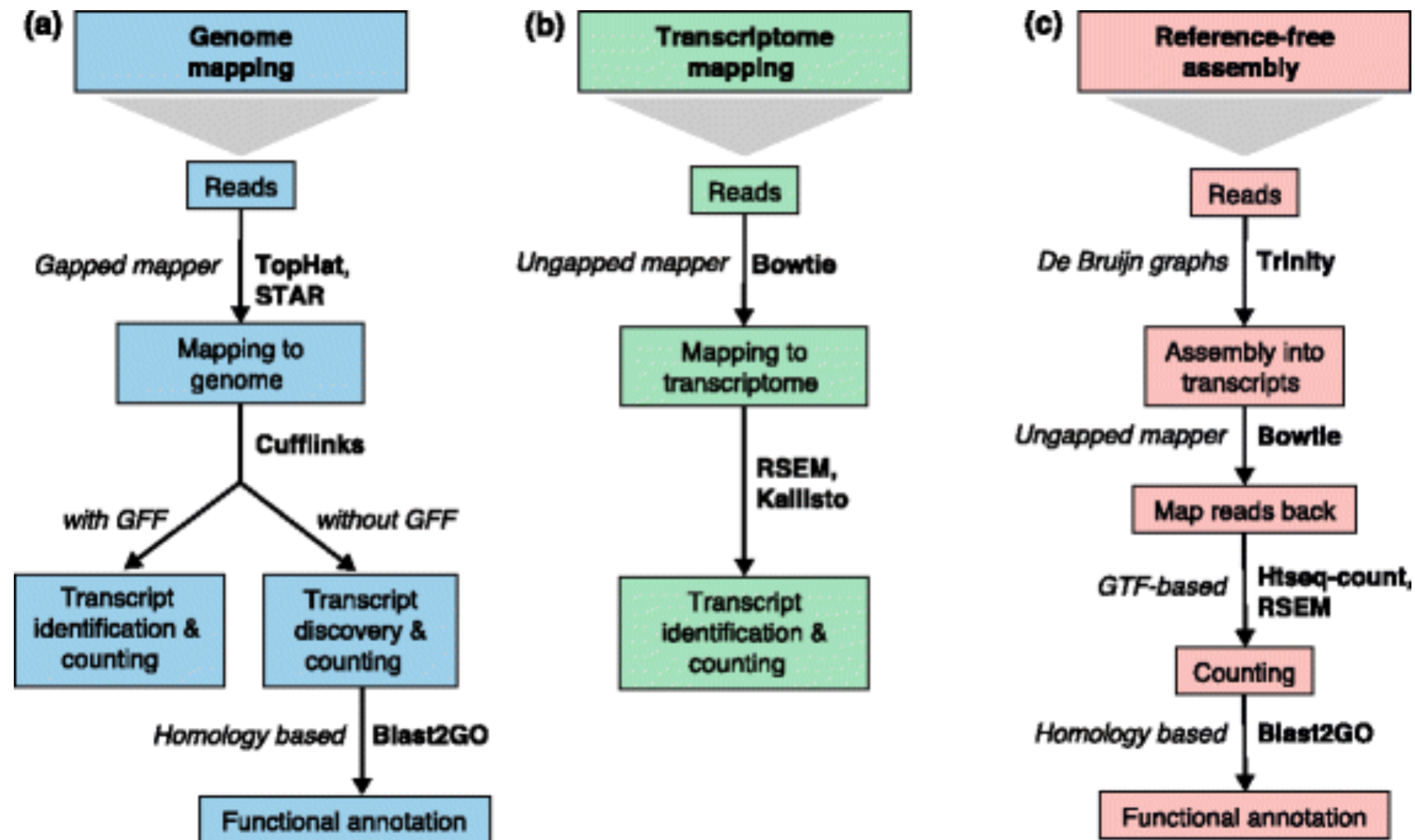
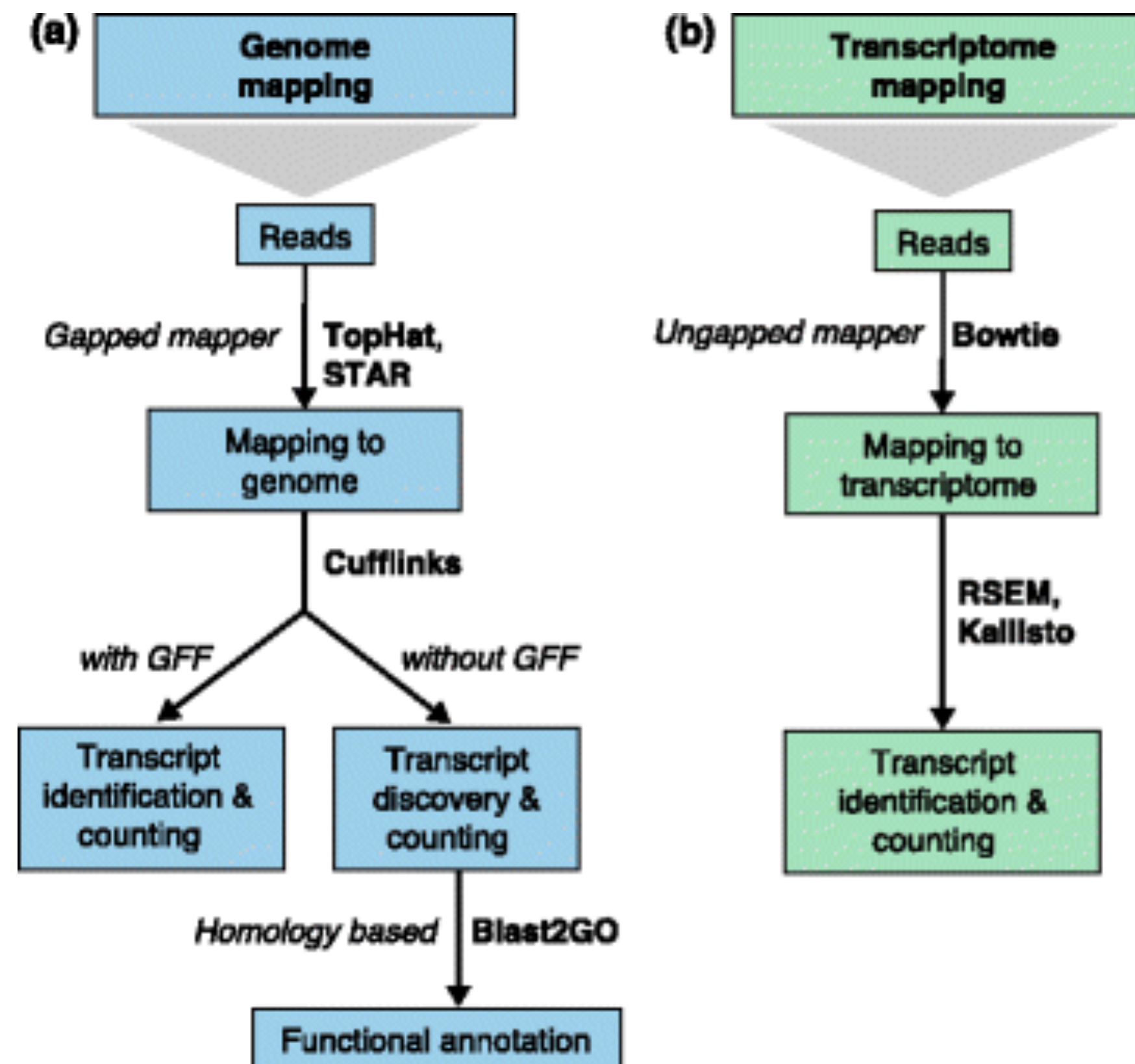


