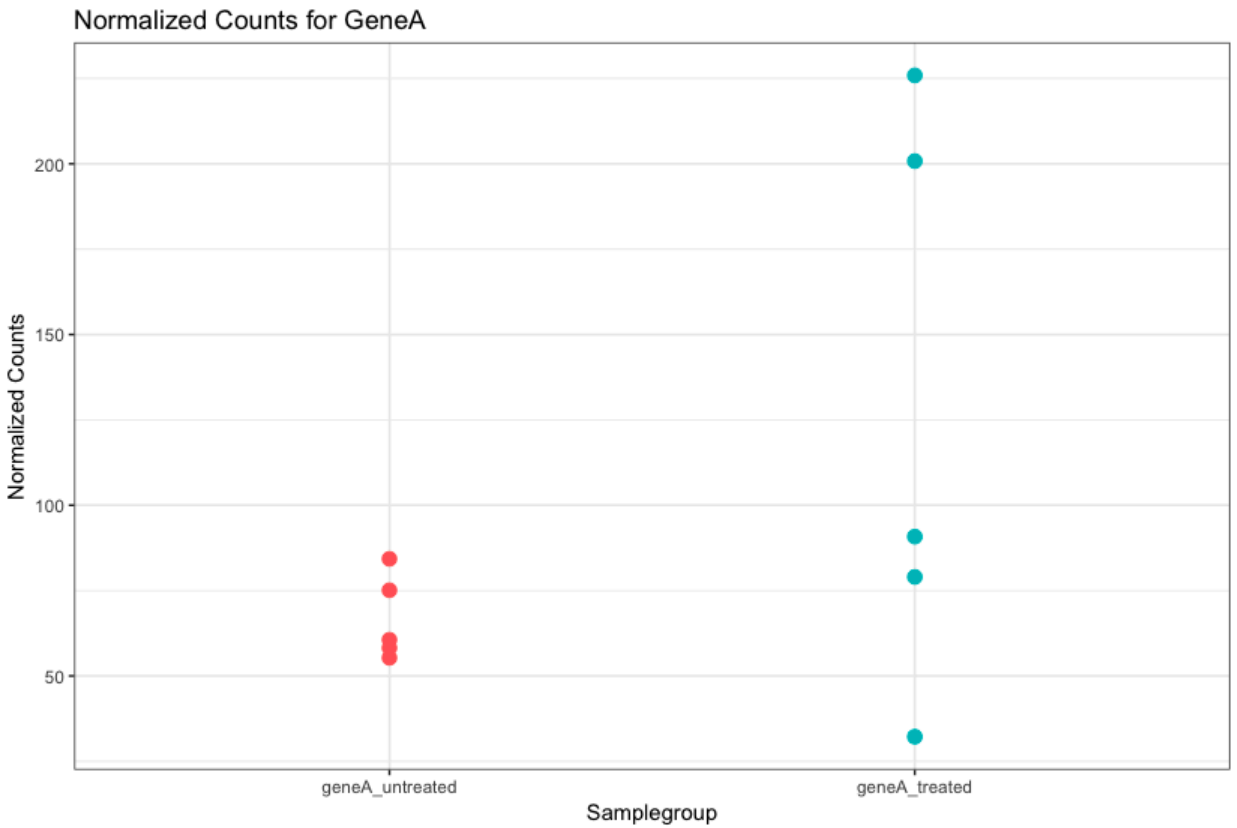
- https://bioinformatics.cvr.ac.uk/featurecounts-or-htseq-count/
- Very similar:
  - Input files: both require for reference -based alignment files (BAM/SAM files) , both could be used for stranded/unstranded reads.
  - Genomic features file: both require for GTF/GFF file from RefSeq or Ensembl.
  - OS: both could be applied in UNIX.
- Different Features: featureCounts supports R version and merge sample results with fast speed.
- Performance Comparison:
  - For example, if two genes were found to both overlap with a reads pair fragment but one gene was found to overlap with only one read and the other with both reads from that fragment, featureCounts will assign that fragment to the gene overlapping with both reads. However, htseq-count will take this fragment as ambiguous and will not assign it to any gene.
  - In my opinion, featureCounts can make a clear distinction for those features that overlap with different numbers of reads from the same fragment, meanwhile I think htseq-count is too conservative and get lower counts.

