TRANSCRIPTOMICS

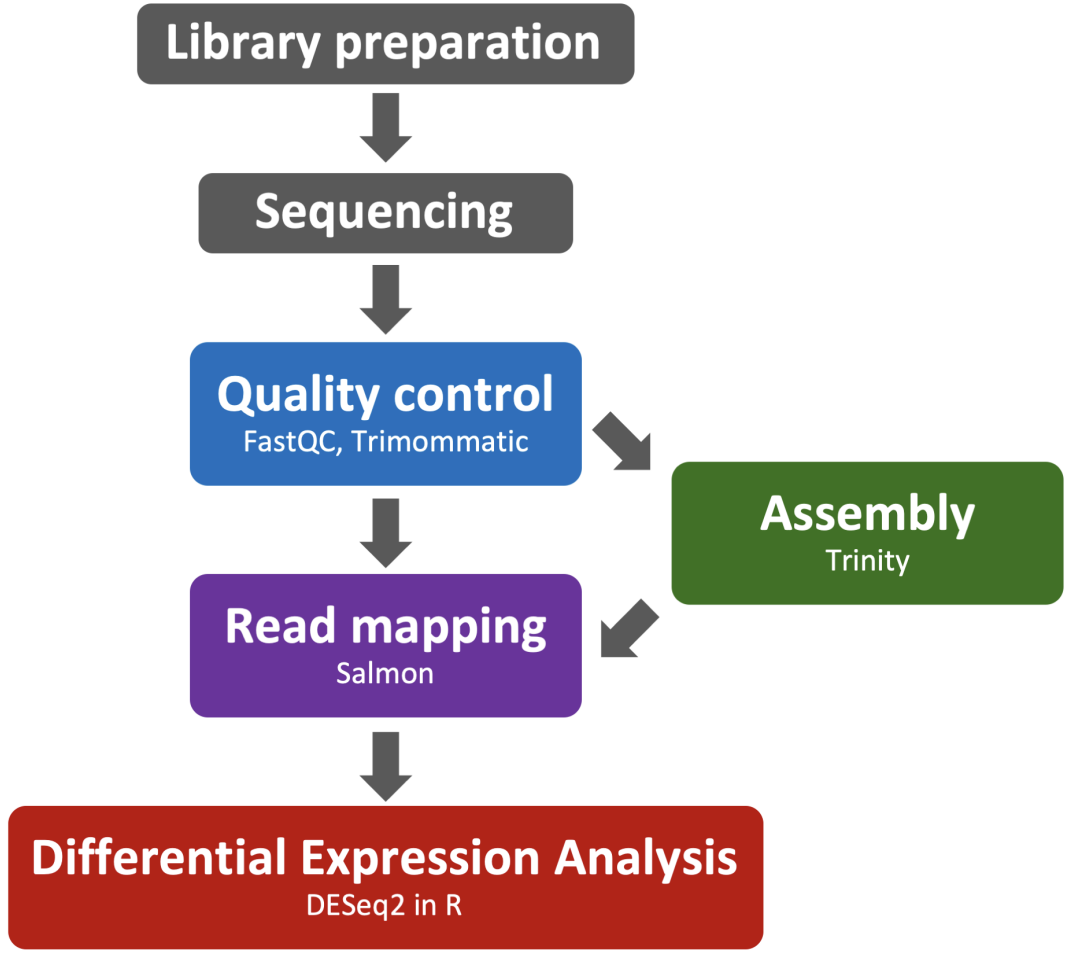
Read Mapping

Day 05

https://tttorres.github.io/transcriptomics/

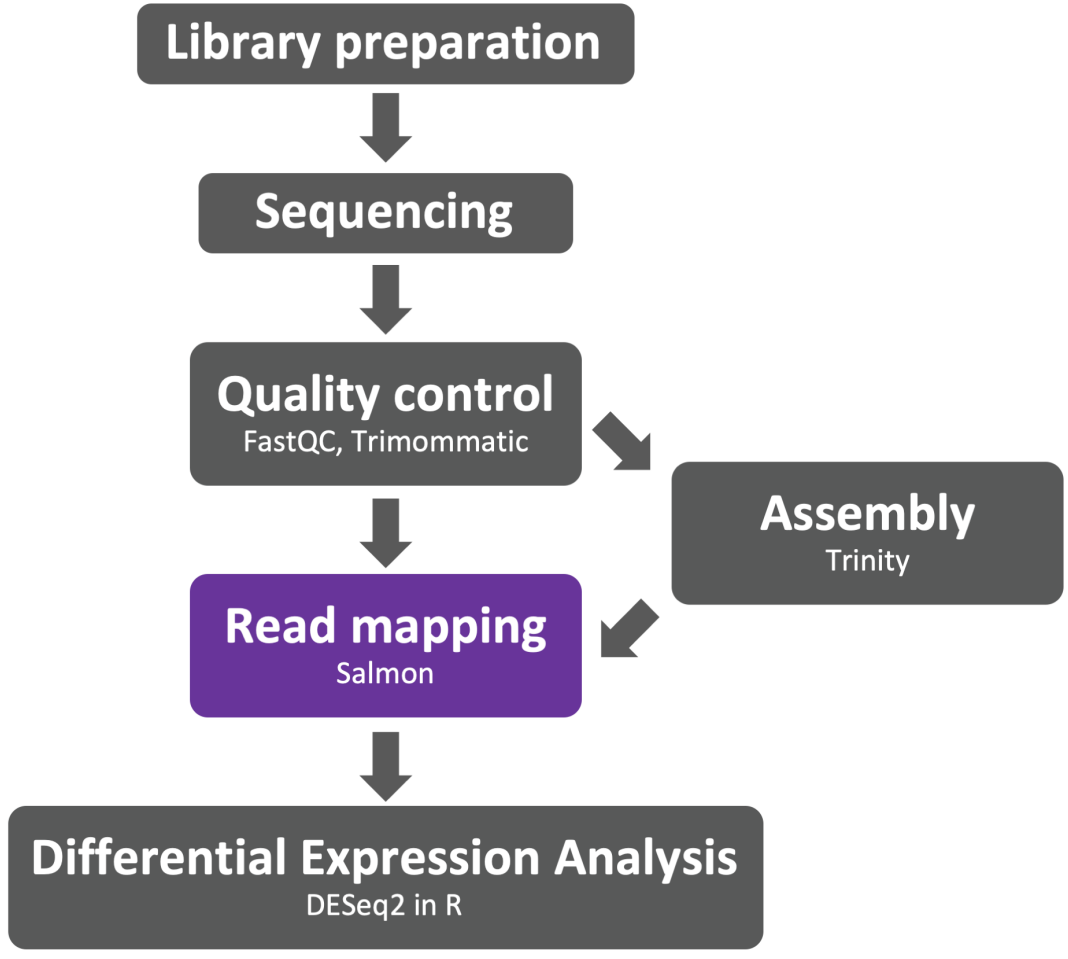
RNA-seq workflow

Read Mapping



RNA-seq workflow

Read Mapping

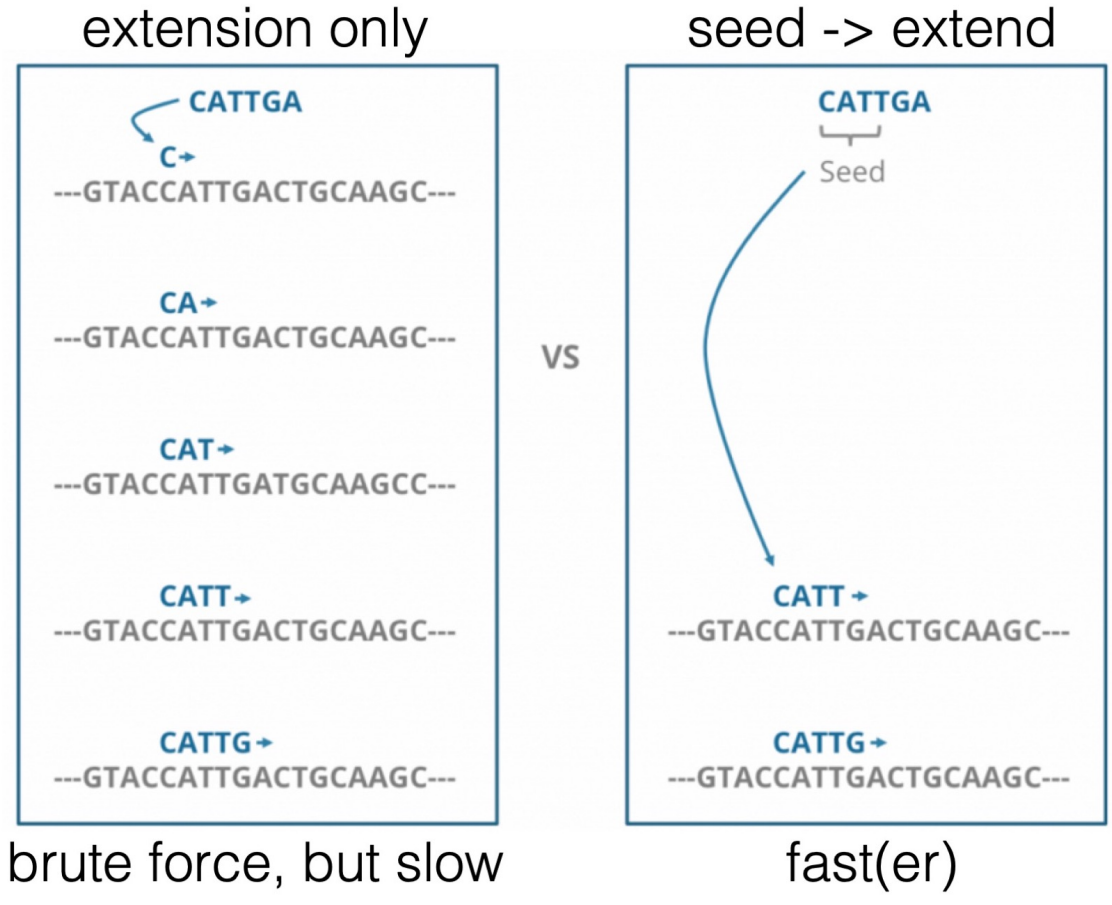


RNA-Seq: Transcript Quantification

Text pattern matching

RNA-Seq: Transcript Quantification

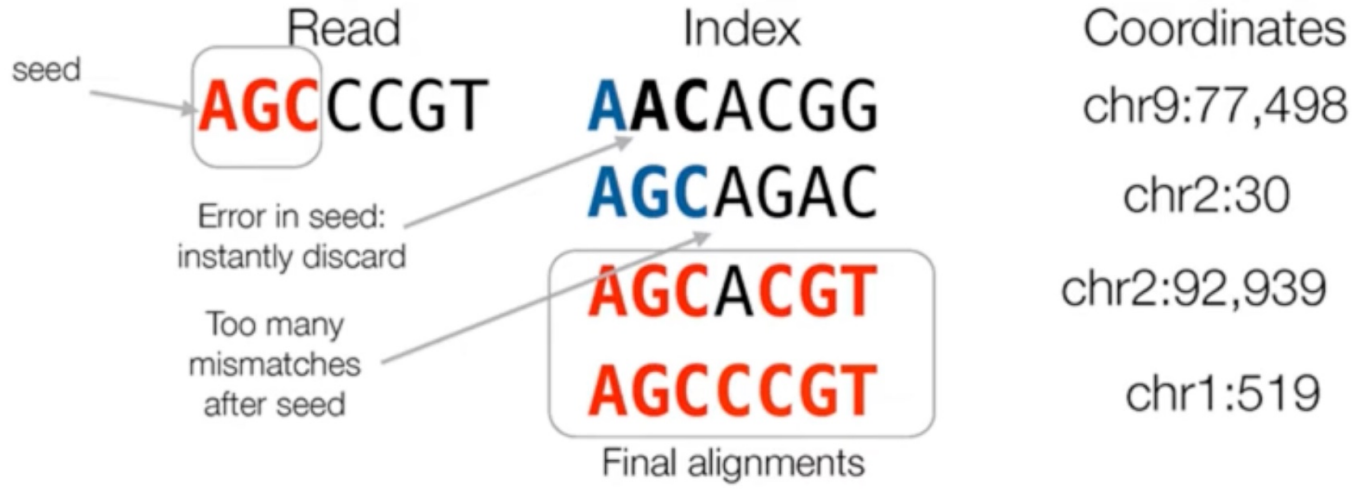
"Seed and Extend" alignments



https://blog.sbgenomics.com/short-read-alignment-seeding/

RNA-Seq: Alignment/Pseudoalignment

"Seed and Extend" alignments: Bowtie



RNA-Seq: Alignment/Pseudoalignment

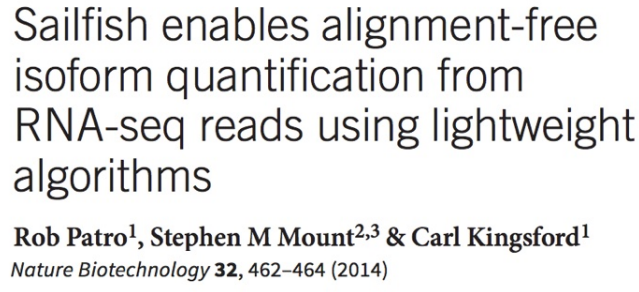
"Seed and Extend" alignments: Bowtie

- Simplification of how many modern aligners work
- Bowtie's seed region only allows for a small number of mismatches
- Bowtie's index allows for a quick lookup and each base reduces the number of possible matches
- This is fine for transcriptome, but reads (usually) come from exons
    - We need a way to deal with "splice reads"

RNA-Seq: Alignment/Pseudoalignment

Sailfish

> **Reads are hard to align, but that's a function of their length. Why not approach the problem by shredding reads into Kmers** - enter "Sailfish"

