Mapping Reads

Purpose of Read Mapping

- Identify the origin of a read
- RNA-seq: Infer the gene that was expressed and that generated the read

- BUT: Is it always possible to find the read origin?

Sequence Alignment

- Optimization problem: Find the alignment with the **highest score**
- Global alignment (end-to-end): Needleman-Wunsch
- Local alignment: Smith-Waterman

Score matrix:

	a	c	g	t
a	+2	-2	-1	-2
c	-2	+2	-2	-1
g	-1	-2	+2	-2
t	-2	-1	-2	+2

Gap score: -3

```
c  c  t  a  g  t  t  c  a  t
      |  |  x  |  x  |  |
t  g  t  a  a  t  -  c  a  c
      +2 +2 -1 +2 -3 +2 +2
```

Alignment score = 6

Local alignments (Smith-Waterman)

- Generally useful for local alignments, determining regions of similarity between two strings (here, DNA sequence)
- Dynamic programming based algorithm: gives optimal alignment, with respect to scoring system
- Need to set scores for **match**, penalties for **mismatch** and **gap** (typically, mismatch penalties are set according to evolutionary knowledge and error profiles)
- Match: +1; Mismatch: -1; Gap: -2

Reference Sequence

	.	A	A	T	G	T
.	0	0	0	0	0	0
A	0					
T	0					
G	0					
A	0					
C	0					

$$H(i,j) = \max \begin{Bmatrix} 0 & \\ H(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1,j) + w(a_i, -) & \text{Deletion} \\ H(i,j-1) + w(-, b_j) & \text{Insertion} \end{Bmatrix}, \ 1 \le i \le m, 1 \le j \le n$$

http://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm

Local alignments (Smith-Waterman)

- Generally useful for local alignments, determining regions of similarity between two strings (here, DNA sequence)
- Dynamic programming based algorithm: gives optimal alignment, with respect to scoring system
- Need to set scores for **match**, penalties for **mismatch** and **gap** (typically, mismatch penalties are set according to evolutionary knowledge and error profiles)
- Match: +1; Mismatch: -1; Gap: -2

	.	A	A	T	G	T
.	0	0	0	0	0	0
A	0	1	1			
T	0					
G	0					
A	0					
C	0					

$$H(i,j) = \max \begin{Bmatrix} 0 & \\ H(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1,j) + w(a_i, -) & \text{Deletion} \\ H(i,j-1) + w(-, b_j) & \text{Insertion} \end{Bmatrix}, \ 1 \le i \le m, 1 \le j \le n$$

http://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm

Local alignments (Smith-Waterman)

- Generally useful for local alignments, determining regions of similarity between two strings (here, DNA sequence)
- Dynamic programming based algorithm: gives optimal alignment, with respect to scoring system
- Need to set scores for **match**, penalties for **mismatch** and **gap** (typically, mismatch penalties are set according to evolutionary knowledge and error profiles)
- Match: +1; Mismatch: -1; Gap: -2

	.	A	A	T	G	T
.	0	0	0	0	0	0
A	0	1	1	0	0	0
T	0	0	0	2	0	1
G	0					
A	0					
C	0					

$$H(i,j) = \max \begin{Bmatrix} 0 & \\ H(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1,j) + w(a_i, -) & \text{Deletion} \\ H(i,j-1) + w(-, b_j) & \text{Insertion} \end{Bmatrix}, \ 1 \le i \le m, 1 \le j \le n$$

http://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm

Local alignments (Smith-Waterman)

- Generally useful for local alignments, determining regions of similarity between two strings (here, DNA sequence)
- Dynamic programming based algorithm: gives optimal alignment, with respect to scoring system
- Need to set scores for **match**, penalties for **mismatch** and **gap** (typically, mismatch penalties are set according to evolutionary knowledge and error profiles)
- Match: +1; Mismatch: -1; Gap: -2

	.	A	A	T	G	T
.	0	0	0	0	0	0
A	0	1	1	0	0	0
T	0	0	0	2	0	1
G	0	0	0	0	3	1
A	0	1	1	0	1	2
C	0	0	0	0	0	0

$$H(i,j) = \max \begin{Bmatrix} 0 & \\ H(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1,j) + w(a_i, -) & \text{Deletion} \\ H(i,j-1) + w(-, b_j) & \text{Insertion} \end{Bmatrix}, \quad 1 \le i \le m, 1 \le j \le n$$

http://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm

Local alignments (Smith-Waterman)

- Traceback:
- Start with maximum score (here, 3)
- Follow path that gives maximum score

	.	A	A	T	G	T
.	0	0	0	0	0	0
A	0	1	1	0	0	0
T	0	0	0	2	0	1
G	0	0	0	0	3	1
A	0	1	1	0	1	2
C	0	0	0	0	0	0

$$H(i,j) = \max \begin{cases} 0 & \\ H(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1,j) + w(a_i, -) & \text{Deletion} \\ H(i,j-1) + w(-, b_j) & \text{Insertion} \end{cases}, \quad 1 \le i \le m, 1 \le j \le n$$

http://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm

Local alignments (Smith-Waterman)

- Traceback:
- Start with maximum score (here, 3)
- Follow path that gives multiple score

- This gives:
- AATGT-
- -ATGAC

	.	A	A	T	G	T
.	0	0	0	0	0	0
A	0	1	1	0	0	0
T	0	0	0	2	0	1
G	0	0	0	0	3	1
A	0	1	1	0	1	2
C	0	0	0	0	0	0

$$H(i,j) = \max \begin{cases} 0 & \\ H(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1,j) + w(a_i, -) & \text{Deletion} \\ H(i,j-1) + w(-, b_j) & \text{Insertion} \end{cases}, \quad 1 \le i \le m, 1 \le j \le n$$

http://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm

Local alignments (Smith-Waterman)

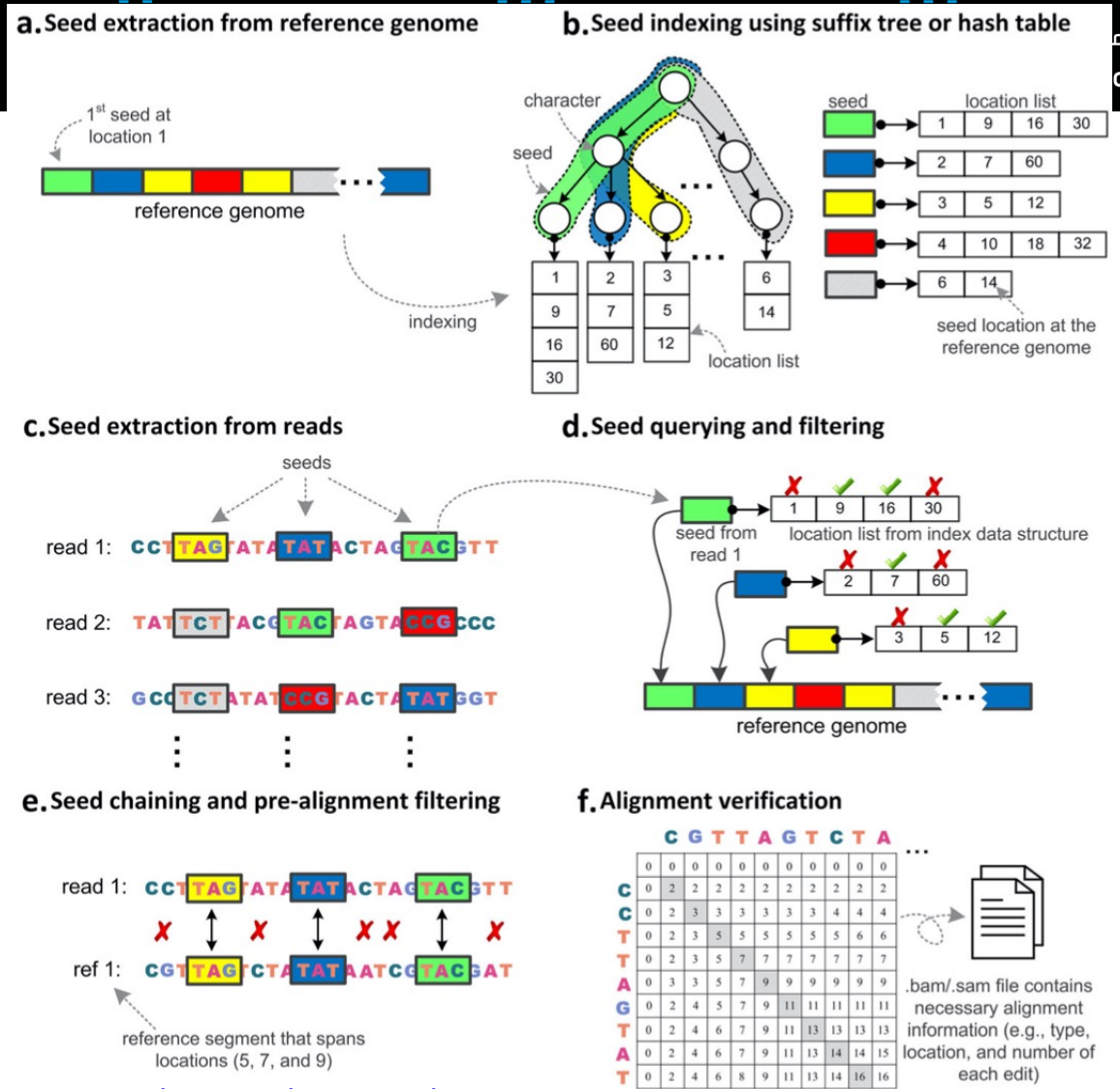
- Other considerations:
- Natural extension is to have a different *gap opening* and a *gap extension* penalty (former generally being larger)
- GO penalty=-2; GE penalty=-1; Mismatch=-1; Match=1
- With above penalties, which is the best scoring alignment?