Statistical models in R



R is a powerful language that can be very useful for NGS data analysis, and there are many popular packages for working with RNA-Seq count data. Some of these packages include edgeR, DESeq2, and limma-voom. All of these tools use statistical modeling of the count data to test each gene against the null hypothesis and evaluate whether or not it is significantly differentially expressed.

RStudio for Differential Expression Analysis



- These methods determine, for each gene, whether the differences in expression (counts) **between groups** is significant given the amount of variation observed **within groups** (replicates). To test for significance, we need an appropriate statistical model that accurately performs normalization (to account for differences in sequencing depth, etc.) and variance modeling (to account for few numbers of replicates and large dynamic expression range).

RStudio for Differential Expression Analysis

- If you do not have RStudio in your laptop, install it. Then, install below packages

  - install.packages("devtools")
  - install.packages(”DESeq2")
  - install.packages("ggplot2")
  - install.packages(" RColorBrewer")
  - install.packages(" reshape2")

p-value

- What a statistical test determines is how likely that null hypothesis is to be true
- The null hypothesis is the hypothesis that nothing is going on (H_0):

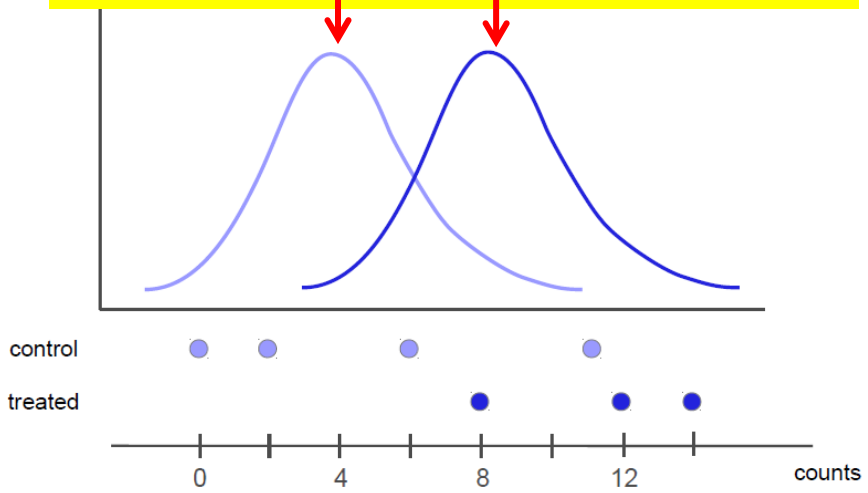
  the gene g is NOT differentially expressed between the conditions
- After a test statistic is computed, it is often converted to a "p-value"
  - If the p-value is small then the null hypothesis is deemed to be untrue and it is rejected in favor of the alternative

http://www.nathalievilla.org/doc/pdf/tutorial-rnaseq.pdf

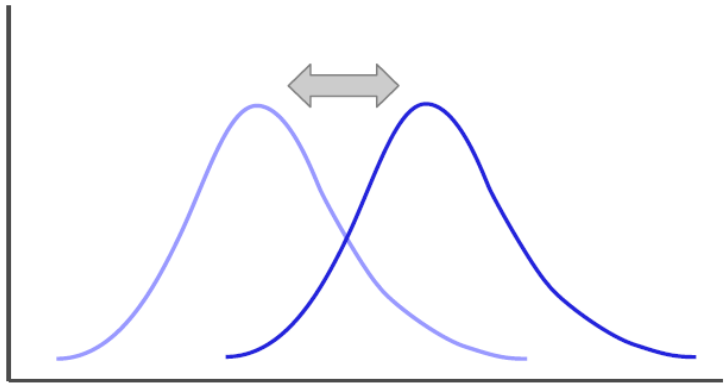
DE: How to quantify the difference

	control				treated		
Gene 1	5	1	0	0	4	0	0
Gene 2	0	2	1	2	1	0	0
Gene 3	92	161	76	70	140	88	70
:	:	:	:	:	:	:	:
Gene g	0	11	2	6	12	8	14
:	:	:	:	:	:	:	:
Gene G	15	25	9	5	20	14	17

