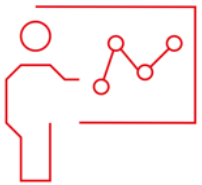
Introduction to RNA-Seq: Mapping & Aligning



Wandrille Duchemin

Alignment vs. pseudoalignment

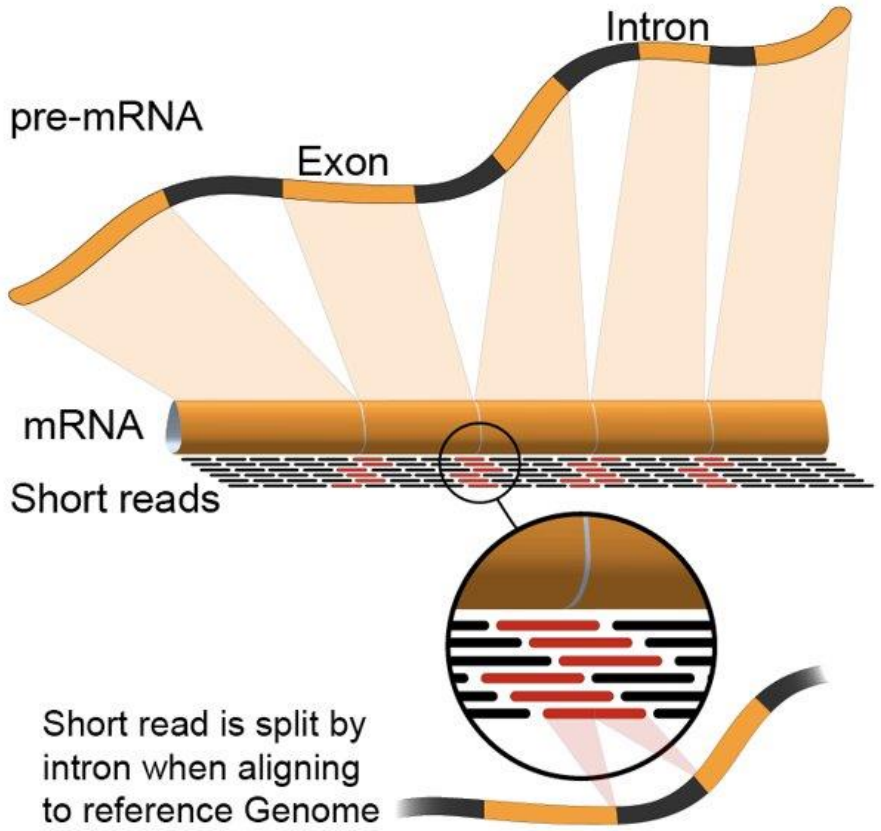
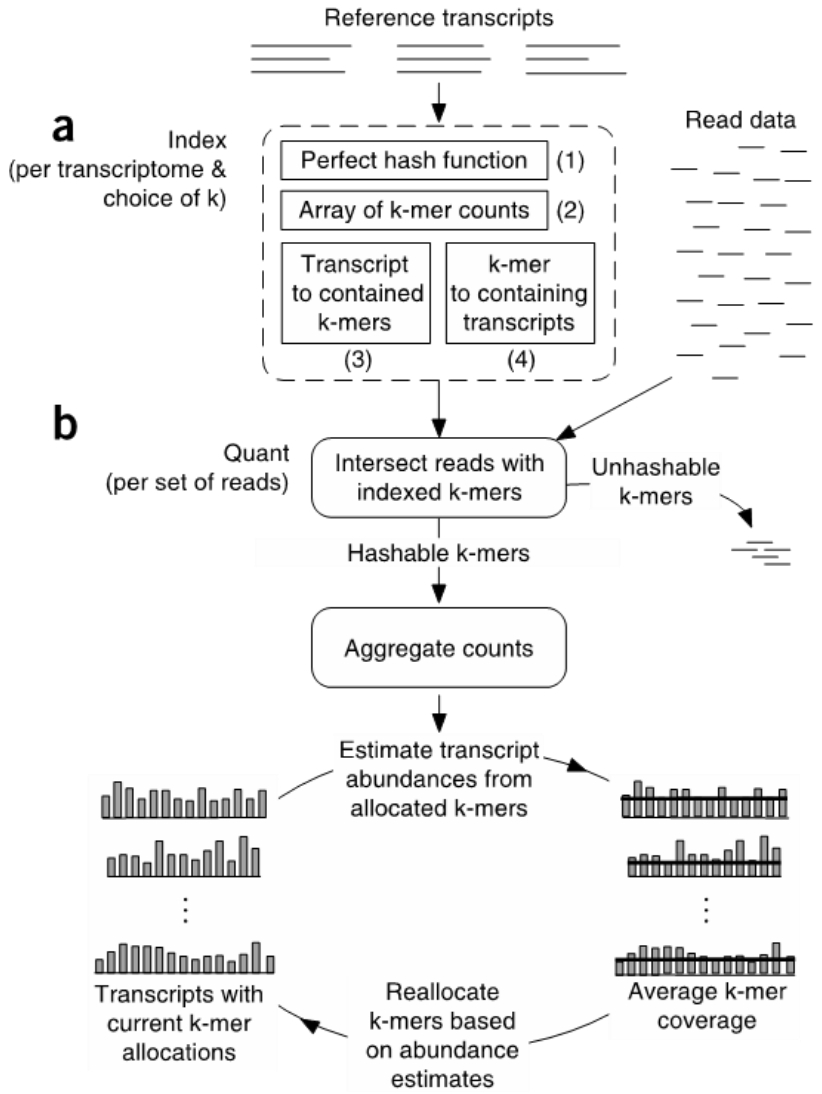