AT-C-GT	ATC--GT	AT-C--GT
ATTTTGT	ATTTTGT	ATT-TTGT

Alignment Tools

- Dynamic programming is slow
- Speed-up with heuristics
  - e.g. exactly align short subsequences and extend these alignments
    - e.g. BLAST / BLAT
  - no longer guaranteed to find the best alignments
  - exact matches are found by index lookup

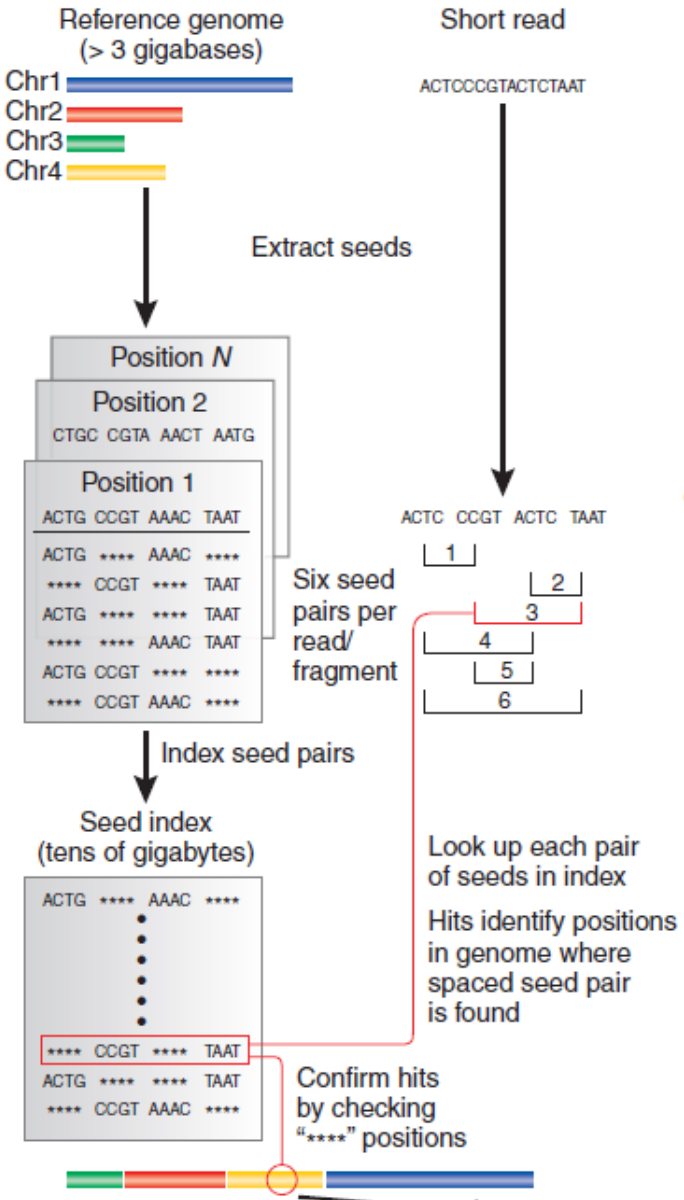
Hashing based alignment



https://genomebiology.biomedcentral.com/articles/10.1186/s13059-021-02443-7

Index Genome: Spaced Seeds

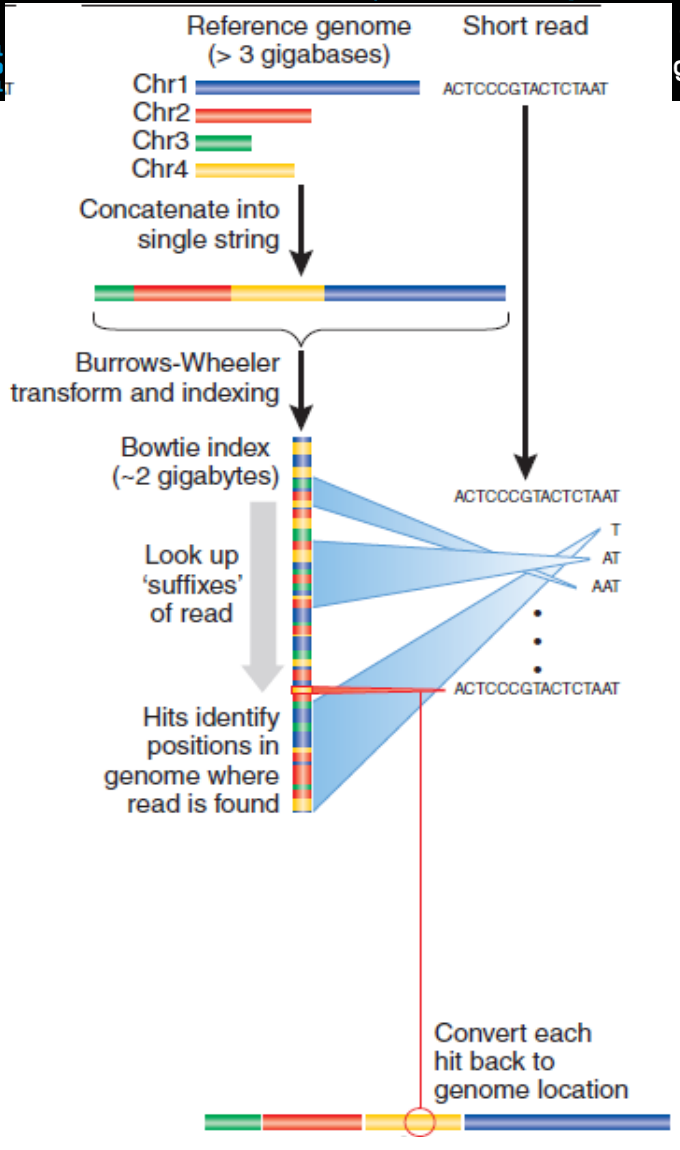
- Tags and tag-sized pieces of reference are cut into small "seeds."
- Pairs of spaced seeds are stored in an index.
- Look up spaced seeds for each tag.
- For each "hit," confirm the remaining positions.



Trapnell etal., Nature Biotechnology, 2009

Index Genome: Burrows-Wheeler Transform

- Store entire reference genome.
- Align tag base by base from the end.
- When tag is traversed, all active locations are reported.
- If no match is found, then back up and try a substitution.

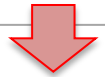


Trapnell etal., Nature Biotechnology, 2009

The Burrows-Wheeler transform (1994; 1983)

c a c a a c g $

c	a	c	a	a	c	g	$
a	c	a	a	c	g	$	c
c	a	a	c	g	$	c	a
a	a	c	g	$	c	a	c
a	c	g	$	c	a	c	a
c	g	$	c	a	c	a	a
g	$	c	a	c	a	a	c
$	c	a	c	a	a	c	g



$	c	a	c	a	a	c	g
a	a	c	g	$	c	a	c
a	c	a	a	c	g	$	c
a	c	g	$	c	a	c	a
c	a	a	c	g	$	c	a
c	a	c	a	a	c	g	$
c	g	$	c	a	c	a	a
g	$	c	a	c	a	a	c

g c c a a $ a c

https://www.geeksforgeeks.org/inverting-burrows-wheeler-transform/

The "Last-First mapping" property

The relative ordering of a particular character (say c) in column 1 is the same as that in the last column

c_1 a c_2 a a c_3 g $

$	c_1	a	c_2	a	a	c_3	g
a	a	c_3	g	$	c_1	a	c_2
a	c_2	a	a	c_3	g	$	c_1
a	c_3	g	$	c_1	a	c_2	a
c_2	a	a	c_3	g	$	c_1	a
c_1	a	c_2	a	a	c_3	g	$
c_3	g	$	c_1	a	c_2	a	a
g	$	c_1	a	c_2	a	a	c_3

The "Last-First mapping" property

The relative ordering of a particular character (say c) in column 1 is the same as that in the last column

c_1 a c_2 a a c_3 g $

$	c_1	a	c_2	a	a	c_3	g
a	a	c_3	g	$	c_1	a	c_2
a	c_2	a	a	c_3	g	$	c_1
a	c_3	g	$	c_1	a	c_2	a
c_2	a	a	c_3	g	$	c_1	a
c_1	a	c_2	a	a	c_3	g	$
c_3	g	$	c_1	a	c_2	a	a
g	$	c_1	a	c_2	a	a	c_3

Proof:

Suppose c X and c Y are cyclic permutations of the input T.
Suppose c X < c Y (in lexicographical ordering)
Then X c < Y c (in lexicographical ordering)
The LF-mapping property follows.

BWT is reversible

$	c	a	c	a	a	c	g
a	a	c	g	$	c	a	c
a	c	a	a	c	g	$	c
a	c	g	$	c	a	c	a
c	a	a	c	g	$	c	a
c	a	c	a	a	c	g	$
c	g	$	c	a	c	a	a
g	$	c	a	c	a	a	c

g c c a a $ a c

$ a a a c c c g

BWT is reversible

$	c	a	c	a	a	c	g
a	a	c	g	$	c	a	c
a	c	a	a	c	g	$	c
a	c	g	$	c	a	c	a
c	a	a	c	g	$	c	a
c	a	c	a	a	c	g	$
c	g	$	c	a	c	a	a
g	$	c	a	c	a	a	c

g$ ca ca aa ac $c ac cg

$c aa ac ac ca ca cg g$

BWT is reversible

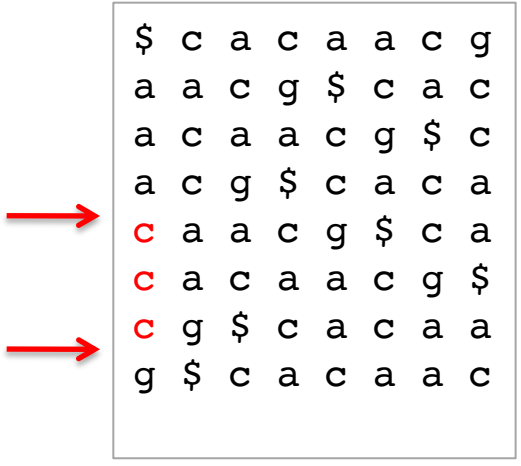
$	c	a	c	a	a	c	g
a	a	c	g	$	c	a	c
a	c	a	a	c	g	$	c
a	c	g	$	c	a	c	a
c	a	a	c	g	$	c	a
c	a	c	a	a	c	g	$
c	g	$	c	a	c	a	a
g	$	c	a	c	a	a	c

Lookup AAC

$	c	a	c	a	a	c	g
a	a	c	g	$	c	a	c
a	c	a	a	c	g	$	c
a	c	g	$	c	a	c	a
c	a	a	c	g	$	c	a
c	a	c	a	a	c	g	$
c	g	$	c	a	c	a	a
g	$	c	a	c	a	a	c

Range ← range of last character in 1st column
While characters left (and nonzero range):
    Lookup first and last match to preceding character in final column
    Range ← LF-mapping of first and last match

Lookup AAC



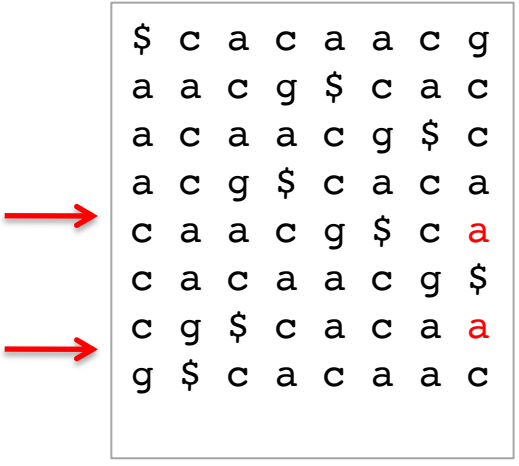
Range ← range of last character in 1st column

While characters left (and nonzero range):

Lookup first and last match to preceding character in final column

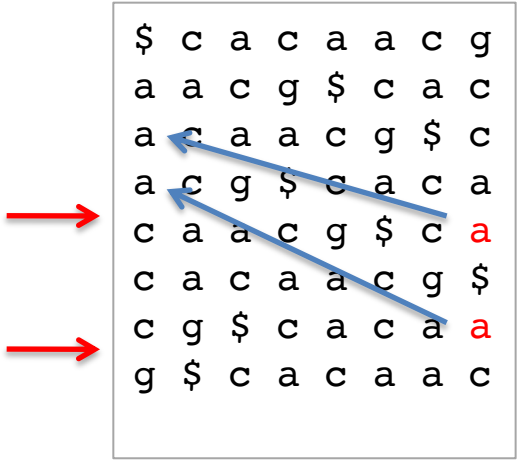
Range ← LF-mapping of first and last match

