RNA Sequencing and Data Processing

National Aeronautics and Space Administration

NASA

GL4U: RNAseq 2024

Amanda M. Saravia-Butler, Ph.D.

NASA GeneLab Science Lead

Contractor: KBR

BPS

Biological & **P**hysical **S**ciences

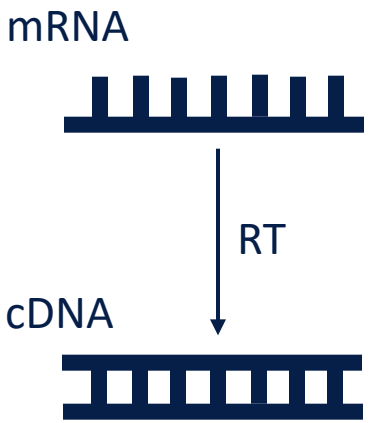
What Came Before RNA Sequencing?

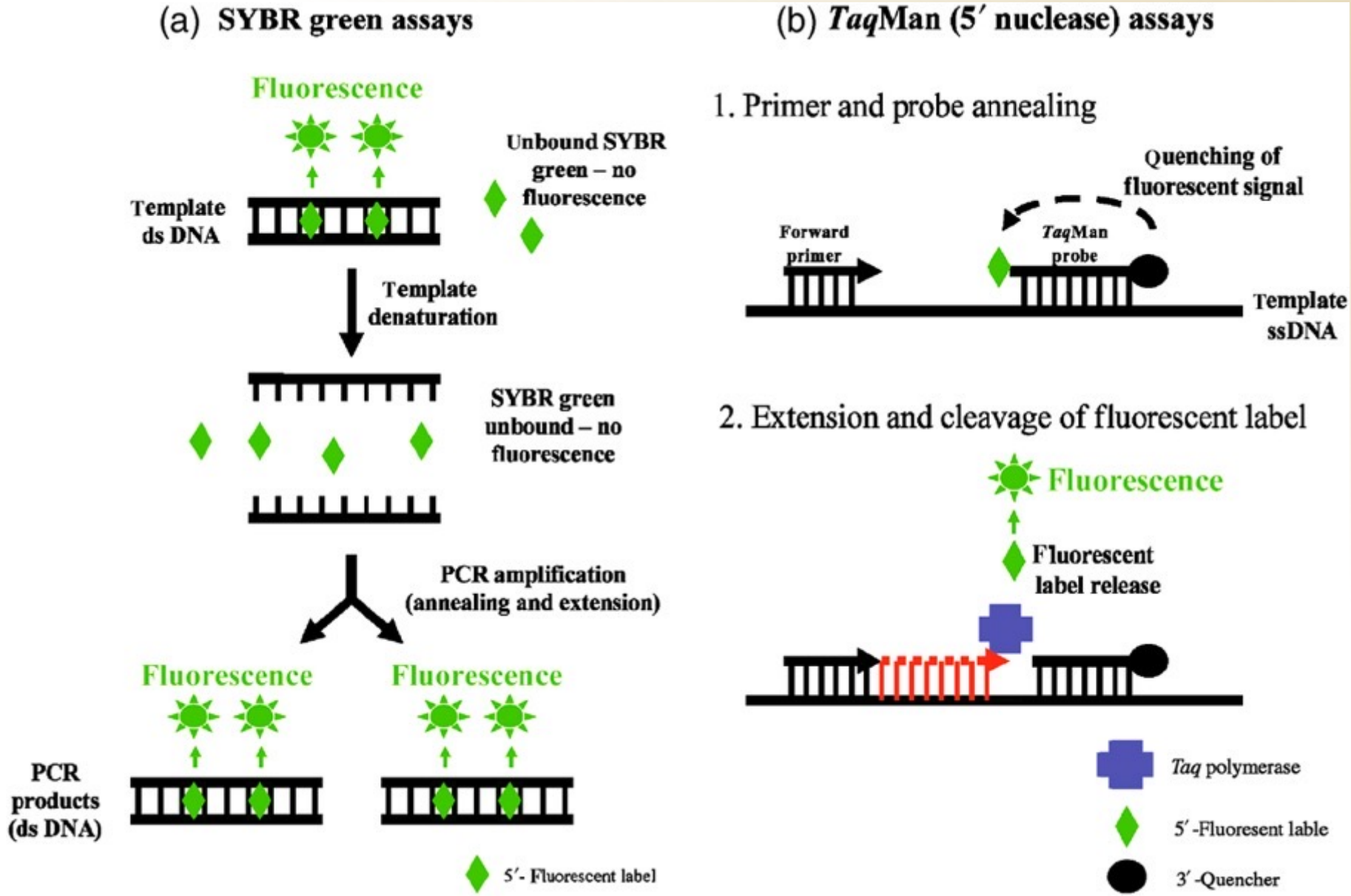
Polymerase Chain Reaction (PCR)

Kary Mullis

Awesome YouTube video of PCR: https://www.youtube.com/watch?v=iQsu3Kz9NYo

Reverse Transcription Quantitative PCR (RTqPCR)

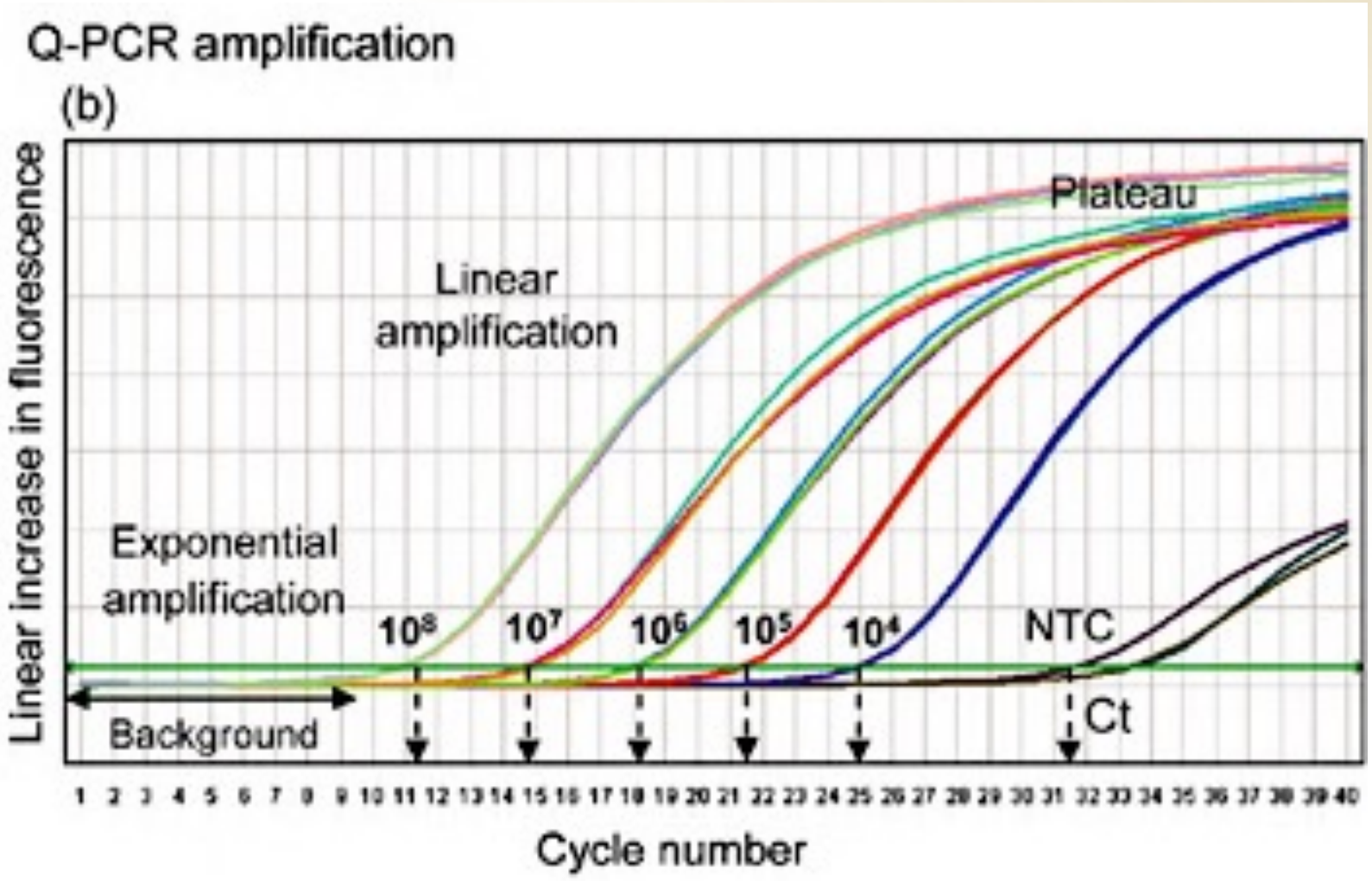




https://www.youtube.com/watch?v=YhXj5Yy4ksQ

FEMS Microbiology Ecology, Volume: 67, Issue: 1, Pages: 6-20, First published: 09 December 2008, DOI: (10.1111/j.1574-6941.2008.00629.x)

Reverse Transcription Quantitative PCR (RTqPCR)

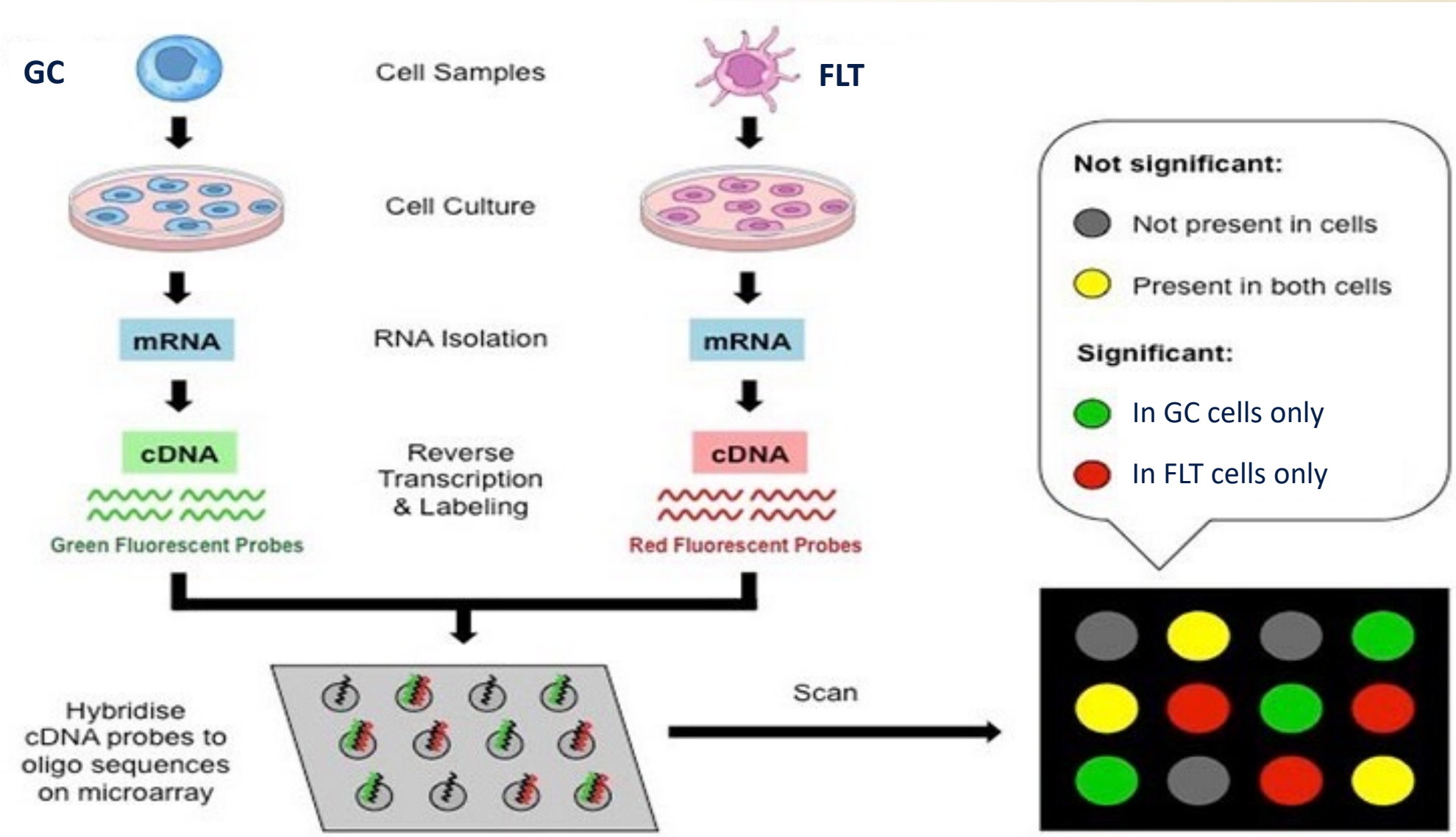


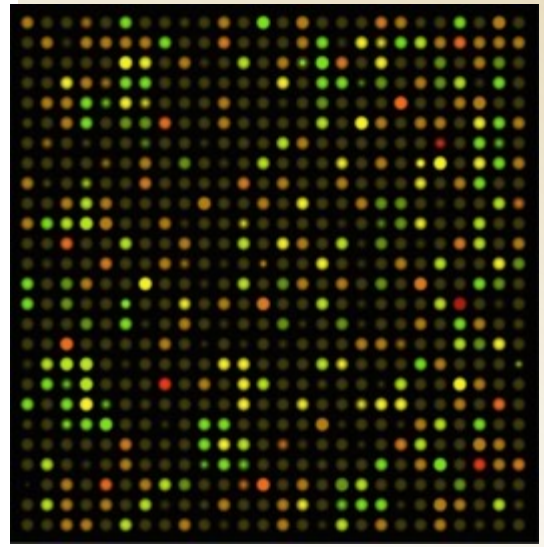
FEMS Microbiology Ecology, Volume: 67, Issue: 1, Pages: 6-20, First published: 09 December 2008, DOI: (10.1111/j.1574-6941.2008.00629.x)

Limitations of RTqPCR

- Only annotated genes can be evaluated
- Potential for non-specific binding (SYBR green)
- Requires primer optimization
- 3 conserved and unique regions are required for TaqMan (probe, F and R primers)
- Requires housekeeping genes (genes with consistent expression irrespective of treatment) for normalization
- ‘Absolute’ quantification of the target gene requires a standard curve, increasing the likelihood for issues if the standard curve is not properly generated
- Limited number of samples and number of genes that can be evaluated simultaneously
- All samples should be run in technical triplicates further limiting the ‘n’ number
- Others?

Microarray Overview





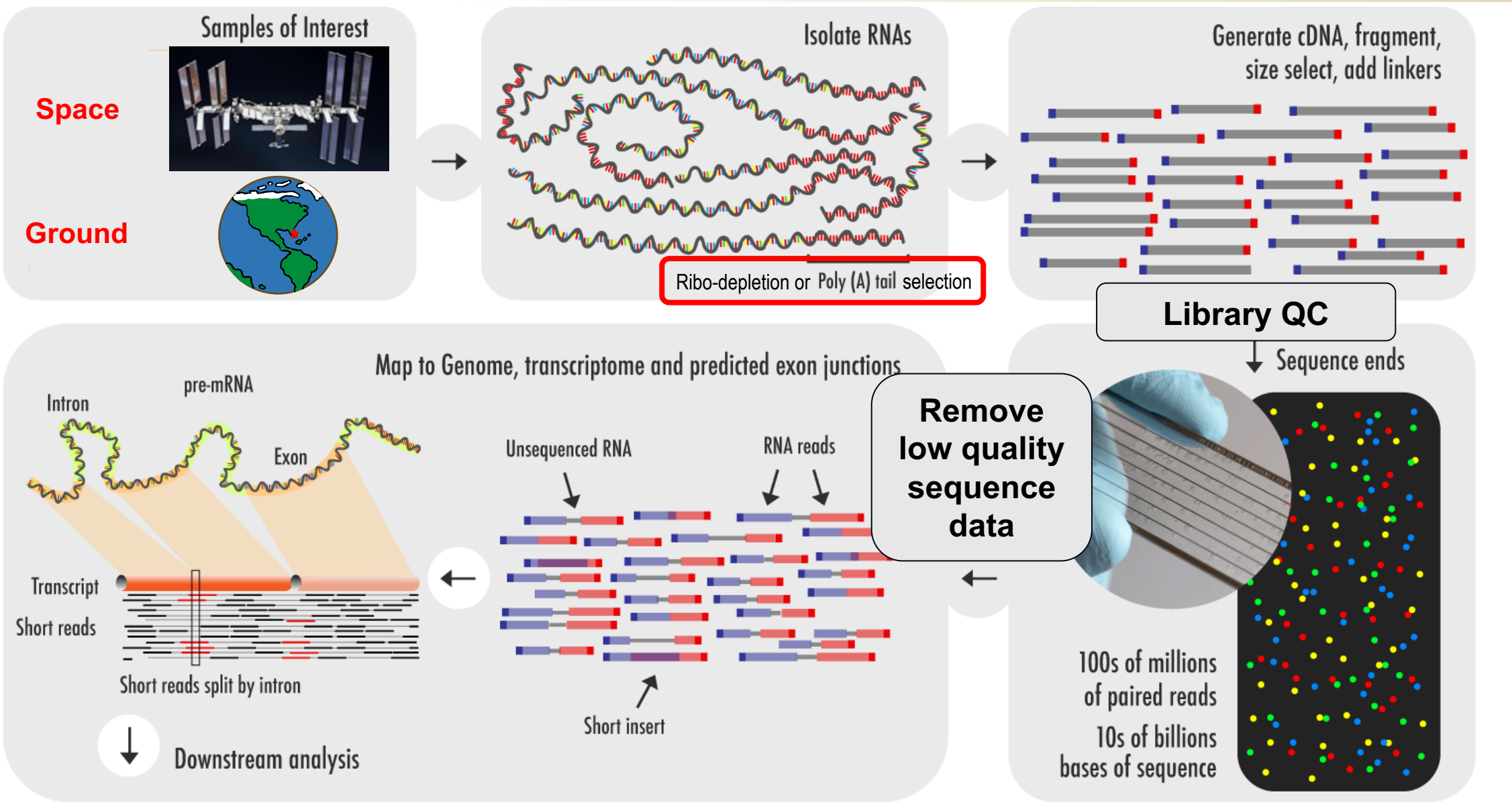
Modified from: https://microbenotes.com/dna-microarray/

Limitations of Microarrays

- Only annotated genes can be evaluated
- Requires hybridization and hence a risk of non-specific binding
  - Cross-hybridization of sequences with high identity
- Difficult to detect splice isoforms
- Dynamic range limited by the scanner (difficult to detect both low and high expressing genes simultaneously; saturation)
- Low expressed genes are highly variable
- Measures relative abundance – absolute quantitation is difficult
- Unequal labeling efficiency of fluorescent dyes
- Chip to chip variation
- Others?

RNA Sequencing (RNAseq)

RNA Sequencing Overview



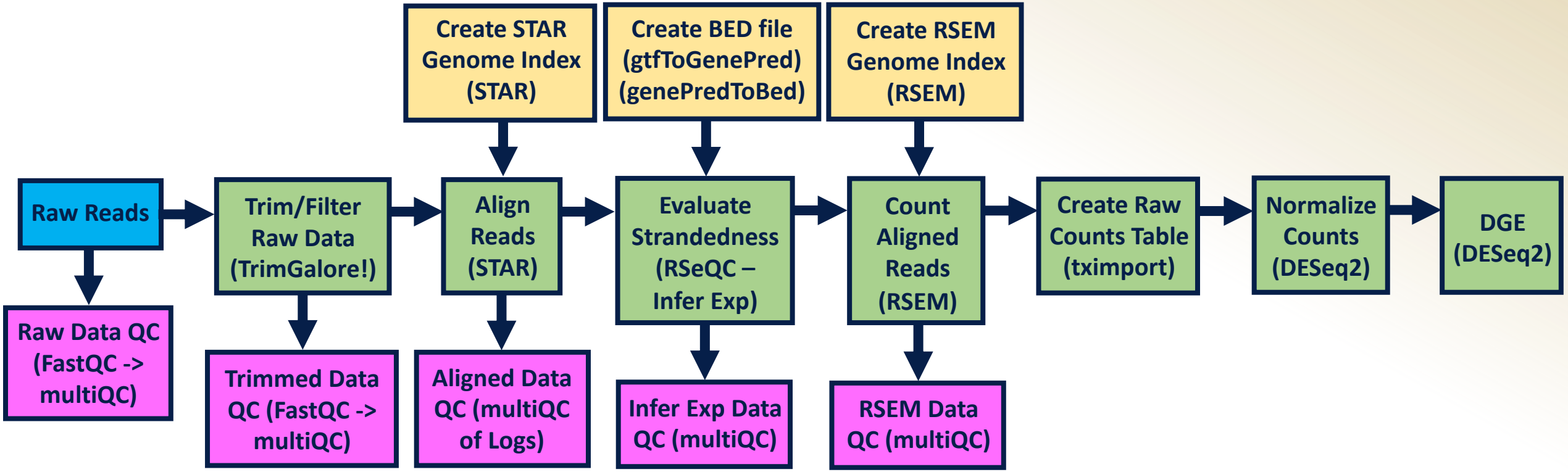
https://www.technologynetworks.com/genomics/articles/rna-seq-basics-applications-and-protocol-299461

RNA Sequencing Parameters

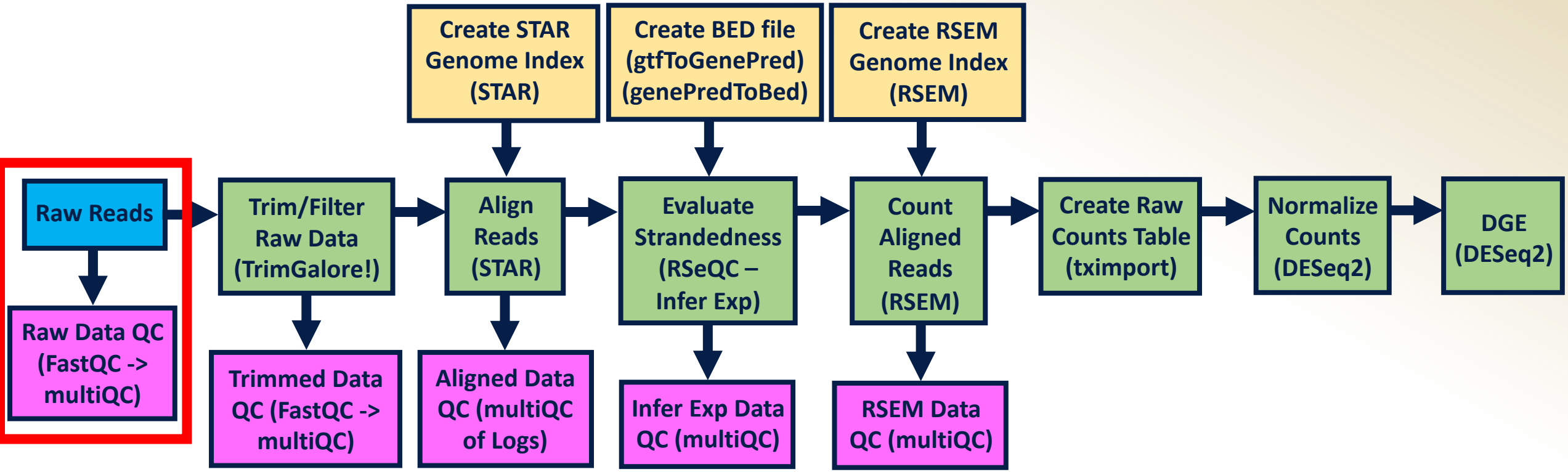
- **Read length**
  - Longer reads increase gene ID confidence
  - GL standard is 150bp
- **Sequencing depth**
  - Greater depth increases the likelihood of sequencing low-abundant transcripts, detecting novel transcripts, and quantifying isoforms
  - Greater depth is necessary for ribo-depleted samples (vs. poly-A enriched samples)
    - GL standard for mammals prepared with ripo-depletion is 40-60M reads/sample
  - More biological replicates is usually preferred over greater depth
- **Paired-end (PE) or Single-end (SE)**
  - PE is preferred

RNAseq
Data Processing

RNAseq Pipeline



RNAseq Pipeline: FastQC



RNAseq Raw Data

- Raw data generated from the sequencer are stored in fastq files or base calls, which are then converted to fastq files
- Fastq files contain multiple reads and each read is formatted as follows:

```
@K00188:264:HG3WJBBXX:1:1101:6289:1595 1:N:0:TAGCTT
GGACTGCCTTTCAGCCCGTCGCAGAGGGAATGGGAGCCTCTGGAGCGGGTGCAGAGGCTCAGCAG
+
AAFFFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJAJJJJJJ
```

Line 1 - Begins with @, followed by information about the sequencing run such as the sequencing platform, run number, flow cell ID and cluster location, read number (forward/reverse), and/or the sample index.

Line 2 - Contains the sequence, written as base calls (A, T, C, or G). Note that the sequence length is equal to the number of cycles in the sequencing run.

Line 3 - A separator line, which begins with a plus (+) sign.

Line 4 - Quality scores of each base call that are Phred +33 encoded and use ASCII characters to represent the quality of the bases.

