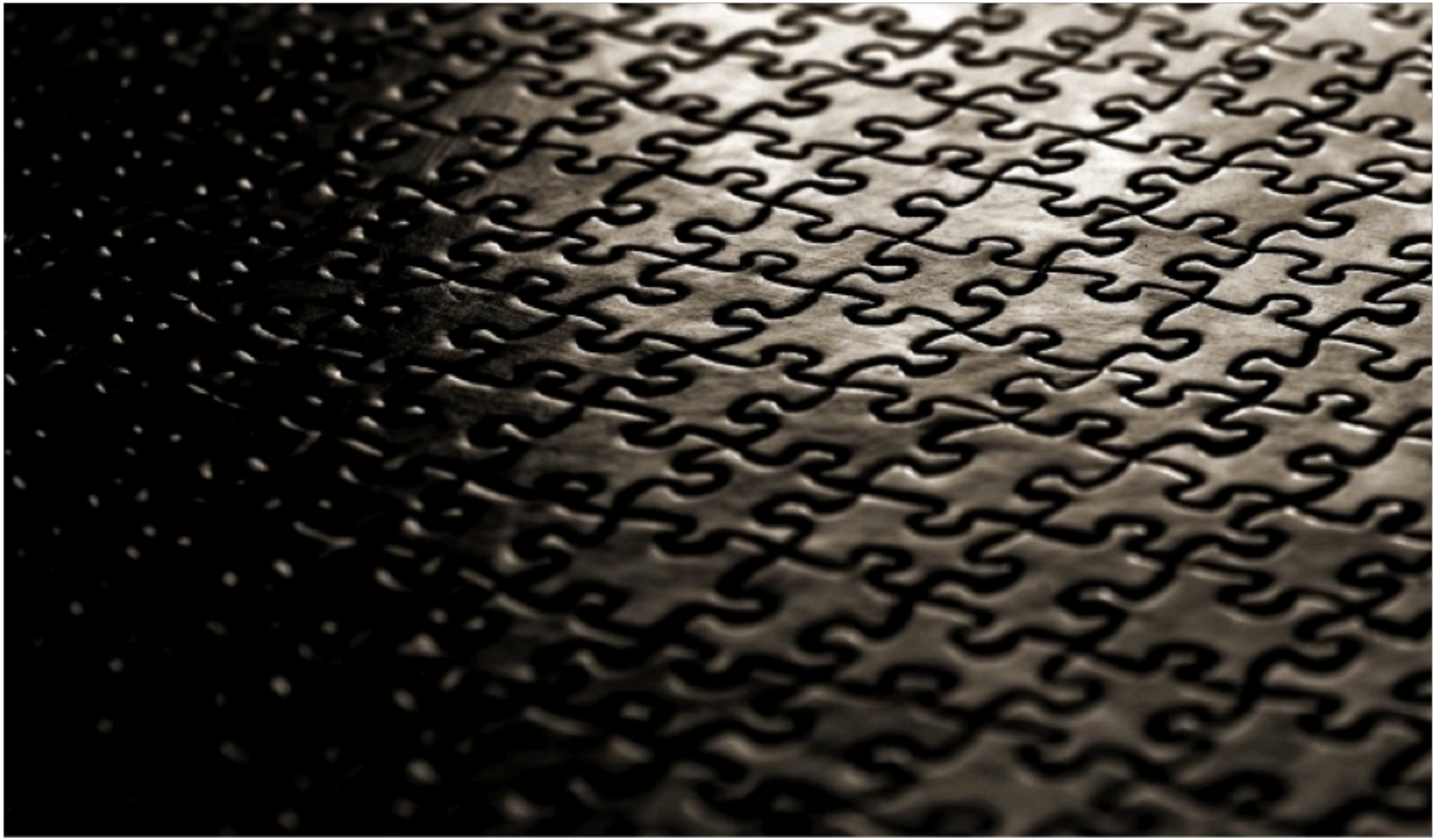
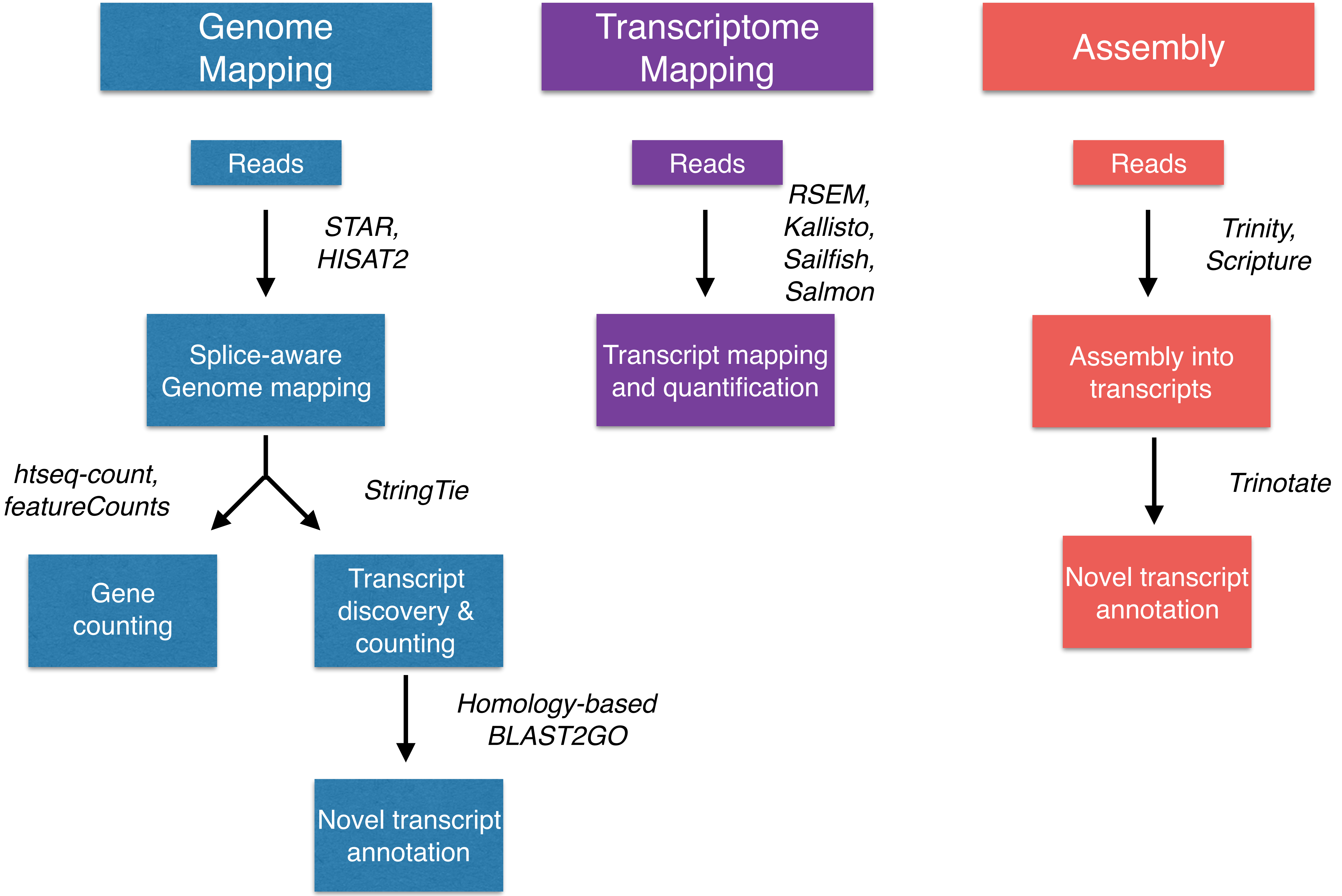