Lookup AAC



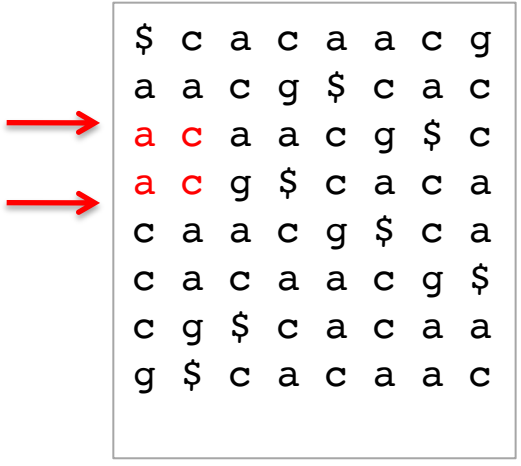
Range ← range of last character in 1st column
While characters left (and nonzero range):
    Lookup first and last match to preceding character in final column
    Range ← LF-mapping of first and last match

Lookup AAC



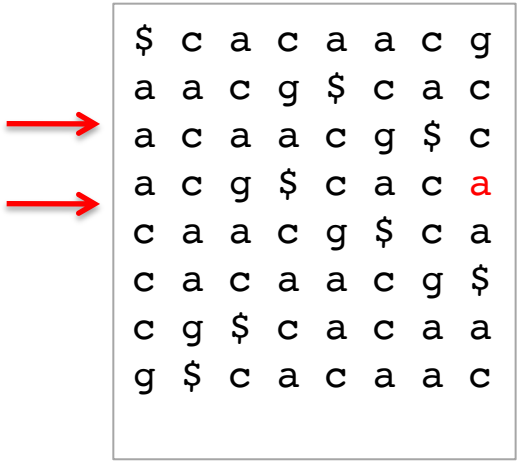
Range ← range of last character in 1st column
While characters left (and nonzero range):
    Lookup first and last match to preceding character in final column
    Range ← LF-mapping of first and last match

Lookup AAC



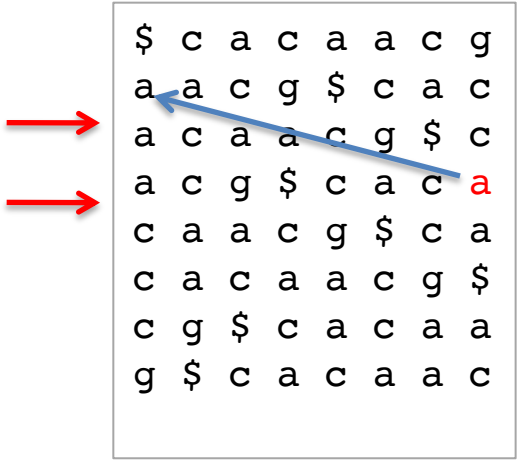
Range ← range of last character in 1st column
While characters left (and nonzero range):
    Lookup first and last match to preceding character in final column
    Range ← LF-mapping of first and last match

Lookup AAC



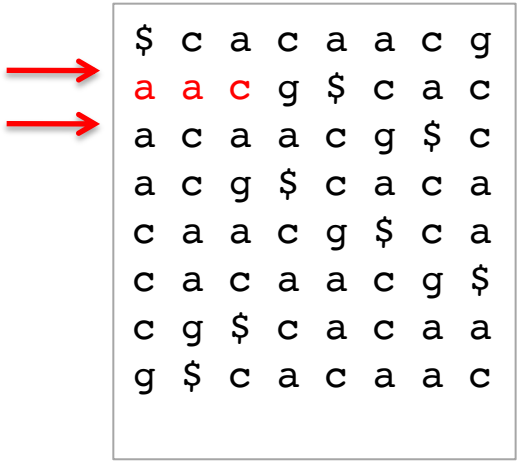
Range ← range of last character in 1st column
While characters left (and nonzero range):
    Lookup first and last match to preceding character in final column
    Range ← LF-mapping of first and last match

Lookup AAC



Range ← range of last character in 1st column
While characters left (and nonzero range):
    Lookup first and last match to preceding character in final column
    Range ← LF-mapping of first and last match

Lookup AAC



Range ← range of last character in 1st column
While characters left (and nonzero range):
    Lookup first and last match to preceding character in final column
    Range ← LF-mapping of first and last match

Comparison

Spaced seeds

- Depending on choices up to 50Gb of memory for humang genome
- More straightforward to program.
- Example:
  - Subread
- More tolerant to
  - sequence variations
  - sequencing errors

Burrows-Wheeler

- Compressed index requires <2Gb of memory for human genome.
- can be very fast
- More complicated to program.
- Examples:
  - bowtie2
  - BWA

Characteristics of algorithms

	Hashing	Suffix tree and BWT-FM
Easy to implement	Yes	No
Search for exact/inexact match	Exact	Exact and inexact
Index size	Large	Compressed (small)
Indexing time	Small	Large
Seed query speed	O(1), fast	Slow
Seed length	Fixed length per index	Can be fixed or variable

Alser et al.

https://genomebiology.biomedcentral.com/articles/10.1186/s13059-021-02443-7

Combining index lookup and pairwise alignment

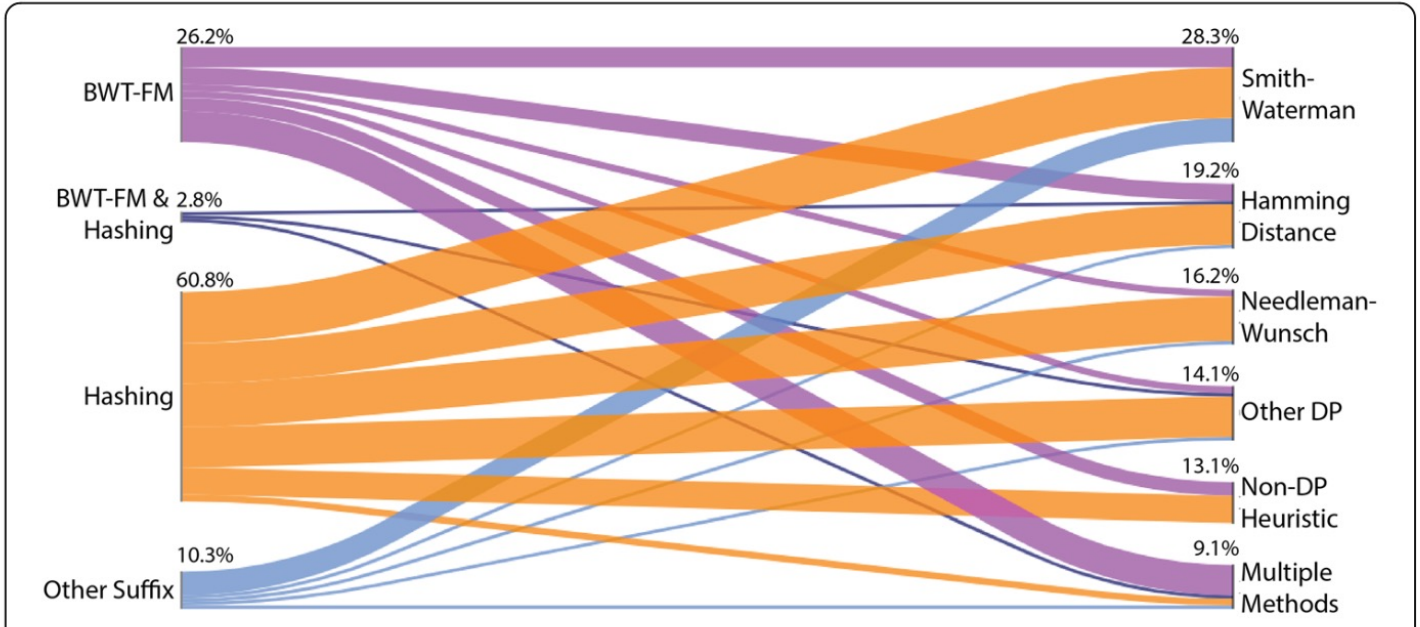
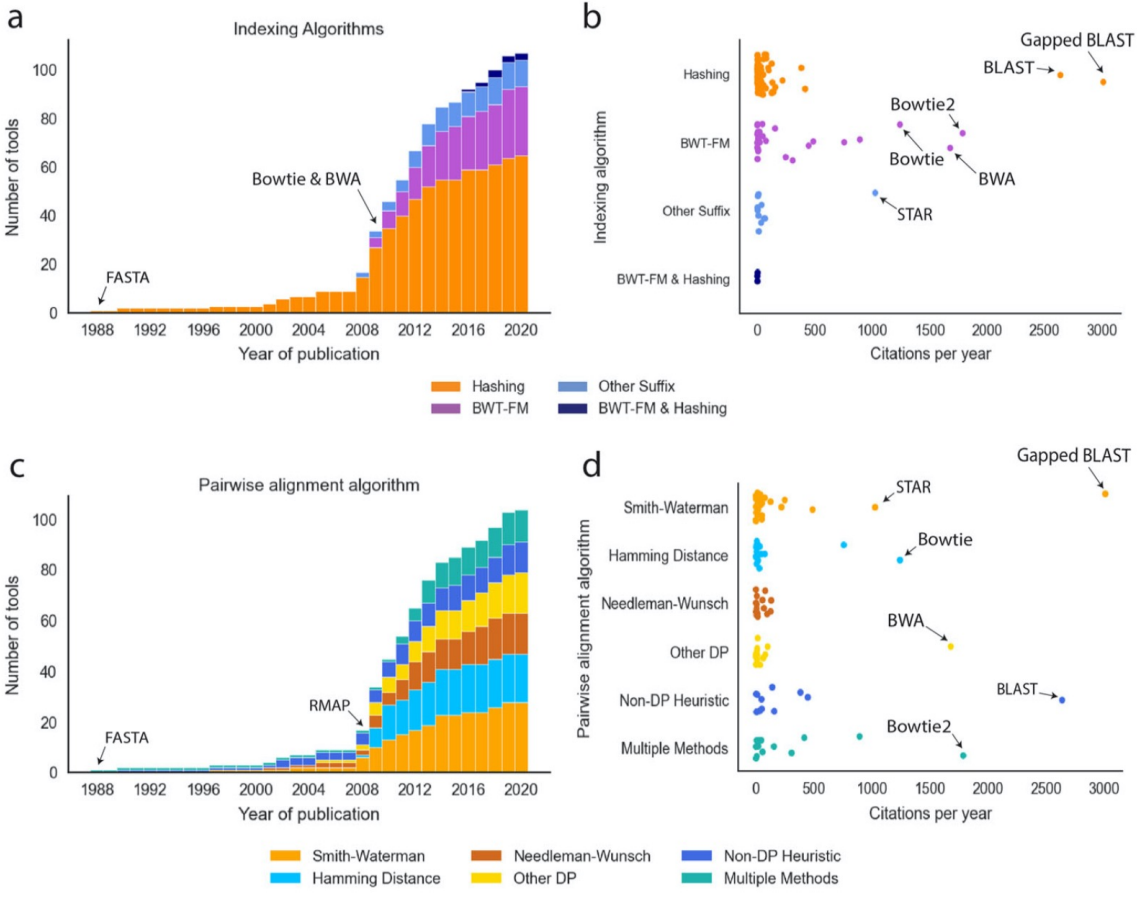


Fig. 2 Combination of algorithms utilized by read alignment tools. Sankey plot displaying the flow of surveyed tools using each indexing technique and pairwise alignment. For every indexing technique, the percentage of surveyed tools using the algorithm is displayed (BWT-FM 26.2%, BWT-FM, and Hashing 2.8%, Hashing 60.8%, Other Suffix 10.3%). For every pairwise alignment technique, the percentage of surveyed tools using the algorithm is displayed (Smith-Waterman 28.3%, Hamming distance 19.2%, Needleman-Wunsch 16.2%, Other DP 14.1%, Non-DP Heuristic 13.1%, Multiple Methods 9.1%)



Alser et al. 2021, Genome Biology

Alignment with Mismatches

- Mismatches can occur because of
  - sequencing error (error rate ~1/500)
  - mutation; (human mutation rate ~1/1e4)
  - → if reads are long (> 100nt) reads with mismatches will not be rare; more than ~10% of the reads may have a mismatch
- If there is a sequence mismatch the index lookup fails! How to find nevertheless an alignment?

