Differential gene expression analysis using RNA-seq

Applied Bioinformatics Core, November 2019

Friederike Dündar with Luce Skrabanek & Paul Zumbo

Day 3: Counting reads

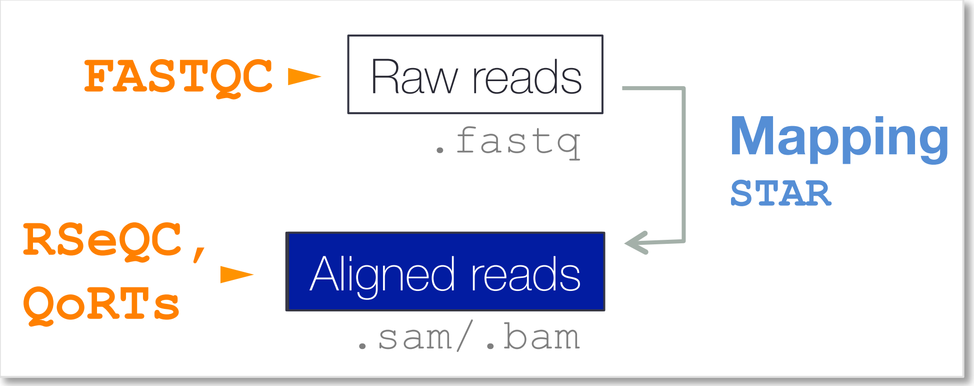
1. Storing aligned reads: **SAM/BAM format**

2. **QC** of aligned reads

3. **counting** reads and quantifying gene expression across different samples
    - working with read counts
    - normalizing
    - transforming

4. **similarity assessments/exploratory analyses**
    - hierarchical clustering
    - PCA

Recap week 1



- We **downloaded fastq.gz** files from the SRA via ENA using wget
- We did **QC** of the raw reads using **FastQC** (1x per sample) and summarized the results for the numerous fastq files per sample it using **MultiQC**
- We **aligned** the raw reads using **STAR** (the genome index that is necessary was provided by us)
- We will do **additional QC** on those BAM files

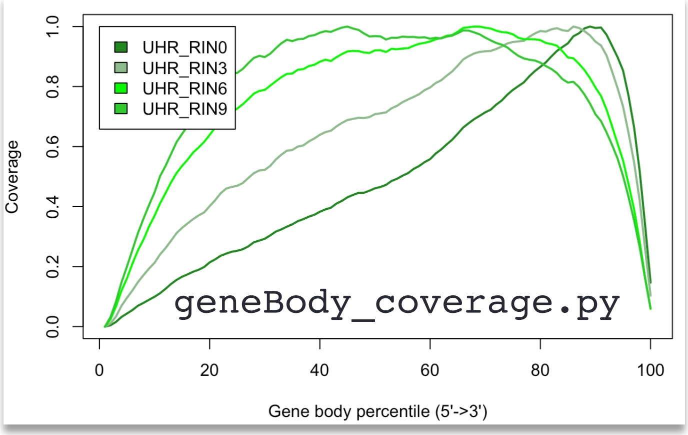
QC recap

- **raw reads** QC
  - adapter/primer/other contaminating and over-represented sequences
  - sequencing quality
  - GC distributions
  - duplication levels

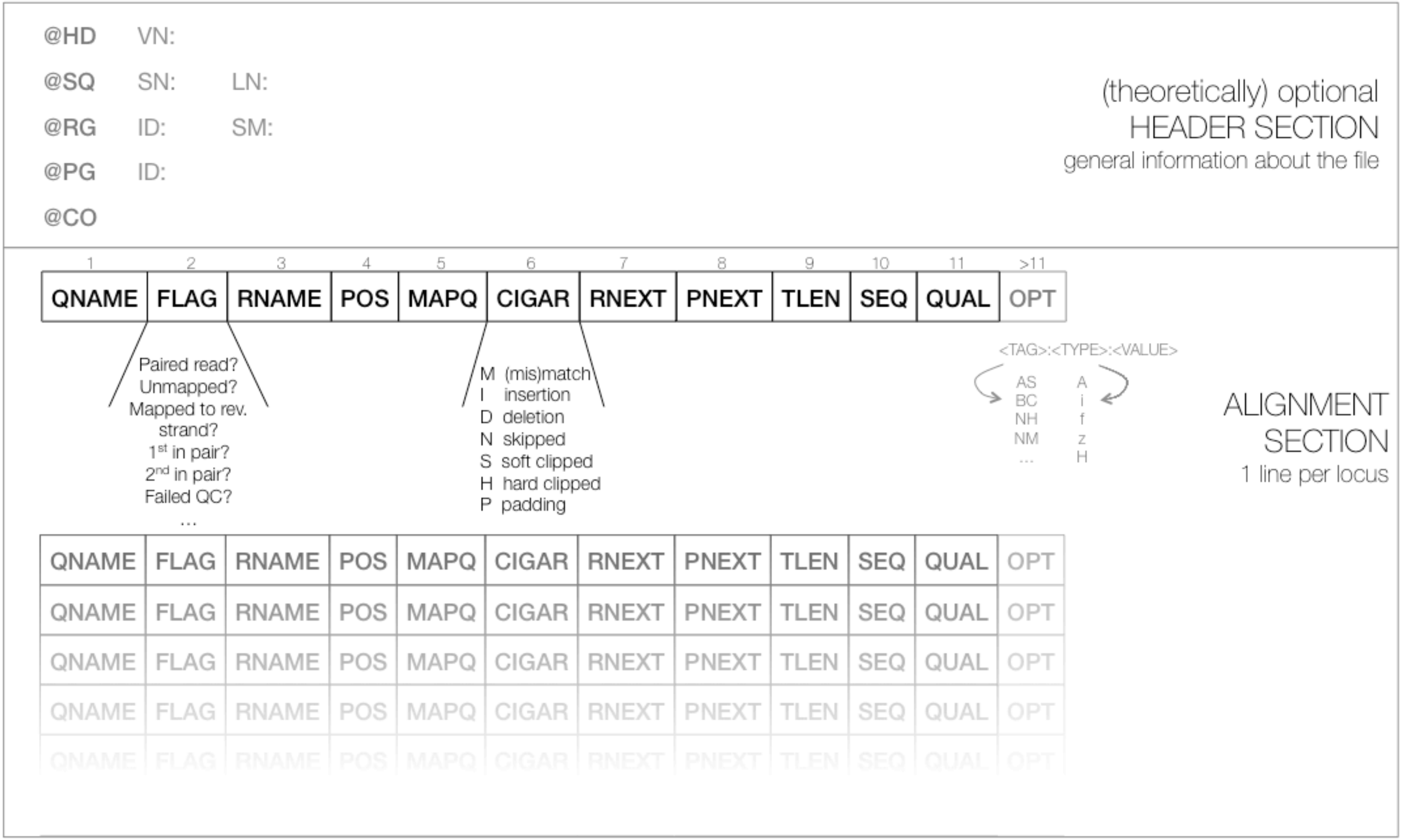
FastQC (QoRTs)

- **aligned reads** QC
  - % (uniquely) aligned reads
  - % exonic vs. intronic/intergenic
  - gene diversity
  - gene body coverage

➢ aligner's log files
➢ samtools flagstat
➢ RSeQC
➢ QoRTs
…
summarize with MultiQC!



Storing aligned reads: SAM/BAM



@HD VN:
@SQ SN: LN:
@RG ID: SM:
@PG ID:
@CO

(theoretically) optional
HEADER SECTION
general information about the file

1	2	3	4	5	6	7	8	9	10	11	>11
QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT

Paired read?
Unmapped?
Mapped to rev. strand?
1st in pair?
2nd in pair?
Failed QC?
...