RNAseq Raw Data Quality Scores

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

Phred quality scores are logarithmically linked to error probabilities

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1,000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%

Raw Data Quality Control with FastQC

- The FastQC program was created to assess the quality of sequence data and provide a QC report for quick evaluation

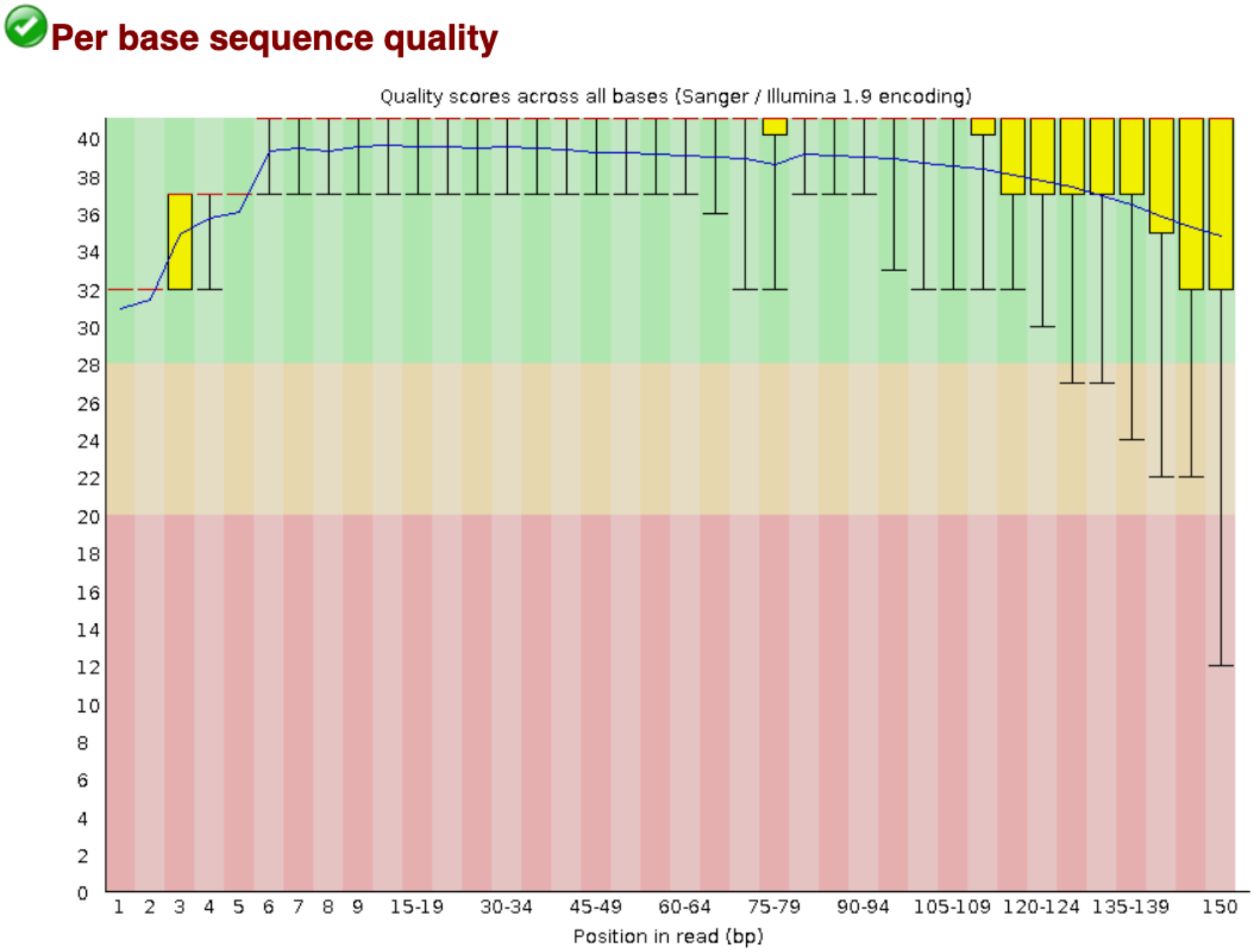
Basic Statistics

Measure	Value
Filename	Mmus_C57-6J_LVR_RR1_FLT_noERCC_Rep1_M25_R2_raw.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	100600211
Sequences flagged as poor quality	0
Sequence length	150
%GC	52

- Are these data for forward or reverse reads? **Reverse**
- What is the read depth for this sample? **100,600,211**
- What is the sample read length? **150**

https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Raw FastQC: Per base sequence quality



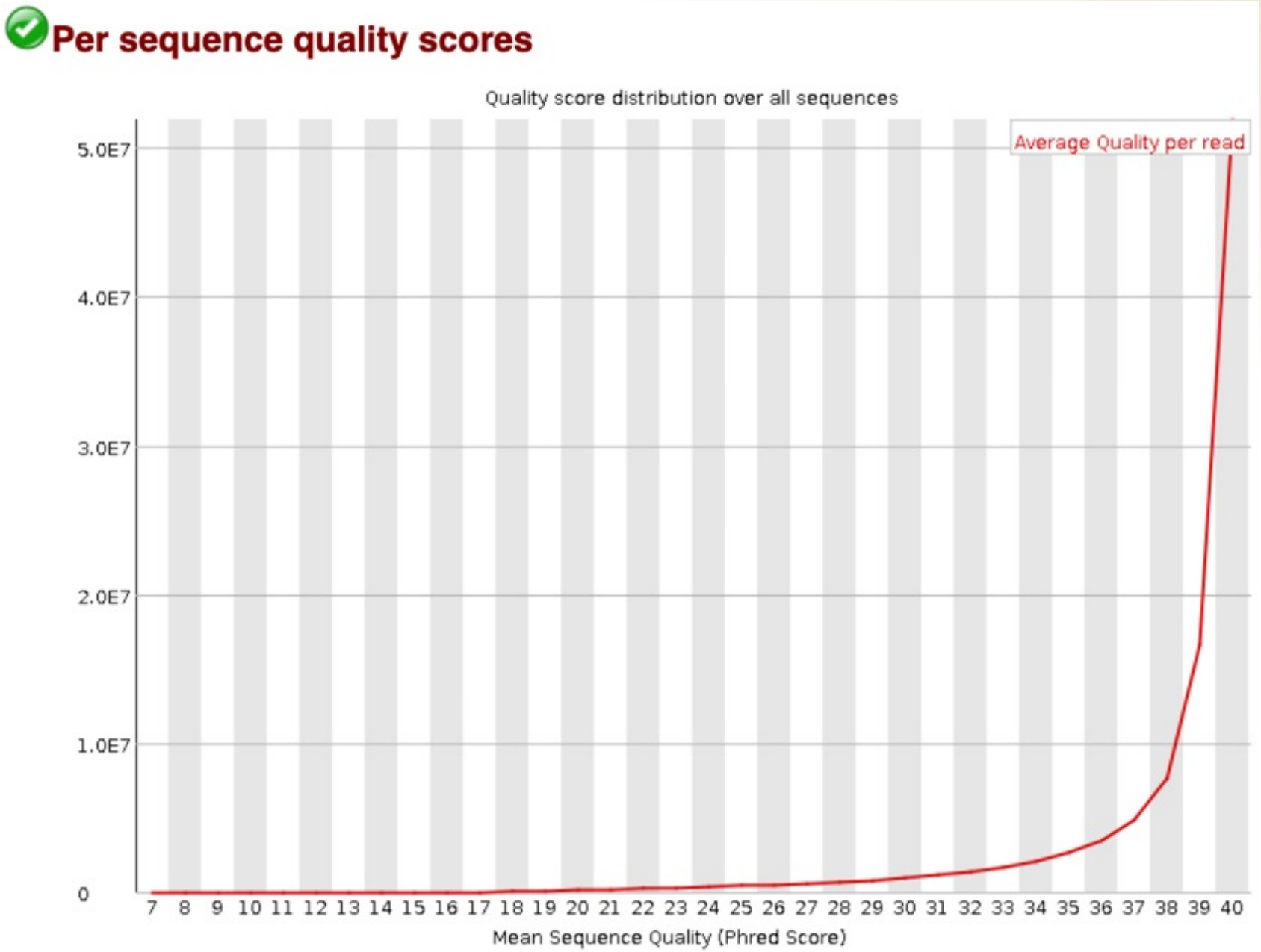
- Do you notice anything about the quality as you move towards the end of the read?

  Quality decreases

- These data are for the reverse reads, do you think the quality would be different for the forward reads?

  You may not think so, but the quality of the forward reads is often better than the reverse – We're not entirely sure why…

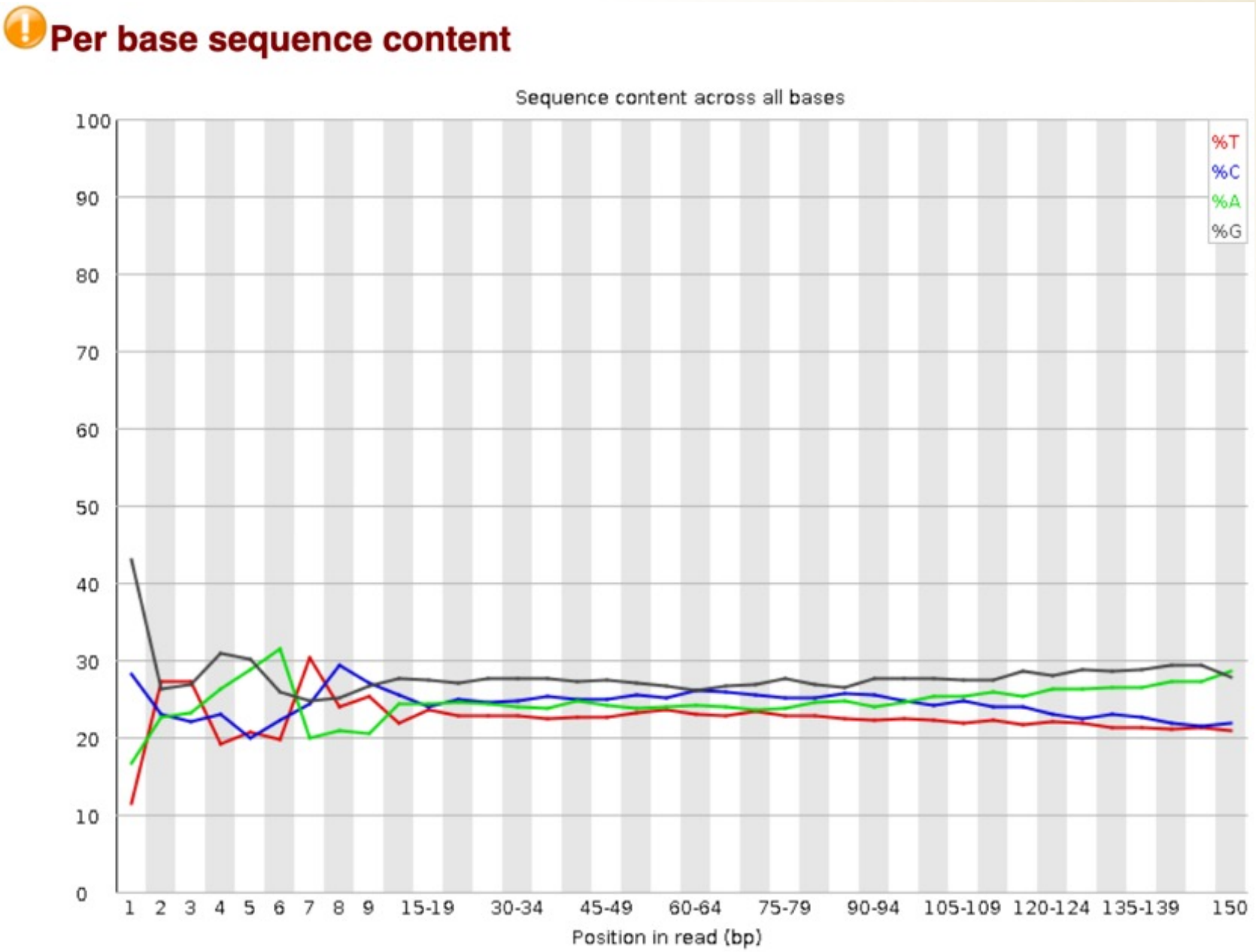
Raw FastQC: Per sequence quality scores



- Would you consider these data to have good quality overall?

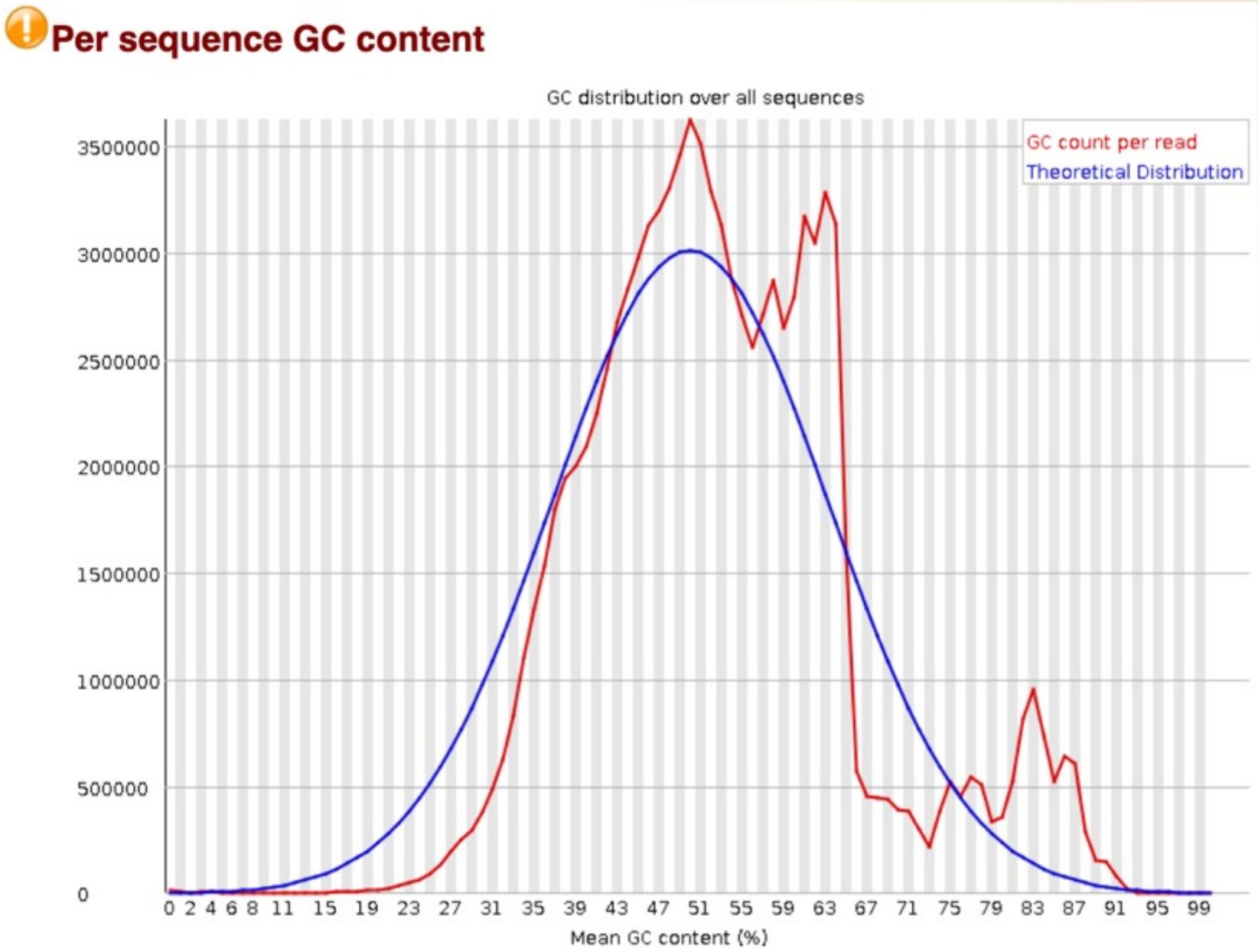
  Yes

Raw FastQC: Per base sequence content



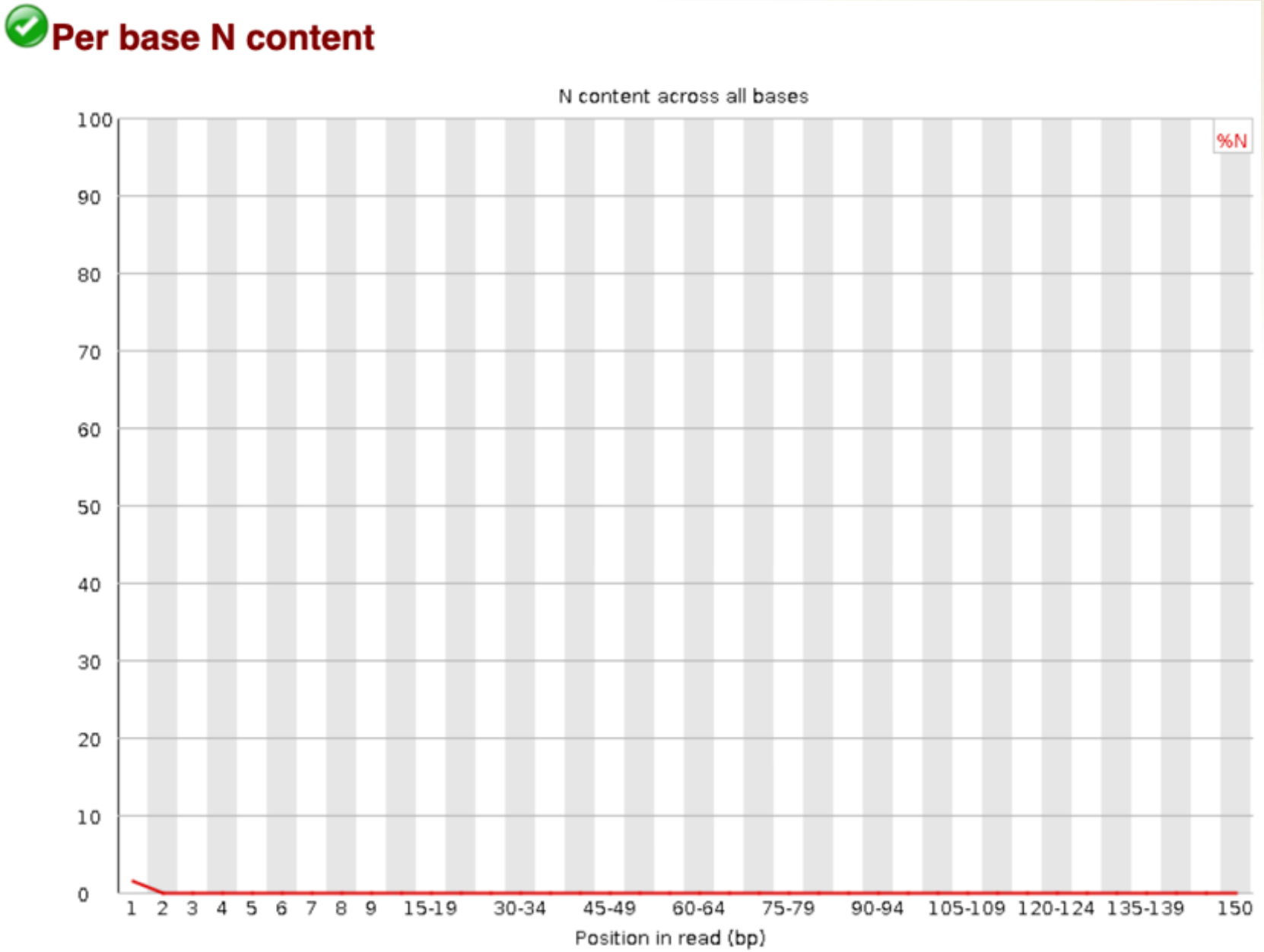
- These values should be fairly consistent across the read length in an unbiased library
- If the library is biased, you will see spikes in the data
- The nature of RNAseq primers tend to cause some wobble at the beginning of the reads

Raw FastQC: Per sequence GC content



- We expect the GC content to be normally distributed (blue line)
- Is the GC content normally distributed across all reads (red line)?
  Kind of?
- What do you think the peaks outside of the normal distribution indicate?
  Contamination

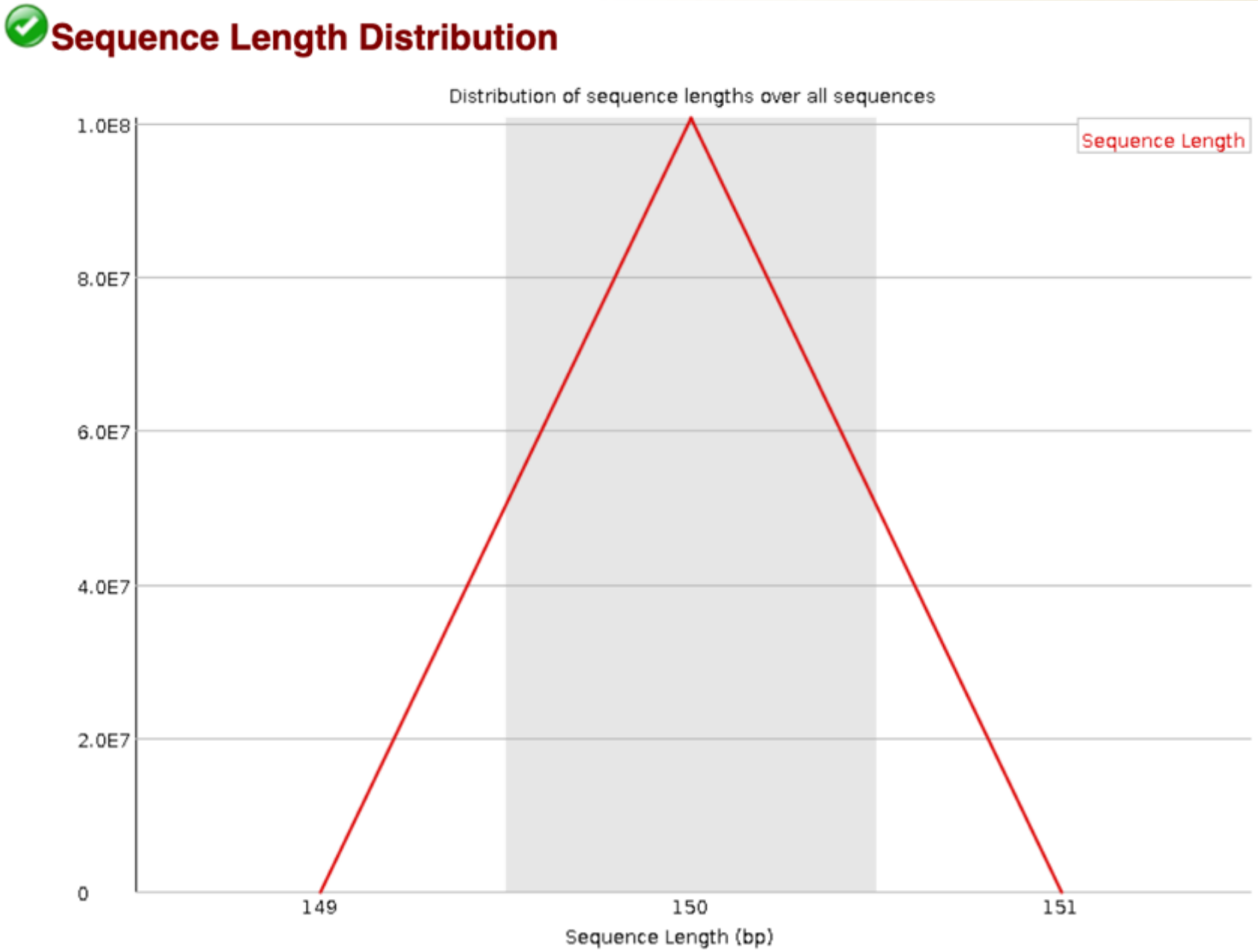
Raw FastQC: Per base N content



- If the sequencer cannot distinguish the base call, an `N` is recorded
- How often was a base unable to be called in these reads?