Aligning reads: tools and theory



Genome Mapping

Reads

STAR, HISAT2

Splice-aware Genome mapping

htseq-count, featureCounts

StringTie

Gene counting

Transcript discovery & counting

Homology-based BLAST2GO

Novel transcript annotation

Transcriptome Mapping

Reads

RSEM, Kallisto, Sailfish, Salmon

Transcript mapping and quantification

Assembly

Reads

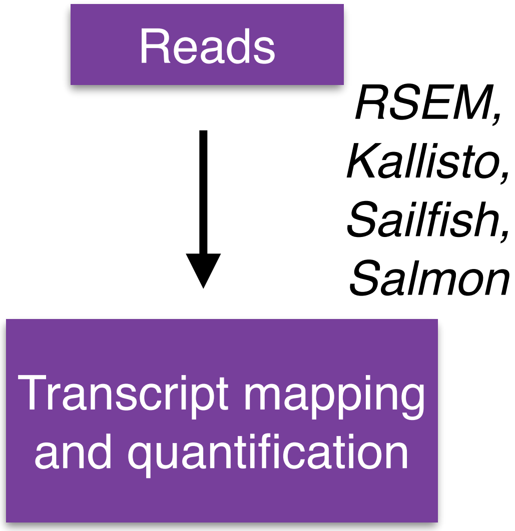
Trinity, Scripture

Assembly into transcripts

Trinotate

Novel transcript annotation

Transcriptome Mapping

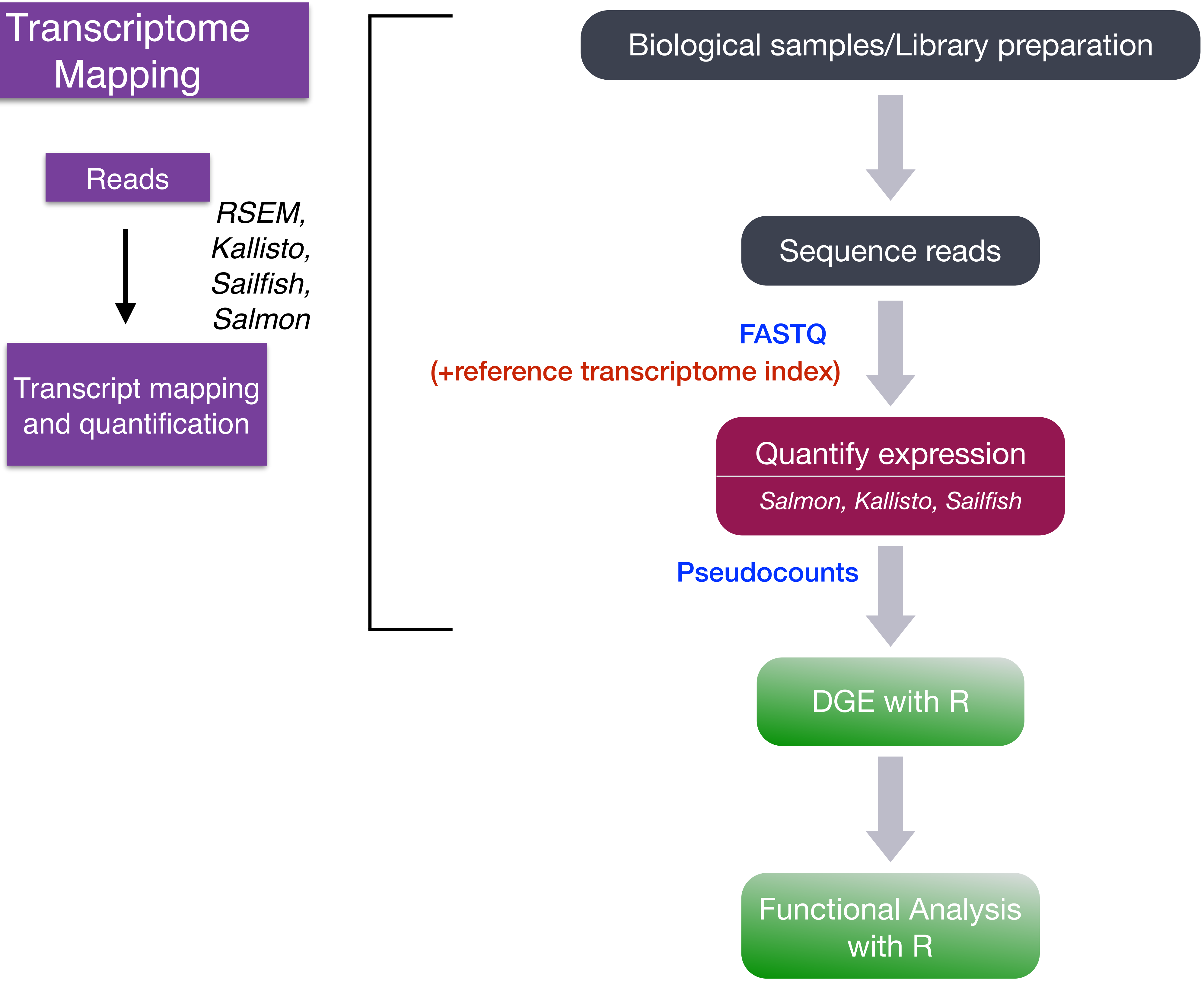


Reads

RSEM,
Kallisto,
Sailfish,
Salmon

Transcript mapping and quantification

Transcriptome Mapping



Reads

RSEM, Kallisto, Sailfish, Salmon

Transcript mapping and quantification

Biological samples/Library preparation

Sequence reads

FASTQ

(+reference transcriptome index)

