

High-throughput sequencing data processing

Charlotte Soneson

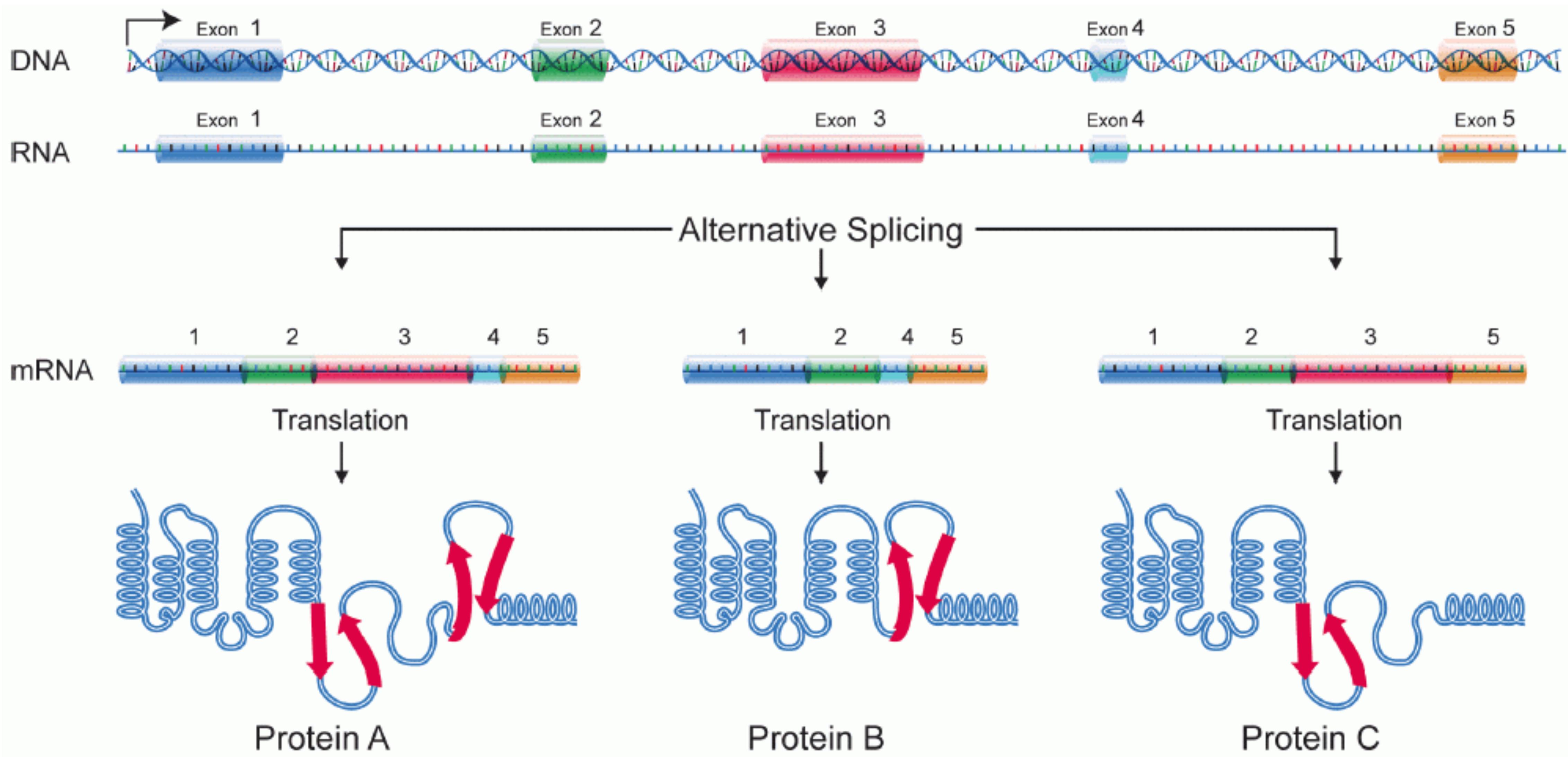
Friedrich Miescher Institute for Biomedical Research &
SIB Swiss Institute of Bioinformatics



Swiss Institute of
Bioinformatics



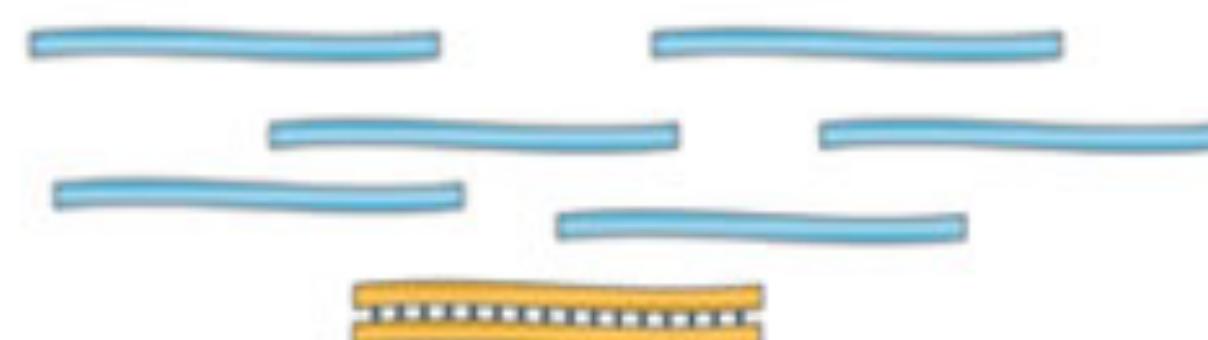
Friedrich Miescher Institute
for Biomedical Research



RNA-sequencing

a Data generation

① mRNA or total RNA

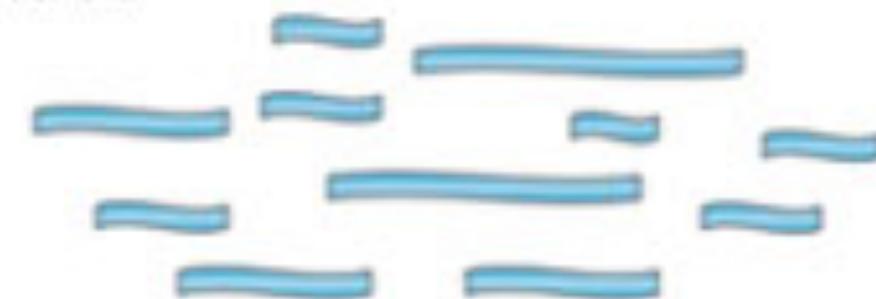


② Remove contaminant DNA

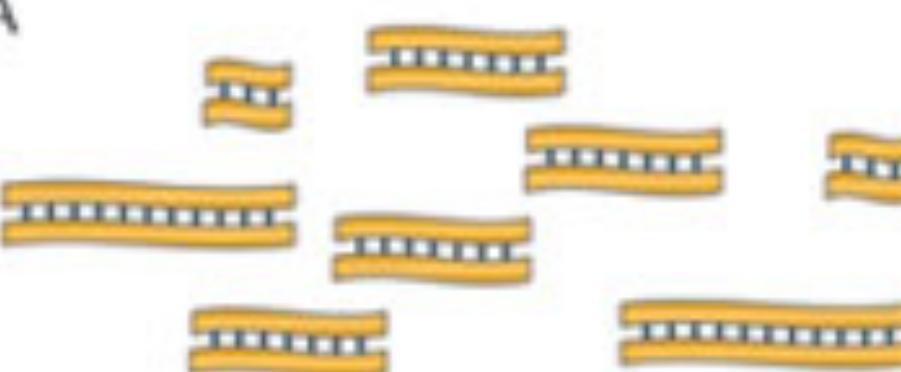


Remove rRNA?
Select mRNA?

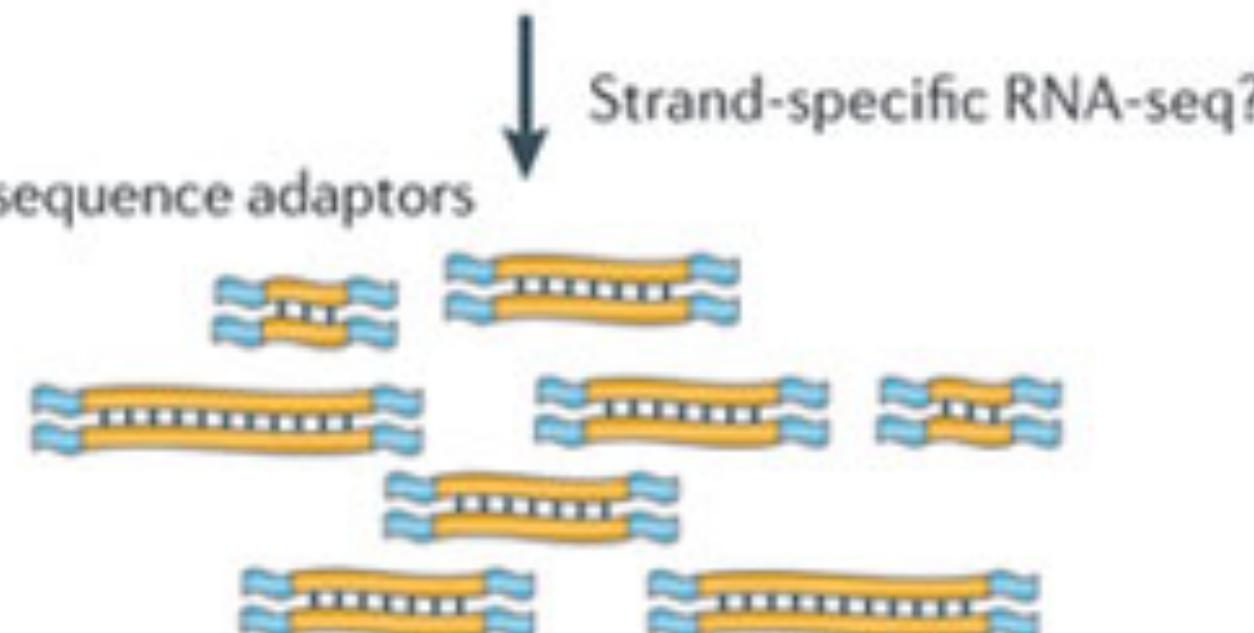
③ Fragment RNA



④ Reverse transcribe
into cDNA

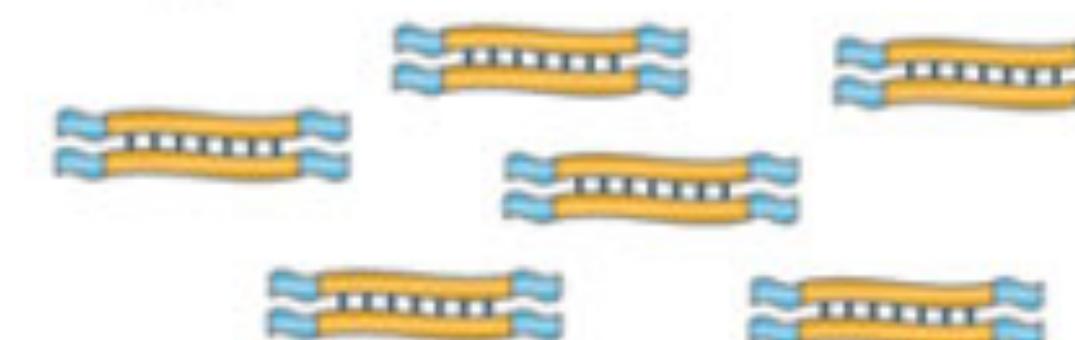


⑤ Ligate sequence adaptors



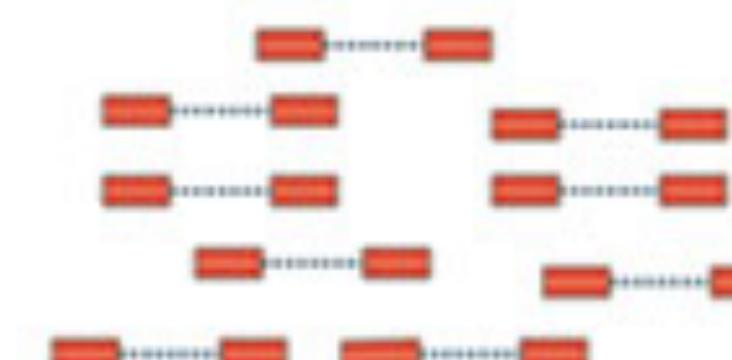
Strand-specific RNA-seq?