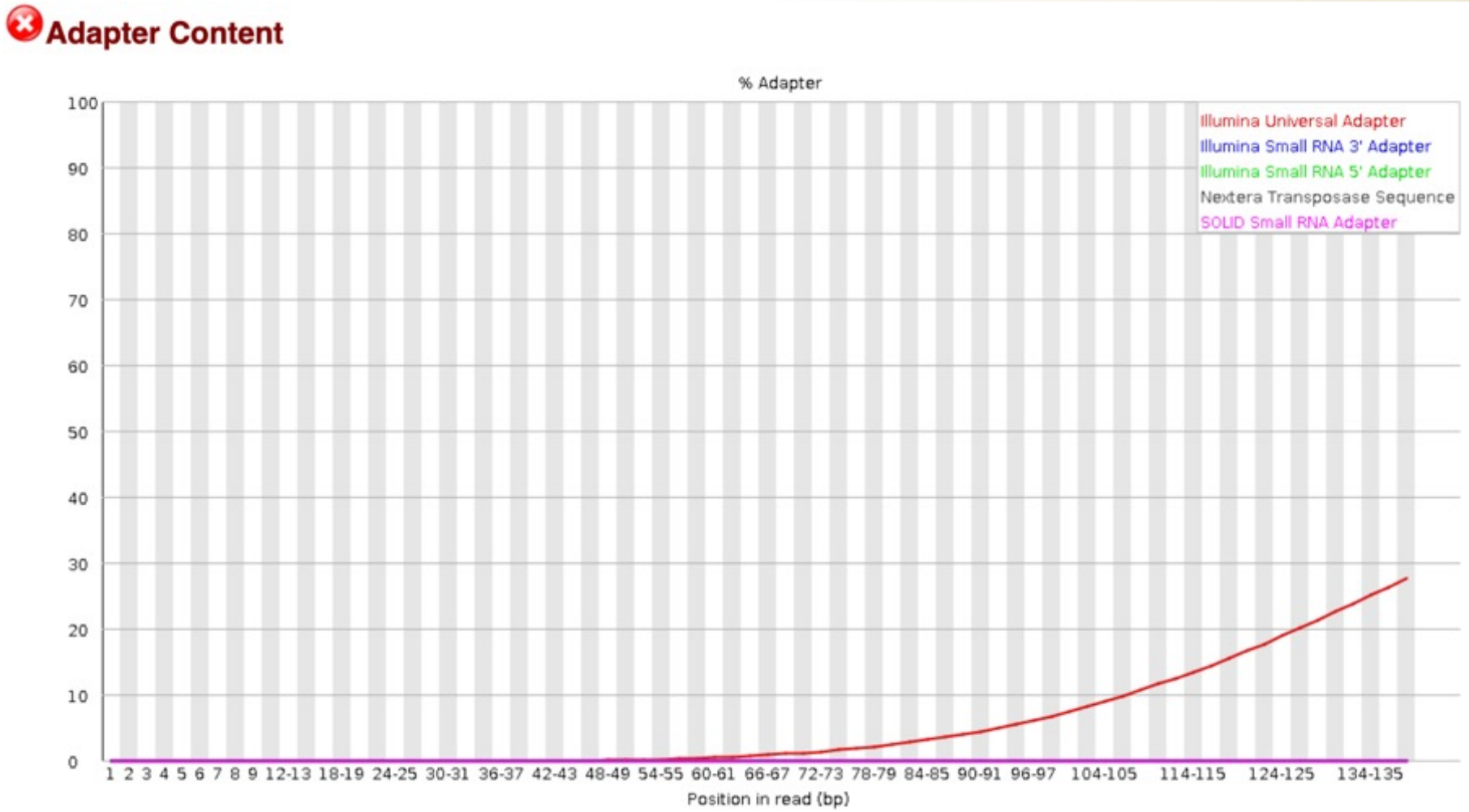
  Not very often at all, only rarely at the beginning of the read

Raw FastQC: Sequence Length Distribution



- The length of all raw reads should be the same
- Are all reads the same length?
  Yes
- What is the read length for these data?
  150bp

Raw FastQC: Adapter Content

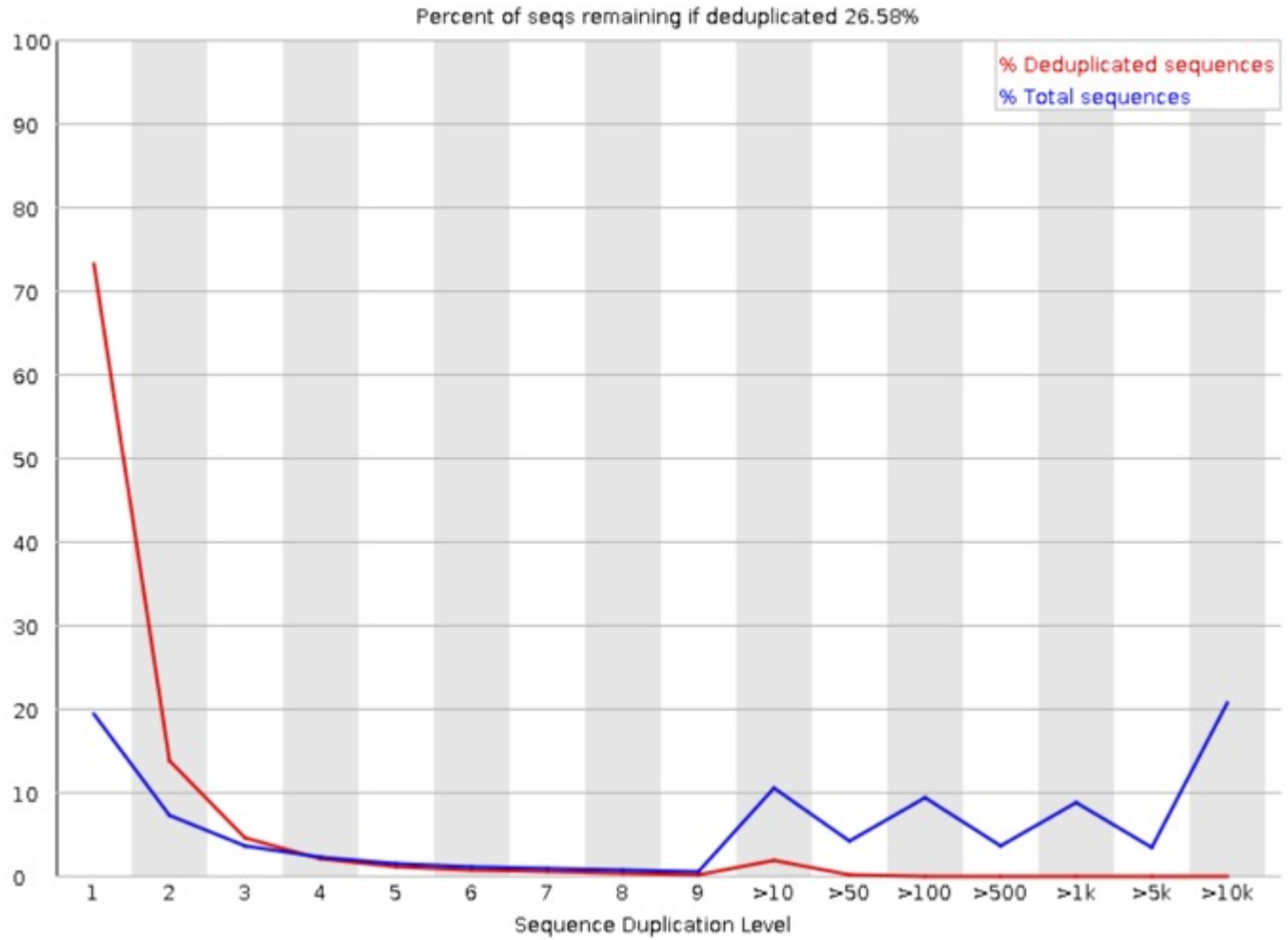


- If your sequence of interest is shorter than the number of sequencing cycles, you will sequence into your adapter.

- Does this example contain adapter sequences?
  Yes
- Where are the adapters relative to the read position?
  At the end of the read
- Do you think this will be a problem?
  Probably

Raw FastQC: Sequence Duplication Levels

Sequence Duplication Levels *Note: These data are based on a 100,000 read ***sampling*** of the total number of reads and ***only the first 50bp are considered***



- Used to determine how unique the reads are
- The blue line indicates the percent of duplicated reads at a specific duplication level (red shows what these data would be post deduplication)
- Read duplication can be either biological or technical in origin – How?
- If you have a very diverse library where all reads are unique, 100% of reads would have a duplication level of 1
- It's common, specifically when working with enriched libraries, to have some level of duplication – Why?
- Lower duplication levels (2-9 copies) are likely reads derived from 'interesting' genomic regions (i.e. mRNA)
- Moderate duplication levels (10-100 copies) are likely reads derived from rRNA and/or highly repetitive genomic regions
- High duplication levels (>100 copies) are indicative of a library issue (i.e. adapter dimers; too much PhiX) or contamination

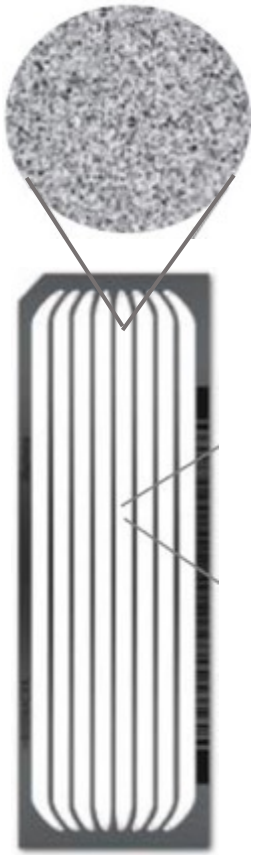
- Are duplicates present in this sample? Should we be concerned?

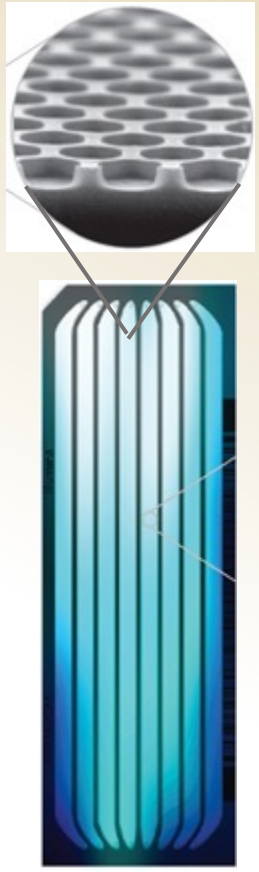
Yes and yes

- Roughly what percent of reads are duplicates? **~73.5%**

RNAseq Raw Data – Reasons for Duplications

A Review of Sequencing Duplicate Types





- A single cluster that has falsely been called as two by RTA
- Third party tools may report patterned flow cell clustering duplicates as optical duplicates

Not on Patterned Flow Cells

Optical

1 Cluster → Template generation → Called as 2

Clustering

- Duplicates in nearby wells on Patterned Flow Cells
- During cluster generation a library occupies two adjacent wells

Unique to Patterned Flow Cells

- Duplicate molecules that arise from over amplification during sample prep

PCR

Sister

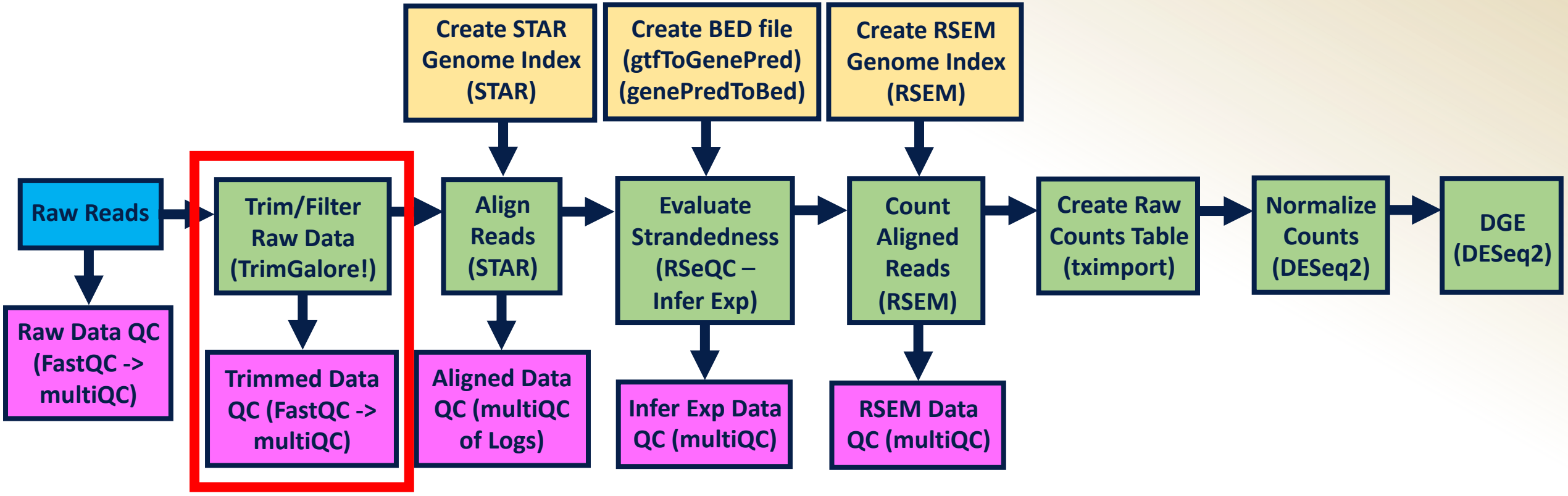
Complement strands of same library form independent clusters

- Treated as duplicates by some informatic pipelines

Present on all systems

illumina

RNAseq Pipeline: Trimming/Filtering



RNAseq Workflow: Trimming/Filtering

Why would we want to trim/filter (aka pre-process) raw sequence data?

- Remove low quality reads
- Trim adapters
- Trim 3’ or 5’ end(s)?
- Remove reads that become too short

Preprocessing Tools:

- Cutadapt
  https://cutadapt.readthedocs.io/en/stable/
- **TrimGalore!** (uses cutadapt)
  https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/
- Trimmomatic
  http://www.usadellab.org/cms/?page=trimmomatic
- HTStream
  https://github.com/s4hts/HTStream

TrimGalore! *Parameters:

- --phred33: instructs cutadapt to use ASCII+33 quality scores as Phred scores for quality trimming
- --quality <INT>: trim low-quality read ends (if not defined, a Phred score cutoff of 20 is applied)
- --length <INT>: remove reads that become shorter than length INT due to quality or adapter trimming (if not defined, a 20bp length threshold is applied)
- --paired: indicates paired-end reads - both reads, forward (R1) and reverse (R2), must pass length threshold or else both reads are removed

*Note: If not specified, adapters are automatically detected (by scanning the first million sequences) and removed

*These parameters are specific to the GeneLab standard pipeline, for a complete list of TrimGalore! parameters, go here: https://github.com/FelixKrueger/TrimGalore/blob/master/Docs/Trim_Galore_User_Guide.md#full-list-of-options-for-trim-galore

Trimmed/Filtered Sequence Quality: Basic Stats

Basic Statistics Raw Data

Measure	Value
Filename	Mmus_C57-6J_LVR_RR1_FLT_noERCC_Rep1_M25_R2_raw.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	100600211
Sequences flagged as poor quality	0
Sequence length	150
%GC	52

Basic Statistics Trimmed/Filtered Data

Measure	Value
Filename	Mmus_C57-6J_LVR_RR1_FLT_noERCC_Rep1_M25_R2_trimmed.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	100453545
Sequences flagged as poor quality	0
Sequence length	20-150
%GC	52

- How many reads were removed during pre-processing?

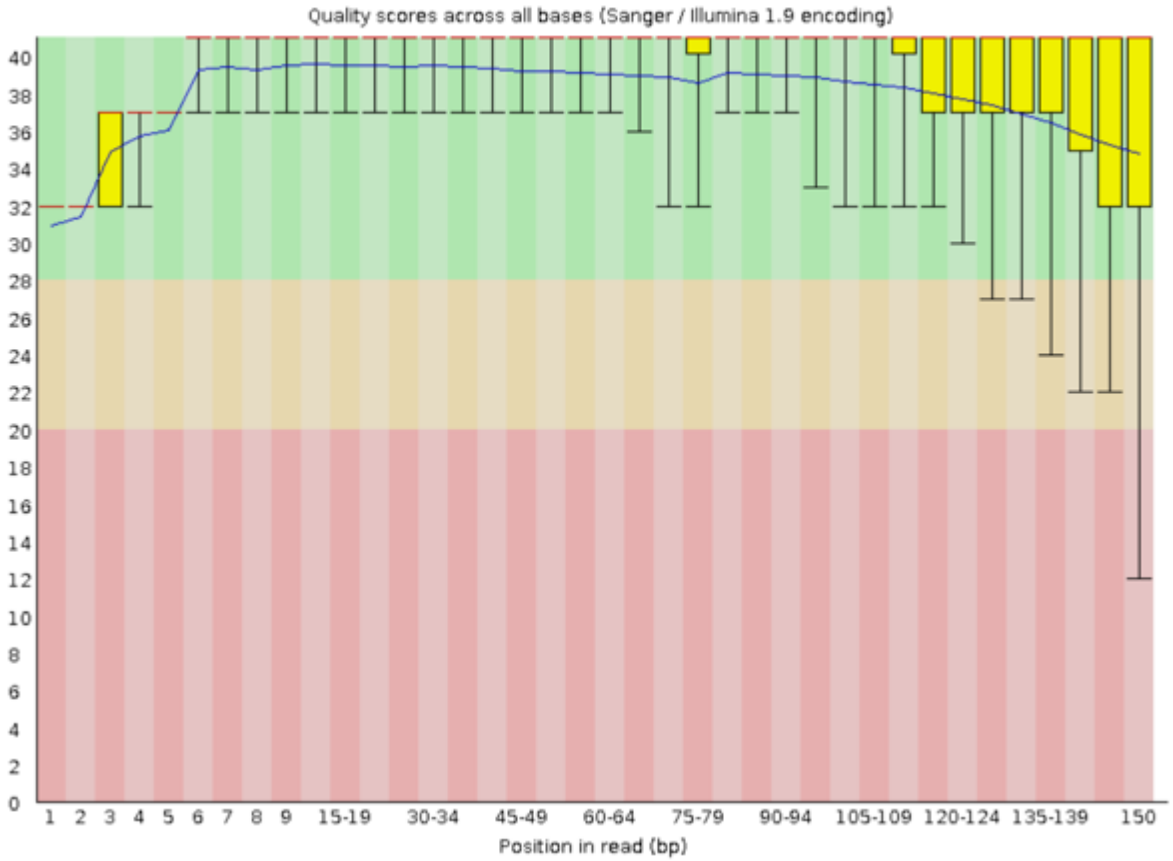
  100,600,211 – 100,453,545 = 146,666

- What is the sample read length after trimming?

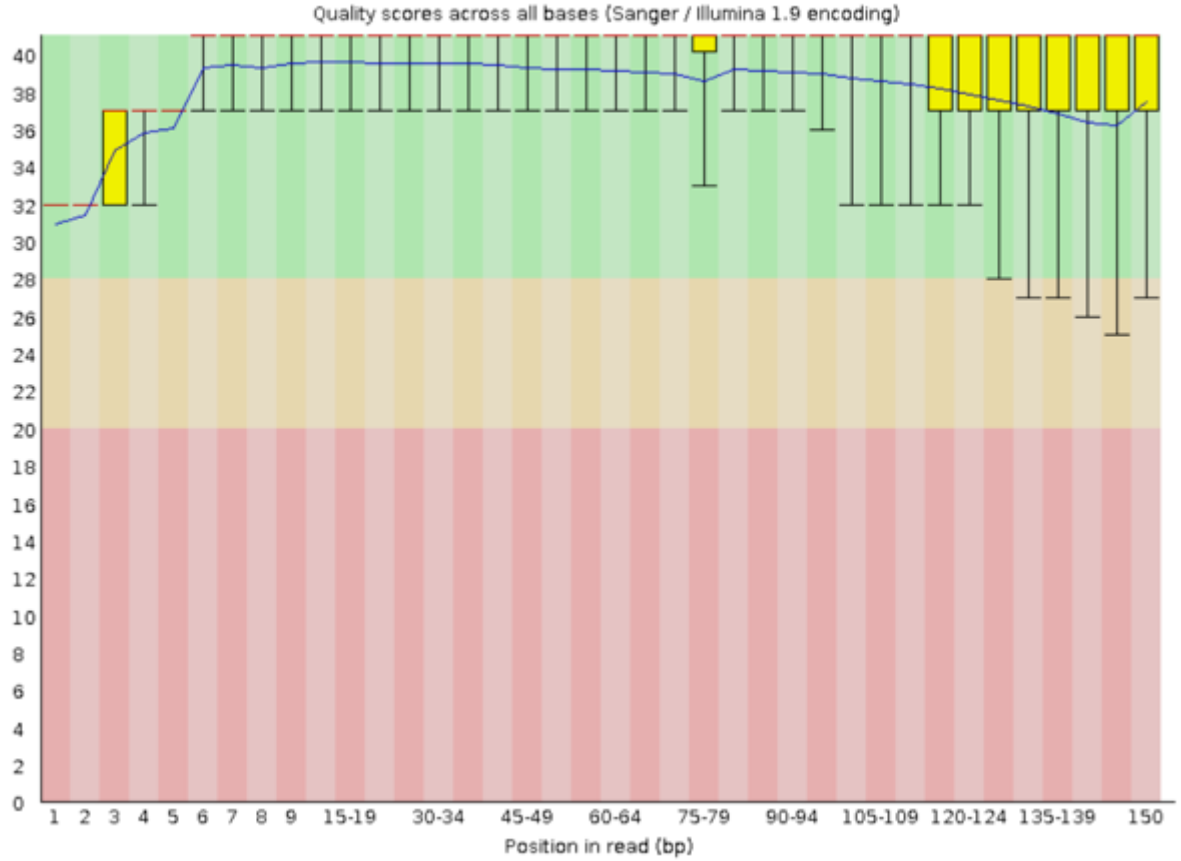
  Ranges from 20 – 150 bp

Trimmed/Filtered Sequence Quality: Per base sequence quality

Per base sequence quality **Raw Data**



Per base sequence quality **Trimmed/Filtered Data**

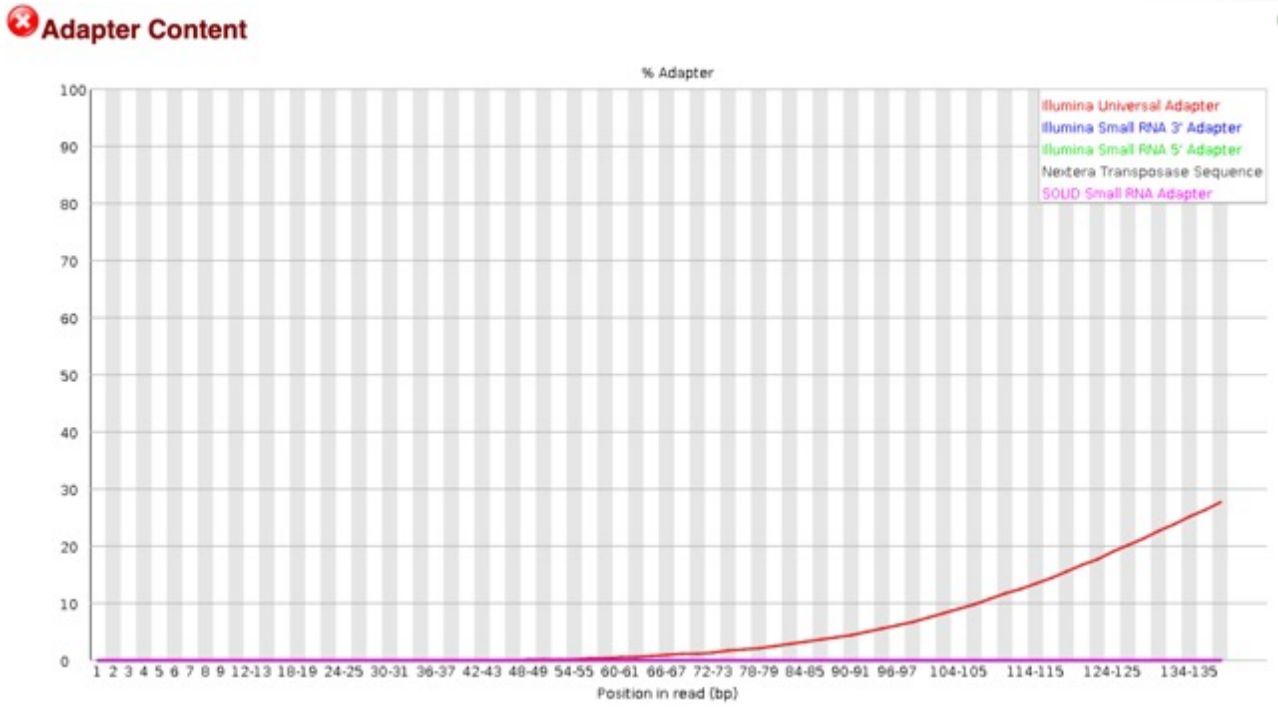


- How has the per-base sequence quality changed after trimming? **It improved!**
- What would happen to the quality if a Phred score cutoff >20 is applied?