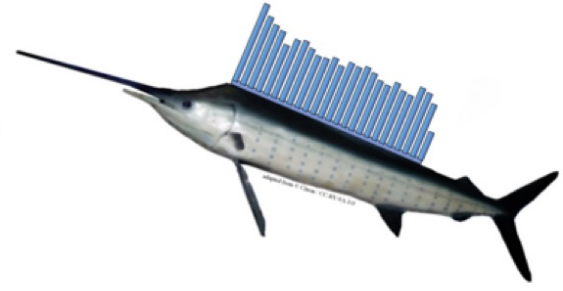


- Sailfish replaces approximate alignment of (error prone) reads with exact alignment of short k-mers

RNA-Seq: Alignment/Pseudoalignment

Sailfish

Sailfish
'lightweight read alignment'

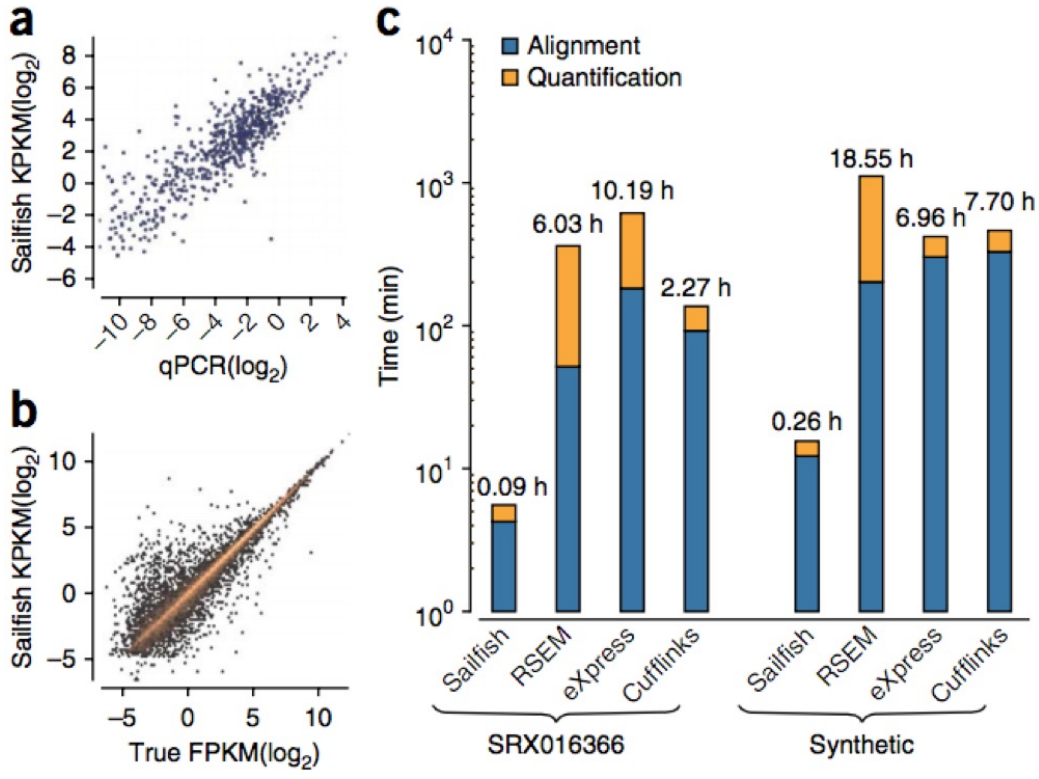


"By not requiring read mapping, **Sailfish avoids parameters specifying**, for example, the number of mismatches to tolerate, total allowable quality of mismatched bases, gap open and extension penalties, whether and how much to trim reads, number and quality of alignments to report from the aligner and pass into the estimation procedure."

Patro et al (2024) Nat Biotech, 32, 462–464.

RNA-Seq: Alignment/Pseudoalignment

Sailfish



Patro et al (2024) Nat Biotech, 32, 462–464.

RNA-Seq: Alignment/Pseudoalignment

Kallisto

BRIEF COMMUNICATIONS

nature biotechnology

Near-optimal probabilistic RNA-seq quantification

Nicolas L Bray[1], Harold Pimentel[2], Páll Melsted[3] & Lior Pachter[2,4,5]

We present kallisto, an RNA-seq quantification program that is two orders of magnitude faster than previous approaches and achieves similar accuracy. Kallisto pseudoaligns reads to a reference, producing a list of transcripts that are compatible with each read while avoiding alignment of individual bases. We use kallisto to analyze 30 million unaligned paired-end RNA-seq reads in <10 min on a standard laptop computer. This removes a major computational bottleneck in RNA-seq analysis.

this information, we develop a method based on pseudoalignment of reads and fragments, which focuses only on identifying the transcripts from which the reads could have originated and does not try to pinpoint exactly how the sequences of the reads and transcripts align.

A pseudoalignment of a read to a set of transcripts, T, is a subset, $S \subseteq T$, without specific coordinates mapping each base in the read to specific positions in each of the transcripts in S. Accurate pseudoalignments of reads to a transcriptome can be obtained using fast hashing of k-mers together with the transcriptome de Bruijn graph (T-DBG). de Bruijn graphs have been crucial for DNA and RNA assembly[8], where they are usually constructed from reads. Kallisto uses a T-DBG, which is a de Bruijn graph constructed from k-mers present in the transcriptome (**Fig. 1a**), and a path covering of the graph, a set of paths whose union covers all edges of the graph, where the paths correspond to transcripts (**Fig. 1b**). This path covering of a T-DBG induces multi-sets on the vertices, called k-compatibility classes. A compatibility class can be associated to an error-free read by

RNA-Seq: Pseudoalignment

De bruijin graph

create transcriptome de Bruijn graph (T-DBG)
using k-mers (31 nucleotides)

Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

create transcriptome de Bruijn graph (T-DBG)
using k-mers (31 nucleotides)

Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

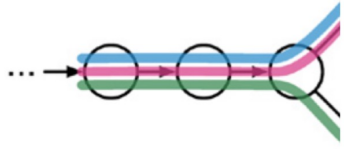
create transcriptome de Bruijn graph (T-DBG)
using k-mers (31 nucleotides)

Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

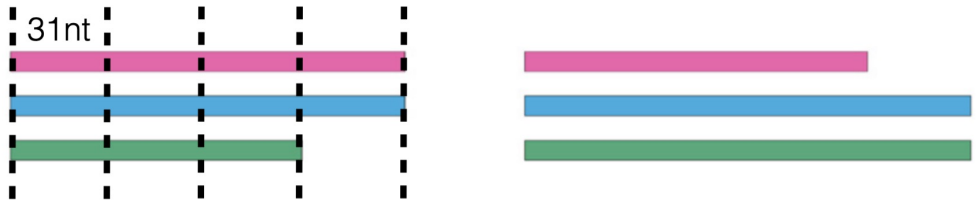
create transcriptome de Bruijn graph (T-DBG)
using k-mers (31 nucleotides)



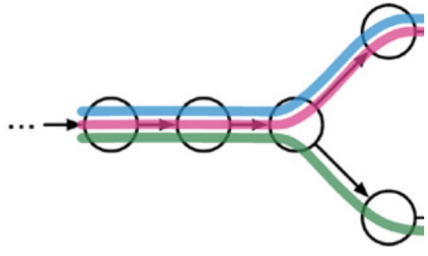
Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph



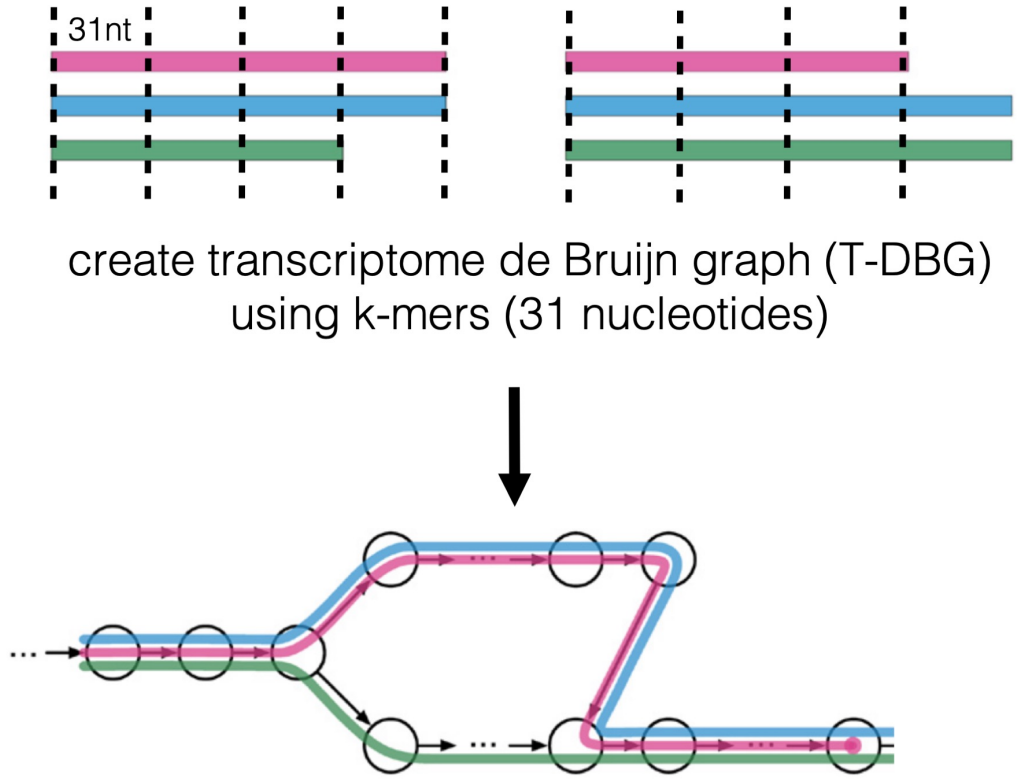
create transcriptome de Bruijn graph (T-DBG)
using k-mers (31 nucleotides)



Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

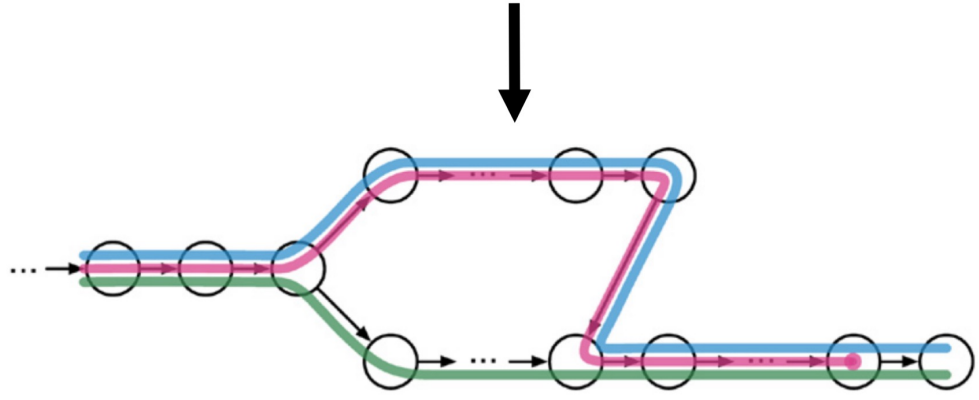


Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

create transcriptome de Bruijn graph (T-DBG)
using k-mers (31 nucleotides)