⑥ Select a range of sizes



PCR amplification?

⑦ Sequence cDNA ends



Single- vs paired-end sequencing



- Each fragment can be sequenced from one end only, or from both ends
- Single-end cheaper and faster
- Paired-end provide improved ability to localize the fragment in the genome and resolve mapping close to repeat regions - less multimapping reads

Strand-specificity

- In “standard” protocols, we don’t know from which strand a read stems
- Various “strand-specific” protocols allow us to keep this information
- Strand-specificity leads to lower number of ambiguous reads (overlapping multiple genes)

RESEARCH ARTICLE | OPEN ACCESS

Differentially expressed genes from RNA-Seq and functional enrichment results are affected by the choice of single-end versus paired-end reads and stranded versus non-stranded protocols

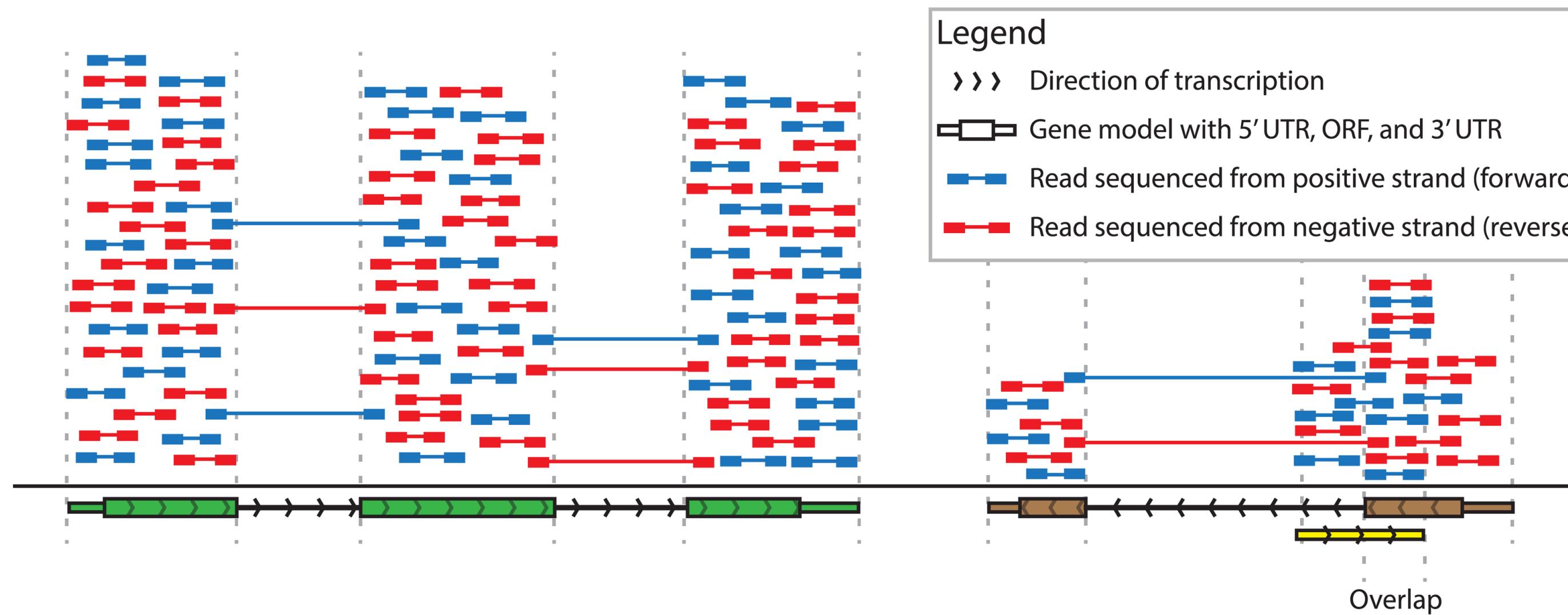
Susan M. Corley  , Karen L. MacKenzie, Annemiek Beverdam, Louise F. Roddam and Marc R. Wilkins

BMC Genomics 2017 18:399 | DOI: 10.1186/s12864-017-3797-0 | © The Author(s). 2017  ReadCube ▾

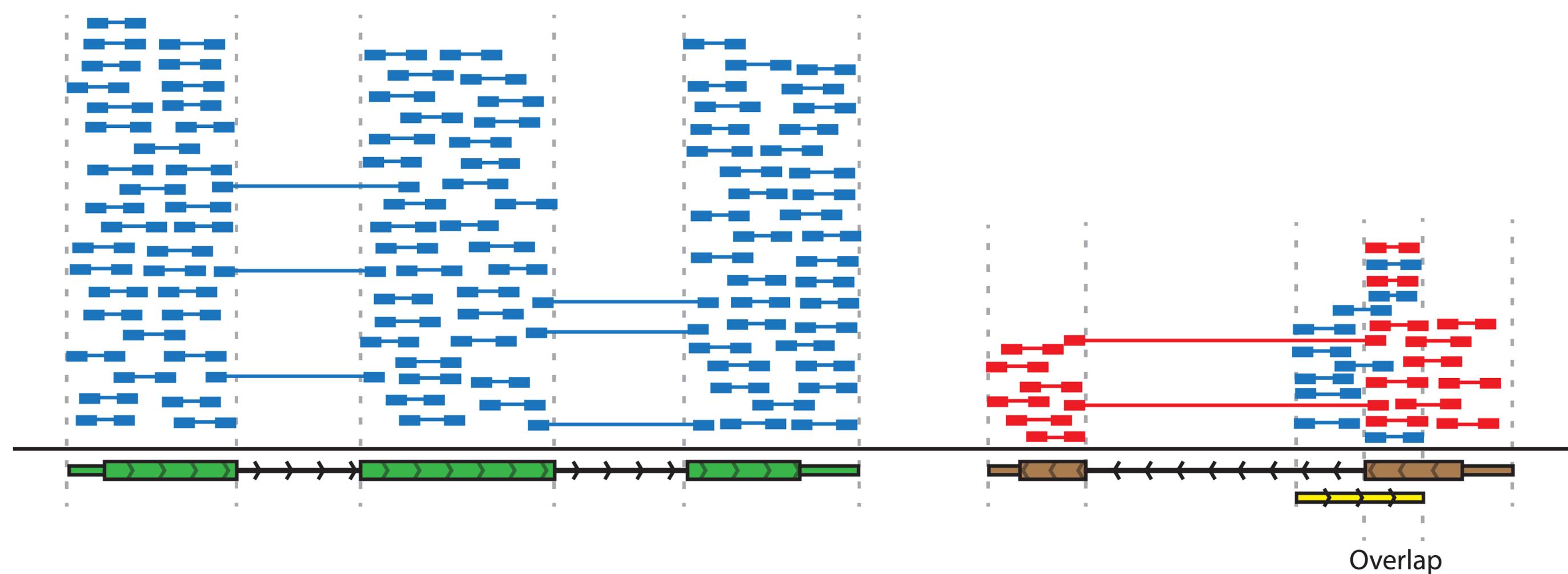
Received: 16 December 2016 | Accepted: 16 May 2017 | Published: 23 May 2017

Strand-specificity

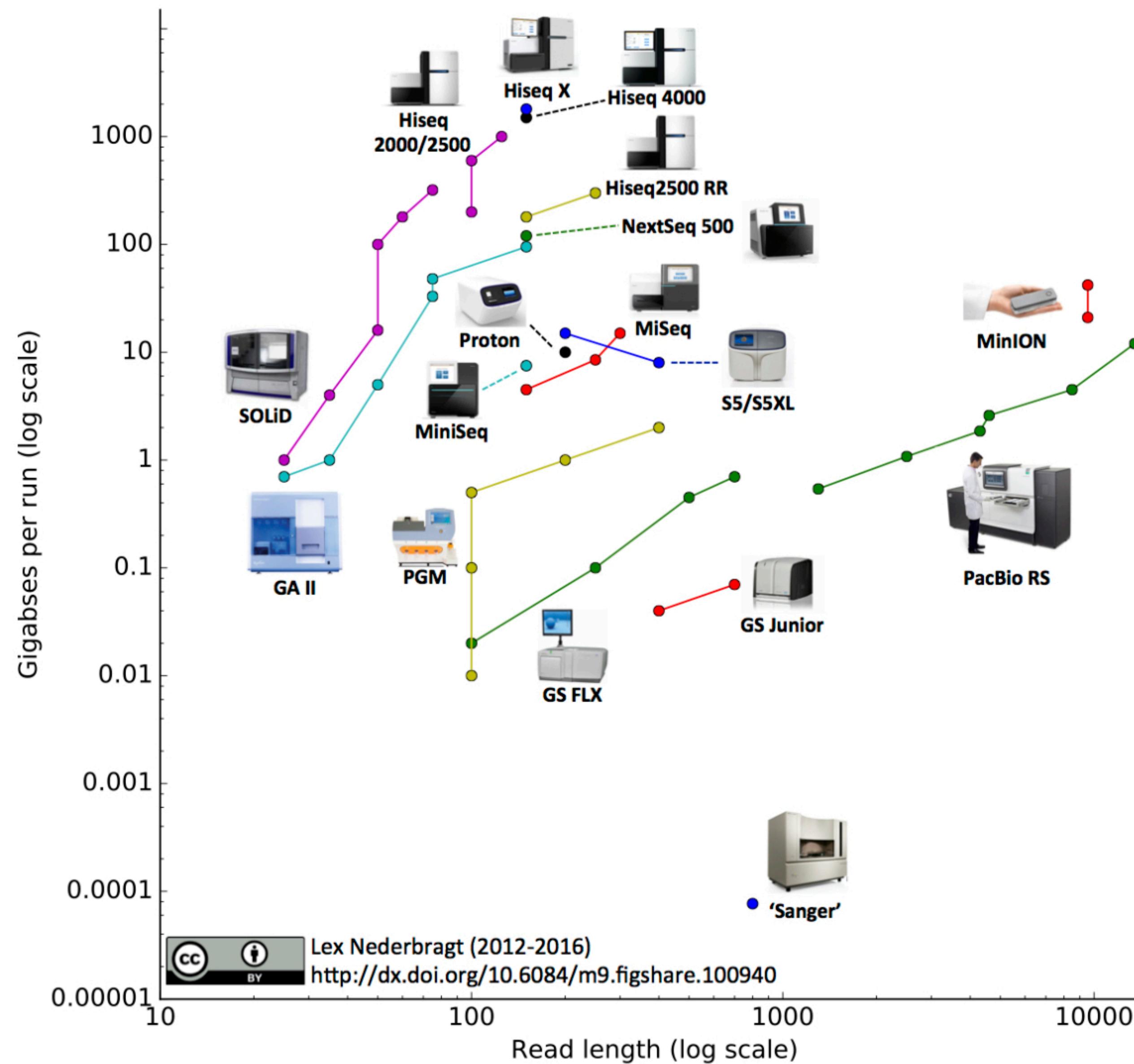
A.



B.