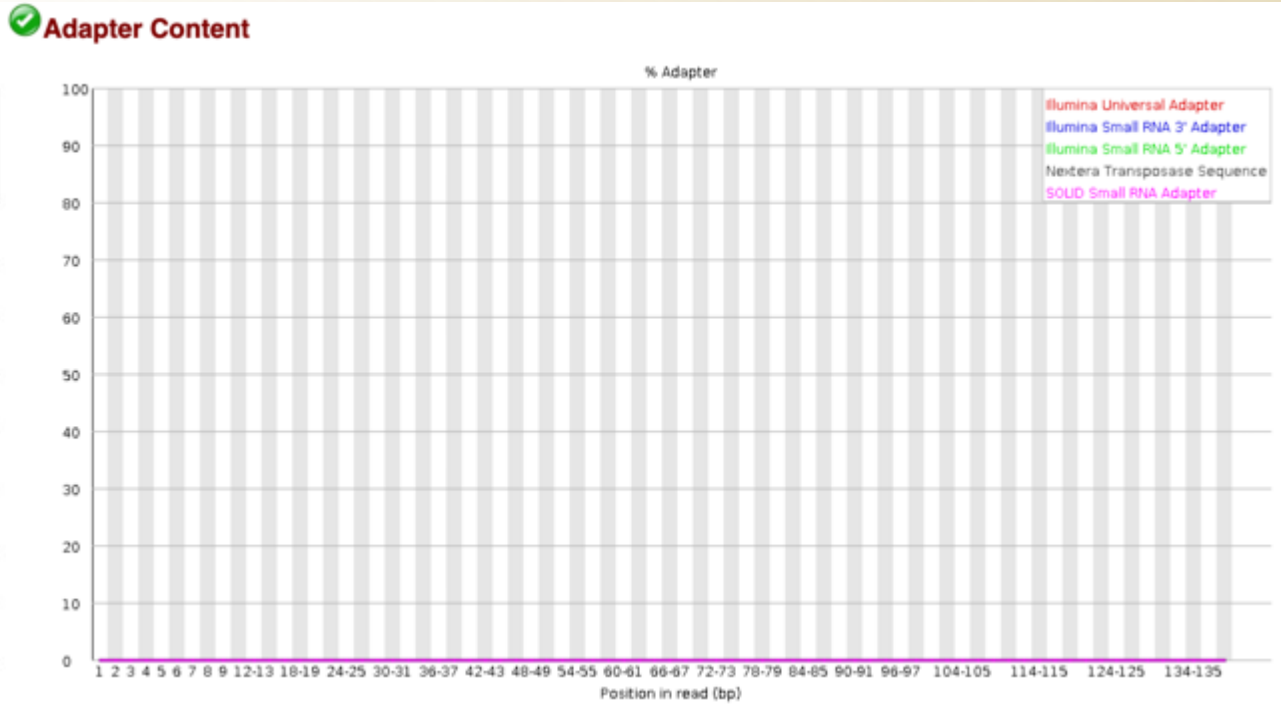
The post-trimmed reads would have better quality

Trimmed/Filtered Sequence Quality: Adapter Content

Raw Data



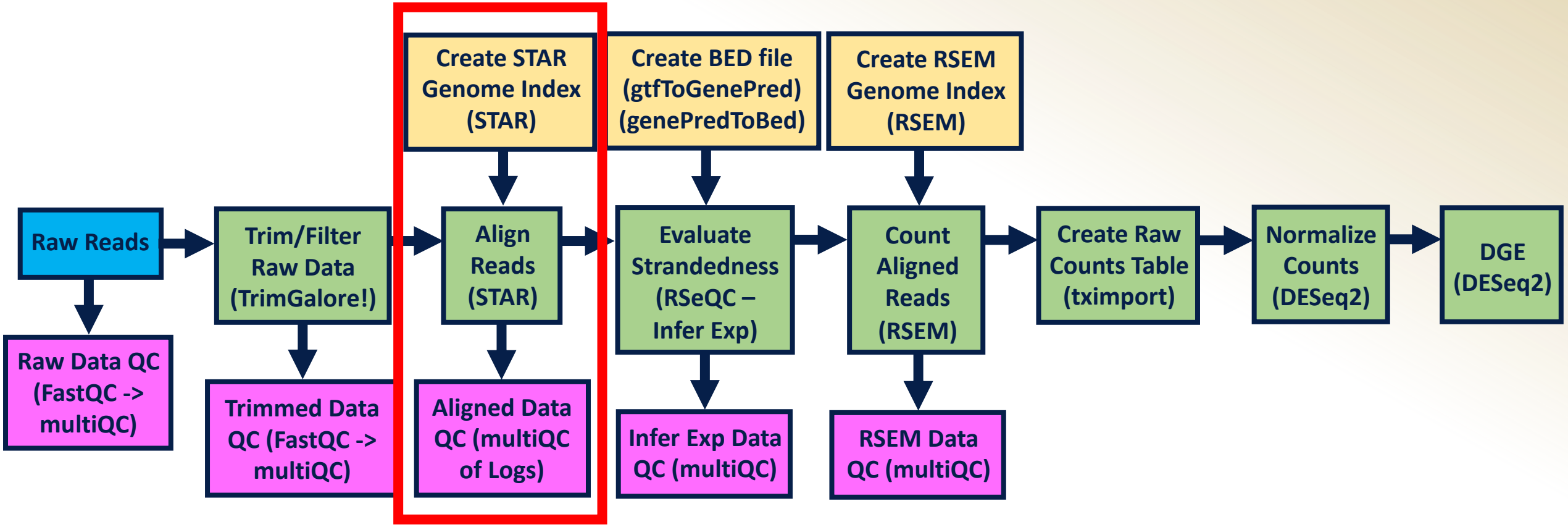
Trimmed/Filtered Data



- What happened to the adapter content after trimming?

Adapters went away!

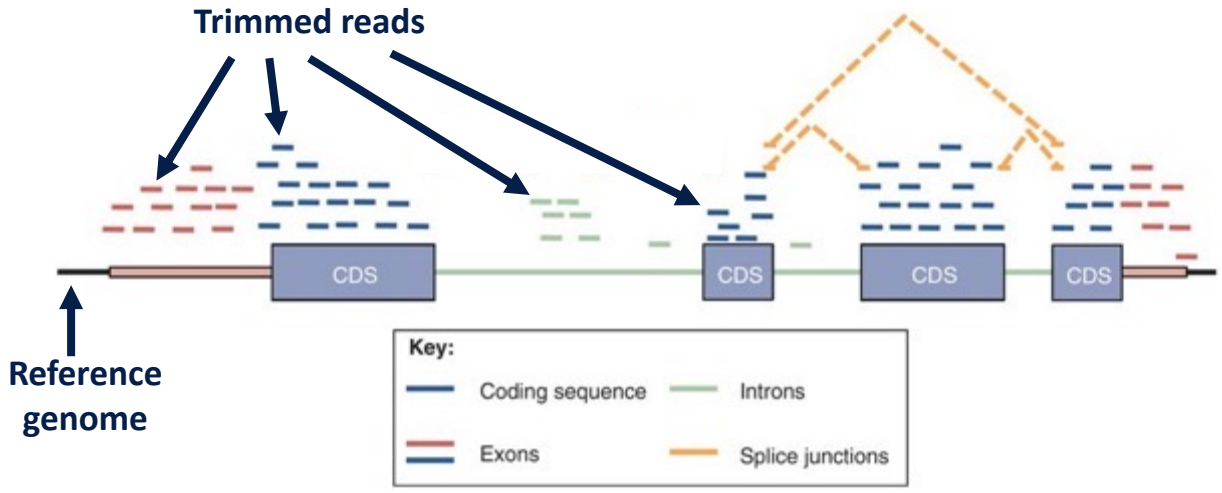
RNAseq Pipeline: Alignment



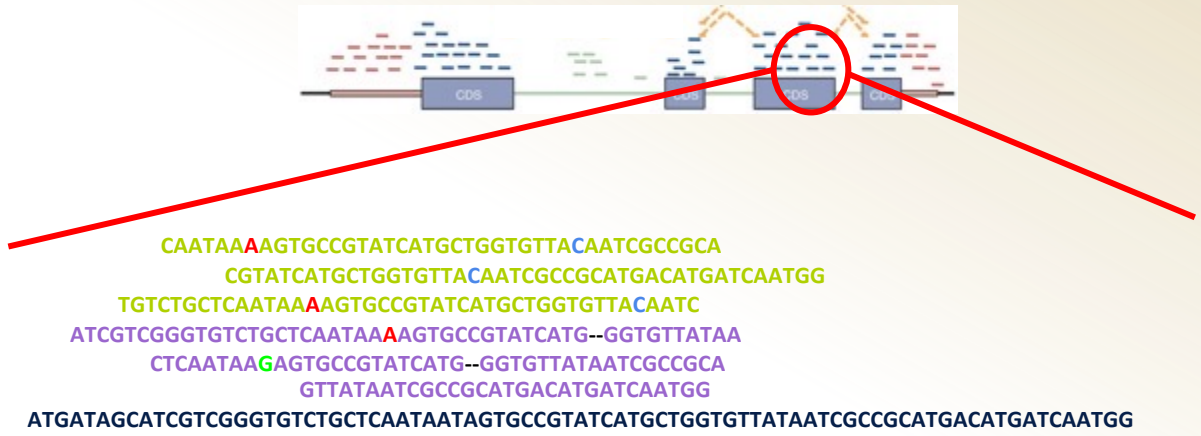
Alignment (aka Mapping?)

- We have millions of short sequences; now how do we figure out where they all came from?
- Luckily, the human genome as well as the genomes of several model organisms have been discovered and these reference genomes are available via public databases – this means we have a place to start looking, a map!

Mapping:



Alignment:



- Mapping reads to a reference genome will tell you where your reads came from (i.e. genomic coordinates)

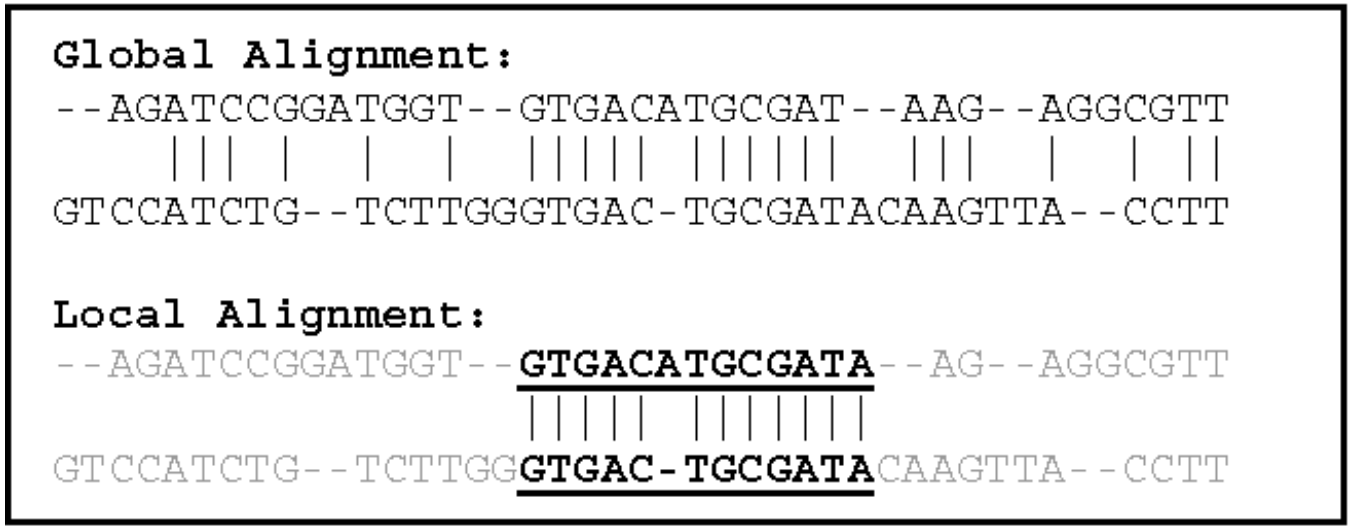
- We may need to fine-tune our mapping to account for differences between our samples and the reference genome
- Aligning reads to a reference genome will also identify single nucleotide differences, gaps, and insertions in addition to the genomic coordinates of origin

Oshlack et al. Genome Biology 2010, 11:220

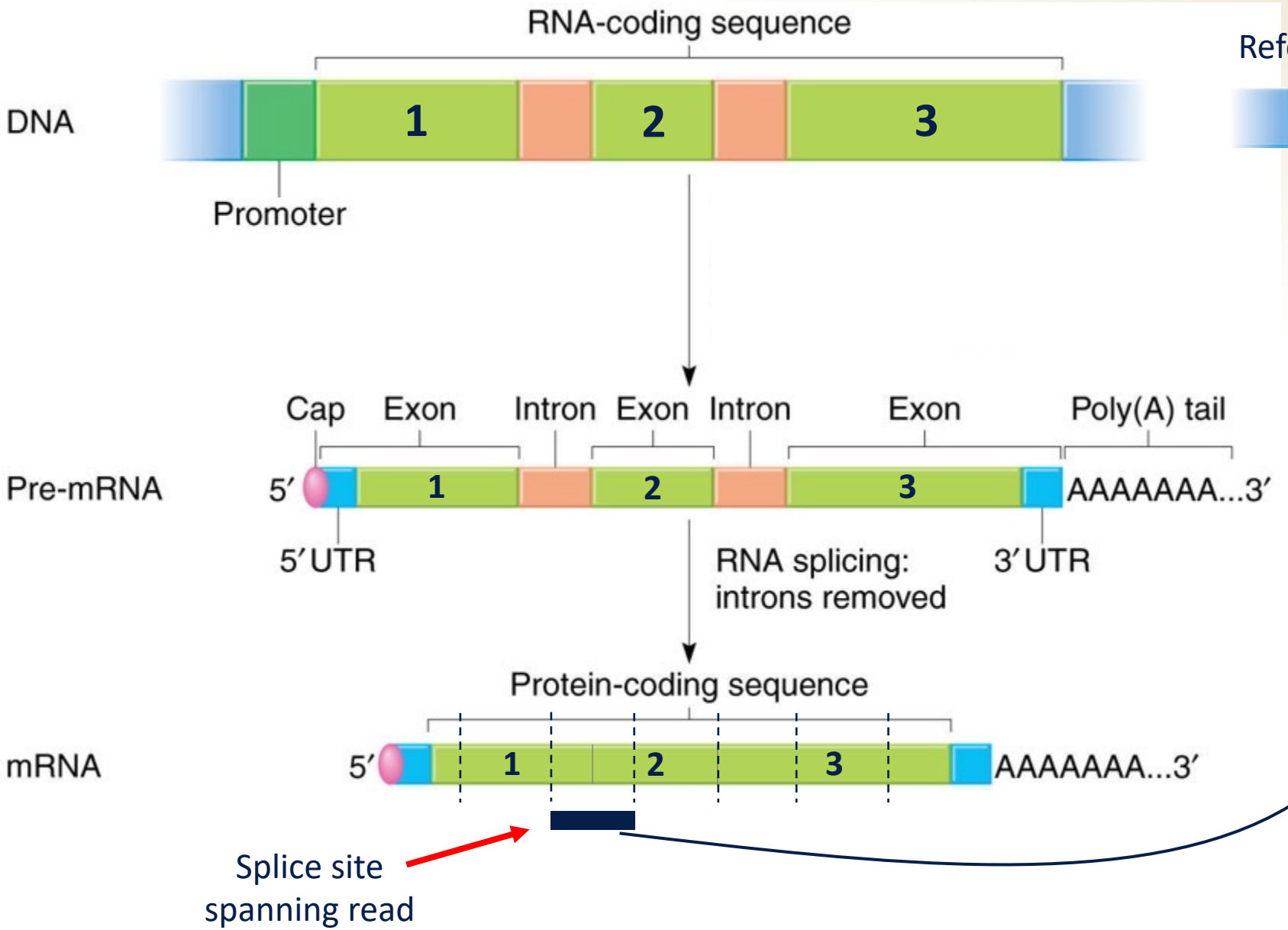
Aligners

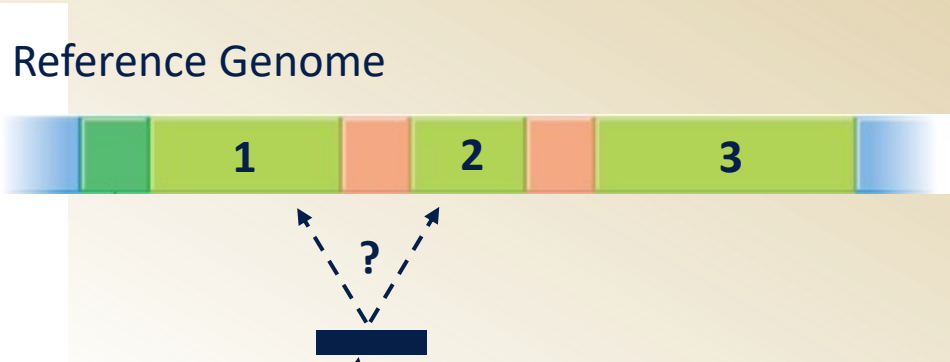
- **Global aligners** (Needleman-Wunsch algorithm) – attempt to align the whole provided sequence, end to end, of both the "query" and the "subject/target" (*examples*: aligning two *Pseudomonas* genomes; aligning the mouse and human transcriptomes)
- **Local aligners** (Smith-Waterman algorithm) – attempt to find "hits" or chains of hits within each provided sequence (*example*: identifying genes that share a domain with a target gene)
- **Glocal aligners** – Initial short read aligners assumed that the whole read came from one location within the reference (target) sequence; thus, **glo**bal with respect to the read and lo**cal** with respect to the reference
  - Are there any issues with this approach?

    What if a whole read comes from multiple locations within the reference (spans a splice junction)
- Most aligners commonly used today are local with respect to both the read and reference, which allows them to ignore poor alignment in low quality read ends and/or adapter sequences



Reads That Span Splice Sites





If we align reads back to a reference genome (DNA), how do we know where a mRNA-derived read came from when it spans a splice site?

Figure © 2010 PJ Russell, **iGenetics** 3rd ed.; all text material © 2014 by Steven M. Carr

Short Read Aligners – Splice Unaware vs Aware

- **Splice unaware aligners** (Needleman-Wunsch algorithm): Unable to properly align reads that span splice junctions and thus more commonly used for DNA-DNA alignment – Could these be used for aligning RNAseq data?
  - **Pseudo-aligners**: Compares read k-mers (overlapping subsequences) to a transcriptome de Bruijn graph (T-DBG) to find transcripts compatible with the read
- **Splice aware aligners** (Smith-Waterman algorithm): Equipped to handle intron-sized gaps, improving alignment of reads that span splice junctions when aligning to a reference genome and thus are commonly used for transcript-derived cDNA-DNA alignment

Splice unaware aligners:

- Burrows-Wheeler Aligner (BWA)
  - http://bio-bwa.sourceforge.net/
- Bowtie (similar to BWA)
  - http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

Pseudo-aligners:

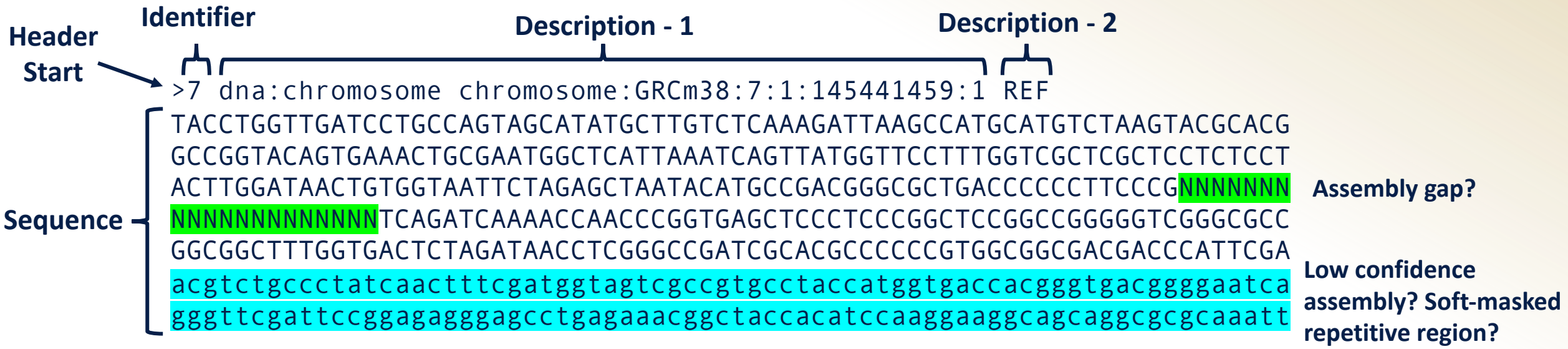
- Kallisto
  - https://pachterlab.github.io/kallisto/
- Salmon
  - https://salmon.readthedocs.io/en/latest/salmon.html

Splice aware aligners:

- **Spliced Transcripts Aligned to a Reference (STAR)**
  - https://github.com/alexdobin/STAR
- Hierarchical indexing for spliced alignment of transcripts 2 (HISAT2)
  - http://daehwankimlab.github.io/hisat2/

Reference Genome (Fasta)

- Short read aligners require a reference – when aligning RNAseq data we use the reference genome of the organism in which the samples were derived
- Reference genomes are stored in fasta files, which have the general format shown below:



- Header Start: All fasta header lines begin with a `>`
- Identifier: Sequence ID – This could be a database-specific ID, a chromosome number (as shown), a gene symbol, etc.
- Description fields: Additional information about the sequence, some databases maintain a standard format for these fields
- Note: Header fields are separated by spaces and some databases also include `|` or `,`

Gene Annotations (GFF/GTF)

- If we want to identify annotated genes (i.e. genes with known genomic coordinates and functions), we need to provide the aligner with a gene annotation file corresponding to reference genome used
  - Gene annotations are stored in Gene Feature Format (GFF) or Gene Transfer Format (GTF) files, which have the following format:

	1	2	3	4	5	6	7	8	9
	3	ensembl_havana	gene	108107280	108146146	.	-	.	gene_id "ENSMUSG00000000001"; gene_version "4"; gene_name "Gnai3"; gene_source "ensembl_havana";
	3	ensembl_havana	transcript	108107280	108146146	.	-	.	gene_id "ENSMUSG00000000001"; gene_version "4"; transcript_id "ENSMUST00000000001";
	3	ensembl_havana	exon	108145888	108146146	.	-	.	gene_id "ENSMUSG00000000001"; gene_version "4"; transcript_id "ENSMUST00000000001";
*	3	ensembl_havana	CDS	108145888	108146005	.	-	0	gene_id "ENSMUSG00000000001"; gene_version "4"; transcript_id "ENSMUST00000000001";
	3	ensembl_havana	start_codon	108146003	108146005	.	-	0	gene_id "ENSMUSG00000000001"; gene_version "4"; transcript_id "ENSMUST00000000001";

1. **Sequence Name**: Name of the chromosome or scaffold [3]
2. **Source**: Program that generated the GTF file or feature [ensembl_havana]
3. **Feature**: Feature type; gene, exon, CDS, start codon, etc. [CDS]
4. **Start**: Start location on reference sequence [108145888]
5. **End**: End location on reference sequence [108146005]

[*]

6. **Score**: Floating point value [.]
7. **Strand**: Forward (+) or reverse (-) [-]
8. **Frame**: Indicates which base, 0, 1, or 2 is the first base of a codon [0]
9. **Attribute**: `;`-deiminated list of tags with additional info [gene_id "ENSMUSG00000000001"; gene_version "4"; transcript_id "ENSMUST00000000001";]

Where do we get reference genome and annotation files?

- Reference genomes and respective annotation files can be downloaded from publicly available databases:
  - **Ensembl**: https://www.ensembl.org/
  - **Ensembl Genomes**: https://ensemblgenomes.org/
  - GENCODE (uses Ensembl IDs): https://www.gencodegenes.org/
  - Illumina igenomes: https://support.illumina.com/sequencing/sequencing_software/igenome.html
  - NCBI genome: https://www.ncbi.nlm.nih.gov/genome/
  - Specialized databases:
    - https://flybase.org/
    - https://wormbase.org/
    - http://www.xenbase.org/
    - https://vectorbase.org/
    - https://phytozome.jgi.doe.gov/
    - https://www.patricbrc.org/

Building a STAR Index

- Before we can align trimmed reads to a reference genome (or transcriptome), the genome must be indexed
- By creating a genomic (or transcriptomic) index, aligners organize and store the genomic context to make searching the entire genome more efficient

STAR *Parameters:

- --runMode genomeGenerate: Instructs STAR to run genome index generation job to create the STAR indexed reference.
- --genomeSAindexNbases <INT>: Length (in bases) of the SA pre-indexing string, usually between 10 and 15. Longer strings require more memory but allow for faster searches. This value should be scaled down for smaller genomes (like bacteria) to min(14, log2(GenomeLength)/2 - 1). For example, for a 1 megaBase genome this value would be 9.
- --genomeDir: Specifies the path to the directory where the STAR indexed reference will be stored. At least 100GB of available disk space is required for mammalian genomes.
- --genomeFastaFiles: Specifies one or more uncompressed fasta file(s) containg the genome reference sequences.
- --sjdbGTFfile: Specifies the uncompressed file(s) containing annotated transcripts in the standard gtf format.
- --sjdbOverhang <INT>: Indicates the length of the genomic sequence around the annotated junction to be used in constructing the splice junctions database. The length should be one less than the length of the reads.

