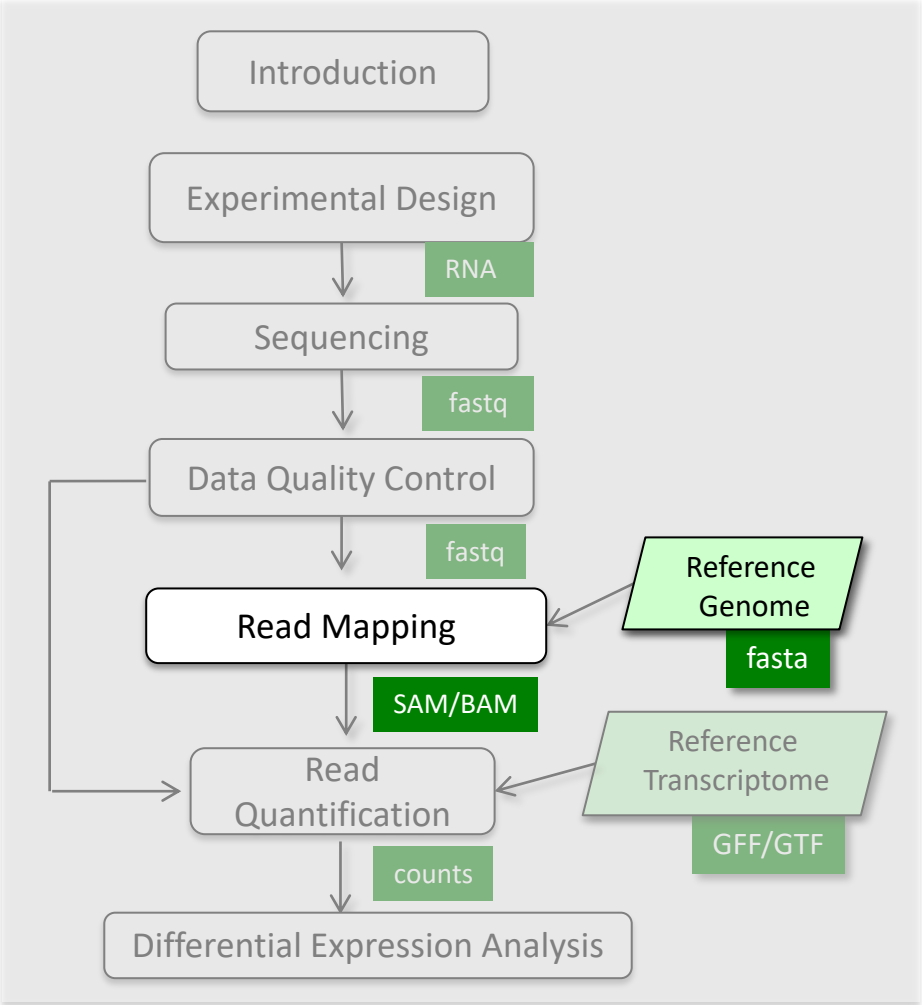
RNA-seq Bioinformatics:
Read Mapping

Falko Noé

University of Zurich[UZH]
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Checking in (Clicker)



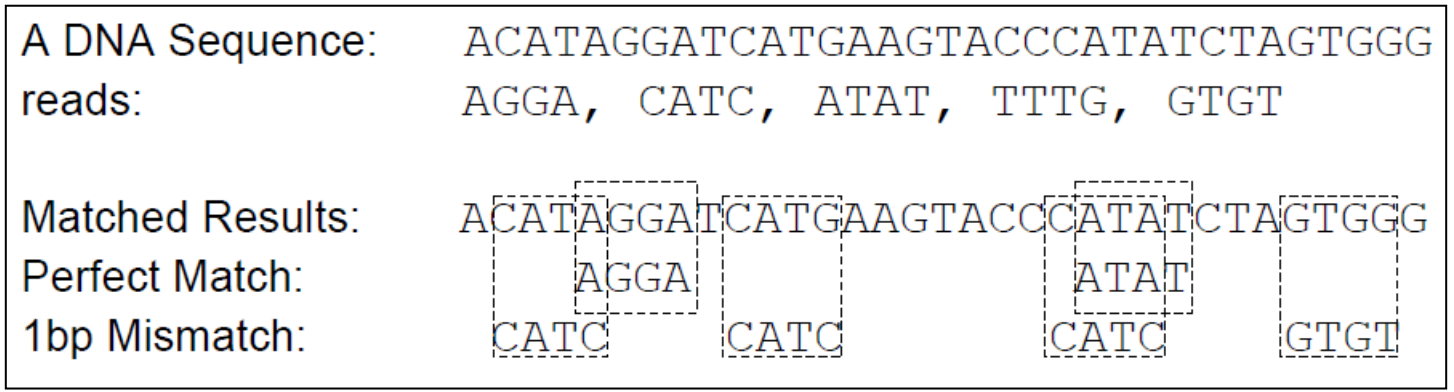
Read Mapping

- General idea
- Different strategies
- File formats

- Quality control
- Visualization

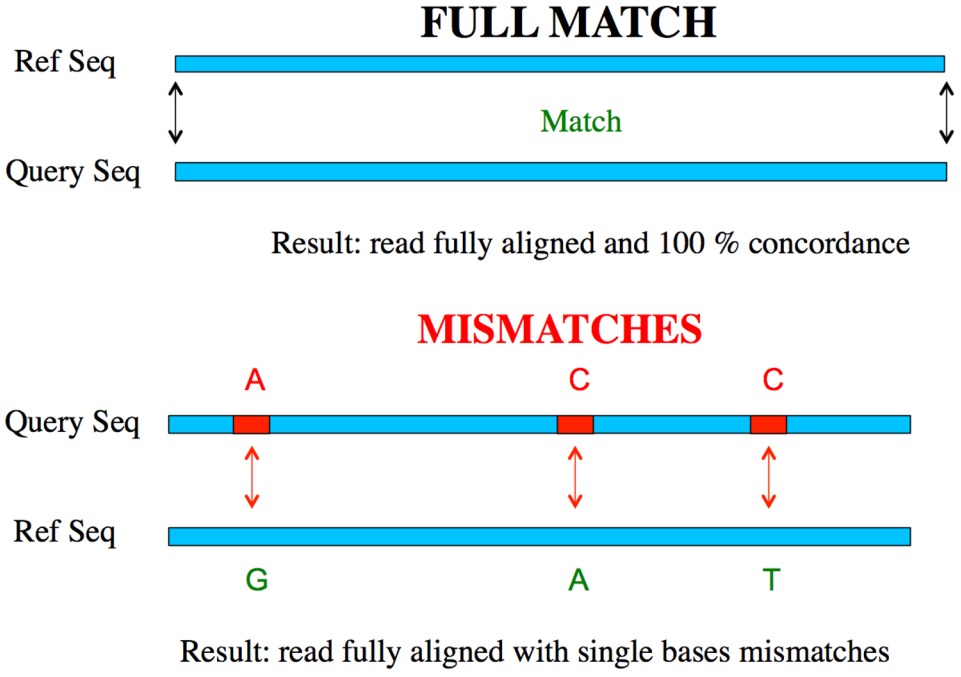
Mapping: locate reads in the reference genome

- Essential step when a reference genome is used
- Many NGS applications:
  - **Quantity: expression quantification, differential analysis (RNA-Seq)**
  - Base mismatch: SNPs, SVs (DNA-Seq)
  - Patterns: protein binding sites, chromatin accessibility, methylation (ChIP-Seq, ATAC-Seq, BS-Seq)



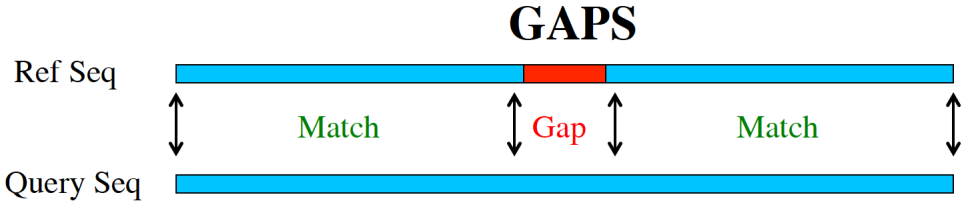
Mapping reads to the reference genome

- Mapping outcomes:

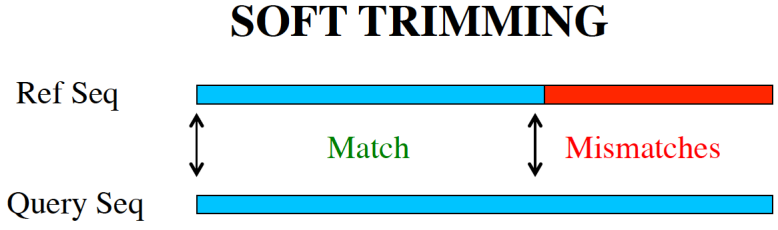


Mapping reads to the reference genome

- Mapping outcomes:



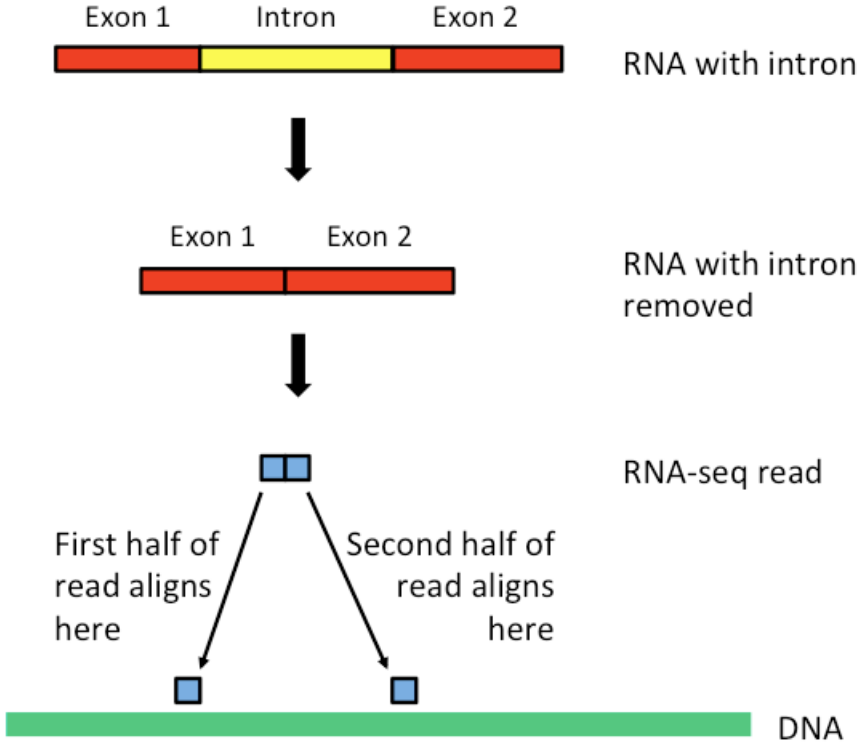
Result: read fully aligned and gap reported



Result: read partially aligned and soft trimming reported

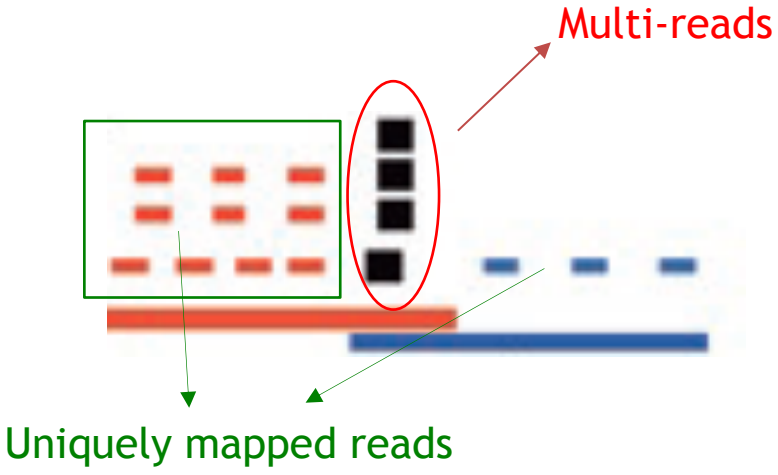
Mapping reads to the reference genome

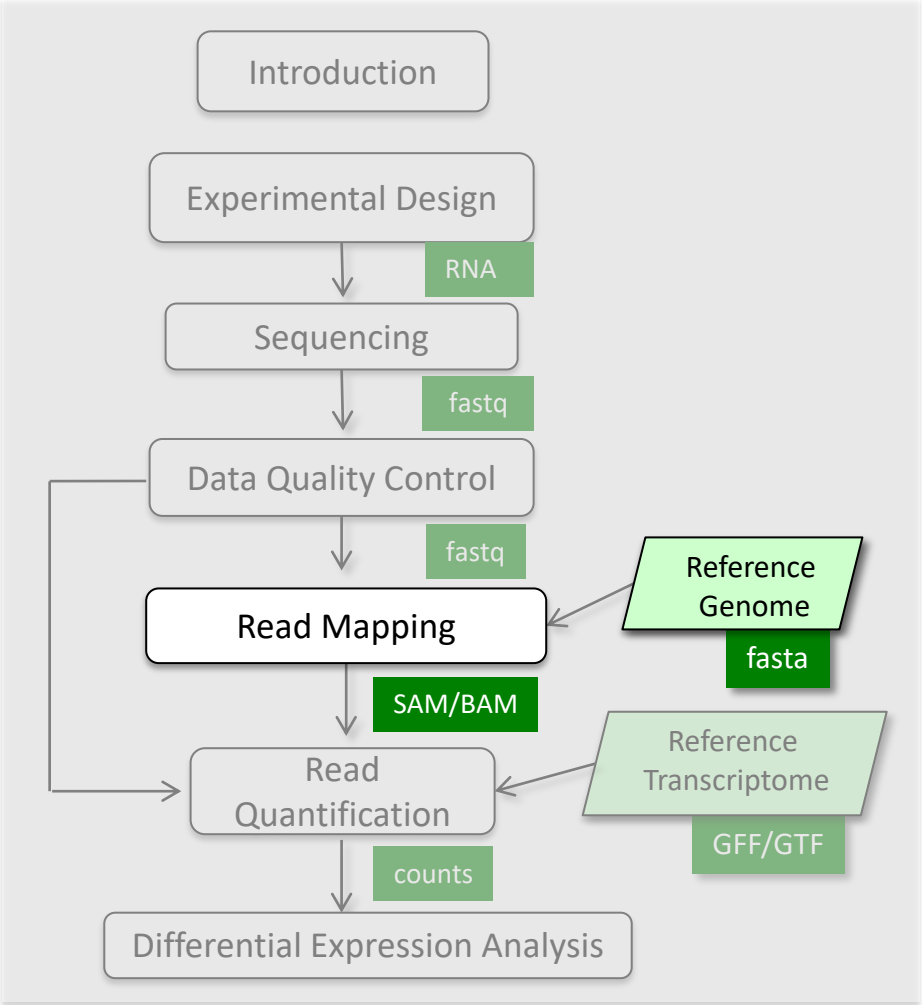
- Mapping outcomes: Spliced alignments



Mapping reads to the reference genome

- Mapping outcomes: Uniquely mapped reads vs multi-reads
  - Repeats, paralogs
  - Shared exons in transcriptome