The world of sequencing technologies



Raw reads - FASTQ format

- Combines sequence and base quality information
- Four lines per sequence (read)
 - ID line (starting with @)
 - sequence
 - another ID line (starting with +)
 - base qualities
- For paired-end sequencing: one file for “first” reads and one for “second” reads

FASTQ format - sequence ID line

```
@D7MHBFN1:202:D1BUDACXX:4:1101:1340:1967 1:N:0:CATGCA
NATCTTCGGATCACTTGGTCAAATTGAAACGATAACAGAGAAGATTGTAAGTAACAATATTACCAAGGTTCGAGTCATACTAAGTGTCCCTATAGT
+
#1=DDFFFHHHHJJJJJJHIIJIIJJIIJGIIIJJJJJIIJJIIJJHIIFGIIIIJJJJIIEHJIIHHGFFF@?ADFEDDEDDBDDBDCDDDDEC
```

- D7MHBFN1 - unique instrument name
- 202 - run ID
- D1BUDACXX - flowcell ID
- 4 - flowcell lane
- 1101 - tile number within lane
- 1340 - x-coordinate of cluster within tile
- 1967 - y-coordinate of cluster within tile
- 1 - member of pair (1 or 2). Older versions: /1 and /2
- Y/N - whether the read failed quality control (Y = bad)
- 0 - none of the control bits are on
- CATGCA - index sequence (barcode)

FASTQ format - base qualities

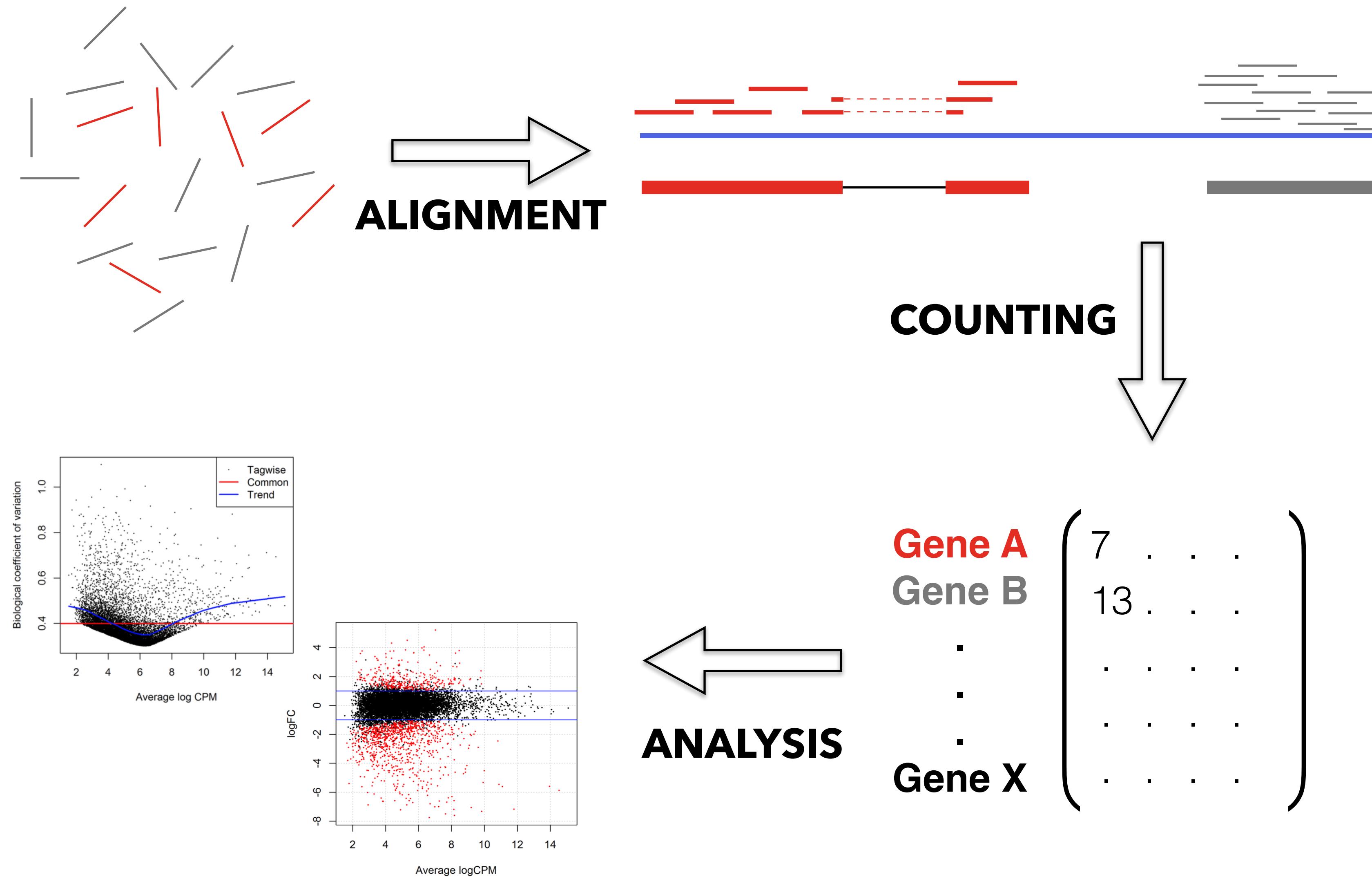
- For each letter, estimate the probability of being erroneous (p)
- Phred score $Q = -10 \cdot \log_{10}(p)$

Phred score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%
50	1 in 100000	99.999%

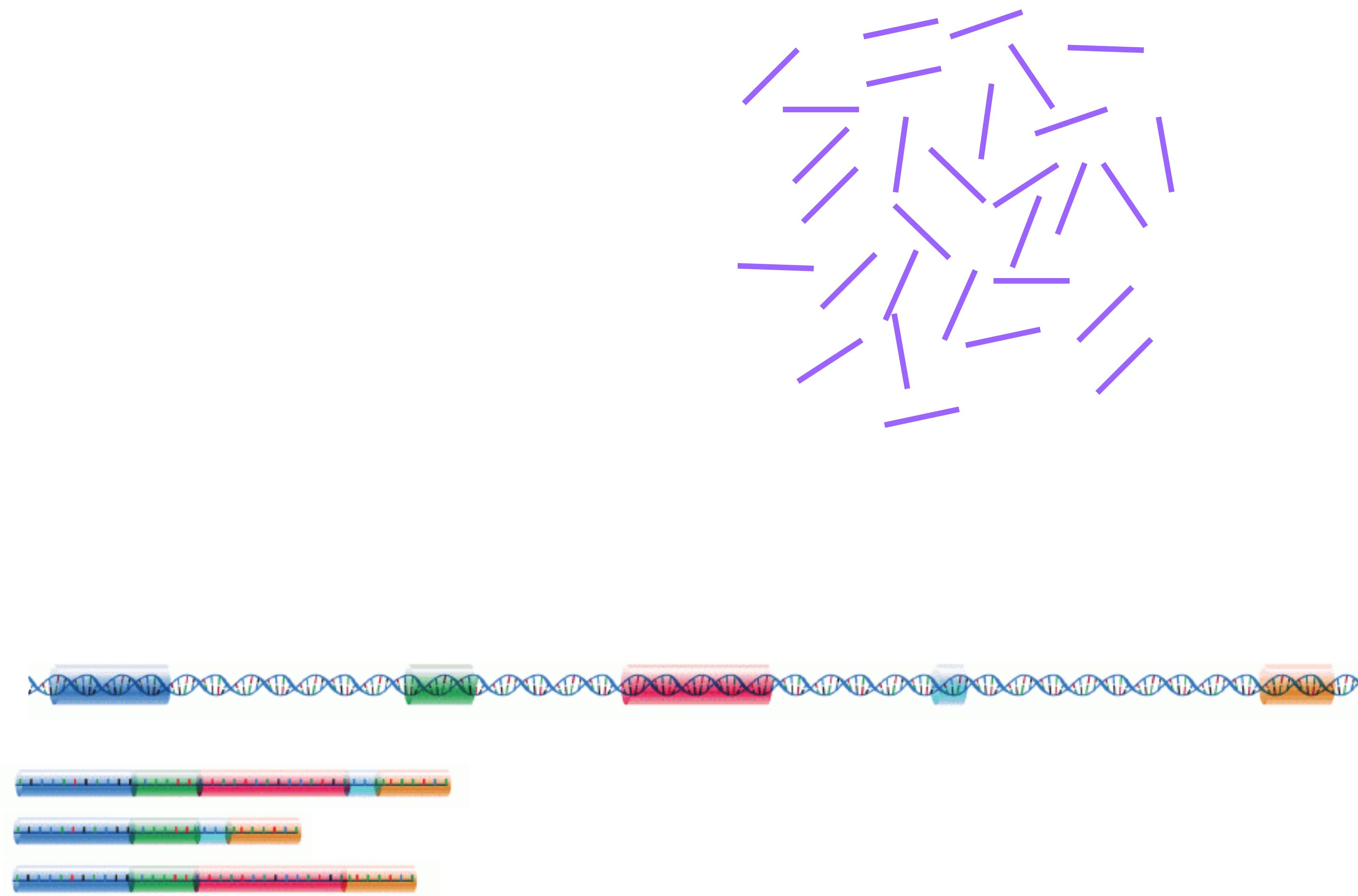
Quality format encoding

“Capital letters = good quality” (with Illumina 1.8+)