Aligning reads: tools and theory



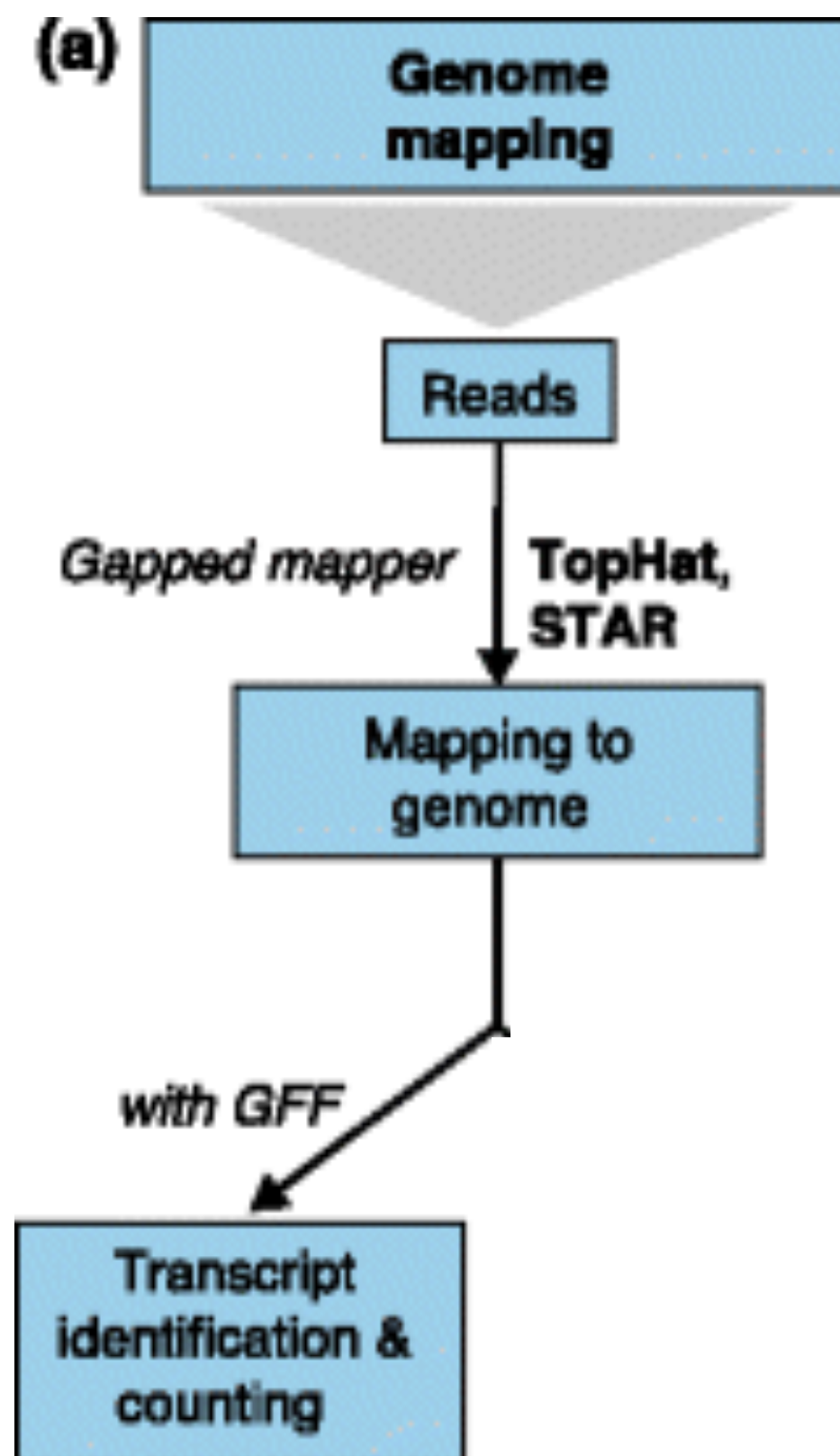
Strategies for read mapping with RNA-seq