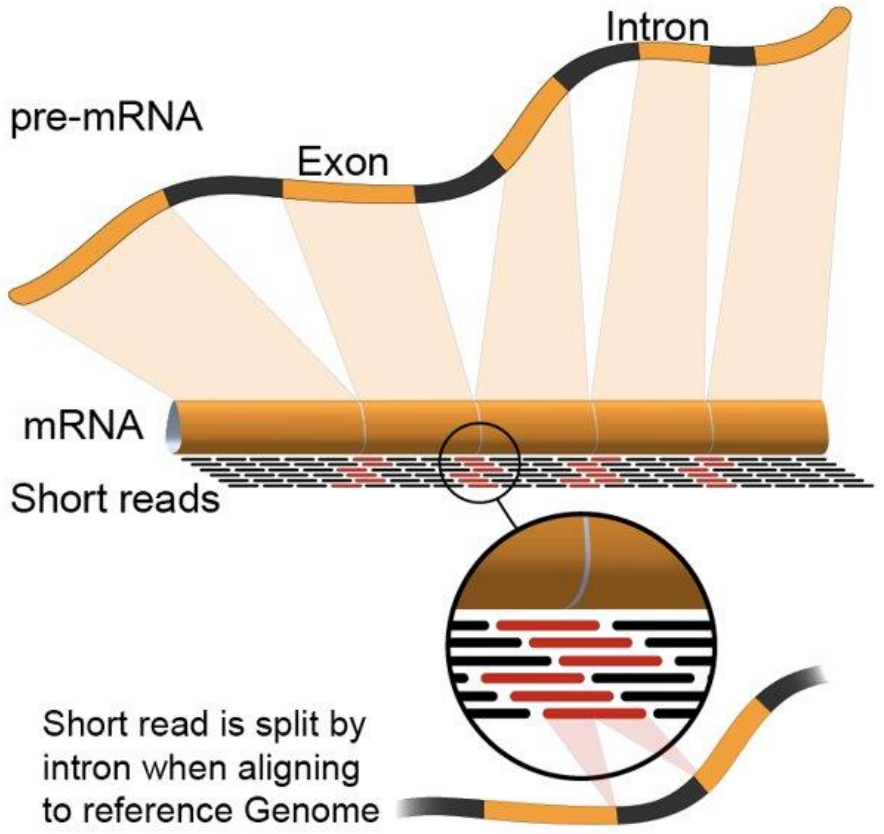
Image credit : wikipedia user Rgocs



Sailfish (Patro et al. 2014),

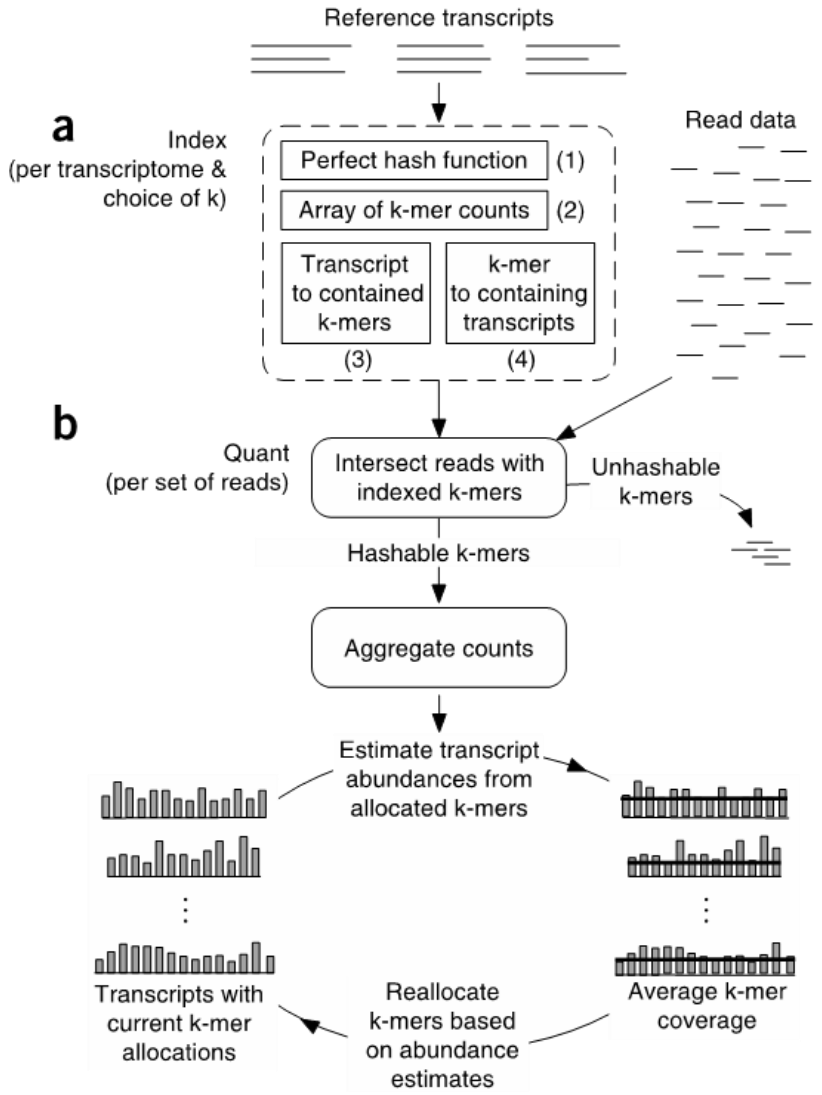
See also Kallisto (Bray et al. 2016)

Alignment vs. pseudoalignment



Resource intensive!