M (mis)match
I insertion
D deletion
N skipped
S soft clipped
H hard clipped
P padding

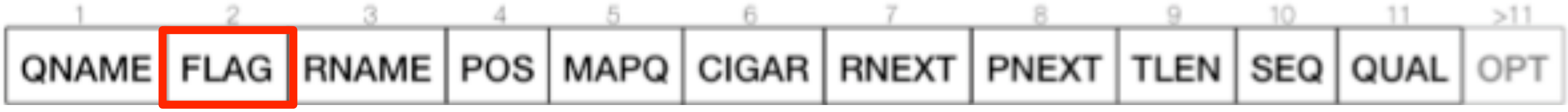
<TAG>:<TYPE>:<VALUE>

AS	A
BC	i
NH	f
NM	z
...	H

ALIGNMENT
SECTION
1 line per locus

QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT
QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT
QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT
QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT
QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT

Storing aligned reads: SAM/BAM



1	2	3	4	5	6	7	8	9	10	11	>11
QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT

2nd field: binary FLAG

Binary (Decimal)	Hex	Description
00000000001 (1)	0x1	Is the read paired?
00000000010 (2)	0x2	Are both reads in a pair mapped “properly” (i.e., in the correct orientation with respect to one another)?
00000000100 (4)	0x4	Is the read itself unmapped?
00000001000 (8)	0x8	Is the mate read unmapped?
00000010000 (16)	0x10	Has the read been mapped to the reverse strand?
00000100000 (32)	0x20	Has the mate read been mapped to the reverse strand?
00001000000 (64)	0x40	Is the read the first read in a pair?
00010000000 (128)	0x80	Is the read the second read in a pair?
00100000000 (256)	0x100	Is the alignment not primary? (A read with split matches may have multiple primary alignment records.)
01000000000 (512)	0x200	Does the read fail platform/vendor quality checks?
10000000000 (1024)	0x400	Is the read a PCR or optical duplicate?

most common FLAGS for SR: 0; 4; 16

```
https://broadinstitute.github.io/
picard/explain-flags.html
```

Storing aligned reads: SAM/BAM

1	2	3	4	5	6	7	8	9	10	11	>11
QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT

6th field: CIGAR string – which hoops did the aligner have to jump through to align the read to the genome locus that it thought was the best fit?

M	alignment (match or **mis**match!!)
I (N)	insertion (large insertions)
D	deletion
S/H	clipping

← spliced out introns = sequences are missing in the read, i.e., they need to be inserted in order to align the read to the genome

	Reference sequence with aligned reads	CIGAR string	Explanation
	C T G C A T G T T A G A T A A * * G A T A G C T G T G C T A		
reads	A A G G A T A * C T G	1M2I4M1D3M	Insertion & Deletion
	G A T A A * G G A T A	5M1P1I4M	Padding & Insertion
	T G T T A ▬ T G C T A	5M13N5M	Spliced read
	a a a C A T G T T A G	3S8M	Soft clipping
	~~A~~ ~~A~~ ~~A~~ C A T G T T A G	3H8M	Hard clipping

Storing aligned reads: SAM/BAM

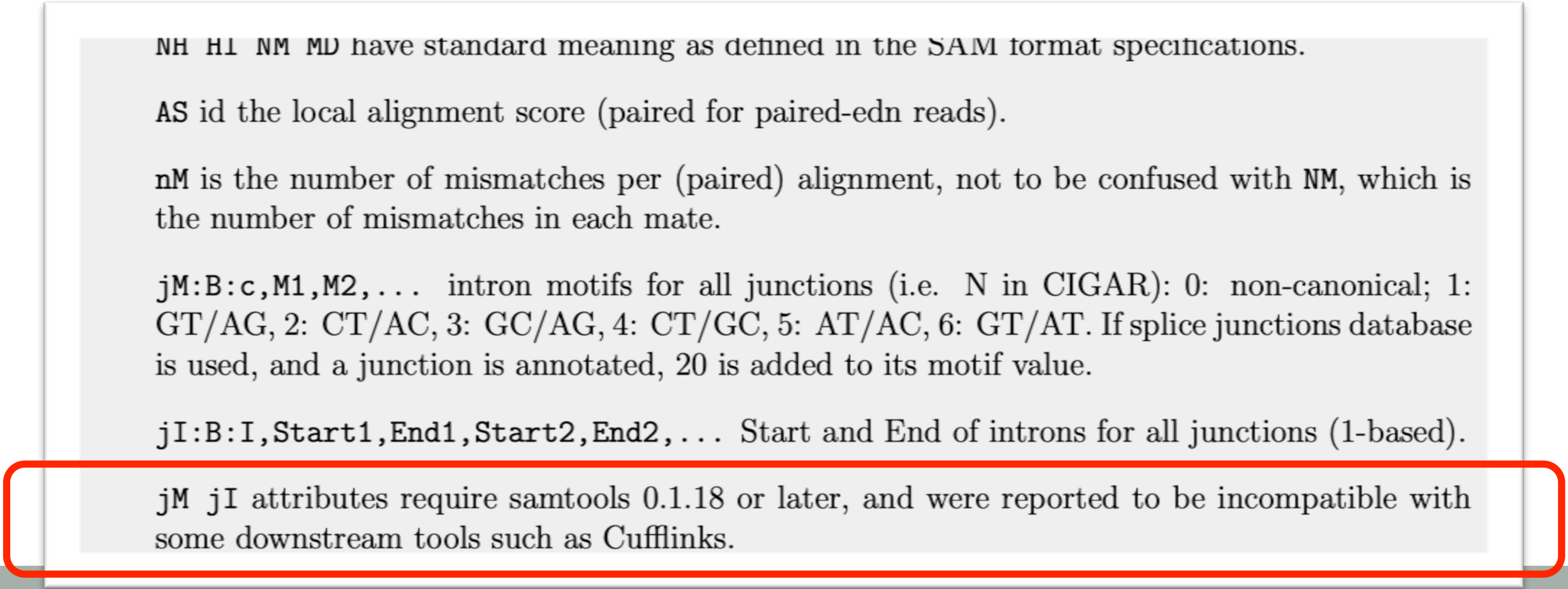
1	2	3	4	5	6	7	8	9	10	11	>11
QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL	OPT

after 11th field: OPTIONAL information

<TAG>:<TYPE>:<VALUE>

tags are not standardized!

AS:i	Alignment score
BC:Z	Barcode sequence
HI:i	Query is *i*-th hit stored in the file
NH:i	Number of reported alignments for the query sequence
NM:i	Edit distance of the query to the reference
MD:Z	String that contains the exact positions of mismatches (should complement the CIGAR string)
RG:Z	Read group (should match the entry after ID if @RG is present in the header.



NH HI NM MD have standard meaning as defined in the SAM format specifications.

AS id the local alignment score (paired for paired-edn reads).

nM is the number of mismatches per (paired) alignment, not to be confused with NM, which is the number of mismatches in each mate.

jM:B:c,M1,M2,... intron motifs for all junctions (i.e. N in CIGAR): 0: non-canonical; 1: GT/AG, 2: CT/AC, 3: GC/AG, 4: CT/GC, 5: AT/AC, 6: GT/AT. If splice junctions database is used, and a junction is annotated, 20 is added to its motif value.

jI:B:I,Start1,End1,Start2,End2,... Start and End of introns for all junctions (1-based).

jM jI attributes require samtools 0.1.18 or later, and were reported to be incompatible with some downstream tools such as Cufflinks.

QC of aligned reads

- How many reads aligned?
  - aligner output (e.g., Log.final.out, `STAR`'s log file)
- How well did the reads align?
  - `samtools flagstat,` RSeQC's `bam_stat`
  - these provide summaries of the `FLAG` field values
- Did we capture mostly exonic RNA?
  - RSeQC's `read_distribution.py`, QoRTS
- Do we see a pronounced 3'/5' bias?
  - `RSeQC's geneBody_coverage.py`, QoRTS

(almost) all of these results can be summarized using MultiQC!
→ Section 3.4.1 of the course notes

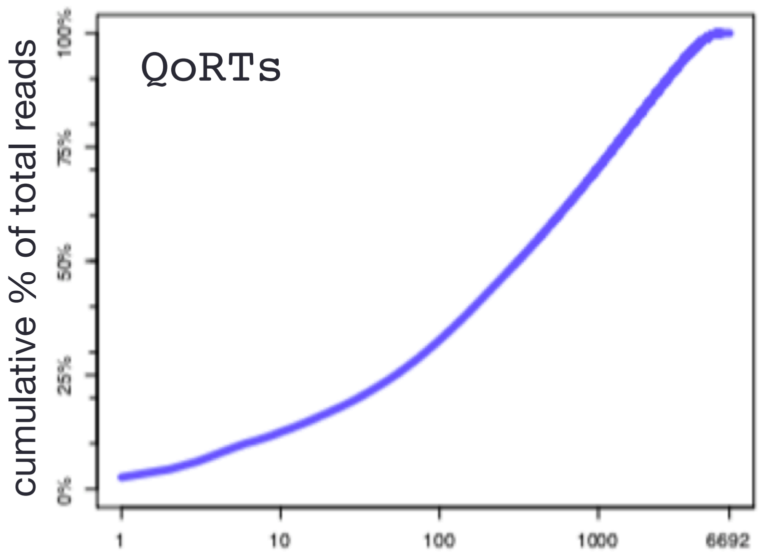
visual inspection!