Conesa A., et al, Genome Biology 2016, 17:13 doi:10.1186/s13059-016-0881-8

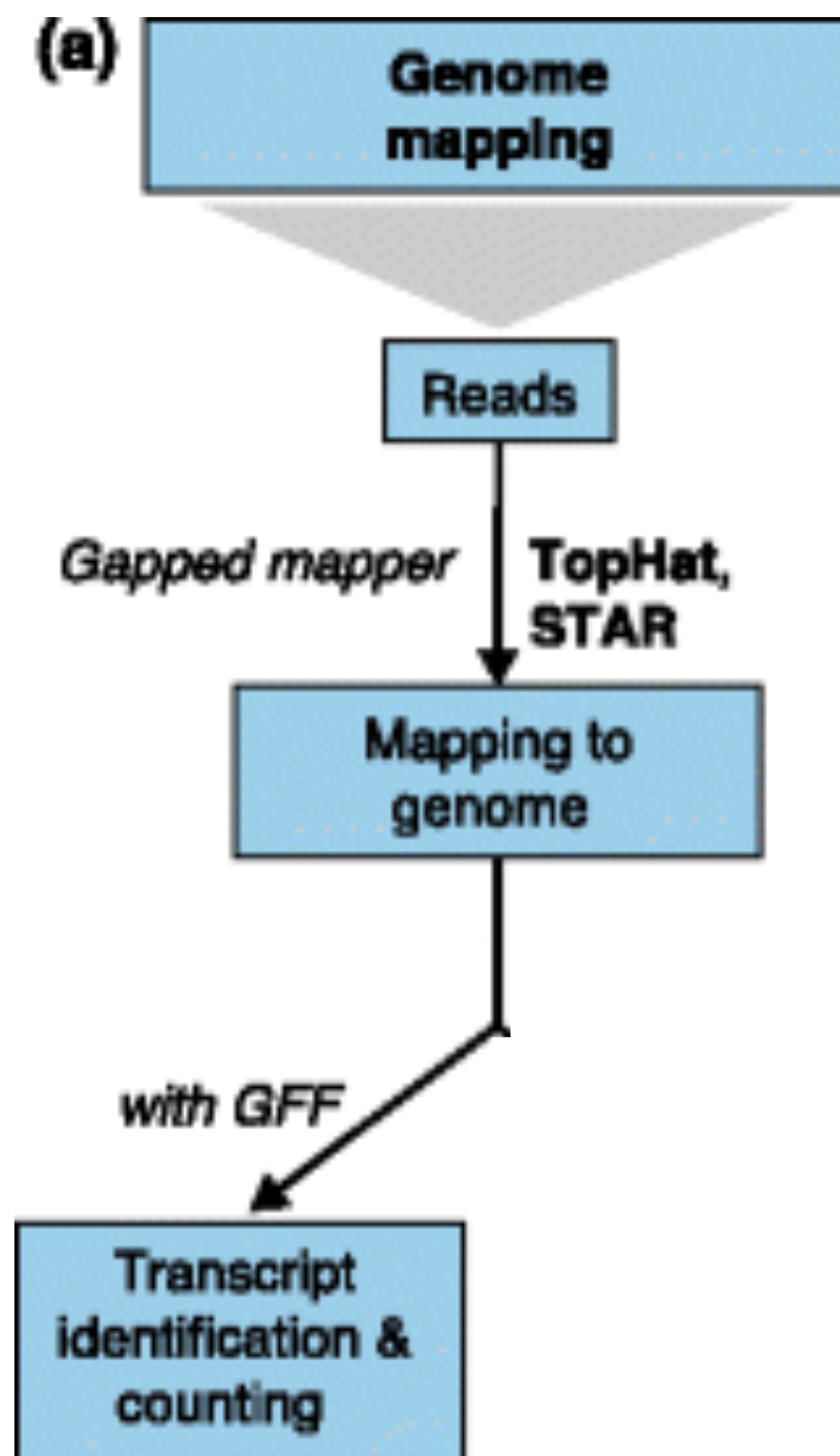


Strategies for read mapping with RNA-seq

Conesa A., et al, Genome Biology 2016, 17:13 doi:10.1186/s13059-016-0881-8



Transcriptome quantification



Biological samples/Library preparation

Sequence reads

FASTQC

Trimming

Splice-aware mapping to genome

Counting reads associated with genes

Statistical analysis to identify
differentially expressed genes

Transcriptome quantification

Genome

chrX: 52139280 152139290 152139300 152139310 152139320 152139330
--->CGCCGTCCCTCAGAAATGGAAACCTCGCTTCTCTCTGCCCCACAATGCGCAAGTCAG

Sequence read

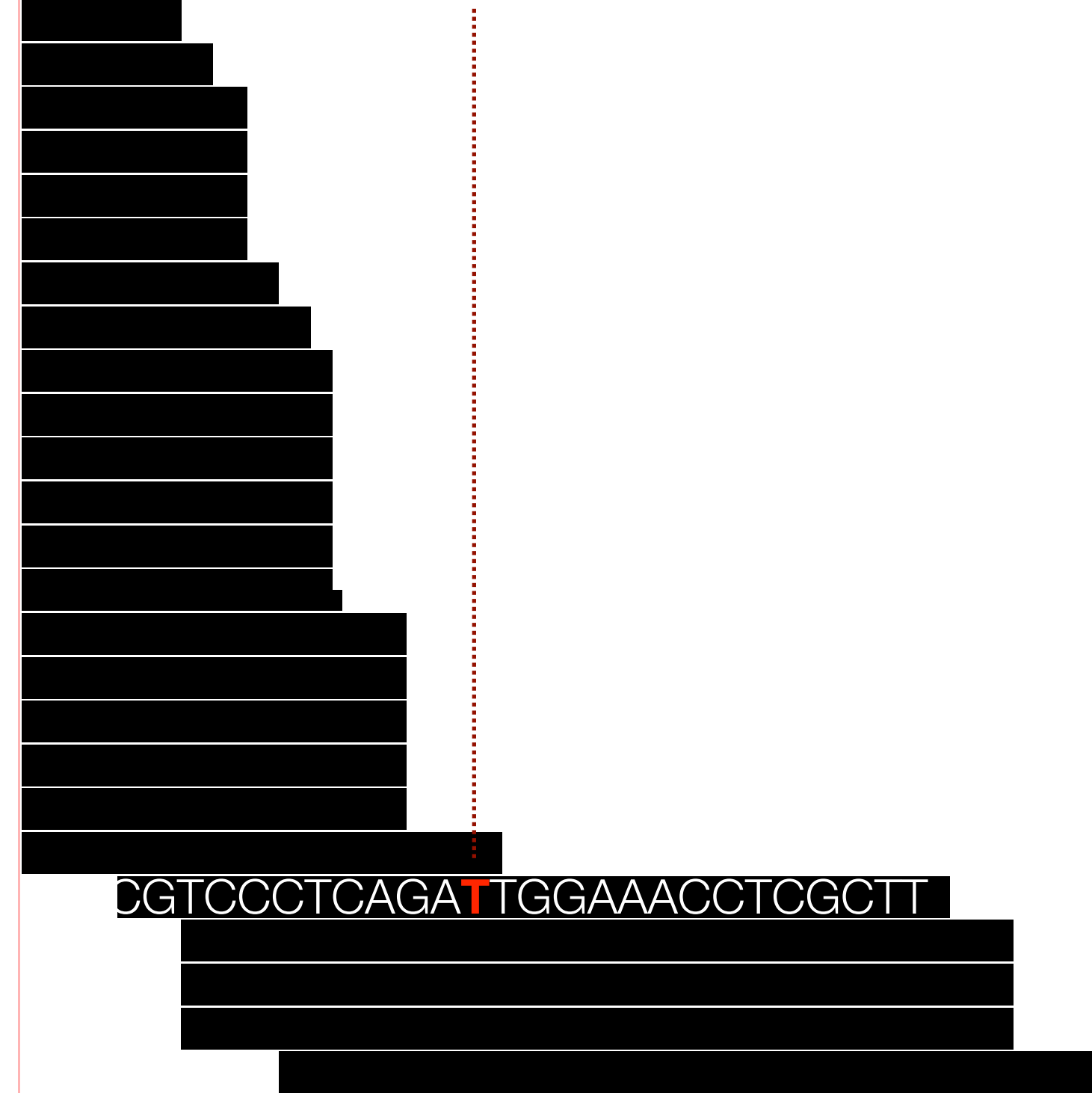
CGTCCCTCAGAAATGGAAACCTCGCTT

A simple case of string matching

Genome

chrX: 52139280 152139290 152139300 152139310 152139320 152139330
--->CGCCGTCCCTCAGAAATGGAAACCTCGCTTCTCTCTGCCCCACAATGCGCAAGTCAG