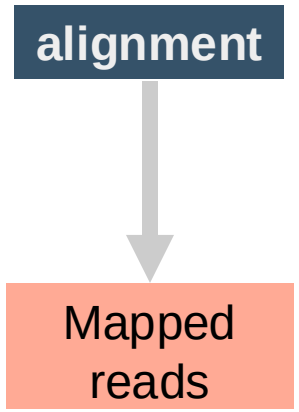
Image credit : wikipedia user Rgocs

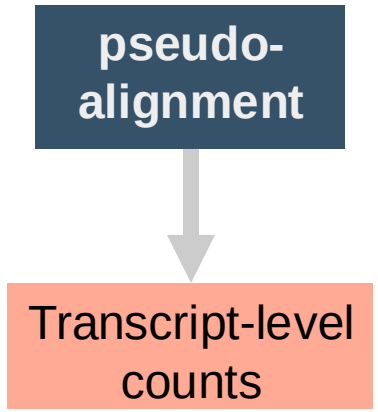


Sailfish (Patro et al. 2014),

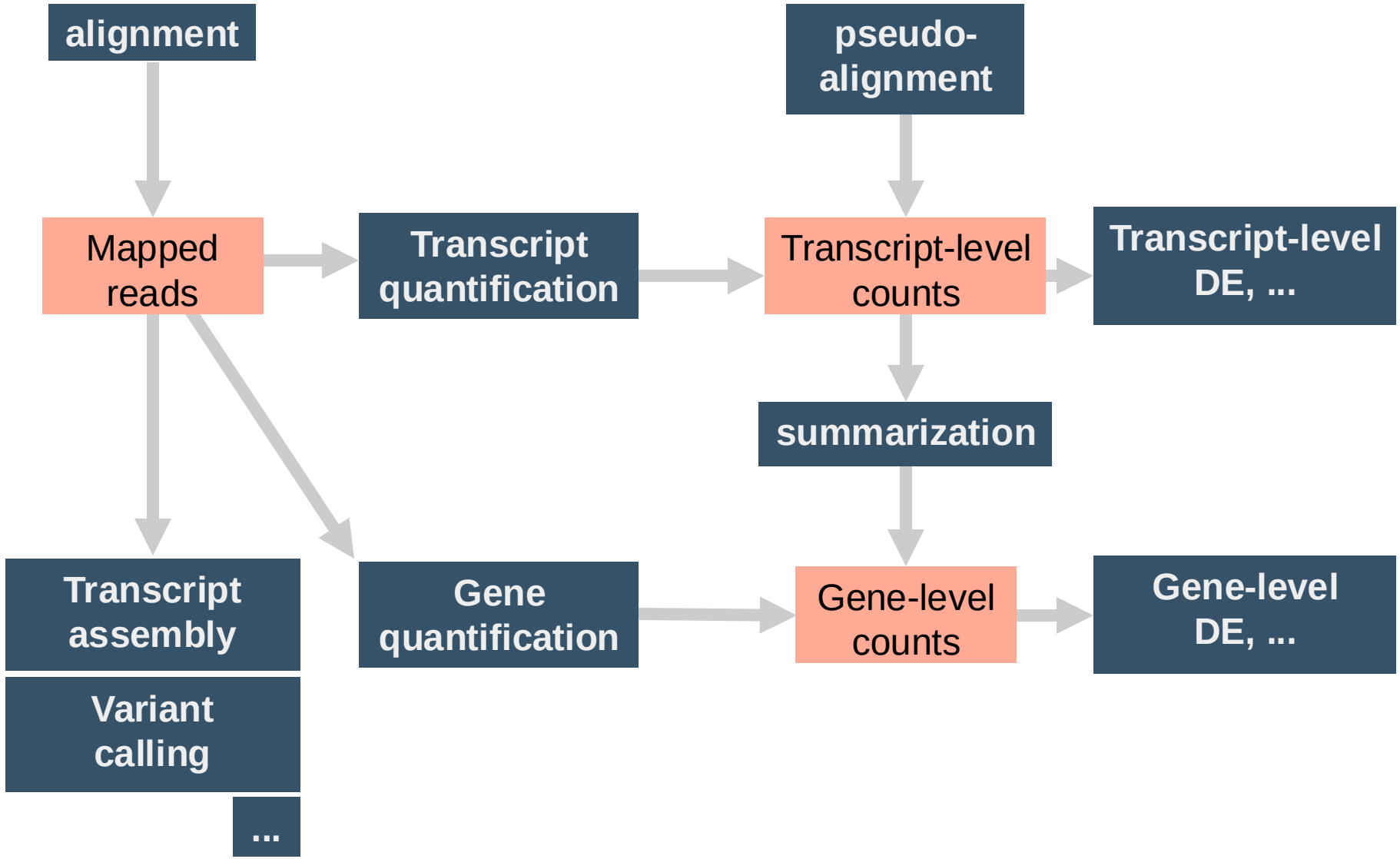
See also Kallisto (Bray et al. 2016)