Read Mapping

- General idea
- Different strategies
- File formats

- Quality control
- Visualization

Mapping challenges

- Quantity
  - 20-30 M reads per sample
- Mapping uncertainties
  - Short reads
  - Sequencing errors, SNPs, Structural variations
  - No exact reference
- Span junctions
  - Exon-exon
  - Structural variations
- Algorithm
  - Fast
  - Able to handle SNPs, indels, sequencing errors
  - Able to allow introns for reference genome alignment

Smith-Waterman algorithm

- Exhaustive search to find the best **local alignment**
  - Alignment with the **highest score**
- Dependent on scoring paradigm

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

Smith-Waterman algorithm example: Where is GATTACA?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
T	G	A	T	T	A	C	A	G	A	T	T	A	C	C	...
G	A	T	T	A	C	A									

Match Score: 1/7

Smith-Waterman algorithm example: Where is GATTACA?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
T	G	A	T	T	A	C	A	G	A	T	T	A	C	C	...
	G	A	T	T	A	C	A								

Match Score: 7/7

Smith-Waterman algorithm example: Where is GATTACA?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
T	G	A	T	T	A	C	A	G	A	T	T	A	C	C	...
		G	A	T	T	A	C	A	...						

Match Score: 1/7

Smith-Waterman algorithm example: Where is GATTACA?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
T	G	A	T	T	A	C	A	G	A	T	T	A	C	C	...
								G	A	T	T	A	C	A	

Match Score: 6/7 <- We may be very interested in these imperfect matches
Especially if there are no perfect end-to-end matches

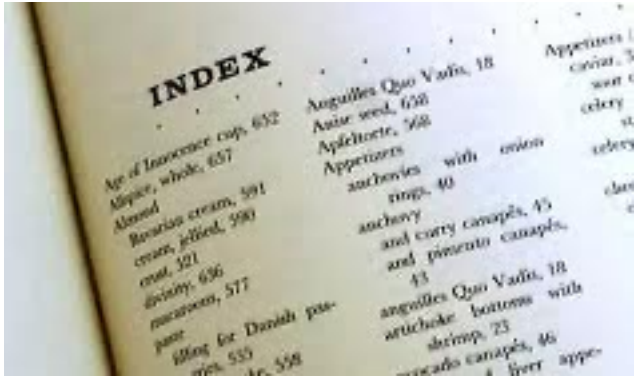
Smith-Waterman algorithm: limitation

- Too slow: one CPU day per million reads
- High sensitivity not needed
  - Only exact and close to exact matches

How to map billions of short reads - index

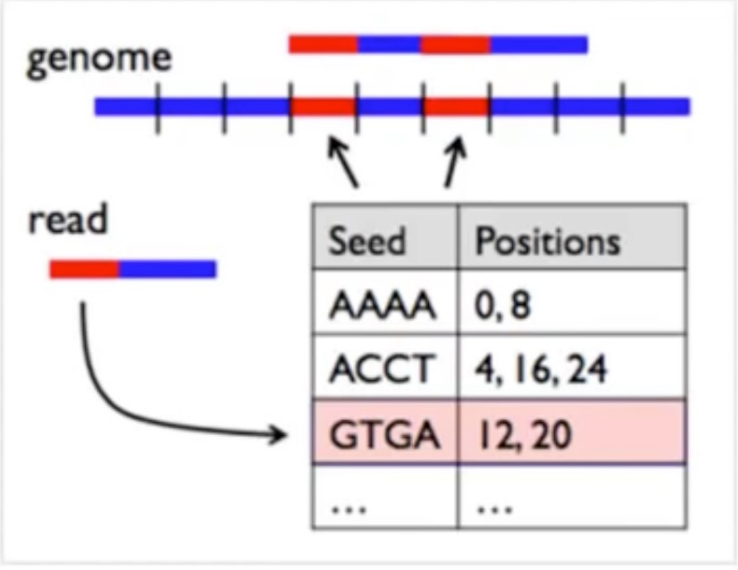
- Index
  - an alphabetical list of names, subjects, etc, with references to the places where they occur
  - Sorted, structured, allow fast search

- NGS mapping software
  - Index reference file
  - Index lookup, not direct sequence comparison
  - Two main categories:
    - Hash table mappers
    - Burrow-Wheeler Transform (BWT) mappers



Hash table (seed) mappers

- Cut reference into small "seeds" (k-mers)
- Store seeds and their positions in a lookup table (hash table)
- Take part of the read and look up the hash table
- Extend seeds to full alignments
  - Smith-Waterman
  - Sensitive but slow
- Shorter seeds -> higher sensitivity, but slower

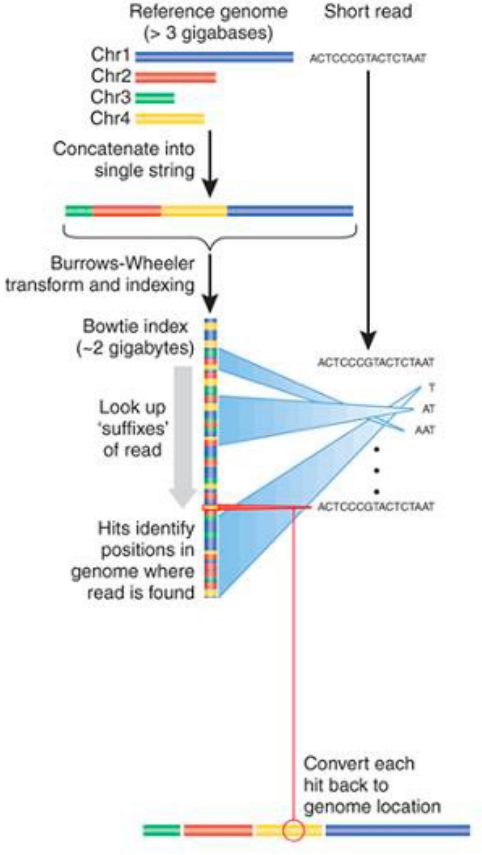


Flicek P. and Birney E. 2009. Nature Method Supplement.
Li H. and Homer N. 2010. Briefs in Bioinformatics.
Garber M. et al. 2011. Nature Methods.

Hash table (seed) mappers: Features and tools

- Pros:
  - Allow more mis-matches
  - Better at indel detection
  - More tolerant to sequence differences (Cross species mapping), higher sensitivity
- Cons:
  - High RAM needed for large genomes (50 Gb for human)
  - Slower
- Tools:
  - Eland, SOAP, MAQ, SHRiMP, GSNAP, …

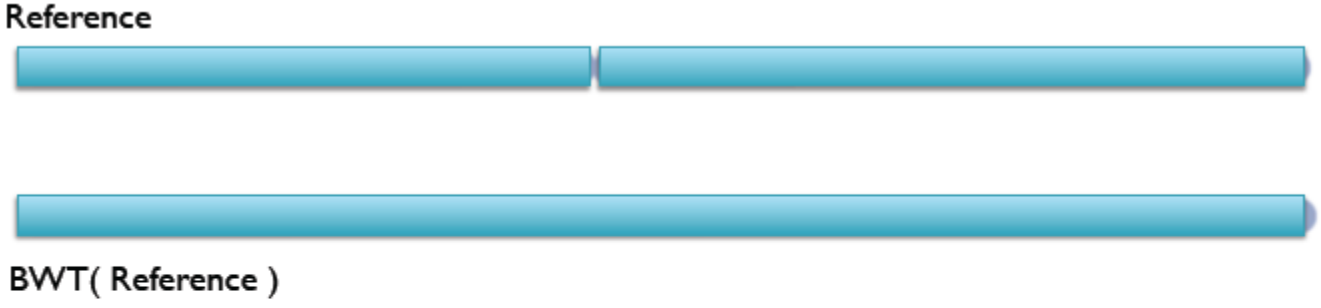
Burrow-Wheeler-Transformation



- Sort and index genome (BWT)
- Align read base by base to find positions in the genome
- Index generation and read alignment examples are provided in the supplementary slides

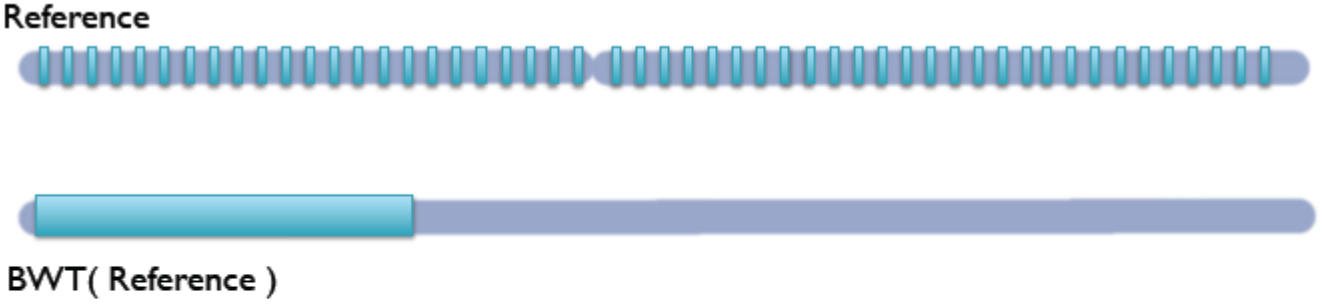
Trapnell & Salzberg(2009) Nature Biotech 27, 455 -457

BWT mappers: Bowtie



Query:
AATGATACGGCGACCACCGAGATCTA

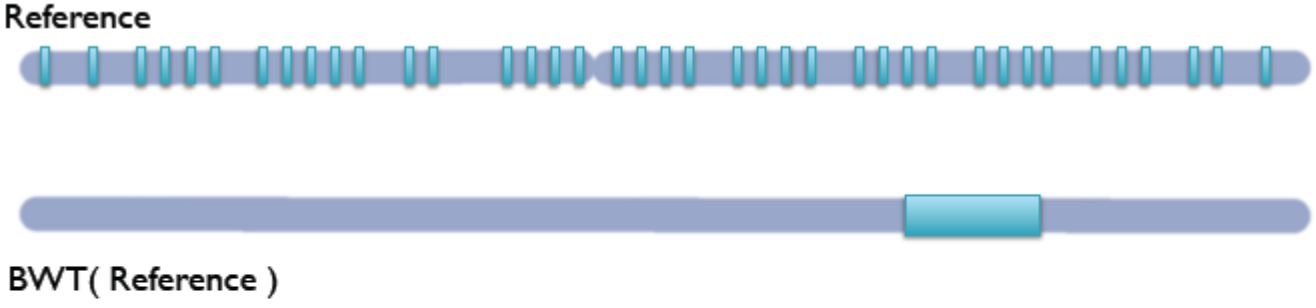
BWT mappers: Bowtie



- Backward search exact matching
- Searched suffix appears consecutively in BWT

Slides courtesy of Ben Langmead (langmead@umiacs.umd.edu)

BWT mappers: Bowtie



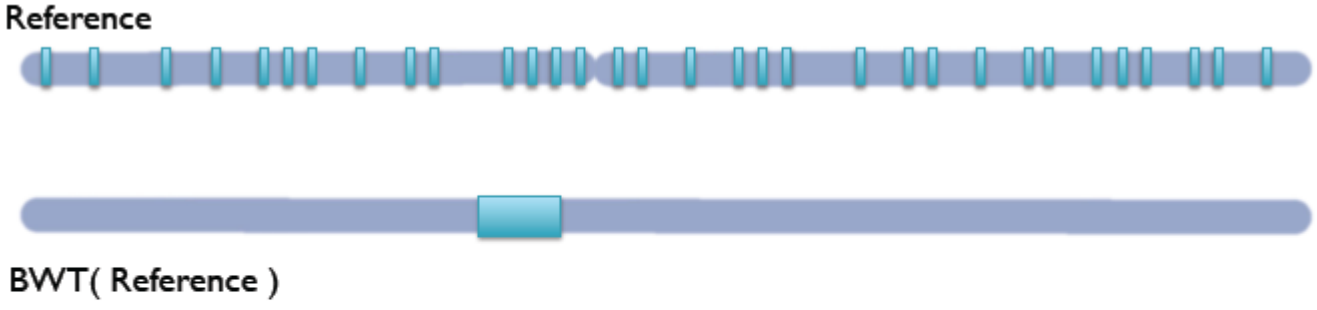
Query:
AATGATACGGCGACCACCGAGATCTA

At each step:

- Searching suffix grows by one character
- The size of the range in BWT shrinks/remains the same