NB model is estimated for each gene
2 parameters: mean and dispersion



Fit the model for each gene



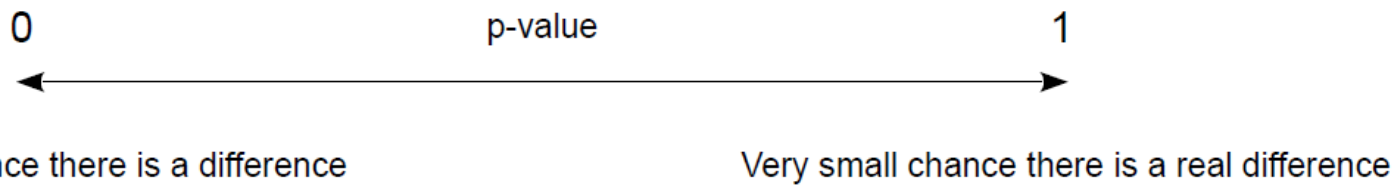
Use statistics to quantify the difference

Difference is put into p-value

http://www.nathalievilla.org/doc/pdf/tutorial-rnaseq.pdf

p-value

- It is a usual convention in biology to use a critical p-value of 0.05 (often called alpha, α).
- There is nothing magical about p-value < 0.05, it is just a convention.
- This means that the probability of observing data as extreme as this if the null hypothesis is true is 0.05 (5% or 1 in 20)
- In other words, it indicates that the null hypothesis is unlikely to be true

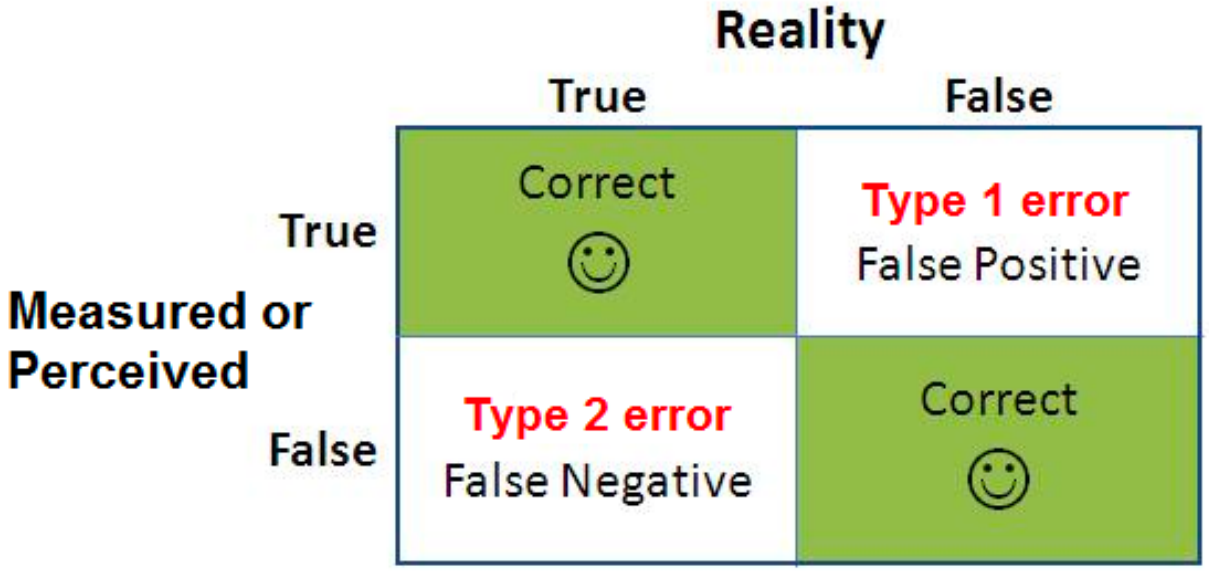


the smaller the p-value the more confident we can be in the conclusions drawn from it.

http://www.nathalievilla.org/doc/pdf/tutorial-rnaseq.pdf

Type I and Type II error

- 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).
- 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.