* These parameters are specific to the GeneLab standard pipeline, for a complete list of STAR parameters, go here: https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf

Alignment with STAR

STAR *Parameters:

- --twopassMode: Specifies 2-pass mapping mode; the `Basic` option instructs STAR to perform the 1st pass mapping, then automatically extract junctions, insert them into the genome index, and re-map all reads in the 2nd mapping pass.
- --genomeDir: Specifies the path to the directory where the STAR indexed reference is stored.
- --outSAMunmapped: Specifies output of unmapped reads in the SAM format; the `Within` option instructs STAR to output the unmapped reads within the main SAM file.
- --outFilterType: Specifies the type of filtering; the `BySJout` option instructs STAR to keep only those reads that contain junctions that passed filtering in the SJ.out.tab output file.
- --outSAMattributes: List of desired SAM attributes in the order desired for the output SAM file; SAM attribute descriptions can be found here: https://samtools.github.io/hts-specs/SAMtags.pdf
- --outFilterMultimapNmax <INT>: Specifies the maximum number of loci the read is allowed to map to; all alignments will be output only if the read maps to no more loci than this value. <20>
- --outFilterMismatchNmax <INT>: Maximum number of mismatches allowed to be included in the alignment output. <10>
- --outFilterMismatchNoverReadLmax <FLOAT>: Ratio of mismatches to read length allowed to be included in the alignment output; the <0.04> value indicates that up to 4 mismatches are allowed per 100 bases.
- --alignIntronMin <INT>: Minimum intron size; a genomic gap is considered an intron if its length is equal to or greater than this value, otherwise it is considered a deletion. <20>

* These parameters are specific to the GeneLab standard pipeline, for a complete list of STAR parameters, go here: https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf

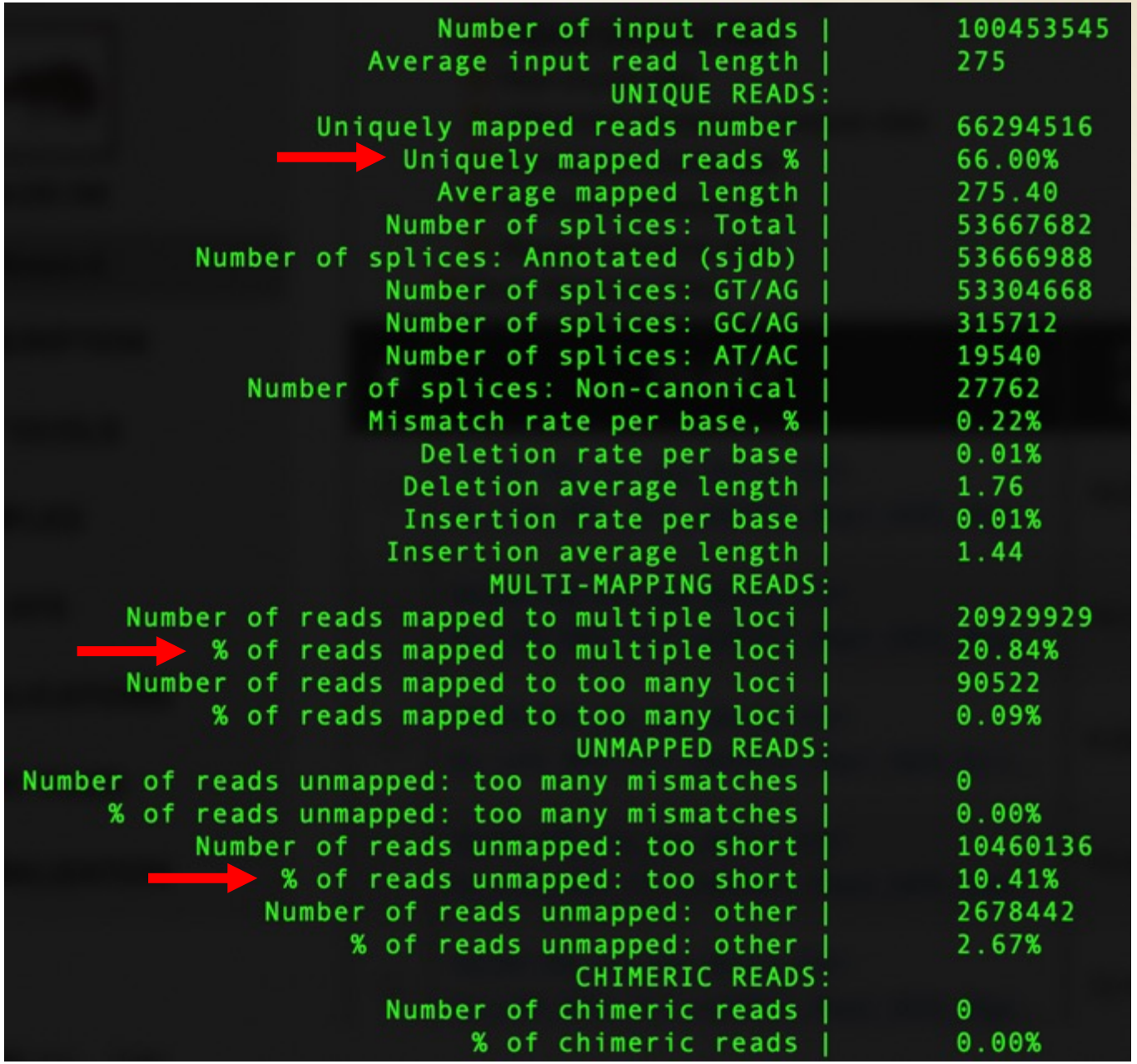
Alignment with STAR

STAR *Parameters:

- --alignIntronMax <INT>: Maximum intron size. <1000000>
- --alignMatesGapMax <INT>: Maximum genomic distance (in bases) between two mates of paired-end reads; this option should be removed for single-end read. <1000000>
- --alignSJoverhangMin <INT>: Minimum overhang (i.e. block size) for unannotated spliced alignments. <8>
- --alignSJDBoverhangMin <INT>: Minimum overhang (i.e. block size) for annotated spliced alignments. <1>
- --sjdbScore <INT>: Additional alignment score for alignments that cross database junctions. <1>
- --outSAMtype: Specifies desired output format; the `BAM SortedByCoordinate` options specify that the output file will be sorted by coordinate and be in the BAM format
- --quantMode: Specifies the type(s) of quantification desired; the `TranscriptomeSAM` option instructs STAR to output a separate sam/bam file containing alignments to the transcriptome.
- --outSAMheaderHD: Indicates a header line for the SAM/BAM file.
- --outFileNamePrefix: Specifies the path to and prefix for the output file names; for GeneLab the prefix is the sample id.
- --readFilesIn: Path to input read 1 (forward read) and read 2 (reverse read); for paired-end reads, read 1 and read 2 should be separated by a space; for single-end reads only read 1 should be indicated.

* These parameters are specific to the GeneLab standard pipeline, for a complete list of STAR parameters, go here: https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf

STAR Alignment Log



```
                         Number of input reads |       100453545
                     Average input read length |       275
                                   UNIQUE READS:
                    Uniquely mapped reads number |       66294516
                         Uniquely mapped reads % |       66.00%
                           Average mapped length |       275.40
                        Number of splices: Total |       53667682
             Number of splices: Annotated (sjdb) |       53666988
                        Number of splices: GT/AG |       53304668
                        Number of splices: GC/AG |       315712
                        Number of splices: AT/AC |       19540
                    Number of splices: Non-canonical |       27762
                    Mismatch rate per base, % |       0.22%
                       Deletion rate per base |       0.01%
                      Deletion average length |       1.76
                      Insertion rate per base |       0.01%
                     Insertion average length |       1.44
                           MULTI-MAPPING READS:
        Number of reads mapped to multiple loci |       20929929
             % of reads mapped to multiple loci |       20.84%
        Number of reads mapped to too many loci |       90522
             % of reads mapped to too many loci |       0.09%
                                UNMAPPED READS:
Number of reads unmapped: too many mismatches |       0
     % of reads unmapped: too many mismatches |       0.00%
          Number of reads unmapped: too short |       10460136
               % of reads unmapped: too short |       10.41%
              Number of reads unmapped: other |       2678442
                   % of reads unmapped: other |       2.67%
                                CHIMERIC READS:
                     Number of chimeric reads |       0
                          % of chimeric reads |       0.00%
```

Alignment File (SAM/BAM)

- Alignment data are stored in Sequence Alignment Map (*SAM) files, which have the following format:

1		2	3	4	5	6								
J00113:339:HMJMNBBXX:3:1101:1144:1367		163	ENSMUST00000031314	1845	255	123M	=	1845	123	NGCAA…GGAAC	#AAFF…JFJJJ	NH:i:	HI:i:1	MC:Z:123M
* J00113:339:HMJMNBBXX:3:1101:1144:1367		83	ENSMUST00000031314	1845	255	123M	=	1845	-123	TGCAA…GGANC	JFJFJ…FFA#A	NH:i:	HI:i:1	MC:Z:123M
J00113:339:HMJMNBBXX:3:1101:5426:1367		419	ENSMUST00000174924	135	3	148M	=	144	159	NGTGA…AACAT	#AAFF…JJJJJ	NH:i:2	HI:i:1	MC:Z:150M
J00113:339:HMJMNBBXX:3:1101:5426:1367		339	ENSMUST00000174924	144	3	150M	=	135	-159	CCGGG…ACCNC	JJJJJ…FFA#A	NH:i:2	HI:i:1	MC:Z:148M
J00113:339:HMJMNBBXX:3:1101:5426:1367		163	ENSMUST00000175032	135	3	148M	=	144	159	NGTGA…AACAT	#AAFF…JJJJJ	NH:i:2	HI:i:2	MC:Z:150M
J00113:339:HMJMNBBXX:3:1101:5426:1367		83	ENSMUST00000175032	144	3	150M	=	135	-159	CCGGG…ACCNC	JJJJJ…FFA#A	NH:i:2	HI:i:2	MC:Z:148M

Column	Name	Description [*]
1	QNAME	Query template name: Info about the sequencing run that generated the read, found in line 1 of the trimmed fastq file [J00113:339:HMJMNBBXX:3:1101:1144:1367]
2	FLAG	bitwise FLAG: Information about the alignment encoded in bits. To easily decode the SAM FLAG, type it into the Broad Institute's SAM FLAG decoder: https://broadinstitute.github.io/picard/explain-flags.html [83]
3	RNAME	Reference sequence name: Name of the reference sequence the read aligned to (this will be the ensembl transcript ID in the transcript-aligned BAM file as shown above, and the chromosome number in the genome-aligned BAM file) [ENSMUST00000031314]

Column	Name	Description [*]
4	POS	1-based leftmost mapping position: The position on the reference genome in which the left most base of the read aligns. [1845]
5	MAPQ	Mapping quality: Equal to the -10log(base10) of the probability that the mapping position is wrong; a value of [255] indicates the mapping quality is not available (it's uniquely mapped).
6	CIGAR	CIGAR string: Aligned read length and associated operation, which encodes information about the alignment relative to the reference (i.e. match/mismatch, insertion/deletion). [123M]

*Binary Alignment Map (BAM) files are compressed SAM files. Detailed descriptions of the (SAM) format specification can be found here: https://samtools.github.io/hts-specs/SAMv1.pdf

Sequencing Through the Ages

- Alignment data are stored in Sequence Alignment Map (*SAM) files, which have the following format:

1		2	3	4	5	6	7	8	9	10	11	12		
	J00113:339:HMJMNBBXX:3:1101:1144:1367	163	ENSMUST00000031314	1845	255	123M	=	1845	123	NGCAA…GGAAC	#AAFF…JFJJJ	NH:i:	HI:i:1	MC:Z:123M
*	J00113:339:HMJMNBBXX:3:1101:1144:1367	83	ENSMUST00000031314	1845	255	123M	=	1845	-123	TGCAA…GGANC	JFJFJ…FFA#A	NH:i:	HI:i:1	MC:Z:123M
	J00113:339:HMJMNBBXX:3:1101:5426:1367	419	ENSMUST00000174924	135	3	148M	=	144	159	NGTGA…AACAT	#AAFF…JJJJJ	NH:i:2	HI:i:1	MC:Z:150M
	J00113:339:HMJMNBBXX:3:1101:5426:1367	339	ENSMUST00000174924	144	3	150M	=	135	-159	CCGGG…ACCNC	JJJJJ…FFA#A	NH:i:2	HI:i:1	MC:Z:148M
	J00113:339:HMJMNBBXX:3:1101:5426:1367	163	ENSMUST00000175032	135	3	148M	=	144	159	NGTGA…AACAT	#AAFF…JJJJJ	NH:i:2	HI:i:2	MC:Z:150M
	J00113:339:HMJMNBBXX:3:1101:5426:1367	83	ENSMUST00000175032	144	3	150M	=	135	-159	CCGGG…ACCNC	JJJJJ…FFA#A	NH:i:2	HI:i:2	MC:Z:148M

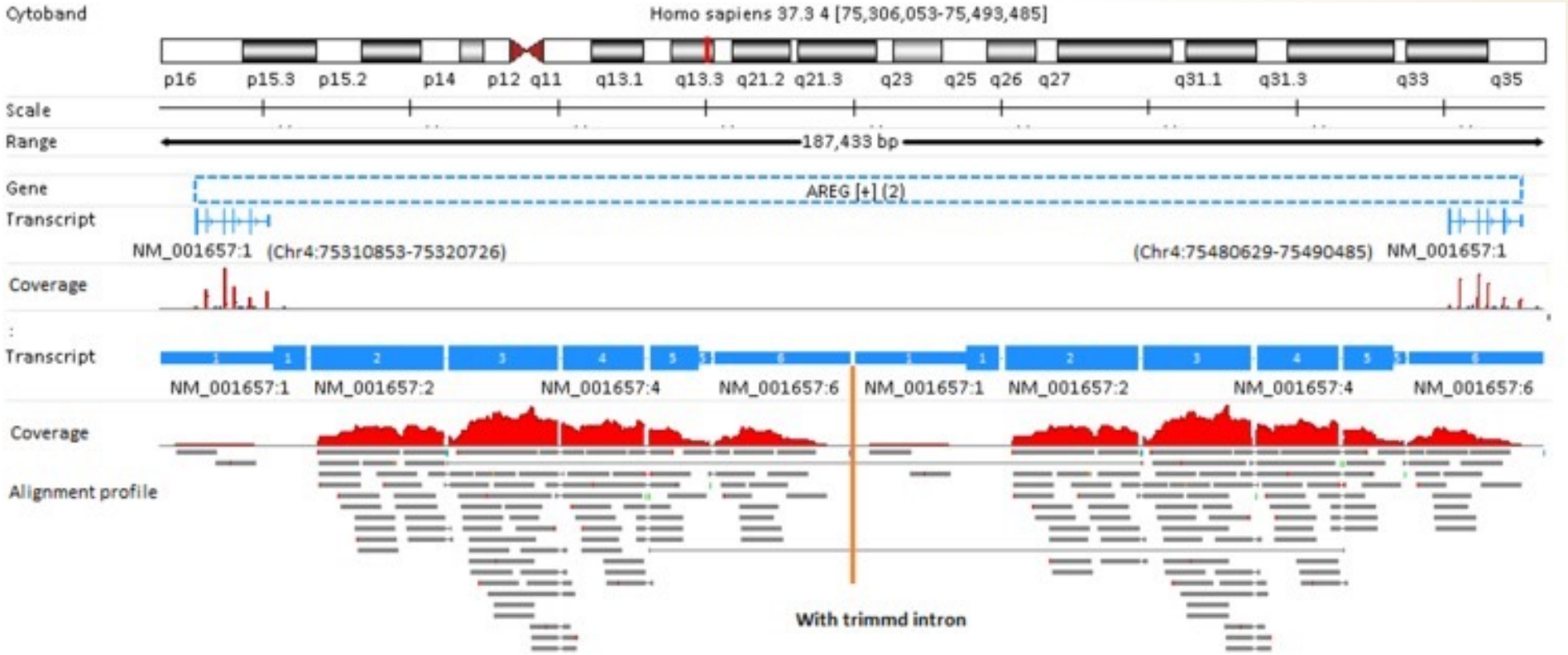
Column	Name	Description [*]
7	RNEXT	Reference name of the mate/next read: Reference sequence name of the next aligned read in the template, if it's the same, this is represented with an equal (=) sign. [=]
8	PNEXT	Position of the mate/next read: 1-based position of the next aligned read in the template. [1845]
9	TLEN	Observed template length: Length from the leftmost position of read 1 to the rightmost position of read 2 for aligned paired-end sequence data. [-123]

Column	Name	Description [*]
10	SEQ	Segment sequence: Sequence of the aligned trimmed read, found in line 2 of the trimmed fastq file. [TGCAA…GGANC]
11	QUAL	ASCII of Phred-scaled base quality +33: Base call quality scores, found in line 4 of the trimmed fastq file. [JFJFJ…FFA#A]
12+	Additional attributes	Additional SAM attributes that were added with the `--outSAMattributes` option in the STAR alignment command. [NH:i: HI:i:1 MC:Z:123M]

*Binary Alignment Map (BAM) files are compressed SAM files. Detailed descriptions of the (SAM) format specification can be found here: https://samtools.github.io/hts-specs/SAMv1.pdf

View Alignment Data

- If you want to take a closer look at how reads map to your reference genome, there are several tools available to help read and interpret BAM/SAM files:
  - Integrate Genomics Viewer (IGV): https://software.broadinstitute.org/software/igv/
  - BAMview: https://www.sanger.ac.uk/tool/bamview/
  - Integrated Genome Browser: https://www.bioviz.org/
  - GenomeView: https://genomeview.org/
  - SAMscope: https://bio.tools/samscope
  - **UCSC Genome Browser**: https://genome.ucsc.edu/

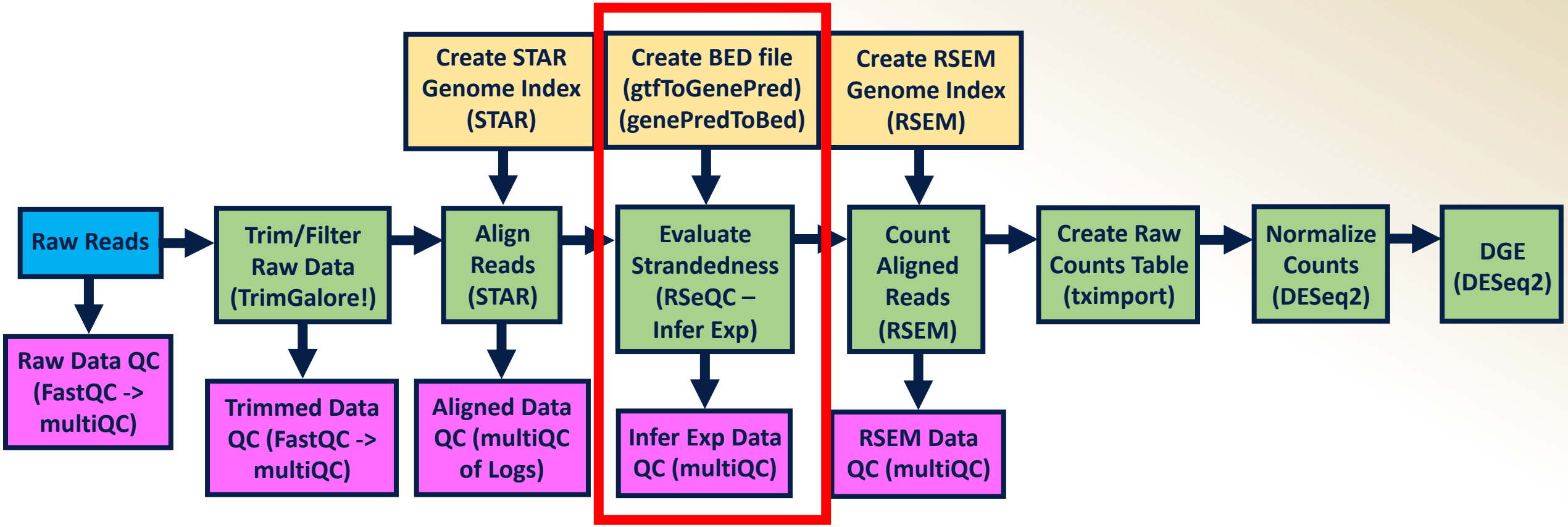


Zhao, Shanrong (2014). PLoS ONE 9(7): e101374. doi:10.1371/ journal.pone.0101374

We have aligned reads, now what?

- A lot of information can be derived from RNAseq alignment data
  - **Differential gene expression (DGE)** **Count aligned reads**
  - Identification of novel transcripts (and quantitation if performing 2-pass alignment)
  - Splice isoform identification and quantitation (splice aware aligners only)
  - Variant calls (SNPs and InDels)
  - Identification (and quantitation) of lncRNAs, pseudogenes, small RNAs (miRNA, small nuclear RNA, piRNA, snoRNA, etc.) (ribo-dpeletion method only)

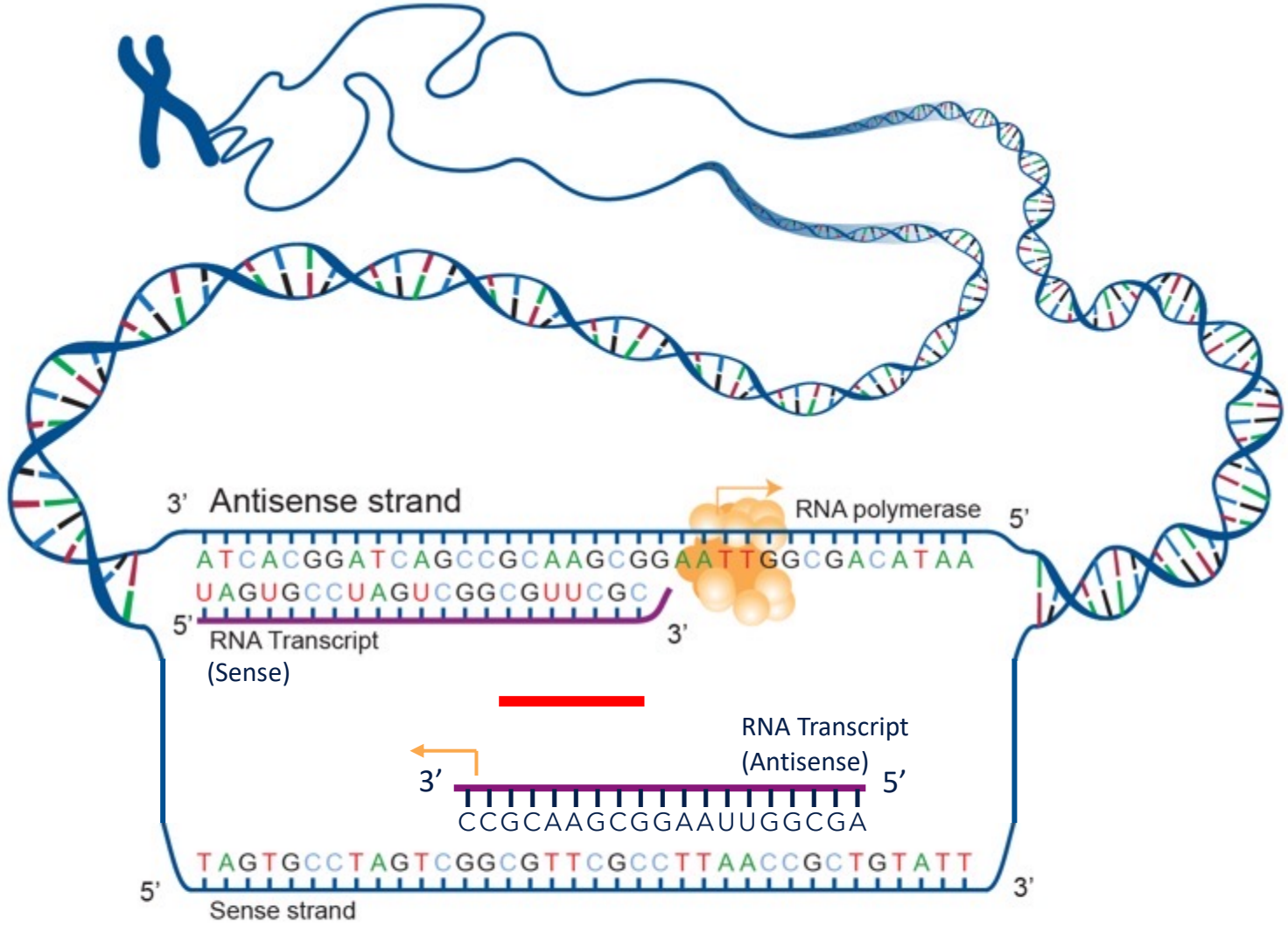
RNAseq Pipeline: Strandedness



Determine Read Strandedness

- RNA can be transcribed from the antisense DNA strand to make sense RNA (protein coding RNA)
- RNA can also be transcribed from the sense DNA strand to make antisense RNA (non-coding RNA)

- If a read aligns to a region where a sense RNA transcript and an antisense RNA transcript overlap, how do we know the transcript of origin?
- Can we know?