Quantify expression

Salmon, Kallisto, Sailfish

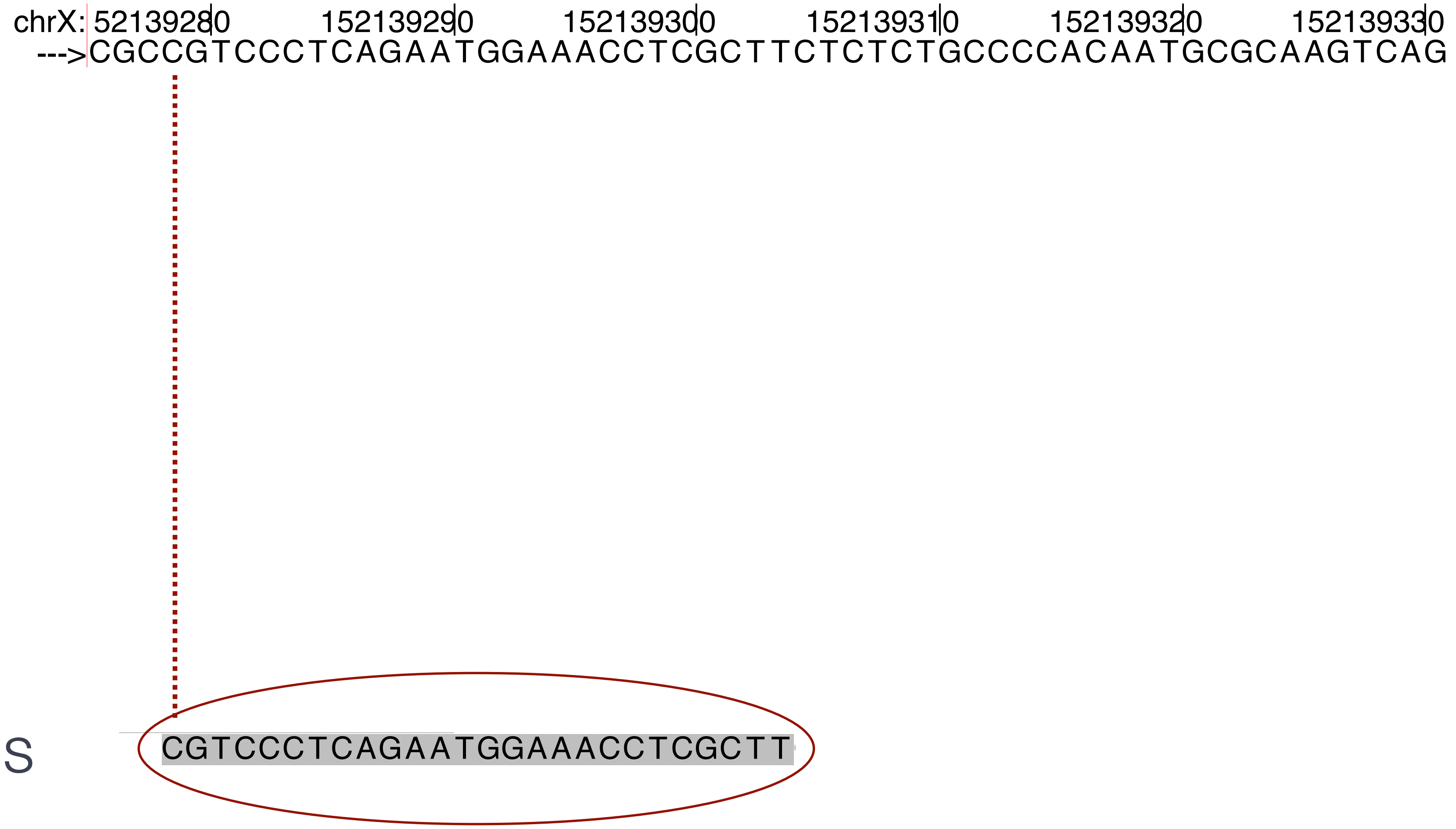
Pseudocounts

DGE with R

Functional Analysis with R

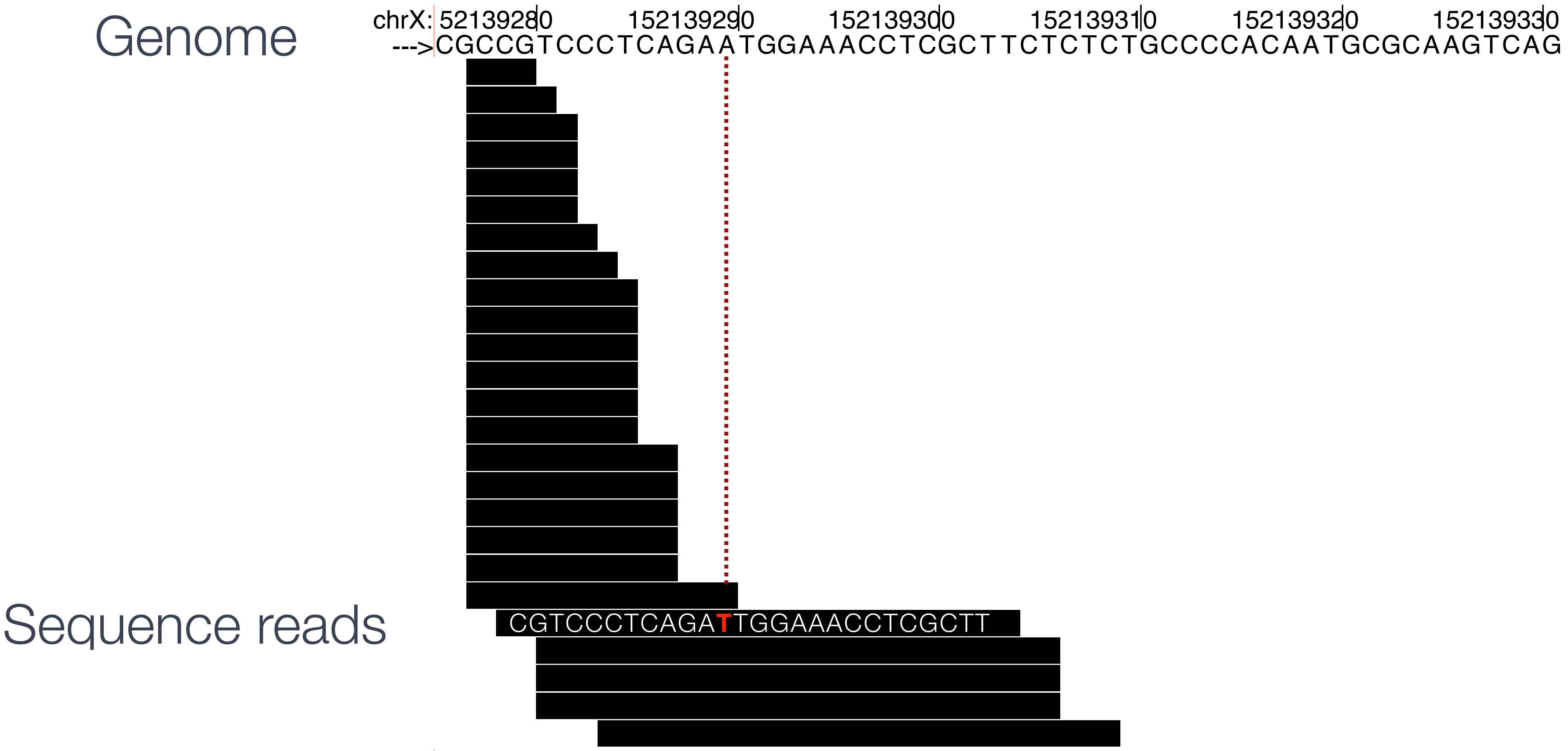
Goal: Finding where in the genome these reads originated from

Reference


chrX: 52139280
152139290
152139300
152139310
152139320
152139330
--->CGCCGTCCCTCAGAATGGAAACCTCGCTTCTCTCTGCCCCACAATGCGCAAGTCAG
CGTCCCTCAGAATGGAAACCTCGCTT

Sequence reads

A simple case of string matching


Genome
chrX: 52139280
152139290
152139300
152139310
152139320
152139330
--->CGCCGTCCCTCAGAATGGAAACCTCGCTTCTCTCTGCCCCACAATGCGCAAGTCAG
Sequence reads
CGTCCCTCAGATTGGAAACCTCGCTT

A simple case of string matching?

Non-comprehensive list of challenges

- Large, incomplete and repetitive genomes OR transcriptomes with overlapping transcripts (isoforms)
- Short reads: 50-150 bp
    - Non-unique alignment
    - Sensitive to non-exact matching (variants, sequencing errors)
- Massive number of short reads
- Small insert size: 200-500 bp libraries
- Compute capacity for efficient mapping

Building an index

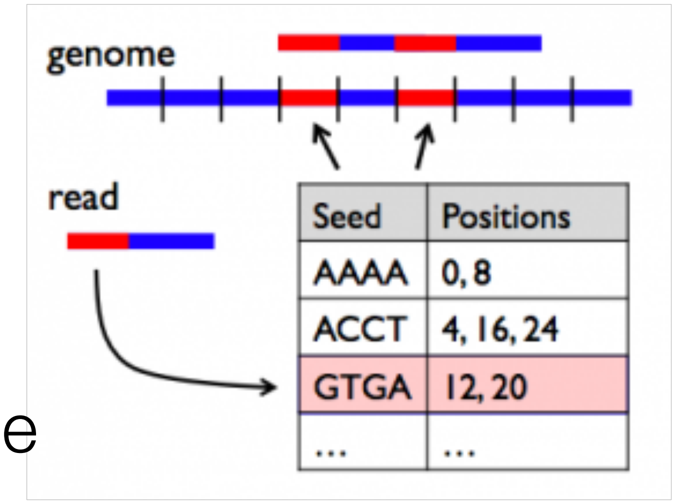