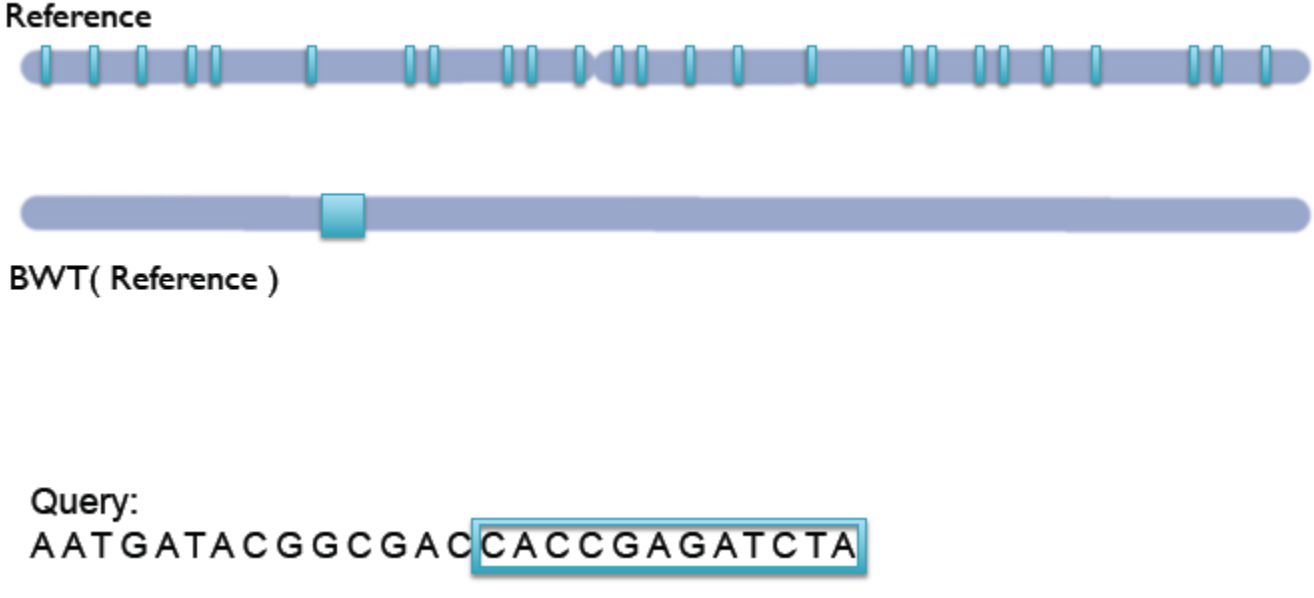
Slides courtesy of Ben Langmead (langmead@umiacs.umd.edu)

BWT mappers: Bowtie



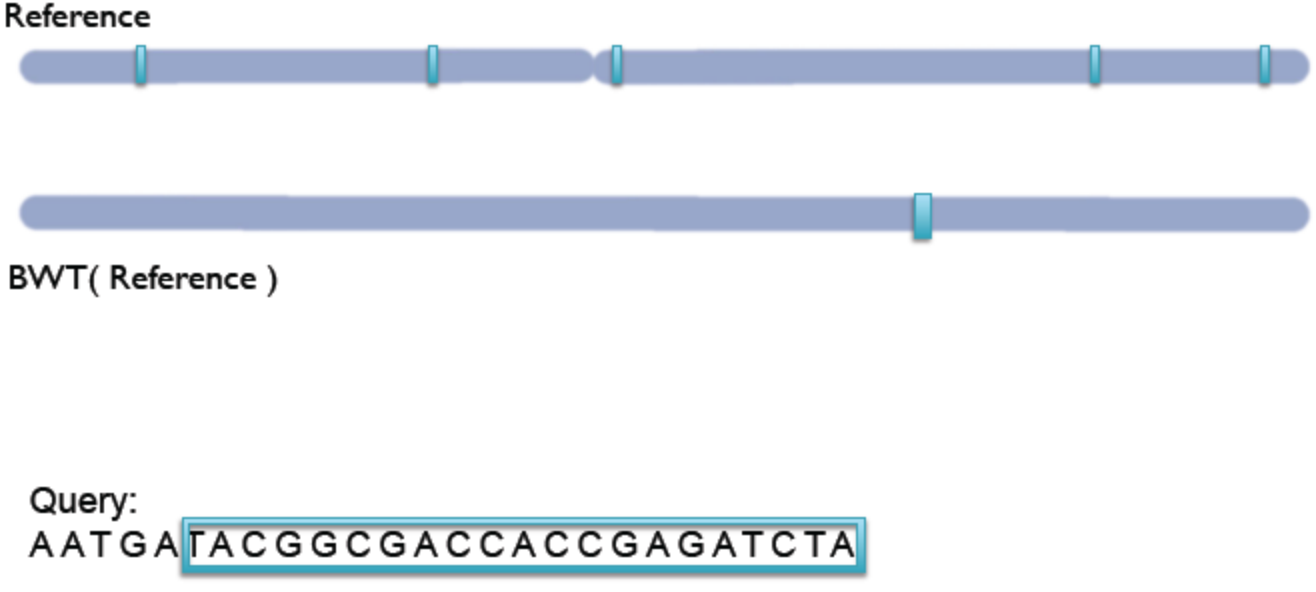
Slides courtesy of Ben Langmead (langmead@umiacs.umd.edu)

BWT mappers: Bowtie



Slides courtesy of Ben Langmead (langmead@umiacs.umd.edu)

BWT mappers: Bowtie



Slides courtesy of Ben Langmead (langmead@umiacs.umd.edu)

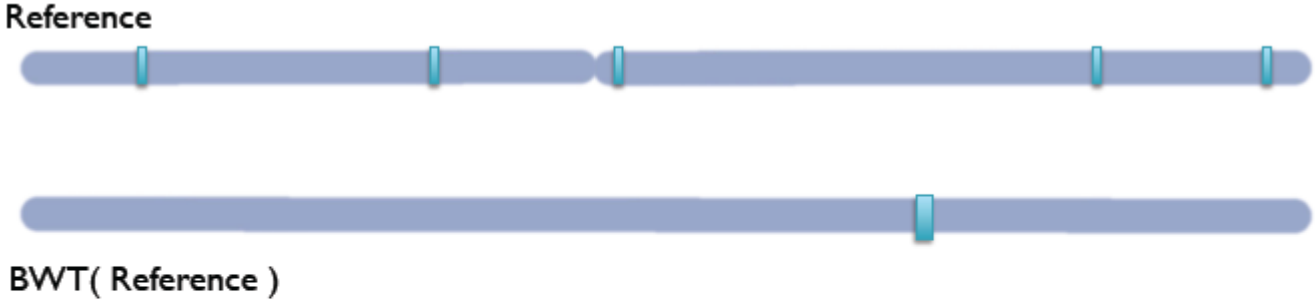
BWT mappers: Bowtie

A mismatch in the search suffix -> empty BWT range/failed index lookup

- Sequencing error (Illumina 1/1000)
- True mutation (*A. thaliana* 7X10^-9 per site per generation)
- Mismatches are not rare event (at least 10% of >100 nt reads)

Slides courtesy of Ben Langmead (langmead@umiacs.umd.edu)

BWT mappers: Bowtie



Empty BWT range activates backtracking

- All different bases are tried at the mismatched position
- Chop reads in short segments (seeds), align those mismatch-free, stitch seed alignments together

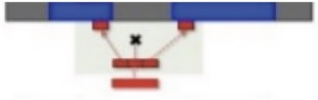
Slides courtesy of Ben Langmead (langmead@umiacs.umd.edu)

BWT mappers: Features and tools

- Pros:
  - Uses BWT index based approach
  - Less RAM is needed (2-8 GB for human genome)
  - Fast for searching perfect or close matches
- Cons:
  - Performance decreases with distant matches
- Tools:
  - BWA, bowtie, bowtie2, STAR, SOAP2, …

Un-spliced vs. spliced mappers

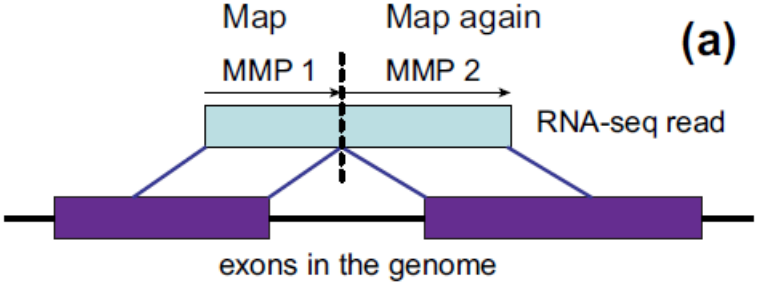
- Un-spliced mappers (DNA-Seq)
  - Align continuous reads (not containing gaps as a result of splicing)
  - When encountering an intron, the aligner **stops and trims the rest of the read**
  - BWA, BOWTIE2, BLAST
  - Splice junctions are **impossible** to detect



- Spliced mappers (RNA-Seq)
  - Aware of the presence of introns
  - When encountering an intron, the aligner **does not stop** to trim the rest of the read but continues to find the next exon
  - TopHat, **STAR**, GMAP, BLAT

Spliced mappers: STAR

- Ultra fast (does BWT as much as possible)
- Start with BWT until a mismatch is encountered, then store the possible alignment candidates, and start a new BWT from scratch instead of doing Smith-Waterman search.



MMP = Maximal Mappable pieces

- Error tolerant (sensitive)
- Both short and long reads

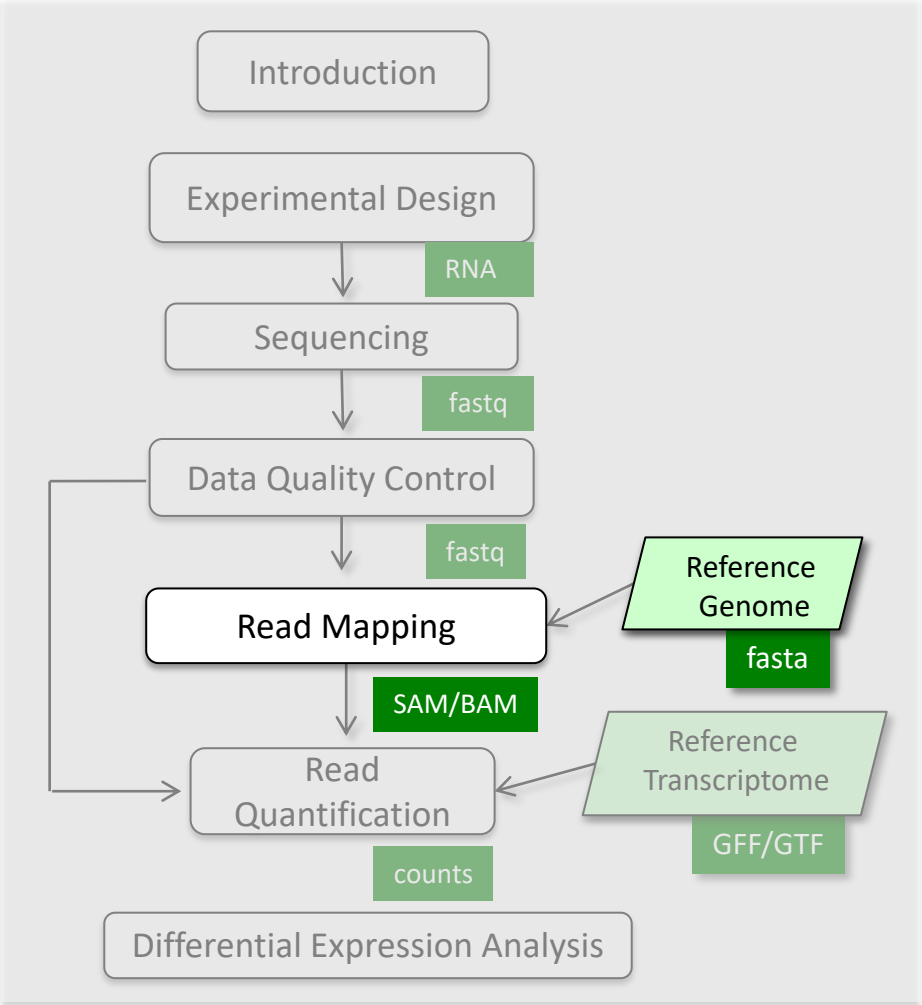
Dobin A. et. al. 2012. Bioinformatics. doi: 10.1093/bts635

Choosing an alignment tool…

- Default options may not be best

"… there is no tool that outperforms all of the others in all the tests. Therefore, the end user should clearly specify his needs in order to choose the tool that provides the best results." - Hatem et al *BMC Bioinformatics* 2013, **14**:184

Mapping Clicker Questions



Read Mapping

- General idea
- Different strategies
- File formats

- Quality control
- Visualization

Sequence / Alignment (SAM) files

- **SAM (Sequence Alignment/Map)**
  - Single unified format for storing read alignments to a reference genome
  - Large plain text file
  - RNA-Seq: >10Gb
  - Exome: >50Gb, Whole Genome: 0.8-1Tb

- **BAM (Binary Alignment/Map)**
  - Binary equivalent of SAM
  - Compressed data plus index (bai)
  - Developed for fast processing/indexing
  - RNA-Seq: >2Gb
  - Exome: 2-10Gb, Whole Genome: 100-300Gb

SAM/BAM read information

Reads (x4)

Mapping quality score

- A score that indicates how well the read is aligned
- Probability that a read is mapped incorrectly

- Phred scaled
  - mapP = probability that a read is mapped incorrectly
  - mapQ = $-10\log_{10}$mP
  - Q30: 1 in 1000 incorrect
  - Range from 1 to 254
  - The higher, the better
  - 255: mapping quality not available, but for unique alignments (STAR)