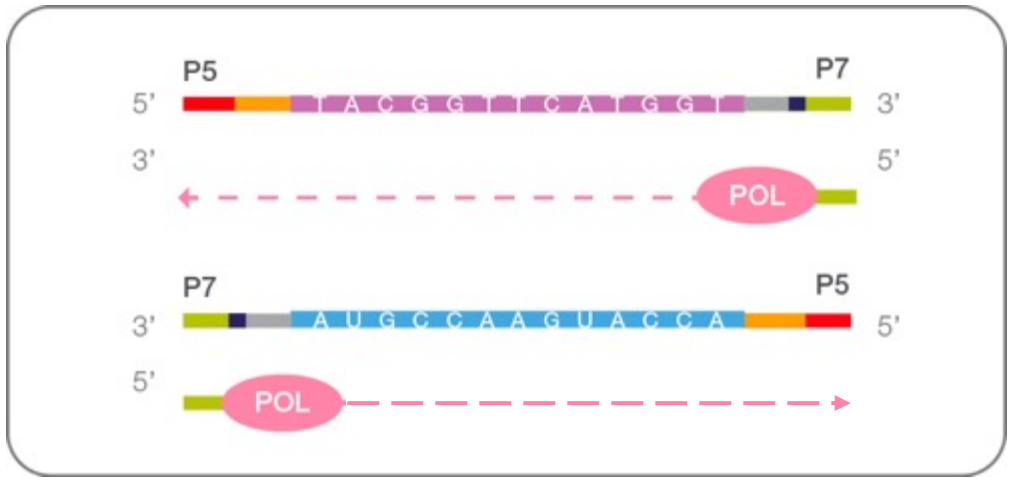


Modified from: https://www.genome.gov/genetics-glossary/antisense

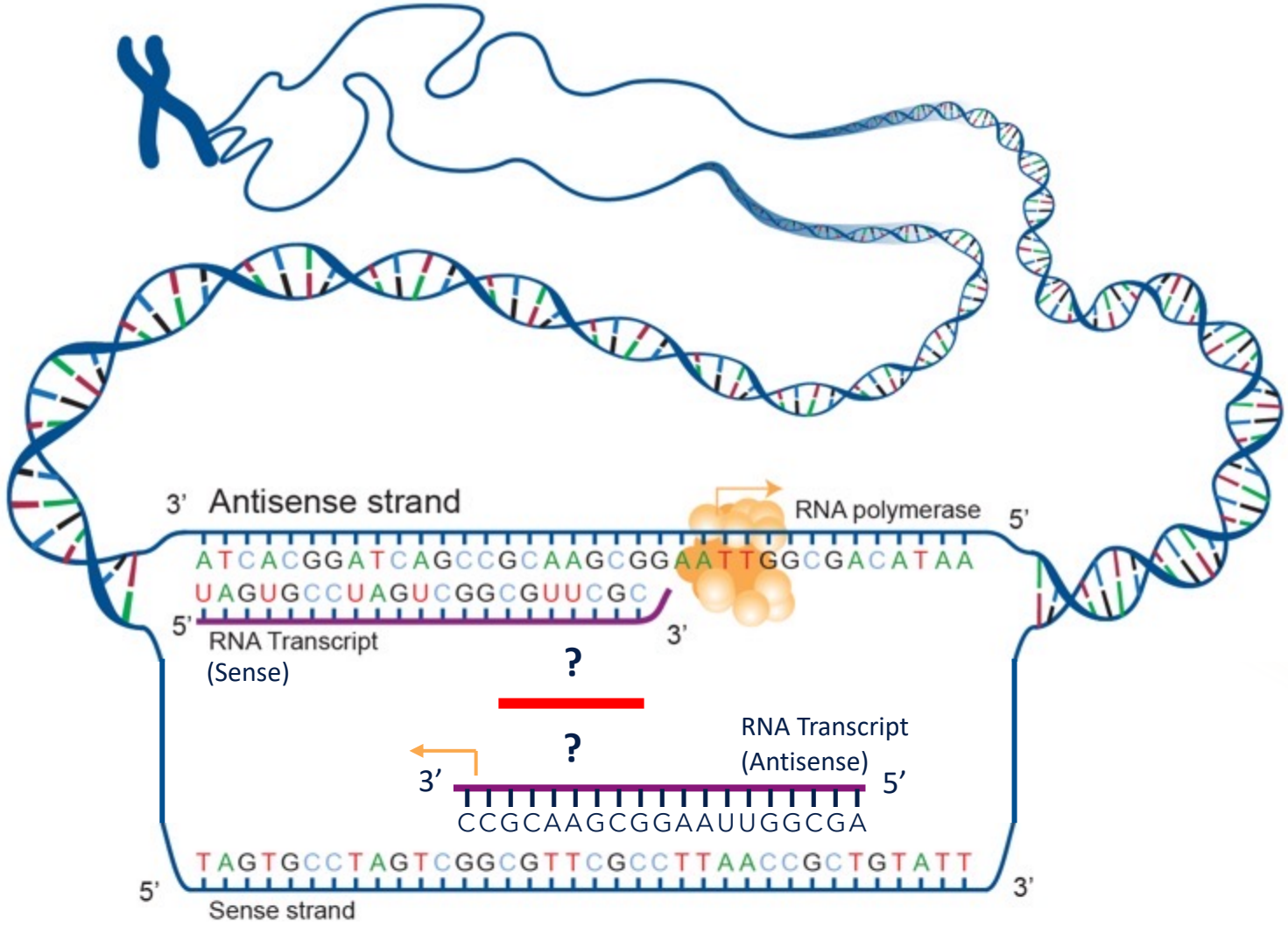
Determine Read Strandedness - Unstranded

- Prior to sequencing, cDNA fragments are enriched
- If both strands of the cDNA fragments are enriched, read orientation is not preserved

Figure 6 Enriching DNA Fragments



For reads that align to a region where a sense RNA transcript and an antisense RNA transcript overlap, we won't be able to know the transcript of origin

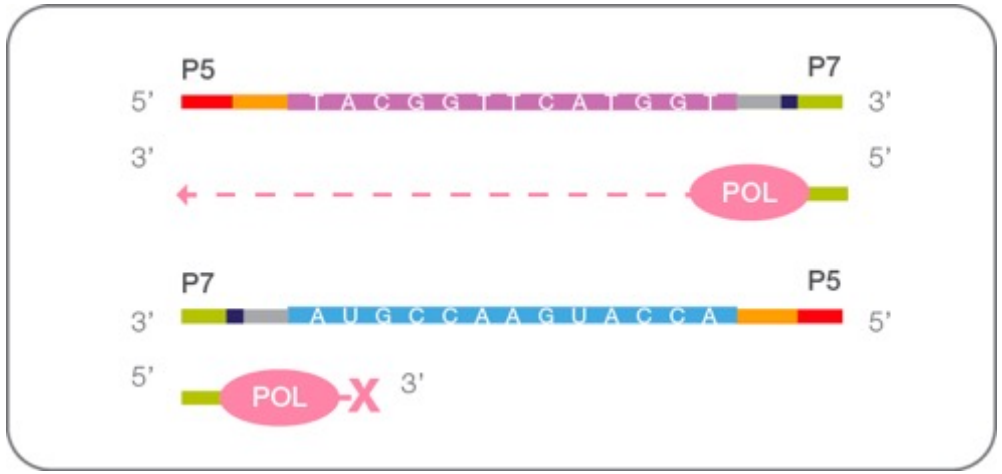


Modified from: https://www.genome.gov/genetics-glossary/antisense

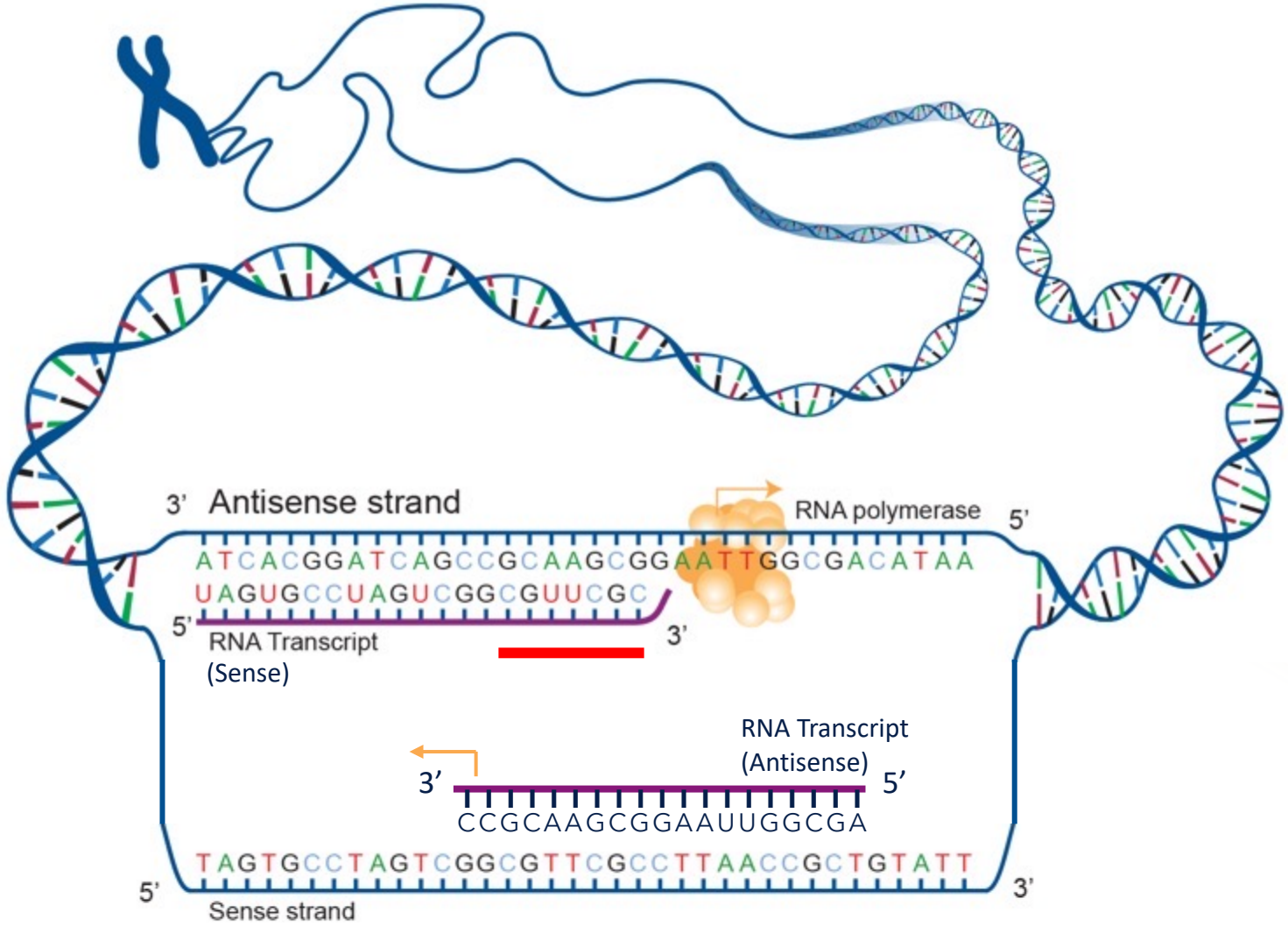
Determine Read Strandedness - Stranded

- If only one strand of the cDNA fragments is enriched, read orientation is preserved
- Preserving strand information during during sequencing helps to resolve read ambiguity in overlapping transcripts transcribed from opposite DNA strands

Figure 6 Enriching DNA Fragments



If we know the orientation of the reads relative to the reference transcripts, we will be able to know the transcript of origin for reads that align to a region where a sense and an antisense RNA transcript overlap.



Modified from: https://www.genome.gov/genetics-glossary/antisense

Following Strandedness from the Illumina TruSeq Stranded Kit

O = Original orientation **RC = Reverse Compliment of original orientation**

Figure 1 Ribo-Zero Depleting and Fragmenting RNA

Figure 2 Synthesizing First Strand cDNA

Figure 3 Synthesizing Second Strand cDNA

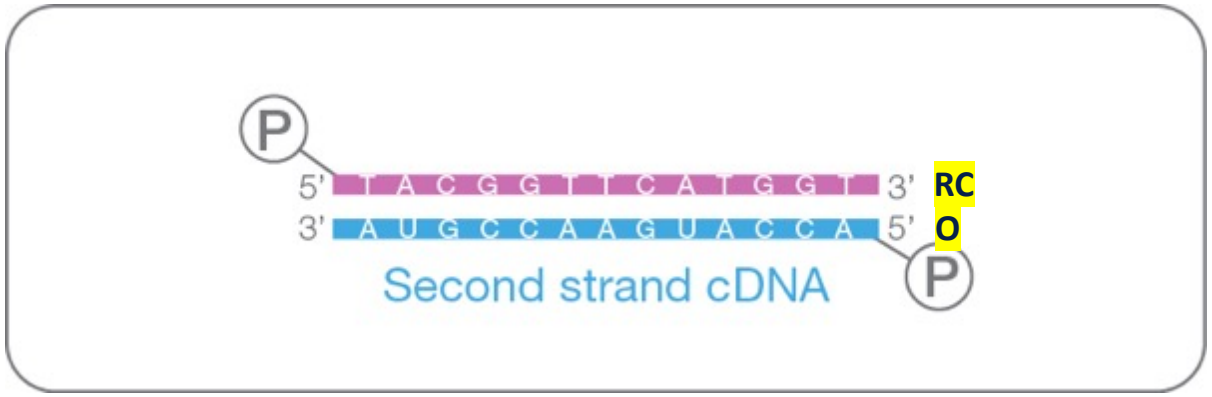
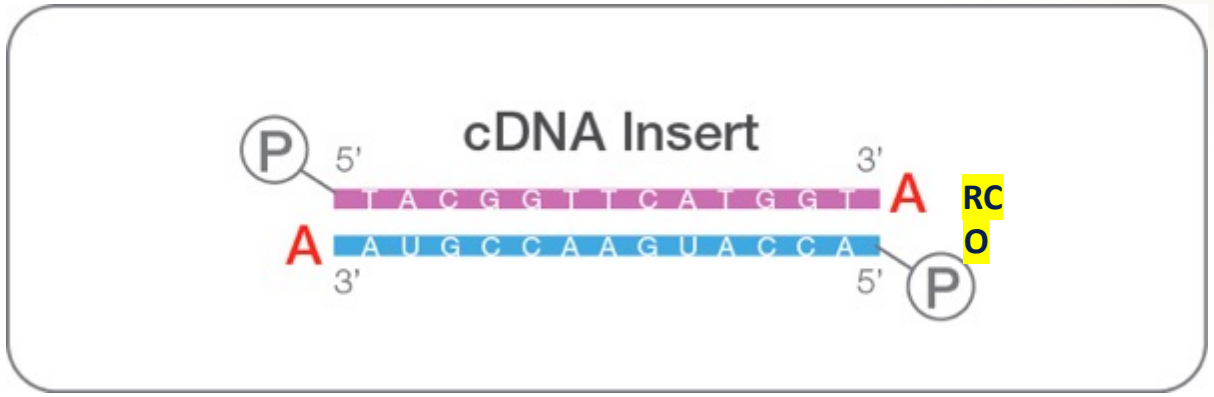


Figure 4 Adenylating 3' Ends



Following Strandedness from the Illumina TruSeq Stranded Kit

O = Original orientation
RC = Reverse Compliment of original orientation

Figure 5 Ligating Adapters

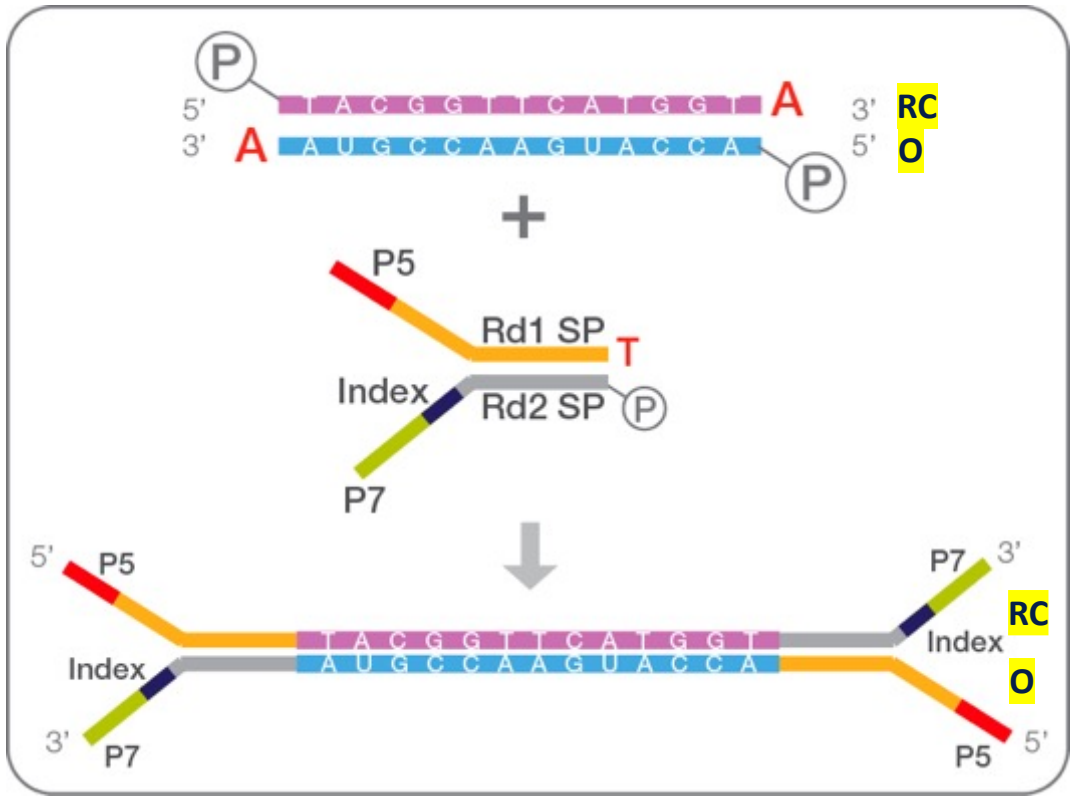


Figure 6 Enriching DNA Fragments

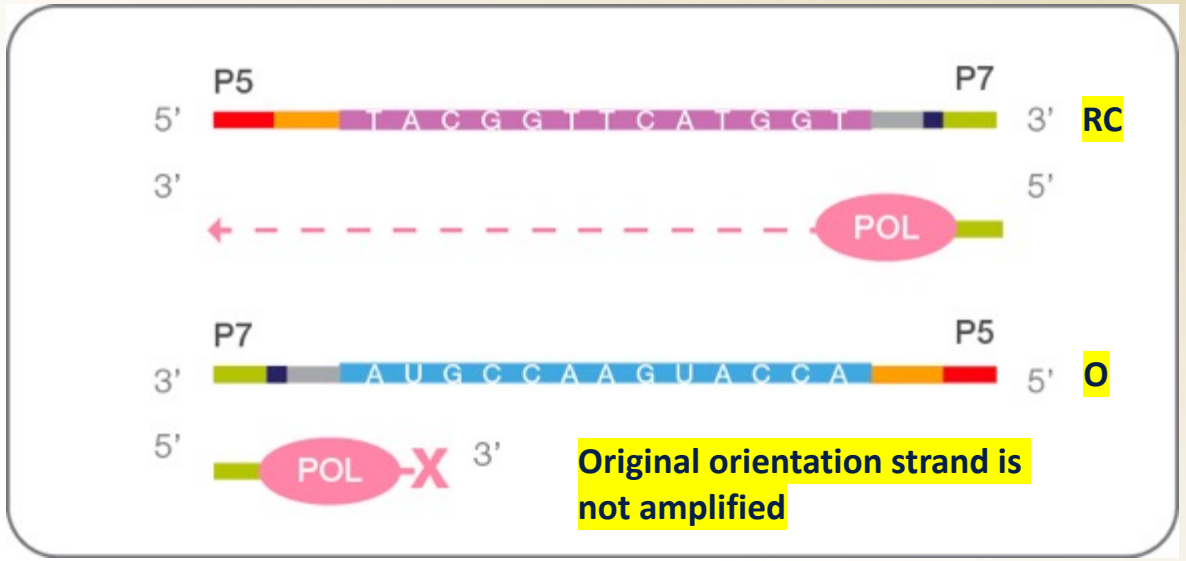
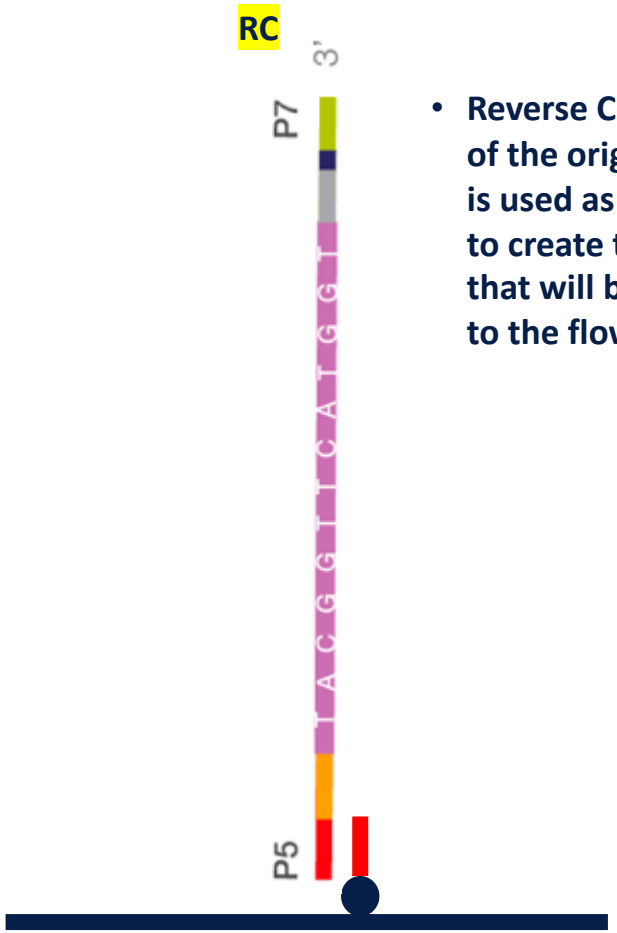
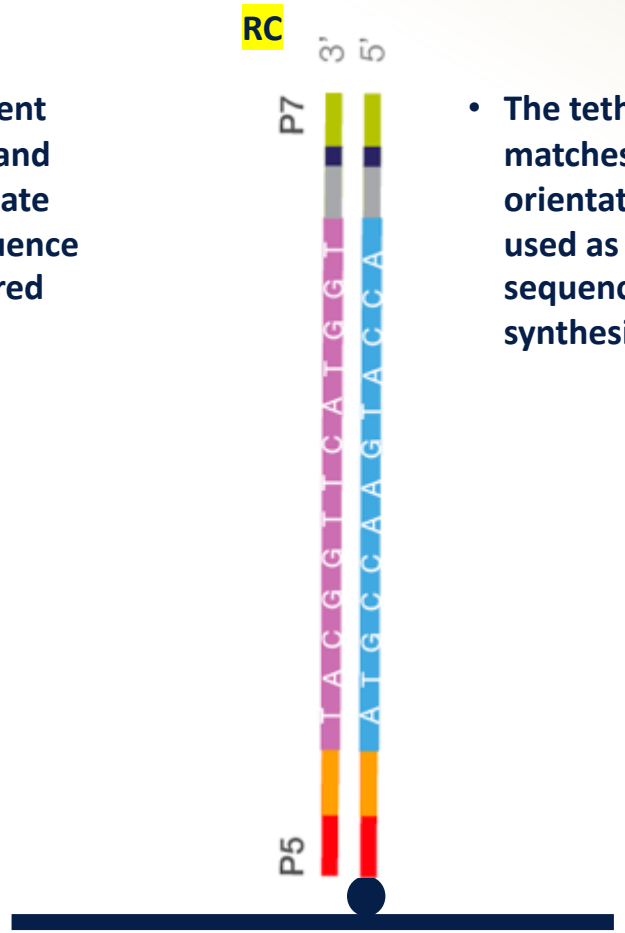


Figure 7 LS Final Library

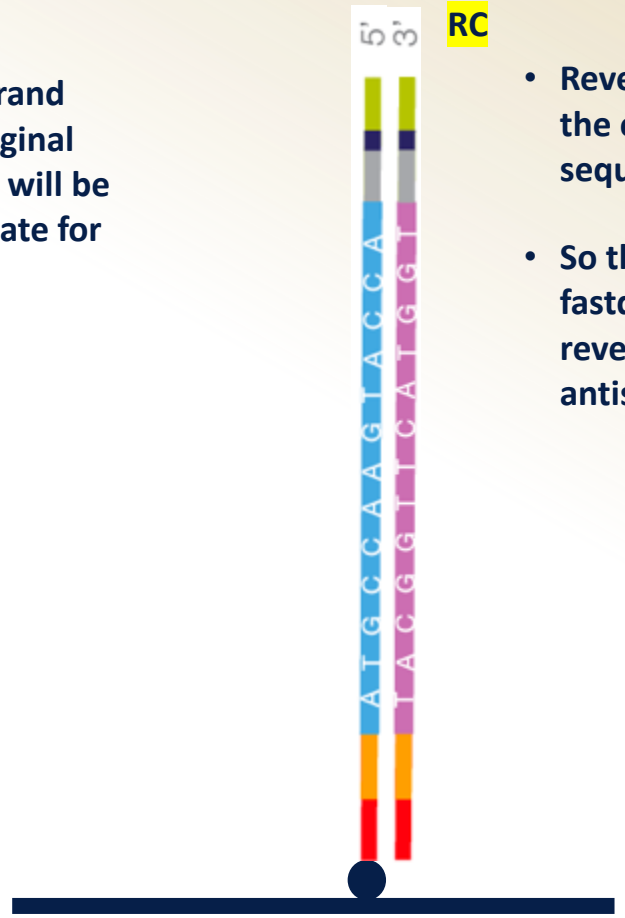
Following Strandedness from the Illumina TruSeq Stranded Kit



- Reverse Compliment of the original strand is used as a template to create the sequence that will be tethered to the flow cell



- The tethered strand matches the original orientation and will be used as a template for sequencing by synthesis



- Reverse Compliment of the original strand is the sequence being read
- So the sequences in the fastq files will be in the reverse compliment (or antisense) orientation

Determine Read Strandedness – RSeQC Infer Experiment

- To properly quantitate aligned reads, we need to know if the libraries were unstranded or stranded and if stranded, we also need to know read orientation relative to the reference
- Library prep kit documents should indicate if they are stranded or unstranded and if stranded, in which direction however:
  - Strandedness info may be difficult to find
  - Mixing and matching library prep kits and adapters could change the strandedness orientation
  - Strandedness info could be incorrectly reported or not reported at all in scientific papers
- To definitively determine strandedness of RNAseq data, GeneLab uses the RNA-seq Quality Control Package (RSeQC) Infer Experiment module
- RSeQC Infer Experiment compares the strandness of the reads with the strandness of transcripts
  - Read strandness is determined from the alignment data (genome aligned bam files)
  - Transcript strandness is determined from the reference transcripts (*note: the reference needs to be in Browser Extensible Data, BED, format*)