- Having an index of the reference sequence provides an efficient way to search
- Once index is built, it can be queried any number of times
- Every genome or transcriptome build requires a new index for the specific tool in question.

Commonly used indexing methods

- Hash-based (Salmon, Kallisto)
- Suffix arrays (Salmon, STAR)
- Burrows-Wheeler Transform (BWA, Bowtie2)

Hash-based alignment (circa 1990)

- Pick k-mer size, build lookup of every k-mer in the reference mapped to its positions (the index)
- Break the query into k-mers
- Seed-and-extend strategy
- For BLAST, 100% match the query k-mer to reference then extend until score drops below 50%
- 0.1 - 1 sec per query; not feasible for NGS data



Seed	Positions
AAAA	0, 8
ACCT	4, 16, 24
GTGA	12, 20
...	...

Hash-based alignment (present day)

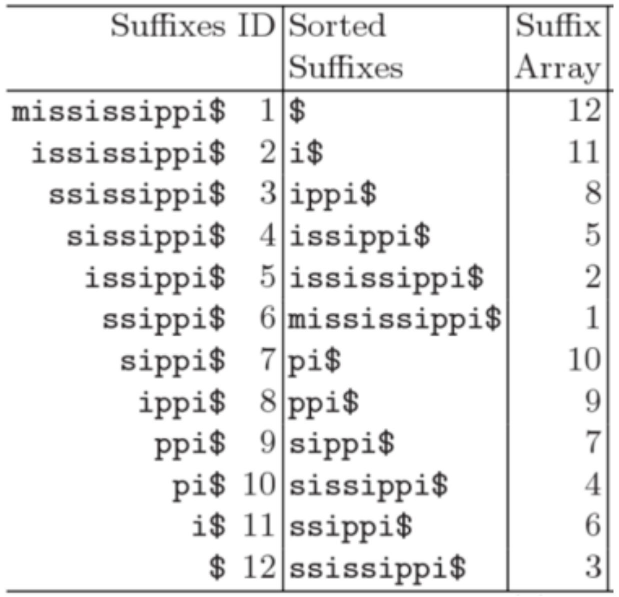
- Need to make some concessions on sensitivity by making adaptations for use on NGS data:
    - allow for mismatches and/or gaps (ELAND, MAQ, SOAP)
    - using multiple seeds (BLAT, ELAND2)
- Memory intensive and slower (~16GB RAM required for hg19)
- Simpler in design but more sensitive