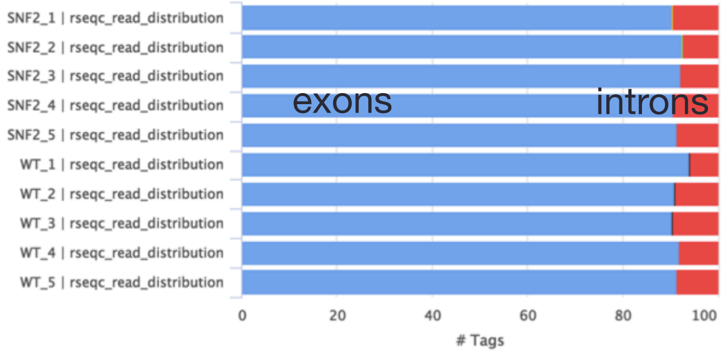
Typical biases of RNA-seq

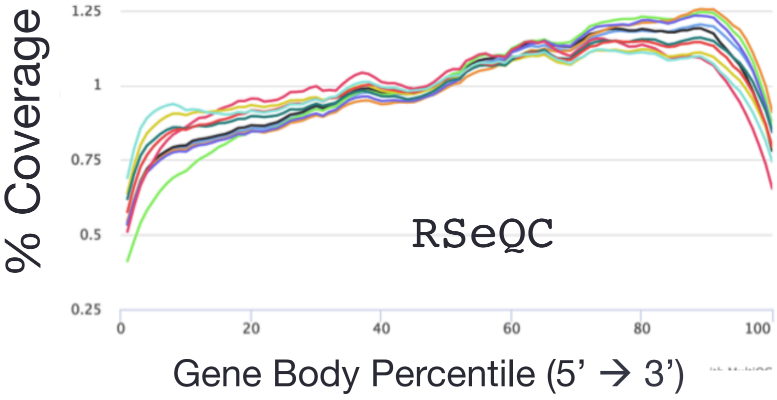
- lack of **gene diversity**:
  - dominance of rRNAs, tRNAs or other highly abundant transcripts

- **read distribution**
  - high intron coverage: incomplete poly(A) enrichment
  - many intergenic reads: gDNA contamination

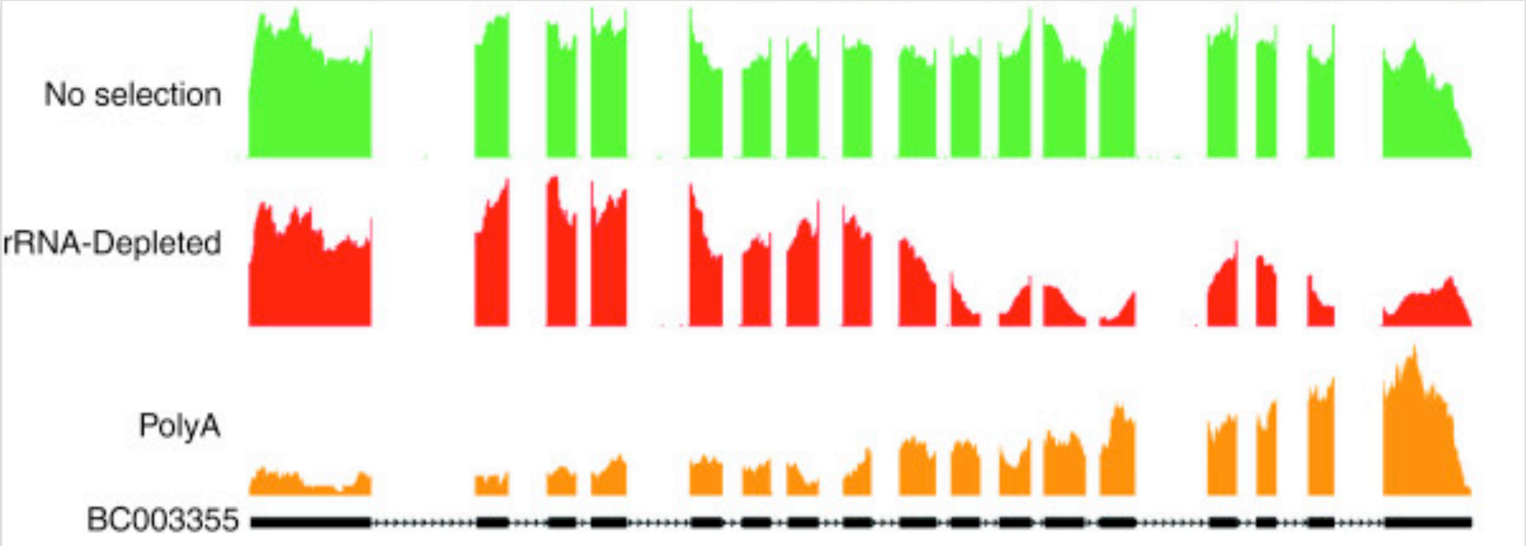
- **gene body coverage**
  - 3’ bias: RNA degradation + poly(A) enrichment

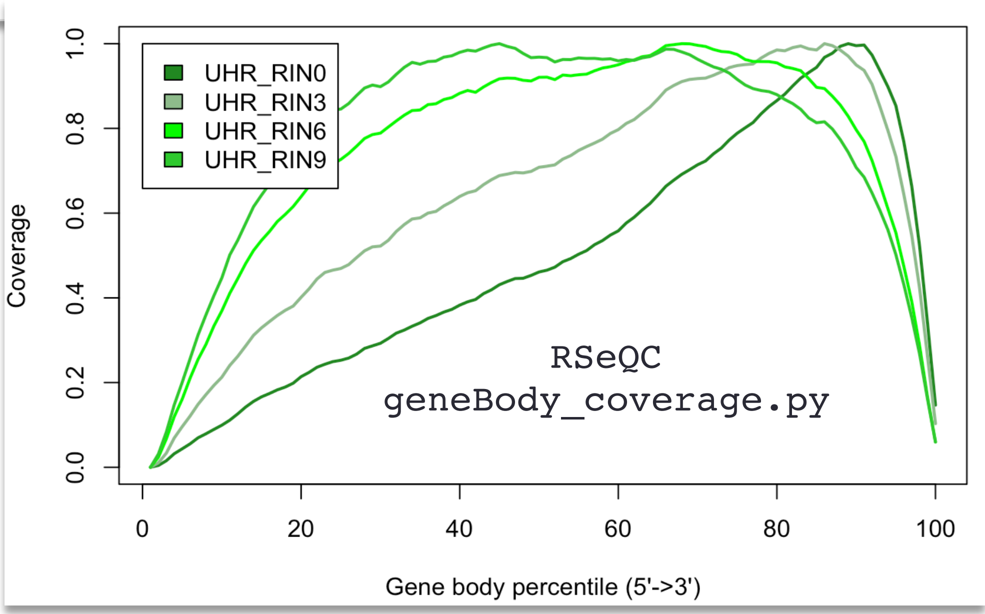






Different protocols have different gene body coverage bias





RSeQC
geneBody_coverage.py

Lahens et al. (2014) Genome Biology 15:R86

2 popular post-alignment QC packages

RSeQC

- commands are not well standardized
- output is not standardized either (text, R scripts, PDF)
- most results can be integrated with the help of MultiQC
- see Table 11 of the course notes for a list of relevant RSeQC scripts (mostly: `read_distribution` and `geneBody_coverage.py`)

