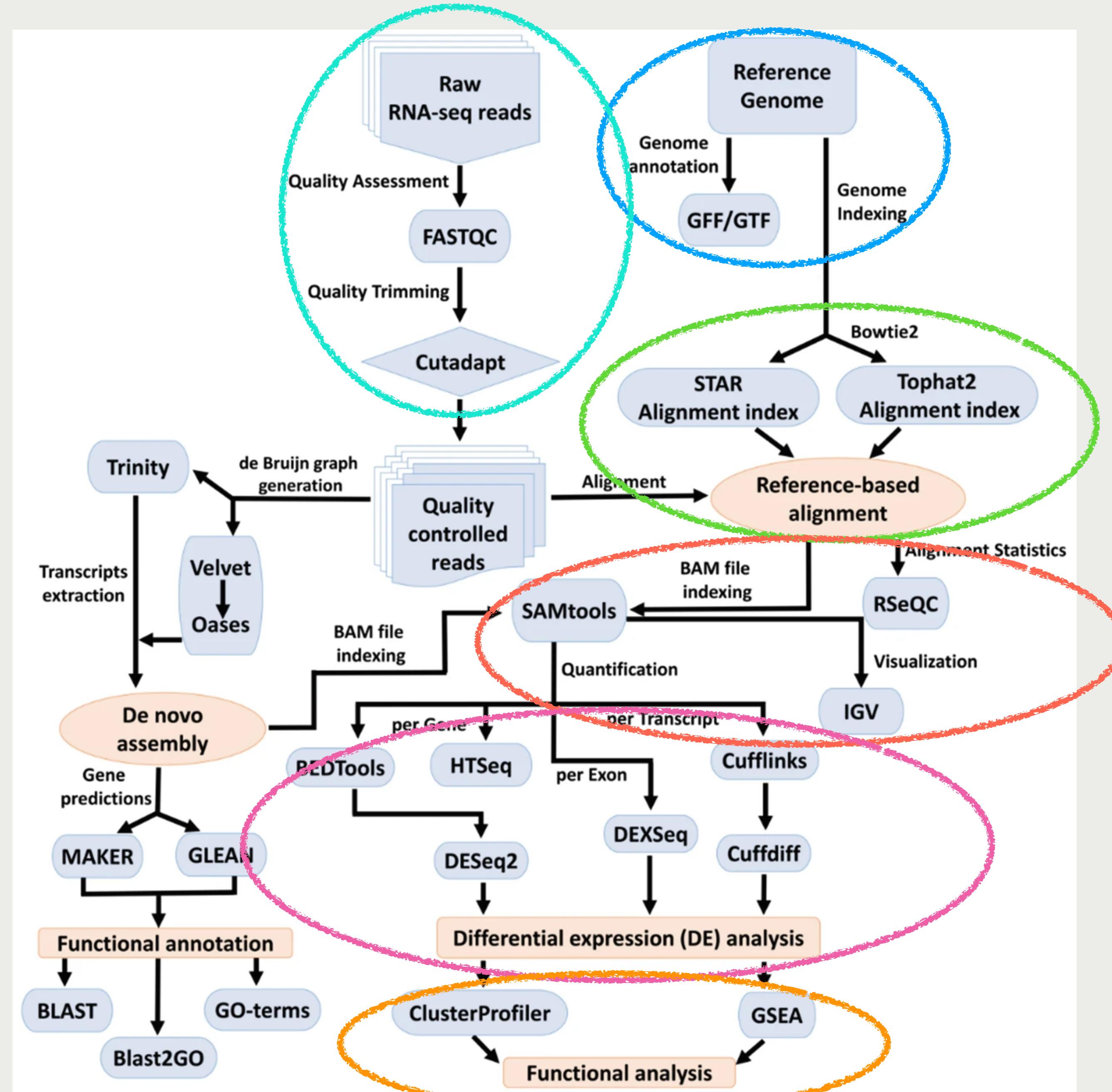


BIOL 343

Applied Bioinformatics I

Alignment/Mapping

Dr. Nic Wheeler

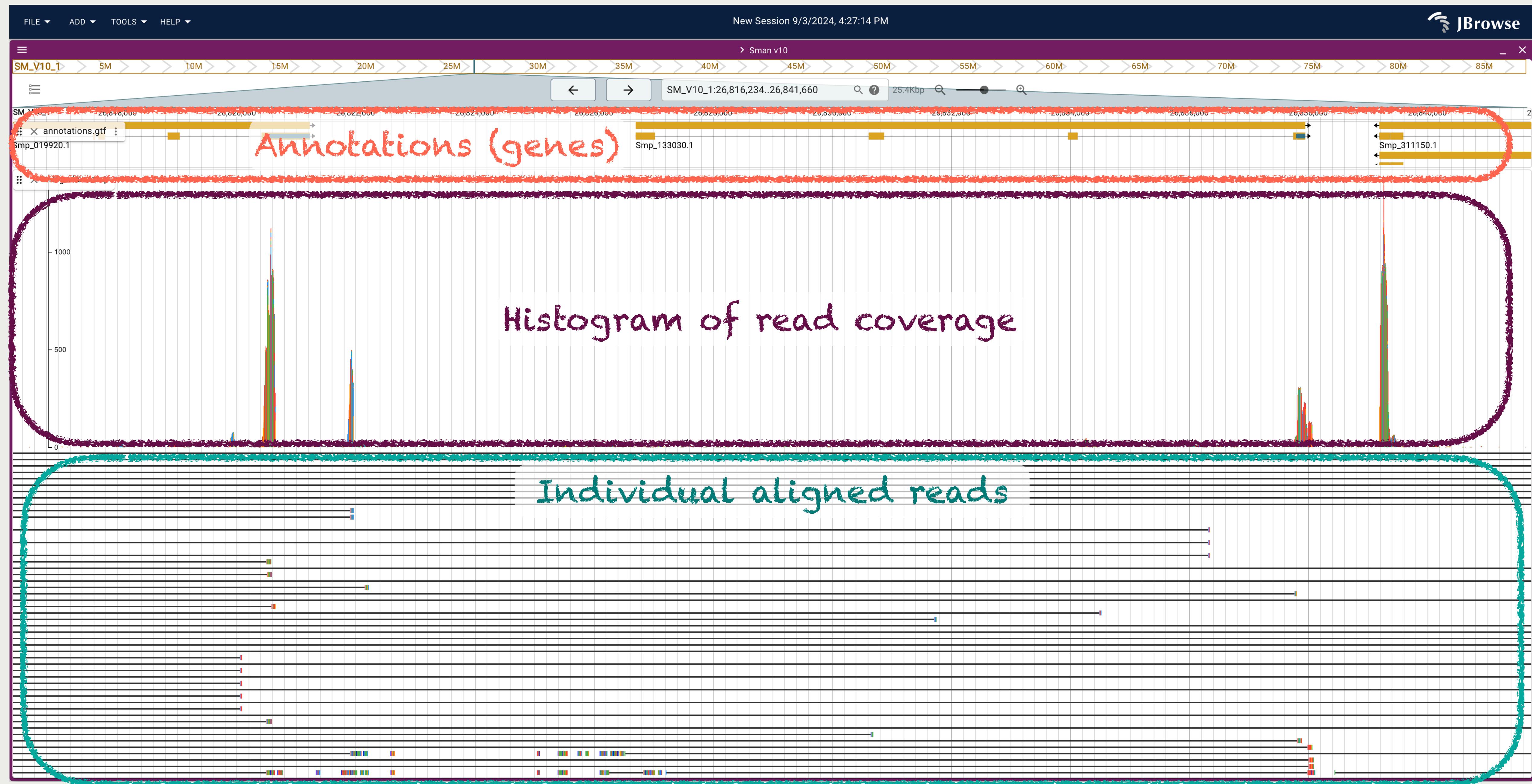


Alignment is the most important step in RNA-seq analysis

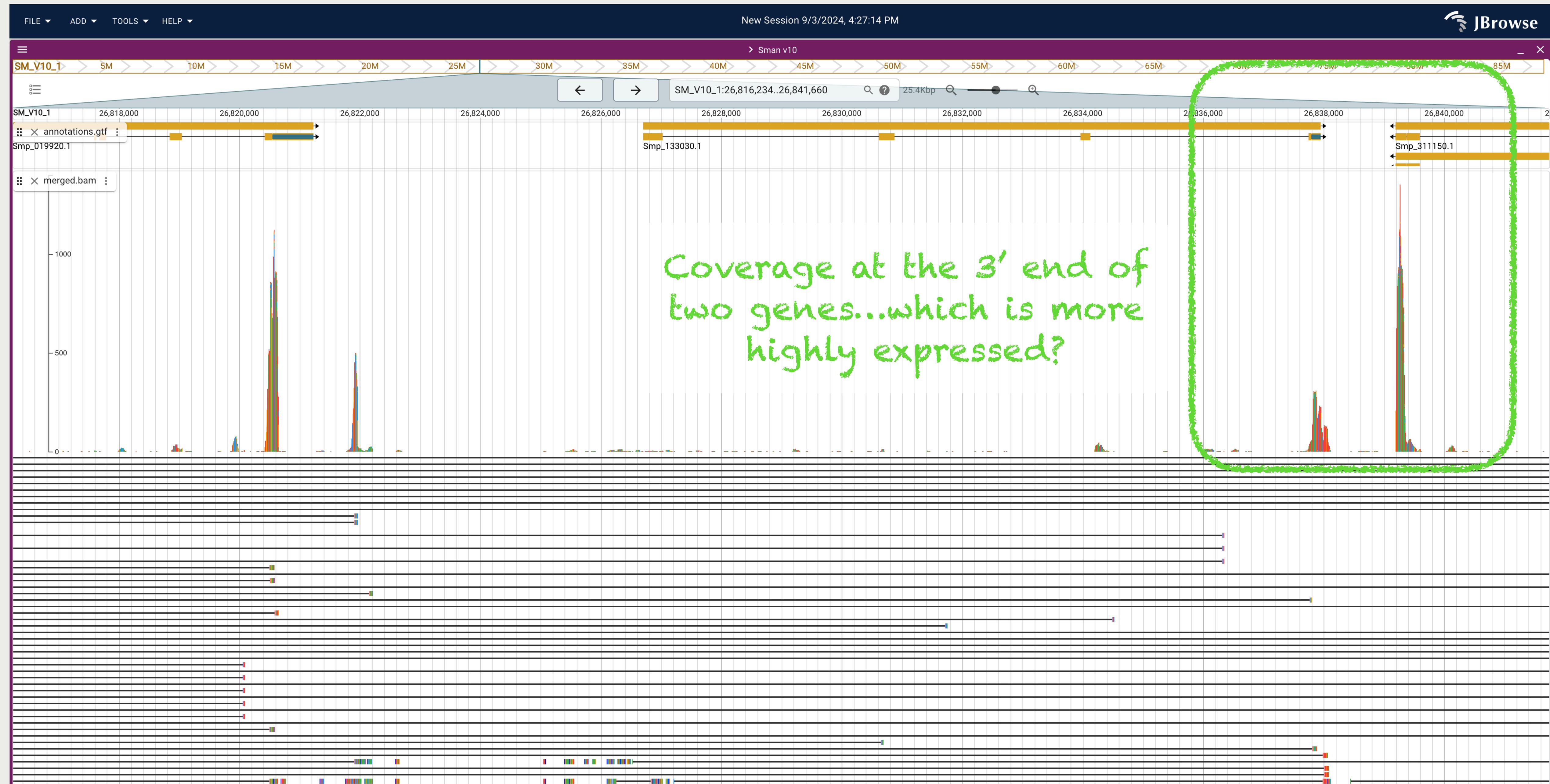
Counting (also important) and DEG ID relies on high-confidence mapping

- Recall the goal of our RNA-seq experiments...
 - Treatment vs Control