Suffix arrays

- A sorted table of all suffixes (substrings) of a given string
- A suffix array will contain integers that represent the starting indexes of the all the suffixes of a given string, after the aforementioned suffixes are sorted
- Requires large amount of memory to load the suffix array and genome sequence prior to alignment
- Popular Tools:

  STAR (2012), Salmon

Let the given string be "`mississippi`"

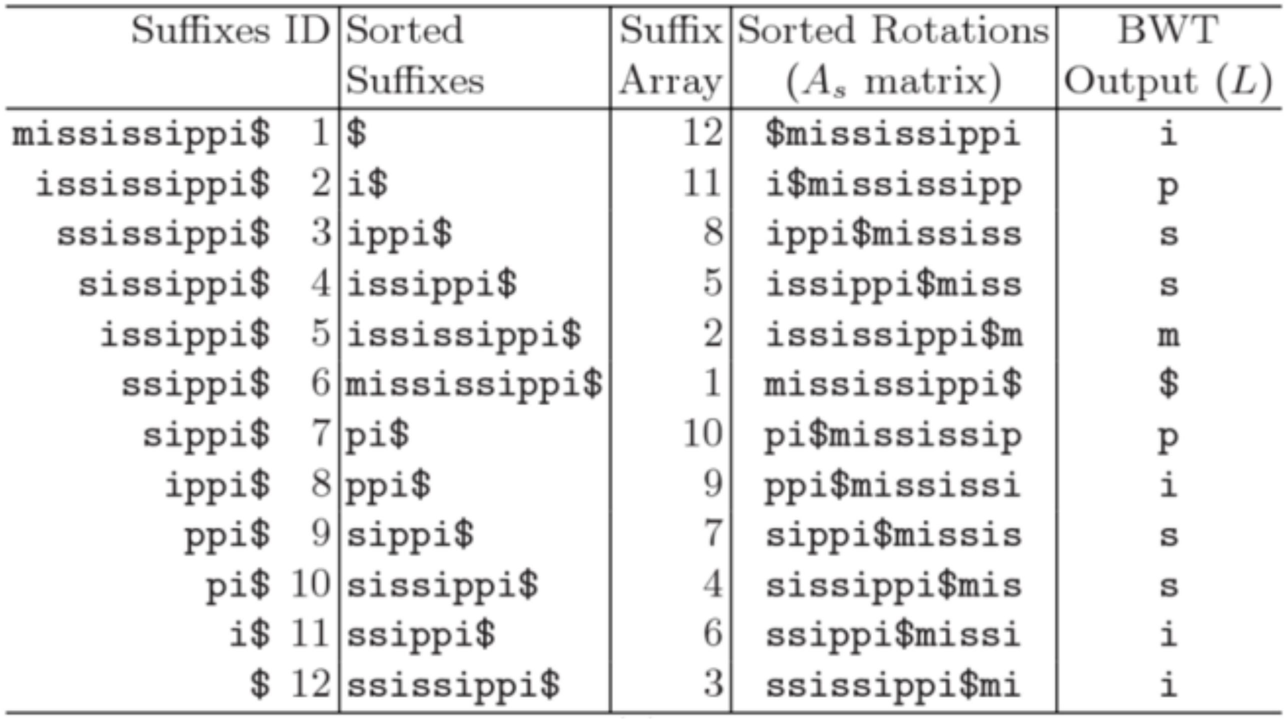


Suffixes	ID	Sorted Suffixes	Suffix Array
`mississippi$`	1	`$`	12
`ississippi$`	2	`i$`	11
`ssissippi$`	3	`ippi$`	8
`sissippi$`	4	`issippi$`	5
`issippi$`	5	`ississippi$`	2
`ssippi$`	6	`mississippi$`	1
`sippi$`	7	`pi$`	10
`ippi$`	8	`ppi$`	9
`ppi$`	9	`sippi$`	7
`pi$`	10	`sissippi$`	4
`i$`	11	`ssippi$`	6
`$`	12	`ssissippi$`	3

The suffix array will be:
{12, 11, 8, 5, 2, 1, 10, 9, 7, 4, 6, 3}

Burrows-Wheeler transform

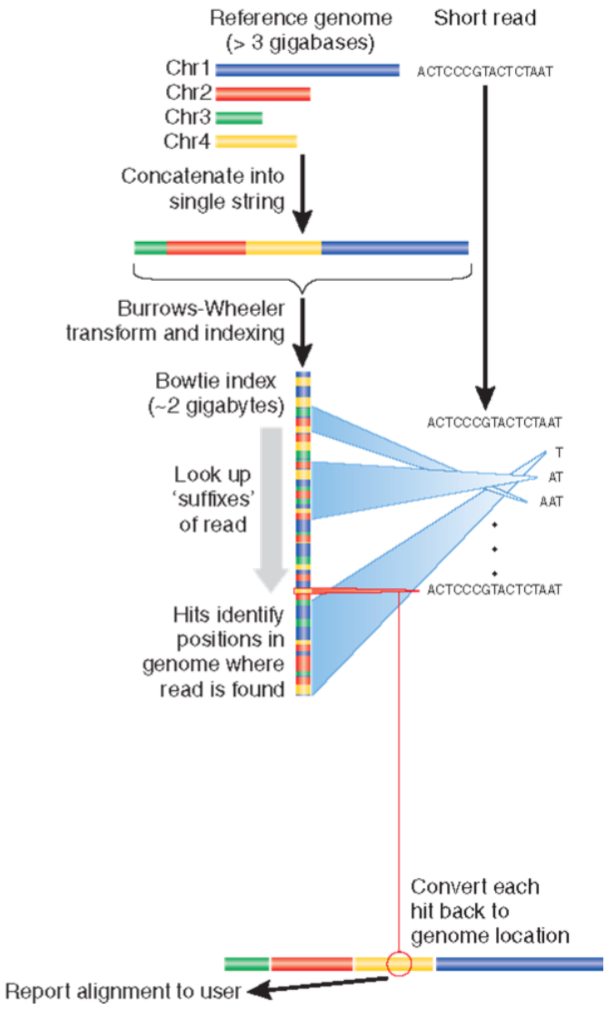
- A compressed form of suffix arrays
- Tends to put runs of the same character together rather than alphabetically, which makes the compression work well