BWT Alignment with Mismatches

- Strategies:
  - If the BWT lookup fails at position k, try all different bases at position k
    → drawback: computing effort grows exponentially with the number of mismatches
    → implemented e.g. in bowtie

  - Chop reads in segments (seeds) and align those mismatch-free and stitch seed alignments together
    → implemented e.g. in bowtie

Consideration: Sequencing Errors – PHRED quality

- Each called base is given a quality score Q
- Q = 10 * log10(estimated probability that call is wrong)

  10 prob = 0.1

  20 prob = 0.01

  30 prob = 0.001

  [Q30 often used as a threshold for useful sequence data]

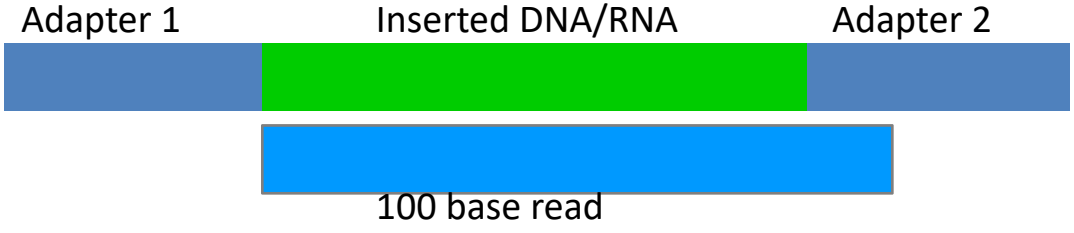
  → down-weight low-quality bases when computing the alignment score
- These quality scores are called PHRED scores.
- PHRED scores are determined by the sequencer that directly rates how reliable the measured signal is

Considerations: Read trimming

- Sequencers have systematic errors
- Illumina sequencers have a higher error rate at the first few bases (1-5)
- Basically, all sequencers have increasing error rate towards the end of the read

- Hard trimming: trim a fixed number of bases from the beginning and/or end
- Quality trimming: cut the end of the read as soon as the base quality drops below a threshold

Considerations: Read trimming



- If the inserted DNA/RNA fragment is too short, the read will contain part of the adapter
- Adapter trimming can be challenging if
  - if the insert is 90 – 100 bases
  - if the read has many sequencing errors

Multiple Alignments

- A read may have multiple valid alignments with identical or similarly good alignment scores
- Aligners allow to choose different reporting strategies:
  - Randomly select one alignment from the top-scoring alignments
  - Report all alignments that are within *delta* of the top-scoring alignment; clip if more than *Nmax* alignments are found
  - Report only alignments if they are unique (no other alignment within *delta* of the alignment score)
  - Do not report anything if more than *Nmax* valid alignments are found
  - …
- Whether a read has a **unique alignment** depends on
  - the read sequence and the sequence homology of the organism
  - the search algorithm of the aligner
  - the reporting strategy of the aligner

Read Alignment Summary (I)

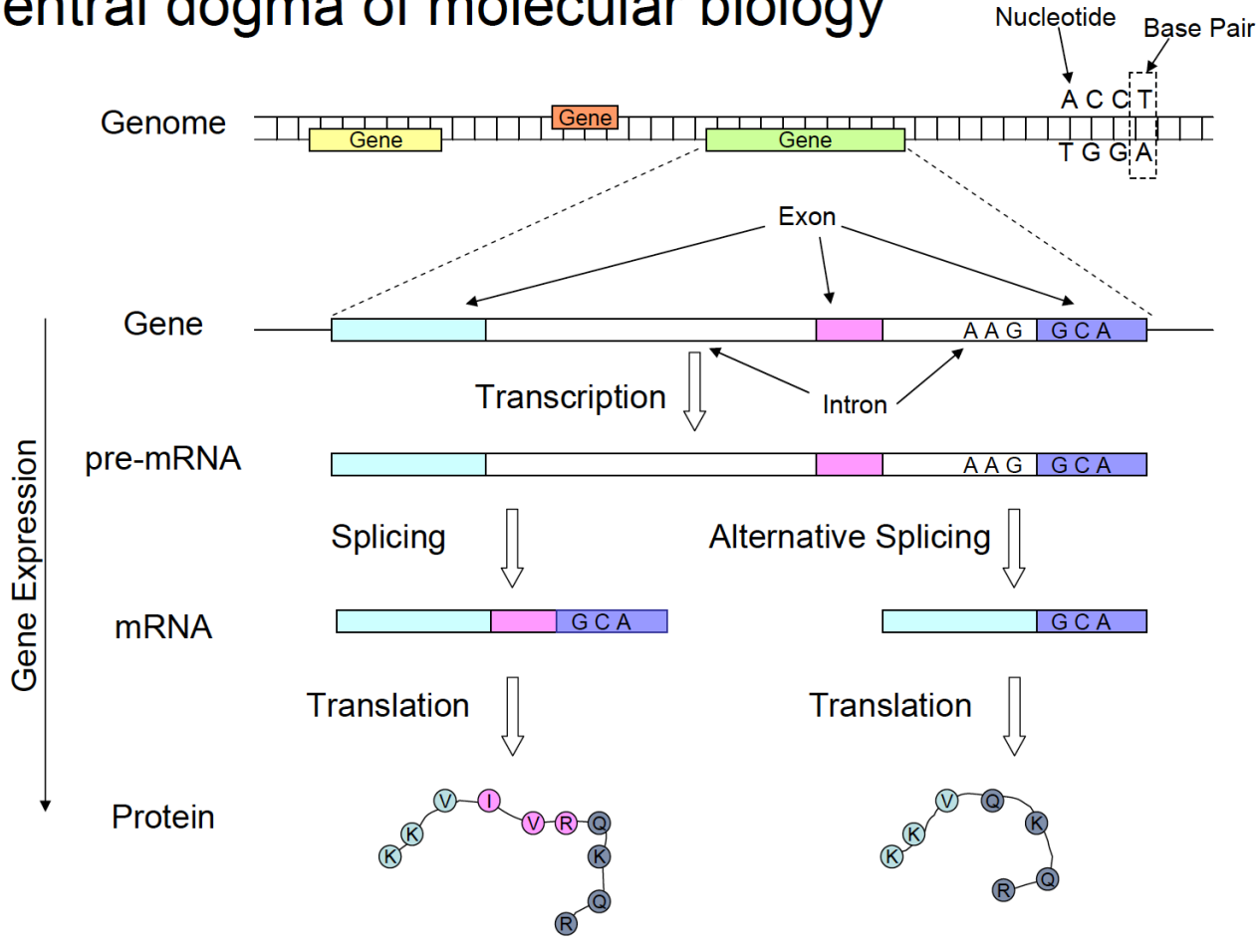
How to map billions of reads?

- The major aligners use the Burrows-Wheeler-Transform to create an index of the reference. Even for the human genome the index fits into 3GB RAM.
- Reads are aligned by **index lookup** not by sequence comparison
- The lookup of a perfect match read is faster than loading the read and writing the alignment coordinates to the output file!
- If the lookup fails because of SNPs or sequencing errors, an actual sequence comparison is done –and this can take time!

Read Alignment Summary (II)

- Alignment is a score optimization problem
- Aligners find the alignment with the **highest alignment score**
- Aligners do not run an extensive search, they use **index lookup and heuristics** to find the alignment with highest score;
  these heuristics are not guaranteed to find the alignment with the maximum score
- If the heuristics are well chosen and the scores are well defined, then the alignment will be very often the highest-scoring alignment and that will be the correct alignment

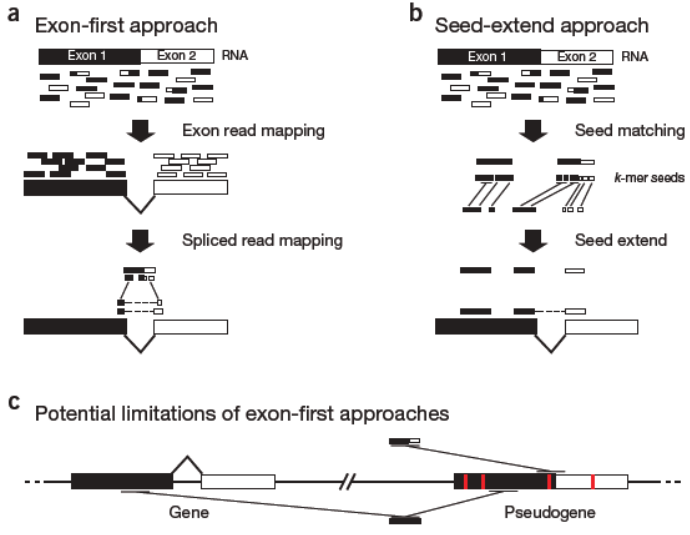
Central dogma of molecular biology



92–94% of human genes undergo alternative splicing,

RNA-seq Mapping

- Mapping targets:
  - transcriptome
  - splice junction library
  - genome
- Mouse retina 60+60bp reads:
  - 41 Mio of 91 Mio map to junctions