 - Mutant vs Wild type
 - ***Identify differentially expressed genes (DEGs)***
- DEGs will be identified using statistical tests comparing ***expression values*** of transcripts/genes
- Expression values will be calculated based on the number of reads that ***align/ map*** to a given genomic locus

Where we're going...



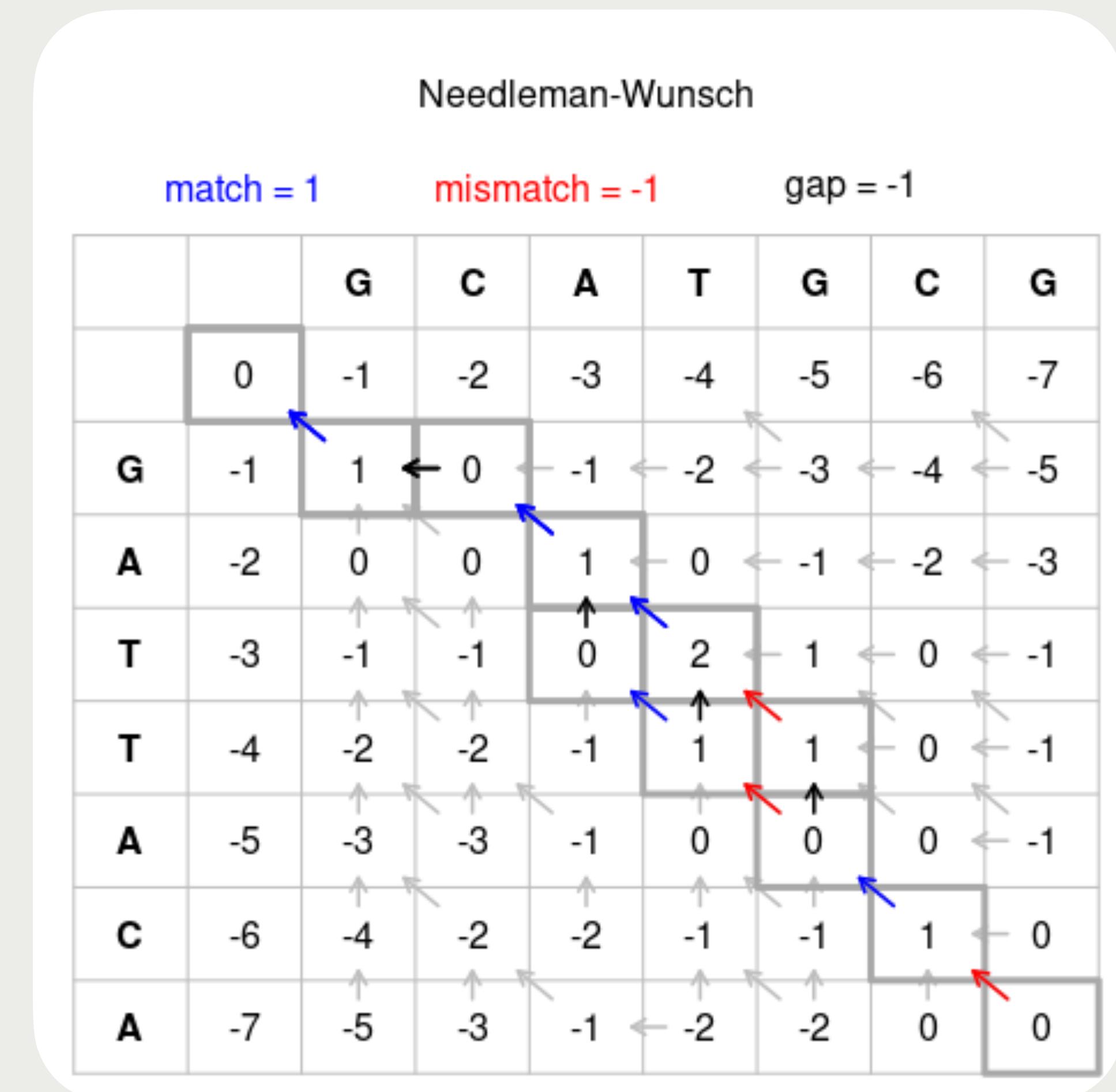
Where we're going...