Suffixes	ID	Sorted Suffixes	Suffix Array	Sorted Rotations (A_s matrix)	BWT Output (L)
mississippi$	1	$	12	$mississippi	i
ississippi$	2	i$	11	i$mississipp	p
ssissippi$	3	ippi$	8	ippi$mississ	s
sissippi$	4	issippi$	5	issippi$miss	s
issippi$	5	ississippi$	2	ississippi$m	m
ssippi$	6	mississippi$	1	mississippi$	$
sippi$	7	pi$	10	pi$mississip	p
ippi$	8	ppi$	9	ppi$mississi	i
ppi$	9	sippi$	7	sippi$missis	s
pi$	10	sissippi$	4	sissippi$mis	s
i$	11	ssippi$	6	ssippi$missi	i
$	12	ssissippi$	3	ssissippi$mi	i

Burrows-Wheeler transform

- Much less memory because of compression; ~1.5 GB of RAM required for hg19 index
- But compression results in diminished efficiency of the string search operations
- Popular Tools:
  - Bowtie2 (2012)
  - SOAP2
  - BWA-MEM (2013)


Reference genome
(> 3 gigabases)
Short read
Chr1
Chr2
Chr3
Chr4
ACTCCCGTACTCTAAT
Concatenate into single string
Burrows-Wheeler transform and indexing
Bowtie index
(~2 gigabytes)
ACTCCCGTACTCTAAT
T
AT
AAT
Look up 'suffixes' of read
ACTCCCGTACTCTAAT
Hits identify positions in genome where read is found
Convert each hit back to genome location
Report alignment to user

Reference data versions matter

- Ensembl, UCSC and NCBI all often use the same genome assemblies or builds (e.g. GrCh38 == hg38)
- Make sure that the annotation file (GTF) is exactly matched with the genome file (fasta), or transcriptome file (fasta)
  - Same build version
  - Same source (e.g. both from FlyBase)

How does Salmon map reads?

Lightweight alignment and quantification using Salmon

- **Reference:** FASTA file of all transcript sequences for the organism
- **Index:** (2 components)
  - Suffix array
  - Hash table (mapping each transcript to its location in the SA)
- **Output:** abundance estimates
  - Number of reads mapping to each transcript listed in the reference