Alignment vs. pseudoalignment

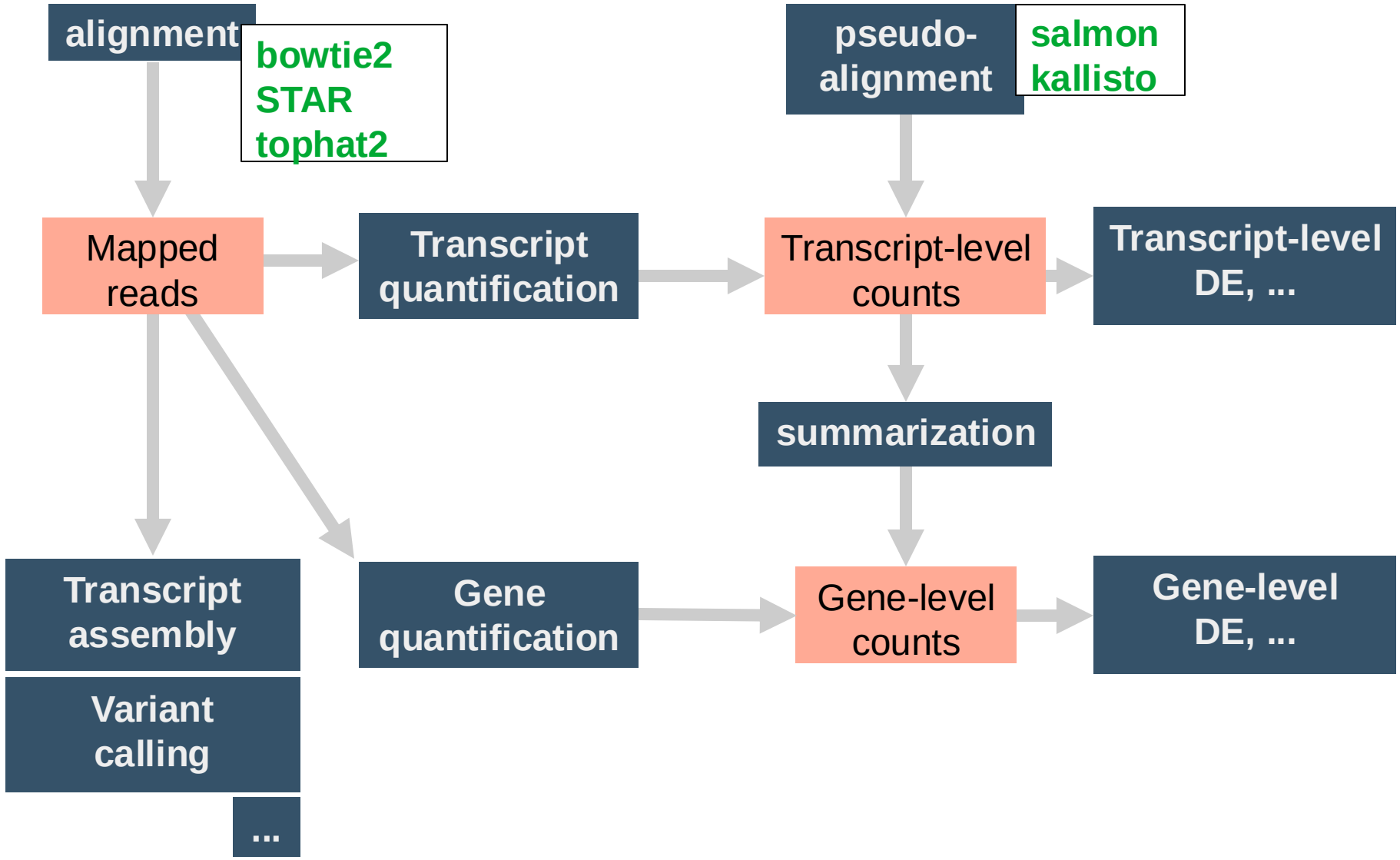




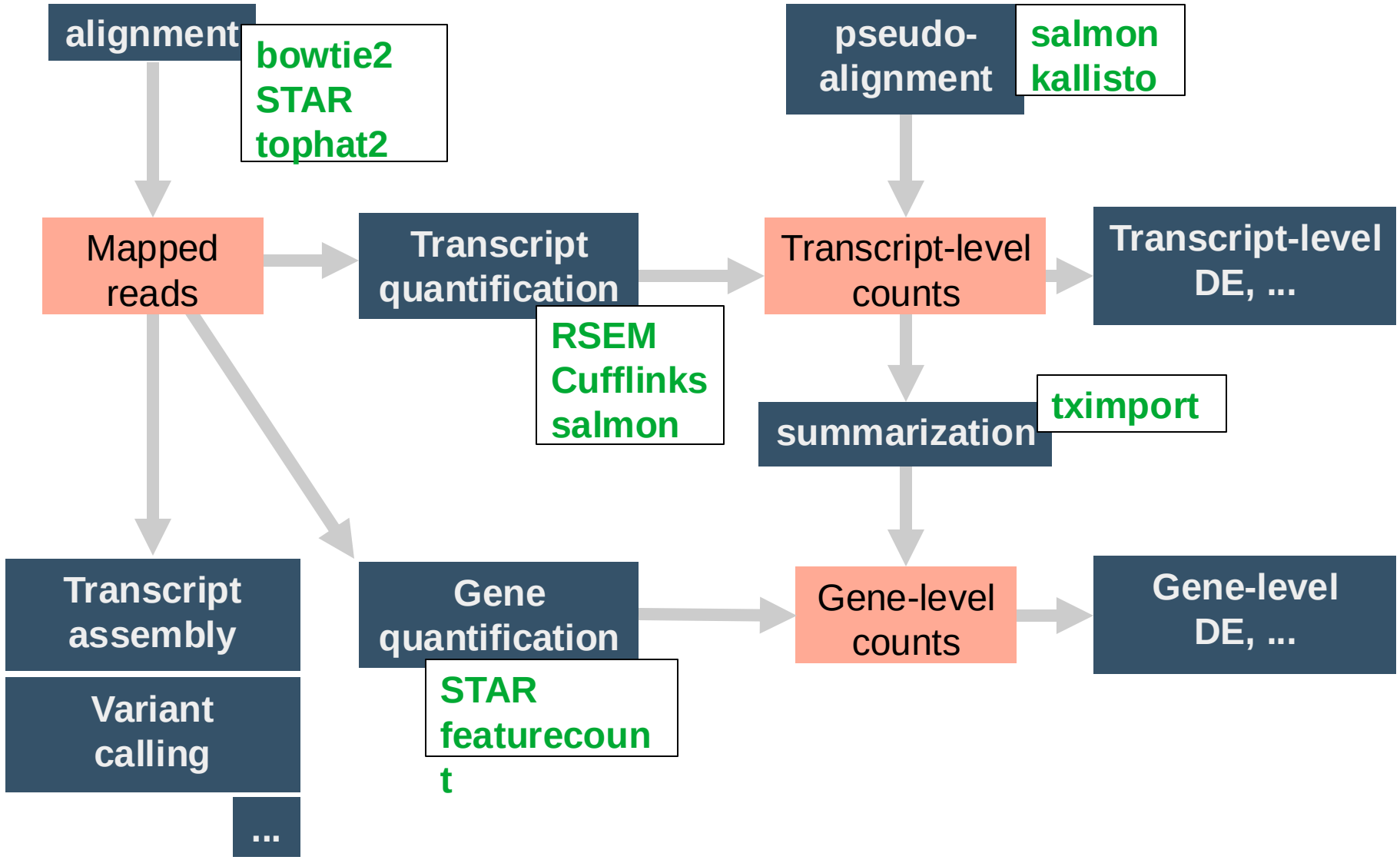
Alignment vs. pseudoalignment



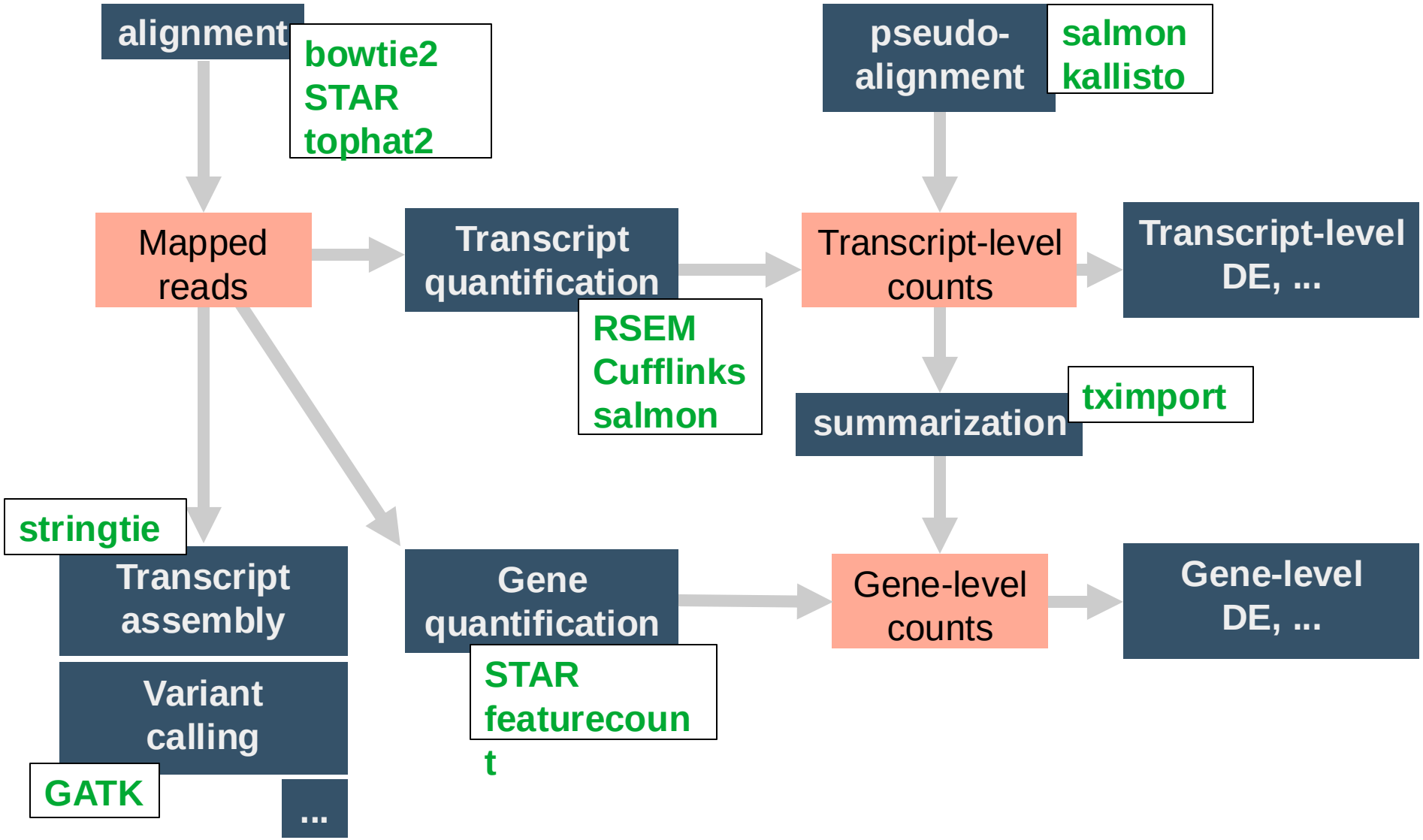
Alignment vs. pseudoalignment



Alignment vs. pseudoalignment

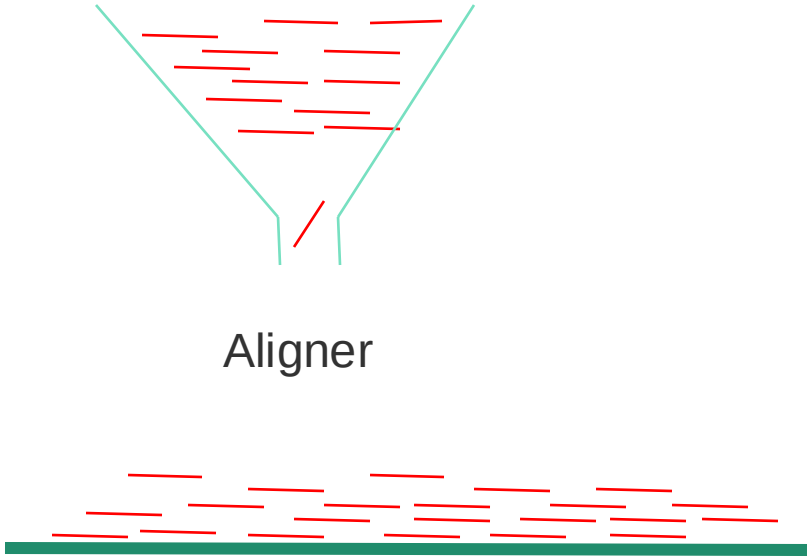


Alignment vs. pseudoalignment



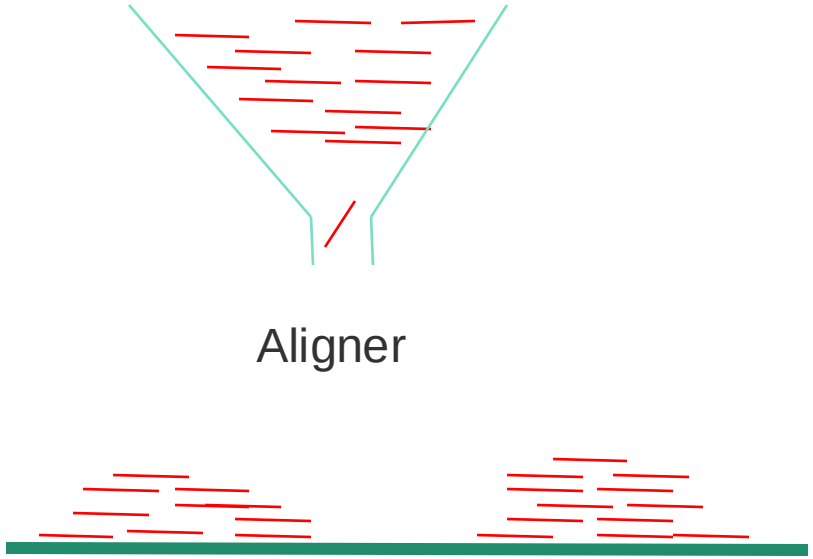
Aligning & mapping sequencing reads

Whole genome re-sequencing
Prokaryote RNAseq



- BWA (Li and Durbin 2009)
- Bowtie (Langemead et al. 2009)

Transcriptome sequencing (RNA-seq)



- Tophat (Trapenell et al. 2009, Kim et al. 2013)
- STAR (Dobin et al. 2013)

Alignment using STAR

Phase 1:

mapping using "Maximum Mappable Prefix"