https://www.abtasty.com/blog/type-1-and-type-2-errors/

Multiple testing issue

- The test for each gene has a probability of producing a type I error
- By performing a large number of hypothesis tests, a substantial number of false positives may accumulate

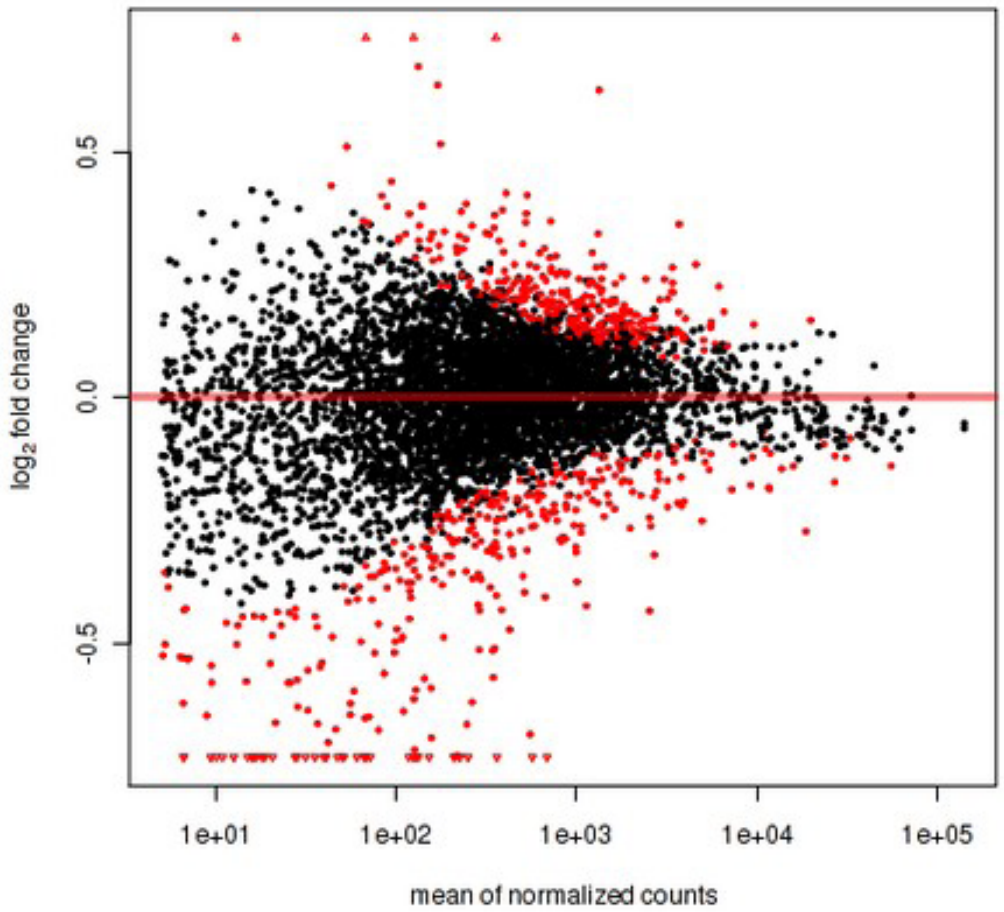
Why Multiple Testing Matters

- Genomics = Lots of Data = Lots of Hypothesis Tests
- Consider:
  - A genome with 10,000 genes (result in performing 10,000 separate hypothesis tests)
  - If we use a standard p-value cut-off of 0.05
  - How many genes are expected to be "significant" by chance?

0.05 × 10,000 = 500 genes.

  - If there are 1,500 genes have p < 0.05 under treament, % of false positive?

500 genes, i.e., 33%



Multiple testing correction

- Goal: to control false discovery rate (FDR)
- 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. This will result in fewer false positives.
  - q-value = 0.05 means that there is a 5% chance that these expression values are from a not differentially expressed gene.
- FDR is the fraction of false positive in the genes that are classified as DE
- If we set a threshold *a* of 0.05, 5% of the detected DE genes will be false positive
- Calculating adjusted p-values (q-values):
  - Bonferroni correction
  - Benjamini/Hochberg correction

DE: How to quantify the difference?