Alignment-based RNA-seq workflow

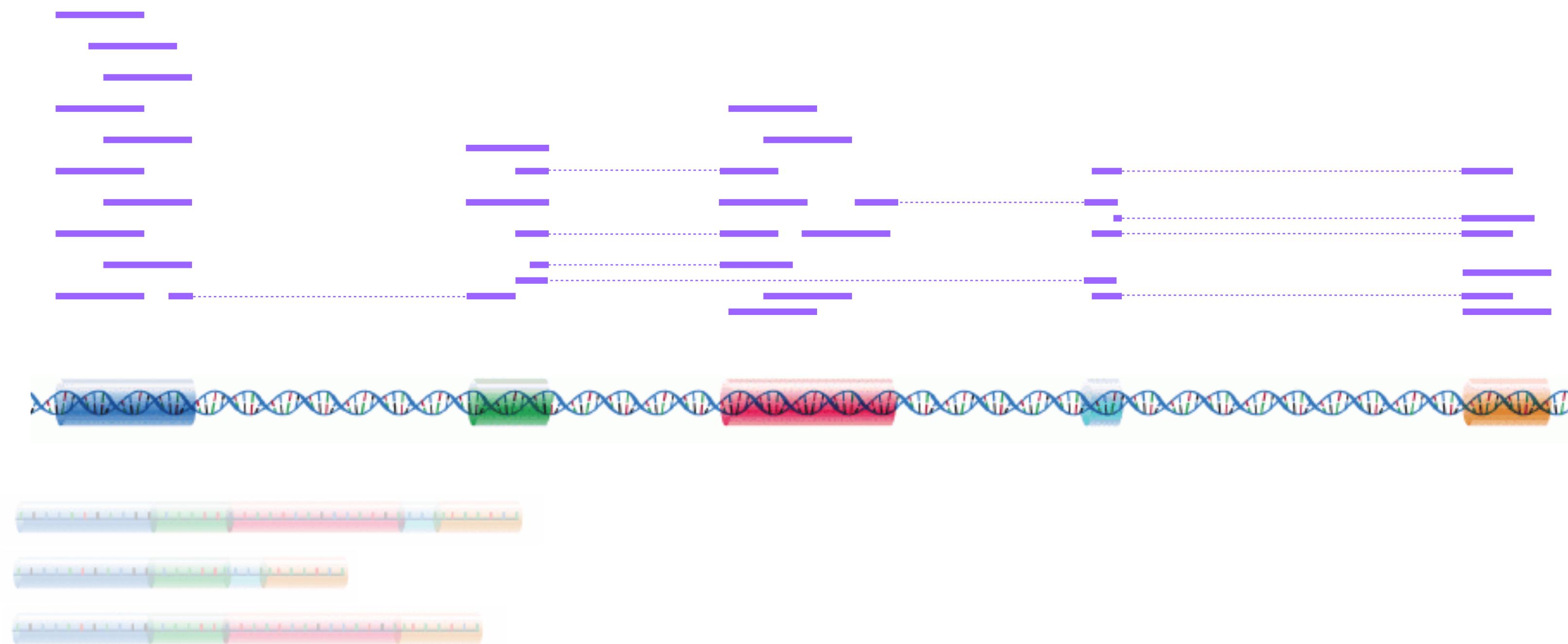


Abundance quantification



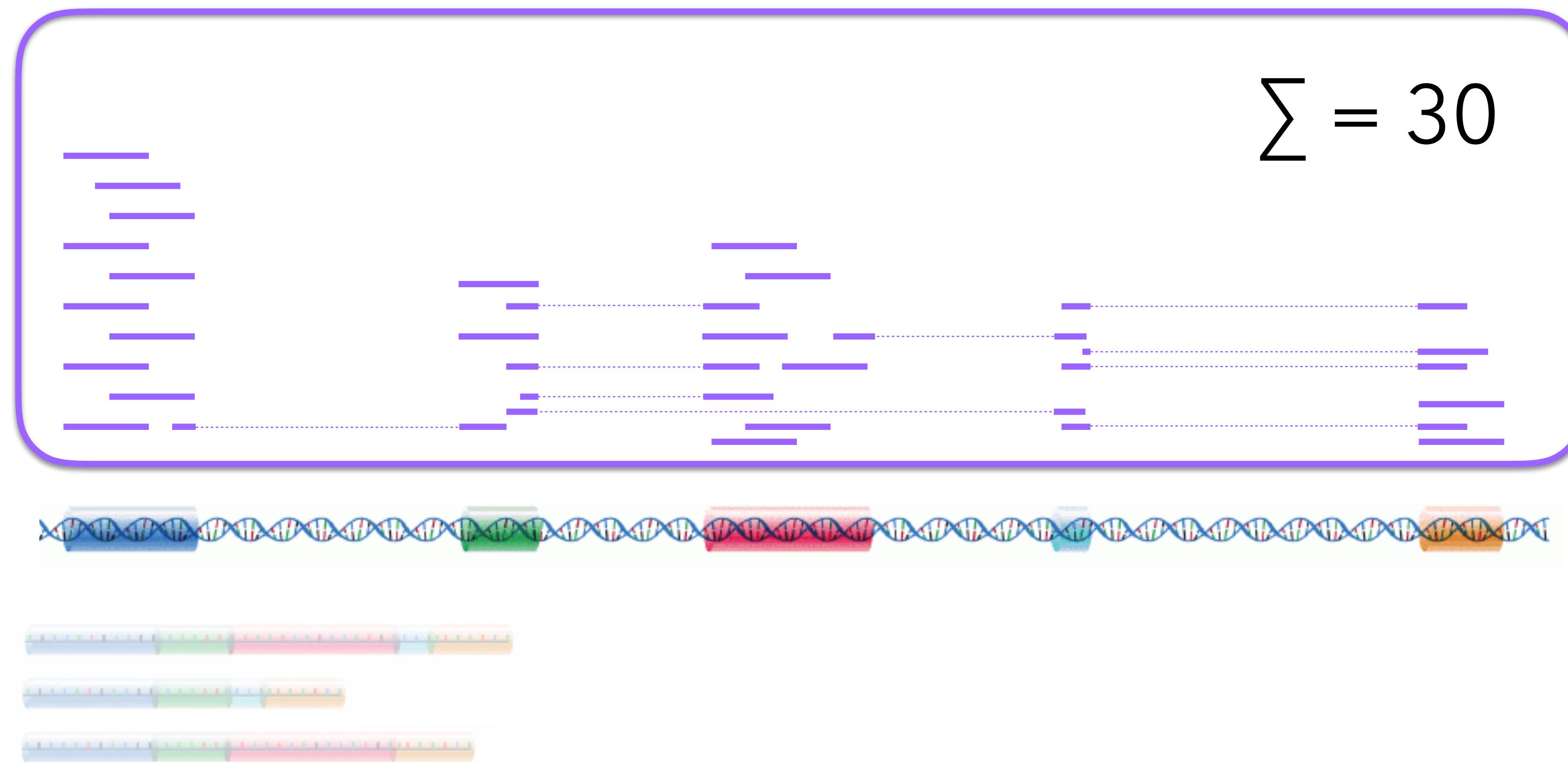
Abundance quantification

Gene-level counts, often obtained by
genome alignment + overlap counting



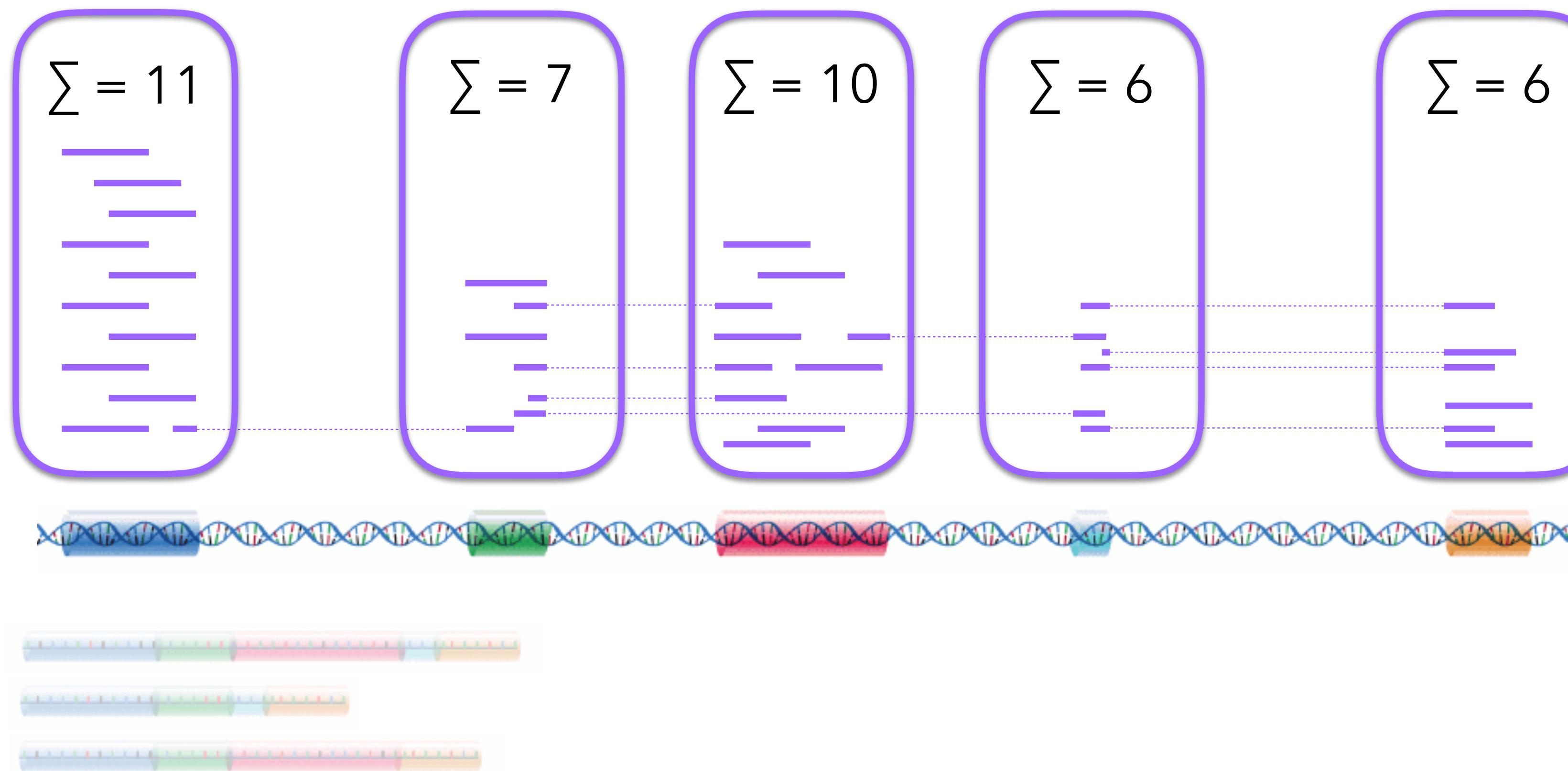
Abundance quantification

Gene-level counts, often obtained by genome alignment + overlap counting



Abundance quantification

Exon-level counts, often obtained by genome alignment + overlap counting



The human reference genome

- A “representative example” of the human genome sequence
- New versions are released periodically (the latest, GRCh38, in December 2013)
- Coordinates are not comparable across versions
- Typically provided as a *fasta* file (generic file format for biological sequences)

```
>chr1
.....
GTTCTTGTCTGTGTTCTTATAACCATACCAAGAATTTCATCACAGA
CAGAGACTAAACTCTTCTTCTTACCTTCCTTGATAATATTTGA
TCCAGGAATGGGGATAATTTGCAGTTAAAATTTCTTTATGATGGAA
GGTGAGGAGGGAGAGAGAGGTTACATTAGAAGTGACCCAACCTCCATTTC
TTCCAATGGTTTTTCAGTTTATTTTAAAGCGTGAACAGAGAATA
GTCACCTGATCAATTAAATATGTCAAAAGTGAAGAAAAATCTCTTT
TTAAAGGAAATGAGGGCAGTAACACAACCAAGGAATCAAATTCAAGGTTG
AGGCTGACCTTGACCTGCAACTATGCTACTCCATGAACAGCAAGTAGGA
AATGGCTGATTCATGAAGGTGGACTGGCATCAGAGGAGGCGAGGGATCC
AGGGTTCTGATGAGTGGCAACATTCTGGTCTTGAGTTGTTGAT
TGGTGAATCAAATTAGGTGACAGCCAGCTAAAGAGAGTGAGGGTGGCTG
TCTTGTGAATGGGAAGTGACCAAGCTTGAAAGCACAGACTgtggtggtc
.....|
```

The human reference genome

www.ensembl.org/info/data/ftp/index.html

Single species data

Popular species are listed first. You can customise this list via our [home page](#)

		Show 10 ↓ entries	Show/hide columns				
★	Species	DNA (FASTA)	cDNA (FASTA)	CDS (FASTA)	ncRNA (FASTA)	Protein sequence (FASTA)	
Y	<u>Human</u> <i>Homo sapiens</i>	FASTA ↗	FASTA ↗				
Y	<u>Mouse</u> <i>Mus musculus</i>	FASTA ↗	FASTA ↗				
Y	<u>Zebrafish</u> <i>Danio rerio</i>	FASTA ↗	FASTA ↗				

 [Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz](#)

840 MB

<https://www.gencodegenes.org/human/>

Fasta files

Content	Regions	Description	Download
Transcript sequences	CHR	<ul style="list-style-type: none">Nucleotide sequences of all transcripts on the reference chromosomes	Fasta
Protein-coding transcript sequences	CHR	<ul style="list-style-type: none">Nucleotide sequences of coding transcripts on the reference chromosomesTranscript biotypes: protein_coding, nonsense-mediated_decay, non_stop_decay, IG_*_gene, TR_*_gene, polymorphic_pseudogene	Fasta
Protein-coding transcript translation sequences	CHR	<ul style="list-style-type: none">Amino acid sequences of coding transcript translations on the reference chromosomesTranscript biotypes: protein_coding, nonsense-mediated_decay, non_stop_decay, IG_*_gene, TR_*_gene, polymorphic_pseudogene	Fasta
Long non-coding RNA transcript sequences	CHR	<ul style="list-style-type: none">Nucleotide sequences of long non-coding RNA transcripts on the reference chromosomes	Fasta
Genome sequence (GRCh38.p12)	ALL	<ul style="list-style-type: none">Nucleotide sequence of the GRCh38.p12 genome assembly version on all regions, including reference chromosomes, scaffolds, assembly patches and haplotypesThe sequence region names are the same as in the GTF/GFF3 files	Fasta
Genome sequence, primary assembly (GRCh38)	PRI	<ul style="list-style-type: none">Nucleotide sequence of the GRCh38 primary genome assembly (chromosomes and scaffolds)The sequence region names are the same as in the GTF/GFF3 files	Fasta

Genomic locations of genes and other features

- Typically provided in a **gtf** (gene transfer format) file
- Similar to **gff**, but more standardized

seqname	source	feature	start	end	score	strand	frame	attribute
2R	protein_coding	exon	5139815	5141712	.	-	.	gene_id "FBgn0020621"; transcript_id "FBtr0112897"; exon_number "10"; gene_name "Pkn"; gene_biotype "protein_coding"; transcript_name "Pkn-RG"; exon_id "FBgn0020621:1";
2R	protein_coding	CDS	5141572	5141712	.	-	0	gene_id "FBgn0020621"; transcript_id "FBtr0112897"; exon_number "10"; gene_name "Pkn"; gene_biotype "protein_coding"; transcript_name "Pkn-RG"; protein_id "FBpp0111810";
2R	protein_coding	stop_codon	5141569	5141571	.	-	0	gene_id "FBgn0020621"; transcript_id "FBtr0112897"; exon_number "10"; gene_name "Pkn"; gene_biotype "protein_coding"; transcript_name "Pkn-RG";

Genomic locations of genes and other features

www.ensembl.org/info/data/ftp/index.html

<https://www.gencodegenes.org/human/>

See this list via our [home page](#).