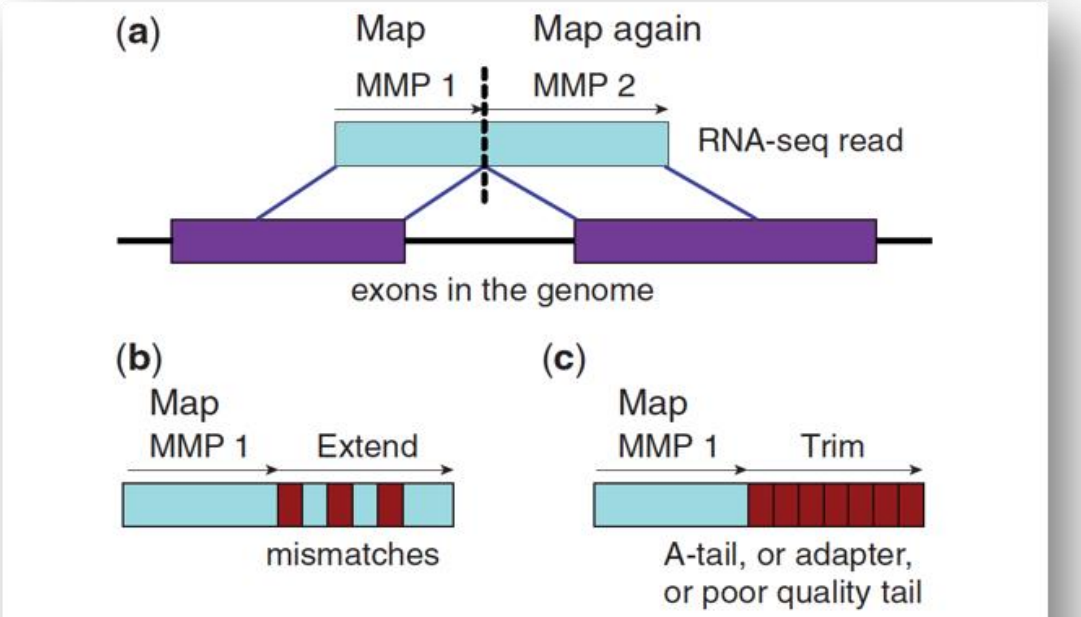


Fig. 1. Schematic representation of the Maximum Mappable Prefix search in the STAR algorithm for detecting (**a**) splice junctions, (**b**) mismatches and (**c**) tails

Phase 2:

"stitching"

Dobin et al. 2013

Benchmarking of RNAseq aligners

	Correctly mapped 200 bases	>=150 bases correctly mapped	Unmapped	True positive junctions		False positive junctions	
				Number	Sensitivity	Number	FDR
Aligner	1	2	4	5	6	7	8
STAR	81.3%	95.0%	4.82%	148,487	92.7%	409	0.3%
TopHat2	82.6%	83.7%	6.70%	135,006	84.3%	1,228	0.9%

Dobin & Gingeras 2013

STAR : 20x faster

Tophat2 : 6x less memory (can be run on recent laptop)

Benchmarking of RNAseq aligners

	Correctly mapped 200 bases	>=150 bases correctly mapped	Unmapped	True positive junctions		False positive junctions	
				Number	Sensitivity	Number	FDR
Aligner	1	2	4	5	6	7	8
STAR	81.3%	95.0%	4.82%	148,487	92.7%	409	0.3%
TopHat2	82.6%	83.7%	6.70%	135,006	84.3%	1,228	0.9%

Dobin & Gingeras 2013

STAR : 20x faster

Tophat2 : 6x less memory (can be run on recent laptop)

Reference index preparation

Different for each software!

Needs a suitable reference genome

- sequence
- annotation

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

https://hgdownload.soe.ucsc.edu/downloads.html

Genome annotation files

Text file describing genomic features

- Gene, CDS, exon, intron, …
- Chromosome, start, end, strand, attributes, …

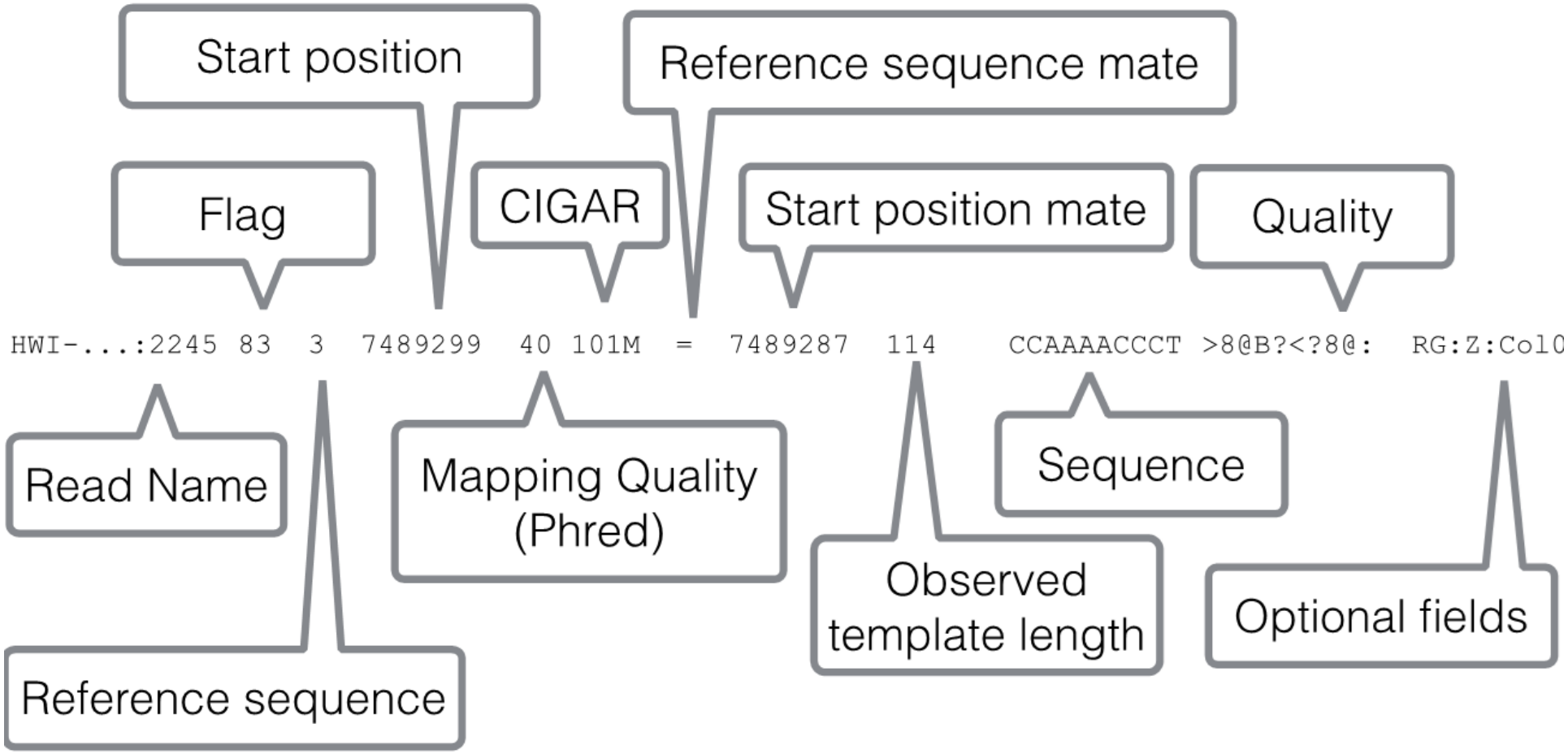
Most common format: gtf / gff3
http://www.ensembl.org/info/website/upload/gff.html