Types of alignment algorithms

Needleman-Wunsch and...

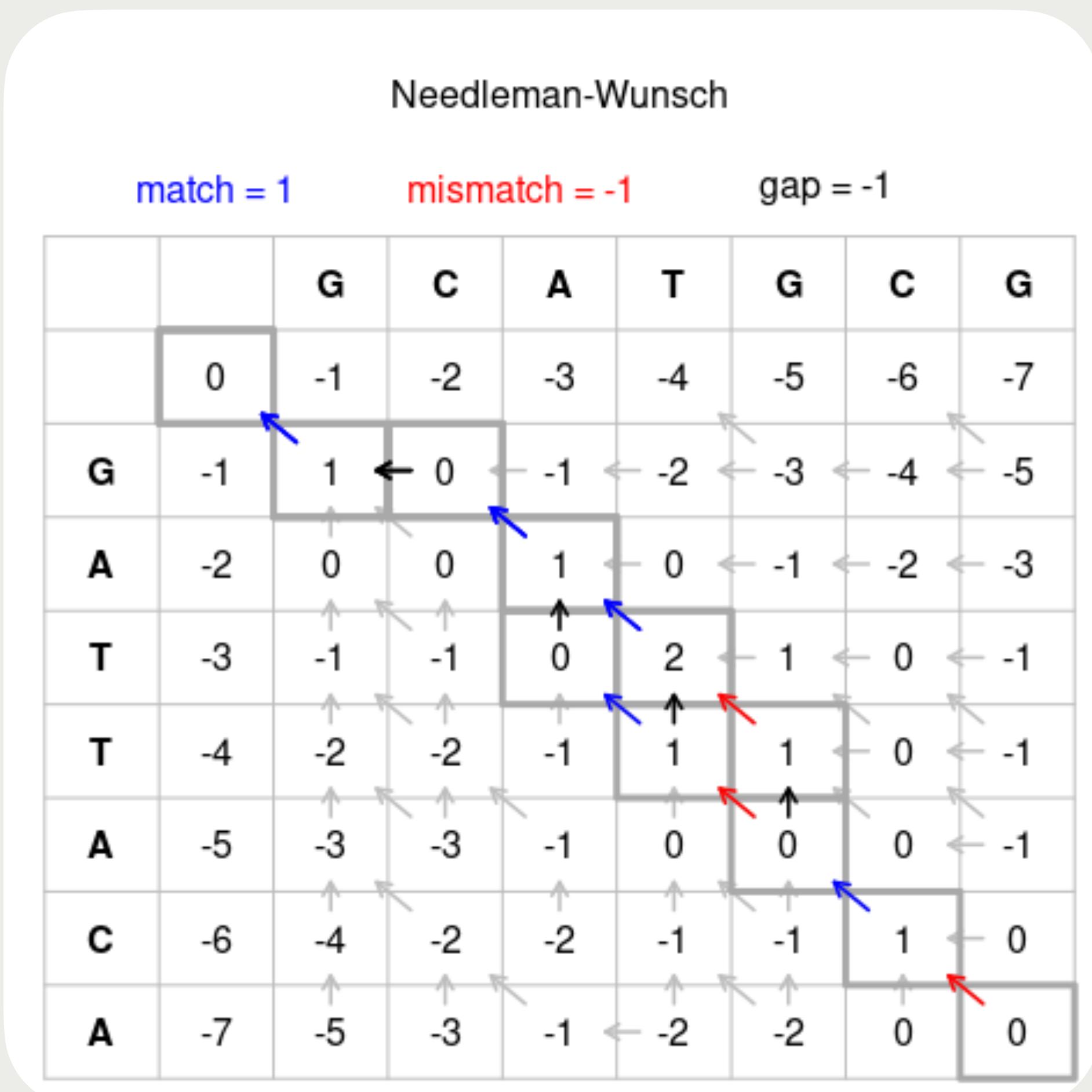
- Needleman-Wunsch (global alignment) and Smith-Waterman (local alignment)
 - Dynamic programming
 - Mismatch penalty (transitions or transversions)
 - Gap penalty



Types of alignment algorithms

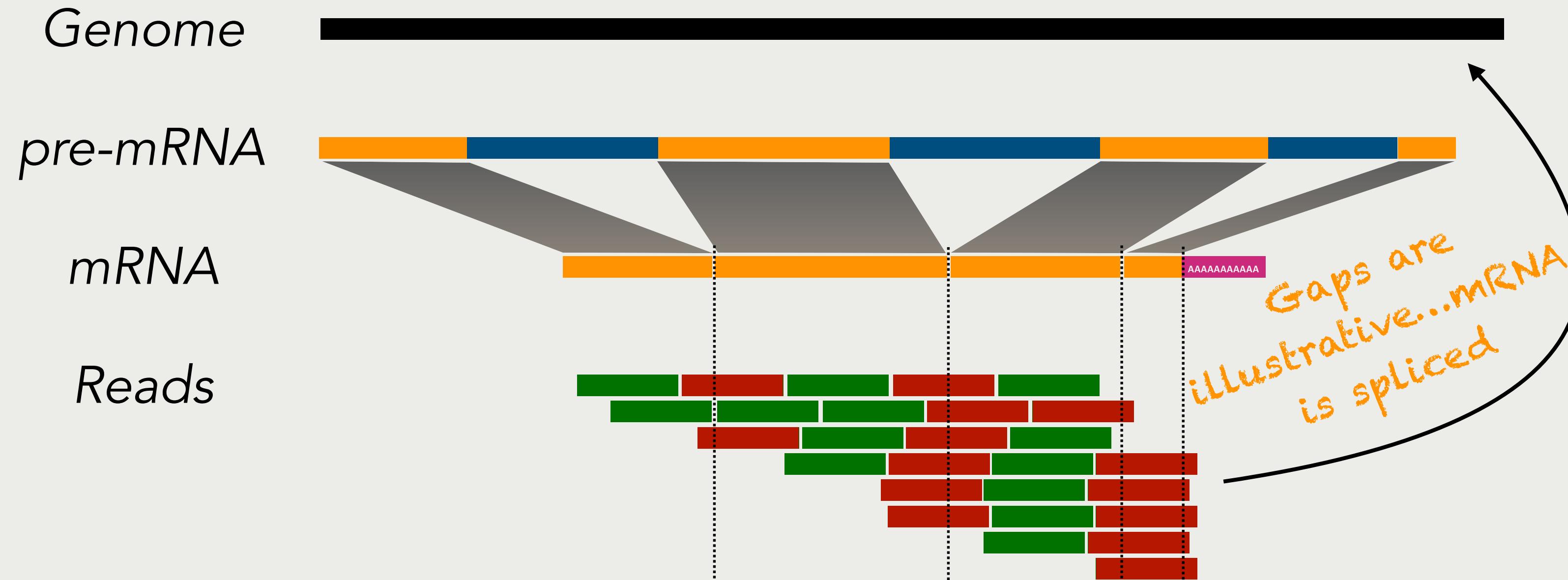
Needleman-Wunsch and...