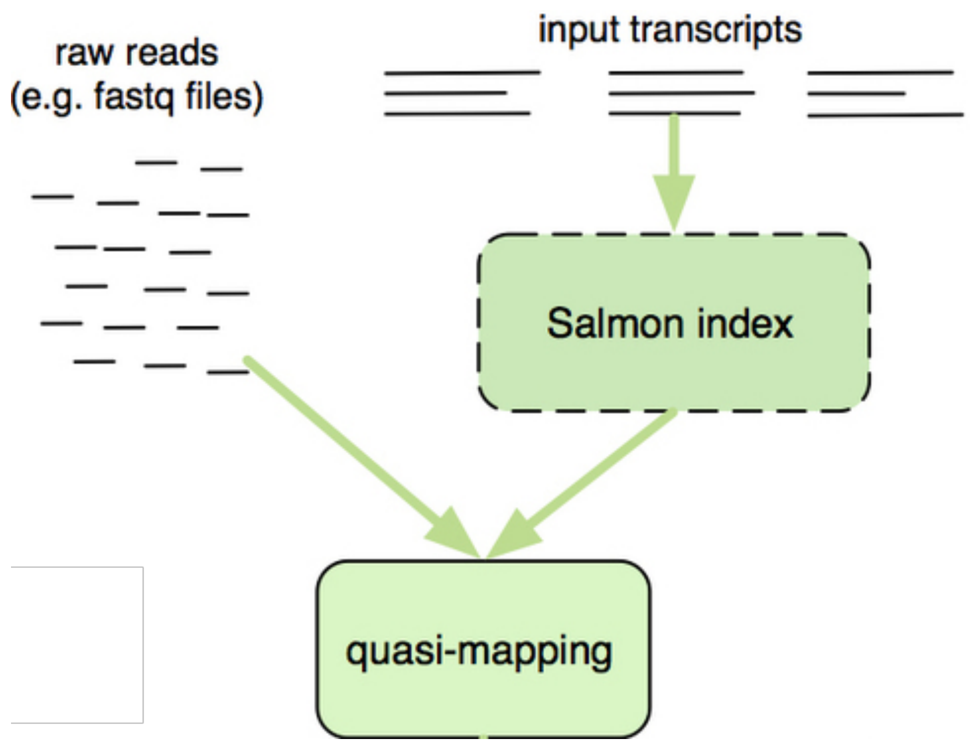

raw reads
(e.g. fastq files)
input transcripts
Salmon index
quasi-mapping

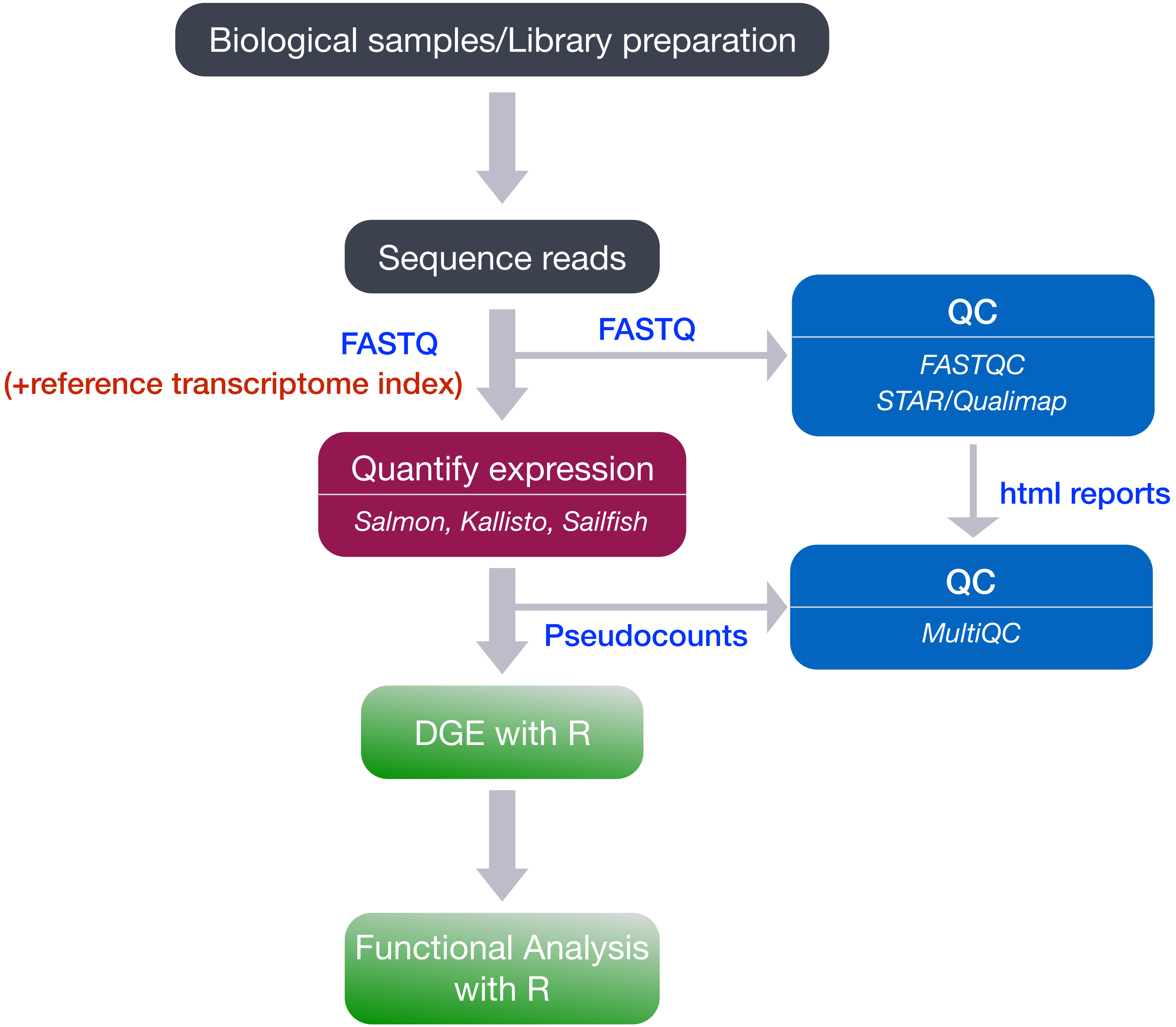
Image source: RNA-seq blog

Note that we don't get any information on *where* each read is mapping!

Why use lightweight alignment?

- Approaches avoid base-to-base alignment
- Faster, more efficient (~ >20x faster than alignment-based)
- Improved accuracy for transcript-level quantification
- Improvements in accuracy for gene-level quantification**
- Other tools include: Kallisto (quasi-aligner), Sailfish (kmer-based), Salmon (quasi-aligner), RSEM

**doi: 10.12688/f1000research.7563.2


Biological samples/Library preparation

Sequence reads

FASTQ

(+reference transcriptome index)

FASTQ

QC

FASTQC
STAR/Qualimap

Quantify expression

Salmon, Kallisto, Sailfish

html reports

Pseudocounts

QC

MultiQC

DGE with R

Functional Analysis
with R

Quality Checks

QC metrics

Various metrics can give important information about the quality of the library:

-- Total % of reads aligning? % of uniquely mapping reads? % of properly paired PE reads?

-- Genomic origin of reads (exonic, intronic, intergenic)

-- Quantity of rRNA

-- Transcript coverage and 5'-3' bias

How do we compute QC metrics?

- Tools like RNA-SeQC and Qualimap
    - *Input:* alignment file formats (i.e. SAM/BAM)
    - *Output:* summary of the different metrics in an HTML report format

SAM/BAM file format

- Sequence Alignment Map (SAM) format contains information on a per-read basis:

  -- Coordinates of alignment, including strand

  -- Mismatches

  -- Mapping information (unique?, properly paired?, etc.)

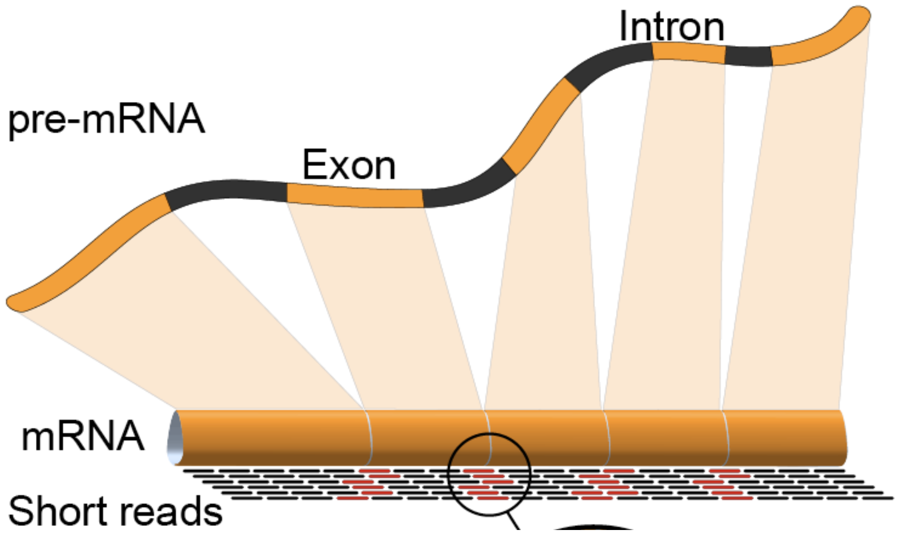
  -- Quality of mapping (tool-specific scoring systems)