MT	RefSeq	gene	2751	3707	.	+	.	gene_id "ENSMUSG00000064341";
MT	RefSeq	transcript	2751	3707	.	+	.	gene_id "ENSMUSG00000064341";
MT	RefSeq	exon	2751	3707	.	+	.	gene_id "ENSMUSG00000064341";
MT	RefSeq	CDS	2751	3704	.	+	0	gene_id "ENSMUSG00000064341";
MT	RefSeq	start_codon	2751	2753	.	+	0	gene_id "ENSMUSG00000064341";
MT	RefSeq	stop_codon	3705	3707	.	+	0	gene_id "ENSMUSG00000064341";

Most mapper produce SAM files

https://samtools.github.io/hts-specs/SAMv1.pdf

Each line contain mapping information about a single read



Most mapper produce SAM files

https://samtools.github.io/hts-specs/SAMv1.pdf

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

Example, flag 83 = 64+16+2+1 means it's first read (0x40) of pair-end reads (0x1) and it's mapped on minus strand (0x10) and both reads mapped (0x2).
https://broadinstitute.github.io/picard/explain-flags.html

Most mapper produce SAM files

https://samtools.github.io/hts-specs/SAMv1.pdf

- Big files: ideally compress in BAM file
- Can be sorted and indexed for easy access by post-processing software

- multiQC can grab interesting information from a folder containing SAM/BAM files
  (as well as the other files created by the mapping software)

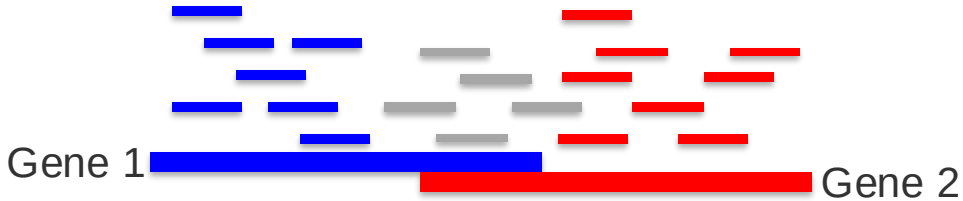
After mapping: counting

Pseudoaligner: transcript-level expression quantification

Aligner: we need to subsequently estimate expression from mapped reads

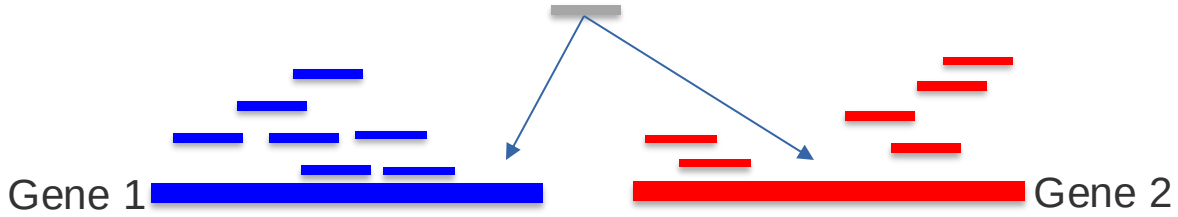
Counting: fundamental problems

Overlapping genes



stranded sequencing
OR
discard reads / count both ?

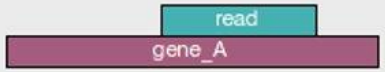
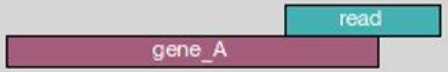
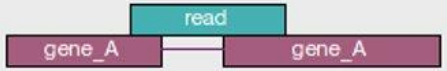
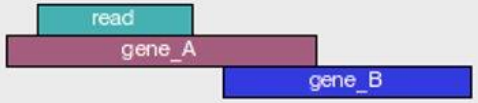
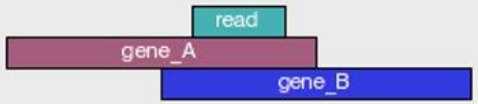
Multi-mapping reads



- discard reads?
- count both?

Counting: gene-level counters

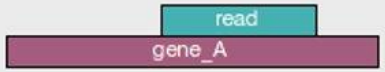
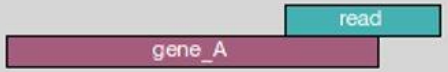
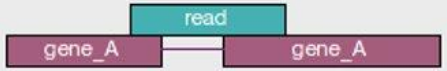
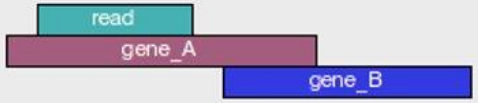
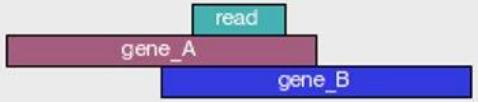
HTSeq

	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous	gene_A	gene_A
	ambiguous	ambiguous	ambiguous