Why log transform?

Fold change = $X_{treatment}/X_{control}$

Original ratio		Log2-tranformed ratio	
Up	Down	Up	Down
2	0.5	1	-1
4	0.25	2	-2
8	0.125	3	-3
16	0.0625	4	-4

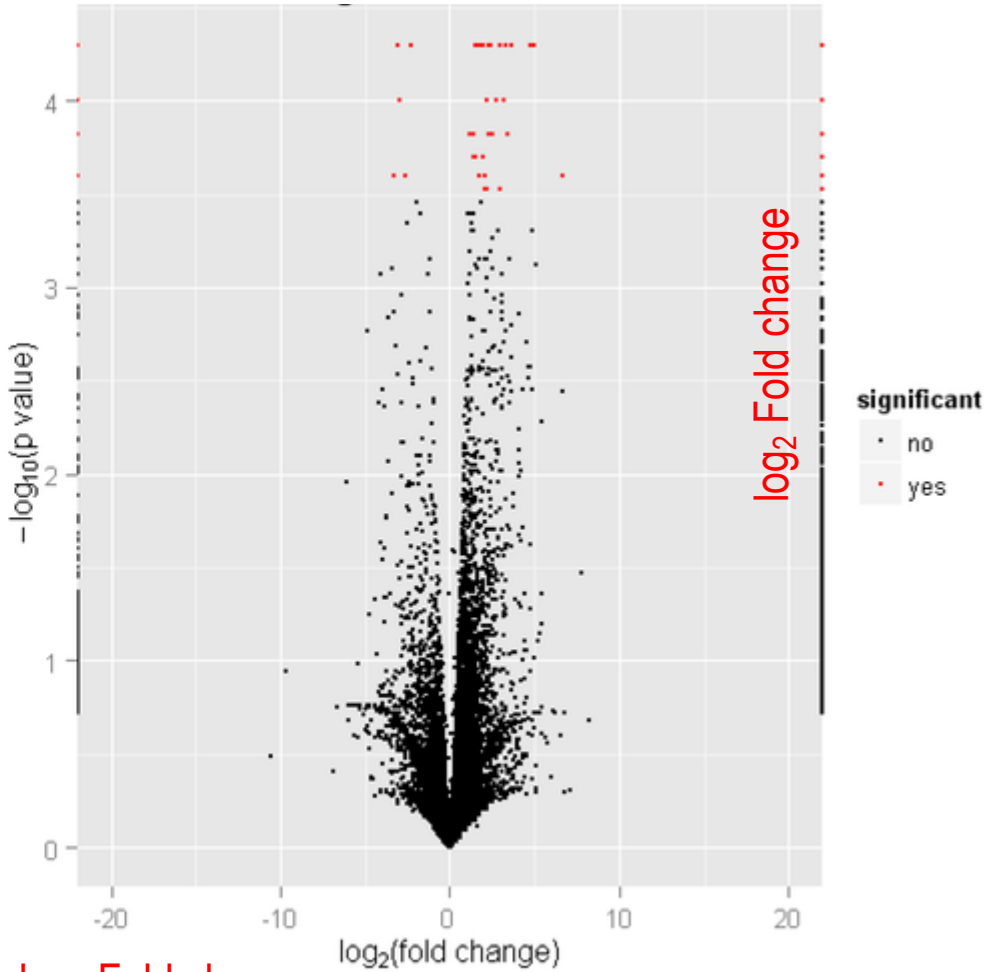
The main reason behind this is in order to be able to compare under expression and over expression on the same scale.