- Problems
 - Genomes are *huge* strings with lots of repetition
 - Short reads are likely to align many locations
 - Reads should align with massive gaps if spanning an intron
 - Solution - Suffix array (STAR) or Burrow-Wheelers transform and FM-index (HISAT)



Splice-aware alignment

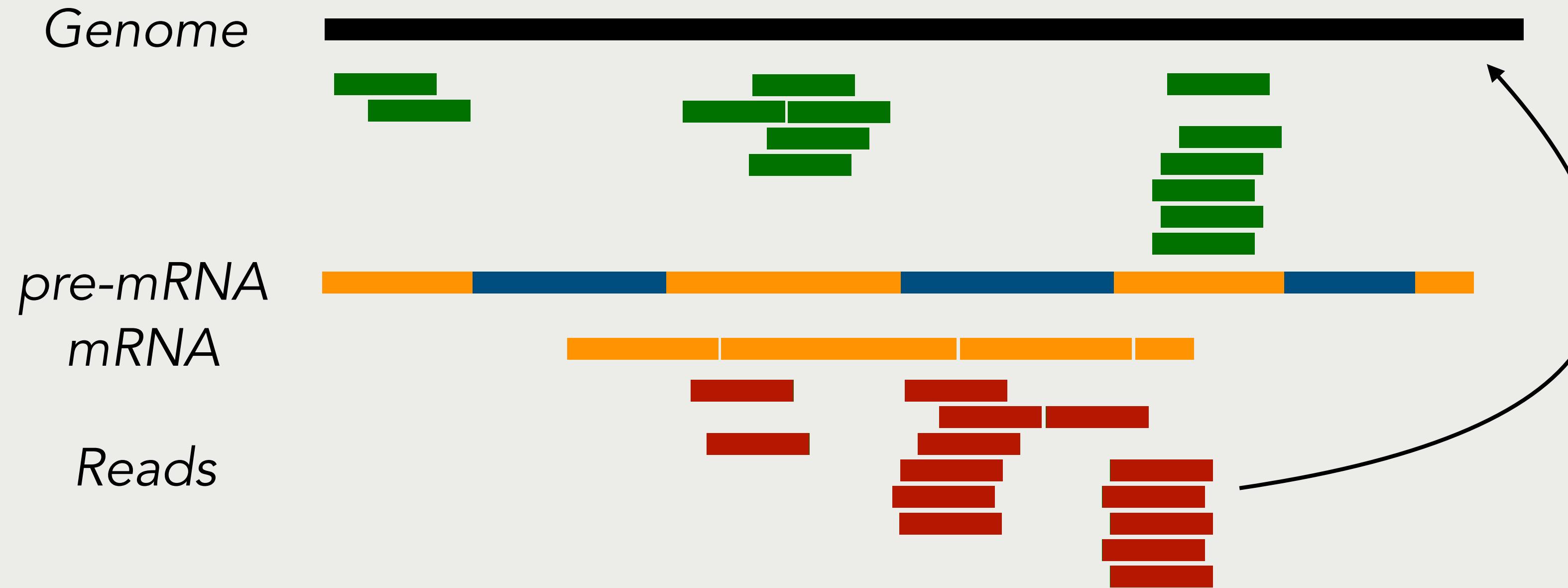
Gaps are large and encouraged



But, reads aren't aligned to a transcriptome (mRNAs), but a genome

Splice-aware alignment

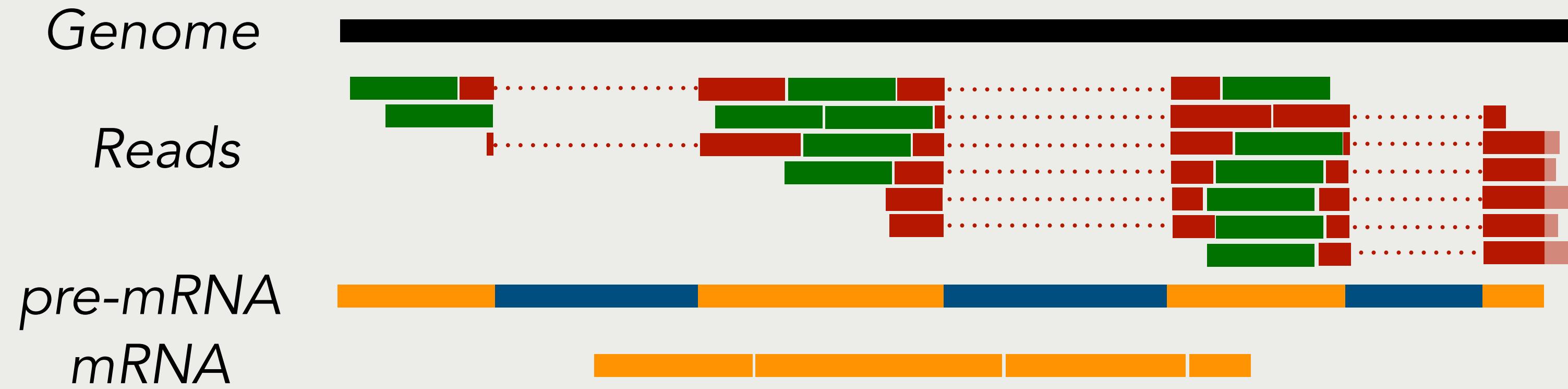
Gaps are large and encouraged



But, reads aren't aligned to a transcriptome (mRNAs), but a genome

Splice-aware alignment

Gaps are large and encouraged



Large gaps - representing introns - and reads from poly(A) tails don't align

Two (main) approaches to splice-aware alignment

STAR and HISAT

STAR

Spliced Transcripts Alignment to
a Reference

Published in 2013

40574 citations

Requires a lot of RAM; ultra fast

HISAT

Hierarchical Indexing for Spliced
Alignment of Transcripts

Published in 2015

17667 citations

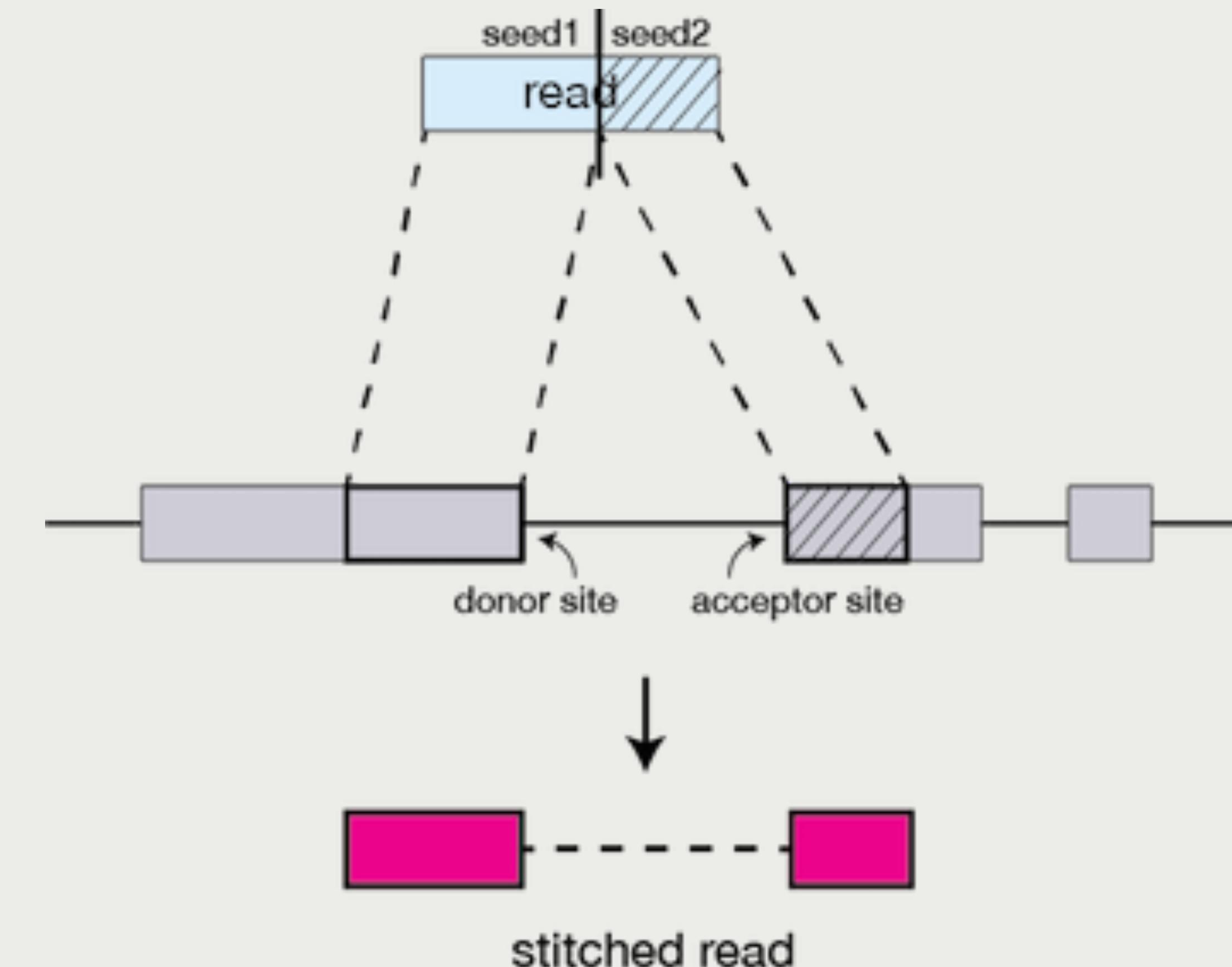
Less RAM needed; still fast

Index/compress/transform (or some combination of all) the reference genome to make searches faster

Two (main) approaches to splice-aware alignment

STAR

1. Starting from the first base, find the Maximal Mappable Prefix (MMP, seed 1) of the read
 - The end of seed 1 will map to a splice donor
2. Find the MMP of the remainder of the read