- Mapper dependent, difficult to compute, difficult to compare

Meaning of mapQ30

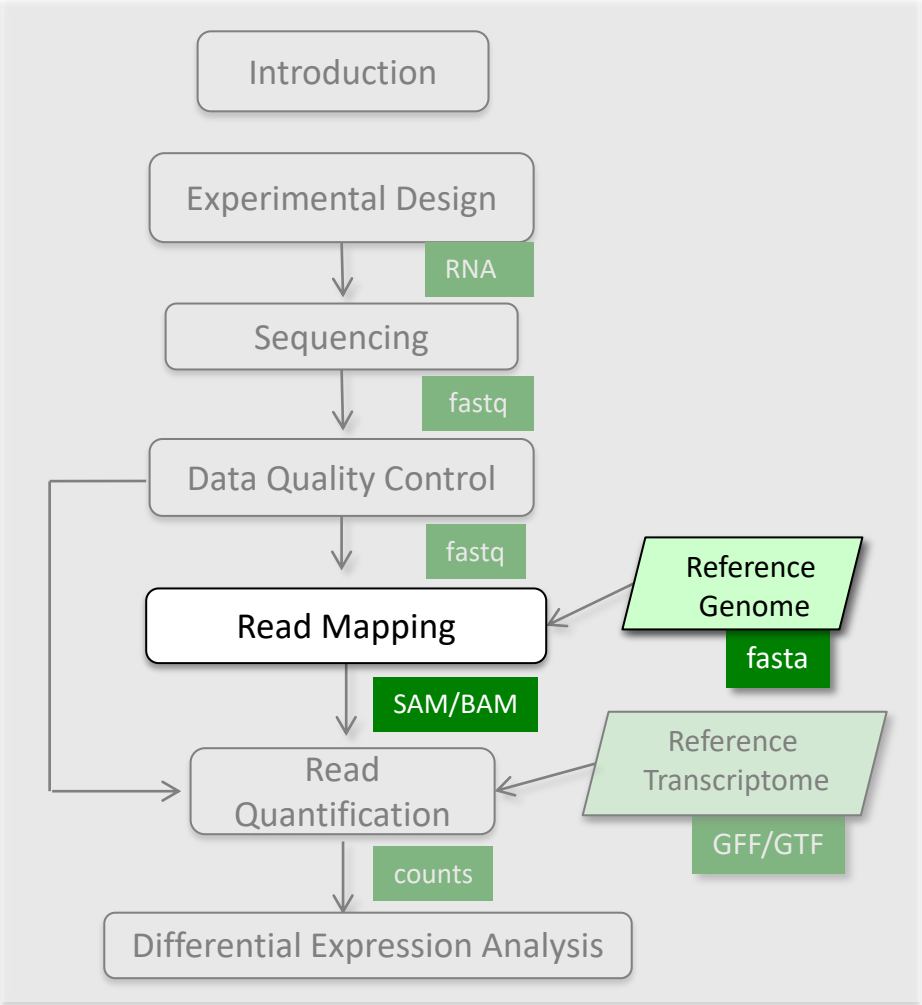
- The overall **base quality** of the read is **good**.
- The read has few or just one `good' hit on the reference -> **best alignment can be easily identified**
- The best alignment has **few mismatches** -> actual mutations or sequencing errors.
- mapQ30 is usually required for SNP calling algorithms and detection of structural variations

Cause of poor mapQ

- Poor quality reads (Low base quality -> low mapping quality)
- Paired end reads or not (Reads mapped in pairs -> more likely to be correct)

- Reference with poor quality
- High divergence between the sequenced population and reference
- Repeats (Reads in repetitive regions -> very low mapping quality)

- Poor choice of the mapping software (An algorithm with low sensitivity -> more mapping errors)
- Improper alignment parameters



Read Mapping

- General idea
- Different strategies
- File formats

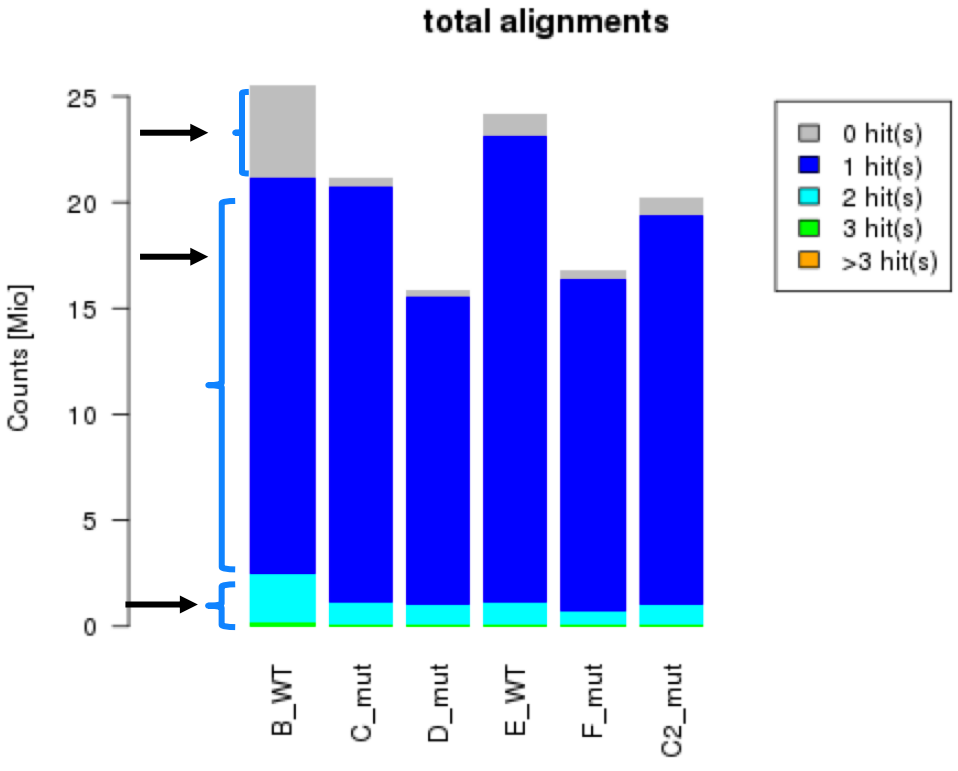
- Quality control
- Visualization

Why QC on mapping results

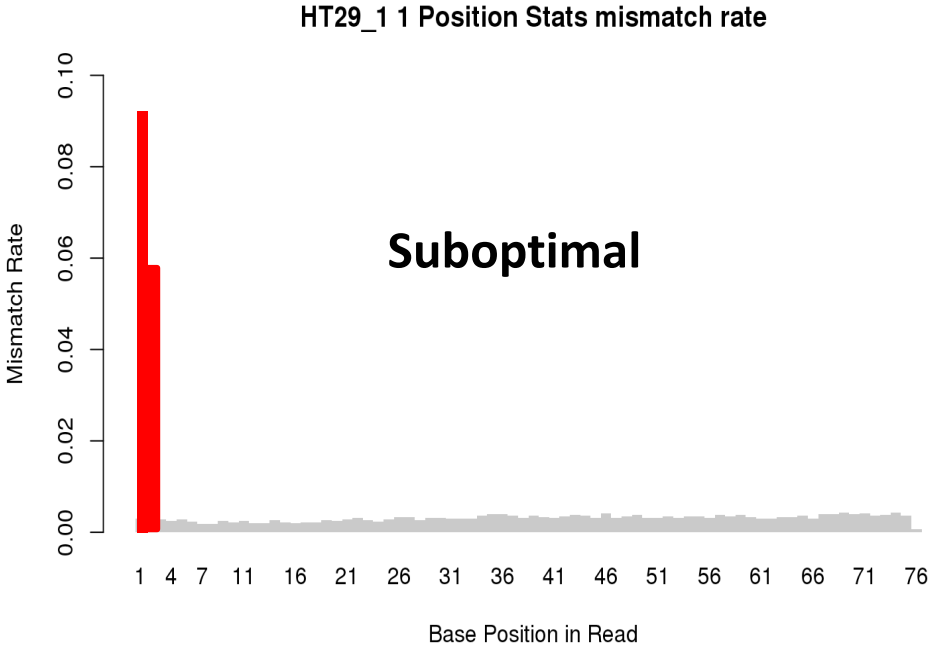
- If the data pre-processing and mapping are done properly
  - Extremely low mapping rate for within species read mapping
- If the alignment results fit expected outcomes
  - abundance of genomic features
  - Expected coverage depth and distribution
- Capturing of technical issues
  - Sample degradation
  - Over-amplification issues (Duplication rate)

Mapping QC metrics: Summary stats

- How well did reads align to the reference?
- Summary statistics
  - % reads with **no alignment**
  - % reads with **unique alignment**
  - % reads with **multiple alignments**
- we aim for >70% unique alignments
- Samples with lower mapping rates should be investigated (contamination vs. quality issues)



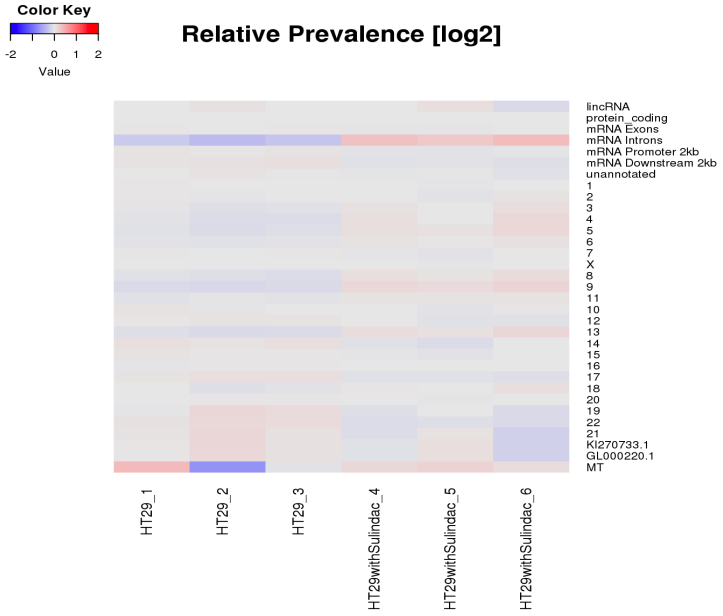
Read position specific error rate



Hard trimming of first 2-3 bases

Mapping QC metrics: Abundance of genomic features

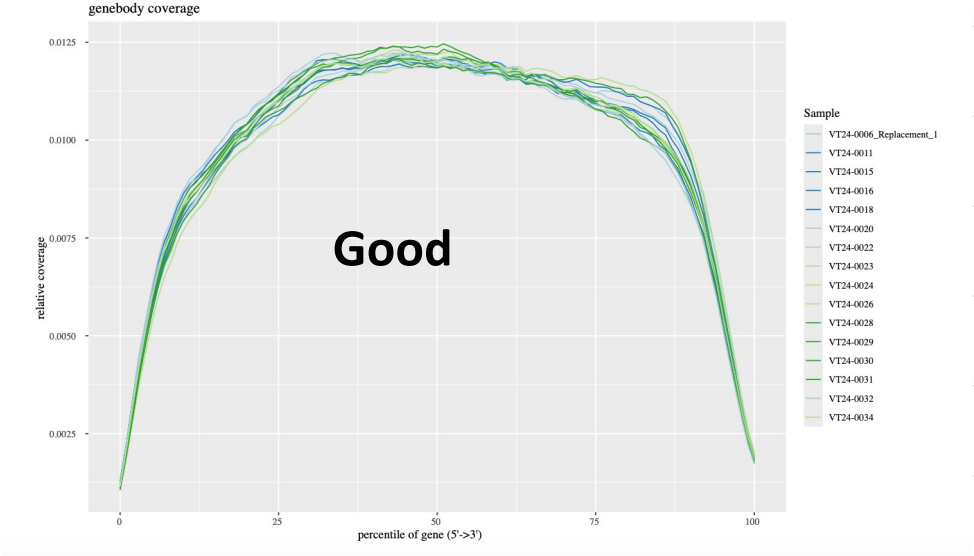
- Relative abundance of annotation features: intron, exon, up/down stream, unannotated
- Are there samples with different abundance of certain features?