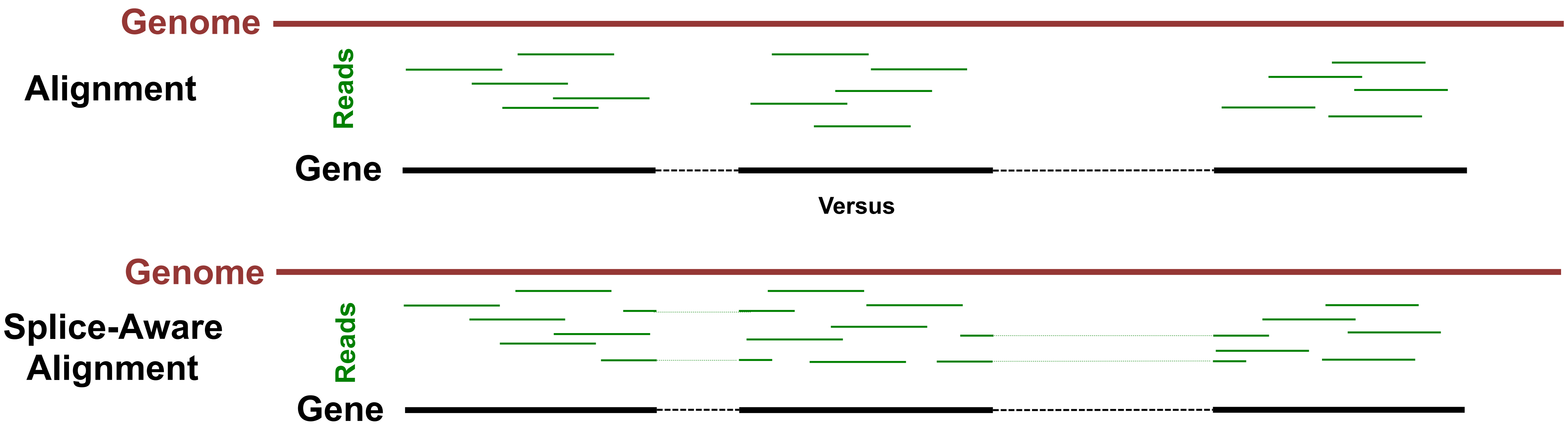
- BAM: Binary version of SAM alignment format files

More information about SAM/BAM

Where do we get this SAM/BAM file from?

- Need to ***align reads to the genome***
- Genome alignment tools perform base-to-base alignment of reads to the genome sequence (FASTA) and can be guided with a gene annotation file (GTF)
- Genome alignment outputs a SAM/BAM file


Intron
pre-mRNA
Exon
mRNA
Short reads


Genome
Alignment
Reads
Gene
Versus
Genome
Splice-Aware Alignment
Reads
Gene

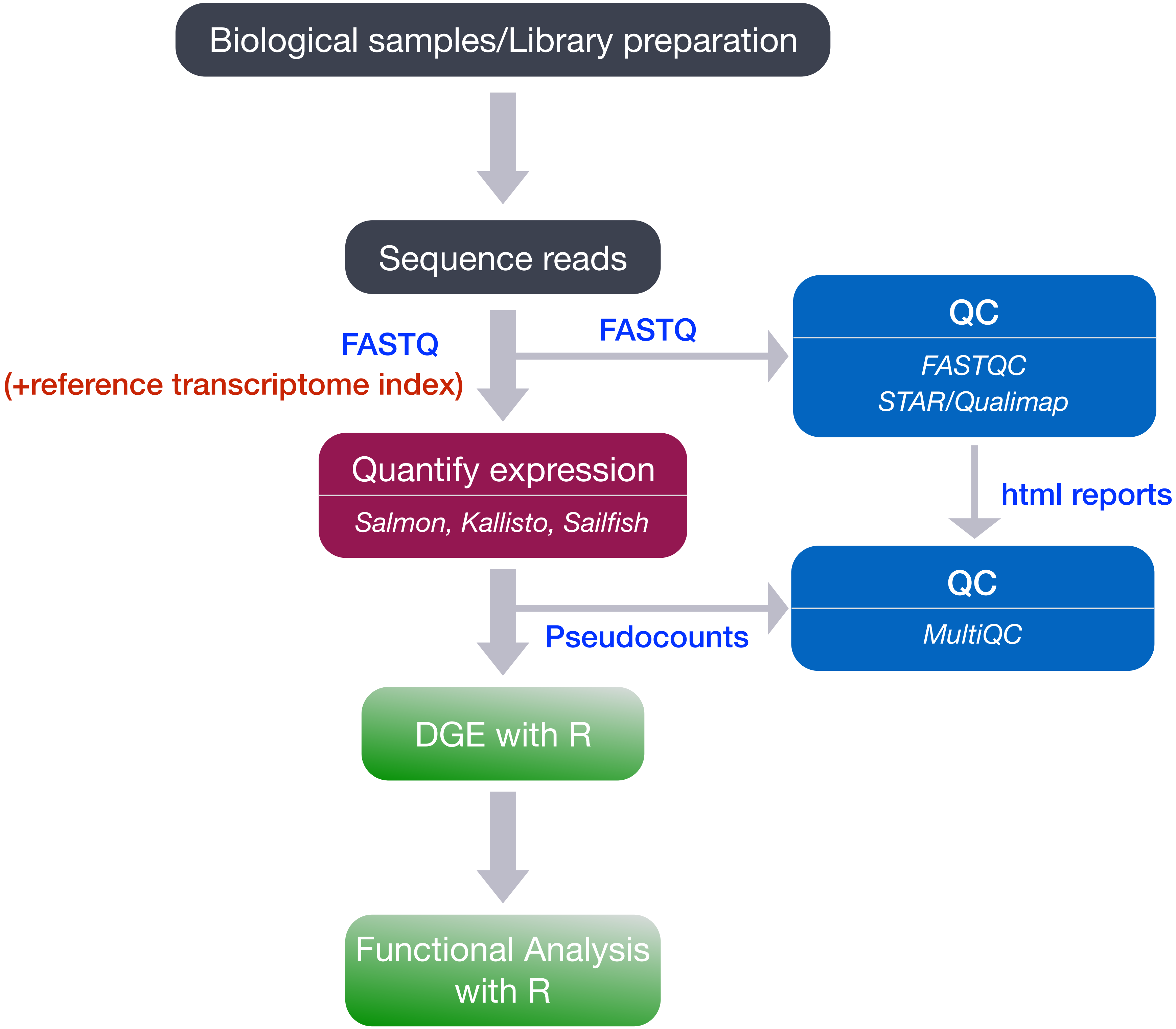
Splice-aware alignment

Splice-aware alignment tools:

HISAT2, STAR, MapSplice, SOAPSplice, Passion, SpliceMap, RUM, ABMapper, CRAC, GSNAP, HMMSplicer, Olego, BLAT

There are excellent genome aligners available that are not splice-aware. These are useful for aligning directly to genes. However, you will lose isoform information.

Bowtie2, BWA, Novoalign (not free), SOAPaligner



Biological samples/Library preparation

Sequence reads

FASTQ
(+reference transcriptome index)

FASTQ

QC

FASTQC
STAR/Qualimap

html reports

Quantify expression

Salmon, Kallisto, Sailfish

Pseudocounts

QC

MultiQC

DGE with R

Functional Analysis
with R

These materials have been developed by members of the teaching team at the Harvard Chan Bioinformatics Core (HBC). These are open access materials distributed under the terms of the Creative Commons Attribution license (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

cc BY