 - The end of seed 2 will map to a splice acceptor
3. Cluster and stitch the seeds based on proximity (if multiple seeds 1/2, stitch those that are closest together)



Two (main) approaches to splice-aware alignment

STAR

- Uses a suffix array of the reference genome
 - Every substring of the genome sorted lexicographically
 - Given a search string P , two binary searches to find the boundaries
 - gtg - binary search to find boundary 1 at index 7, binary search to find boundary 2 at index 9
 - Counts the number of occurrences ($9-7+1$)
 - Many developments (ongoing) in 1) generating the SA and 2) searching the SA

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
	t	g	t	g	t	g	t	g	c	a	c	c	g	\$
0														
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														

0	13	\$
1	9	accg\$
2	8	caccg\$
3	10	ccg\$
4	11	cg\$
5	12	g\$
6	7	gcaccg\$
7	5	gtgcaccg\$
8	3	gtgtgcaccg\$
9	1	gtgtgtgcaccg\$
10	6	tgcaccg\$
11	4	tgtgcaccg\$
12	2	tgtgtgcaccg\$
13	0	tgtgtgtgcaccg\$

Two (main) approaches to splice-aware alignment

STAR

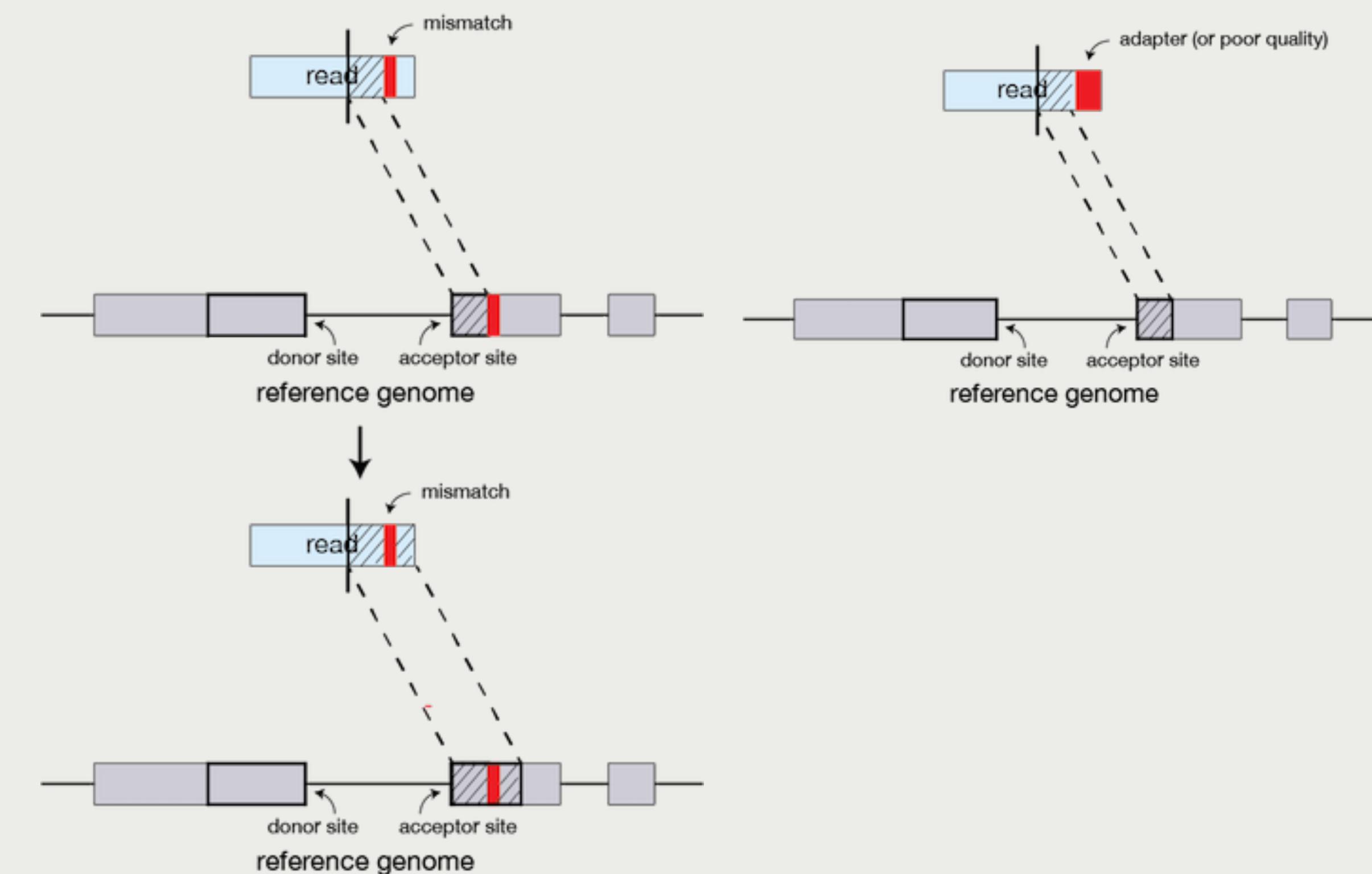
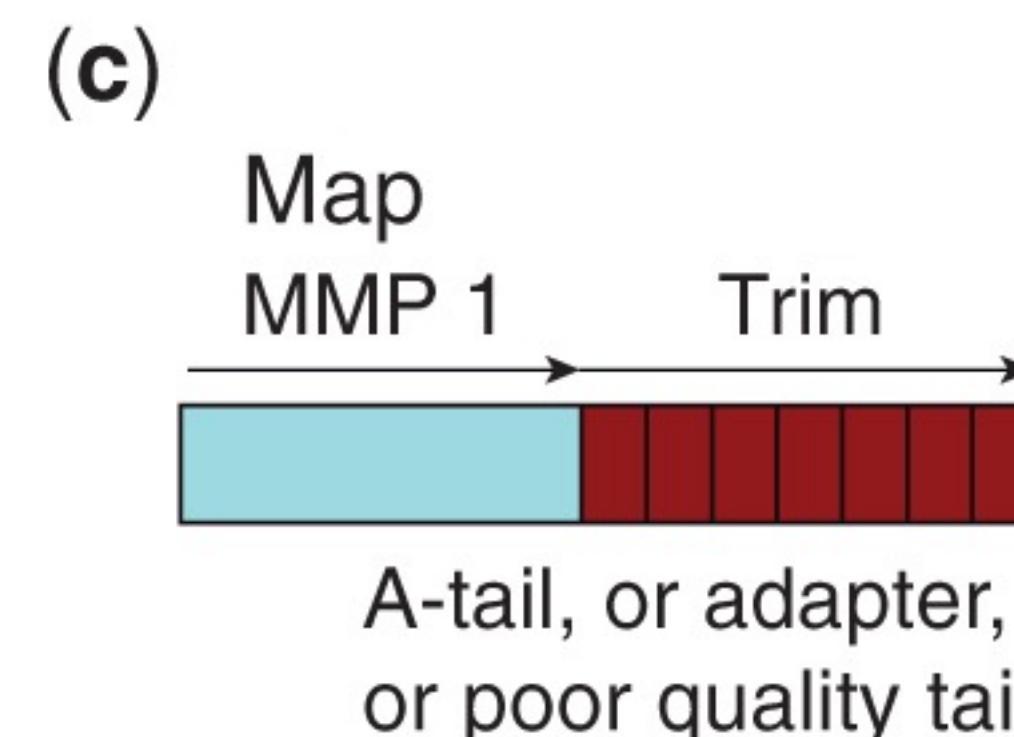
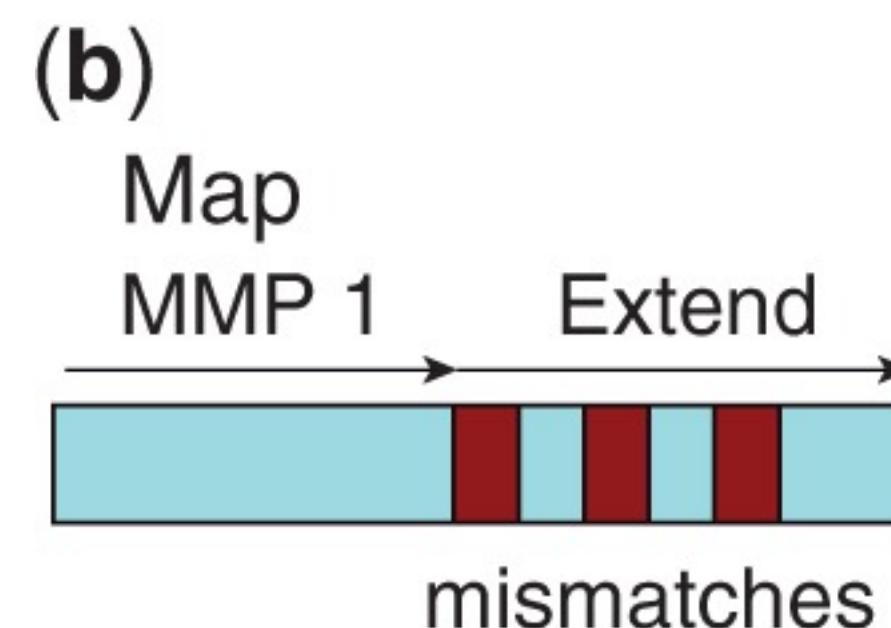
- Only two binary searches - ultrafast!
- Suffix array of a large genome (16x) - very big RAM!
 - 32 GB RAM for human genome
- Generating suffix array - kinda slow!
- For more on suffix arrays, see <https://www.cs.cmu.edu/~dga/csa.pdf>