Sequence reads



Difficult in practice

- Volume of data: ~3 Gbp
- ~50% of genome is repeat regions that cannot be covered by reads
 - Simple repeats, tandem, interspersed
 - Transposons
 - Segmental duplications where mapping is unclear
- Gap or unfinished regions
 - peri-centromere, sub-telomere
 - ~5Mb unique to ethnic groups (e.g., African, Asian)
- Finishing errors(1/10,000bp), miscalled base incorporated

Challenges:

Human genome is large and complex ⁷

- Short reads: 50-150 bp (versus a very long reference)
 - Non-unique alignment
 - Sensitive to sequencing errors
- Massive amount of short reads: one lane produces ≥ 150 million 100 nucleotide reads
- Small insert size: 200-500 bp libraries

Challenges: short read NGS data

Reference ATCTCCATAGGACTAGAAGTAG

Substitution ATCTCCATAG**C**ACTAGAAGTAG

Deletion ATCTCCATAGGAC**-**AGAAGTAG

Insertion ATCTCCATAGGACTAGAAGT**T**AG

3bp deletion ATCTC**---**AGGACTAGAAGTAG

Challenges: non-exact matching

Local alignment vs Global alignment

- ▶ **Local alignment** matches the query with a *substring* (k-mer) of the reference
 - ▶ Tailored towards finding *regions of highly similar sequence* and aligning around those by working outwards to align the rest

Local Alignment

```
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
      |||| ||||| ||||| ||||| |||||
5' TACTCACGGATGAGGTACTTTAGAGGC 3'
```

Global Alignment

```
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
||||| ||||| ||||| ||||| |||||
5' ACTACTAGATT----ACGGATC--GTACTTTAGAGGCTAGCAACCA 3'
```

- ▶ A **global alignment** performs end-to-end alignment between the query and the reference

Reference ATCTCCATAGGACTAGGAAGTAG

Substitution ATCTCCATAG**C**ACTAGGAAGTAG

Deletion ATCTCCATAGGAC**-**AGGAAGTAG

Insertion ATCTCCATAGGACTAGGAAGT**T**AG

3bp deletion ATCTC**---**AGGACTAGGAAGTAG

General concepts: edit distance

Reference CGTCCCTCAGATTGGAA—CCTCGCTT

Read TCCCTCAGAATGGAAACCTCGCT

Edit distance =3

General concepts: edit distance

Building an index

- ▶ For each read we need to scan the entire corpus as fast as possible
- ▶ Having an index of the reference genome provides an efficient way to search
- ▶ Once index is built, it can be queried any number of times
- ▶ Indexes are genome and tool-specific



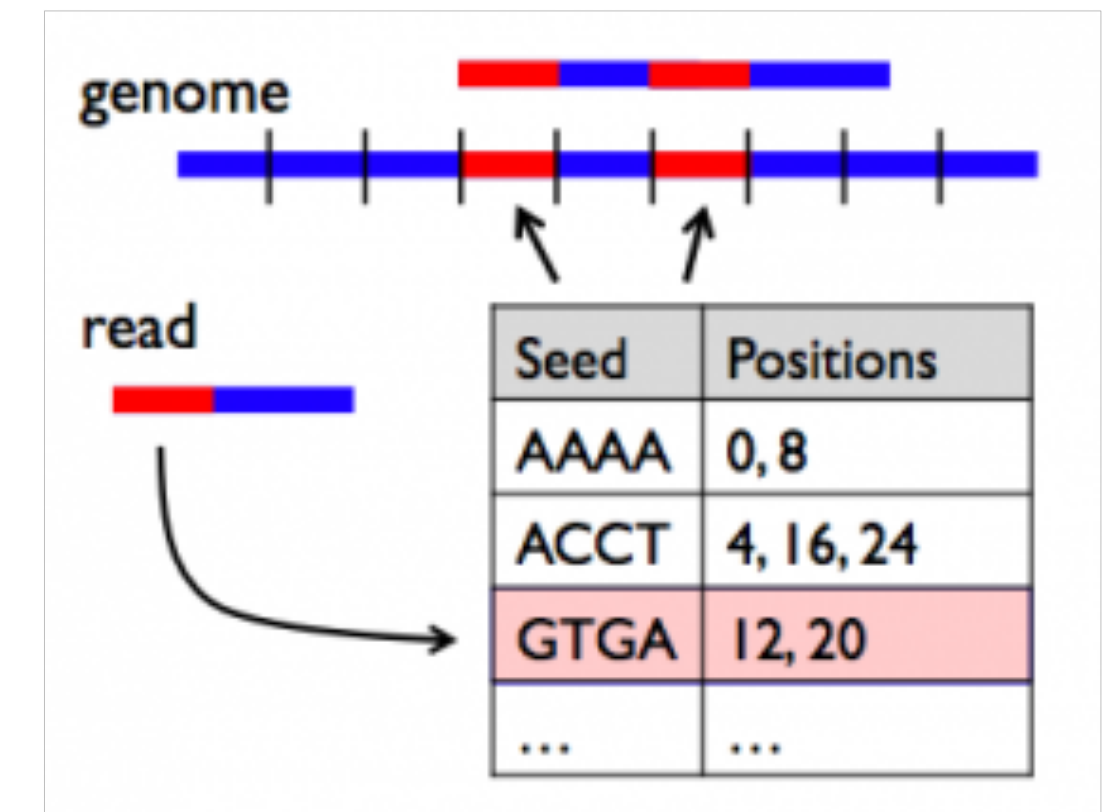
Alignment tools can be grouped based on indexing method

- ▶ Some examples include:
 - ▶ Hash-based
 - ▶ Suffix arrays
 - ▶ Burrows-Wheeler Transform

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



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)

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
sippi\$	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}