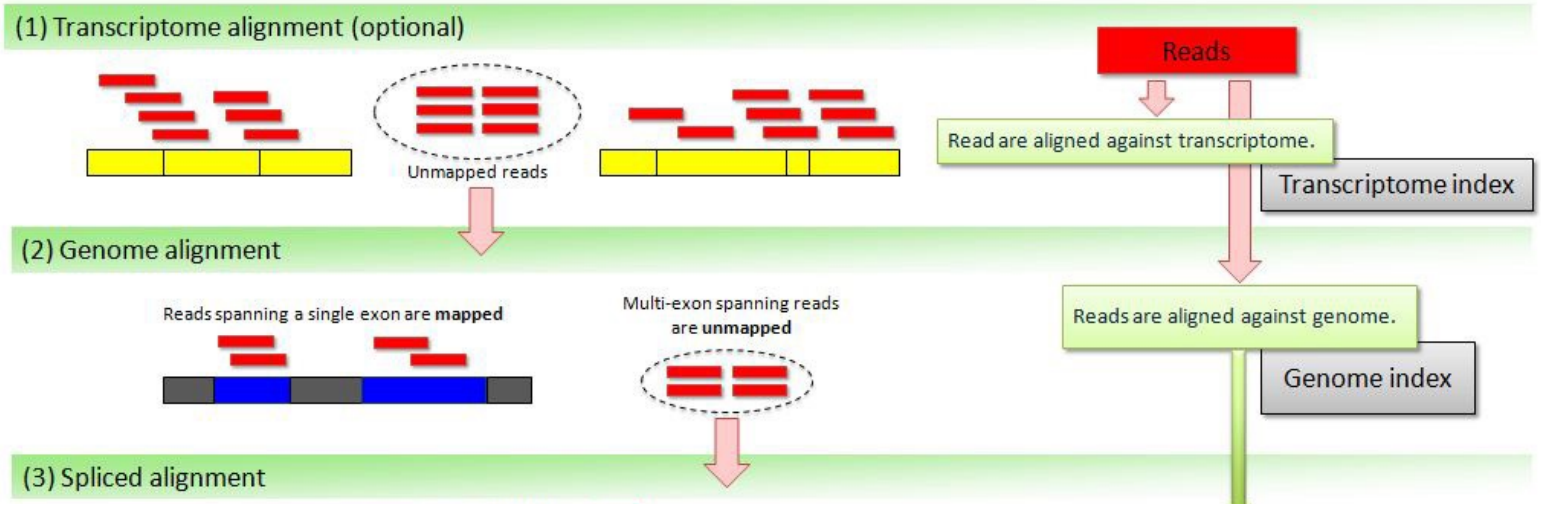
Strategies:

- Exon-first: fast
- Seed-extend:
  - good with polymorphisms
  - simultaneous spliced/unspliced mapping

Garber etal., Nature Methods, 2011

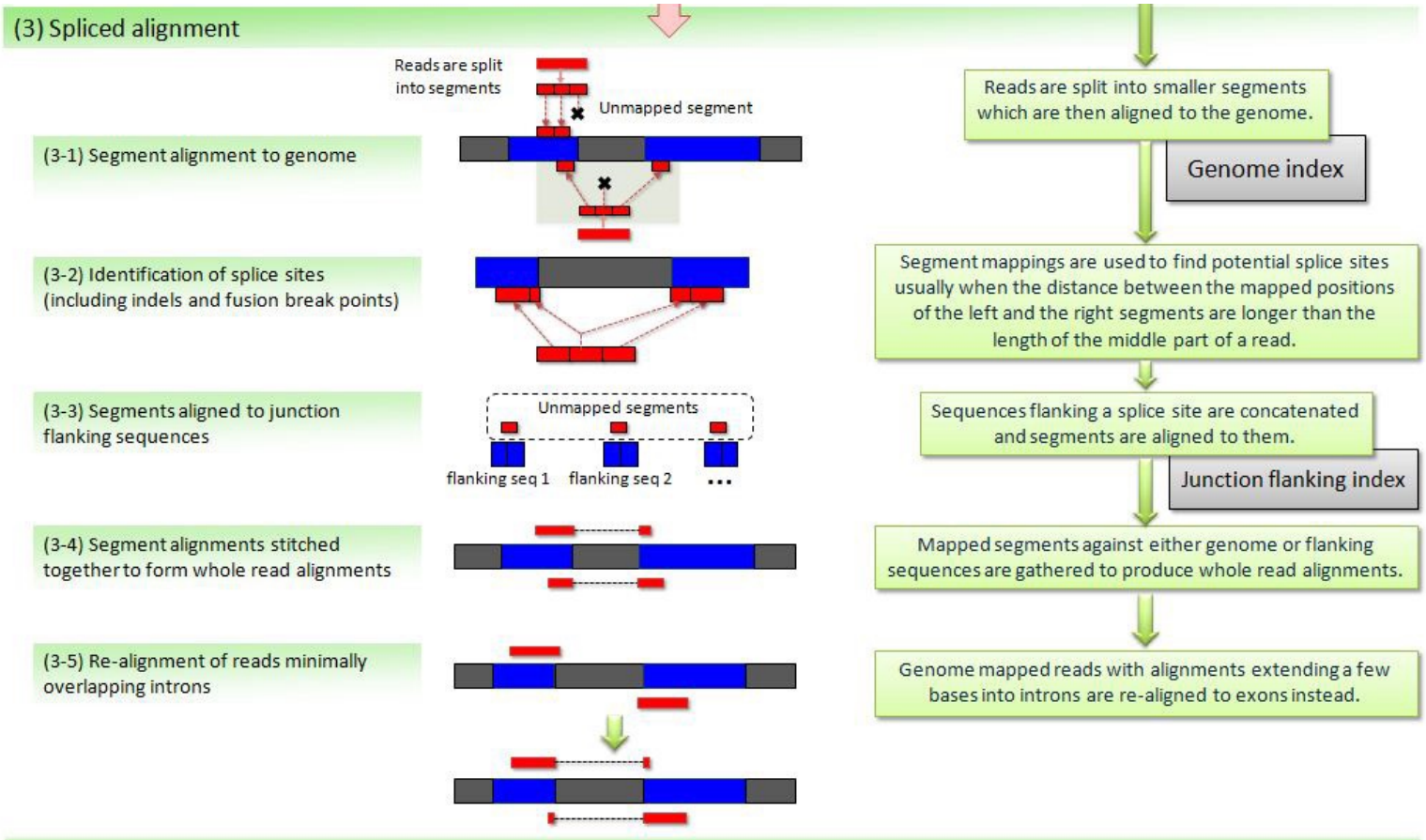
Tophat2 pipeline – illustrative of the challenge of mapping RNA-seq reads

- Note: Tophat2 is outdated, use STAR, HISAT, Subread, …



Kim et al. Genome Biology

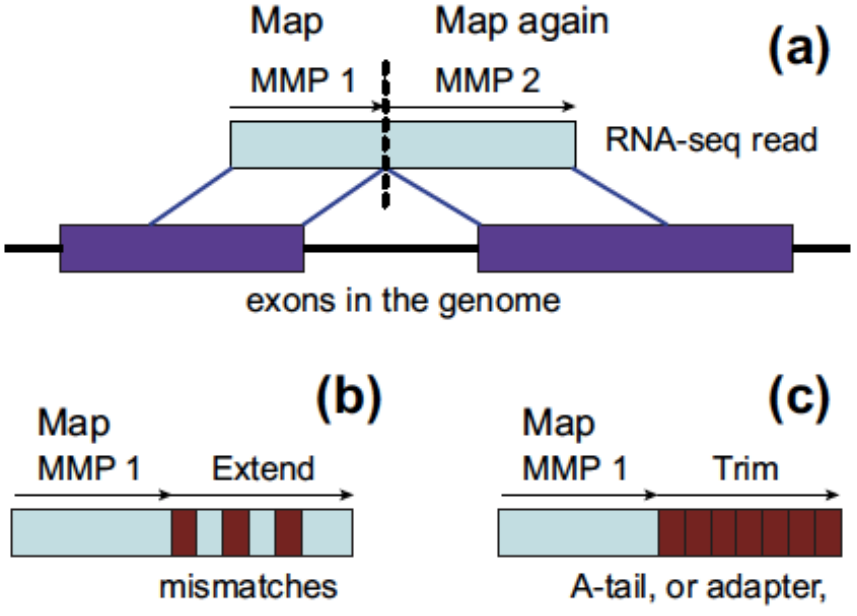
Tophat2 pipeline



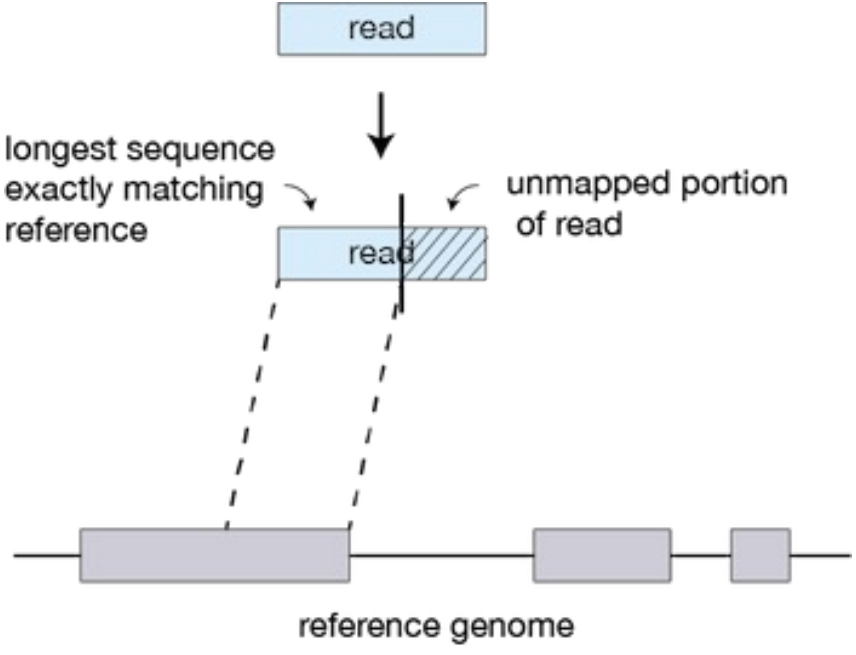
Kim et al. Genome Biology 2013

STAR: universal RNA-seq aligner

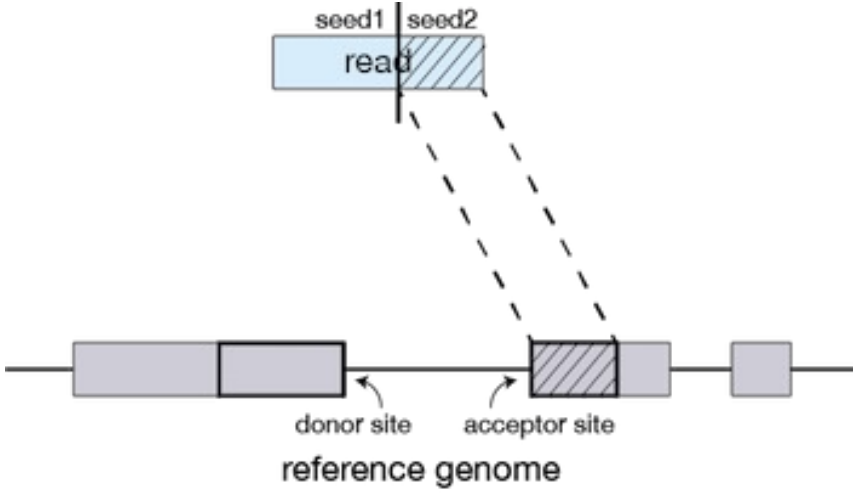
- Designed to align the non-contiguous sequences directly to the reference genome
- Steps:
  - Search Maximal Mappable Prefix (MMP)
  - clustering/stitching/scoring