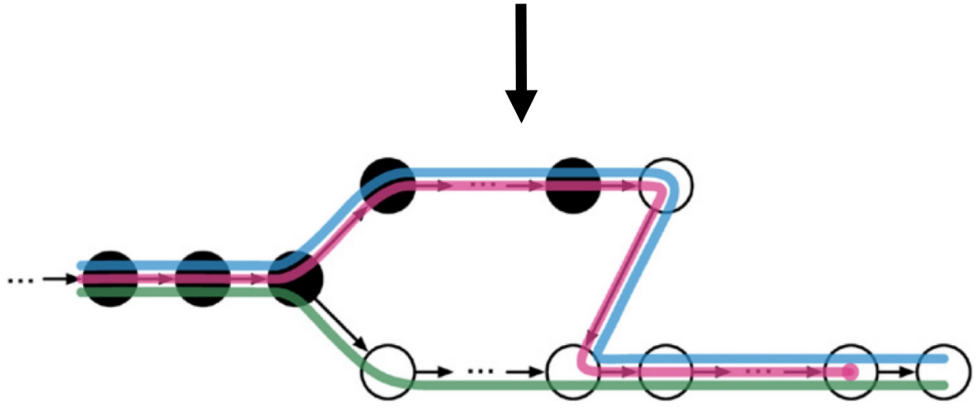


Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

Don’t align, instead find transcripts that are compatible with the read

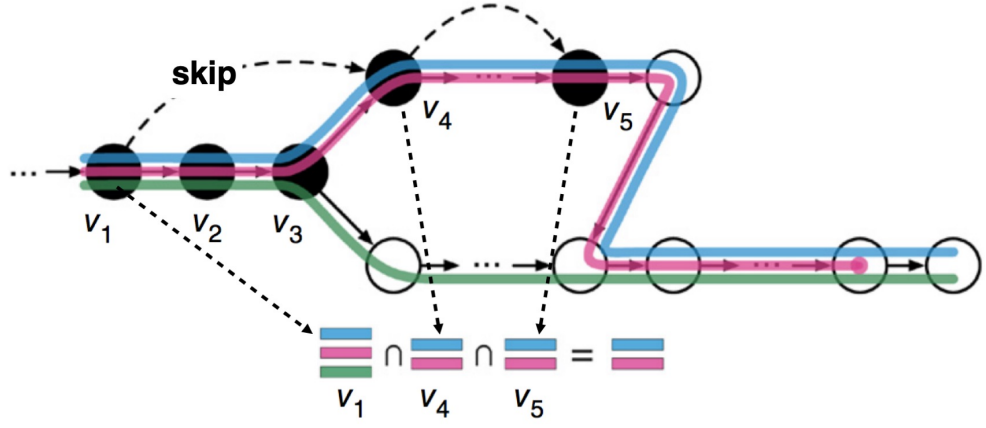


Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

Don’t even need to search for all possible K-mers

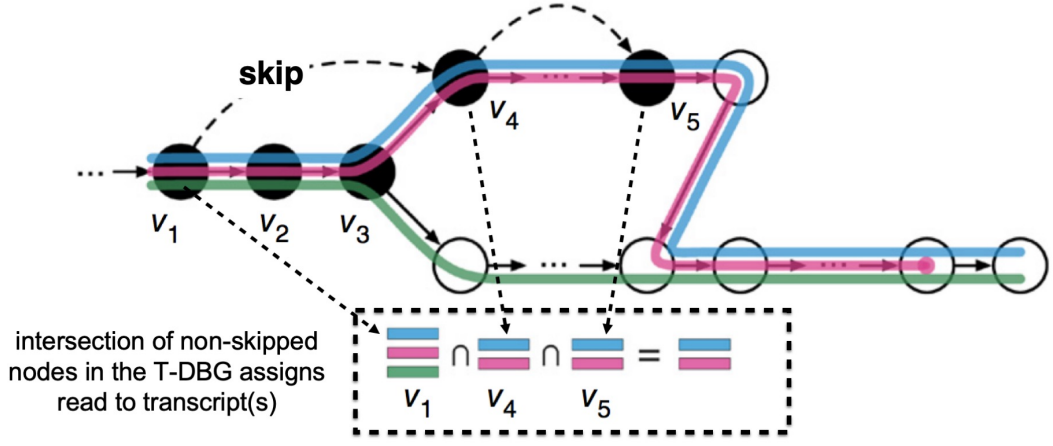


Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

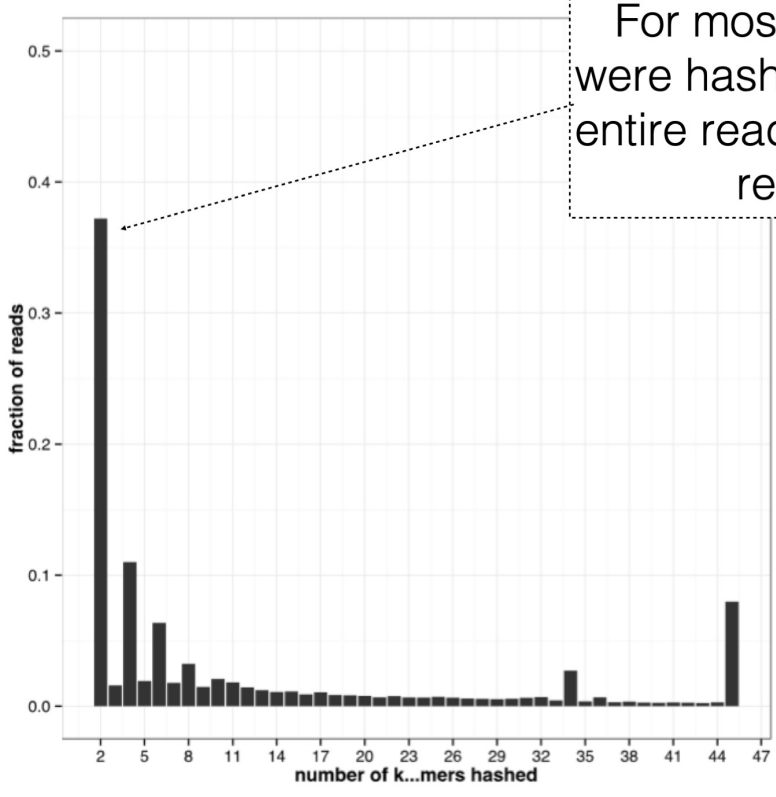
Don’t even need to search for all possible K-mers



Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

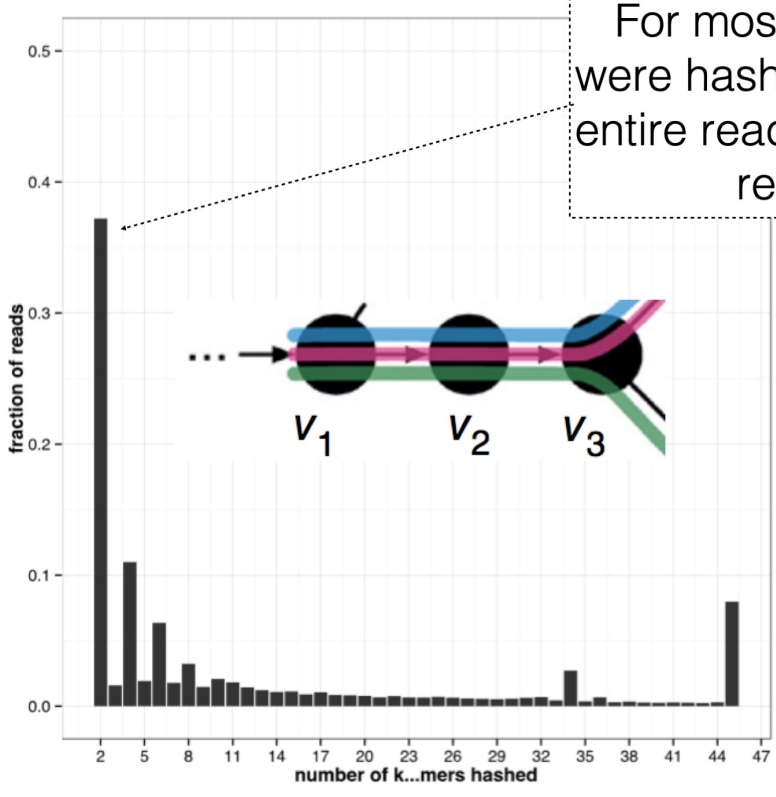


For most reads, only two k-mers were hashed b/c all k-mers from the entire read mapped to a contiguous region in the T-DBG

Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

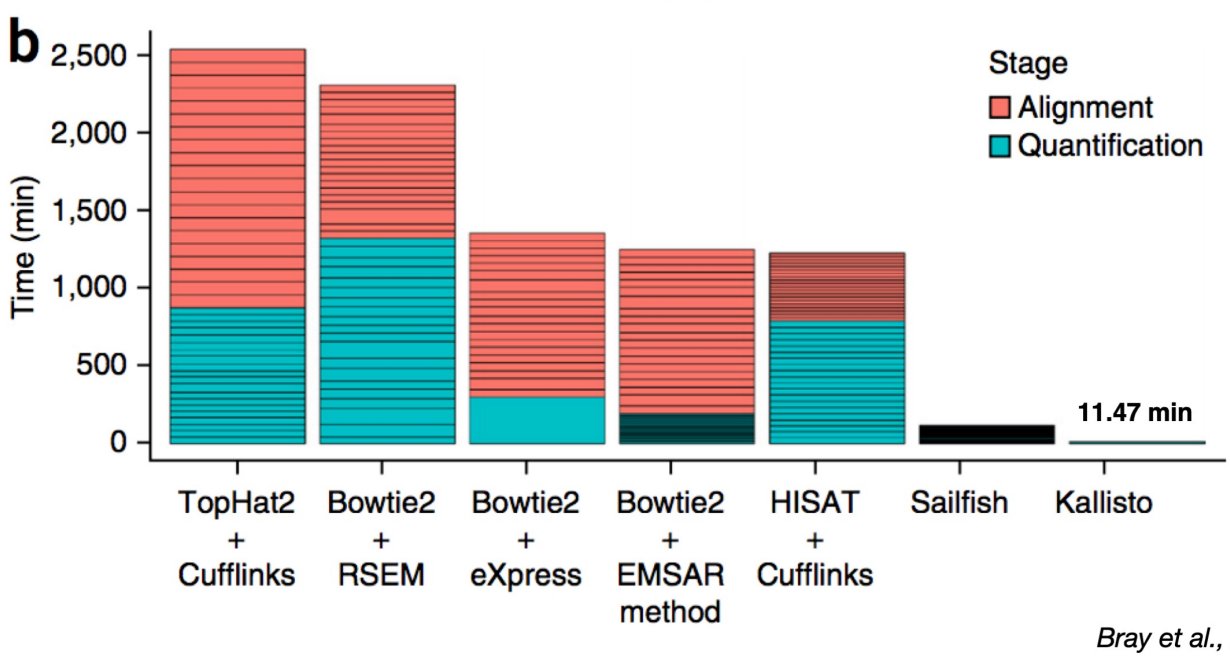


For most reads, only two k-mers were hashed b/c all k-mers from the entire read mapped to a contiguous region in the T-DBG

Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph



Bray et al., Nat. Biotech., 2016

RNA-Seq: Pseudoalignment

De bruijin graph

"The implication for biologists is that RNA-Seq analysis now becomes interactive. Instead of "freezing" an analysis that might take weeks or even months, data can be explored dynamically, e.g. easily quantified against different transcriptomes, or re-quantified as transcriptomes are updated. The ability to analyze data locally instead of requiring cloud computation means that analysis is portable, and also easily secure."

- Lior Pachter, Bits of DNA blog, 2015.

RNA-Seq: Pseudoalignment

De bruijin graph

Salmon provides fast and bias-aware quantification of transcript expression

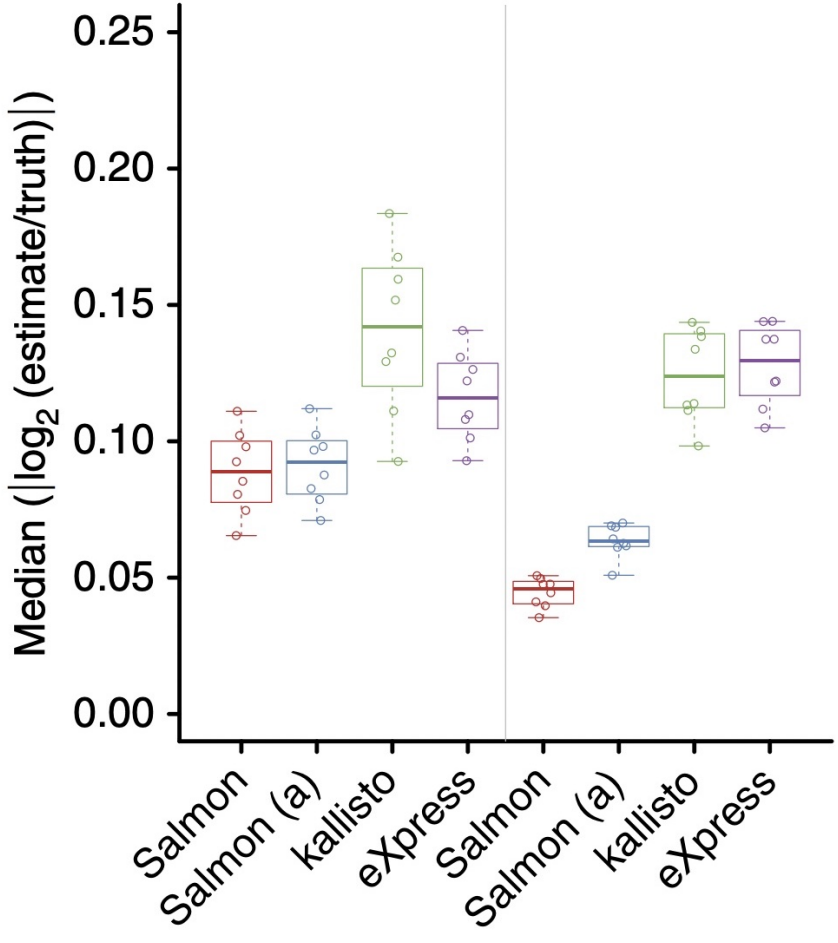
Rob Patro[1], Geet Duggal[2], Michael I Love[3,4], Rafael A Irizarry[3,4] & Carl Kingsford[5]

We introduce Salmon, a lightweight method for quantifying transcript abundance from RNA–seq reads. Salmon combines a new dual-phase parallel inference algorithm and feature-rich bias models with an ultra-fast read mapping procedure. It is the first transcriptome-wide quantifier to correct for fragment GC-content bias, which, as we demonstrate here, substantially improves the accuracy of abundance estimates and the sensitivity of subsequent differential expression analysis.