`http://rseqc.sourceforge.net/`

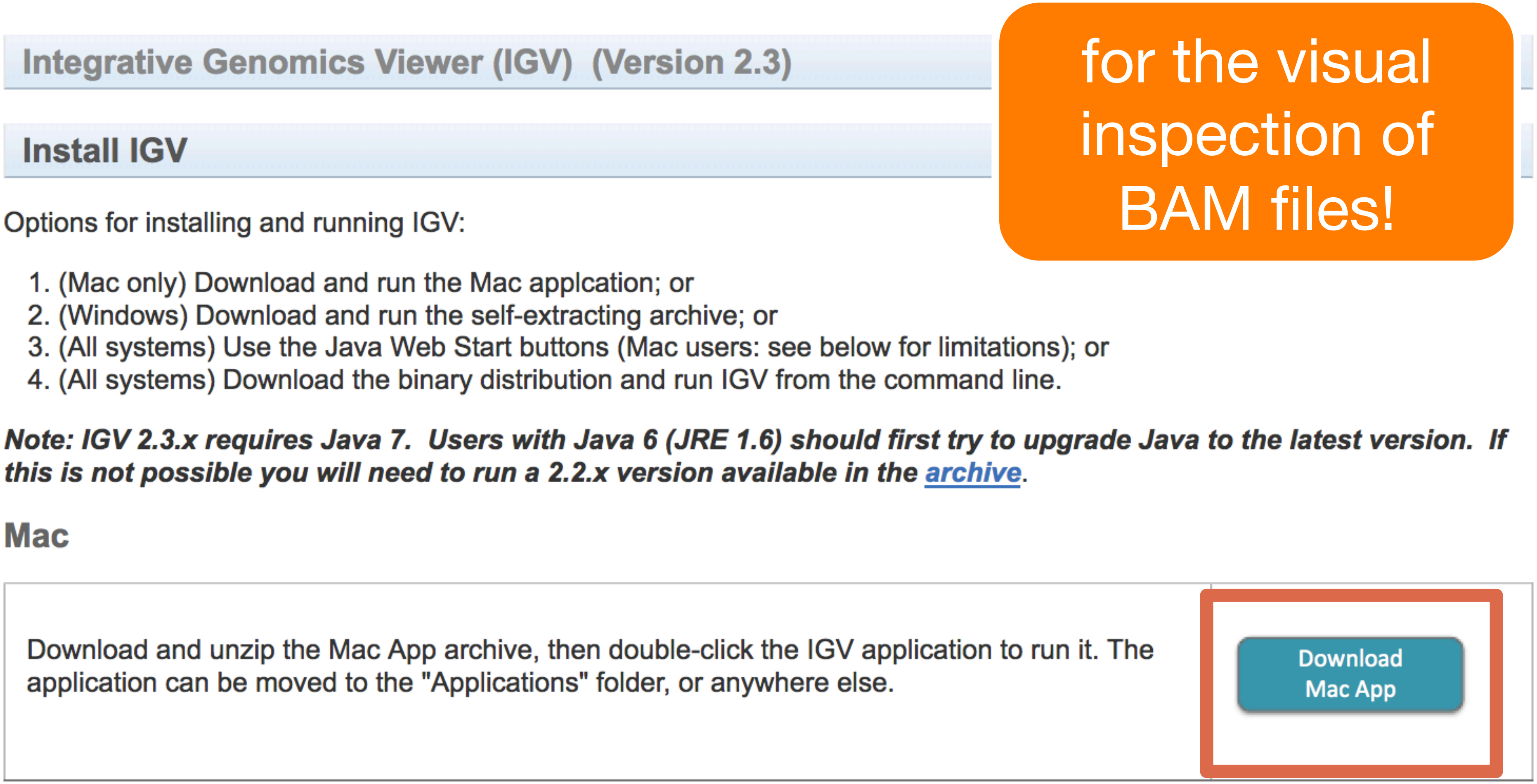
QoRTs

- less clunky than RSeQC
- offers many checks that are already part of FastQC
- stratifies genes by expression strength for many checks
- gene diversity plot is very useful!
- can bundle numerous samples into one PDF, but may run for a long (!) time
- output is not easily integrated with MultiQC

`https://hartleys.github.io/QoRTs/`

Integrative Genomics Viewer

http://software.broadinstitute.org/software/igv/download



Integrative Genomics Viewer (IGV) (Version 2.3)

Install IGV

Options for installing and running IGV:

1. (Mac only) Download and run the Mac applcation; or
2. (Windows) Download and run the self-extracting archive; or
3. (All systems) Use the Java Web Start buttons (Mac users: see below for limitations); or
4. (All systems) Download the binary distribution and run IGV from the command line.

Note: IGV 2.3.x requires Java 7. Users with Java 6 (JRE 1.6) should first try to upgrade Java to the latest version. If this is not possible you will need to run a 2.2.x version available in the archive.

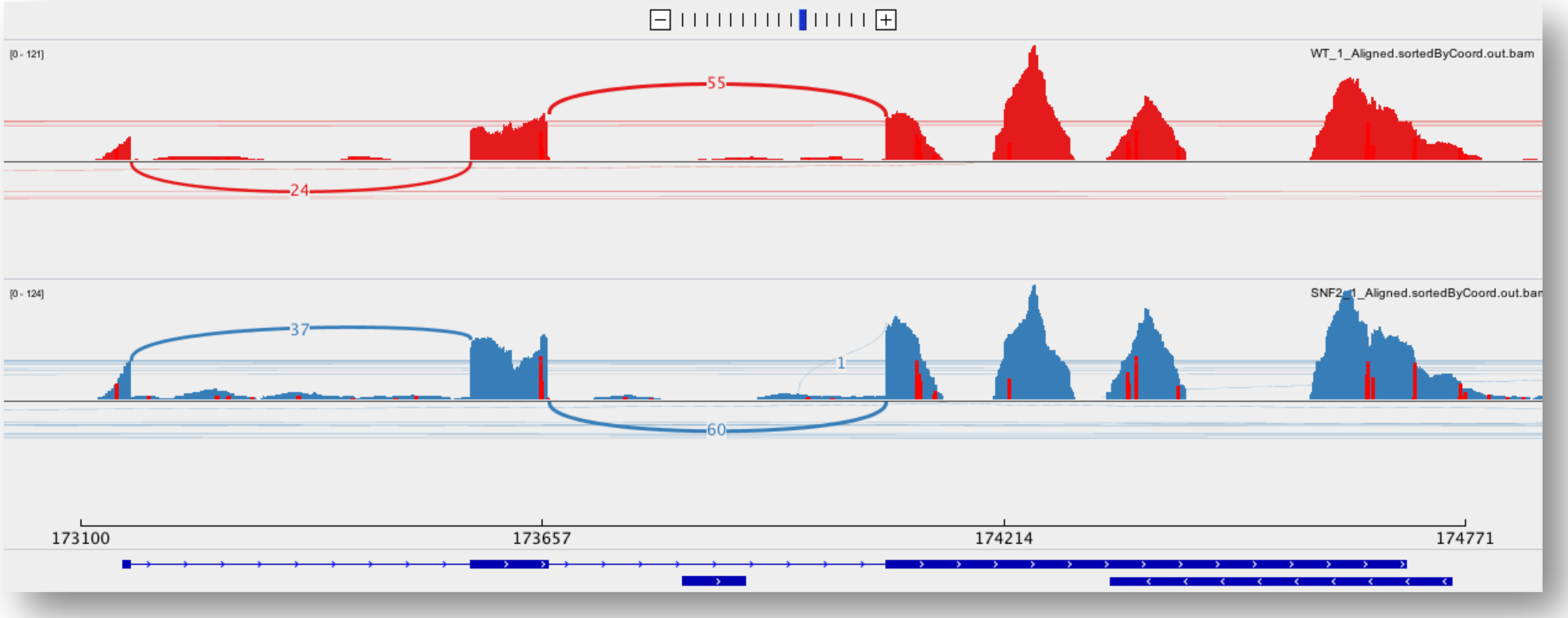
Mac

Download and unzip the Mac App archive, then double-click the IGV application to run it. The application can be moved to the "Applications" folder, or anywhere else.

for the visual inspection of BAM files!