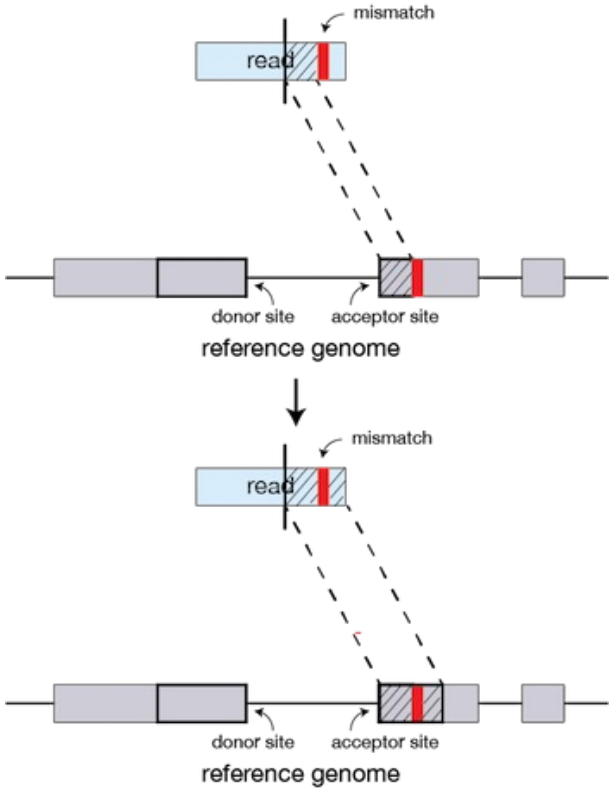
Seed lookup



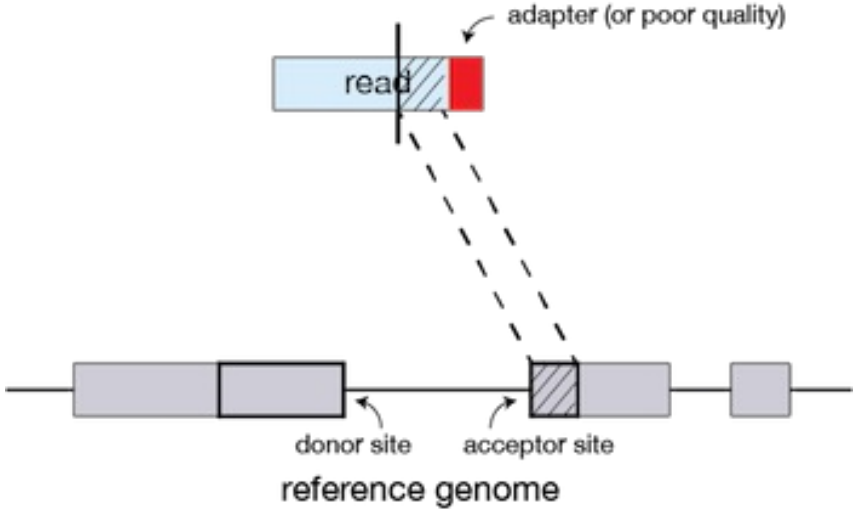
Reseed



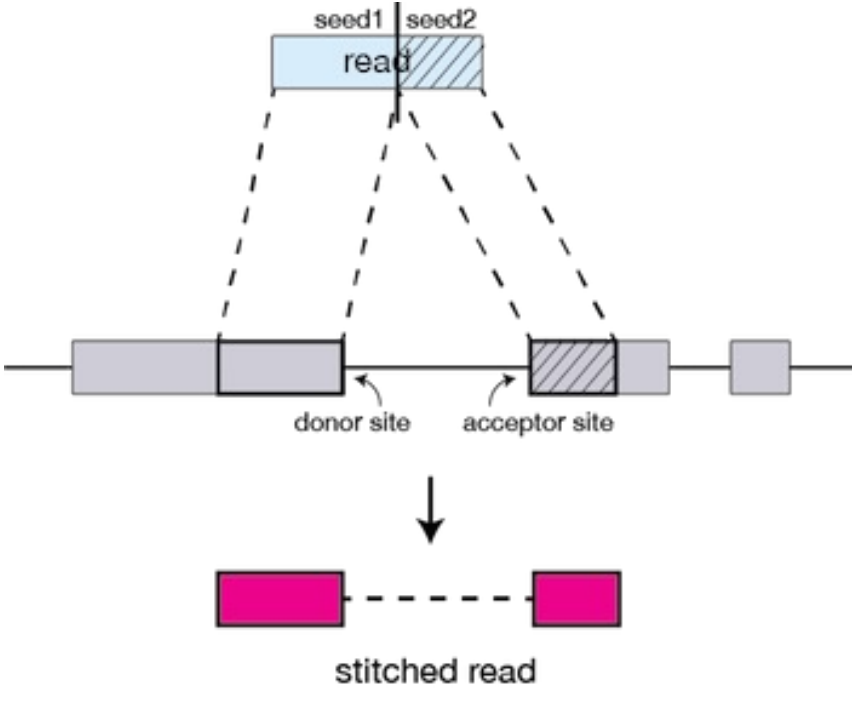
Handle mismatches



Soft clip ends



Stitch spliced



Comparison of Aligners

- Tradeoff:
  - speed
  - RAM
  - accuracy
  - sensitivity

- http://www.ecseq.com/support/benchmark

Performance comparison of RNA-seq mappers

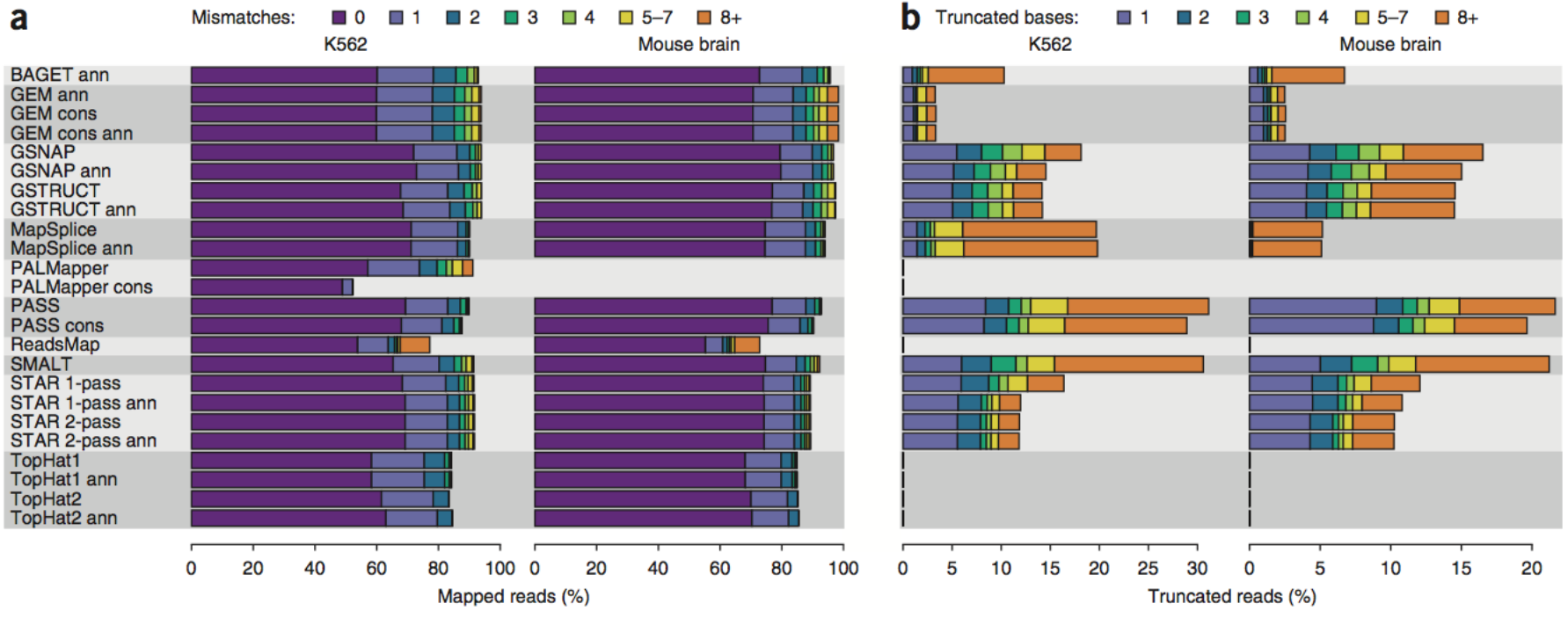


Figure 2 | Mismatch and truncation frequencies. (**a**) Percentage of sequenced reads mapped with the indicated number of mismatches. (**b**) Percentage of sequenced reads truncated at either or both ends. Bar colors indicate the number of bases removed.

Engström et al. Systematic evaluation of spliced alignment programs for RNA-seq data. Nat Methods. 2013.

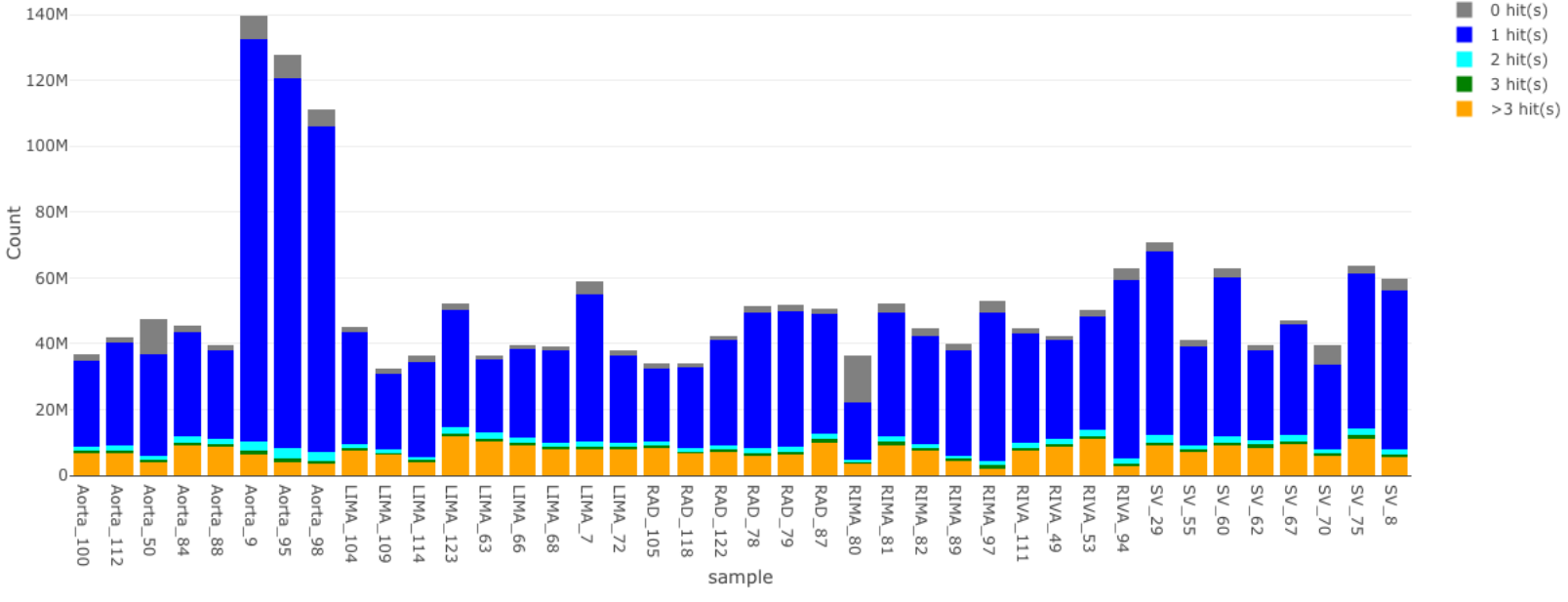
STAR 2-pass mode

- goal: improve alignment around novel splice junctions
- does not discover more splice junctions but aligns more reads to discovered splice junctions
- initial mapping might fail to align reads where only a short part of the read spans the splice junction

How long should a read be so that it can be uniquely aligned??