Generate A Reference BED File From A GTF File

- To evaluate strandedness, the RSeQC Infer Experiment module requires the reference to be in BED format
- We first create a gene predictions (genePred) table from the GTF file using UCSC's gtftogenepred program

genePred Table:

1	2	3	4	5	6	7	8	9	10
ENSMUST00000193812	1	+	3073252	3074322	3074322	3074322	1	3073252,	3074322,
ENSMUST00000082908	1	+	3102015	3102125	3102125	3102125	1	3102015,	3102125,
ENSMUST00000162897	1	-	3205900	3216344	3216344	3216344	2	3205900,3213608,	3207317,3216344,
ENSMUST00000159265	1	-	3206522	3215632	3215632	3215632	2	3206522,3213438,	3207317,3215632,

1	Transcript ID
2	Chromosome name
3	Strand orientation
4	Transcription start position
5	Transcription end position
6	Coding region start
7	Coding region end
8	Number of exons
9	Exon start positions
10	Exon end positions

Generate A Reference BED File From A GTF File

- To evaluate strandedness, the RSeQC Infer Experiment module requires the reference to be in BED format
- We first create a gene predictions (genePred) table from the GTF file using UCSC's gtftogenepred program
- Then convert the genePred table into the BED format using UCSC's genepredtobed program

BED File:

1	2	3	4	5	6	7	8	9	10	11	12
1	3073252	3074322	ENSMUST00000193812	0	+	3074322	3074322	0	1	1070,	0,
1	3102015	3102125	ENSMUST00000082908	0	+	3102125	3102125	0	1	110,	0,
1	3205900	3216344	ENSMUST00000162897	0	-	3216344	3216344	0	2	1417,2736,	0,7708,
1	3206522	3215632	ENSMUST00000159265	0	-	3215632	3215632	0	2	795,2194,	0,6916,

1	chrom - Chromosome name
2	chromStart - Transcription start position
3	chromEnd - Transcription end position
4	name - Transcript ID
5	score - 0-1000, associated with the Genome Browser track line
6	strand - Strand orientation
7	thickStart - Coding region (exon) start
8	thickEnd - Coding region (exon) end
9	itemRgb – R,G,B (255,0,0), associated with the Genome Browser display color
10	blockCount - Number of exons
11	blockSizes - Exon end – Exon start
12	blockStarts - Exon start – chromStart

Evaluate Strandedness with RSeQC Infer Experiment

infer_experiment.py *Parameters:

- -r: Specifies the path to the reference gene model in BED format.
- -i: Specifies the path to the input BAM file.
- -s: Specifies the number of reads to sample from the input BAM file.

infer_experiment.py output examples

Single-end non strand specific:

This **is** SingleEnd Data
Fraction of reads failed to determine: 0.0170
Fraction of reads explained by "++,--": 0.4834
Fraction of reads explained by "+-,-+": 0.4996

Note: ~half the reads are in the same orientation relative to the reference transcripts and ~half are in the opposite orientation (unstranded)

Single-end strand specific:

This **is** SingleEnd Data
Fraction of reads failed to determine: 0.0170
Fraction of reads explained by "++,--": 0.9669
Fraction of reads explained by "+-,-+": 0.0161

Note: Most reads are in the same orientation relative to the reference transcripts (sense)

http://rseqc.sourceforge.net/#infer-experiment-py

Evaluate Strandedness with RSeQC Infer Experiment

infer_experiment.py output examples

Paired-end non strand specific:

```
This is PairEnd Data
Fraction of reads failed to determine: 0.0172
Fraction of reads explained by "1++,1--,2+-,2-+": 0.4903
Fraction of reads explained by "1+-,1-+,2++,2--": 0.4925
```

Note: ~half the forward reads are in the same orientation relative to the reference and ~half are in the opposite orientation; same is true for the reverse reads (unstranded)

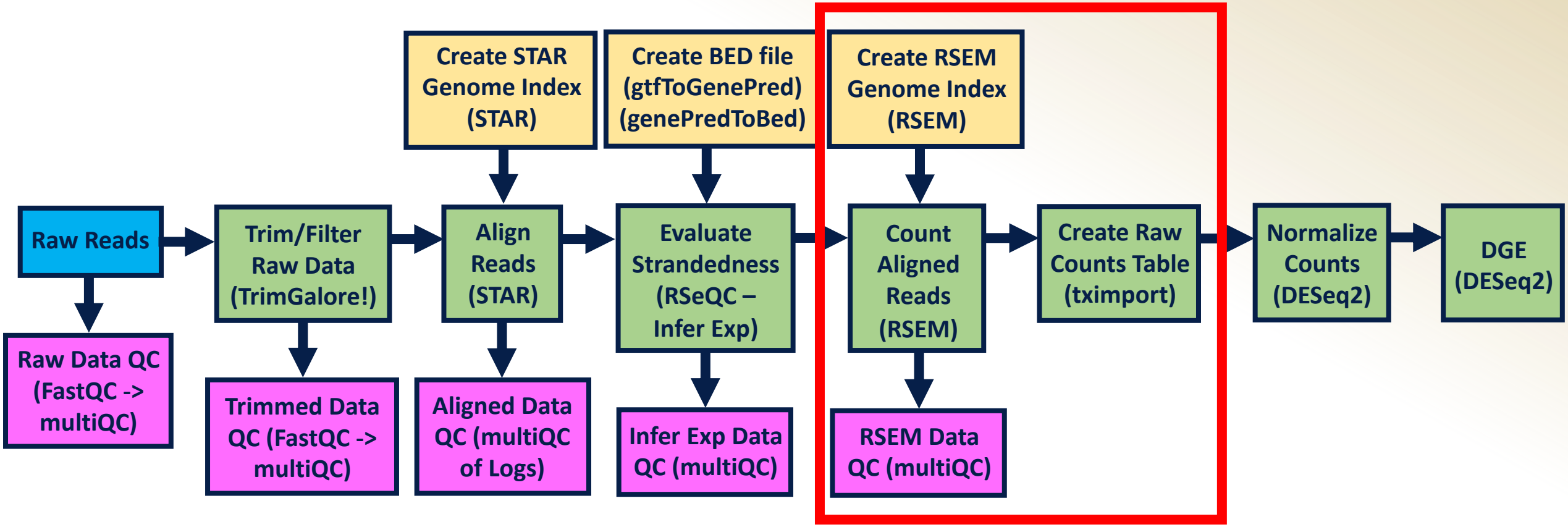
Paired-end strand specific:

```
This is PairEnd Data
Fraction of reads failed to determine: 0.0072
Fraction of reads explained by "1++,1--,2+-,2-+": 0.0487
Fraction of reads explained by "1+-,1-+,2++,2--": 0.9441
```

Note: Most of the forward reads are in the opposite orientation relative to the reference and most of the reverse reads are in the same orientation relative to the reference transcripts (antisense)

http://rseqc.sourceforge.net/#infer-experiment-py

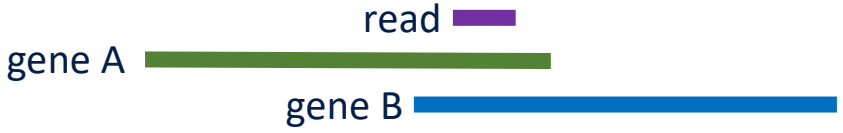
RNAseq Pipeline: Quantitation



Count Aligned Reads

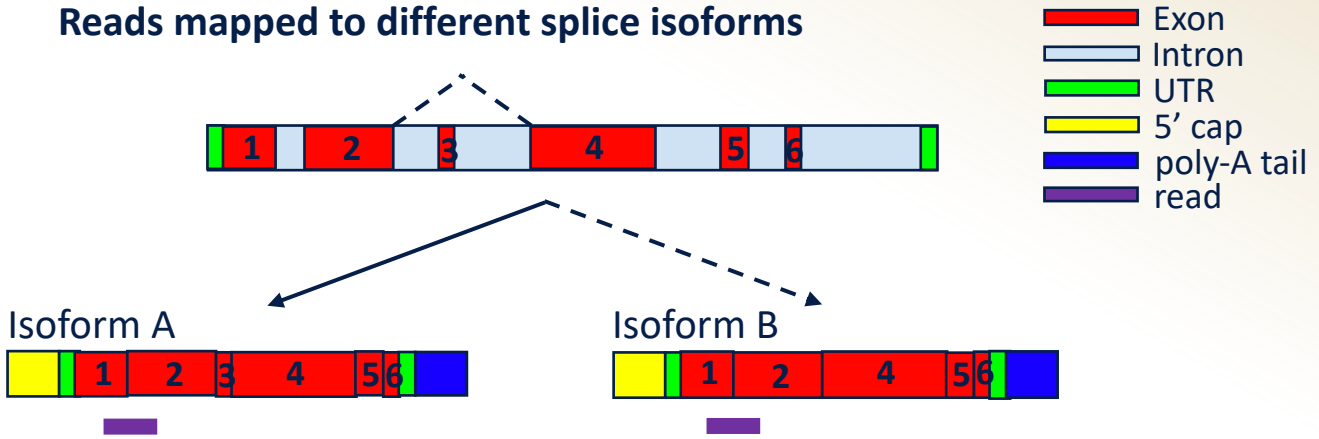
- To determine the expression level of each gene in each sample, we next have to assign/count the aligned reads
- Several tools are available to quantitate gene expression data, some of which are also aligners, and they use different algorithms (non-weighted and weighted) for quantitation
- Some of the main differences among quantification tools are how they handle multi-mapped reads, reads mapped to multiple genes or splice isoforms, and the outputs generated

Multi-mapped reads (different genes)



Should the read be assigned to gene A or gene B?

Reads mapped to different splice isoforms



Should the read be assigned to Isoform A or Isoform B?

Non-weighted vs Weighted Quantification

Non-weighted quantitation tools:

- A read is counted if it overlaps (1nt or more) one gene.
- Multi-mapped reads will either not be quantitated or quantitation will be assigned based on the tool option selected
- Generates gene quantitation in raw counts
- Examples include:

➢ **HTSeq-count**: Options for multi-mapped reads: none, all, fraction, random. https://htseq.readthedocs.io/en/master/count.html

➢ **featureCounts**: Options for multi-mapped reads: none or all. http://bioinf.wehi.edu.au/featureCounts/

➢ **STAR** (uses default HTSeq-count): Uses default HTSeq-count (no multi-mapped reads are counted). https://github.com/alexdobin/STAR

Weighted quantitation tools:

- Utilizes the maximum likelihood estimation (MLE) method to assign reads to genes and/or transcripts.
- Examples include:

➢ **Salmon/Kallisto** (pseudo-aligners): Both use MLE, Salmon also uses the Variational Bayes (VB) method. Generates transcript quantitation (TPM and predicted raw counts). https://salmon.readthedocs.io/en/latest/salmon.html https://pachterlab.github.io/kallisto/

➢ **RNA-Seq by Expectation-Maximization (RSEM)**: RSEM quantitates uniquely aligned reads then uses MLE abundance estimates to assign multi-mapped reads. Generates gene and transcript quantitation (weighted **raw counts**, TPM, FPKM). https://deweylab.github.io/RSEM/

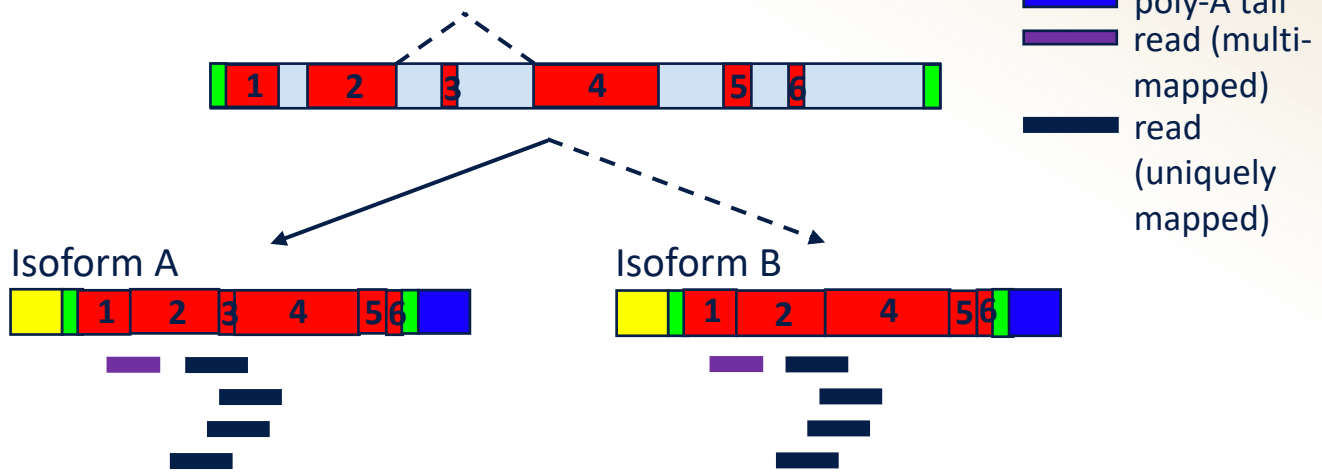
Count Aligned Reads with RSEM

- How do we deal with multi-mapping and splice isoform quantification issues?
- The RSEM software package was designed to address these issues and "rescue" reads that are not uniquely mapped by allocating them to transcripts using a 3-step algorithm:
  1. Estimate abundances based on uniquely mapped reads only.
  2. For each read that maps to multiple locations (multiread), divide it between the transcripts to which it maps, proportional to their abundances estimated in the first step. – Uses the maximum likelihood estimation (MLE) method
  3. Compute abundances based on updated counts for each transcript.

Multi-mapped reads (different genes)

Should the read be assigned to gene A or gene B?

Reads mapped to different splice isoforms



Should the read be assigned to Isoform A or Isoform B?

Building a RSEM Index

- To interpret the information provided in the BAM file, similar to STAR, the RSEM program first requires the generation of a RSEM index using the reference genome and associated GTF files from the sample organism.

RSEM *Parameters:

- rsem-prepare-reference: Calls the RSEM program that will extract reference transcripts from the reference genome using the gene transfer file and create a RSEM index.
- --gtf: Specifies the uncompressed file(s) containing annotated transcripts are in the standard GTF format.
- Positional argument to specify one or more uncompressed fasta file(s) containing the genome reference sequences.
- Positional argument to specify the path to the directory where the RSEM index will be stored and the prefix desired for the RSEM reference files.

* These parameters are specific to the GeneLab standard pipeline, for a complete list of RSEM prepare reference parameters, go here: https://deweylab.github.io/RSEM/rsem-prepare-reference.html

Quantitation with RSEM

RSEM *Parameters:

- rsem-calculate-expression: Calls the RSEM program that will estimate gene and isoform expression from RNAseq data.
- --alignments: Indicates that the input file contains alignments in sam, bam, or cram format.
- --bam: Specifies that the input alignments are in bam format.
- --paired-end: Indicates that the input reads are paired-end reads.
- --seed: The seed for the random number generators used in calculating posterior mean estimates and credibility intervals; must be a non-negative 32-bit integer.
- --seed-length: Instructs RSEM to ignore any aligned read if it or its mate's (for paired-end reads) length is less than the value indicated (20bp)
- --estimate-rspd: Instructs RSEM to estimate the read start position distribution (rspd) from the data.
- --no-bam-output: Instructs RSEM not to output any bam file.
- --strandedness: Defines the strandedness of the RNAseq reads; none = reads are unstranded, forward = most (forward) reads are sense relative to the reference, reverse = most (forward) reads are antisense relative to the reference.
- Positional argument to specify the path to the input BAM file(s).
- Positional argument to specify the path to the directory where the RSEM reference is stored and its prefix.
- Positional argument to specify the path to and prefix for the output file names.

* These parameters are specific to the GeneLab standard pipeline, for a complete list of RSEM calculate expression parameters, go here: https://deweylab.github.io/RSEM/rsem-calculate-expression.html

RSEM Count Data (*genes.results)

- RSEM outputs two files containing expression estimates per gene (*genes.results) and per isoform (*isoforms.results)
- For DGE analysis we will use estimates per gene, so let's take a look at the *genes.results file format:

1	2	3	4	5	6	7
gene_id	transcript_id(s)	length	effective_length	expected_count	TPM	FPKM
ENSMUSG00000000001	ENSMUST00000000001	3262.00	3083.74	5984.00	12.21	30.30
ENSMUSG00000000003	ENSMUST00000000003,ENSMUST00000114041	799.50	621.24	0.00	0.00	0.00
ENSMUSG00000000028	ENSMUST00000000028,ENSMUST00000096990,ENSMUST00000115585,ENSMUST00000231819	1921.73	1743.46	68.00	0.25	0.61
* ENSMUSG00000000056	ENSMUST00000103015,ENSMUST00000151088,ENSMUST00000154047	3894.46	3716.21	2108.00	3.57	8.86
ENSMUSG00000000058	ENSMUST00000000058,ENSMUST00000115459,ENSMUST00000115462	2604.61	2426.35	150.00	0.39	0.97
ENSMUSG00000000078	ENSMUST00000000080,ENSMUST00000221734,ENSMUST00000222857	3190.68	3012.42	838.00	1.75	4.34

Column	Name	Description [*]
1	gene_id	Gene name according to the database used (in this example, we used ensembl genome and GTF files, so the gene names are ensembl IDs) [ENSMUSG00000000056]
2	transcript_id(s)	Comma-separated list of all the transcripts derived from the respective gene in column 1 [ENSMUST00000103015,ENSMUST00000151088, ENSMUST00000154047]
3	length	The weighted average of the respective gene's transcripts' lengths [3894.46]
4	Effective_length	The weighted average of the respective gene's transcripts' effective lengths, which are weighted by each transcript's isoform percentage [3716.21]

Column	Name	Description [*]
5	**expected_count**	The sum of the estimates of the number of read fragments that are derived from each transcript of the respective gene (we will use these count values to calculate DE). [2108.00]
6	TPM	Transcripts per million, which is a relative measure of transcript abundance - this value is summed over all transcripts for each respective gene to generate the gene TPM value. [3.57]
7	FPKM	Fragments Per Kilobase of transcript per Million mapped reads, which is another relative measure of transcript abundance - this value is summed over all transcripts for each respective gene to generate the gene FPKM value. [8.86]

*Detailed descriptions of the RSEM calculate expression outputs can be found here: http://deweylab.biostat.wisc.edu/rsem/rsem-calculate-expression.html#output

Raw Counts Table

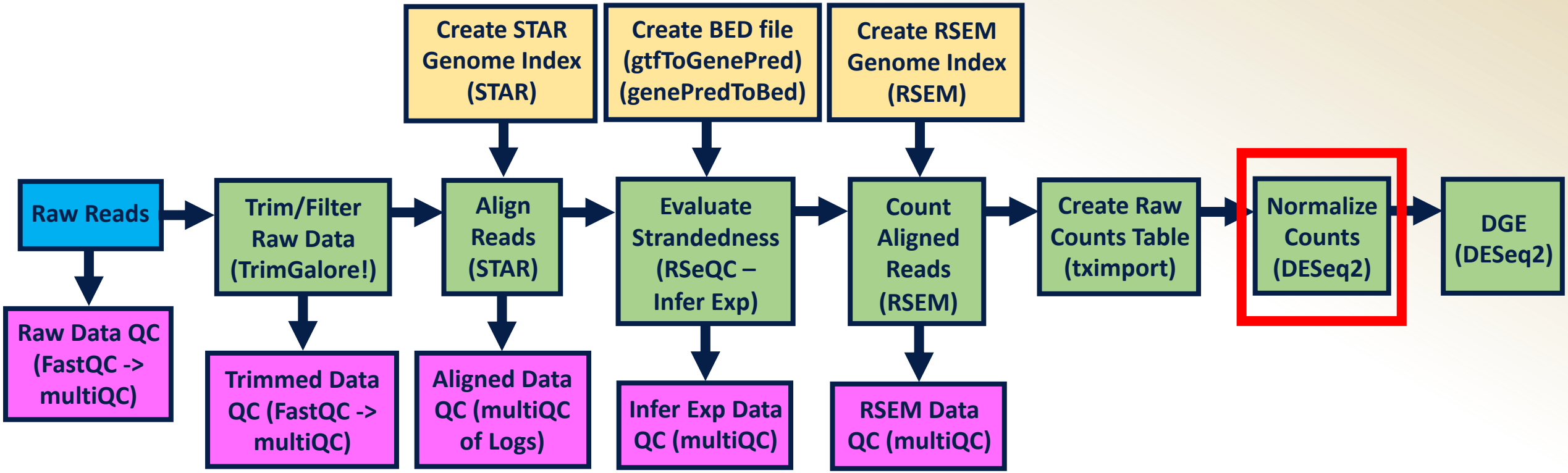
Sample Names

Gene IDs

	FLT_Rep1	FLT_Rep2	FLT_Rep3	FLT_Rep4	FLT_Rep5	GC_Rep1	GC_Rep2	GC_Rep3	GC_Rep4	GC_Rep5
ENSMUSG00000000001	5984	4900	5928	3654	5835	5633	6847	7420	6371	7912
ENSMUSG00000000003	0	0	0	0	0	0	0	0	0	0
ENSMUSG00000000028	68	36	55	65	45	32	43	68	24	50
ENSMUSG00000000031	13	18	9	13	9	12	11	11	32	32
ENSMUSG00000000037	10	9	5	7	6	8	2	1	7	4
ENSMUSG00000000049	68198	48651	67572	54265	79771	71035	105690	93444	92665	118393
ENSMUSG00000000056	2108	703	1336	1547	1947	1966	3470	2625	2432	4253
ENSMUSG00000000058	150	189	158	83	138	164	211	280	208	278
ENSMUSG00000000078	838	654	912	409	823	908	984	1000	981	934
ENSMUSG00000000085	392	239	444	455	455	263	404	339	380	323
ENSMUSG00000000088	3689	2896	3108	2286	3829	4084	4934	4742	3723	5283
ENSMUSG00000000093	281	129	210	170	298	144	208	238	257	188
ENSMUSG00000000094	0	5	1	0	1	1	2	4	0	1
ENSMUSG00000000103	0	0	0	0	0	0	0	0	0	0
ENSMUSG00000000120	379	196	314	156	265	354	343	411	346	352
ENSMUSG00000000125	0	3	0	0	0	0	0	0	0	0
ENSMUSG00000000126	18	13	8	24	13	15	19	17	14	10
ENSMUSG00000000127	685	479	853	719	733	537	824	652	672	721
ENSMUSG00000000131	2241	1483	2713	2371	2835	1620	3007	2357	2551	2571
ENSMUSG00000000134	801	694	1010	794	889	579	788	905	815	766

RNAseq
Data Analysis

RNAseq Pipeline: Normalize Counts

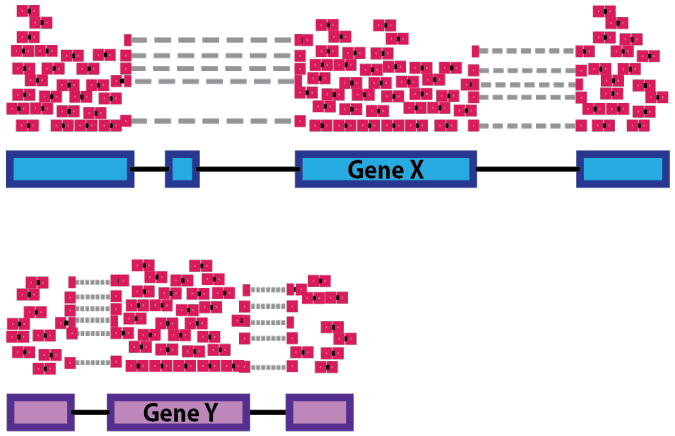


Normalize Count Data

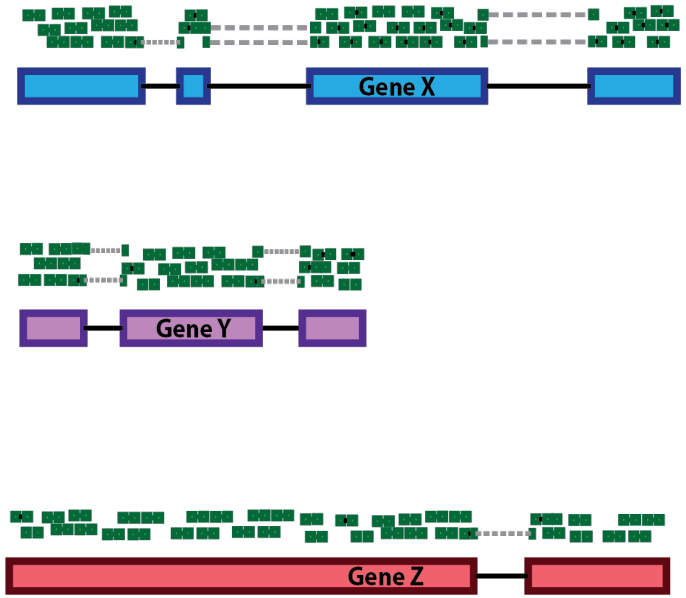
- We now have quantitated gene expression data – are we ready to run DGE analysis? **Not quite…**
- Although we've counted the number of reads aligned to each gene, we have not accounted for differences in read depth, gene length, or RNA composition – why is this necessary?