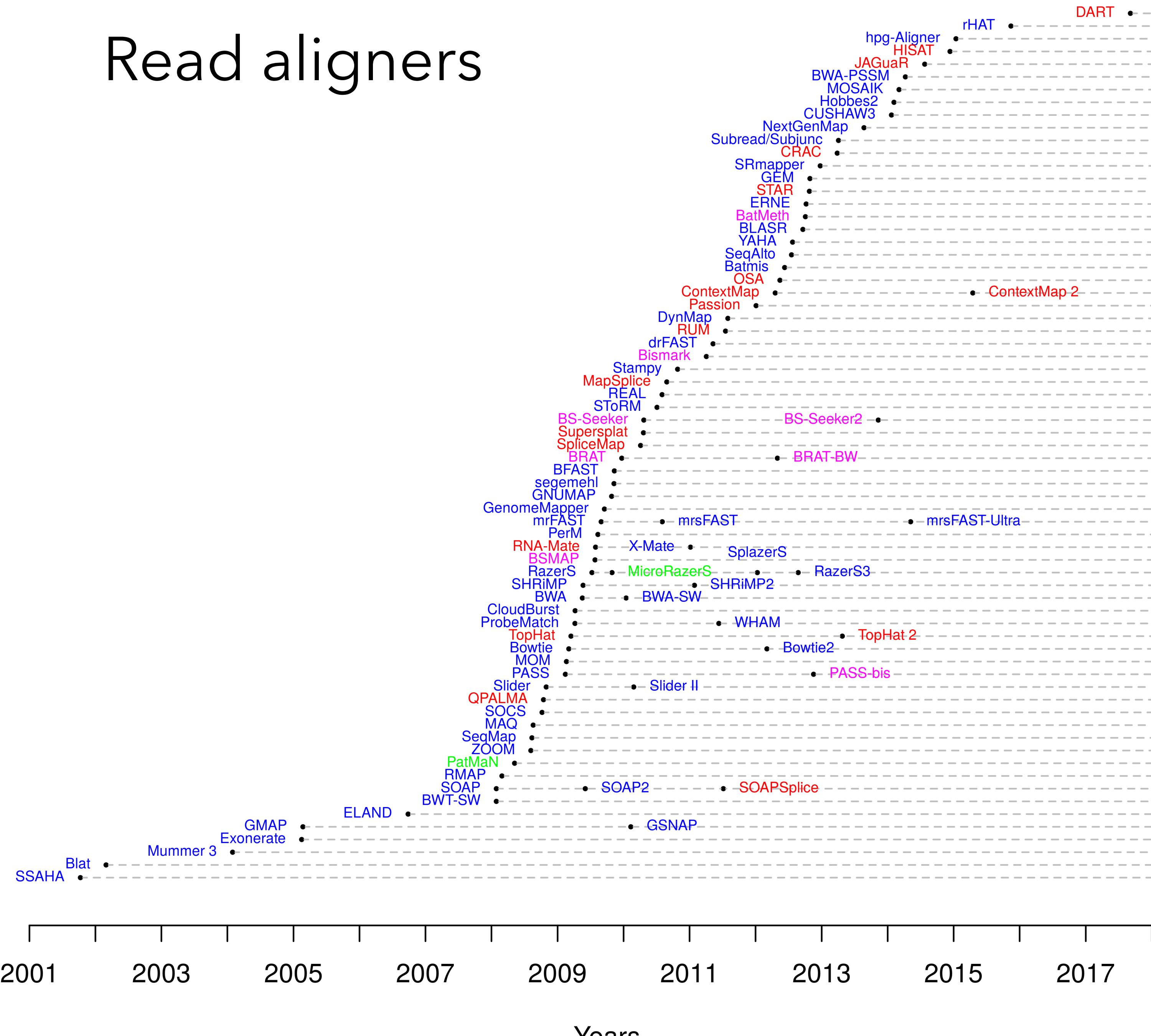
Show/hide columns							
DS STA)	ncRNA (FASTA)	Protein sequence	Annotated sequence (FASTA)	Annotated sequence (EMBL)	Gene sets (GenBank)	Whole databases (GV)	Vari-
GTF	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	G'
GTF	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	G'
GTF	FASTA	FASTA	EMBL	GenBank	GTF	MySQL	G'

GTF / GFF3 files

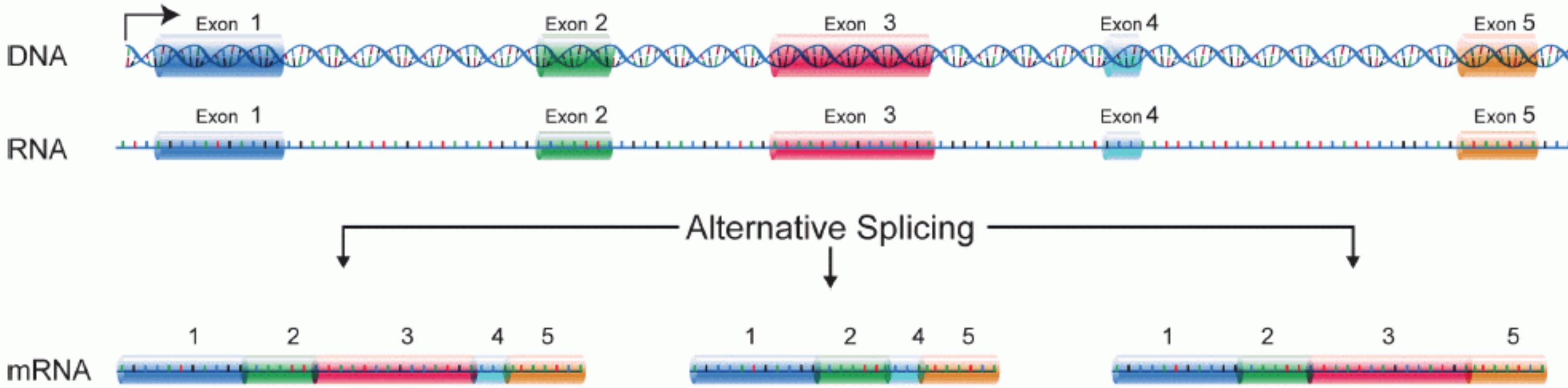
Content	Regions	Description	Download
Comprehensive gene annotation	CHR	<ul style="list-style-type: none">It contains the comprehensive gene annotation on the reference chromosomes onlyThis is the main annotation file for most users	GTF GFF3
Comprehensive gene annotation	ALL	<ul style="list-style-type: none">It contains the comprehensive gene annotation on the reference chromosomes, scaffolds, assembly patches and alternate loci (haplotypes)This is a superset of the main annotation file	GTF GFF3
Comprehensive gene annotation	PRI	<ul style="list-style-type: none">It contains the comprehensive gene annotation on the primary assembly (chromosomes and scaffolds) sequence regionsThis is a superset of the main annotation file	GTF GFF3
Basic gene annotation	CHR	<ul style="list-style-type: none">It contains the basic gene annotation on the reference chromosomes onlyThis is a subset of the corresponding comprehensive annotation, including only those transcripts tagged as 'basic' in every gene	GTF GFF3

Read aligners

DNA
RNA
miRNA
bisulfite



Aligning RNA-seq reads



- Need a splice-aware aligner
- Common choices:
 - STAR
 - HISAT2

STAR: step 1 - indexing the genome

```
$ STAR --runThreadN 24 \  
    --runMode genomeGenerate \  
    --genomeDir my_genome \  
    --genomeFastaFiles my_genome.fa \  
    --sjdbGTFfile my_genes.gtf \  
    --sjdbOverhang 99
```

number of threads



output folder -
name according

to genome



read length - 1



STAR: step 2 - aligning the reads

```
$ STAR --runThreadN 24 \  
    --runMode alignReads \  
    --genomeDir my_genome \  
    --readFilesIn S1_read1.fq.gz \  
        S1_read2.fq.gz \  
        --readFilesCommand zcat \  
        --outFileNamePrefix output/S1/ \  
        --outSAMtype BAM SortedByCoordinate \  
        --quantMode GeneCounts
```

count reads

created index

read file(s)

[include sample ID]

for compressed read files

STAR: output

 SRR1039508	 SRR1039508_Aligned.sortedByCoord.out.bam
 SRR1039509	 SRR1039508_Log.final.out
 SRR1039512	 SRR1039508_Log.out
 SRR1039513	 SRR1039508_Log.progress.out
 SRR1039516	 SRR1039508_ReadsPerGene.out.tab
 SRR1039517	 SRR1039508_SJ.out.tab
 SRR1039520	
 SRR1039521	

Representing alignments - SAM format

- # • Header

```
@SQ SN:chr1 LN:249250621
@SQ SN:chr2 LN:243199373
@SQ SN:chr3 LN:198022430
@SQ SN:chr4 LN:191154276
```

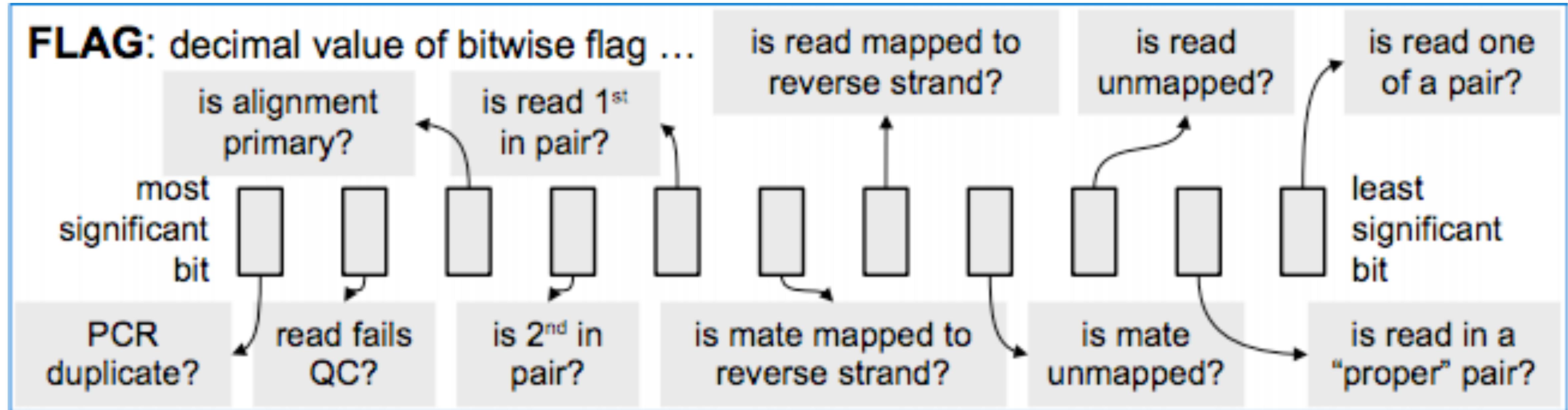
- Body

- Typically, one line per alignment
 - BAM = binary SAM

Representing alignments - SAM format

- Column 1 - sequence ID
 - Column 2 - flag. Ex:
 - 0 - non-paired read, mapping to forward strand
 - 16 - non-paired read, mapping to reverse strand
 - 4 - unmapped read
 - Column 3 - reference sequence name for the alignment
 - Column 4 - position of alignment
 - Column 5 - mapping quality
 - 255 - not available
 - 0 - multiple best hits
 - Column 6 - CIGAR string
 - Column 7-8 - reference name/position of mate/next segment
 - Column 9 - observed template length
 - Column 10 - sequence (represented as mapped on the reference (forward) strand!)
 - Column 11 - base quality
 - Remaining columns are optional, and are of the type TAG:TYPE:VALUE

The SAM flag



- ex: 83 = 00001010011 = first in pair, read on reverse strand, part of properly mapped pair

The CIGAR string

Op	BAM	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch

- Describes the mapping in more detail
- See also the MD tag

The CIGAR string - example

RefPos:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Reference:	C	C	A	T	A	C	T	G	A	A	C	T	G	A	C	T	A	A	C
Read:	ACTAGAATGGCT																		

Aligning these two:

RefPos:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Reference:	C	C	A	T	A	C	T	G	A	A	C	T	G	A	C	T	A	A	C
Read:				A	C	T	A	G	A	A		T	G	G	C	T			

With the alignment above, you get:

POS:	5
CIGAR:	3M1I3M1D5M

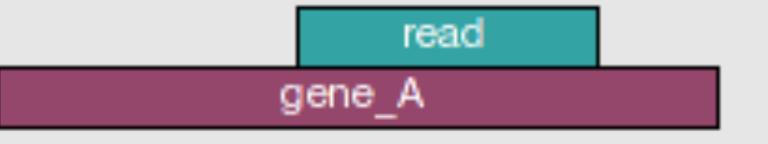
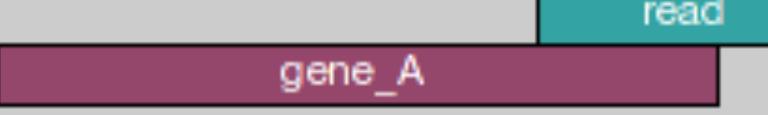
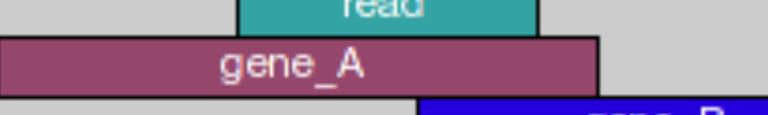
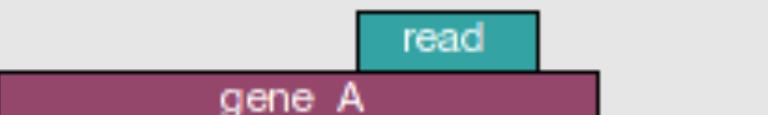
Working with SAM/BAM files

- SAMtools
 - convert between SAM/BAM
 - sort/index
 - view alignments
 - ...
- R interface in the Rsamtools package

Estimating abundances via overlap counting

- STAR
- HTseq-count (Python)
- Rsubread::featureCounts (R)
- GenomicAlignments::summarizeOverlaps (R)

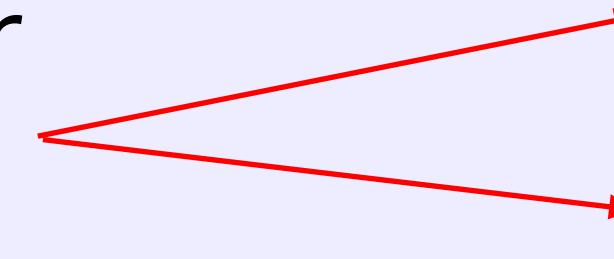
Counting modes

	union	intersection _strict	intersection _nonempty
 A single read (cyan bar) overlaps with gene_A (purple bar).	gene_A	gene_A	gene_A
 A read (cyan bar) starts within gene_A (purple bar).	gene_A	no_feature	gene_A
 A read (cyan bar) ends within gene_A (purple bar).	gene_A	no_feature	gene_A
 Two reads (cyan bars) overlap with gene_A (purple bar).	gene_A	gene_A	gene_A
 A read (cyan bar) spans across gene_A (purple bar) and gene_B (blue bar).	gene_A	gene_A	gene_A
 A read (cyan bar) starts in gene_A (purple bar) and ends in gene_B (blue bar).	ambiguous	gene_A	gene_A
 A read (cyan bar) ends in gene_A (purple bar) and starts in gene_B (blue bar).	ambiguous	ambiguous	ambiguous

featureCounts

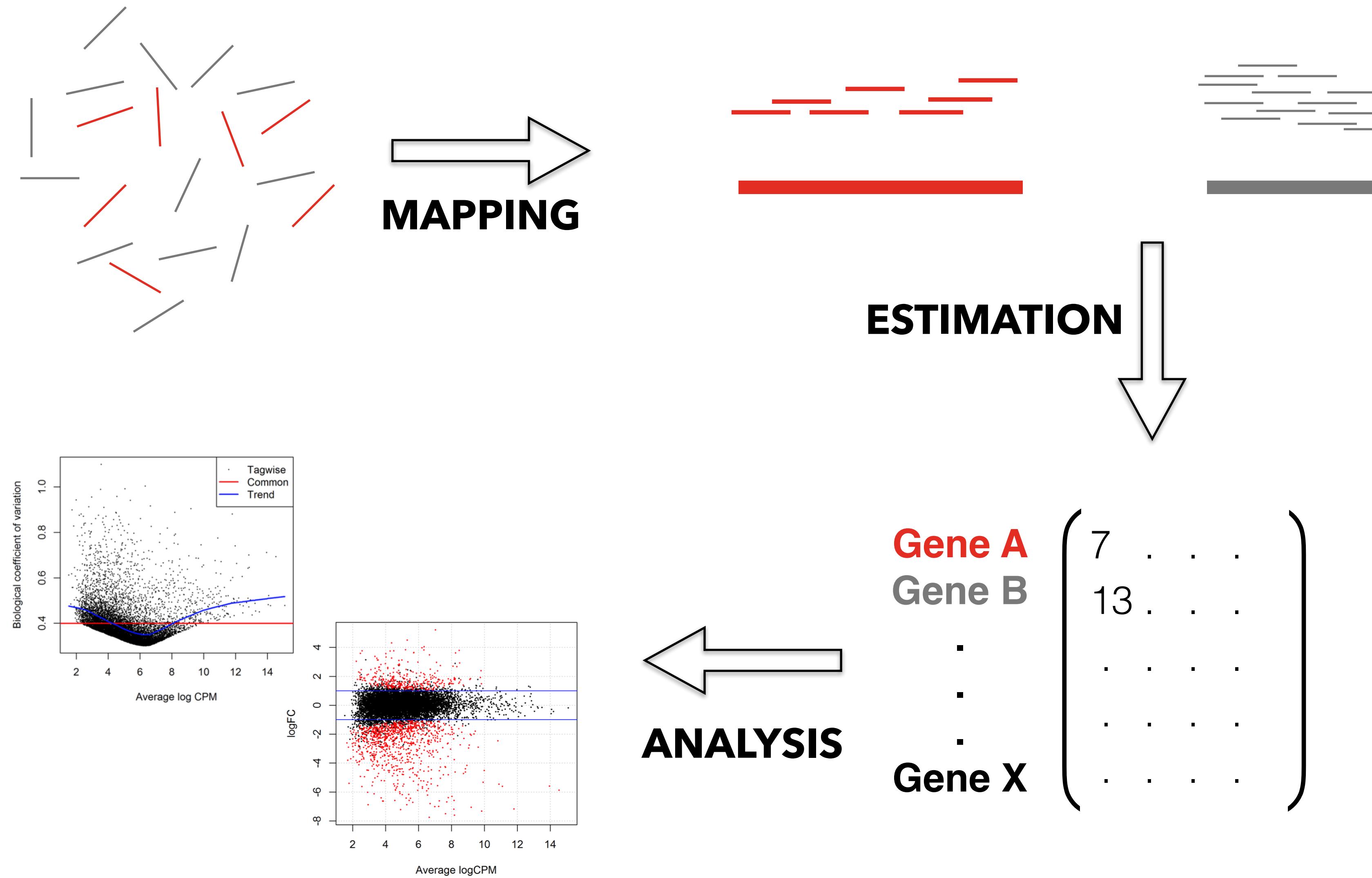
```
> featureCounts(files = bamfiles,
                  annot.ext = "my_genes.gtf",
                  isGTFAnnotationFile = TRUE,
                  GTF.featureType = "exon",
                  GTF.attrType = "gene_id",
                  useMetaFeatures = TRUE,
                  isPairedEnd = TRUE,
                  strandSpecific = 0)
```

check your
GTF file!



```
2R  protein_coding exon 5139815 5141712 . - . gene_id "FBgn0020621"; transcript_id
"FBtr0112897"; exon_number "10"; gene_name "Pkn"; gene_biotype "protein_coding"; transcript_name "Pkn-RG";
exon_id "FBgn0020621:1";
```

Alignment-free RNA-seq workflow



Abundance quantification

Equivalence class counts, often obtained by
“alignment-free” estimation methods



- **Salmon** (Patro et al, *Nat Methods* 2017)
- **kallisto** (Bray et al, *Nat Biotechnol* 2016)

Abundance quantification

Transcript-level counts, often obtained by
“alignment-free” estimation methods



Abundance quantification

Gene-level counts, obtained by summation of transcript-level counts



Reference transcript sequences

www.ensembl.org/info/data/ftp/index.html

Single species data

Popular species are listed first. You can customise this list via our [home page](#)

		Show 10 entries	Show/hide columns				
★	Species	DNA (FASTA)	cDNA (FASTA)	CDS (FASTA)	ncRNA (FASTA)	Protein sequence (FASTA)	
Y	Human <i>Homo sapiens</i>	FASTA ↗					
Y	Mouse <i>Mus musculus</i>	FASTA ↗					
Y	Zebrafish <i>Danio rerio</i>	FASTA ↗					

<https://www.gencodegenes.org/human/>

Fasta files

Content	Regions	Description	Download
Transcript sequences	CHR	<ul style="list-style-type: none">Nucleotide sequences of all transcripts on the reference chromosomes	Fasta
Protein-coding transcript sequences	CHR	<ul style="list-style-type: none">Nucleotide sequences of coding transcripts on the reference chromosomesTranscript biotypes: protein_coding, nonsense-mediated_decay, non_stop_decay, IG_*_gene, TR_*_gene, polymorphic_pseudogene	Fasta
Protein-coding transcript translation sequences	CHR	<ul style="list-style-type: none">Amino acid sequences of coding transcript translations on the reference chromosomesTranscript biotypes: protein_coding, nonsense-mediated_decay, non_stop_decay, IG_*_gene, TR_*_gene, polymorphic_pseudogene	Fasta
Long non-coding RNA transcript sequences	CHR	<ul style="list-style-type: none">Nucleotide sequences of long non-coding RNA transcripts on the reference chromosomes	Fasta
Genome sequence (GRCh38.p12)	ALL	<ul style="list-style-type: none">Nucleotide sequence of the GRCh38.p12 genome assembly version on all regions, including reference chromosomes, scaffolds, assembly patches and haplotypesThe sequence region names are the same as in the GTF/GFF3 files	Fasta
Genome sequence, primary assembly (GRCh38)	PRI	<ul style="list-style-type: none">Nucleotide sequence of the GRCh38 primary genome assembly (chromosomes and scaffolds)The sequence region names are the same as in the GTF/GFF3 files	Fasta

Step 1: build transcriptome index

kallisto

```
$ kallisto index -i my_transcripts.idx \  
my_transcripts.fasta
```

name of index

transcriptome fasta file

Salmon

```
$ salmon index -i my_transcripts.idx \  
-t my_transcripts.fasta
```

Step 2: estimate transcript abundances

number of cores

kallisto

```
$ kallisto quant -i my_transcripts.idx \
-o results/sample1 -b 30 -t 10 \
sample1_1.fastq sample1_2.fastq
```

output folder

name of index

bootstraps

Salmon

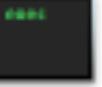
input fastq files

libtype

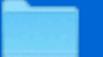
```
$ salmon quant -i my_transcripts.idx -l A \
-1 sample1_1.fastq -2 sample1_2.fastq \
-p 10 -o results/sample1 --validateMappings \
--numBootstraps 30 --seqBias --gcBias
```

Output

kallisto

-
-  abundance.h5 ✓
 -  abundance.tsv ✓
 -  run_info.json ✓

Salmon

 aux_info	▶	 ambig_info.tsv
 cmd_info.json	▶	 eq_classes.txt
 lib_format_counts.json	▶	 exp_gc.gz
 libParams	▶	 exp3_seq.gz
 logs	▶	 exp5_seq.gz
 quant.sf		 expected_bias.gz
		 fld.gz
		 meta_info.json
		 obs_gc.gz
		 obs3_seq.gz
		 obs5_seq.gz
		 observed_bias_3p.gz
		 observed_bias.gz