Our novel quantification procedure, Salmon, employs a new dual-phase statistical inference procedure and sample-specific bias models that account for sequence-specific, fragment GC-content, and positional biases. It achieves the same order-of-magnitude benefits in speed as kallisto and Sailfish but with greater accuracy. Salmon consists of three components: a lightweight mapping model, an online phase that estimates initial expression levels and model parameters, and an offline phase that refines expression estimates (**Supplementary Fig. 1**). This two-phase inference procedure allows Salmon to build a probabilistic model of the sequencing experiment that incorporates information—like terms contributing to the conditional probability of drawing a fragment of a given transcript—not considered by Sailfish[9] and kallisto[10]. Salmon is capable of either mapping sequencing reads itself by using a fast and lightweight procedure called quasi-mapping or accepting precomputed read alignments in the form of a SAM or BAM file. In this paper, we mainly employ Salmon's lightweight mapping mode, although we also test using precomputed alignments. Furthermore, Salmon provides

RNA-Seq: Pseudoalignment

De bruijin graph



RNA-Seq: Pseudoalignment

De bruijin graph

- Sailfish https://www.ncbi.nlm.nih.gov/pubmed/24752080
  "Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms." 2014
- RNA-Skim https://www.ncbi.nlm.nih.gov/pubmed/24931995
  "RNA-Skim: a rapid method for RNA-Seq quantification at transcript level." 2014
- Kallisto https://www.ncbi.nlm.nih.gov/pubmed/27043002
  "Near-optimal probabilistic RNA-seq quantification." 2016
- Salmon https://www.ncbi.nlm.nih.gov/pubmed/28263959
  "Salmon provides fast and bias-aware quantification of transcript expression." 2017

RNA-Seq: Alignment/Pseudoalignment

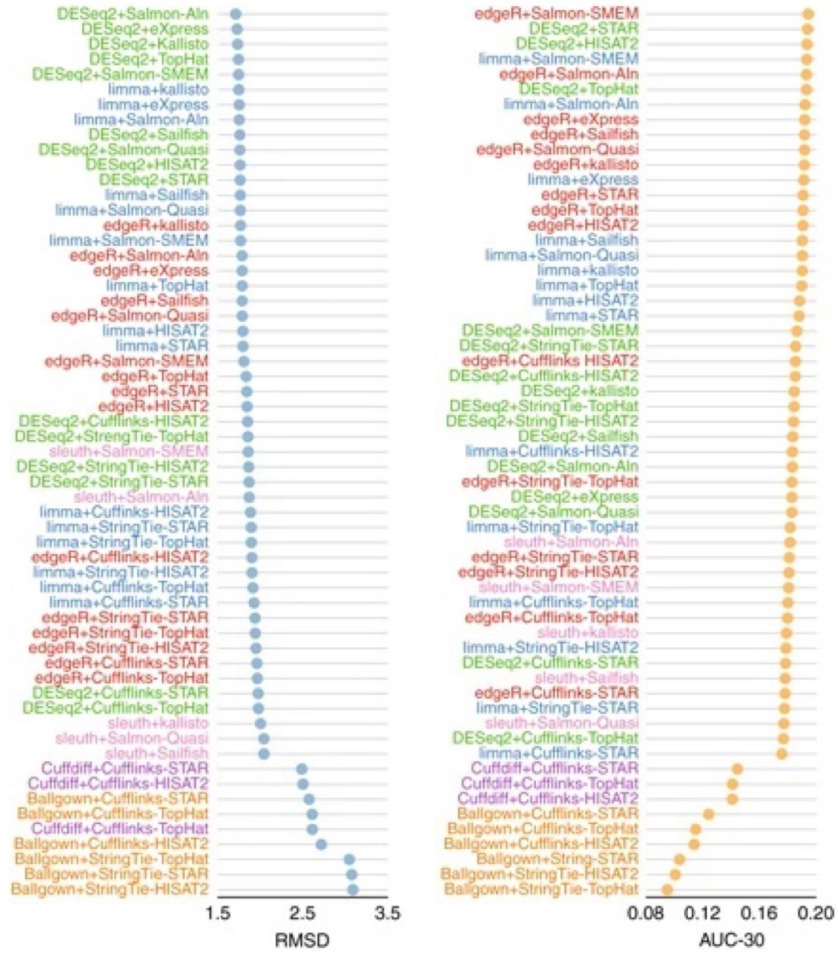
Comparison

"Unlike other studies, our work not only compared various differential analysis tools, but also studied the impact of different alignment-based and alignment-free approaches on the accuracy of differential analysis. Overall, alignment-free techniques such as Salmon or kallisto were observed to be capable of delivering high-quality predictions."

"Alignment-free tools like Salmon-SMEM and kallisto yielded the most consistent and accurate quantifications, and, thus, can serve as the most accurate yet computationally cheap solutions if isoform discovery is not important."

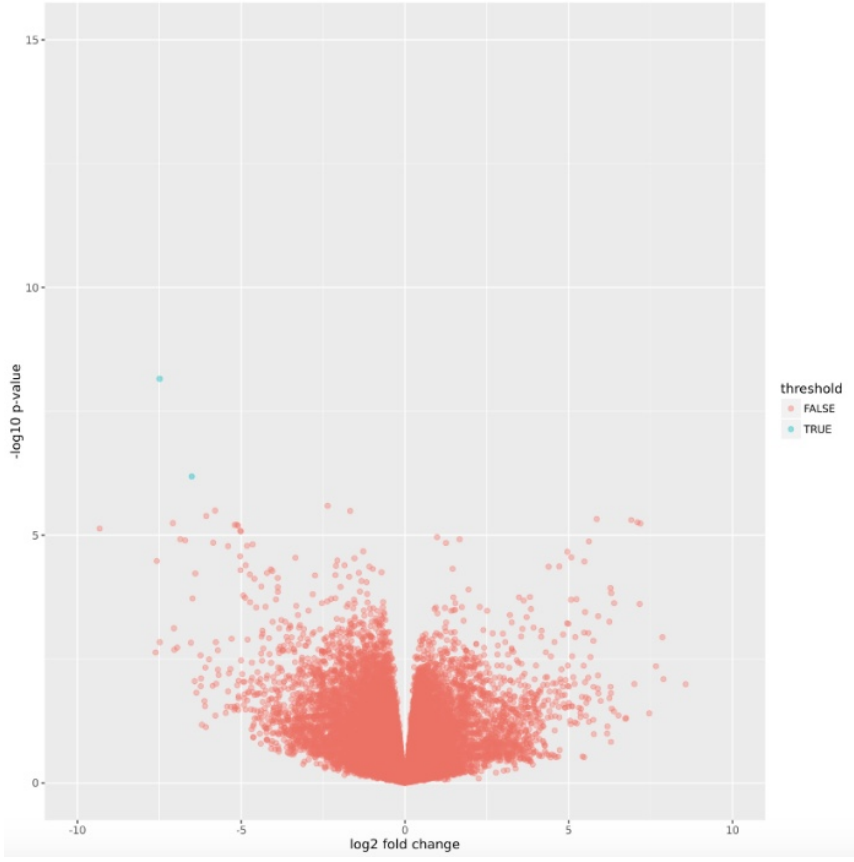
traditional vs. alignment-free

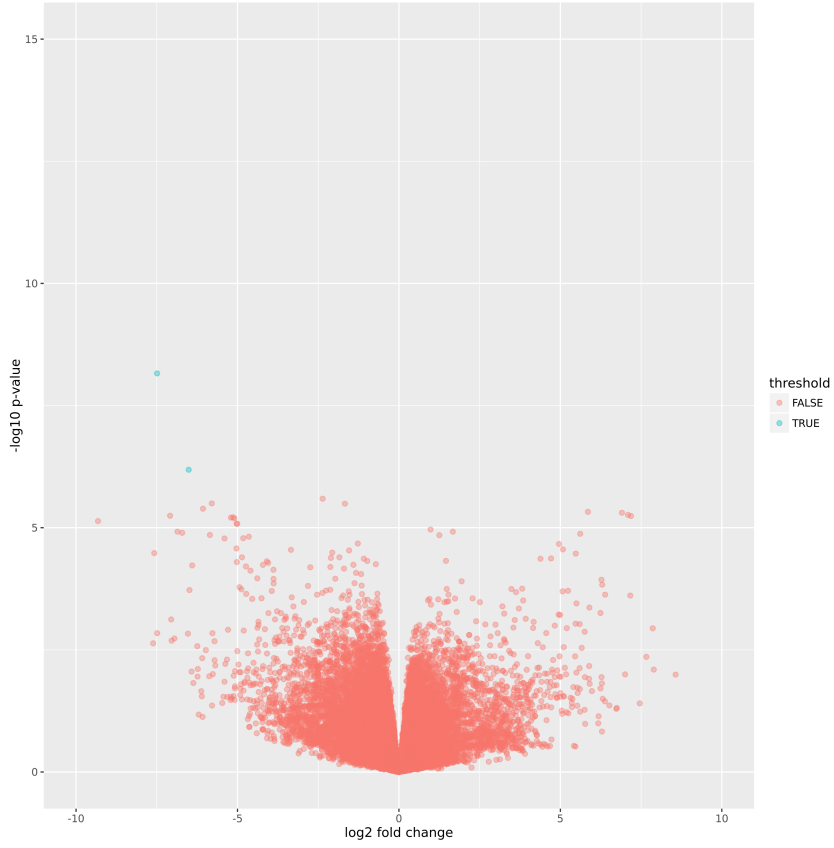
Sahraeian et al., Nature Communications, 2017



RNA-Seq: Alignment/Pseudoalignment

What is the best one?

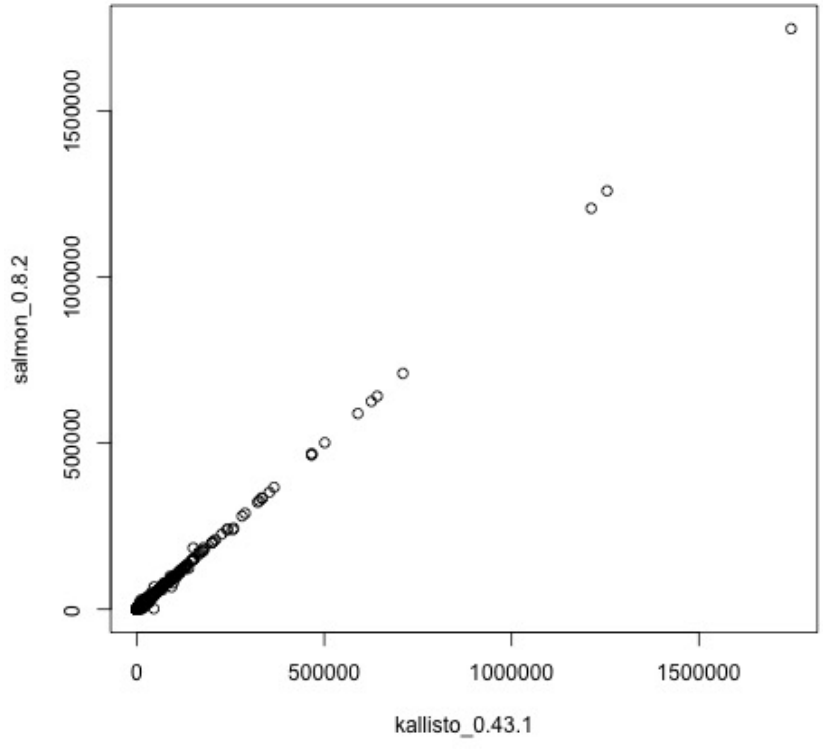




https://liorpachter.wordpress.com/2017/08/02/how-not-to-perform-a-differential-expression-analysis-or-science/

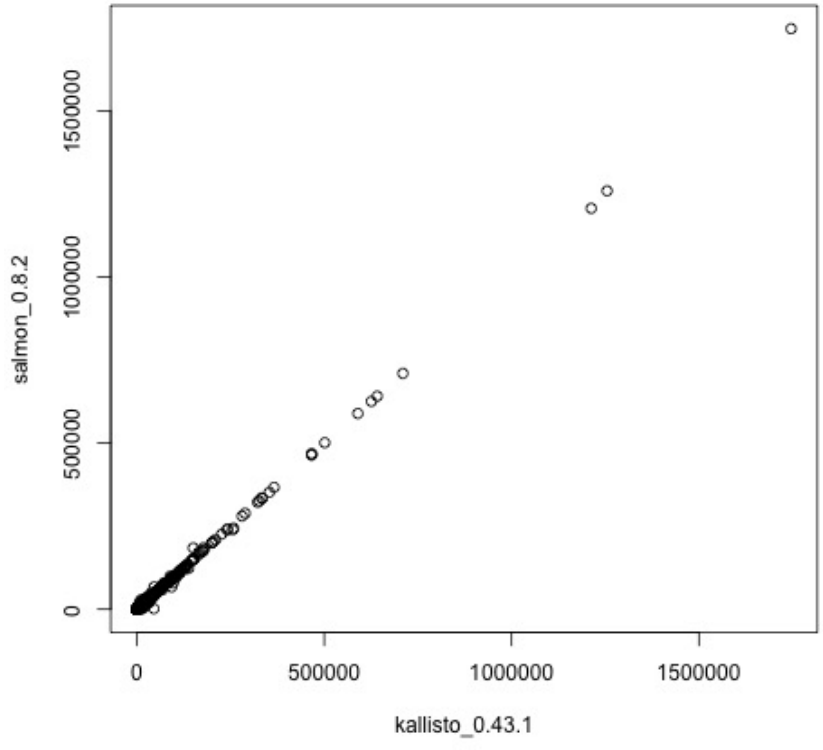
RNA-Seq: Alignment/Pseudoalignment

What is the best one?



RNA-Seq: Alignment/Pseudoalignment

What is the best one?



Pearson correlation coefficient is 0.9996026

RNA-Seq: Alignment/Pseudoalignment

What is the best one?

Various sources suggest that Salmon, Kallisto, and Sailfish results are quite comparable.