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
t		g	t	g	t	g	t	g	c	a	c	c	g	\$
	0	13	\$											
	1	9	accg\$											
	2	8	caccg\$											
	3	10	ccg\$											
	4	11	cg\$											
	5	12	g\$											
	6	7	gcaccg\$											
	7	5	gtgcaccg\$											
	8	3	gtgtgcaccg\$											
	9	1	gtgtgtgcaccg\$											
	10	6	tgcaccg\$											
	11	4	tgtgcaccg\$											
	12	2	tgttgtgcaccg\$											
	13	0	tgttgtgtgcaccg\$											

Two (main) approaches to splice-aware alignment

STAR

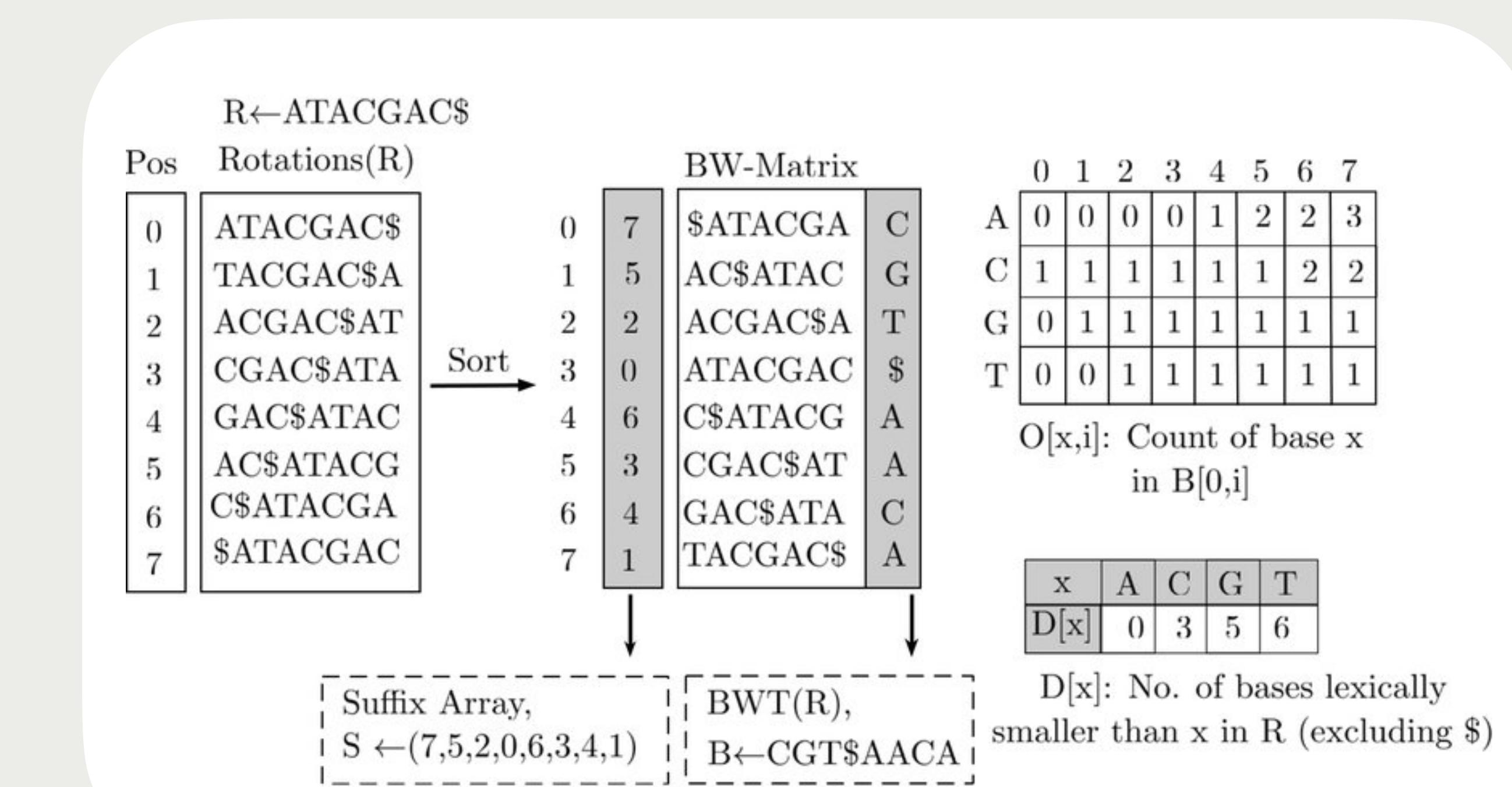
- Other advantages to this approach:
 - Robust to mismatches - MMPs can be extended
 - Can trim (**soft clip**) if extension of MMP results in many mismatches