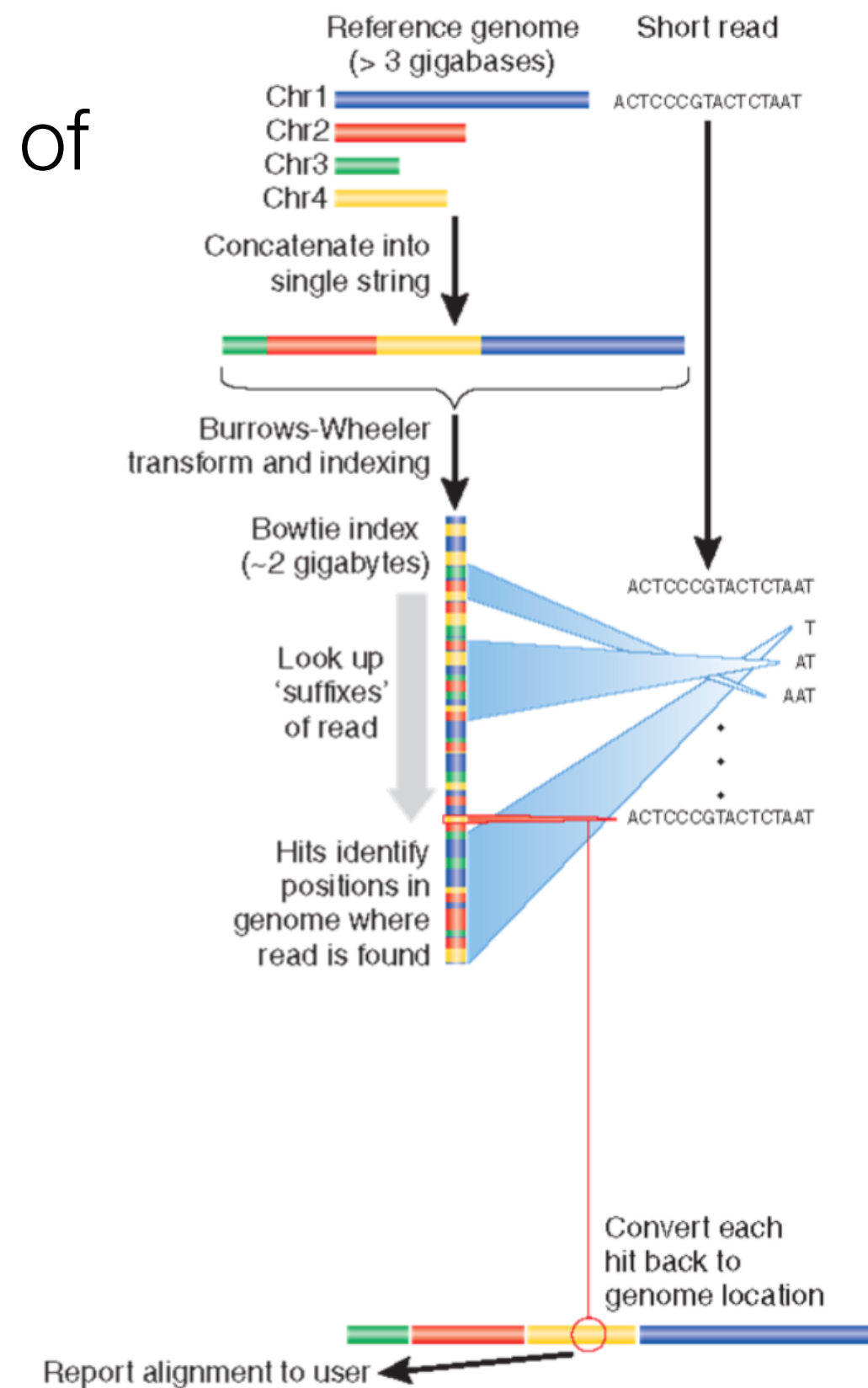
Burrows-Wheeler transform

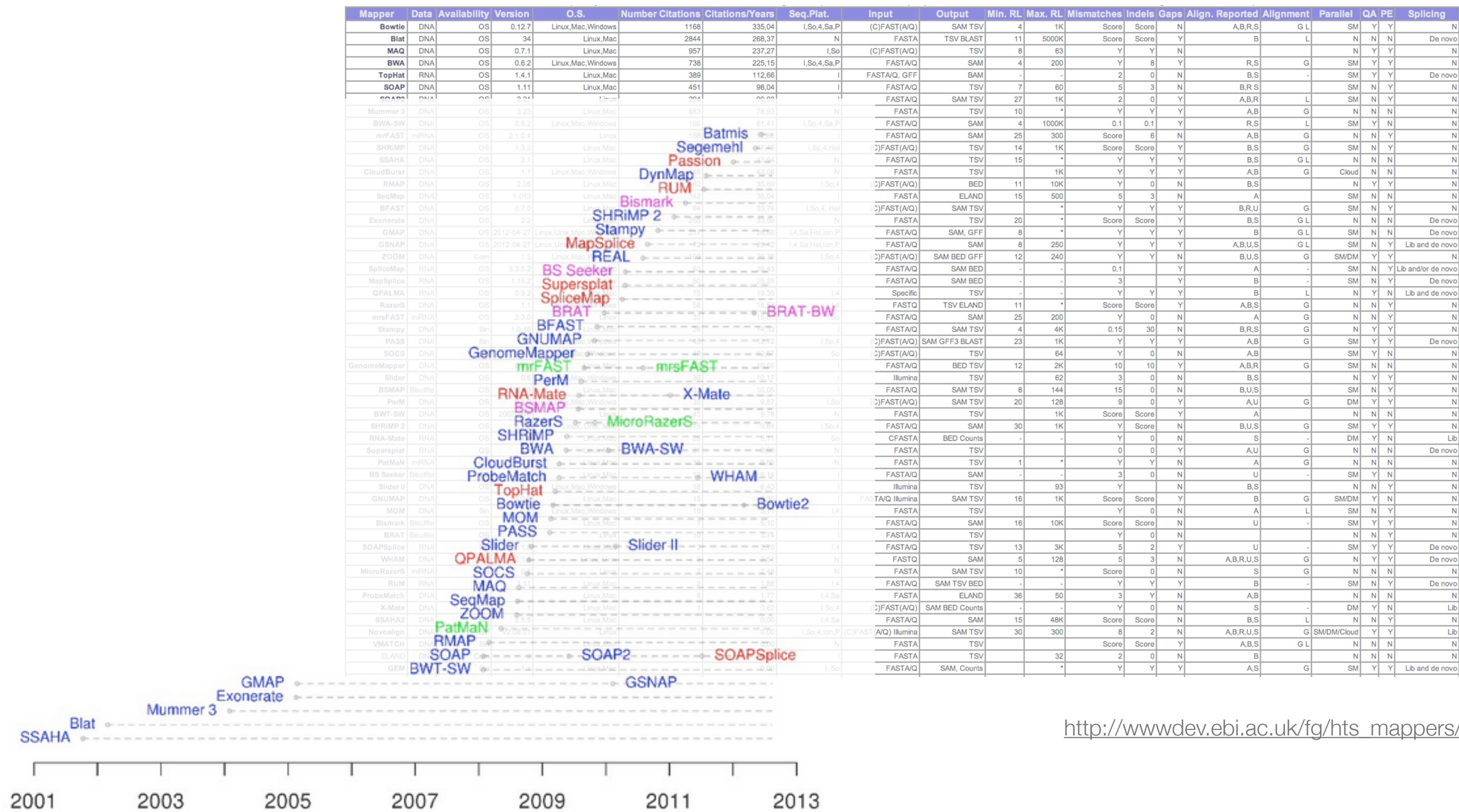
- ▶ A compressed form of suffix arrays
- ▶ Tends to put runs of the same character together rather than alphabetically, which makes the compression work well