Usability, documentation, and supporting downstream tools could be used to decide

RNA-Seq: Alignment/Pseudoalignment

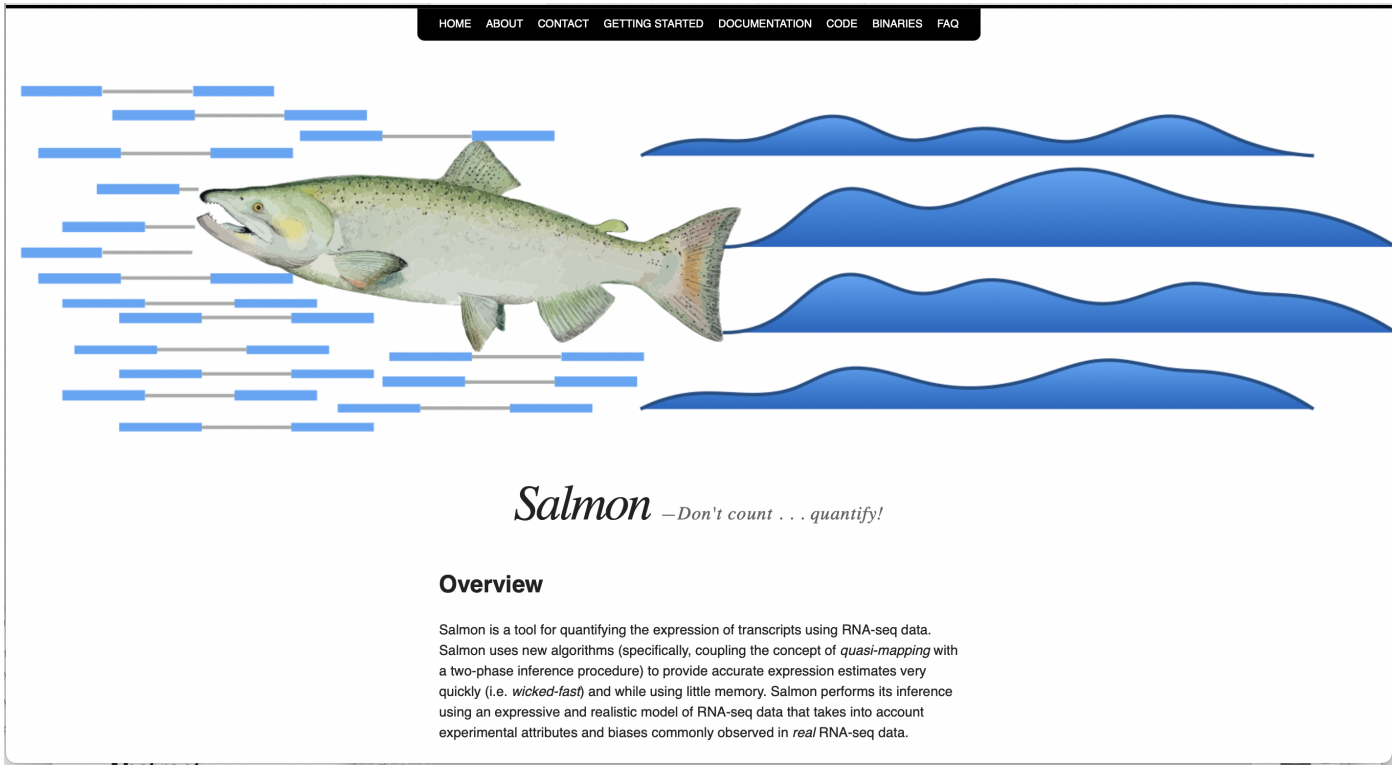
Very fast, BUT...

> If you're interested in sequence variants (SNPs)
> If you're interested in splice forms

You'll need alignments

RNA-Seq: Transcript Quantification

Salmon



https://combine-lab.github.io/salmon/

RNA-Seq: Transcript Quantification

Salmon

> Salmon is a tool for **quantifying the expression** of transcripts using **RNA-seq** data. Salmon uses new algorithms (specifically, coupling the concept of quasi-mapping with a two-phase inference procedure) to provide accurate expression estimates very quickly **(i.e. wicked-fast)** and while using little memory. Salmon performs its inference using an expressive and realistic model of RNA-seq data that takes into account experimental attributes and biases commonly observed in real RNA-seq data

RNA-Seq: Transcript Quantification

Salmon

The mapping-based mode of Salmon runs in two phases:

- indexing
- quantification

https://salmon.readthedocs.io/en/latest/salmon.html

RNA-Seq: Transcript Quantification

Salmon

- Indexing step: independent of the reads, and only needs to be run once for a particular set of reference transcripts.
- Quantification step: specific to the set of RNA-seq reads and is thus run more frequently.

RNA-Seq: Transcript Quantification

Running Salmon

1. Navigate to the `00-Databases` folder

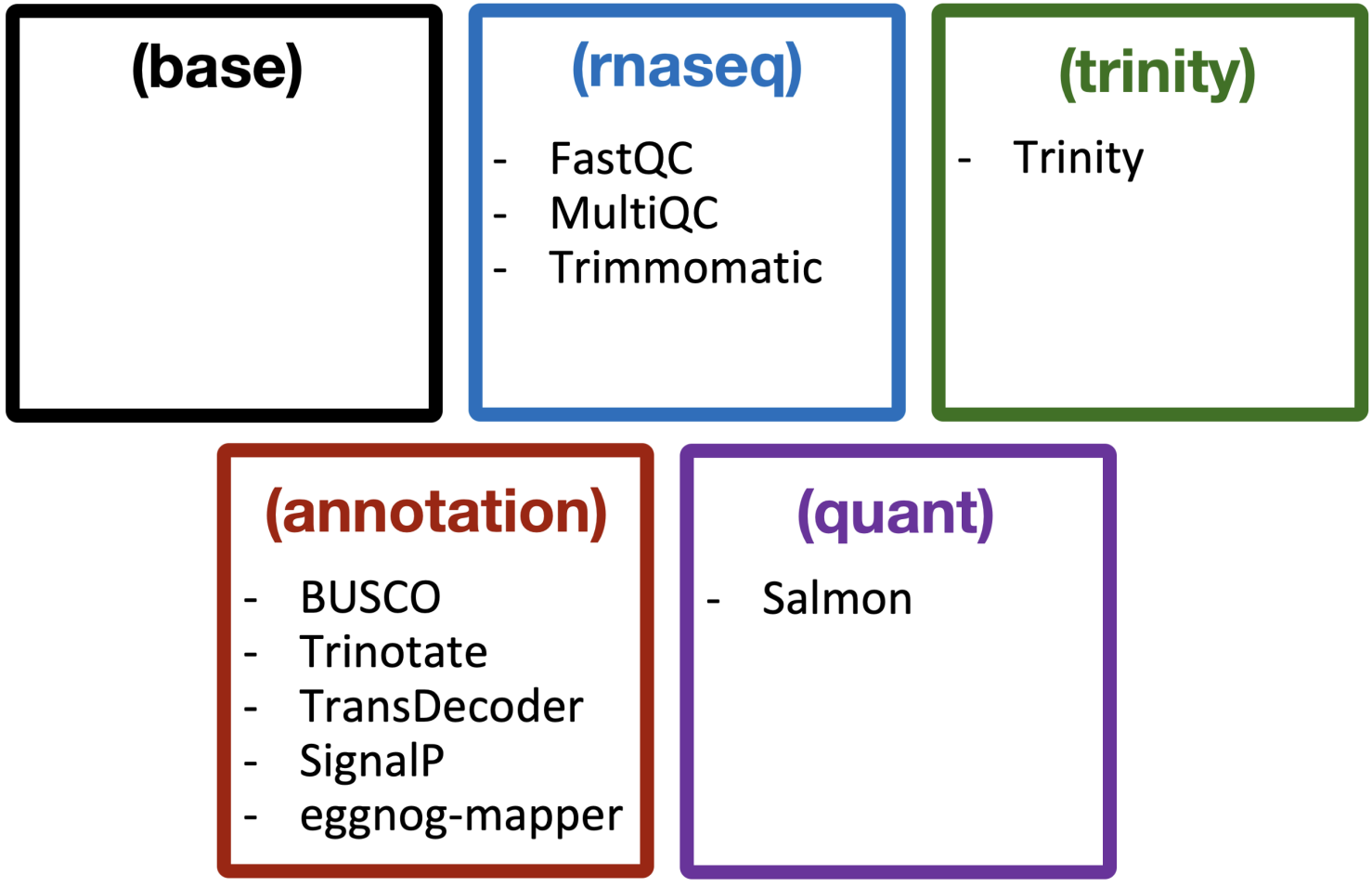
```
cd ~/rnaseq/00-Databases
```

2. Check if you have the reference databases of the mitochondrial genomes. If you haven't done it yet, download databases from course website

```
ls -la
```

RNA-Seq: Transcript Quantification

Conda environments



RNA-Seq: Transcript Quantification

Running Salmon

3. Activate 'quant' environment

```
conda activate quant
```

4. Call salmon to read help

```
salmon index -h
salmon quant -h
salmon quant --help-reads
```

RNA-Seq: Transcript Quantification

Running Salmon

5. Create reference index (example, **do not run!**)

```
# Basename
SPECIES="Chom"

#salmon
salmon index \
-t ~/rnaseq/04-Annotation/${SPECIES}/transdecoder/longest_orfs.cds \
-i ~/rnaseq/00-Databases/${SPECIES}_orfs_index
```

➡️ -t: Transcriptome file

➡️ -o: Output folder for index

RNA-Seq: Transcript Quantification

Running Salmon

5. Create reference index for mtDNA CDSs

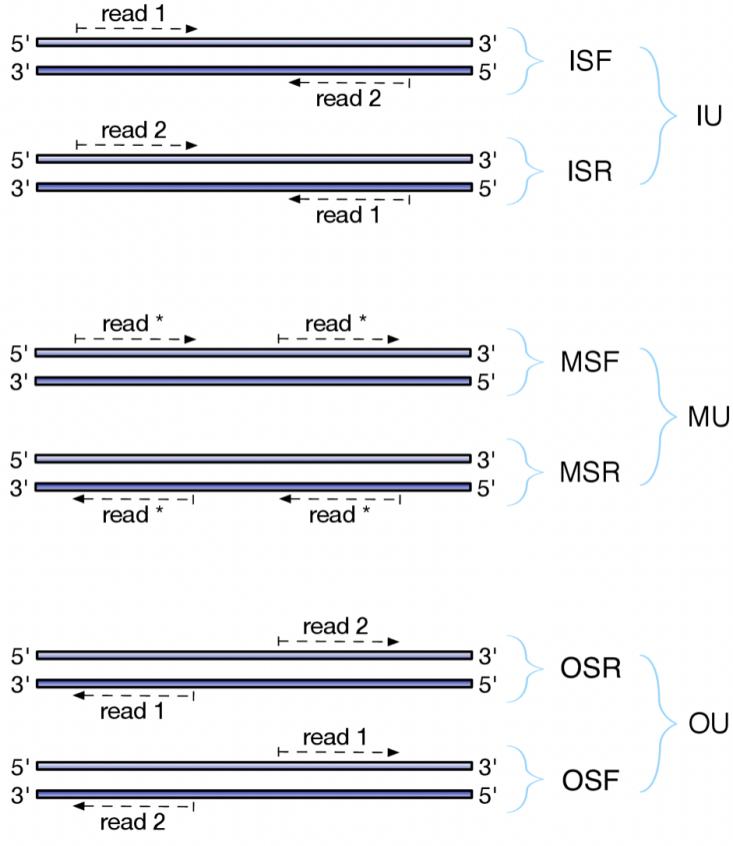
```
# Basename
SPECIES="Chom"


salmon index -t ${SPECIES}_mtDNAgenes.fasta \
             -i ${SPECIES}_mtDNAgenes_index
```

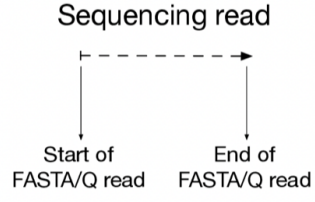
- -t: Transcriptome file
- -o: Output folder for index

RNA-Seq: Transcript Quantification

Running Salmon

Fragment Library Types





RNA-Seq: Transcript Quantification

Running Salmon

Fragment Library Types

> The first part of the library string (relative orientation) is only provided if the library is paired-end. The possible options are:

```
I = inward
O = outward
M = matching
```

> The second part of the read library string specifies whether the protocol is stranded or unstranded; the options are:

```
S = stranded
U = unstranded
```

RNA-Seq: Transcript Quantification

Running Salmon

6. Create and navigate to the folder `05-Mapping`

```
mkdir ~/rnaseq/05-Mapping
cd ~/rnaseq/05-Mapping
```

RNA-Seq: Transcript Quantification

Running Salmon: single-end

7. Salmon for quantification (do not run)

```
salmon quant -i <transcriptome_index> \
             -l <library_type> \
             -r <single_end_reads.fastq> \
             --validateMappings \
             -o <output_directory>
```

- -i: salmon index
- -l: string describing the library type
- -1: file containing the #1 mates
- -2: file containing the #2 mates
- -o: output quantification directory.

RNA-Seq: Transcript Quantification

Running Salmon: paired-end

7. Salmon for quantification (do not run)

```
salmon quant -i <transcriptome_index> \
             -l <library_type> \
             -1 <R1_reads.fastq> \
             -2 <R2_reads.fastq> \
             --validateMappings \
             -o <output_directory>
```

- -i: salmon index
- -l: string describing the library type
- -1: file containing the #1 mates
- -2: file containing the #2 mates
- -o: output quantification directory.