Two (main) approaches to splice-aware alignment

HISAT

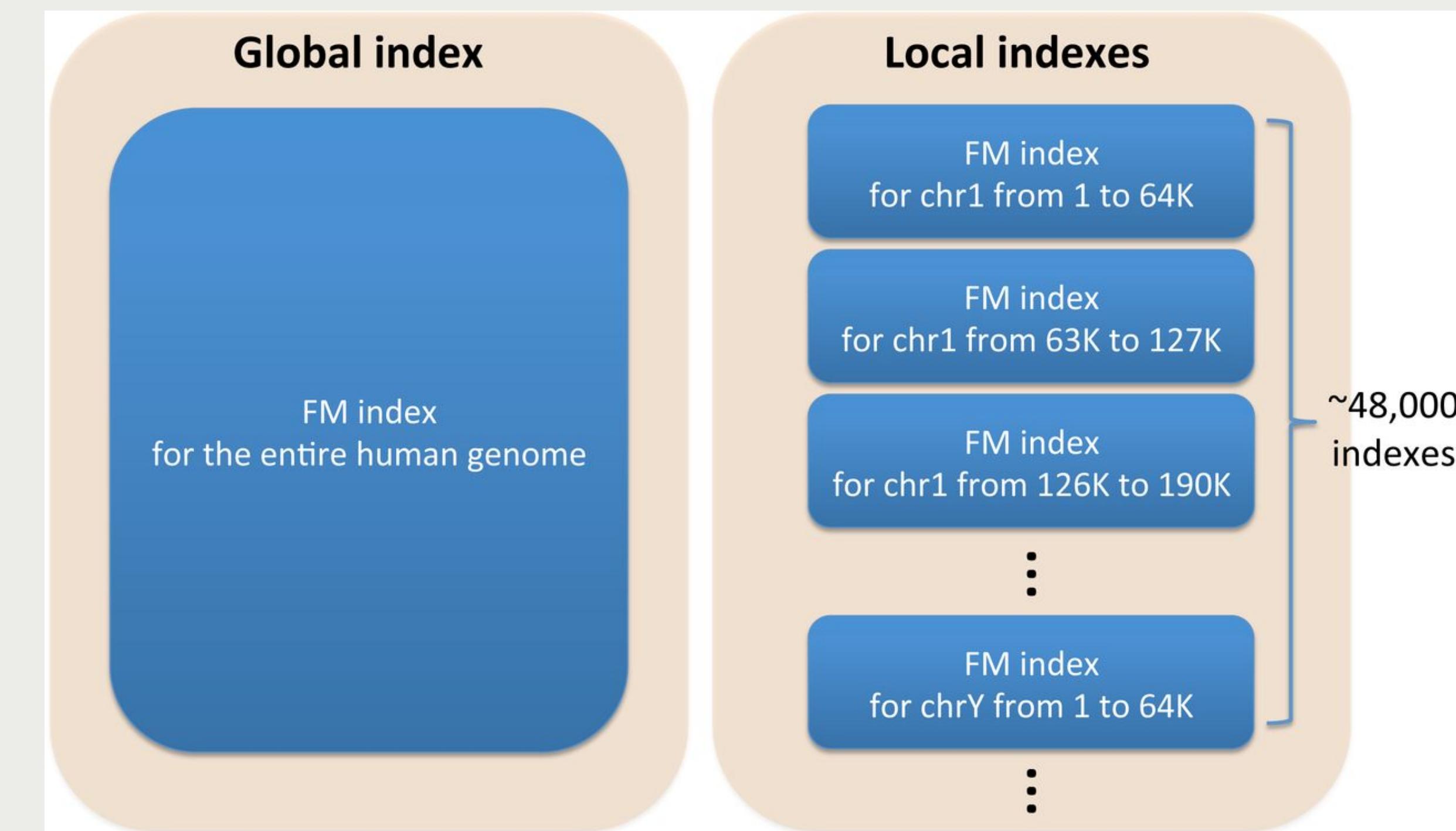
- Burrows-Wheeler Tranform (BWT) to losslessly and reversibly compress the genome
- Stored in an FM-index
 - Related to a SA (compressed SA)
 - Search time is independent of genome length (not true of SA)
 - Includes the SA, the BWT, a “milestone table,” and a count array (C)
- Substrings can quickly be located and counted



Two (main) approaches to splice-aware alignment

HISAT

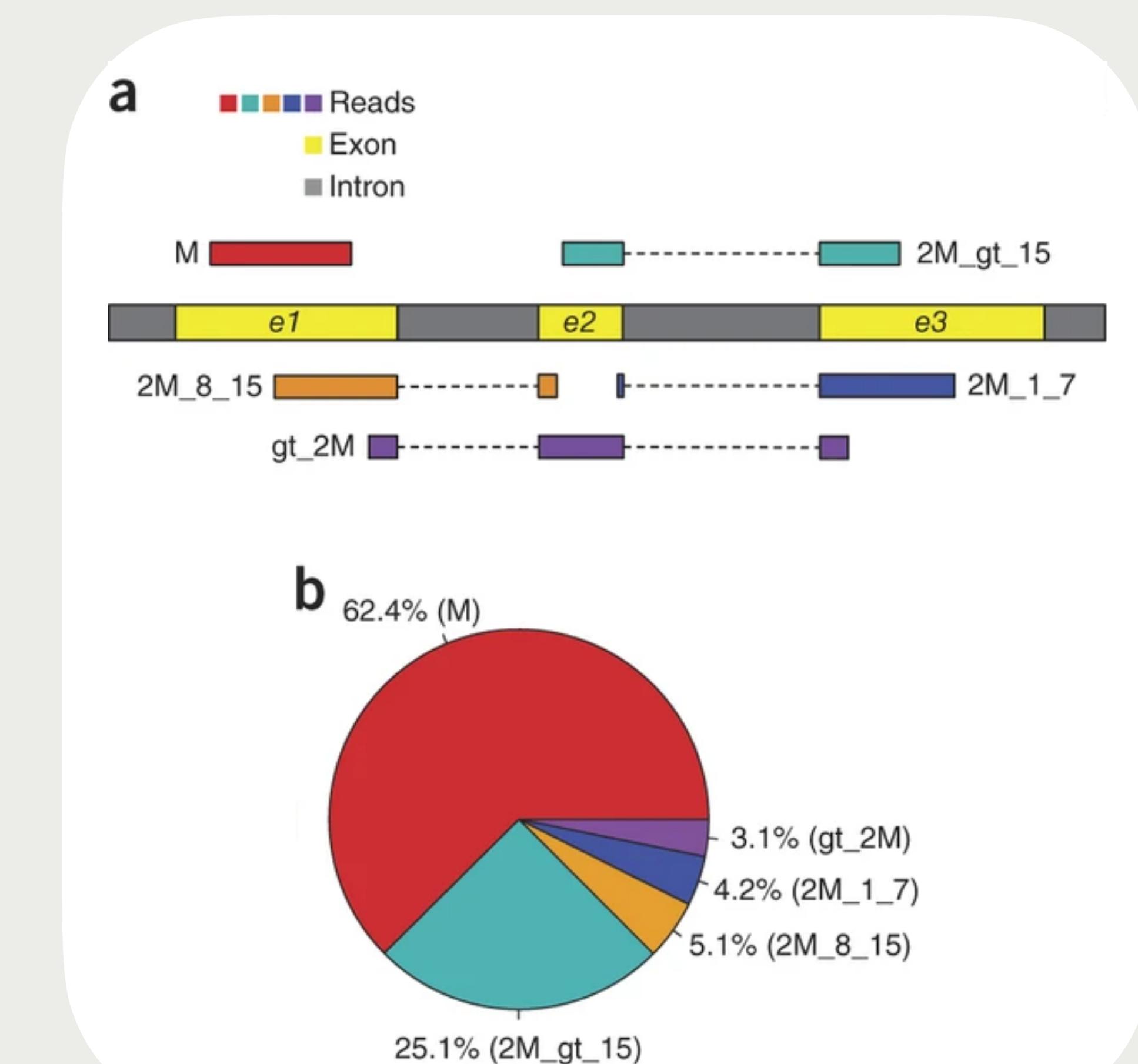
- How to make splice-aware?
- HISAT approach - two types of FM-indexes:
 - Global - entire genome
 - Local - 64,000 bp with 1,024 bp overlaps (~48,000 for the human genome)
 - 90% of human introns fit within a single local index
 - 8 bp will map ~1 time in a local index
- 4 GB total space