Illustrating DE results

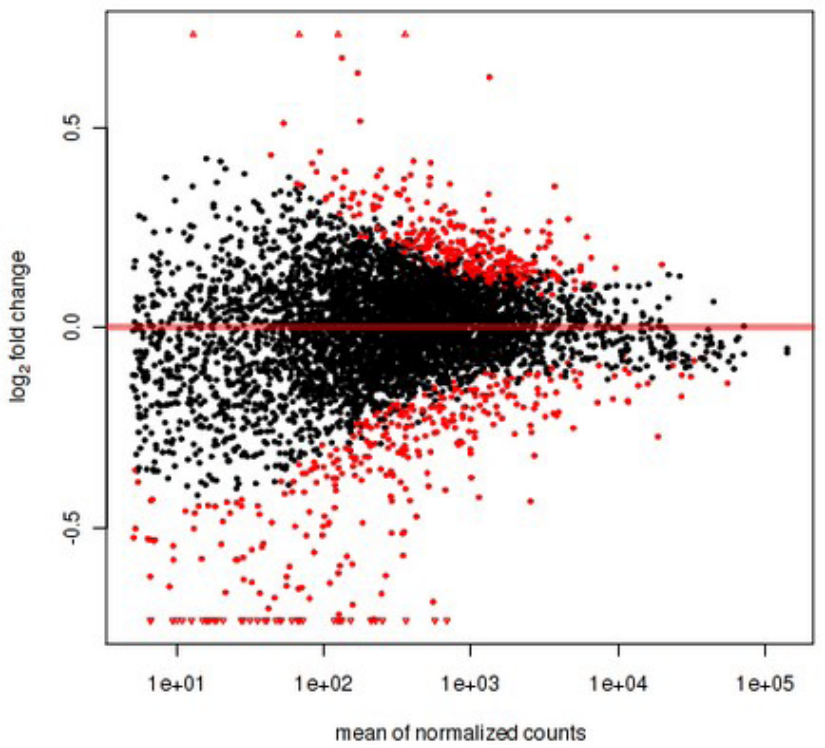
Volcano plot
- x: log2(fold change)
- y: -log10(p-value)

MA plot
- x: mean expression
- y: log2(fold change)

A dot represents one gene
Red dots are significant



volcano-plot: p value versus the log2 fold change between 2 conditions.



MA-plot: mean of counts versus the log2 fold change between 2 conditions.

Multiple papers for comparisons

- http://crazyhottommy.blogspot.com/2016/07/comparing-salmon-kalliso-and-star-htseq.html

RPKM, FPKM, TPM

- RPKM (Reads Per Kilobase Million) and FPKM (Fragments Per Kilobase Million) were used, but TPM (Transcripts Per Million) is now becoming quite popular.
- RPKM
  - Count up the total reads in a sample and divide that number by 1,000,000 – this is our “per million” scaling factor.
  - Divide the read counts by the “per million” scaling factor. This normalizes for sequencing depth, giving you reads per million (RPM)
  - Divide the RPM values by the length of the gene, in kilobases. This gives you RPKM.
- FPKM is very similar to RPKM. RPKM was made for single-end RNA-seq, where every read corresponded to a single fragment that was sequenced. FPKM was made for paired-end RNA-seq
- TPM is very similar to RPKM and FPKM. The only difference is the order of operations. Here’s how you calculate TPM
  - Divide the read counts by the length of each gene in kilobases. This gives you reads per kilobase (RPK).
  - Count up all the RPK values in a sample and divide this number by 1,000,000. This is your “per million” scaling factor.
  - Divide the RPK values by the “per million” scaling factor. This gives you TPM.

RPKM, FPKM and TPM, Clearly Explained!!!

- https://www.youtube.com/watch?v=TTUrtCY2k-w

To understand the differences between TPM and RPKM and FPKM, we'll work through the math using an imaginary RNA-seq data with three replicates (Rep1, 2 and 3) for a genome with 4 genes (A, B, C and D).

Gene Name	Rep1 Counts	Rep2 Counts	Rep3 Counts
A (2kb)	10	12	30
B (4kb)	20	25	60
C (1kb)	5	8	15
D (10kb)	0	0	1

Genes

RPKM

RPKM – step 1: normalize for read depth.

Gene Name	Rep1 Counts	Rep2 Counts	Rep3 Counts
A (2kb)	10	12	30
B (4kb)	20	25	60
C (1kb)	5	8	15
D (10kb)	0	0	1
Total reads:	35	45	106
Tens of reads:	3.5	4.5	10.6

RPM - scaled using the "per million" factors.