Integrative Genomics Viewer

- **load BAM file(s)** from your computer ("File")
- take a **snapshot** of the reads around gene ***YPL198W***


[0 - 121]
WT_1_Aligned.sortedByCoord.out.bam
55
24
[0 - 124]
SNF2_1_Aligned.sortedByCoord.out.bam
37
1
60
173100
173657
174214
174771

starting with IGV 2.3, **Sashimi plots** can easily be created

http://software.broadinstitute.org/software/igv/Sashimi

many options!

Katz (2015). Bioinformatics. doi:10.1093/bioinformatics/btv034

Summary

- aligning unspliced reads is not too difficult, but it still takes a long time (depending on the size of the genome)

- spliced reads are quite tricky, and identifying novel splice junctions is error-prone and far from being solved

- the file format for storing aligned reads (SAM/BAM) is fairly standardized, but the optional fields (and how alignment tools interpret some of the mandatory entries) leave lots of room for variability

- the file format(s) for storing genome annotation (e.g. genes, transcripts) tend to be even stricter defined and even less well followed (aka it’s a mess!)

- historically, `samtools` are the most widely used tools when it comes to exploring and manipulating `SAM/BAM` files (although there are alternatives, e.g. `bamtools`)

- **QC of aligned read files is at least as important as QC of the raw reads, if not more so!**

removing rRNAs

Can be done at virtually every step of the analysis. Choose the version that makes most sense to you.

- **sortMeRNA**: http://bioinfo.lifl.fr/RNA/sortmerna/
    - input: reads in fastq file + rRNA sequences
    - will extract those reads that do not match to the rRNA sequences
    - https://www.ncbi.nlm.nih.gov/nuccore/U13369 (human rRNA), https://www.ncbi.nlm.nih.gov/nuccore/BK000964 (mouse)

raw reads filtering

- make a **“genome” index for rRNAs only** (and perhaps tRNAs), then align your reads and only use those that do not map for the next round of alignment

alignment-based filtering

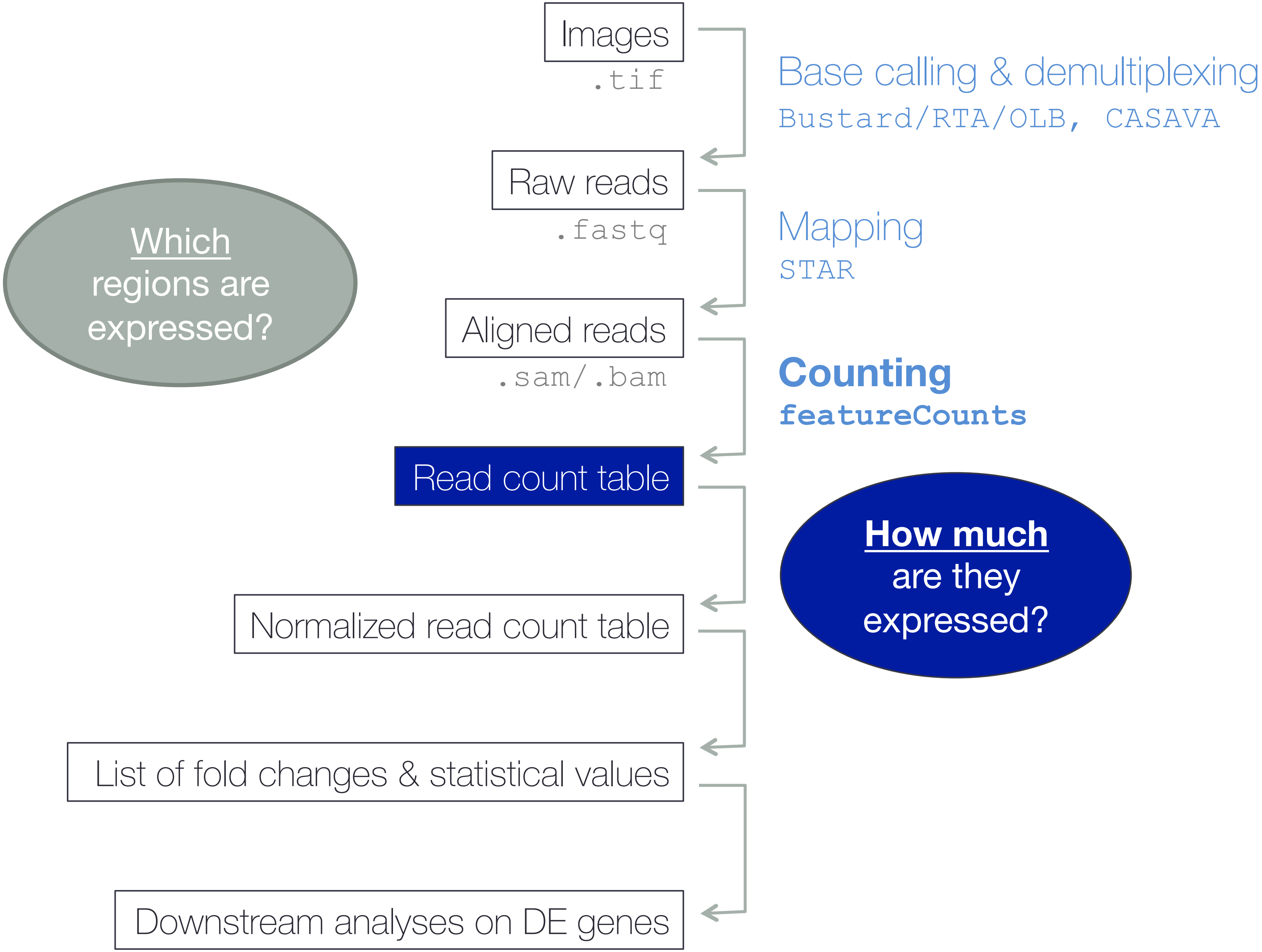
- do your alignment and counting as is, simply **ignore the rRNA genes** in your subsequent downstream analysis

ignoring information about some RNA classes

COUNTING READS

from alignments to count tables

Bioinformatics workflow of RNA-seq analysis



Images
.tif

Base calling & demultiplexing
`Bustard/RTA/OLB, CASAVA`

Raw reads
.fastq

Mapping
`STAR`

Aligned reads
.sam/.bam

Counting
`featureCounts`

Read count table

Normalized read count table

List of fold changes & statistical values

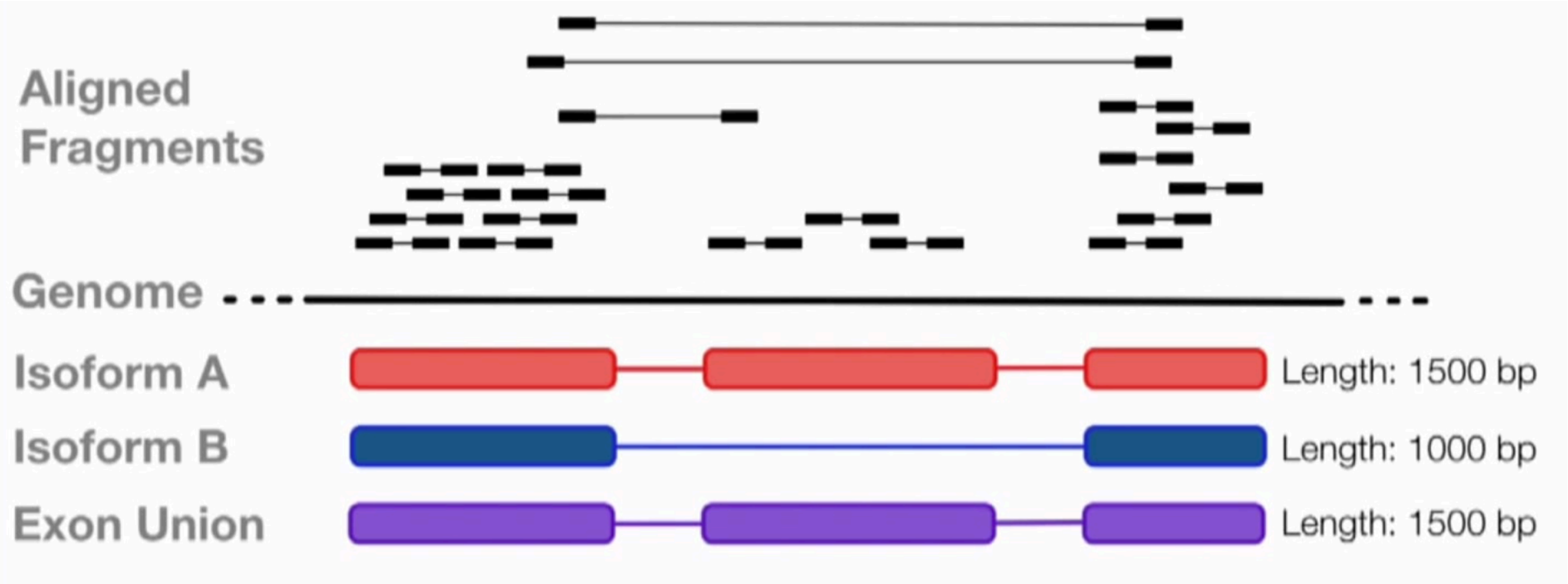
Downstream analyses on DE genes

Which regions are expressed?