Output

kallisto

[abundance.tsv]

target_id	length	eff_length	est_counts	tpm
ENST00000406070	2025	1874.91	0	0
ENST00000446844	2227	2076.91	3.37465	0.129755
ENST00000599620	686	535.97	0	0
ENST00000471557	505	355.404	2.84168	0.638509
ENST00000338761	1456	1305.91	1.3122e-05	8.02414e-07
ENST00000417509	1444	1293.91	5.15988	0.318455
ENST00000484946	610	460.029	17.4159	3.02326
ENST00000490656	660	509.97	7.51996	1.17756
ENST00000439537	1161	1010.91	14.432	1.14006
ENST00000493251	641	491.006	2.63203	0.428073
ENST00000460127	408	259.526	0	0

Salmon

[quant.sf]

Name	Length	EffectiveLength	TPM	NumReads
ENST00000406070	2025	1869.81	0	0
ENST00000446844	2227	2071.81	0.137334	3.71695
ENST00000599620	686	530.936	0	0
ENST00000471557	505	350.256	0.731211	3.3457
ENST00000338761	1456	1300.81	0	0
ENST00000417509	1444	1288.81	7.58582e-08	1.27717e-06
ENST00000484946	610	455.039	2.87905	17.1142
ENST00000490656	660	504.969	1.46703	9.67744
ENST00000439537	1161	1005.81	1.47611	19.3952
ENST00000493251	641	485.994	0.597774	3.79512
ENST00000460127	408	253.708	0	0

Reading the estimated values into R

```
> library(tximport)
> salmon_files
      SRR1039508          SRR1039509          SRR1039512
"salmon/SRR1039508/quant.sf" "salmon/SRR1039509/quant.sf" "salmon/SRR1039512/quant.sf"
      SRR1039513          SRR1039516          SRR1039517
"salmon/SRR1039513/quant.sf" "salmon/SRR1039516/quant.sf" "salmon/SRR1039517/quant.sf"
      SRR1039520          SRR1039521
"salmon/SRR1039520/quant.sf" "salmon/SRR1039521/quant.sf"
> head(tx2gene)
    tx          gene
1 ENST00000456328.2 ENSG00000223972.5
2 ENST00000450305.2 ENSG00000223972.5
3 ENST00000488147.1 ENSG00000227232.5
4 ENST00000619216.1 ENSG00000278267.1
5 ENST00000473358.1 ENSG00000243485.5
6 ENST00000469289.1 ENSG00000243485.5
```

Reading the estimated values into R

```
> txi <- tximport::tximport(files = salmon_files, type = "salmon",
+                               tx2gene = tx2gene)
reading in files with read_tsv
1 2 3 4 5 6 7 8
summarizing abundance
summarizing counts
summarizing length
> names(txi)
[1] "abundance"          "counts"             "length"            "countsFromAbundance"
```

Reading the estimated values into R

```
> head(txr$counts, 3)
          SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517
ENSG00000000003.14    707.21    463.445    896.252    421.186   1185.87   1086.289
ENSG00000000005.5      0.00      0.000      0.000      0.000      0.00      0.000
ENSG00000000419.12    454.00    508.999    606.000    352.999    583.00    773.000
          SRR1039520 SRR1039521
ENSG00000000003.14     802      596.220
ENSG00000000005.5       0      0.000
ENSG00000000419.12     410      500.001
> head(txr$abundance, 3)
          SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517
ENSG00000000003.14  29.87883  21.13732  30.40148  23.50615  38.90846  31.20858
ENSG00000000005.5  0.00000  0.00000  0.00000  0.00000  0.00000  0.00000
ENSG00000000419.12  44.97429  54.55385  47.13428  44.61619  44.52030  51.50911
          SRR1039520 SRR1039521
ENSG00000000003.14  36.73055  25.60277
ENSG00000000005.5  0.00000  0.00000
ENSG00000000419.12  43.01582  49.56923
> head(txr$length, 3)
          SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517
ENSG00000000003.14 1869.3397 1871.3781 1924.5532 2029.1670 2001.5240 1919.7958
ENSG00000000005.5  783.9375  783.9375  783.9375  783.9375  783.9375  783.9375
ENSG00000000419.12  797.2506  796.3534  839.3260  895.9974  859.9590  827.7119
          SRR1039520 SRR1039521
ENSG00000000003.14 1969.7615 1973.4448
ENSG00000000005.5  783.9375  783.9375
ENSG00000000419.12  859.8489  854.7984
```

counts

TPMs

“ATL”
offsets

Even cooler: tximeta

	SampleName	cell	dex	albut	names	avgLength	Experiment	Sample
SRR1039508	GSM1275862	N61311	untrt	untrt	SRR1039508	126	SRX384345	SRS508568
SRR1039509	GSM1275863	N61311	trt	untrt	SRR1039509	126	SRX384346	SRS508567
SRR1039512	GSM1275866	N052611	untrt	untrt	SRR1039512	126	SRX384349	SRS508571
SRR1039513	GSM1275867	N052611	trt	untrt	SRR1039513	87	SRX384350	SRS508572
SRR1039516	GSM1275870	N080611	untrt	untrt	SRR1039516	120	SRX384353	SRS508575
SRR1039517	GSM1275871	N080611	trt	untrt	SRR1039517	126	SRX384354	SRS508576
SRR1039520	GSM1275874	N061011	untrt	untrt	SRR1039520	101	SRX384357	SRS508579
SRR1039521	GSM1275875	N061011	trt	untrt	SRR1039521	98	SRX384358	SRS508580
	BioSample				files			
SRR1039508	SAMN02422669				salmon/SRR1039508/quant.sf			
SRR1039509	SAMN02422675				salmon/SRR1039509/quant.sf			
SRR1039512	SAMN02422678				salmon/SRR1039512/quant.sf			
SRR1039513	SAMN02422670				salmon/SRR1039513/quant.sf			
SRR1039516	SAMN02422682				salmon/SRR1039516/quant.sf			
SRR1039517	SAMN02422673				salmon/SRR1039517/quant.sf			
SRR1039520	SAMN02422683				salmon/SRR1039520/quant.sf			
SRR1039521	SAMN02422677				salmon/SRR1039521/quant.sf			

Even cooler: tximeta

```
> st <- tximeta::tximeta(meta)
importing quantifications
reading in files with read_tsv
1 2 3 4 5 6 7 8
found matching transcriptome:
[ Gencode - Homo sapiens - release 29 ]
loading existing TxDb created: 2019-03-24 16:32:30
generating transcript ranges
fetching genome info
> sg <- tximeta::summarizeToGene(st)
loading existing TxDb created: 2019-03-24 16:32:30
obtaining transcript-to-gene mapping from TxDb
summarizing abundance
summarizing counts
summarizing length
```

Even cooler: tximeta

```
> st
class: RangedSummarizedExperiment
dim: 205870 8
metadata(5): tximetaInfo quantInfo countsFromAbundance txomeInfo txdbInfo
assays(3): counts abundance length
rownames(205870): ENST00000456328.2 ENST00000450305.2 ... ENST00000387460.2
ENST00000387461.2
rowData names(3): tx_id gene_id tx_name
colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
colData names(9): SampleName cell ... Sample BioSample
> sg
class: RangedSummarizedExperiment
dim: 58294 8
metadata(5): tximetaInfo quantInfo countsFromAbundance txomeInfo txdbInfo
assays(3): counts abundance length
rownames(58294): ENSG0000000003.14 ENSG0000000005.5 ... ENSG0000285993.1
ENSG0000285994.1
rowData names(1): gene_id
colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
colData names(9): SampleName cell ... Sample BioSample
```

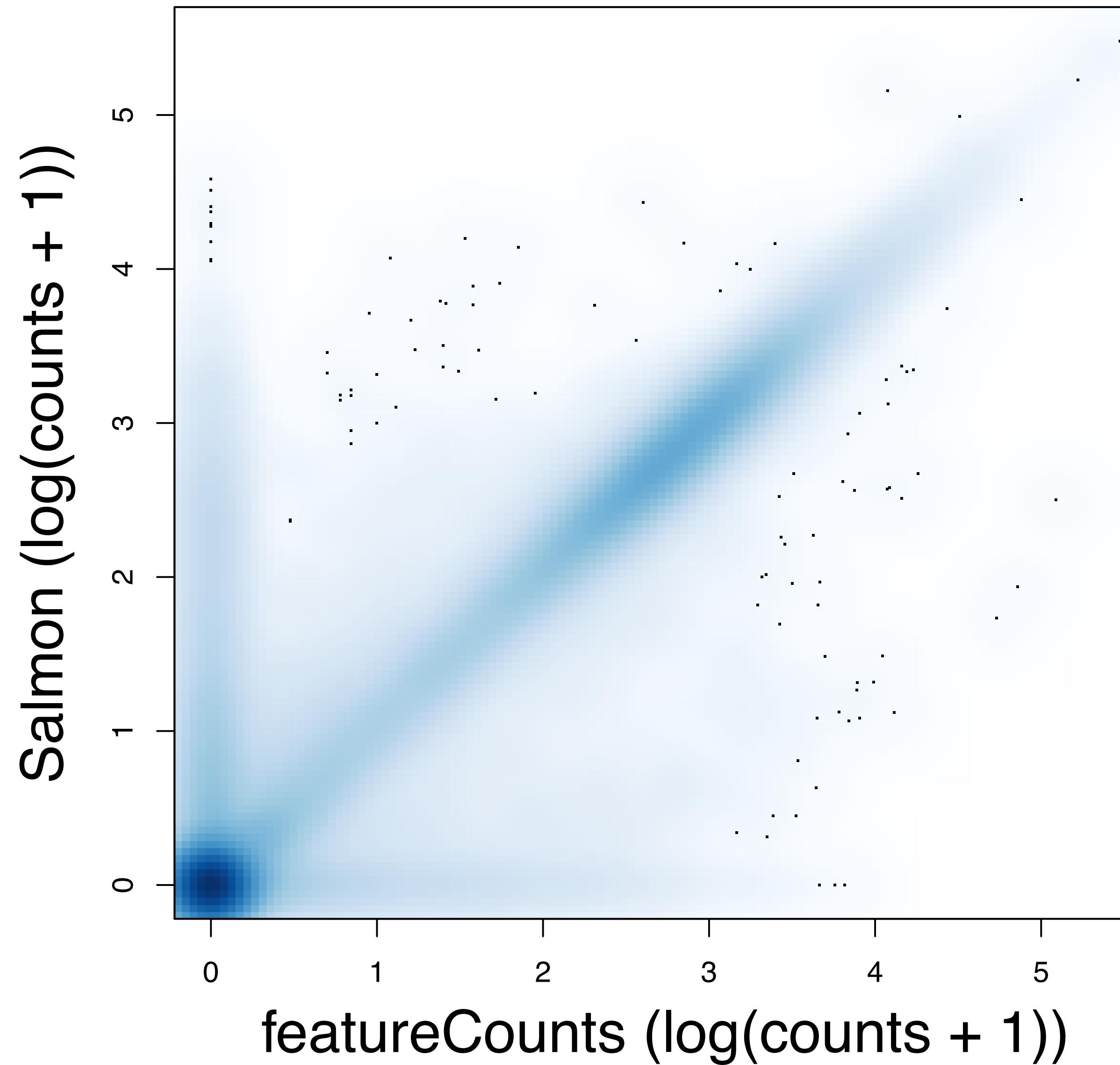
Comparison to alignment-based workflow

Alignment-free methods...