RNA-Seq: Transcript Quantification

Running Salmon: paired-end

7. Run salmon for quantification

```
salmon quant \
       -i ~/rnaseq/00-Databases/${SPECIES}_mtDNAgenes_index \
       -l A \
       -r ~/rnaseq/01-RawReads/${SPECIES}${SAMPLE}_R1.fastq.gz \
       -o ~/rnaseq/05-Mapping/${SPECIES}${SAMPLE}_mtDNA-salmon
```

RNA-Seq: Transcript Quantification

Running Salmon

Quantification File (Trinity contigs)

Name	Length	EffectiveLength	TPM	NumReads
TRINITY_DN6_c0_g1_i27.p3	327	153.671	2171.851997	9088.233
TRINITY_DN6_c0_g1_i28.p1	1131	951.945	5213.066502	135133.017
TRINITY_DN6_c0_g1_i28.p3	303	131.788	6872.061058	24661.403
TRINITY_DN6_c0_g1_i29.p1	1131	951.945	173.575637	4499.425
TRINITY_DN6_c0_g1_i29.p2	414	236.689	0.000000	0.000
TRINITY_DN6_c0_g1_i29.p3	390	213.411	0.000000	0.000
TRINITY_DN6_c0_g1_i3.p2	447	269.022	1176.557572	8619.020
TRINITY_DN6_c0_g1_i3.p3	360	184.654	9.118244	45.849
TRINITY_DN6_c0_g1_i6.p1	1131	951.945	870.636106	22568.614

RNA-Seq: Transcript Quantification

Running Salmon

Quantification File (mtDNA genes)

Name	Length	EffectiveLength	TPM	NumReads
ND2	1017	840.387	19495.097280	94155.999
COX1	1539	1362.387	191863.560563	1502229.185
COX2	688	511.403	188051.013249	552690.368
ATP8	165	27.164	1651.052638	257.753
ATP6	678	501.404	81711.503320	235458.584
COX3	789	612.391	264354.551853	930375.771
ND3	357	183.851	52965.972446	55963.632
ND5	1720	1543.387	7352.180412	65213.001
ND4	1339	1162.387	29129.181680	194590.775

RNA-Seq: Organizing data

Summarizing outputs

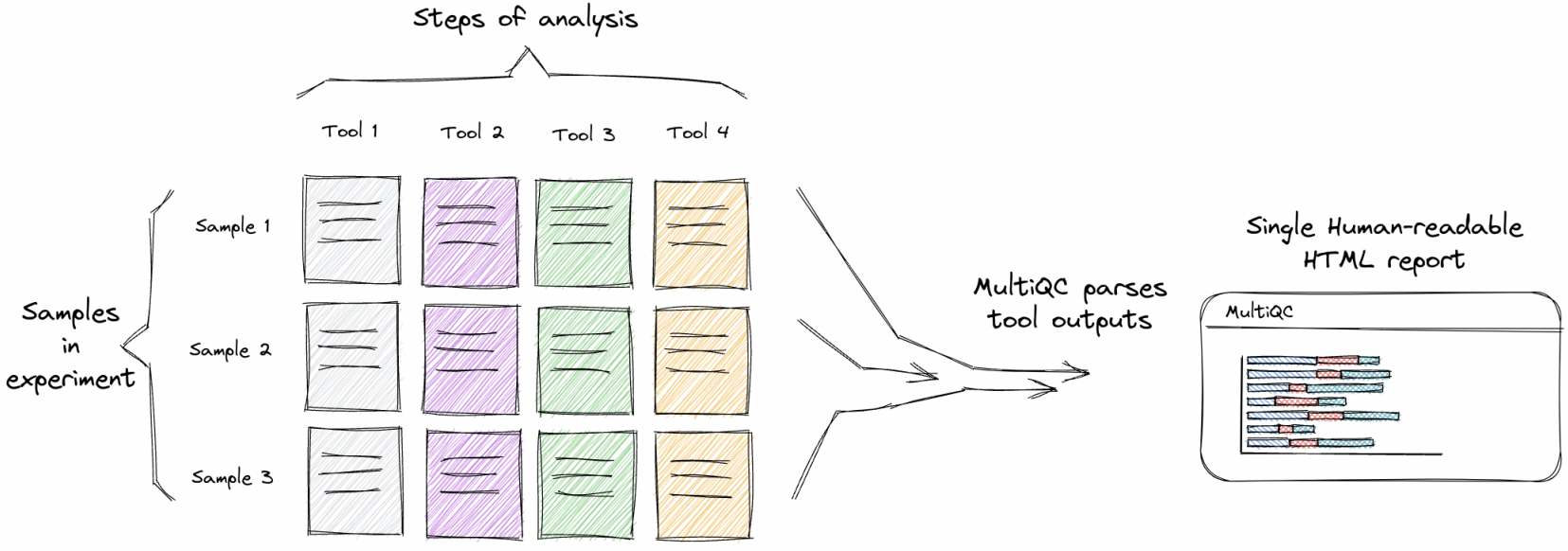
Many different files:

- Fastq (R1 and R2) for each replicate
- Trimmed fastq (R1 and R2) for each replicate
- Fastqc reports
- Salmon quatification files

RNA-Seq: Organizing data

Summarizing output

MultiQC



RNA-Seq: Organizing data

Summarizing output

MultiQC

MultiQC aggregates results from bioinformatics analyses across many samples into a single report.

It searches a given directory for analysis logs and compiles an HTML report. It's a general use tool, perfect for summarising the output from numerous bioinformatics tools.

RNA-Seq: Organizing data

Summarizing output with MultiQC

1. Activate the `rnaseq` environment

```
conda activate rnaseq
```

2. See multiqc help

```
multiqc --help
```

3. Go to the folder `~/rnaseq/` and run MultiQC

```
cd ~/rnaseq/
multiqc .
```

RNA-Seq: Organizing data

Summarizing output

MultiQC

General Statistics

Copy table | Configure columns | Scatter plot | Violin plot | Showing $^{0}/_{100}$ rows and $^{8}/_{11}$ columns. | Export as CSV

Sample Name	% Aligned	M Aligned	CFR	M Bias	Dups	GC	Median len	Seqs
CalbF_1_mtDNA-salmon	4.9 %	0.0 M	100.0 %	0.5				
CalbF_2_mtDNA-salmon	6.9 %	0.1 M	100.0 %	0.5				
CalbF_4_mtDNA-salmon	8.8 %	0.1 M	100.0 %	0.5				
CalbL_1_mtDNA-salmon	2.4 %	0.0 M	100.0 %	0.5				
CalbL_2_mtDNA-salmon	2.4 %	0.0 M	100.0 %	0.5				
Calb_R1					22.4 %	37.0 %	101 bp	0.1 M
Calb_R1_paired					22.6 %	37.0 %	100 bp	0.1 M
Calb_R2					22.0 %	36.0 %	101 bp	0.1 M
Calb_R2_paired					22.6 %	36.0 %	100 bp	0.1 M
CalbF_1_R1					74.0 %	37.0 %	101 bp	35.9 M
CalbF_1_R2					72.7 %	36.0 %	101 bp	35.9 M
CalbF_4_R1					74.8 %	37.0 %	100 bp	21.3 M
CalbF_4_R2					72.5 %	36.0 %	100 bp	21.3 M
CalbL_1_R1					82.0 %	41.0 %	101 bp	33.3 M
CalbL_1_R2					80.8 %	40.0 %	101 bp	33.3 M

RNA-Seq: Automating analyses

Automate using shell scripts

Running BUSCO for many samples

```
# Species list
ALLSPECIES=("Cmac" "Cmeg" "Lexi" "Calb")

# Loop
for SPECIES in "${ALLSPECIES[@]}"; do
  echo "Processing sample: ${SPECIES}"

  # busco command
  busco -i ${SPECIES}.Trinity.fasta -o ${SPECIES}_busco -

done
```

RNA-Seq: Automating analyses

Automate using shell scripts

Running Salmon for many samples

```
#!/bin/bash

# Defining species
SPECIES="Chom"

# List of samples
SAMPLES=("F_1" "F_3" "L_1" "L_3")

# Paths
INDEX_DIR="~/rnaseq/00-Databases/${SPECIES}_mtDNAgenes"
READS_DIR="~/rnaseq/02-FilteredReads/${SPECIES}"
OUTPUT_DIR="~/rnaseq/05-Mapping/${SPECIES}"

#continues
```

RNA-Seq: Automating analyses

Automate using shell scripts

Running Salmon for many samples

```
# Loop
for SAMPLE in "${SAMPLES[@]}"; do
  echo "Processing sample: ${SAMPLE}"

  # Running salmon
  salmon quant \
    -i "${INDEX_DIR}" \
    -l A \
    -r "${READS_DIR}/${SPECIES}${SAMPLE}_R1.fastq" \
    -o "${OUTPUT_DIR}/${SPECIES}${SAMPLE}_mtDNA-salmon"
done

echo "Done!"
```

RNA-Seq: Automating analyses

Automate using shell scripts

Shell script: text file listing all commands

runBusco.sh

```
#!/bin/bash

# Species list
ALLSPECIES=("Cmac" "Cmeg" "Lexi" "Calb")

# Loop
for SPECIES in "${ALLSPECIES[@]}"; do
  echo "Processing sample: ${SPECIES}"

  # busco command
  busco -i ${SPECIES}.Trinity.fasta -o ${SPECIES}_busco -

done
```

RNA-Seq: Automating analyses

Automate using shell scripts

- Save shell script as `runBusco.sh` or `runSalmon.sh`
- Change file permissions:

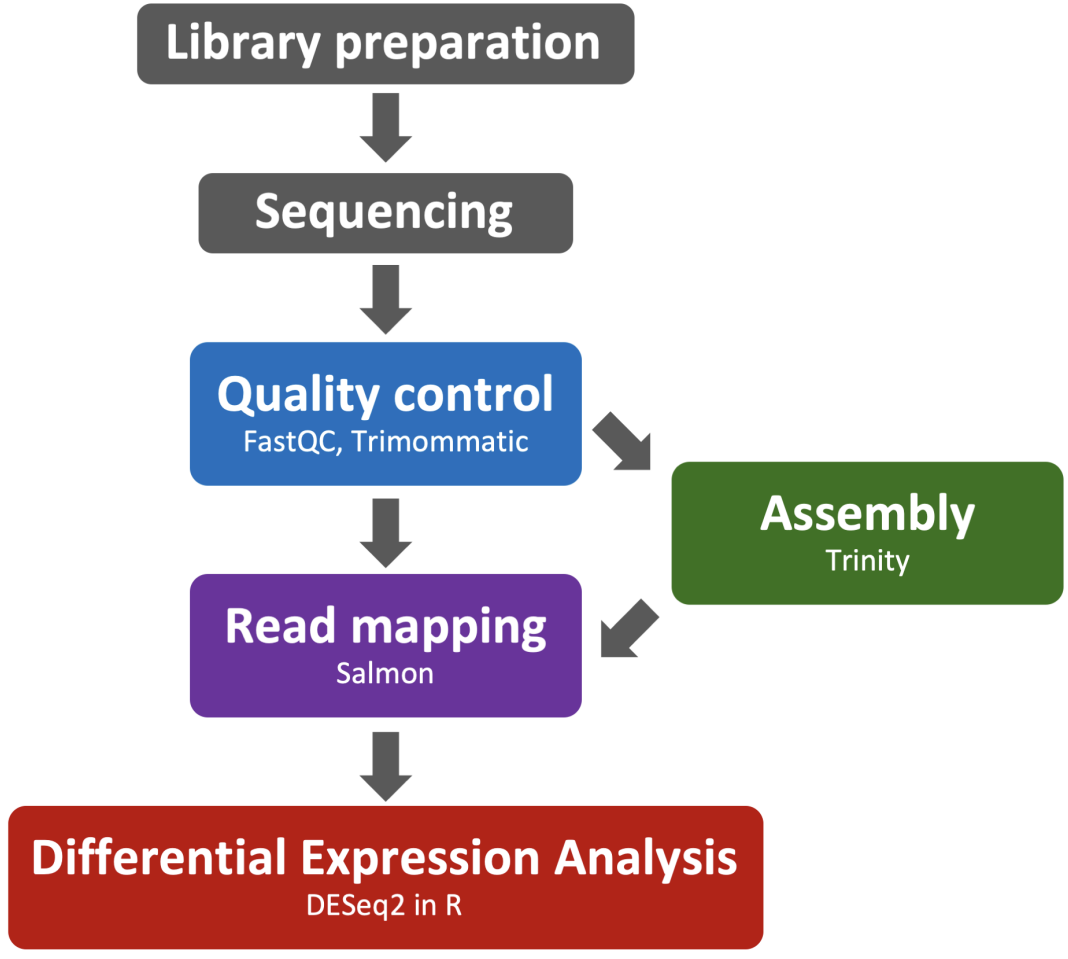
```
chmod +x runSalmon.sh
```

- Run script

```
~/rnaseq/runSalmon.sh
```

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis



RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

Quantification File

Name	Length	EffectiveLength	TPM	NumReads
ND2	1017	840.387	19495.097280	94155.999
COX1	1539	1362.387	191863.560563	1502229.185
COX2	688	511.403	188051.013249	552690.368
ATP8	165	27.164	1651.052638	257.753
ATP6	678	501.404	81711.503320	235458.584
COX3	789	612.391	264354.551853	930375.771
ND3	357	183.851	52965.972446	55963.632
ND5	1720	1543.387	7352.180412	65213.001
ND4	1339	1162.387	29129.181680	194590.775

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

Effective transcript length

> The Effective Length in Salmon output represents an adjusted length of a transcript, which accounts for the fragment length distribution of the sequencing library. It's used in the process of estimating transcript abundance more accurately.

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

Effective transcript length: how it works

1. Raw Transcript Length: This is the actual length of the transcript, as determined from the reference genome or transcriptome.
2. Adjustment for Fragment Length:
    - In sequencing, we expect more fragments to come from longer transcripts simply because they are longer and can generate more fragments.
    - For shorter transcripts, the inability to generate fragments from every position reduces their effective length, making it less likely for reads to map to them, even if they are expressed at similar levels.

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

Correcting Bias:

- Effective Length adjusts the raw length to reflect how likely a transcript is to generate fragments, given the fragment length distribution.
- This correction helps to reduce bias against short transcripts during quantification.

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

The **Effective Length** (L_{eff}) is often calculated as:

$$L_{\text{eff}} = L_{\text{raw}} - \mu + 1$$

Where:

- L_{raw} is the raw transcript length.
- μ is the mean fragment length.