How much are they expressed?

Quantifying expression

genes != transcripts



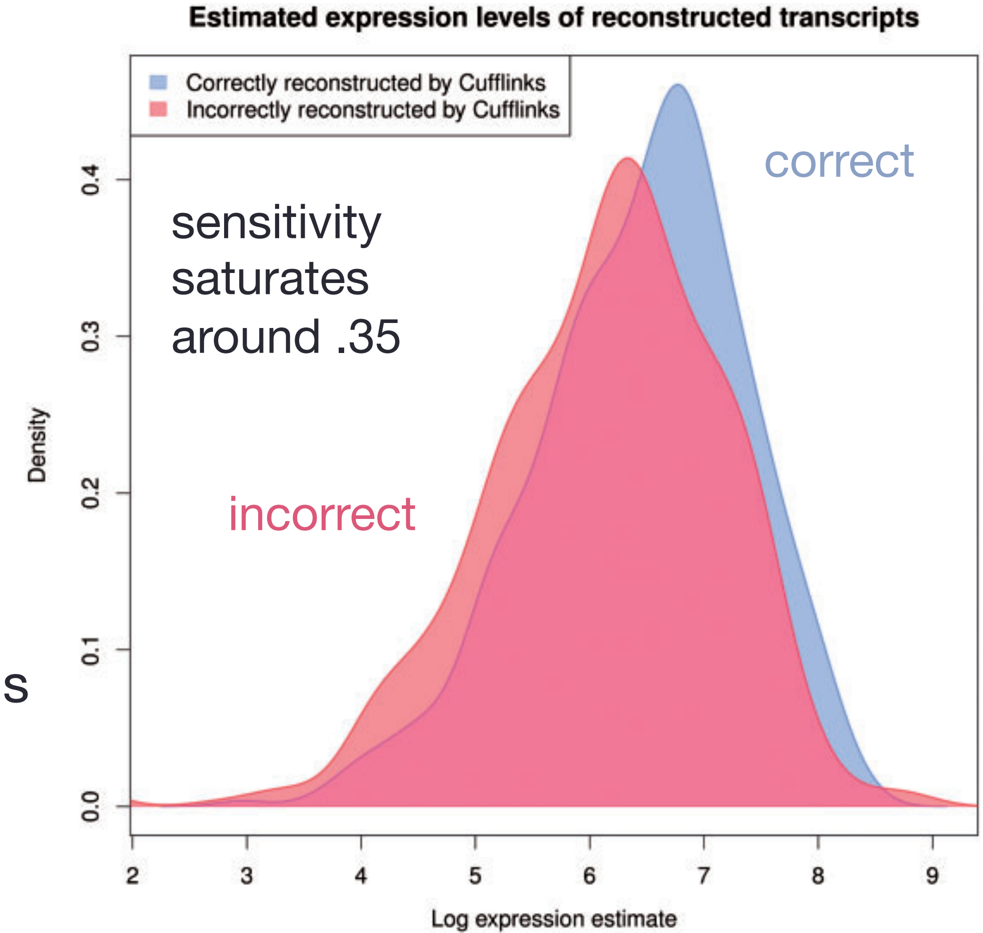
Disclaimer:
There are 2 (maybe 3) schools of thought when it comes to how expression values should be generated. We currently present the one that's based on the **raw reads and gene overlaps.** See the course notes for references for the other strategies' arguments.

Van den Berge, K. et al. (2019). doi: 10.1146/annurev-biodatasci-072018-021255

Please don’t rely on transcriptome reconstruction unless you really need to

This includes Cufflinks!

- Transcriptome reconstruction suffers from **bad precision** and **bad sensitivity** => many FP transcripts (esp. for tricky transcriptomes)!
- False transcripts capture a considerable portion of the reads



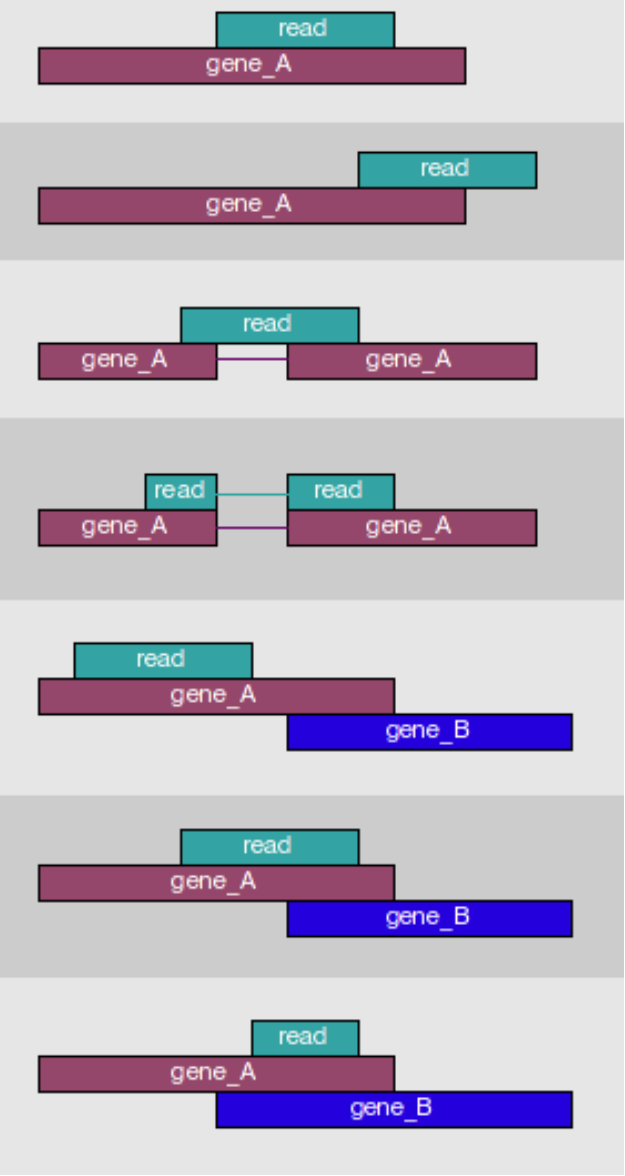
sensitivity saturates around .35

instead of transcript reconstruction, perhaps resort to either one of these alternatives:

- transcript quantification with **pseudo-alignments** → kallisto, salmon
- **exon counts** → DEXSeq
- focus on **specific splice events** → MISO

Janes et al. (2015). Briefings in Bioinformatics, (January), 1–9. doi:10.1093/bib/bbv007

Counting read–gene overlaps



`featureCounts` will use read-gene overlaps as small as 1 bp

multi-overlap reads will be discarded

http://www-huber.embl.de/users/anders/HTSeq/doc/count.html#count

Let's count some reads & read the results into R!

Please save the .RData and the commands!