Two (main) approaches to splice-aware alignment

HISAT

1. Align part of each to the global FM-index
2. Align remainder of the read to a smaller 64,000 bp index (presumably contains the next exon)
 - Depending on length of match of Alignment #1, adaptive strategy with different algorithms
 - All include global search, local search, and read extension

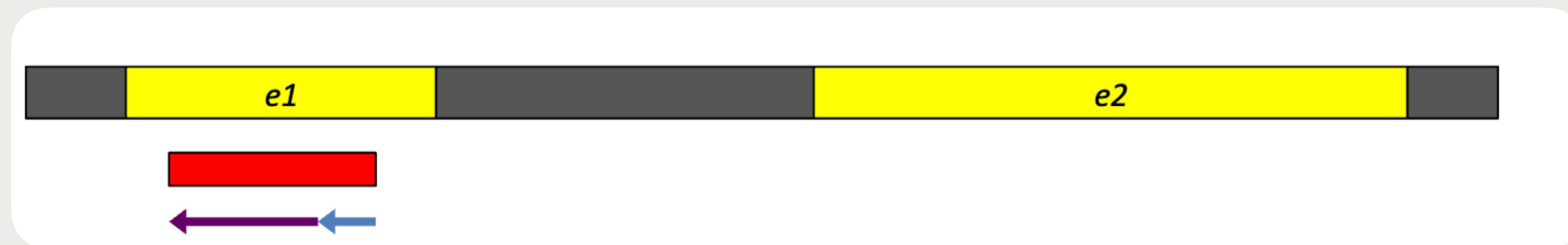


Two (main) approaches to splice-aware alignment

HISAT

Simple scenario - entire read fits within one exon

1. Align read to global FM-index until 28 nt at one location are exactly matching (slow)



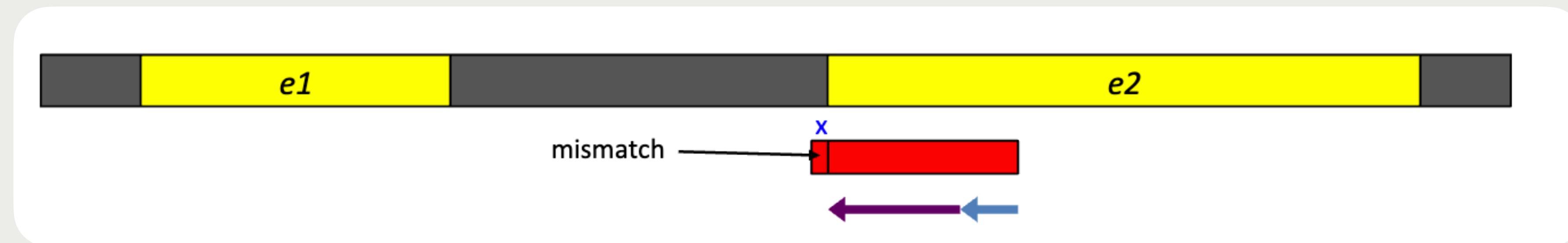
2. Switch to extension mode for the remainder of the read (very fast)

Two (main) approaches to splice-aware alignment

HISAT

Complex scenario - most of read fits in one exon

1. Align read to global FM-index until 28 nt at one location are exactly matching (slow)

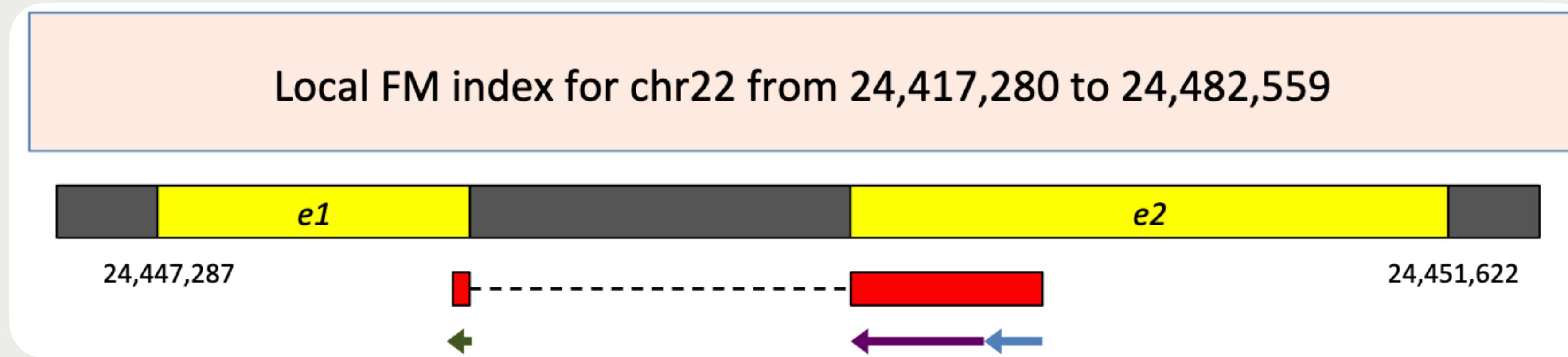


2. Switch to extension mode until reaching a mismatch (very fast)

Two (main) approaches to splice-aware alignment

Complex scenario - most of read fits in one exon

1. Align read to global FM-index until 28 nt at one location are exactly matching (slow)



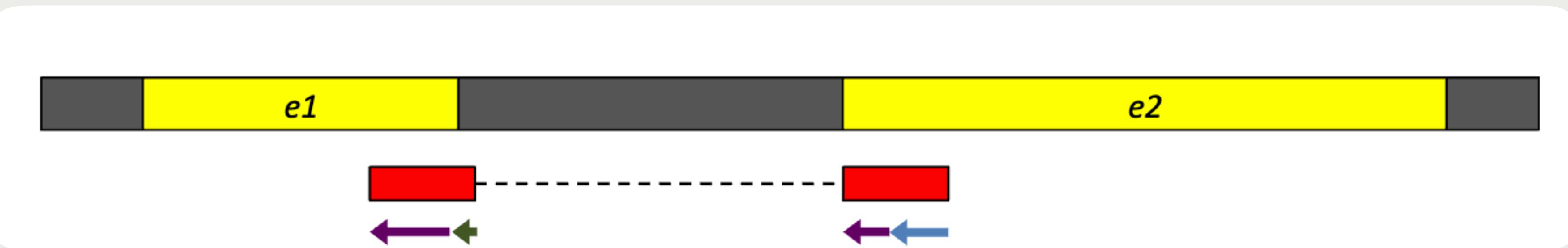
2. Switch to extension mode until reaching a mismatch
(very fast)
 3. Align remainder of read to local FM-index

Two (main) approaches to splice-aware alignment

HISAT

Complex scenario - long anchors in two exons

1. Align read to global FM-index until 28 nt at one location are exactly matching (slow)



2. Switch to extension mode until reaching a mismatch (very fast)
3. Align 8 bp of unaligned portion of read to local FM-index (longer prefix if too many potential local alignments)
4. Switch to extension mode

Two (main) approaches to splice-aware alignment

STAR and HISAT

STAR

Requires a lot of RAM; ultra fast

Soft clips

Few parameters

HISAT

Less RAM needed; still fast

Soft clips

Some hard coded parameters (local index size \approx max intron length; local search seed correlated with expected matches in local index)

If you have the RAM...go with STAR