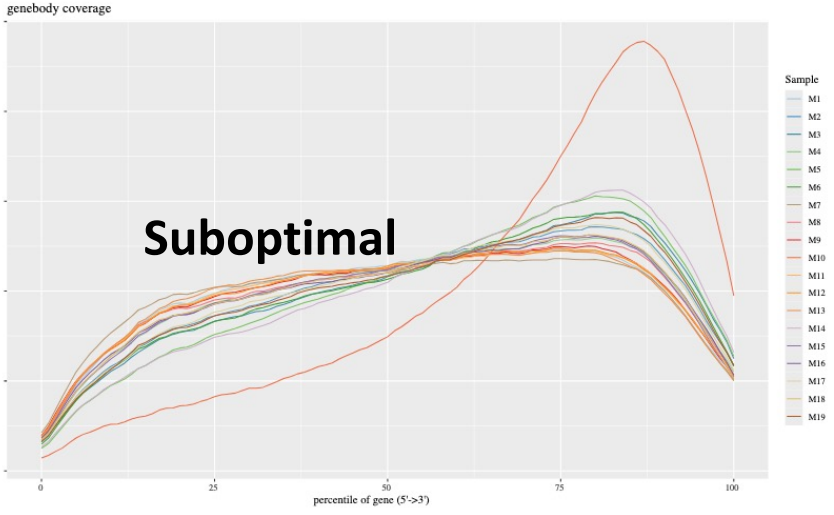


Match Count Percentages

	HT29_1	HT29_2	HT29_3	HT29withSulindac_4	HT29withSulindac_5	HT29withSulindac_6
lincRNA	3.43	3.590	3.470	3.41	3.63	3.20
protein_coding	78.40	77.700	78.100	78.00	77.20	78.70
mRNA Exons	75.10	74.700	74.900	73.20	72.60	73.80
mRNA Introns	3.30	3.010	3.200	4.81	4.67	4.96

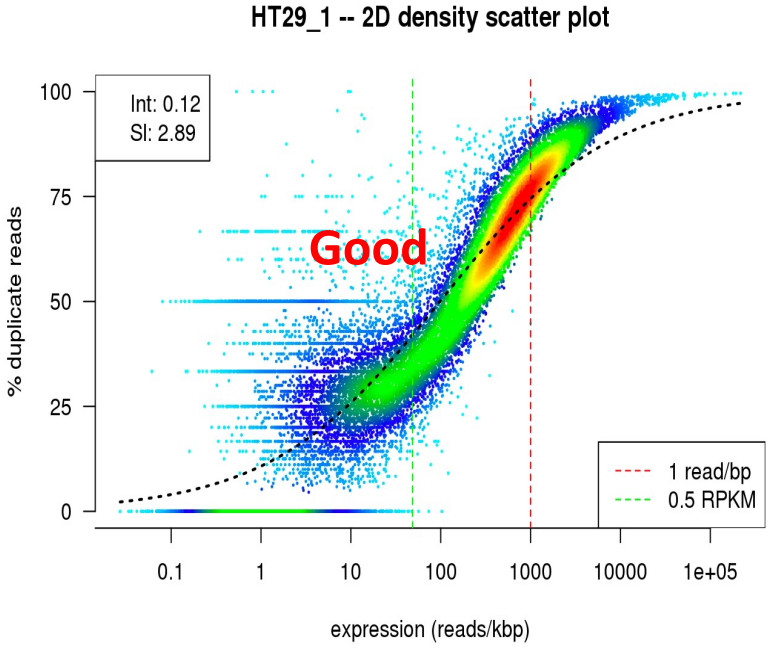
Sample degradation check: Transcript Coverage Bias

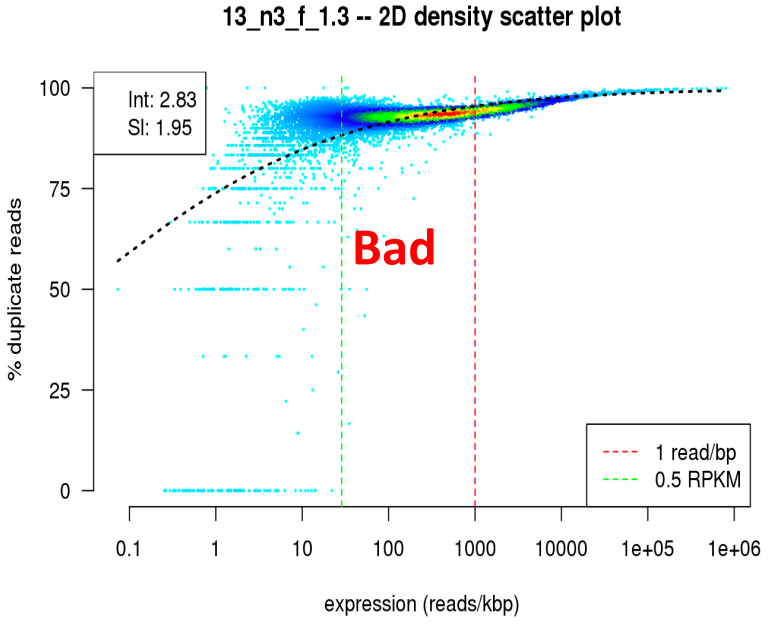




Exclude degraded sample

Overamplification check: Duplication rate QC

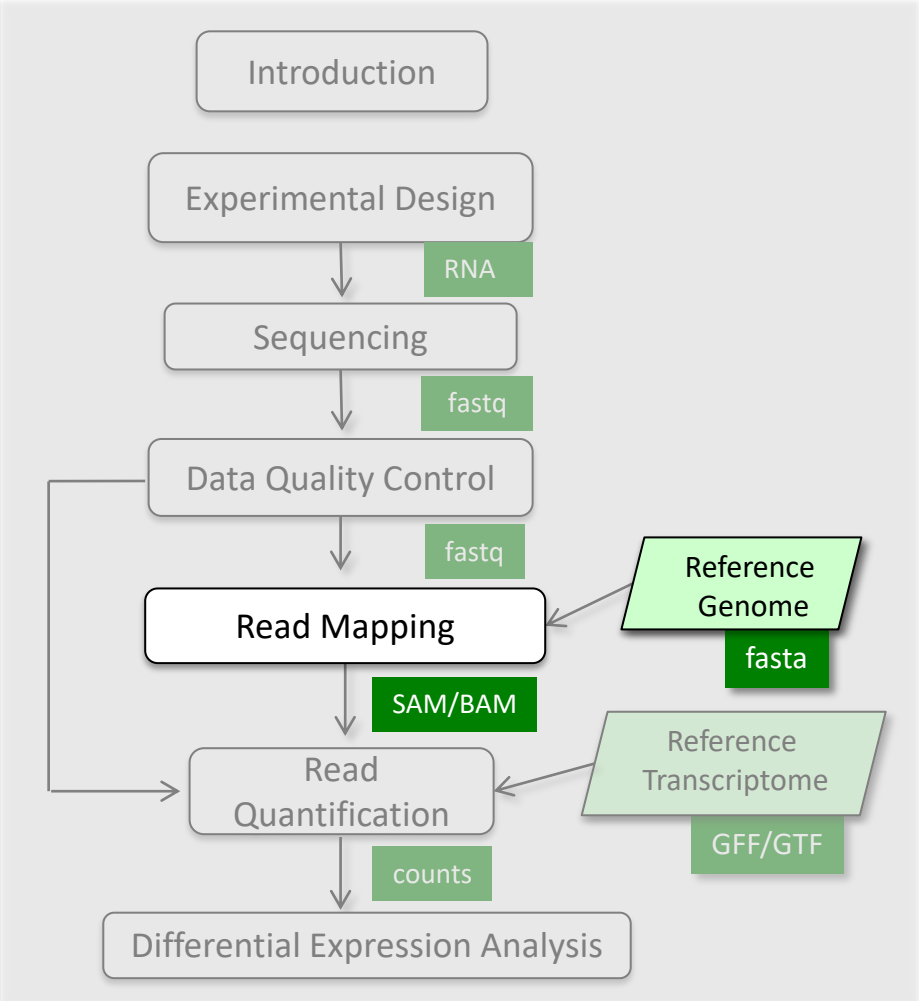




Repeat library prep

RNA-seq mapping QC tools

- RNA-SeQC
  - https://confluence.broadinstitute.org/display/CGATools/RNA-SeQC
- EVER-seq (RSeQC)
  - http://code.google.com/p/rseqc/
- Qualimap
  - http://qualimap.bioinfo.cipf.es/



Read Mapping

- General idea
- Different strategies
- File formats

- Quality control
- Visualization

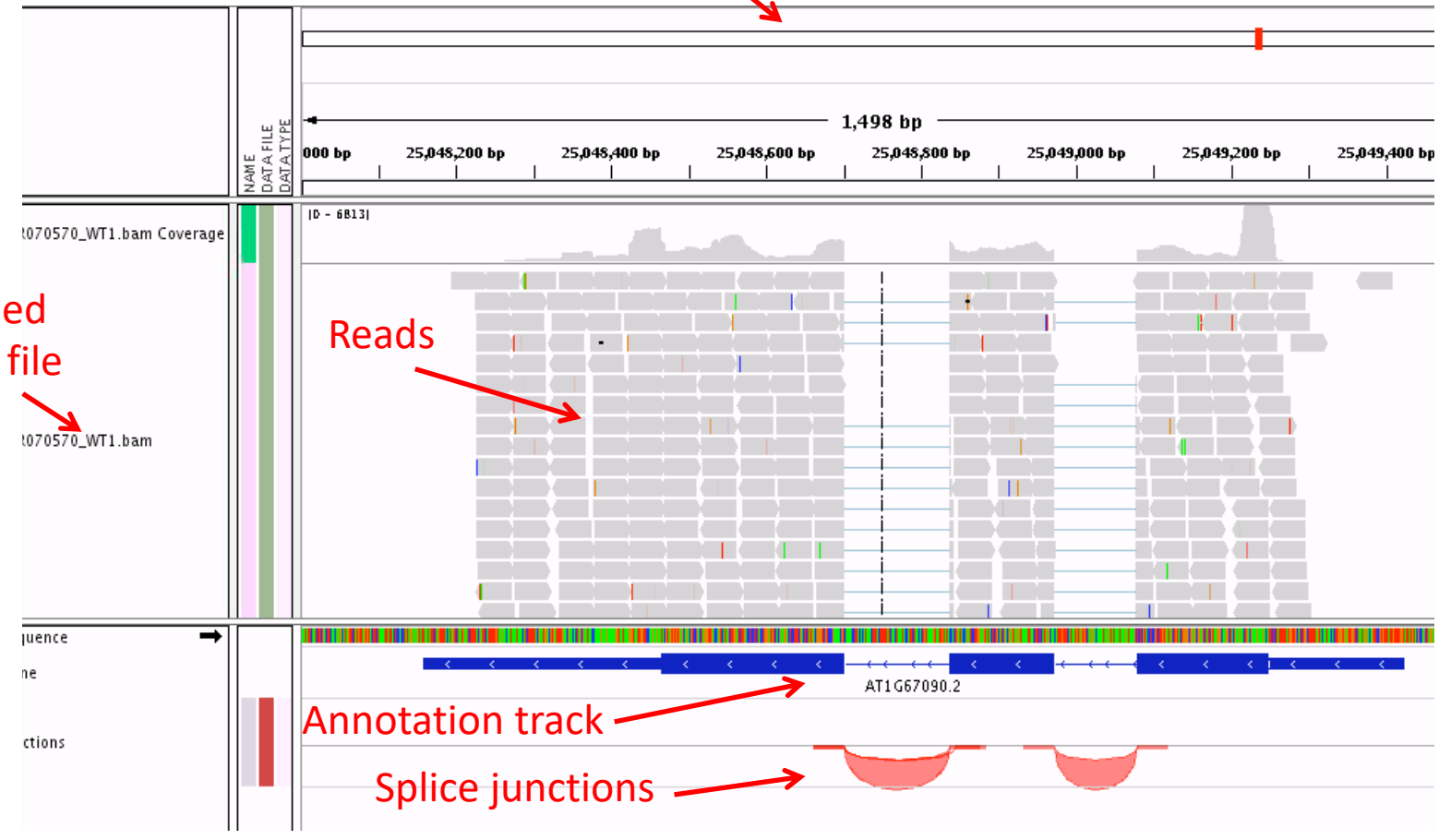
How to visualize mapping results

- Using a genome browser
  - Software enable users to browse multiple data types and annotations in the context of the genome

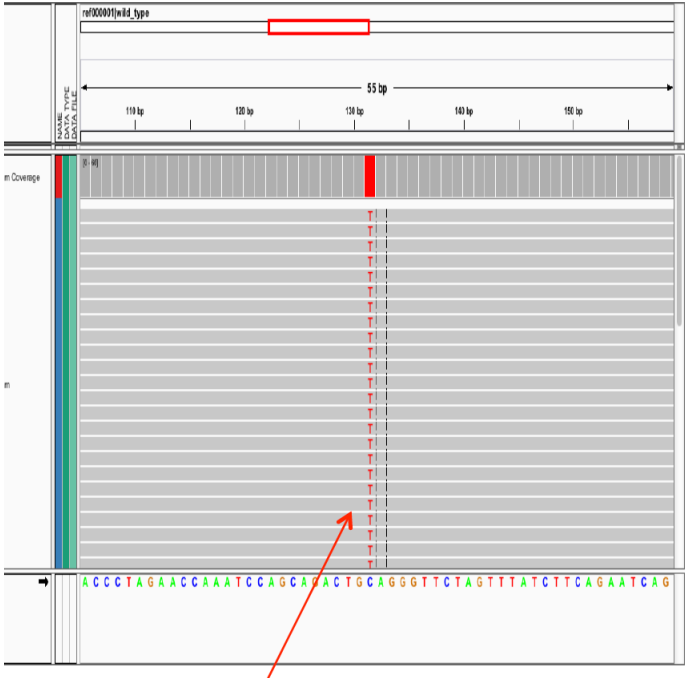
- UCSC and Ensembl browser
  - Reference genome should be hosted on the UCSC/Ensembl server
  - Upload customized data files or import via URL

- **Integrative Genomics Viewer** (IGV) (Broad institute)
  - Open source
  - Well maintained, actively developed
  - Platform independent, easy to use