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

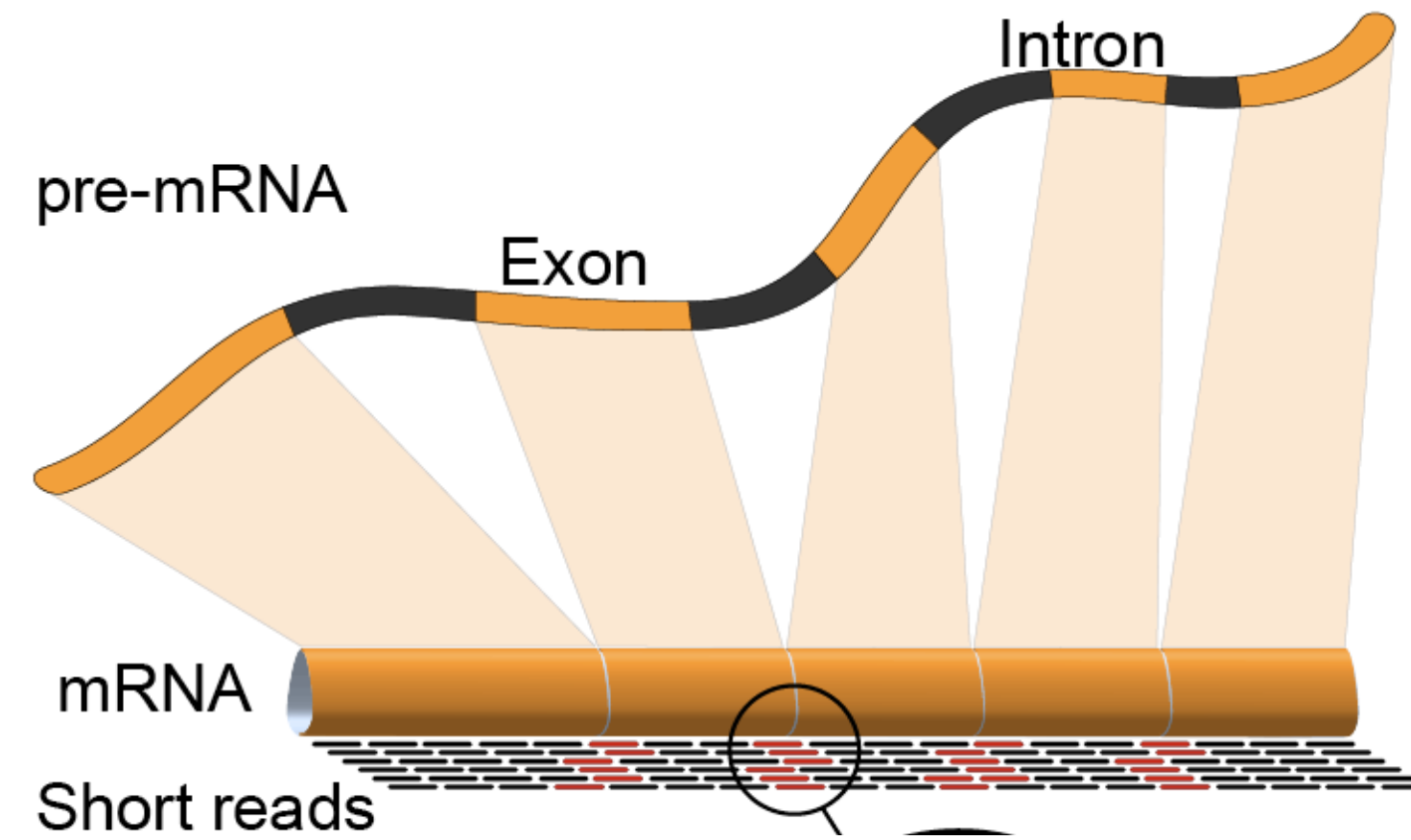
- ▶ Much less memory because of compression; ~1.5 GB of RAM required for hg19 index
- ▶ But compression results in diminished efficiency of the string search operations
- ▶ Popular Tools:
 - Bowtie2 (2012)
 - SOAP2
 - BWA-MEM (2013)