Gene Name	Rep1 RPM	Rep2 RPM	Rep3 RPM
A (2kb)	2.86	2.67	2.83
B (4kb)	5.71	5.56	5.66
C (1kb)	1.43	1.78	1.43
D (10kb)	0	0	0.09

RPKM – step 2: normalize for gene length.

Gene Name	Rep1 RPM	Rep2 RPM	Rep3 RPM
A (2kb)	2.86	2.67	2.83
B (4kb)	5.71	5.56	5.66
C (1kb)	1.43	1.78	1.42
D (10kb)	0	0	0.09

Reads are scaled for depth (M) and gene length (K).

Gene Name	Rep1 RPKM	Rep2 RPKM	Rep3 RPKM
A (2kb)	1.43	1.33	1.42
B (4kb)	1.43	1.39	1.42
C (1kb)	1.43	1.78	1.42
D (10kb)	0	0	0.009

TPM

TPM – step 1: normalize for gene length

Original data:

Gene Name	Rep1 Counts	Rep2 Counts	Rep3 Counts
A (2kb)	10	12	30
B (4kb)	20	25	60
C (1kb)	5	8	15
D (10kb)	0	0	1

RPK – scaled by gene length:

Gene Name	Rep1 RPK	Rep2 RPK	Rep3 RPK
A (2kb)	5	6	15
B (4kb)	5	6.25	15
C (1kb)	5	8	15
D (10kb)	0	0	0.1

TPM – step 2: normalize for sequencing depth

Gene Name	Rep1 RPK	Rep2 RPK	Rep3 RPK
A (2kb)	5	6	15
B (4kb)	5	6.25	15
C (1kb)	5	8	15
D (10kb)	0	0	0.1
Total RPK:	15	20.25	45.1
Tens of RPK:	1.5	2.025	4.51

TPM – scaled by gene length and sequencing depth (M):

Gene Name	Rep1 TPM	Rep2 TPM	Rep3 TPM
A (2kb)	3.33	2.96	3.326
B (4kb)	3.33	3.09	3.326
C (1kb)	3.33	3.95	3.326
D (10kb)	0	0	0.02

RPKM vs TPM

RPKM vs TPM

RPKM

Gene Name	Rep1 RPKM	Rep2 RPKM	Rep3 RPKM
A (2kb)	1.43	1.33	1.42
B (4kb)	1.43	1.39	1.42
C (1kb)	1.43	1.78	1.42
D (10kb)	0	0	0.009
Total:	4.29	4.5	4.25

... the sums of each column are very different.

TPM

Gene Name	Rep1 TPM	Rep2 TPM	Rep3 TPM
A (2kb)	3.33	2.96	3.326
B (4kb)	3.33	3.09	3.326
C (1kb)	3.33	3.95	3.326
D (10kb)	0	0	0.02
Total:	10	10	10

RPKM vs TPM

RPKM

Gene Name	Rep1 RPKM	Rep2 RPKM	Rep3 RPKM
A (2kb)	1.43	1.33	1.42
B (4kb)	1.43	1.39	1.42
C (1kb)	1.43	1.78	1.42
D (10kb)	0	0	0.009
Total:	4.29	4.5	4.25

With RPKM, it is harder to compare the proportion of total reads because each replicate has different total (each pie has a different size)