Current practice

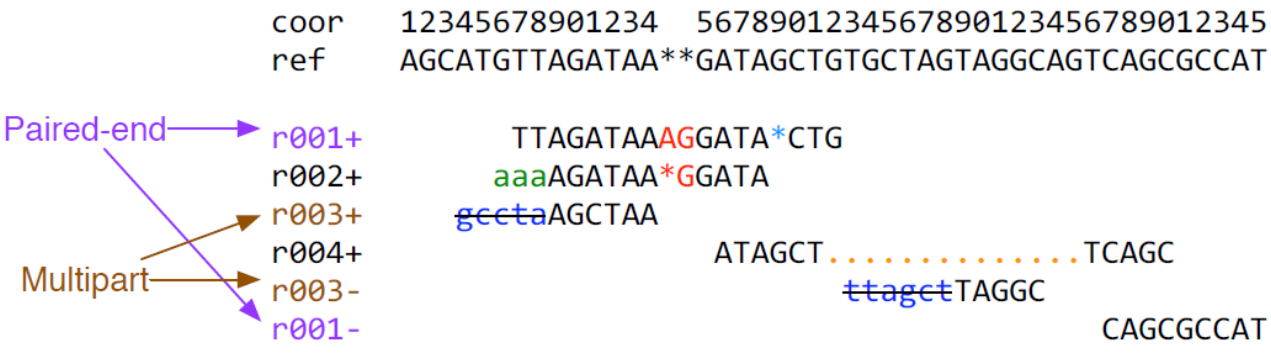
- use >= 75nt for aligning RNA-reads to mammals
- use >= 100+100nt for aligning DNA reads for variant calling
- use 21nt for aligning micro-RNA reads to miRbase



Sequence / Alignment (SAM) files

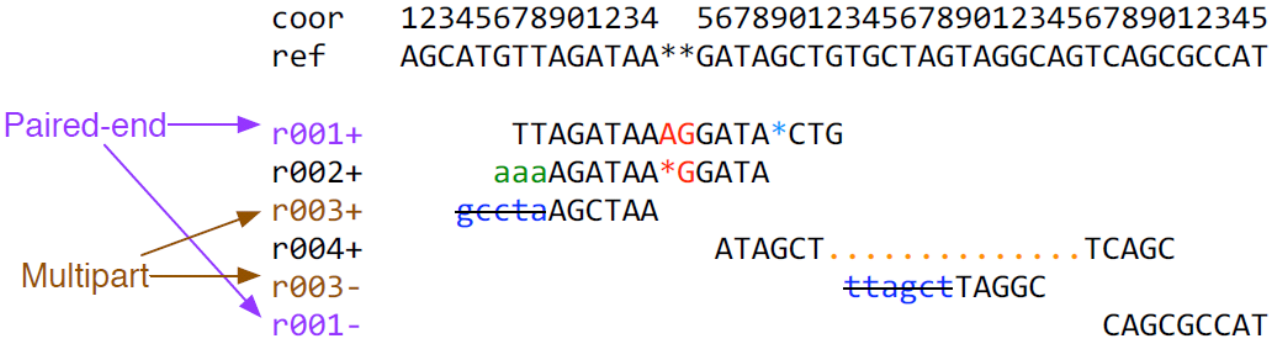
- **SAM (Sequence Alignment/Map)**
  - Single unified format for storing read alignments to a reference genome
  - Large plain text file
- **BAM (Binary Alignment/Map)**
  - Binary equivalent of SAM
  - Compressed data plus index (bai)
  - Developed for fast processing/indexing

An example of read mapping



Slide modified from Heng Li's presentation

Corresponding SAM file



Ins & padding
Soft clipping
Splicing
Hard clipping

```
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTA *
r002   0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA     *
r003   0 ref  9 30 5H6M       *  0   0 AGCTAA    *   NM:i:1
r004   0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003  16 ref 29 30 6H5M       *  0   0 TAGGC     *   NM:i:0
r001  83 ref 37 30 9M         =  7 -39 CAGCGCCAT         *
```

QNAME RNAME MAPQ? MRNM ISIZE QUAL
FLAG POS CIGAR? MPOS SEQ

Slide modified from Heng Li's presentation

CIGAR string - compact representation of an alignment

- M - match or mismatch
- I – insertion
- D – deletion
- S - soft clip
  - Clipped sequences stored in SAM
- H - hard clip
  - Clipped sequences not stored in SAM
- N – skipped reference bases, splicing

```
Match/mismatch,indels

Ref:   ACGCAGTG--GT
Read:  ATGCA-TGCAGT
Cigar:5M1D2M2I2M
```

```
Soft clipping

REF:   ATCGTGTAACCTGACTAGTTAA
READ:  gggGTGTAACC-GACTAGgggg
Cigar:  3S8M1D5M4S
```

```
Hard clipping

REF:   ATCGTGTAACCTGACTAGTTAA
READ:  gggGTGTAACC-GACTAGgggg
Cigar:  3H8M1D5M4H
```

Mapping quality – the MAPQ tag

- The MAPQ tag does indicate the quality of an alignment
- Equivalent to the PHRED score for base

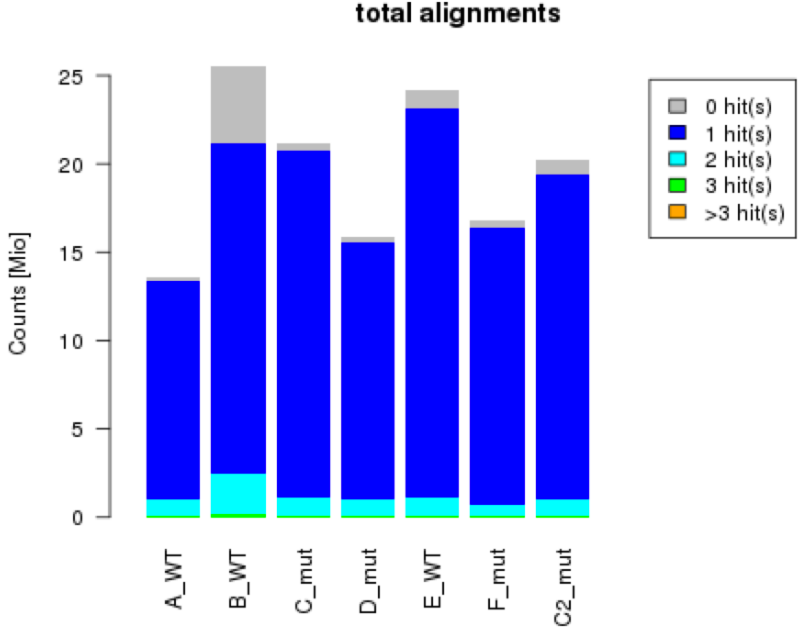
- However – the error probability can not be as accurately determined as the base quality

- The different aligners have different methods for assigning the quality

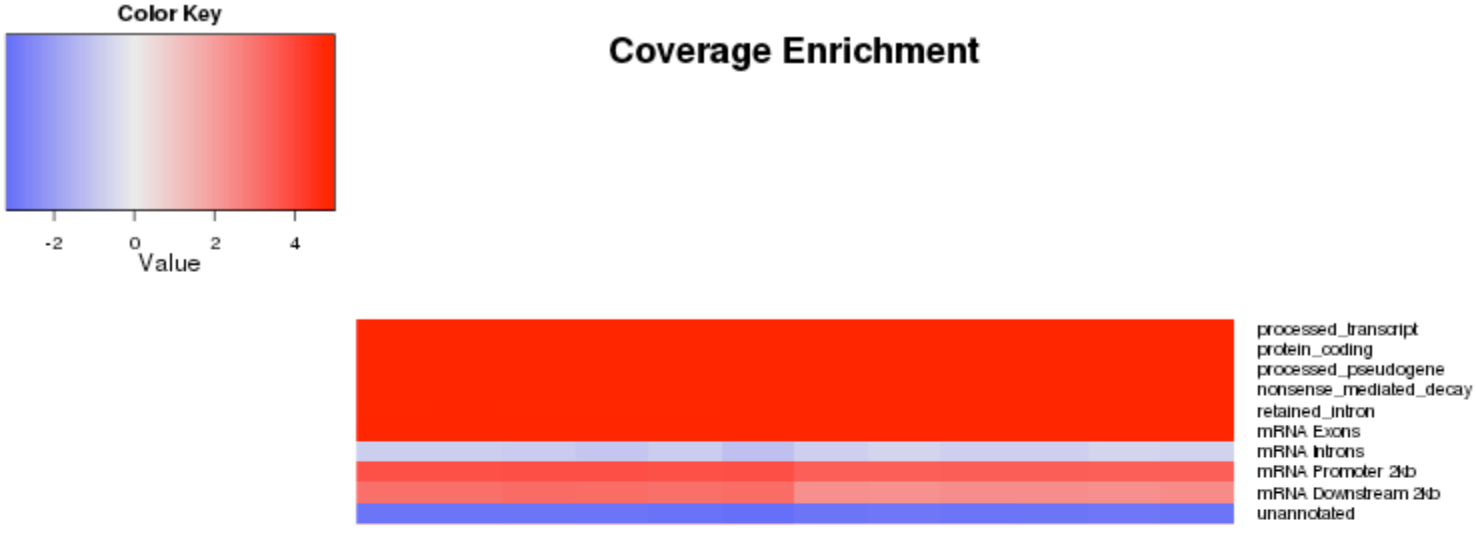
Mapping QC: Mapping statistics

- How well did my sequence library align to my reference?
- Summary statistics (per read library)
  - % reads with unique alignment
  - % reads with multiple alignmen
  - % reads with no alignment
  - % reads properly paired

  (for paired end libraries)



Mapping QC: Profiling efficiency by transcript annotation



- Transcript annotation: intron, exon, up/down stream, unannotated
- Expression profiling efficiency