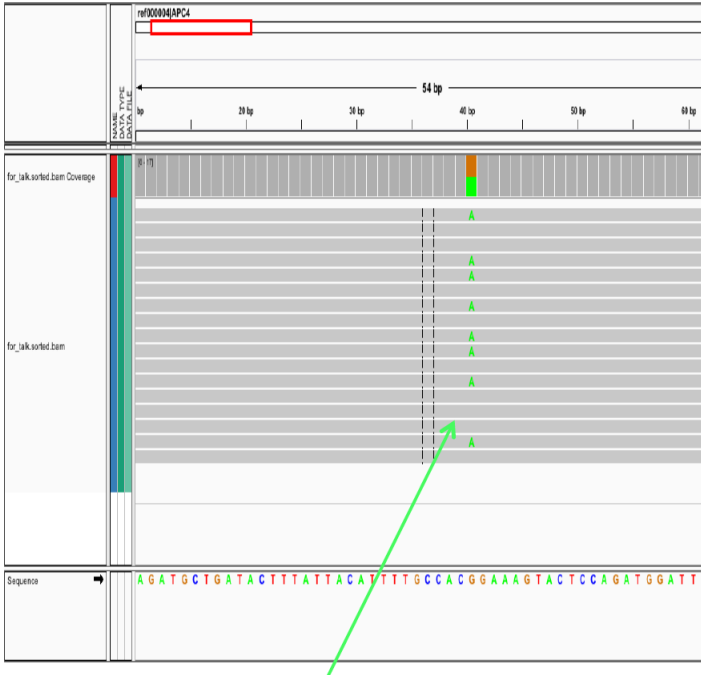
Mapped reads in IGV



SNPs

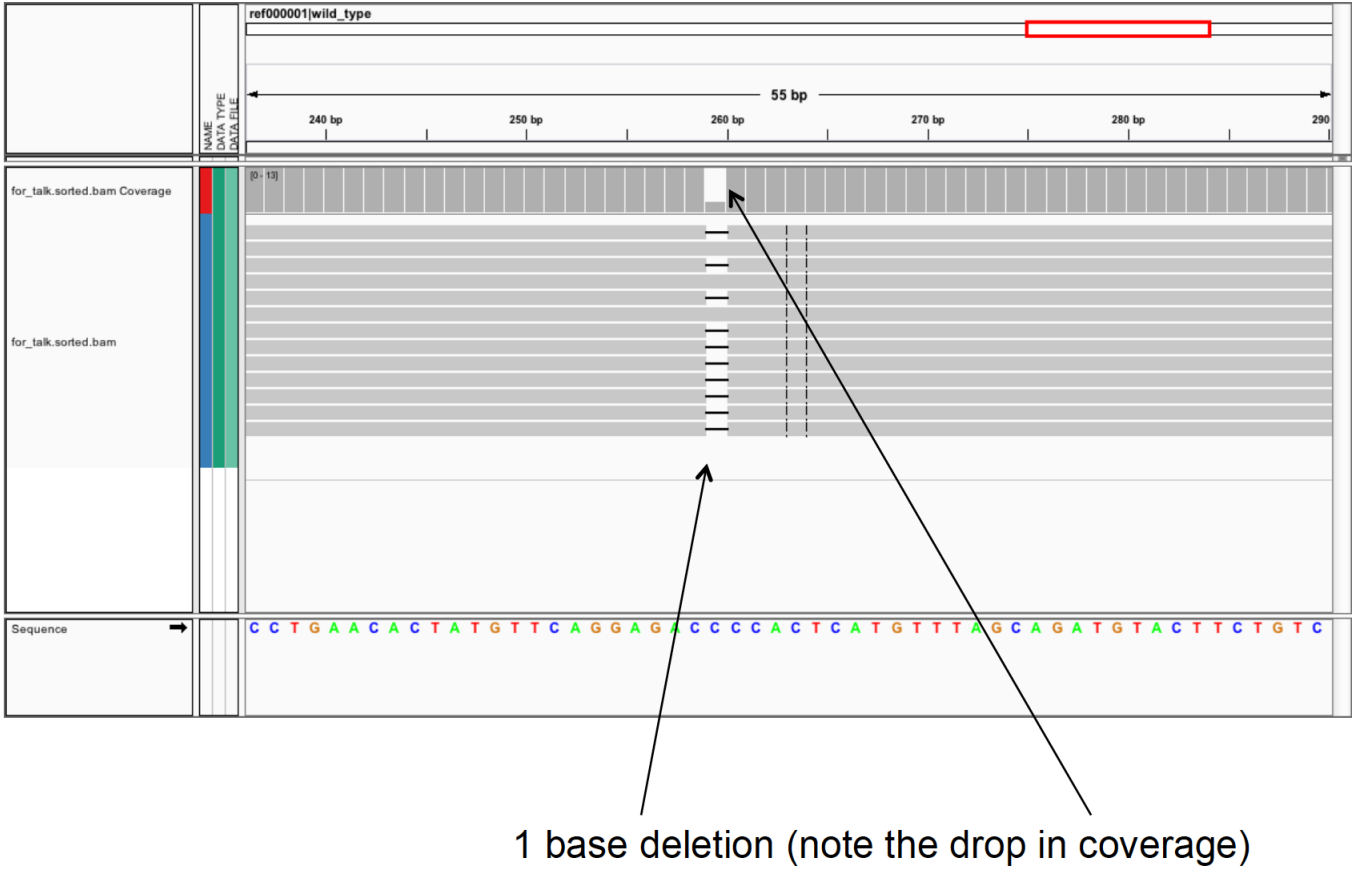


C > T homozygous substitution

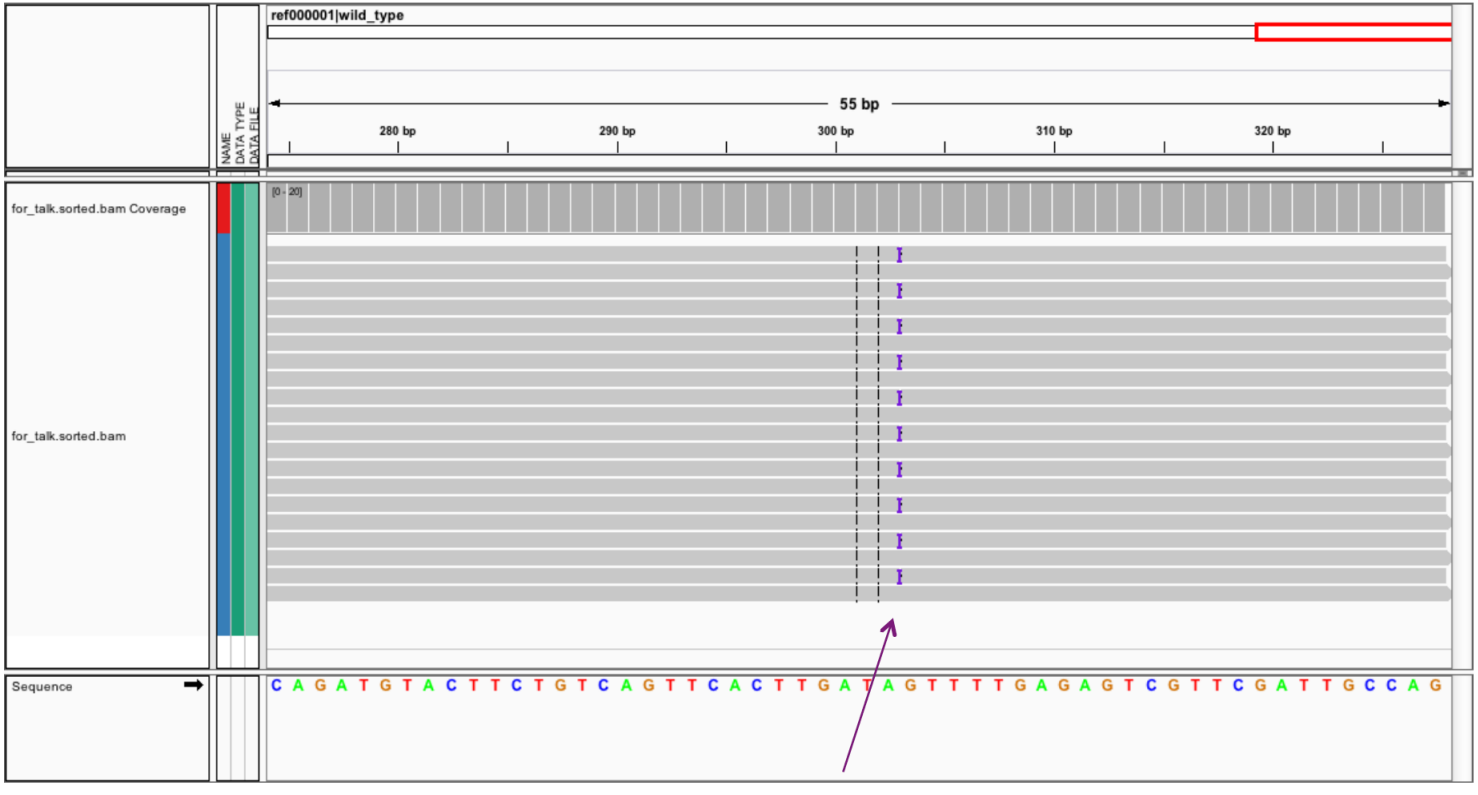


G > A heterozygous substitution

Single base deletions

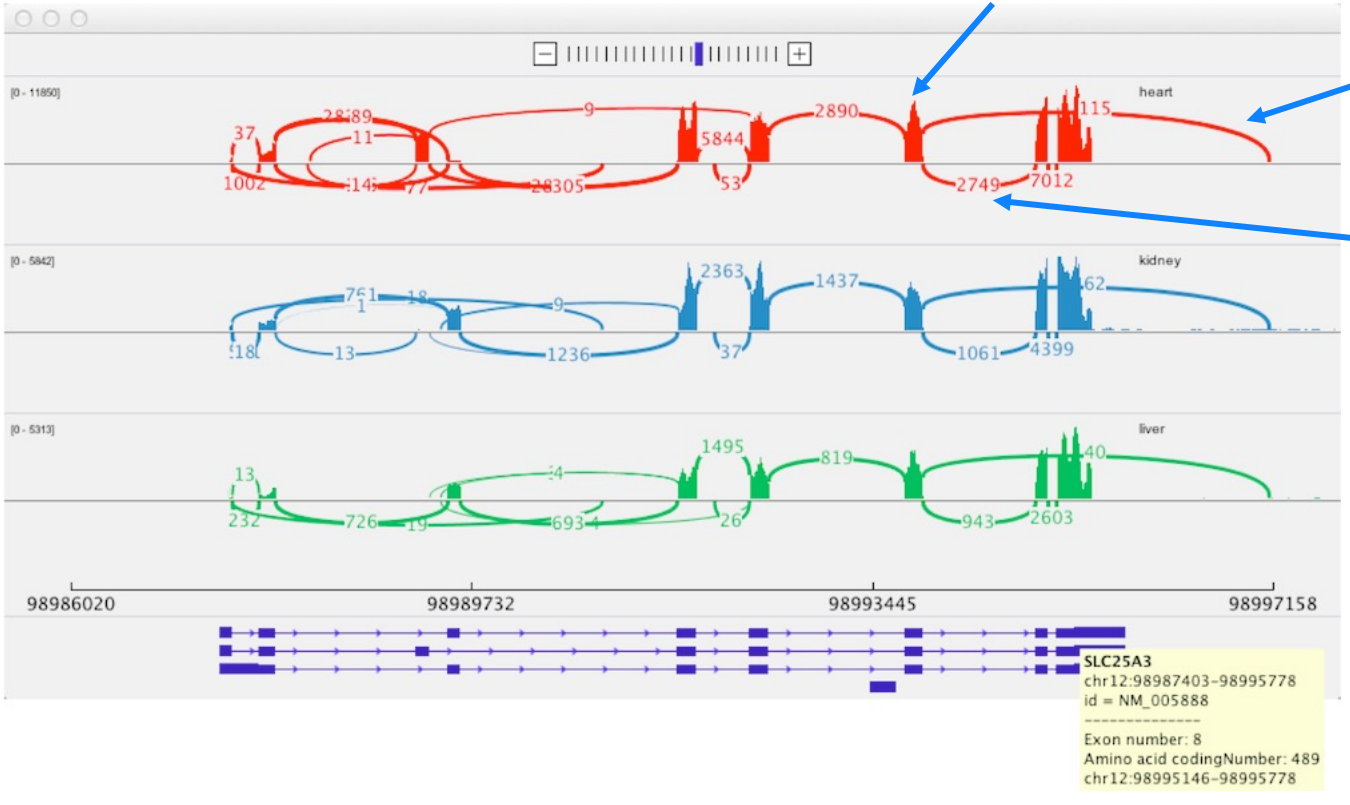


Single base insertions



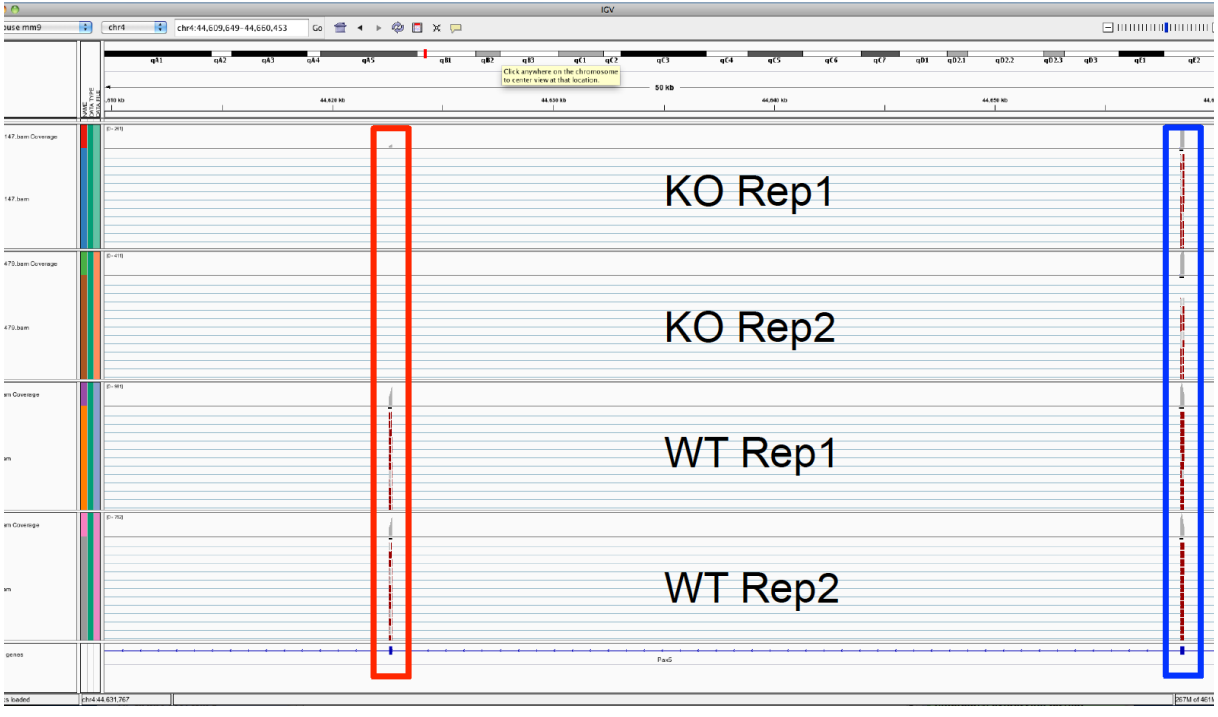
Sashimi plots

- Quantitatively visualize splice junctions



Interactive mapping QC using IGV

- Are my data behaving as expected?
  - No reads mapped to the knock-out site



Read mapping: Summary

- Different mapping tasks
  - No splicing: general mappers
  - Splicing: spliced mappers
- General mappers
  - Index lookup instead of direct sequence comparison
  - Hash table indexing – seed methods
  - Burrow-Wheeler transform methods (Suffix/prefix trees)
- Spliced mappers
  - Build upon general mappers
- Explore the mapping results to spot unexpected trends early
  - Mapping QC tools
  - IGV