Read depth

Sample A Reads (100M)

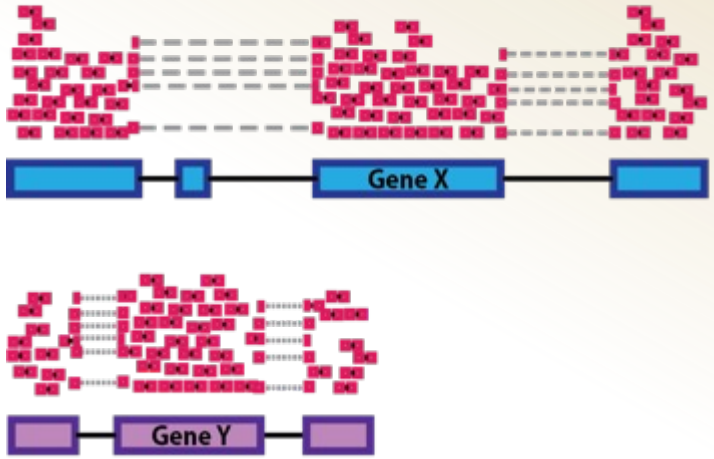


Sample B Reads (50M)



Gene length

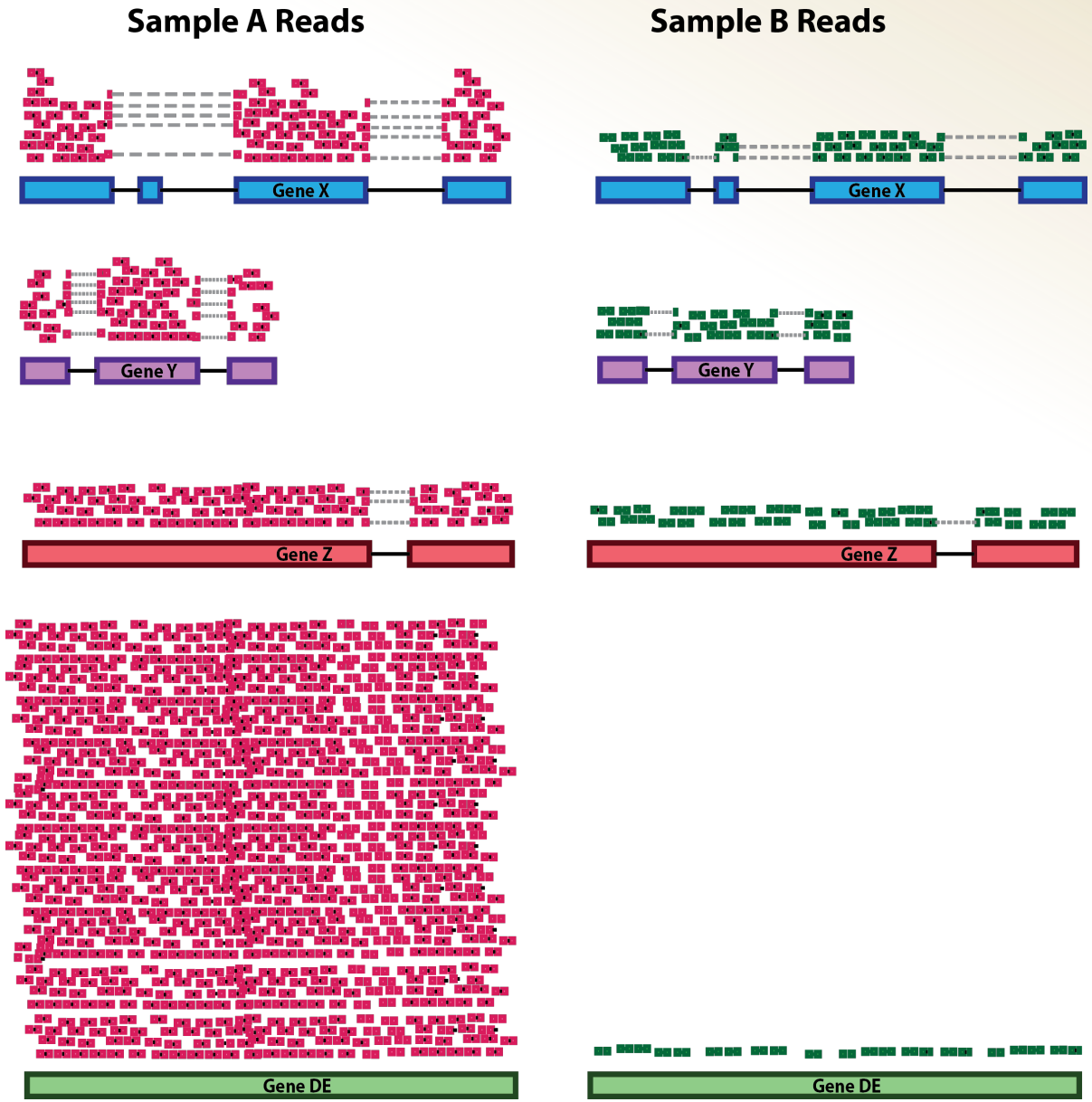
Sample A Reads



https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html

Normalize Count Data

RNA Composition



Which of the 3 factors discussed need to be accounted for when performing DE analysis?

Read depth and RNA composition

https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html

Normalization Methods

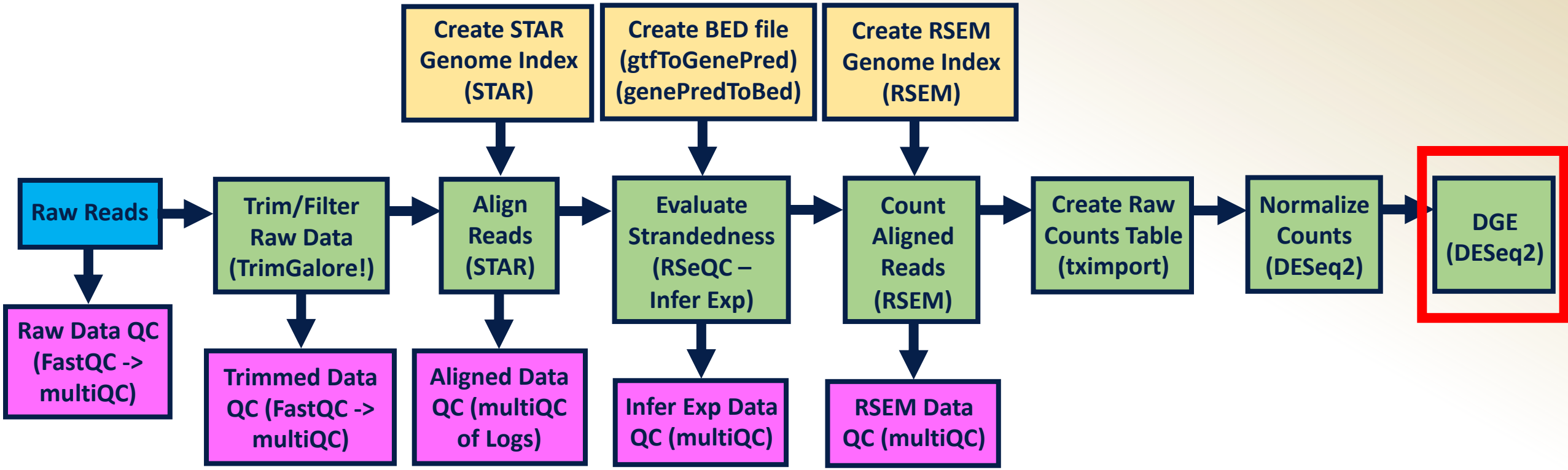
Normalization method (tools)	Description	Accounts for differences in	Uses
Counts per million, CPM (limma-voom)	The total reads in each sample is divided by 1,000,000 to create a scaling factor. Gene counts are then divided by the ”per million” scaling factor.	Sequencing depth	To compare gene counts between replicates of the same group (not for within sample comparisons or DE analysis).
Reads/fragments per kilobase million, RPKM/FPKM (RSEM)	CPM values are divided by the length of the gene (in kilobases).	Sequencing depth and gene length	To compare gene counts between genes within a sample (not for between sample comparisons or DE analysis).
Transcripts per kilobase million, TPM (RSEM, Salmon, Kallisto)	Similar to RPKM/FPKM but differs in the order of operations. Read counts are divided by the length of each gene (in kilobases). The sum of all RPK values in each sample is divided by 1,000,000 then each RPK value is divided by the “per million” scaling factor.	Sequencing depth and gene length	To compare gene counts within a sample or between samples of the same group (not for DE analysis).

Normalization Methods

Normalization method (tools)	Description	Accounts for differences in	Uses
Median of ratios (DESeq2)	A *size factor* is calculated for each sample by dividing the median ratio of all gene counts by the geometric mean of each gene across all samples. The raw counts for each sample are then divided by the sample-specific size factor for each gene.	Sequencing depth and RNA composition	To compare gene counts between samples (not for within sample comparisons) and for DE analysis.
Trimmed mean of M values, TMM (EdgeR) and gene length corrected TMM (GeTMM)	The total read count (or total RPK for GeTMM) is corrected by a normalization factor (calculated using a weighted mean of the log expression ratios between samples after trimming) and scaled to per million reads.	Sequencing depth and RNA composition (note: GeTMM accounts for gene length)	To compare gene counts between and within (only if using GeTMM) samples and for DE analysis.

- The type of between-sample normalization used greatly impacts differential expression analysis.
- We will discuss the DESeq2 normalization method in greater detail in the subsequent RNAseq statistics review lecture.
- After count data are normalized, we can begin differential gene expression analysis.

RNAseq Pipeline: DGE



Differential Gene Expression (DGE) Analysis

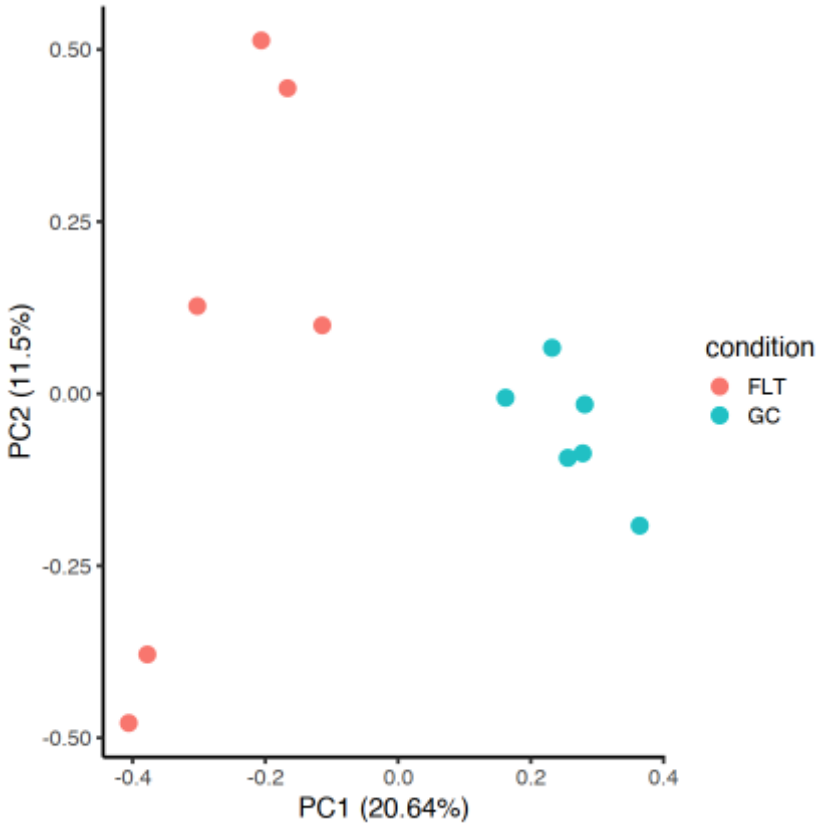
- We are now ready to determine the probability of each gene's expression being significantly different in one of our groups of interest via pair-wise (FLT vs. GC) and/or multiple comparisons testing.
- Several tools are available for DGE analysis; they differ in the type of distribution used to create a model to fit the count data for statistical testing and the statistical test used.

DE tool	Distribution assumption/model	Statistical test
DESeq2 https://bioconductor.org/packages/DESeq2/	Negative binomial	Wald test for pairwise comparisons; likelihood ratio test for multiple comparison testing
limma-voom https://bioconductor.org/packages/limma/	Similar to *t*-distribution with empirical Bayes approach	Moderated *t*-test
EdgeR https://bioconductor.org/packages/edgeR/	Negative binomial	Exact test
baySeq https://bioconductor.org/packages/baySeq	Negative binomial	Posterior probability through Bayesian approach
EBSeq https://bioconductor.org/packages/EBSeq/	Negative binomial-beta empirical Bayes model	Posterior probability through Bayesian approach

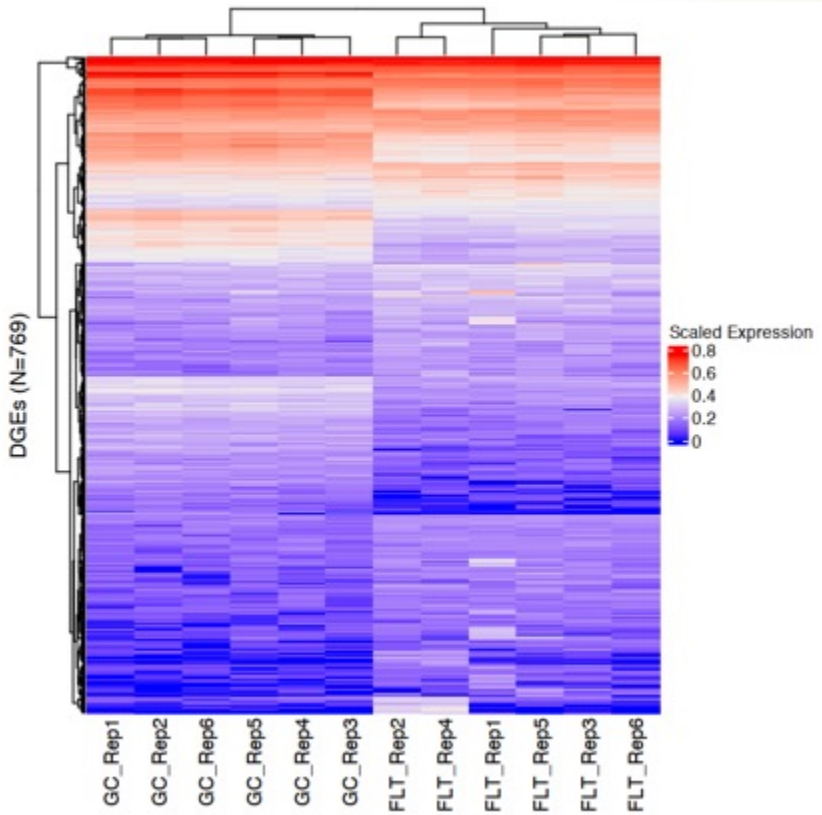
Visualize Count and DGE Data

Data visualization is used to help interpretate count and DGE data - we will generate the following plots during this bootcamp:

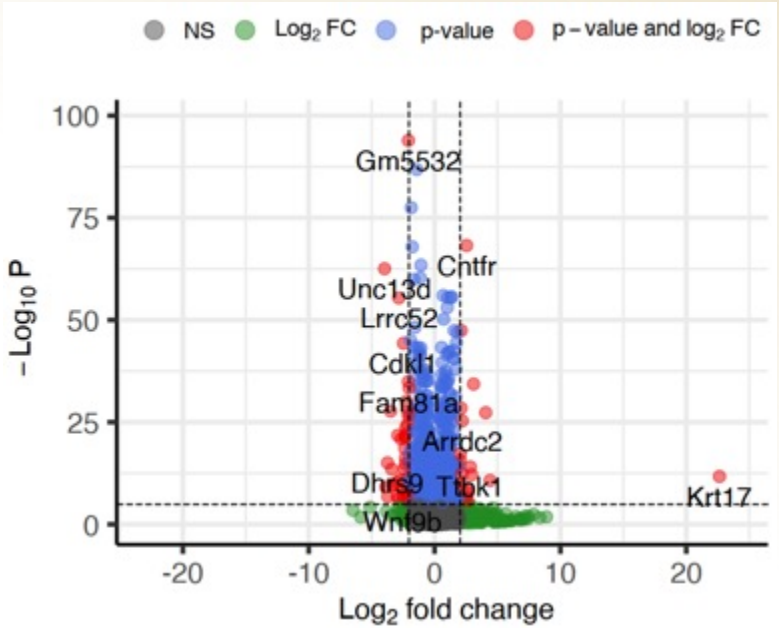
Principal Component Analysis (PCA)



Clustered Heatmap



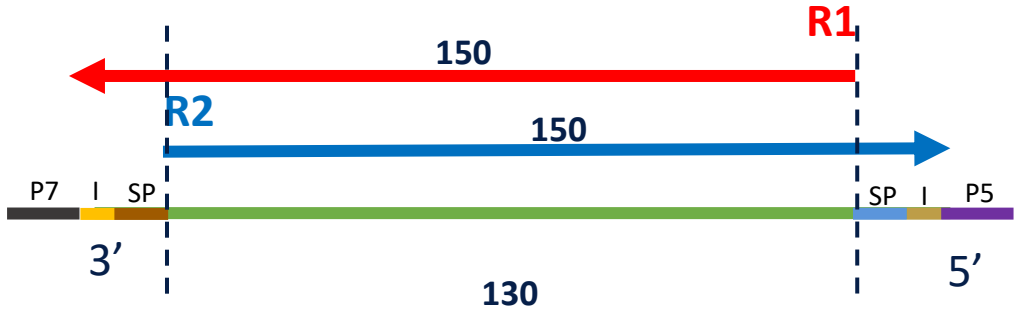
Volcano Plot



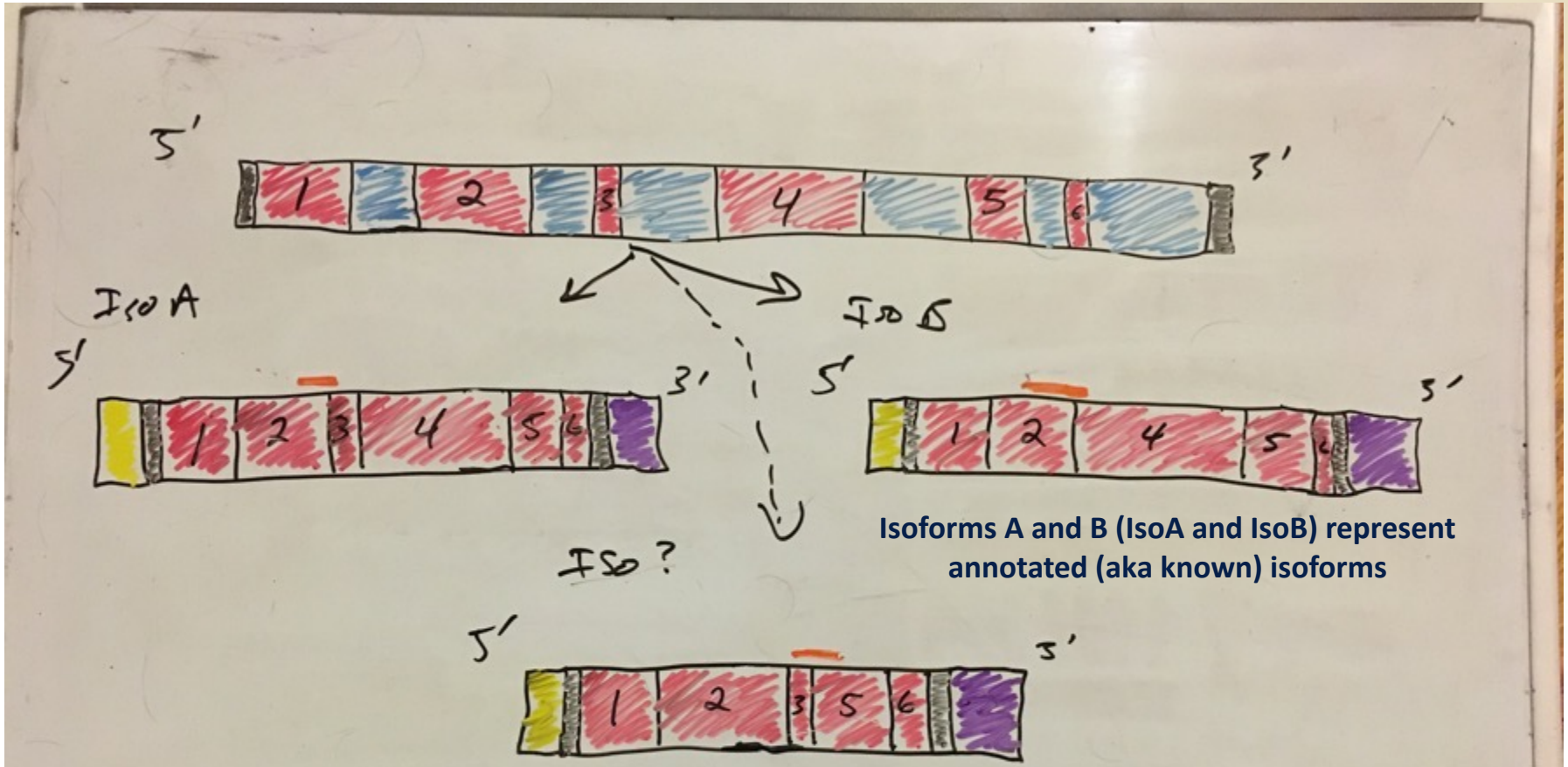
Questions?

Extra Slides

How do adapters get sequenced?



STAR Overhang Options for Isoforms



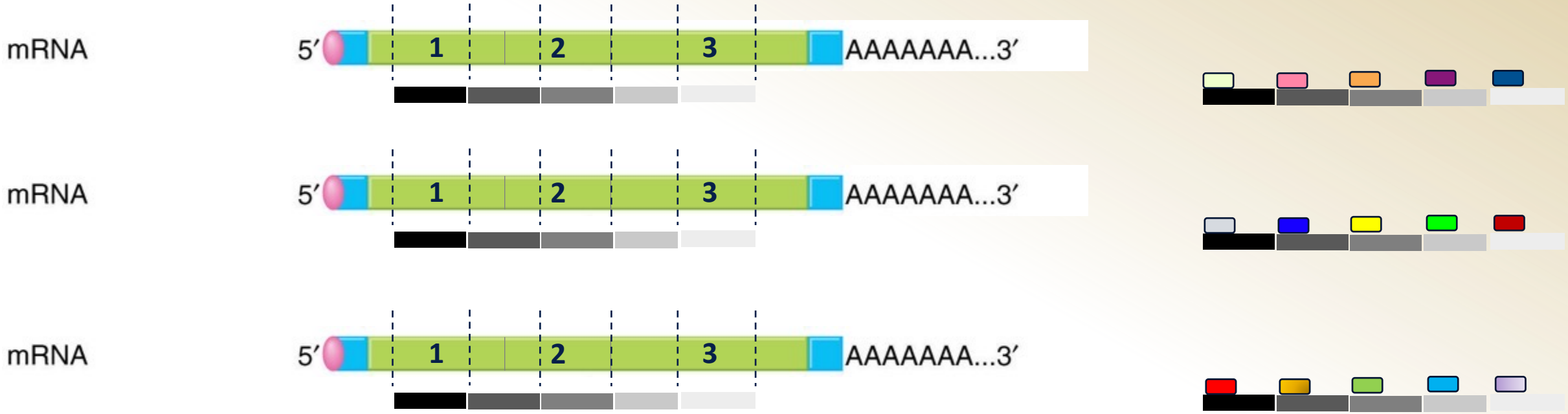
Isoforms A and B (IsoA and IsoB) represent annotated (aka known) isoforms

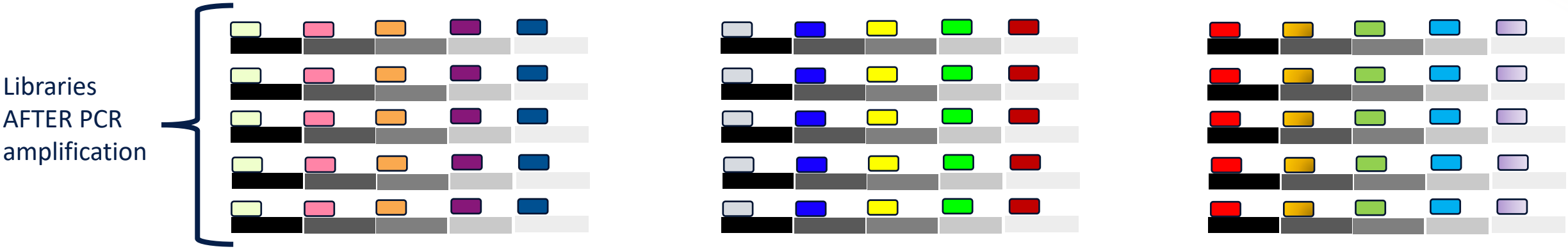
Iso? represents an unannotated (aka unknow) isoform

The amount of overhang allowed across splice junctions for annotated isoforms is defined with the `**--alignSJDBoverhangMin` STAR option**

The amount of overhang allowed across splice junctions for unannotated isoforms is defined with the `**--alignSJoverhangMin` STAR option**

UMIs to Detect Technical Duplicates





Determine Insert Length