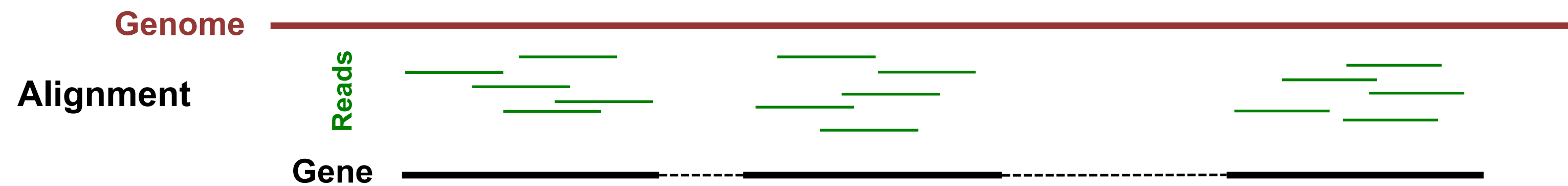
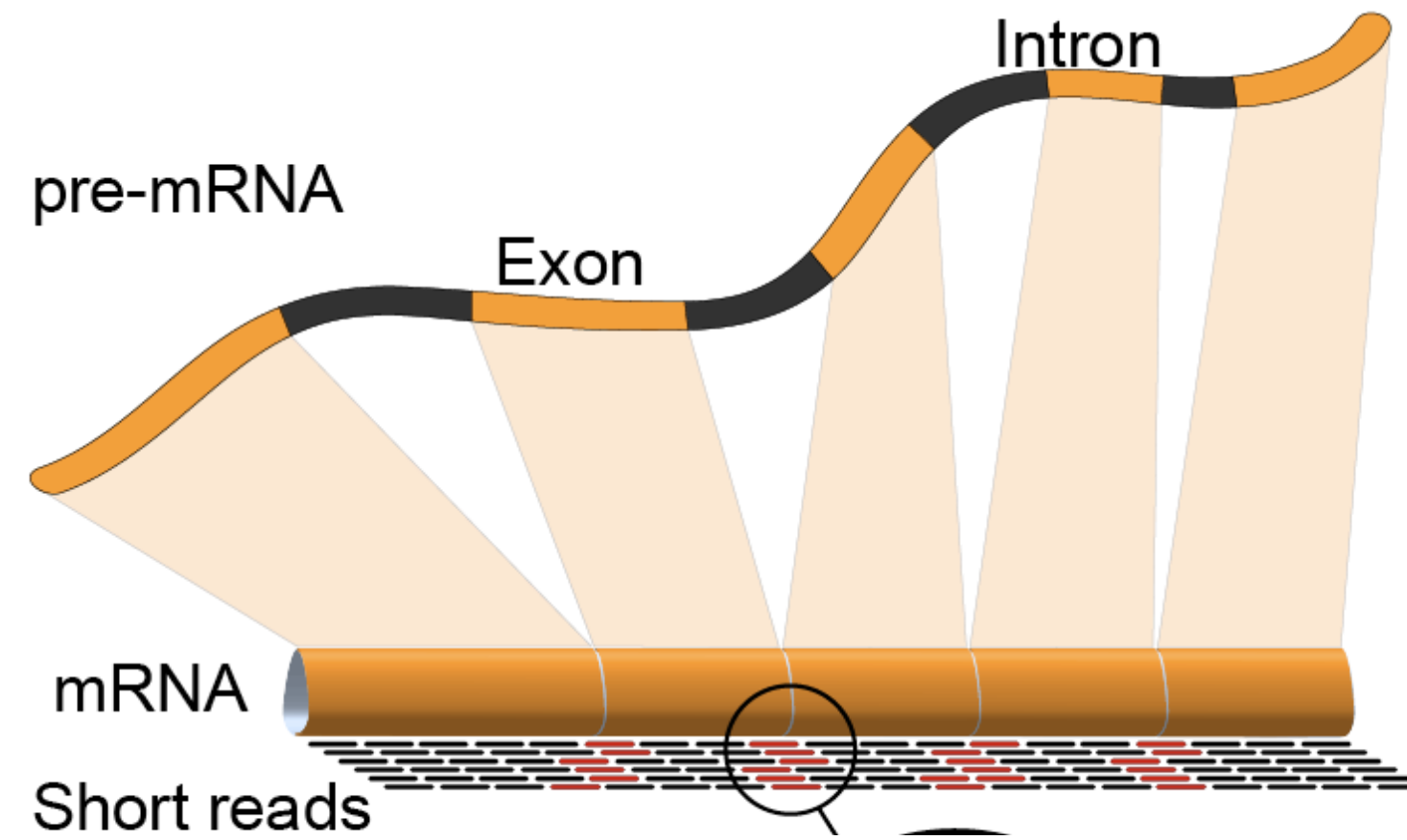


http://wwwdev.ebi.ac.uk/fg/hts_mappers/

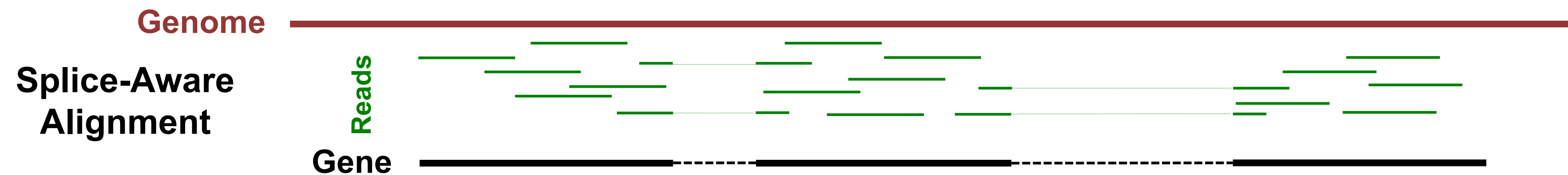
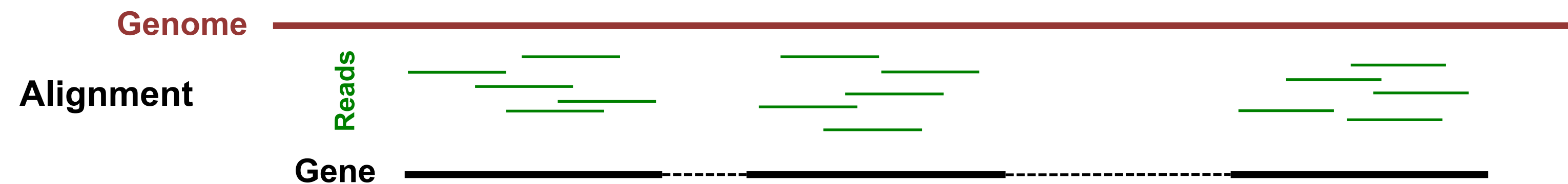
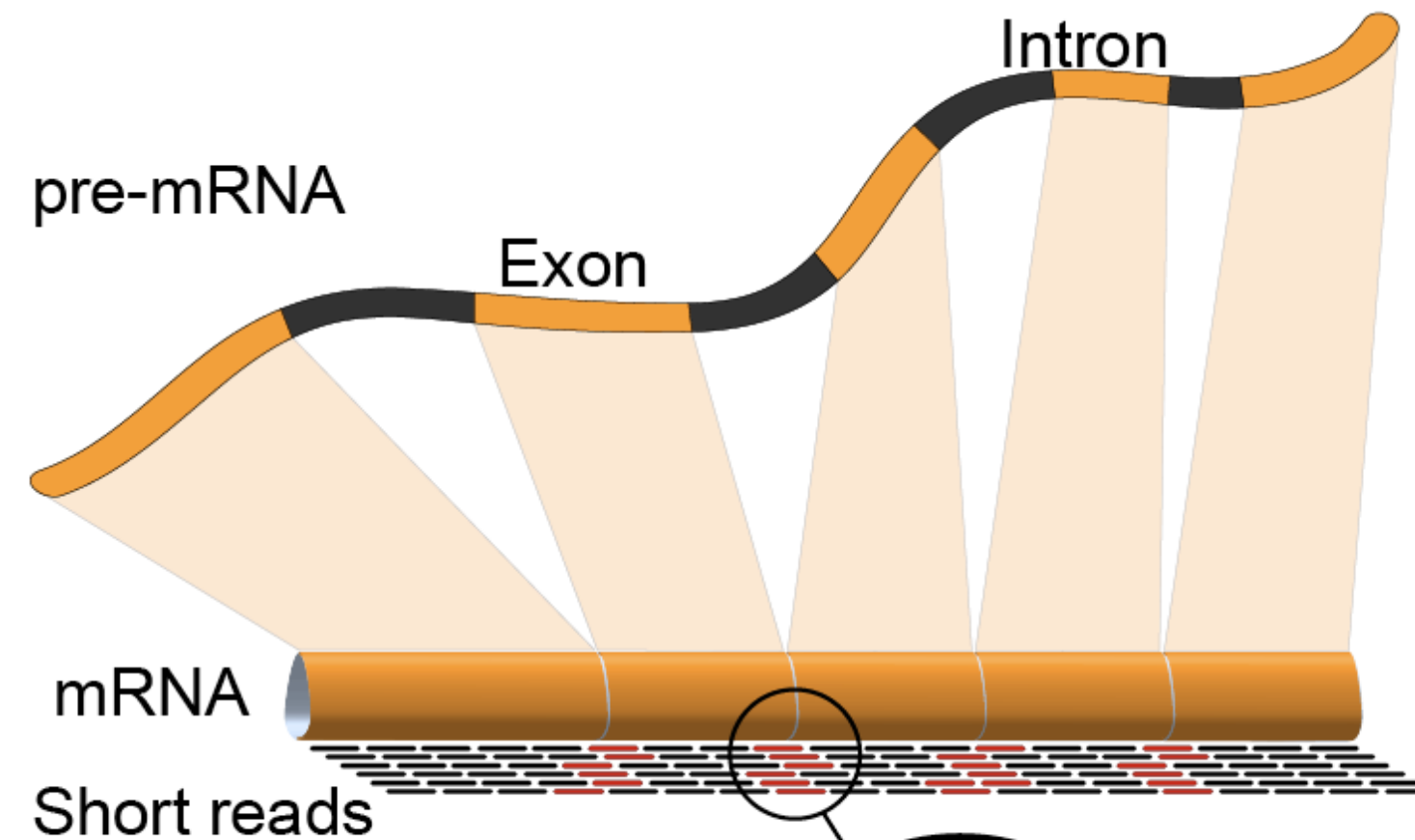
Short-read aligners: choices