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

The **Effective Length** (L_{eff}) is often calculated as:

$$L_{\text{eff}} = L_{\text{raw}} - \mu + 1$$

Where:

- L_{raw} is the raw transcript length.
- μ is the mean fragment length.

transcript	ATGCGTAACATG	L_{raw} = 12
fragment	NNN	μ = 3

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

The **Effective Length** (L_{eff}) is often calculated as:

$$L_{\text{eff}} = L_{\text{raw}} - \mu + 1$$

Where:

- L_{raw} is the raw transcript length.
- μ is the mean fragment length.

transcript	**ATGCGTAACATG**	$L_{raw} = 12$
fragment	**NNN**	$\mu = 3$

> Effective length (L_{eff}) = 10

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

```
ATGCGTAACATG
ATG
 TGC
  GCG
   CGT
    GTA
     TAA
      AAC
       ACA
        CAT
         ATG
```

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

The **Effective Length** (L_{eff}) is often calculated as:

$$L_{\text{eff}} = L_{\text{raw}} - \mu + 1$$

Where:

- L_{raw} is the raw transcript length.
- μ is the mean fragment length.

transcript	**ATGCGTAACATG**	$L_{raw} = 12$
fragment	**NNN**	$\mu = 3$

> Effective length (L_{eff}) = 10

RNA-Seq: Transcript Quantification

Preparing for Diferential Expression Analysis

The Effective Length is used to:

1. Normalize read counts for transcript abundance estimation.
2. Correct for biases introduced by sequencing and transcript properties.
3. Calculate the Transcripts Per Million (TPM) values, which depend on both the number of reads mapping to a transcript and its Effective Length.

❗ If you are analyzing Salmon output, understanding the Effective Length ensures you interpret abundance values correctly and account for transcript-specific biases.

RNA-Seq: Transcript Quantification

Normalization

> Various techniques used to adjust RNA-seq data in order to correct for technical biases, such as differences in sequencing depth, gene length, and sample composition, thereby enhancing the accuracy and reliability of detecting differentially expressed genes

RNA-Seq: Transcript Quantification

Normalization

> Various techniques used to adjust RNA-seq data in order to correct for technical biases, such as differences in sequencing depth, gene length, and sample composition, thereby enhancing the accuracy and reliability of detecting differentially expressed genes

⚠️ RNA-Seq quantification is relative

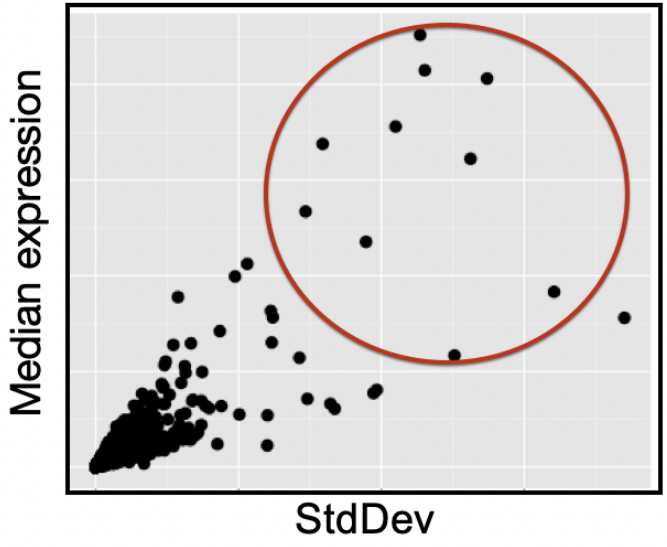
RNA-Seq: Transcript Quantification

Normalization

Gene	Fly 1	Fly 2	Fly 3	Fly 4	Fly 5	StDev
Gene A	5	10	10	15	10	3.5
Gene B	115	110	100	115	118	7.1
Gene C	1000	1100	1050	1045	1030	36.4
Gene D	8000	9000	10000	6000	7030	1576.4

RNA-Seq: Transcript Quantification

Normalization



Genes with higher expression levels have higher standart deviation

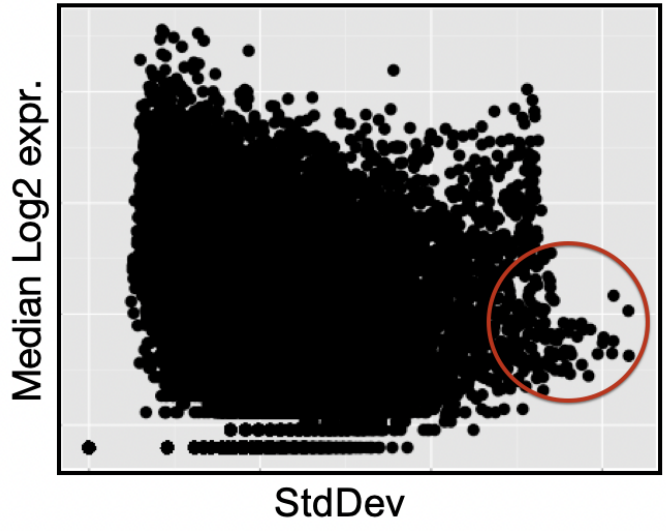
RNA-Seq: Transcript Quantification

Log 2 transformation

Gene	Fly 1	Fly 2	Fly 3	Fly 4	Fly 5	StDev
Gene A	2.3	3.3	3.3	3.9	3.3	0.57
Gene B	6.8	6.8	6.6	6.8	6.9	0.09
Gene C	10.0	10.1	10.0	10.0	10.0	0.05
Gene D	13.0	13.1	13.3	12.6	12.8	0.29