Supplementary slides

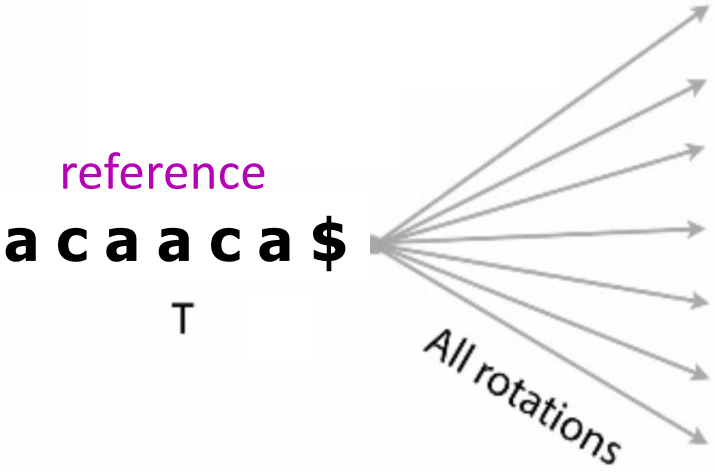
Burrow - Wheeler Transform (BWT) mappers: Index

reference

a c a a c a $

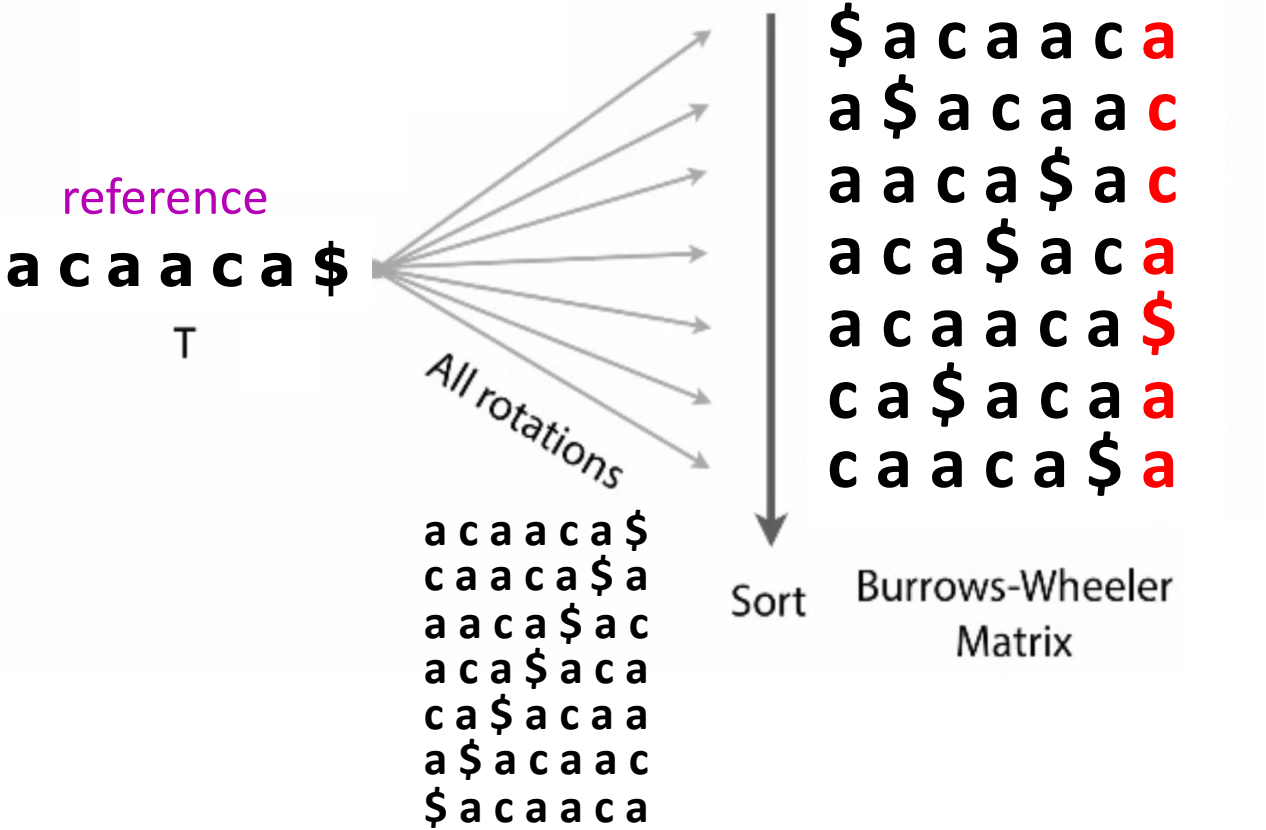
T

Burrow - Wheeler Transform (BWT) mappers: Index

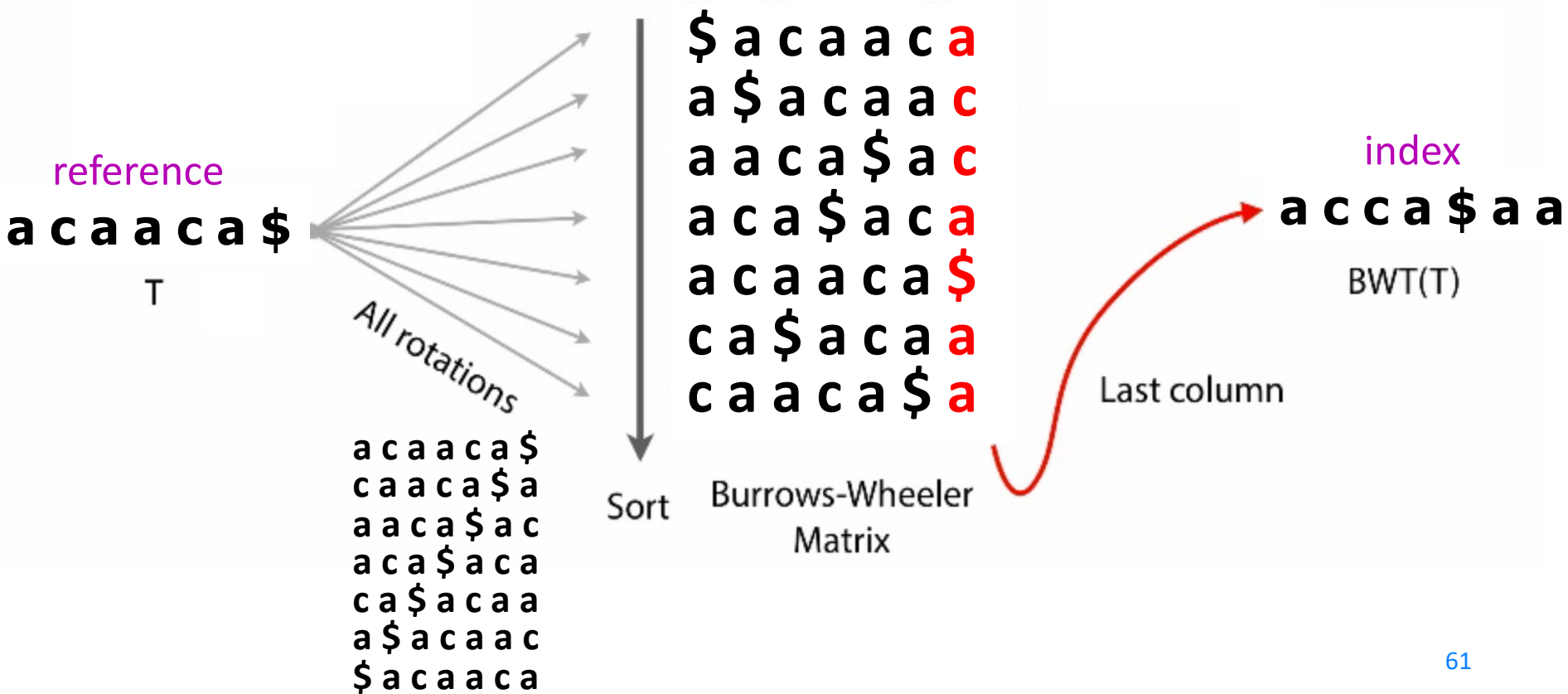


a c a a c a $
c a a c a $ a
a a c a $ a c
a c a $ a c a
c a $ a c a a
a $ a c a a c
$ a c a a c a

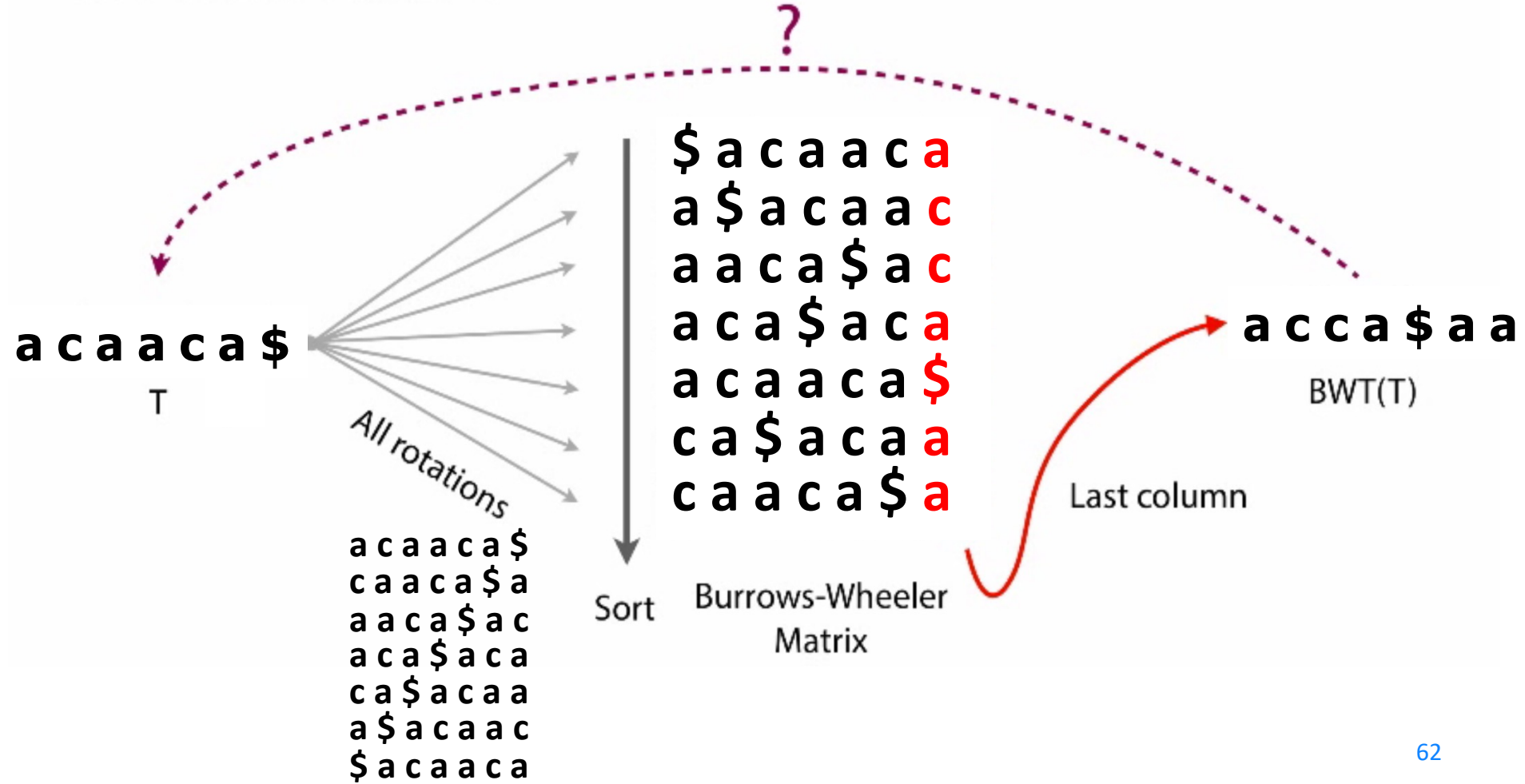
Burrow - Wheeler Transform (BWT) mappers: Index



Burrow - Wheeler Transform (BWT) mappers: Index



Burrow - Wheeler Transform (BWT) mappers: Index



BWT mappers: Find T knowing BWT(T)

- We give each character a rank, equal to the times the character appear

$$a_1\ c_1\ a_2\ a_3\ c_2\ a_4\ \$$$

- Transform produces a BWT matrix

```
$ a c a a c a
a $ a c a a c
a a c a $ a c
a c a $ a c a
a c a a c a $
c a $ a c a a
c a a c a $ a
```

BWT matrix

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix

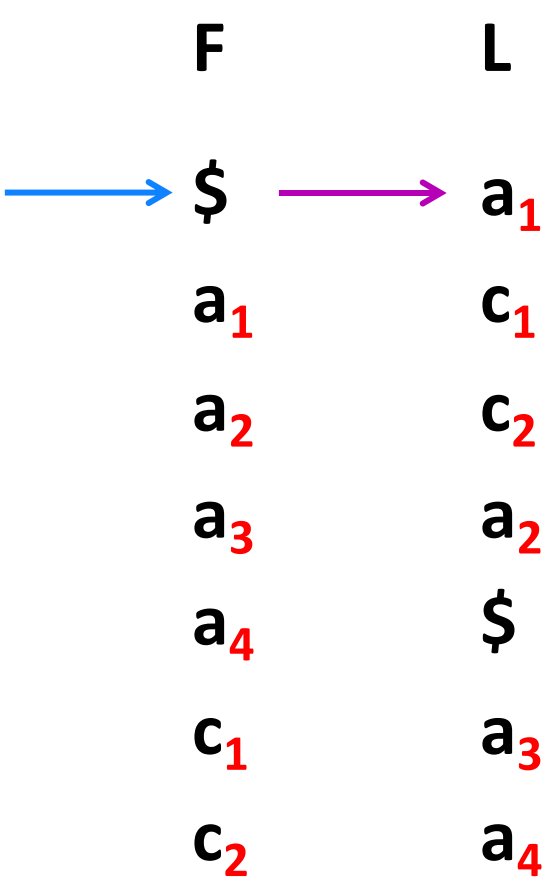
BWT matrices have a property called the Last First (LF) Mapping: the ith occurrence of character c in the last column corresponds to the same text character as the ith occurrence of c in the first column.