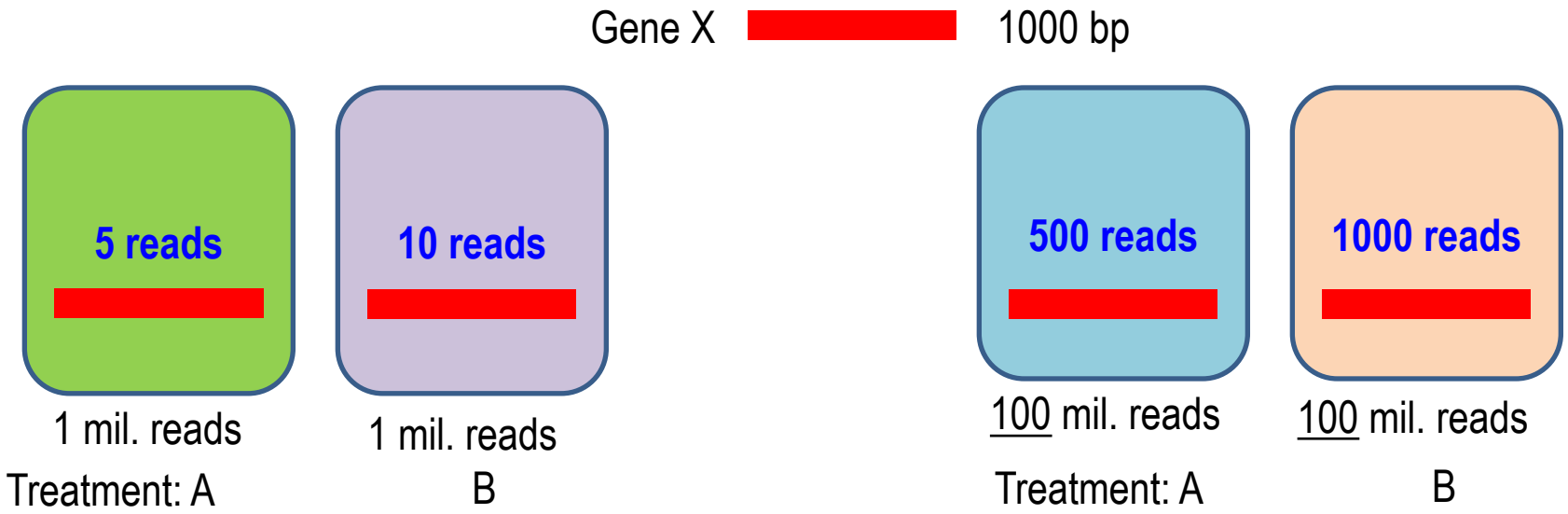
A 1.43 size slice represents a different proportion of reads in in different pies.

Summary: RPKM, FPKM, TPM

- So you see, when calculating TPM, the only difference is that you normalize for gene length first, and then normalize for sequencing depth second. However, the effects of this difference are quite profound.
- When you use TPM, the sum of all TPMs in each sample are the same. This makes it easier to compare the proportion of reads that mapped to a gene in each sample. In contrast, with RPKM and FPKM, the sum of the normalized reads in each sample may be different, and this makes it harder to compare samples directly.

- https://statquest.org/2015/07/09/rpkm-fpkm-and-tpm-clearly-explained/
- http://crazyhottommy.blogspot.com/2016/07/comparing-salmon-kalliso-and-star-htseq.html

Be Careful with RPKM/FPKM Values



RKPM of Gene X

In A, X_{FPKM}= 5 reads/(1kb*1mil)=5

In B, X_{FPKM}= 10 reads/(1kb*1mil)=10

RKPM of Gene X

In A, X_{FPKM}= 500 reads/(1kb*100mil)=5

In B, X_{FPKM}= 1000reads/(1kb*100mil)=10

The RPKM values would be the same for both scenarios

In the latter case, we can be much more confident that there is a true difference between the two treatments than in the first one

Thus, RPKM/FPKM are useful for reporting expression values, but NOT for statistical testing!

Why raw count?

- In principle, counting reads that map to a catalog of features is straightforward.
- Raw read counts is required to correctly model the Poisson component of the sample-to-sample variation
- raw read counts is required for statistical inference based on the negative binomial distribution.

Anders S, et al 2013. Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. Nat Protoc 8:1765-1786.

Why raw count?

- Both DESeq and edgeR internally keep the raw counts and normalization factors separate, as this full information is needed to correctly model the data.
- No prior normalization or other transformation should be applied, including quantities such as RPKM, FPKM or otherwise depth-adjusted read counts.

Anders S, et al 2013. Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. Nat Protoc 8:1765-1786.

Counting rules

- Count reads, not base-pairs
- Count each read at most once.
- Discard a read if
  - it cannot be uniquely mapped
  - its alignment overlaps with several genes
  - the alignment quality score is bad
  - (for paired-end reads) the mates do not map to the same gene

Very good RNA-seq resource

- http://www.nathalievialaneix.eu/doc/pdf/tutorial-rnaseq.pdf
- https://www.encodeproject.org/rna-seq/