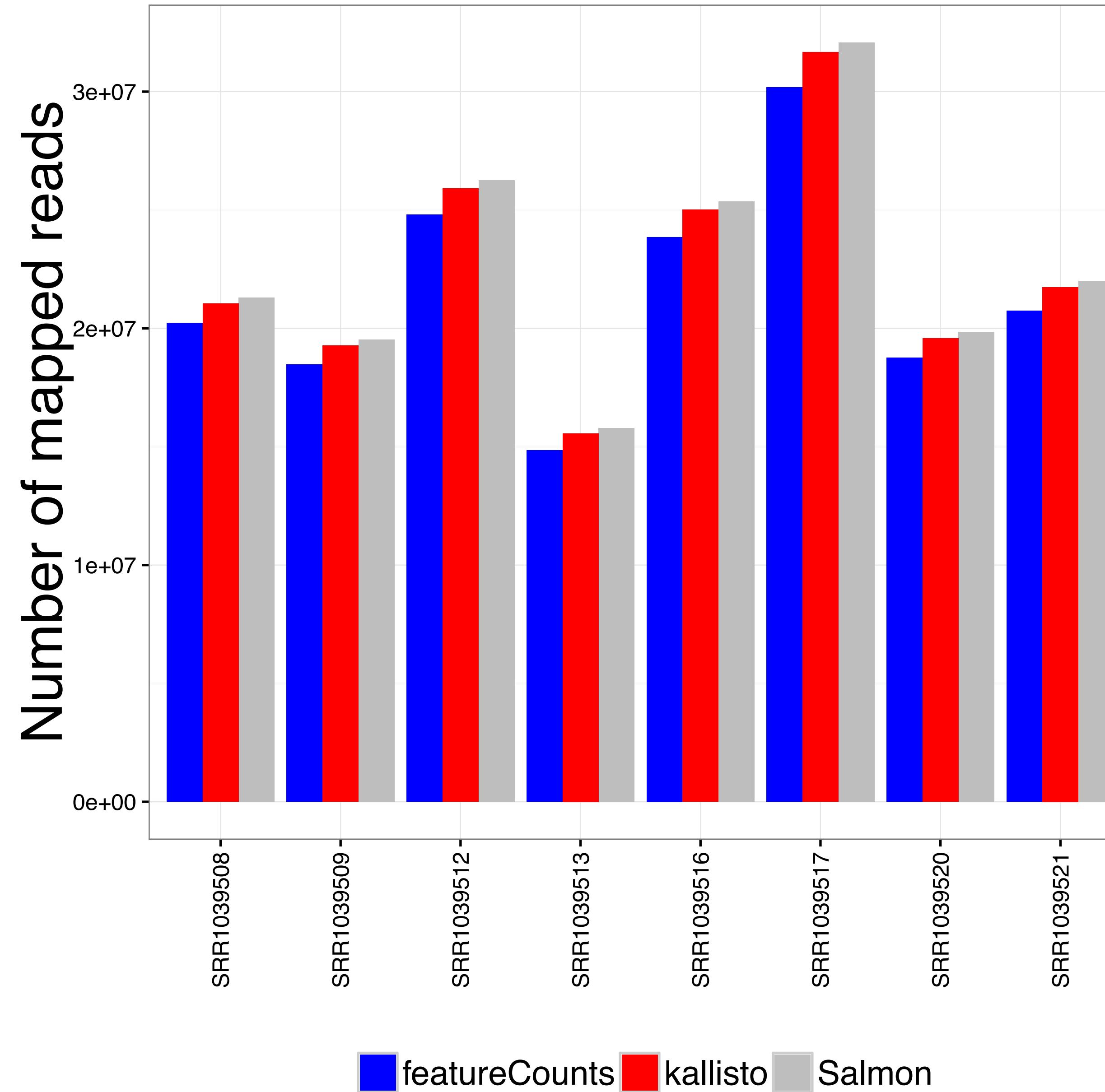
- ... are considerably faster than traditional alignment+counting -> allow bootstrapping
- ... provide more highly resolved estimates (transcripts rather than gene) - can be aggregated to gene level
- ... can use a larger fraction of the reads
- ... don't give precise alignments (for e.g. visualization in genome browser)
- but avoid large alignment files

Counts are overall similar between workflows

SRR1039508

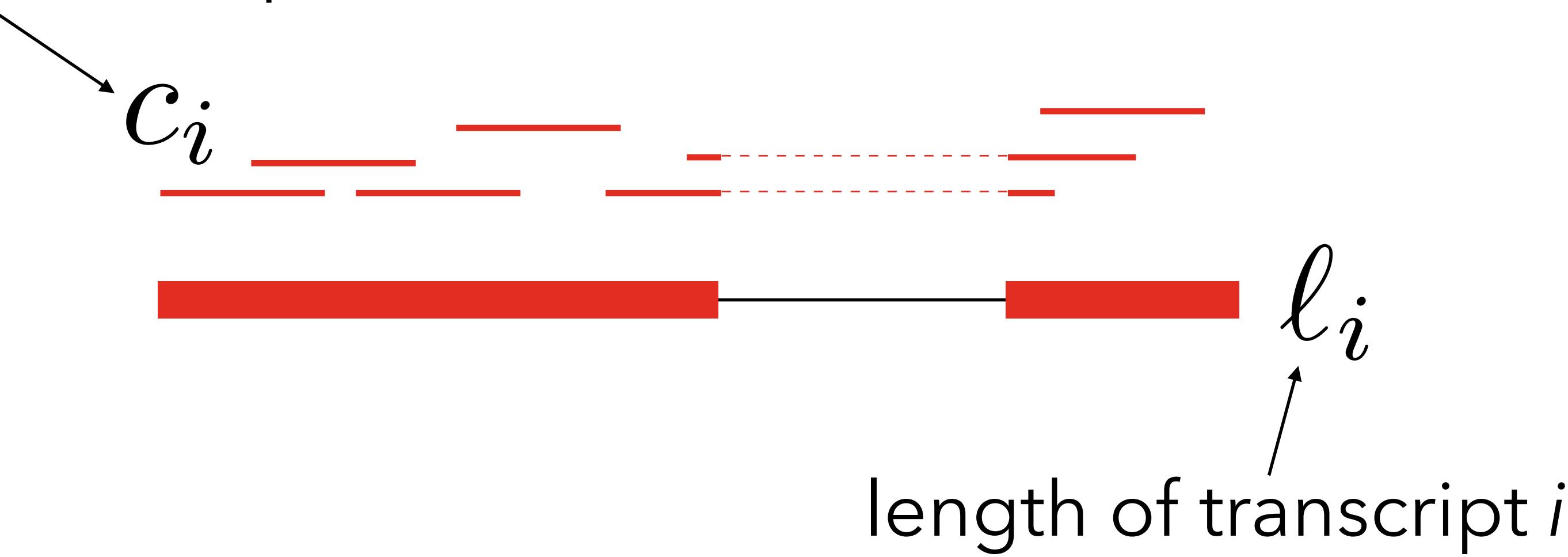


Alignment-free methods can use slightly more reads



Abundance units

read count for transcript i



Abundance units

read count for transcript i

c_i



$$t_i = \frac{c_i r}{\ell_i}$$

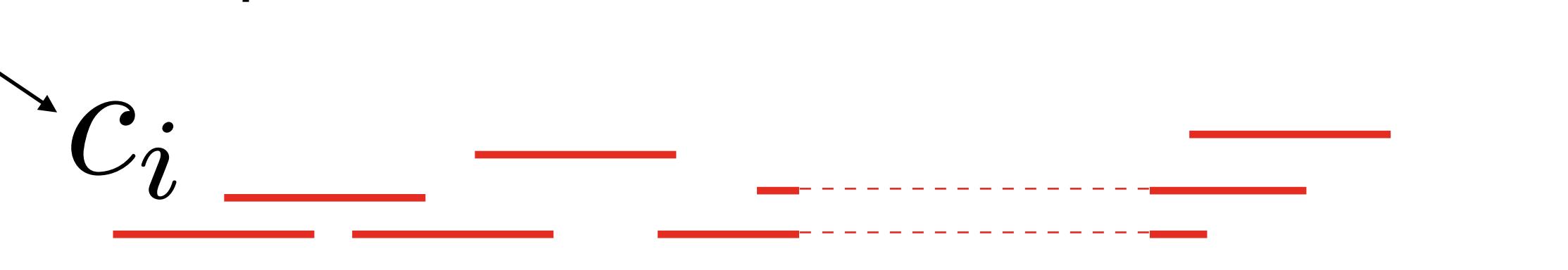
fragment length

ℓ_i

length of transcript i

Abundance units

read count for transcript i



fragment
length

$$t_i = \frac{c_i r}{\ell_i}$$

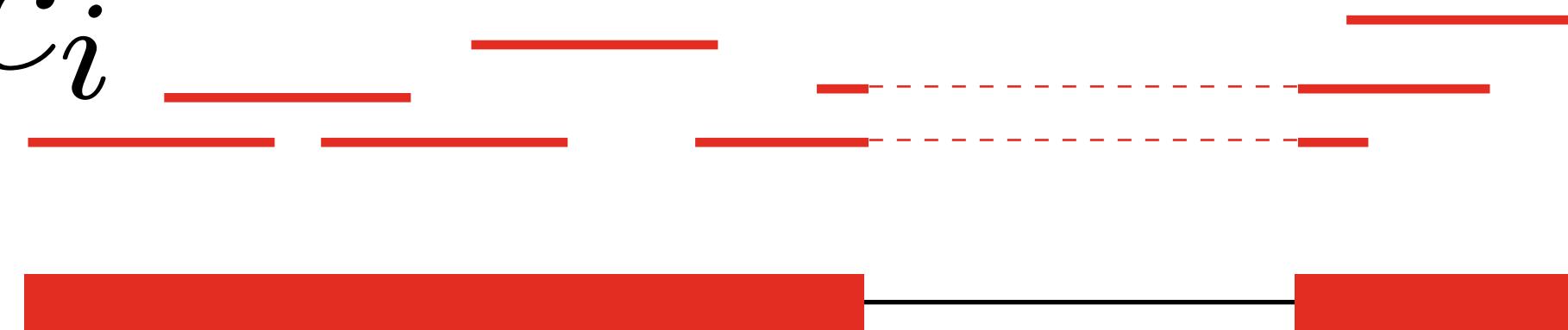
ℓ_i
length of transcript i

$$TPM_i = 10^6 \cdot \frac{t_i}{\sum_k t_k}$$

Abundance units

read count for transcript i

c_i



fragment
length

$$t_i = \frac{c_i r}{\ell_i}$$

ℓ_i
length of transcript i

$$TPM_i = 10^6 \cdot \frac{t_i}{\sum_k t_k}$$

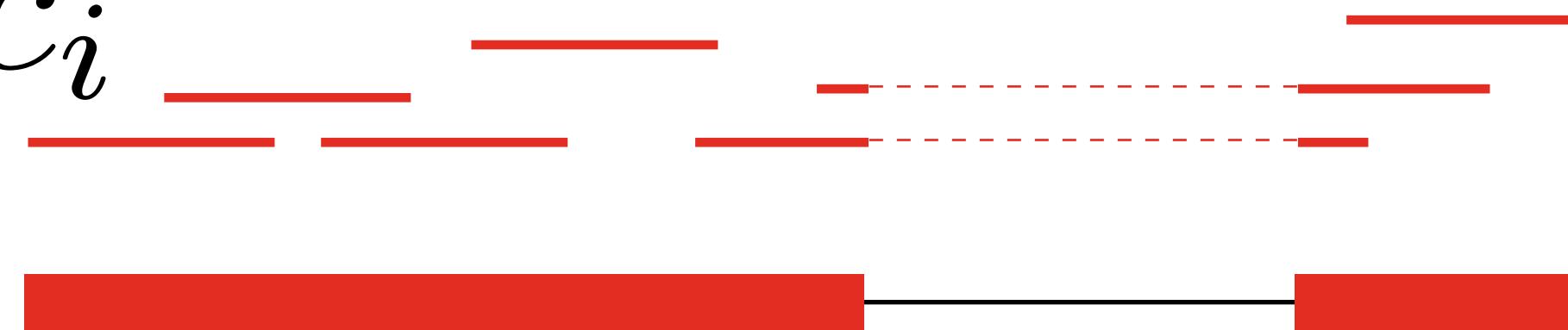
library size

$$RPKM_i = 10^9 \cdot \frac{c_i}{\ell_i \boxed{\sum_k c_k}} = 10^9 \cdot \frac{t_i}{\sum_k (t_k \ell_k)}$$

Abundance units

read count for transcript i

c_i



fragment
length

$$t_i = \frac{c_i r}{\ell_i}$$

ℓ_i
length of transcript i

$$TPM_i = 10^6 \cdot \frac{t_i}{\sum_k t_k}$$

library size

$$RPKM_i = 10^9 \cdot \frac{c_i}{\ell_i \boxed{\sum_k c_k}} = 10^9 \cdot \frac{t_i}{\sum_k (t_k \ell_k)}$$

$$TPM_i \propto RPKM_i$$

$$\sum_i TPM_i = 10^6$$