Counting: gene-level counters

HTSeq

FeatureCount
+ options for
fractional counts

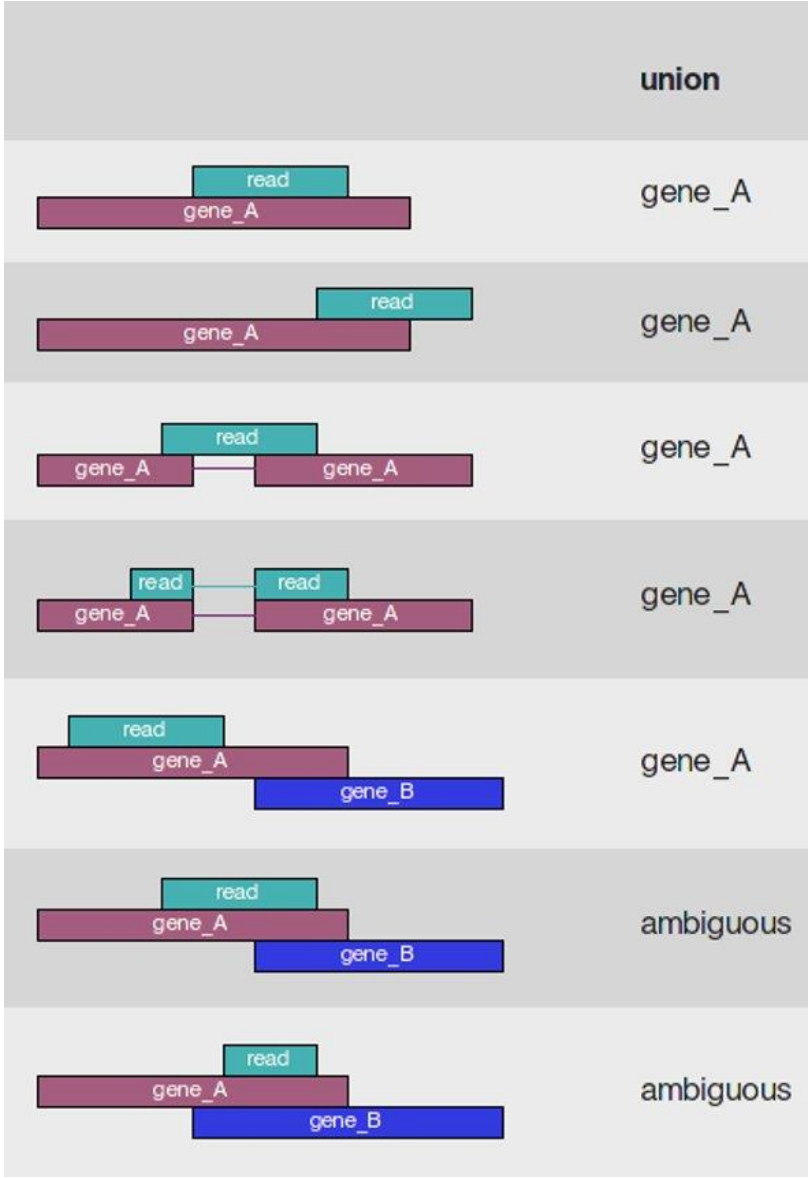
	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous	gene_A	gene_A
	ambiguous	ambiguous	ambiguous

Counting: gene-level counters

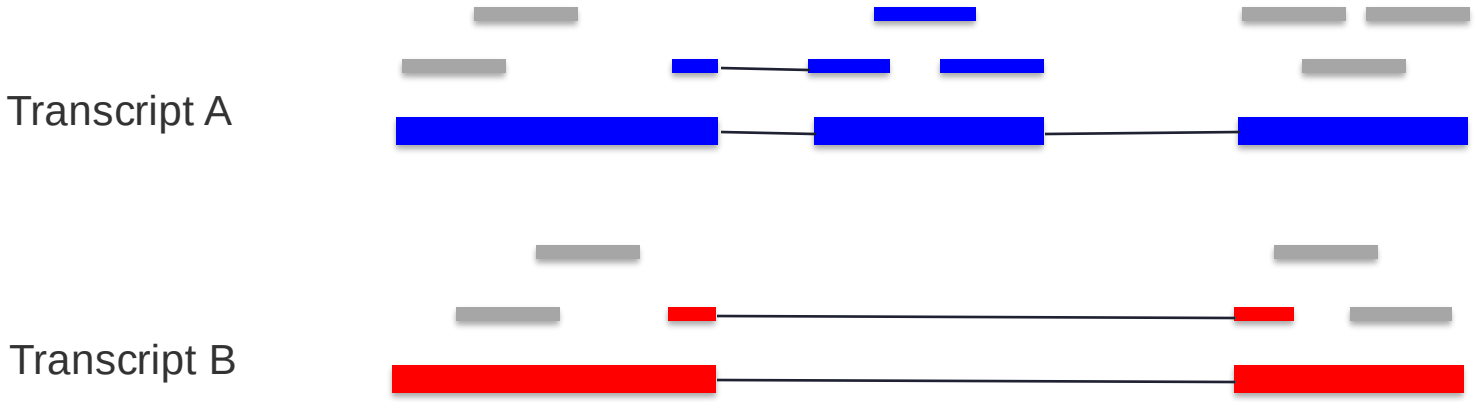
HTSeq

FeatureCount
+ options for
fractional counts

STAR



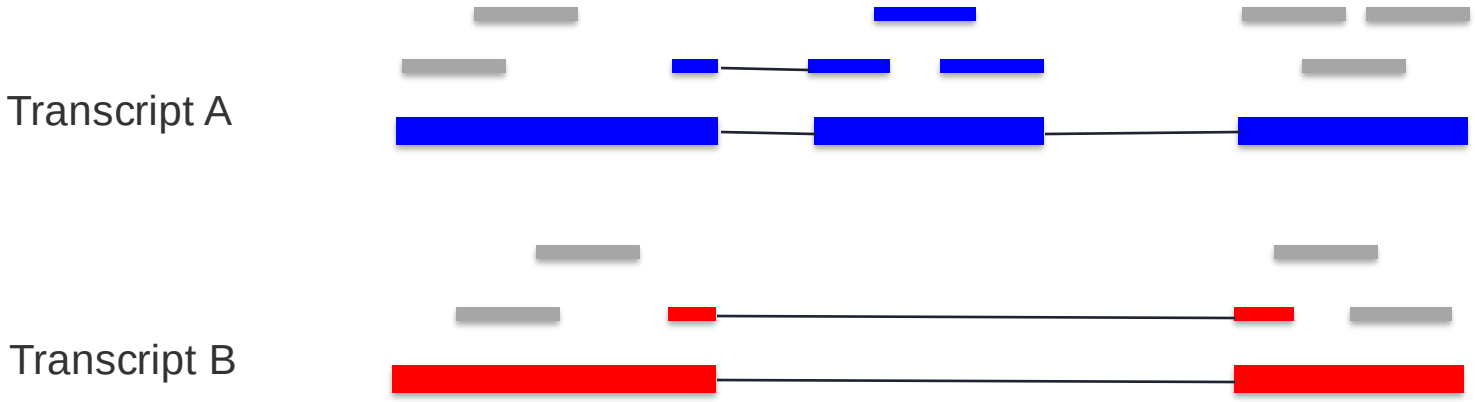
Counting: transcript-level counter



RSEM, cufflink, salmon, stringtie, …

Practical

Counting: transcript-level counter



RSEM, cufflink, salmon, stringtie, …