RNA-Seq: Transcript Quantification

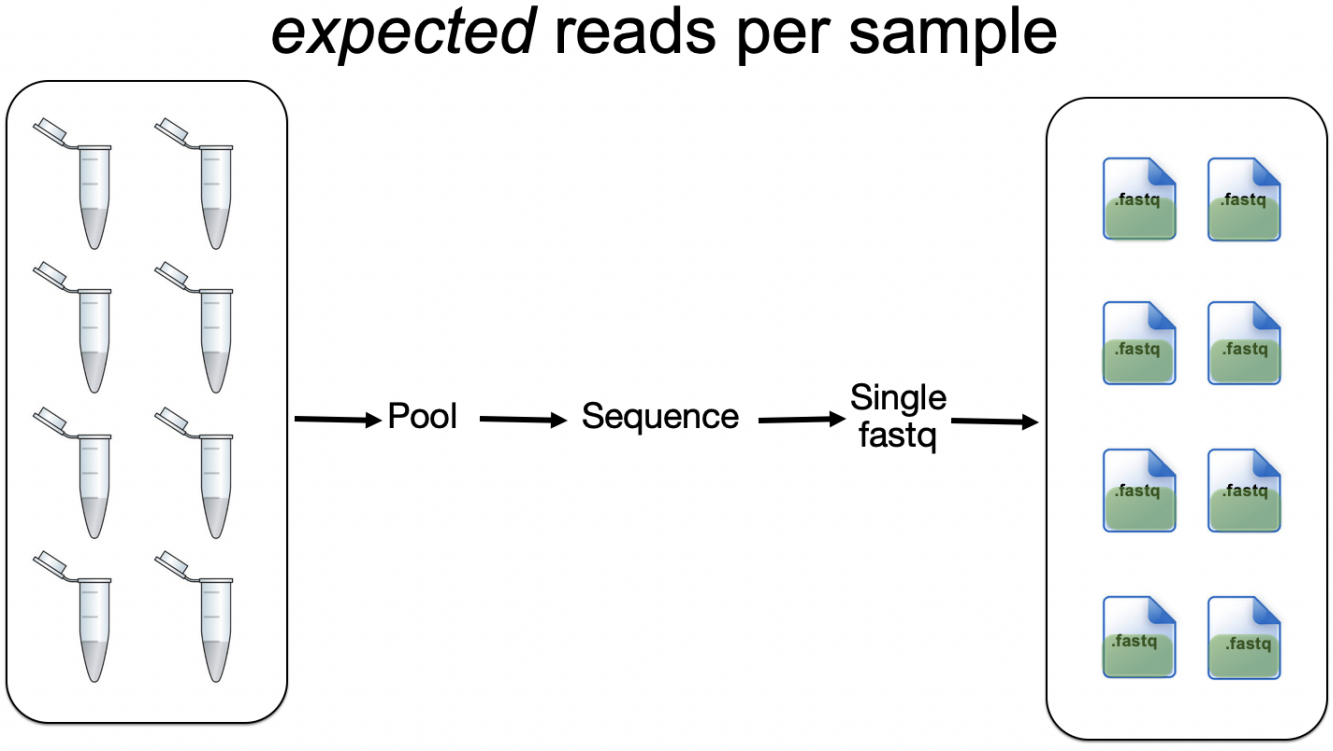
Normalization



Now, genes with lower expression levels have higher standart deviation

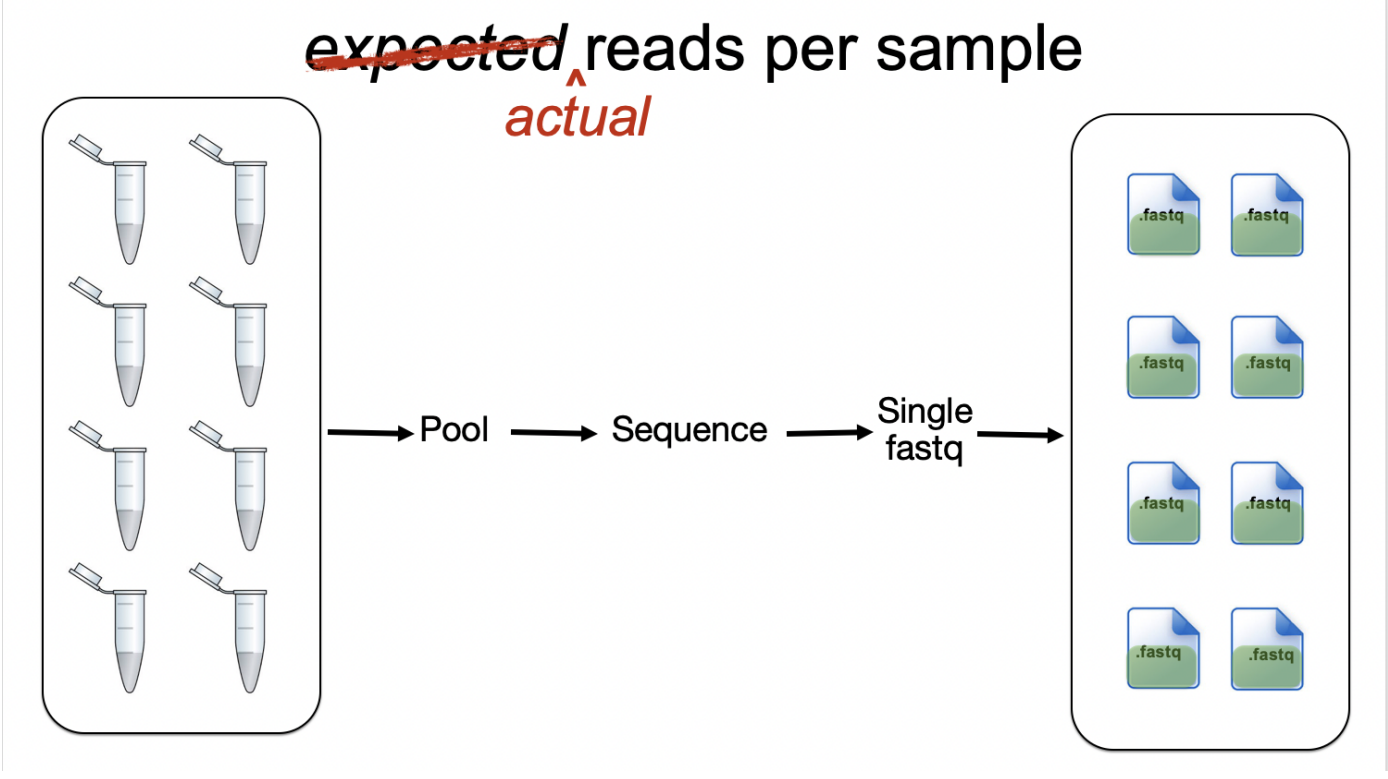
RNA-Seq: Transcript Quantification

Normalization



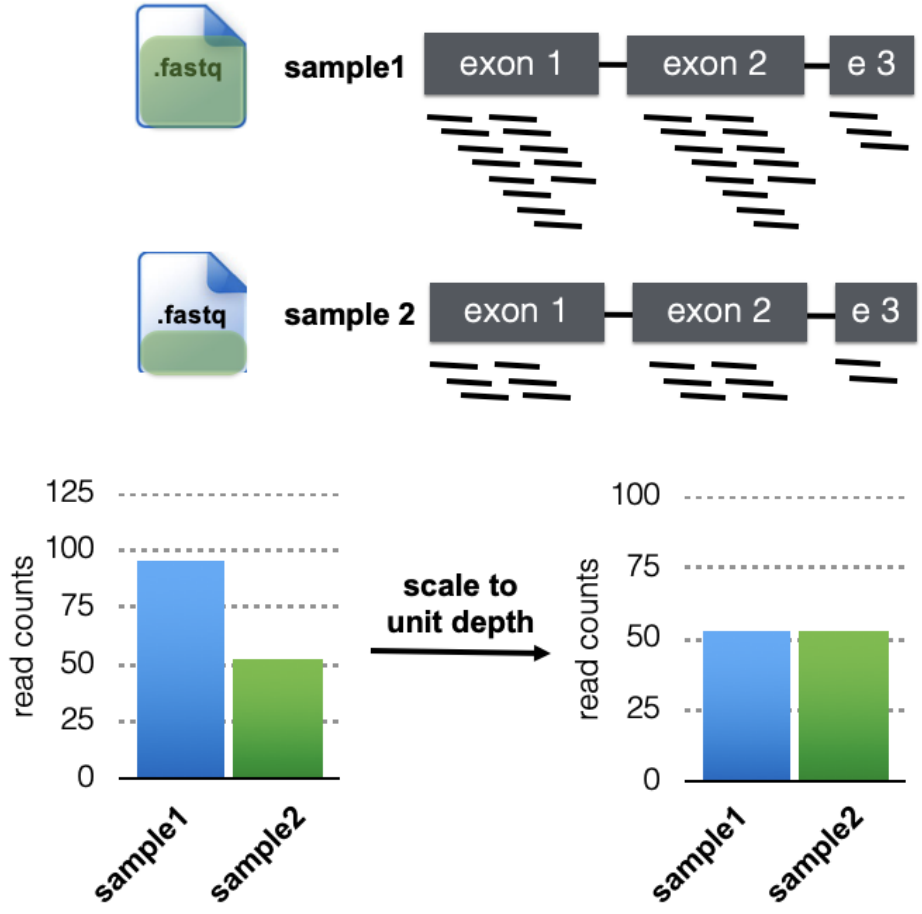
RNA-Seq: Transcript Quantification

Normalization



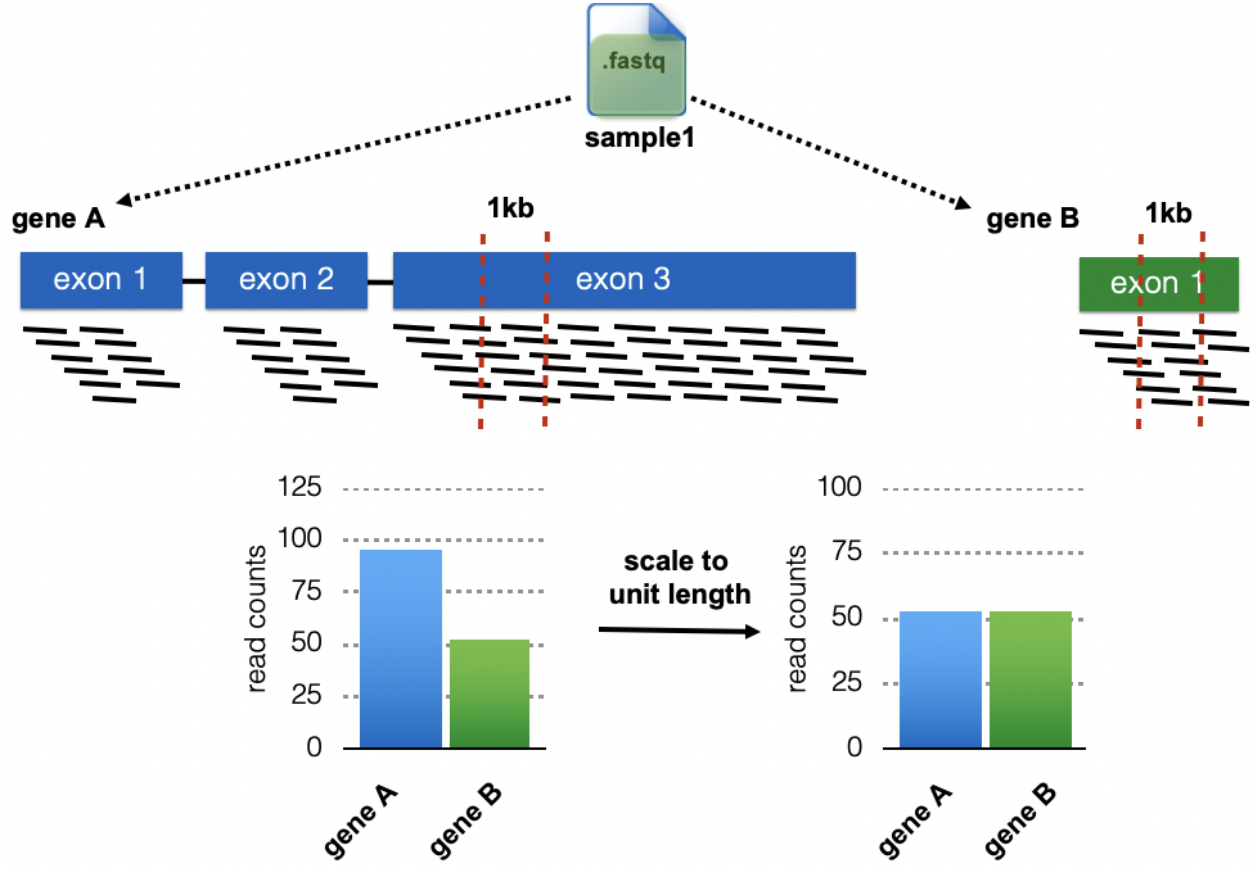
RNA-Seq: Transcript Quantification

Scaling between libraries



RNA-Seq: Transcript Quantification

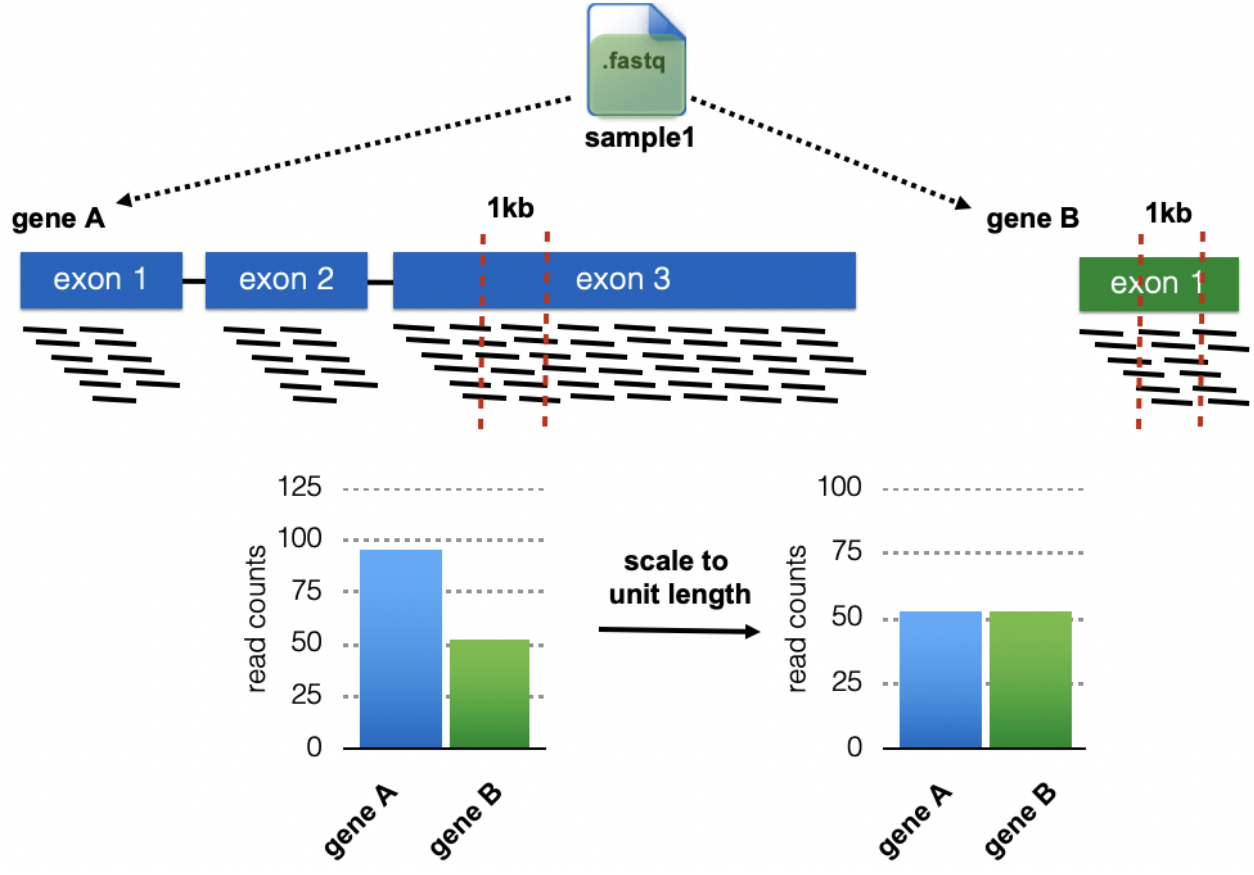
Scaling within libraries



RNA-Seq: Transcript Quantification

Scaling between and within libraries

Reads per Kilobase, per Million reads sequenced (RPKM)



RNA-Seq: Transcript Quantification

Scaling between and within libraries

Reads per Kilobase, per Million reads sequenced (RPKM)

$$\text{RPKM} = \frac{\text{\# reads mapped to genomic region}}{(\text{region length in kb})(\text{total \# reads})}$$

RNA-Seq: Transcript Quantification

Scaling between and within libraries

Reads per Kilobase, per Million reads sequenced (RPKM)

$$\text{RPKM} = \frac{\text{\# reads mapped to genomic region}}{(\text{region length in kb})(\text{total \# reads})}$$

$$\text{RPKM} = \frac{1000}{(5)(20000000)} = 0.00001$$

RNA-Seq: Transcript Quantification

Scaling between and within libraries

Reads per Kilobase, per Million reads sequenced (RPKM)

$$\text{RPKM} = \frac{\text{\# reads mapped to genomic region}}{(\text{region length in kb})(\text{total \# reads})} \times 10^6$$

$$\text{RPKM} = \frac{1000}{(5)(20000000)} = 10$$

RNA-Seq: Transcript Quantification

Scaling between and within libraries

Reads per Kilobase, per Million reads sequenced (RPKM)

$$\text{RPKM} = \frac{\#\text{ reads mapped to genomic region}}{(\text{region length in kb})(\text{total \# reads})} \times 10^6$$

$$\text{RPKM} = \frac{1000}{(5)(20000000)} = 10$$

⚠️ RPKM alone, is not sufficient for normalization

RNA-Seq: Transcript Quantification

Problem with RPKM

Read count

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total
Sample 1	80	10	6	3	1	100
Sample 2	20	20	10	50	400	500

$$\text{RPKM} = \frac{\text{\# reads mapped to gene}}{(\text{gene length in kb})(\text{total \# reads})} * 10^6$$

RNA-Seq: Transcript Quantification

Problem with RPKM

RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total
Sample 1	8000	2000	2400	6000	10000	28400
Sample 2	400	800	800	20000	800000	822000

RNA-Seq: Transcript Quantification

Problem with RPKM

RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total AVG
Sample 1	8000	2000	2400	6000	10000	28400 5680
Sample 2	400	800	800	20000	800000	822000 164000

RNA-Seq: Transcript Quantification

Problem with RPKM

RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total AVG
Sample 1	8000	2000	2400	6000	10000	28400 5680
Sample 2	400	800	800	20000	800000	822000 164000

> “The average relative molar concentration for each and every sample of RNA-seq data mapped to the same genome is the same constant value."
> Wagner, Theories in Biosci., 2012

RNA-Seq: Transcript Quantification

Fixing RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total
Sample 1	80	10	6	3	1	100
Sample 2	20	20	10	50	400	500

First, scale by gene length:

RNA-Seq: Transcript Quantification

Fixing RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total
Sample 1	80	10	6	3	1	100
Sample 2	20	20	10	50	400	500

First, scale by gene length:

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total RPK
Sample 1	80/100	10/50	6/25	3/5	1/1	100
Sample 2	20/100	20/50	10/25	50/5	400/1	500

RNA-Seq: Transcript Quantification

Fixing RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total
Sample 1	80	10	6	3	1	100
Sample 2	20	20	10	50	400	500

First, scale by gene length:

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total RPK
Sample 1	0.8	0.2	0.24	0.6	1	2.84
Sample 2	0.2	0.4	0.4	10	400	411

RNA-Seq: Transcript Quantification

Fixing RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total
Sample 1	80	10	6	3	1	100
Sample 2	20	20	10	50	400	500

First, scale by gene length:

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total RPK
Sample 1	0.8	0.2	0.24	0.6	1	**2.84**
Sample 2	0.2	0.4	0.4	10	400	**411**

➡️ Total RPK will be used as a normalization factor

RNA-Seq: Transcript Quantification

Fixing RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total
Sample 1	80	10	6	3	1	100
Sample 2	20	20	10	50	400	500

Then, normalize with Total RPK per sample:

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total RPK
Sample 1	0.8/2.84	0.2/2.84	0.24/2.84	0.6/2.84	1/2.84	**2.84**
Sample 2	0.2/411	0.4/411	0.4/411	10/411	400/411	**411**

RNA-Seq: Transcript Quantification

Fixing RPKM

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total
Sample 1	80	10	6	3	1	100
Sample 2	20	20	10	50	400	500

Then, normalize with Total RPK per sample:

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total RPK
Sample 1	281690	70423	84507	211268	352113	1000000
Sample 2	487	973	973	24331	973236	1000000

RNA-Seq: Transcript Quantification

Fixing RPKM = TPM

This is TPM (Transcripts Per Million)

Gene (length)	Gene A (100kb)	Gene B (50kb)	Gene C (25kb)	Gene D (5kb)	Gene E (1kb)	Total RPK
Sample 1	281690	70423	84507	211268	352113	1000000
Sample 2	487	973	973	24331	973236	1000000

$$\text{TPM} = \frac{\text{reads per Kb}}{\text{total RPK in sample}} * 10^6$$

RNA-Seq: Transcript Quantification

Additional factors

> “Library size scaling is too simple for many biological applications. The number of fragments expected to map to a gene is not only dependent on the expression level and length of the gene, but also the composition of the RNA population that is being sampled. Thus, if a large number of genes are unique to, or highly expressed in, one experimental condition, the sequencing 'real estate' available for the remaining genes in that sample is decreased. If not adjusted for, this sampling artifact can force the DE analysis to be skewed towards one experimental condition. ”

Robinson and Oshlack, Genome Biology, 2010