NORMALIZING READ COUNTS

from counts to expression value estimates

From counting reads to expression units

- **Raw counts:** number of reads (or fragments) overlapping with the union of exons of a gene

X_i

raw counts != expression strength

strongly influenced by:

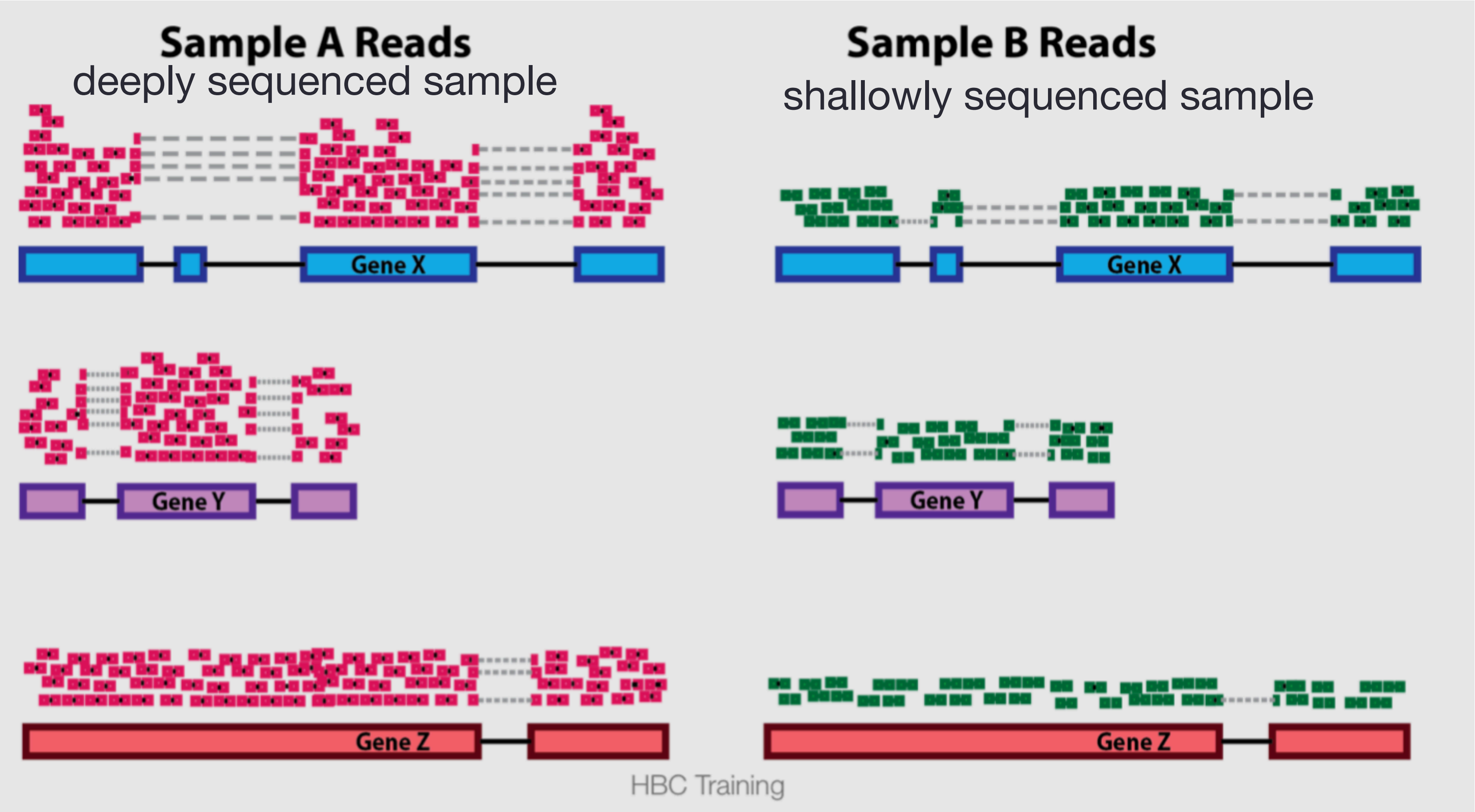
- gene length
- transcript sequence (% GC)
- sequencing depth
- expression of all other genes in the same sample

may cause variations for **different genes** expressed at the same level

may cause variations for the **same gene** in different samples

Influences on read count numbers

Sequencing depth, i.e. total number of reads/sample

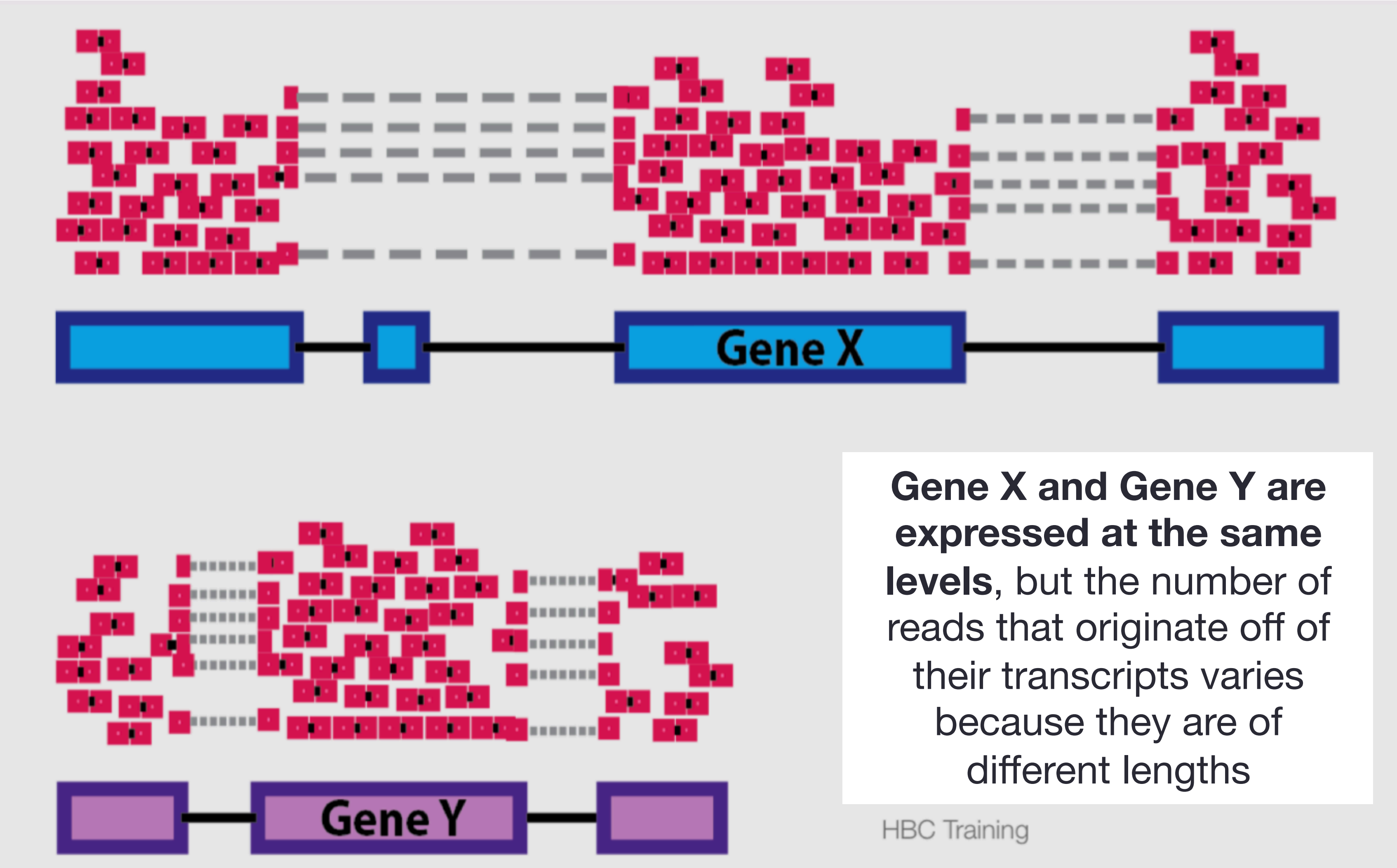


seq. depth of Sample A >> Sample B automatically leads to larger counts for the genes of Sample A even if the expression levels are the same

illustrations from https://github.com/hbctraining/DGE_workshop

Influences on read count numbers

Gene lengths (and GC bias)