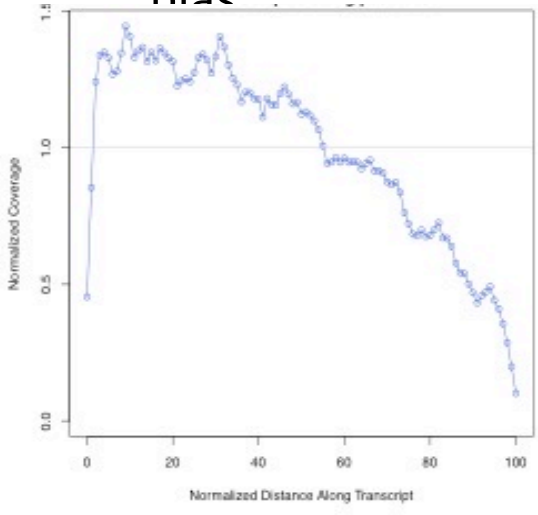
Mapping QC: Coverage bias

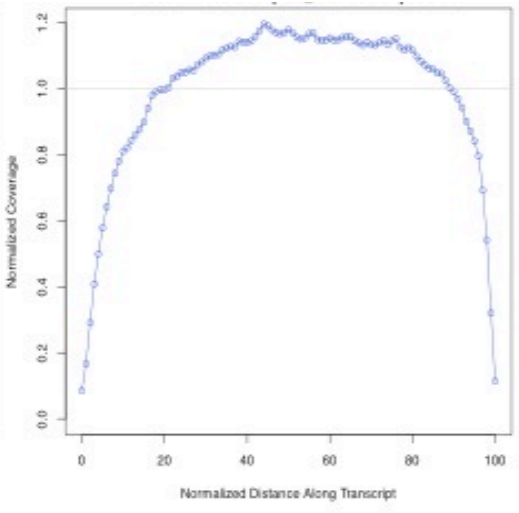
3’ bias

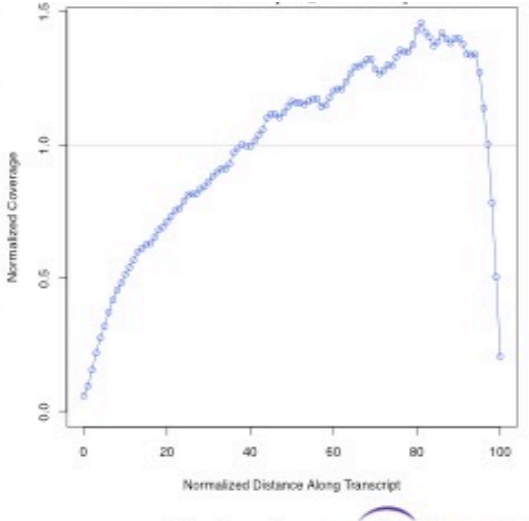
bias

No bias

5’

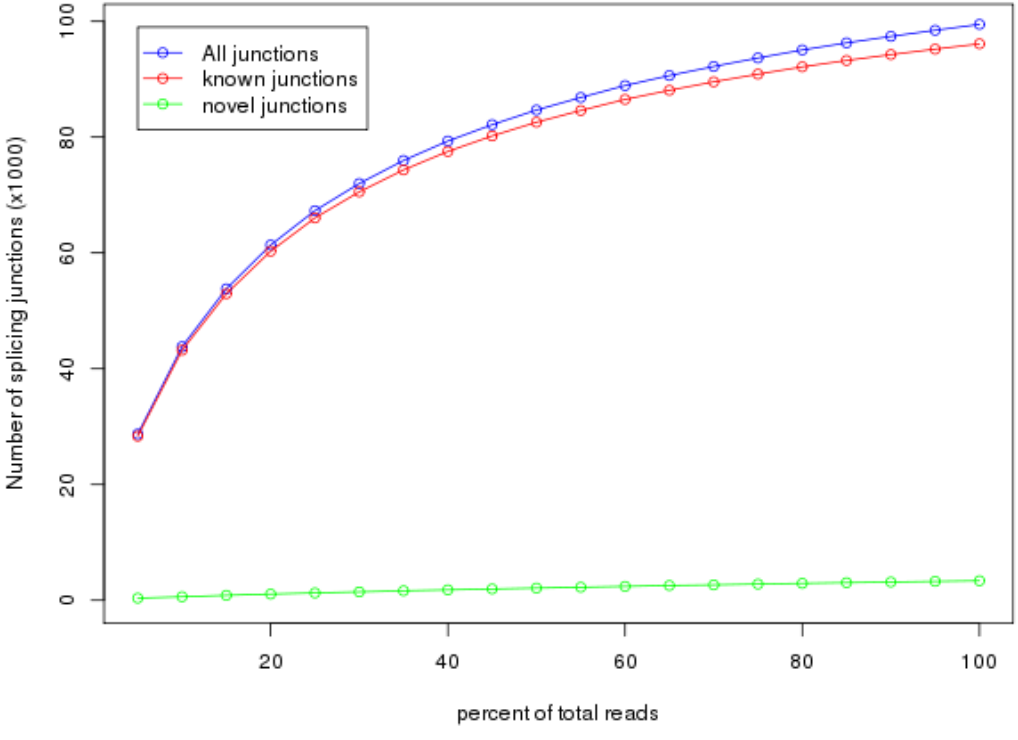






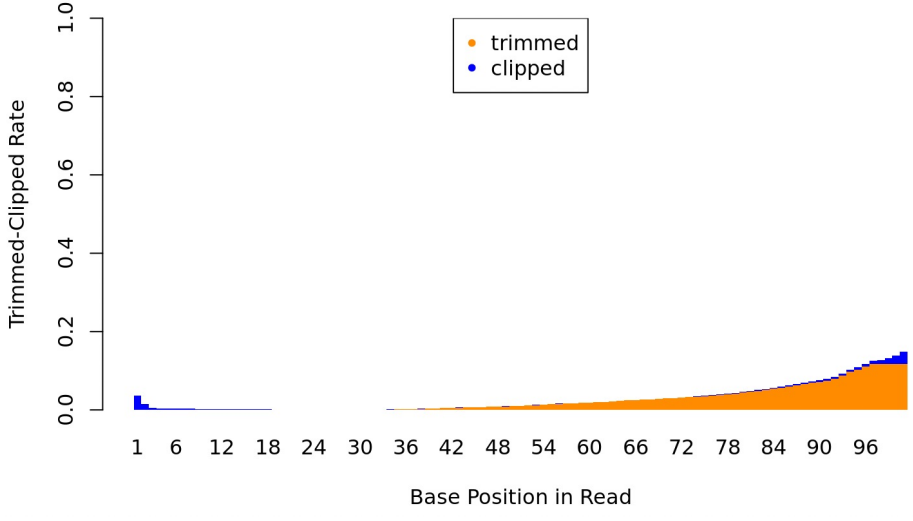
Zuojian Tang NYU Langone MEDICAL CENTER

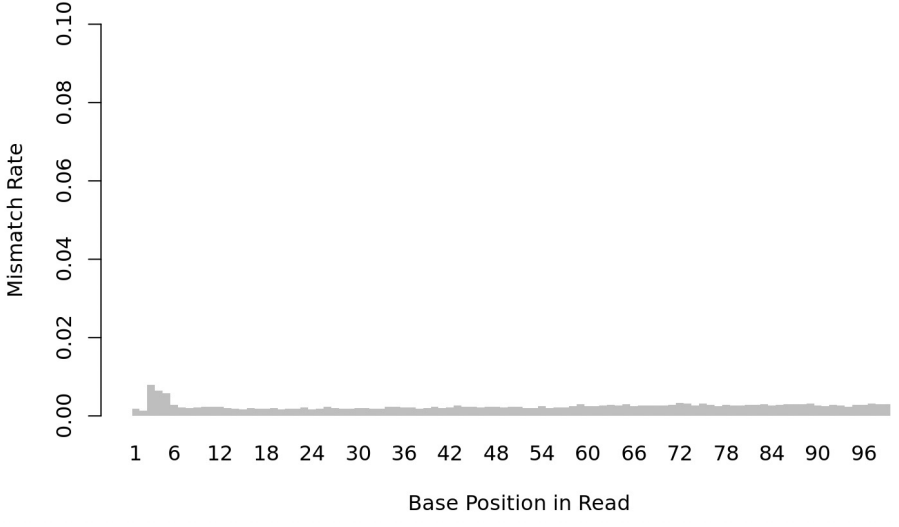
Mapping QC: Junction saturation



- Depth needed for alternative splicing analysis
- All annotated splice junctions are detected - a saturated RNA-seq dataset

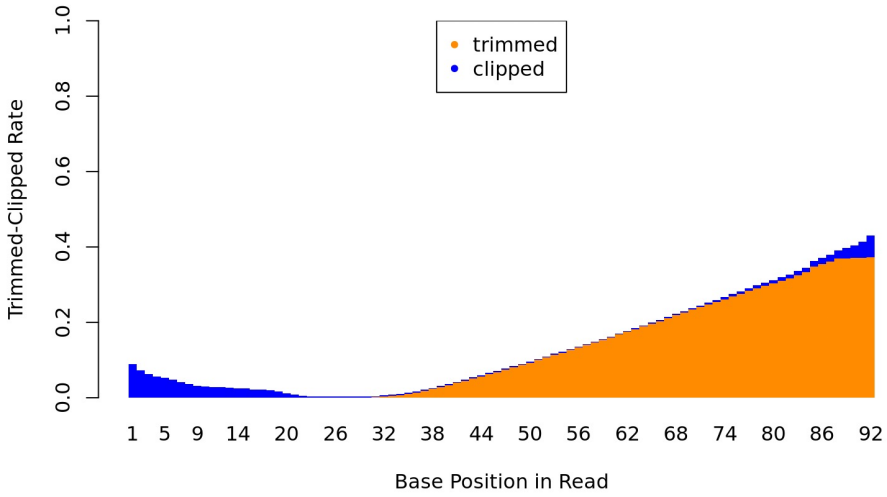
Statistics on Trimming, Soft Clipping and Mismatches

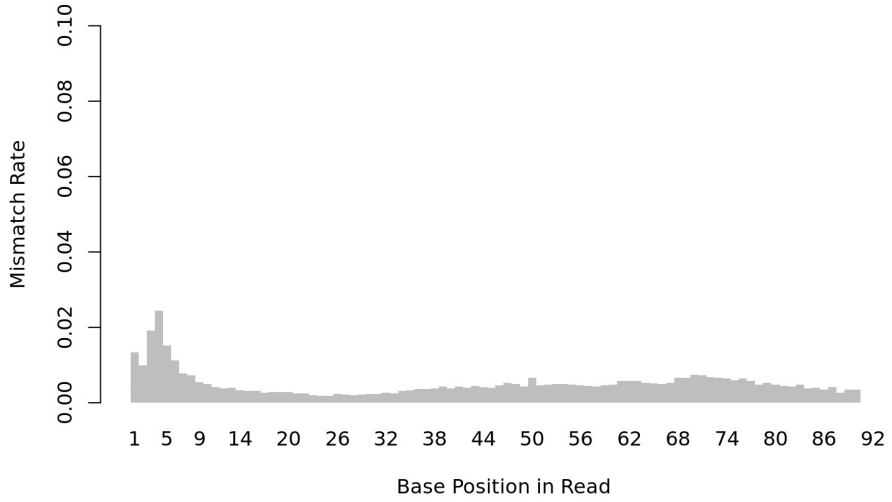




- Mismatches depend on position in the read
- If bases 1 or 2 do not match, the STAR aligner soft-clips them

Statistics on Trimming, Soft Clipping and Mismatches





- Example from a low quality sample (low input <1ng, slight degradation)
- Short inserts, increased mismatch rate

Where mappers have issues

- paralog genes that exis on X and Y chromosomes
- MHC genes (part of the adaptive immune response)
  - poly-morphic: several alleles (versions) exist on that gene's locus

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6007662/

RNA-seq mapping QC tools

- RNA-SeQC
  - https://github.com/getzlab/rnaseqc
- RSeQC
  - http://rseqc.sourceforge.net/