Splice-aware alignment



Splice-aware alignment



Splice-aware alignment

Splice-aware alignment tools:

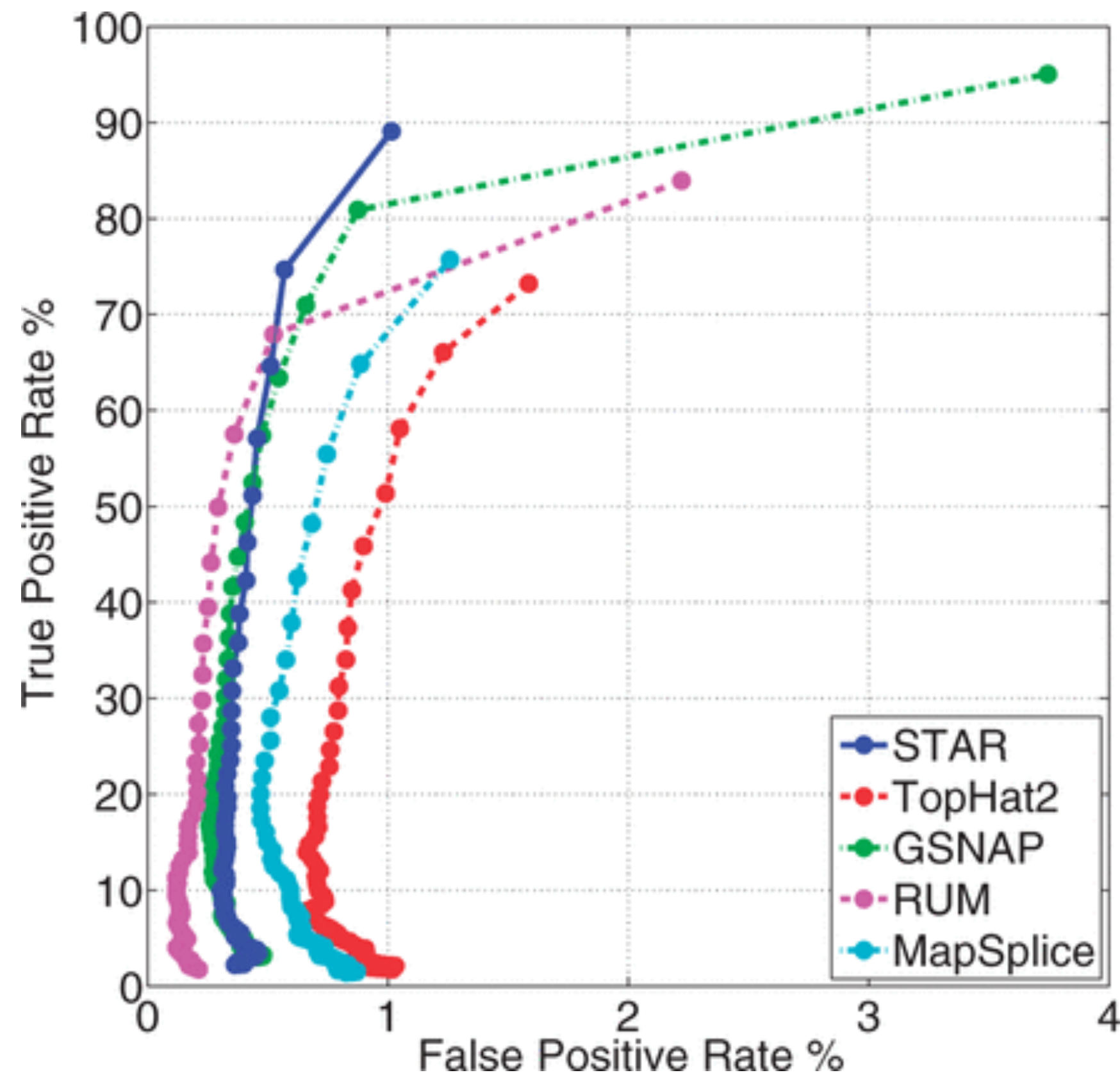
TopHat2, STAR, MapSplice, SOAPSsplice, Passion, SpliceMap,
RUM, ABMapper, CRAC, GSNAP, HMMSplicer, Olego, BLAT

There are excellent 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

Splice-aware alignment



Aligner	Mapping speed: million read pairs/hour		Peak physical RAM, GB	
	6 threads	12 threads	6 threads	12 threads
STAR	309.2	549.9	27.0	28.4
STAR sparse	227.6	423.1	15.6	16.0
TopHat2	8.0	10.1	4.1	11.3
RUM	5.1	7.6	26.9	53.8
MapSplice	3.0	3.1	3.3	3.3
GSNAP	1.8	2.8	25.9	27.0

Bioinformatics (2013) 29 (1): 15-21

The RNA-Seq specific tools

Alignment for RNA-Seq

- ▶ Use the strategy that is most relevant based on the quality of your genome and GTF
- ▶ Choose an aligner that can allow for a read to be “split” across distant regions to account for splice events
- ▶ Evaluate your computational resources and use an aligner that would work best within the confines of the available memory and CPU