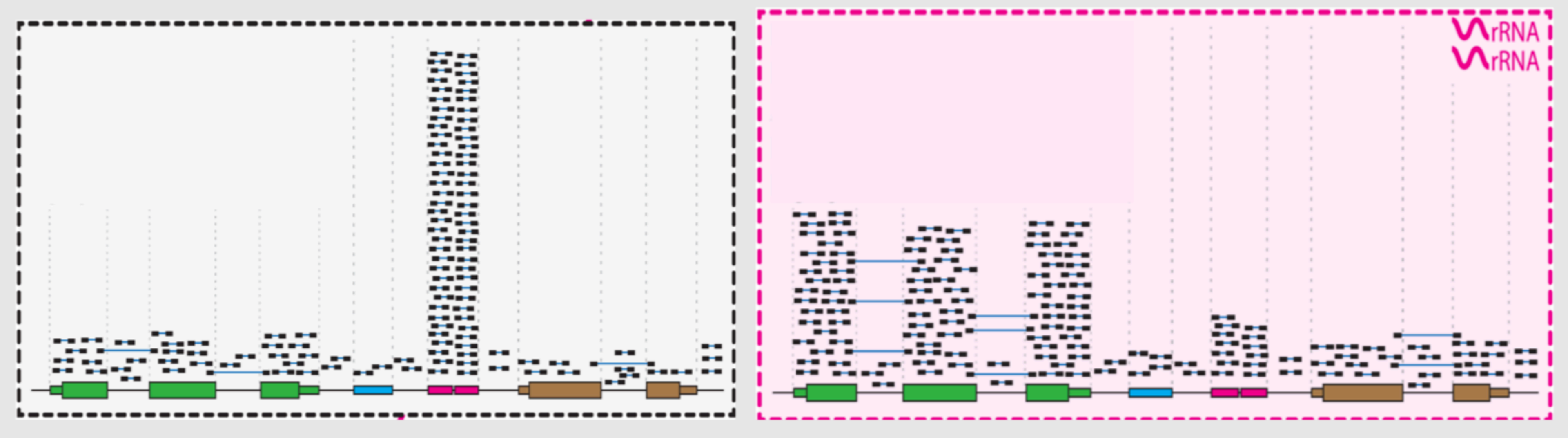
Gene X and Gene Y are expressed at the same levels, but the number of reads that originate off of their transcripts varies because they are of different lengths

HBC Training

Influences on read count numbers

RNA pool composition/library diversity

the reads assigned to individual genes depend on the number of reads that are allocated to all other transcripts in the same sample



one (or more) **very abundant transcript** makes up a significant portion of all reads
→ dynamic range for the remaining transcripts is limited

in the absence of that abundant transcript ("read sponge"), the remaining transcripts' expression differences have a greater chance of being detected

Influences of read count numbers
Summary

GENE-SPECIFIC

- gene length
- transcript sequence (% GC)

need to be corrected when comparing different **genes**

SAMPLE-SPECIFIC

- sequencing depth
- expression of all other genes within the same sample

need to be corrected when comparing the same gene across different **samples**

Different expression units you will hear about

- **Raw counts**: number of reads/fragments overlapping with the union of exons of a gene

$$X_i$$

- **[RF]PKM**: Reads/Fragments per Kilobase of gene per Million reads mapped

$$RPKM_i = \frac{X_i}{(\frac{l_i}{10^3})(\frac{N}{10^6})}$$

gene length seq. depth

- **TPM**: Transcripts Per Million

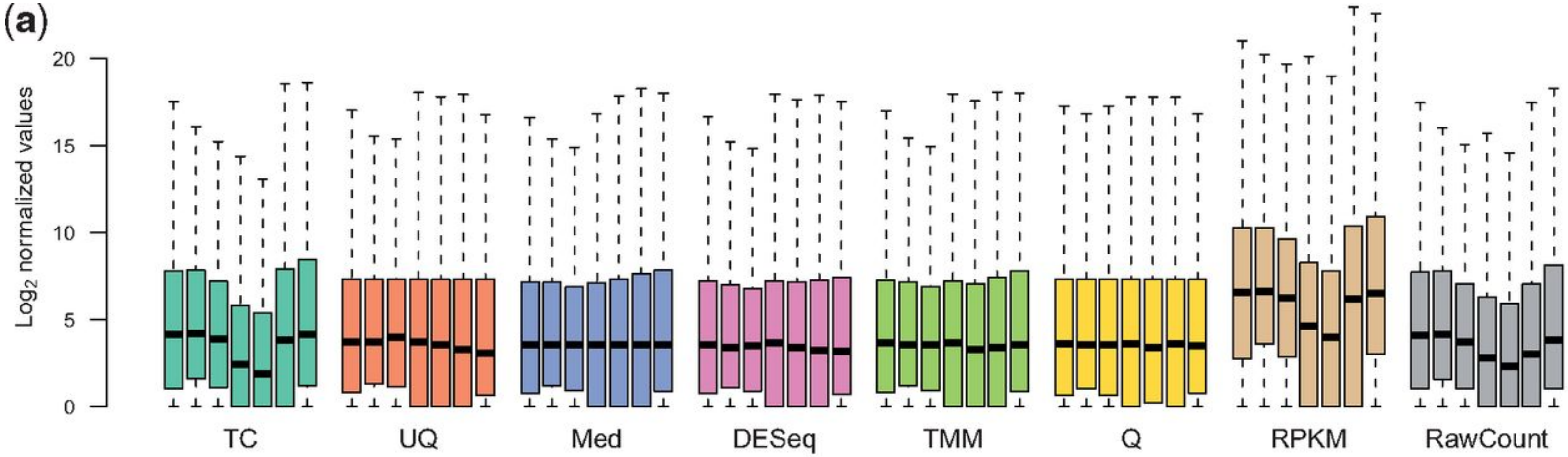
$$TPM_i = \frac{X_i}{l_i} * \frac{1}{\sum_j \frac{X_j}{l_k}} * 10^6$$

gene read counts per bp

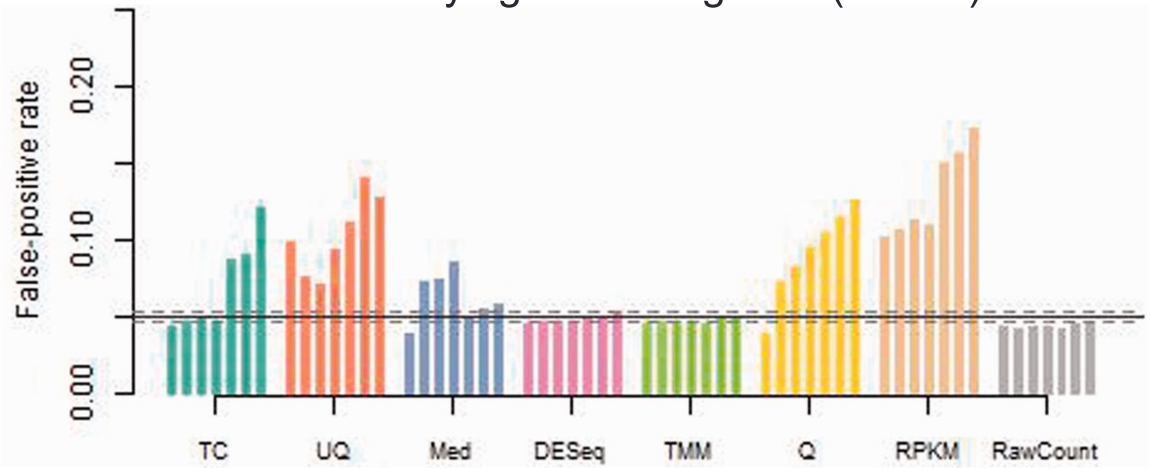
all gene counts over *all* gene bp

- **rlog**: log2-transformed count data normalized for small counts and library size (DESeq2)

Effects of normalization methods on FC calculation and DGE analysis



FP rates for varying % of DE genes (0-30%)



Avoid **[RF]PKM** and **total read count** normalization for DGE!

if you need normalized expression values, use **TPM or DESeq's rlog**

Dillies et al.(2012). Briefings in Bioinformatics. doi:10.1093/bib/bbs046

rlog values of DESeq2

- **Normalization** for differences in sequencing depth & sample composition
  - median of the ratios of the j-th sample's counts to those of the mean expression of each gene across all samples
- **variance-stabilization** to alleviate the heteroskedasticity of the normalized read counts
- **log2-transformation** to compact the range and bring it closer to normally distributed values

The rlog values are good (but far from perfect!) proxies of the “real” expression strength of a given gene across different samples.

These are the values that you should use for exploratory analyses and visualizations!

Let’s normalize (+ variance stabilize + transform) some reads & explore in R!

Please save the .RData and the commands!