F	L
$	a_1
a_1	c_1
a_2	c_2
a_3	a_2
a_4	$
c_1	a_3
c_2	a_4

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix



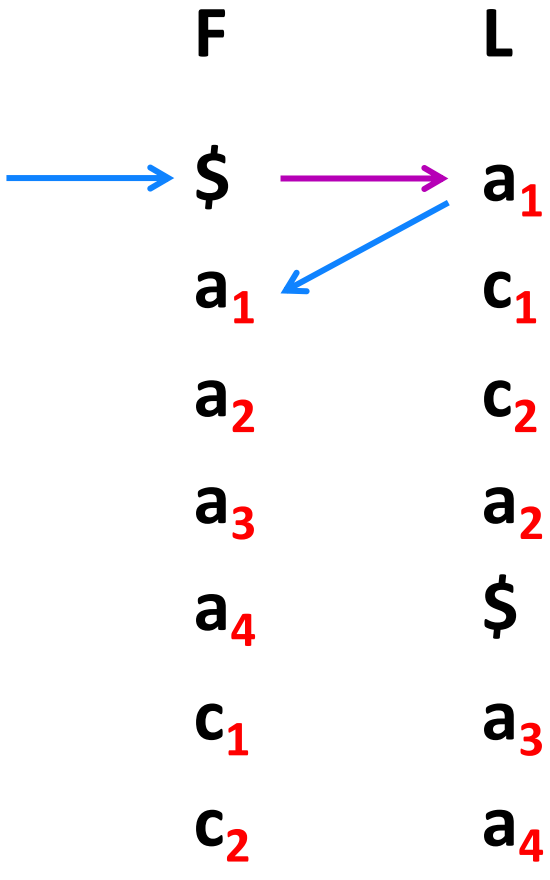
Find T knowing BWT(T):

T = $

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix



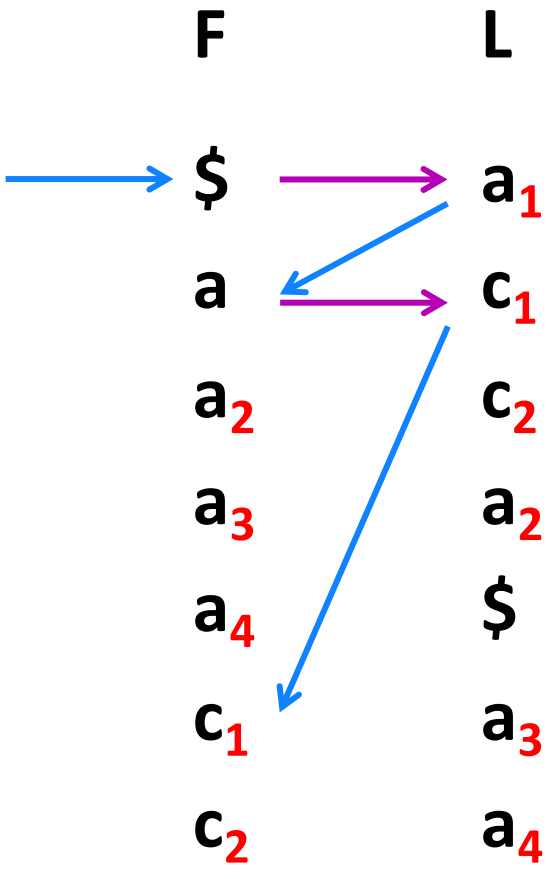
Find T knowing BWT(T):

T = a $

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix



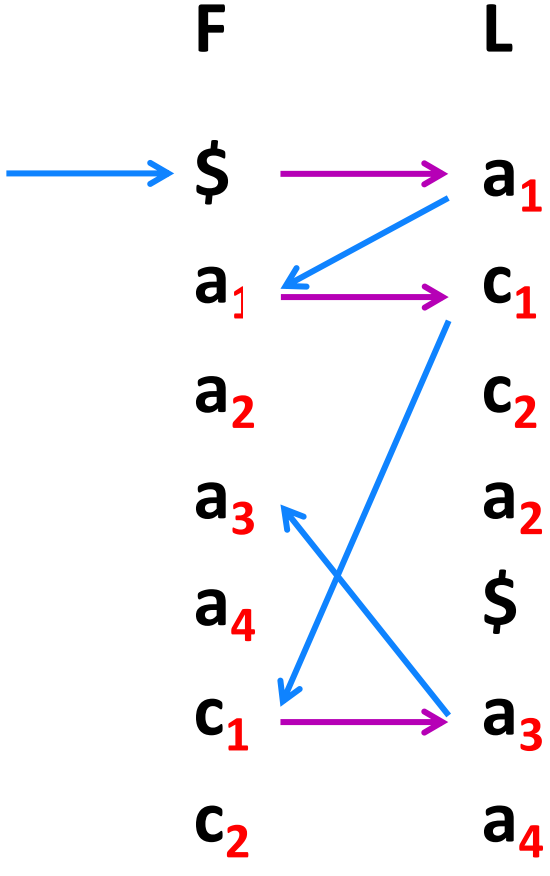
Find T knowing BWT(T):

T = c a $

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix



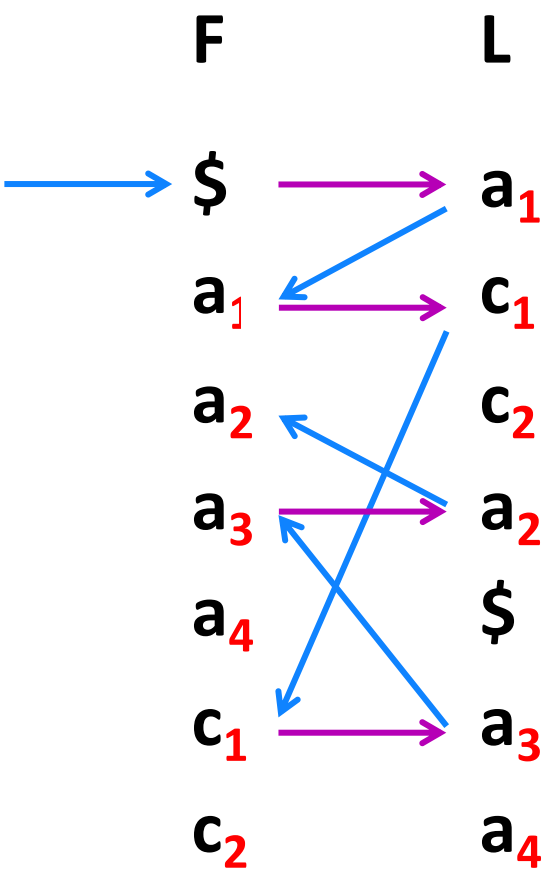
Find T knowing BWT(T):

T = a c a $

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix



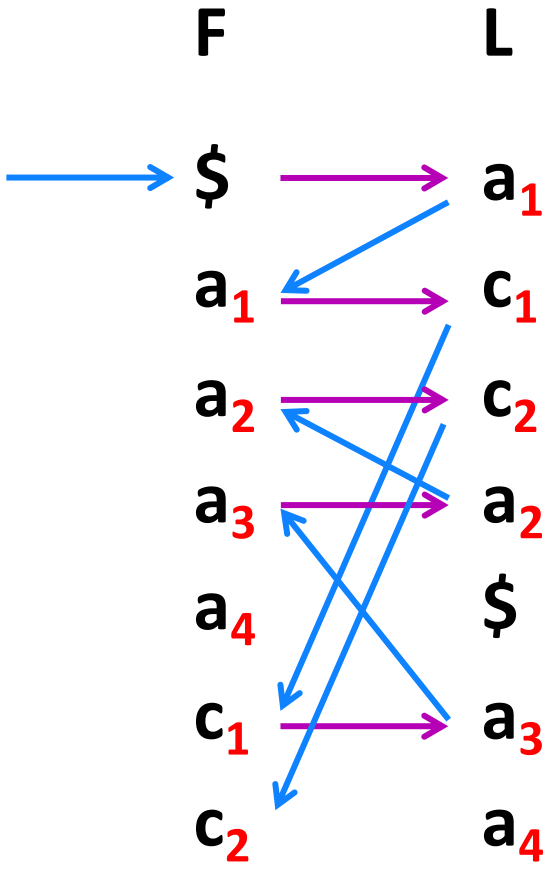
Find T knowing BWT(T):

T = a a c a $

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix



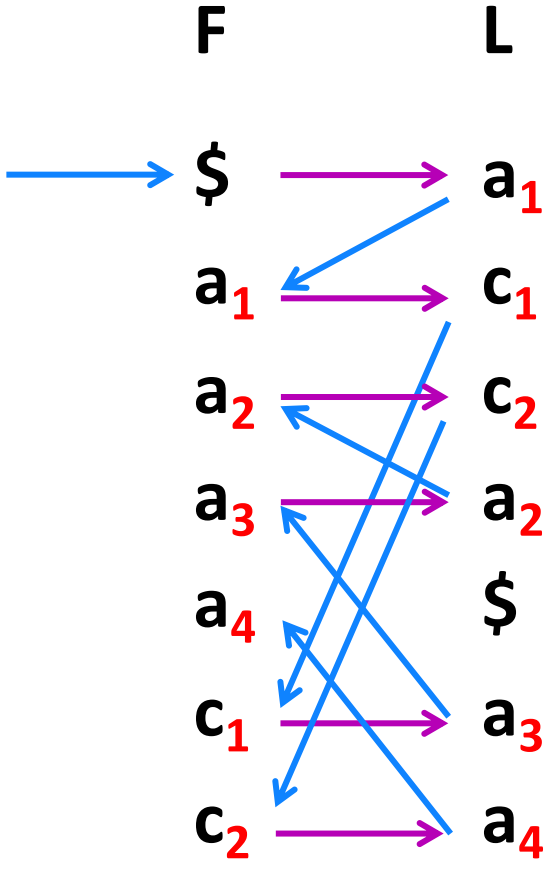
Find T knowing BWT(T):

T = c a a c a $

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix



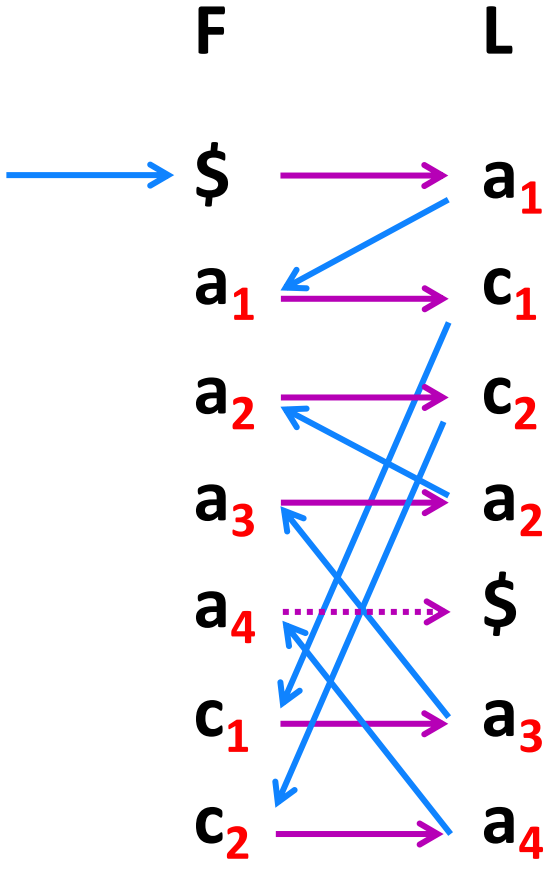
Find T knowing BWT(T):

T = a c a a c a $

BWT mappers: Find T knowing BWT(T)

$	a	c	a	a	c	a
a	$	a	c	a	a	c
a	a	c	a	$	a	c
a	c	a	$	a	c	a
a	c	a	a	c	a	$
c	a	$	a	c	a	a
c	a	a	c	a	$	a

BWT matrix



Find T knowing BWT(T):

T = a c a a c a $

BWT mappers: Example

reference: **a c a a c a**

read: **a c a**

F						L
$	a	c	a	a	c	a_4
a_4	$	a	c	a	a	c_2
a_2	a	c	a	$	a	c_1
a_3	c	a	$	a	c	a_2
a_1	c	a	a	c	a	$
c_2	a	$	a	c	a	a_3
c_1	a	a	c	a	$	a_1

BWT mappers: Example

reference: **a c a a c a**

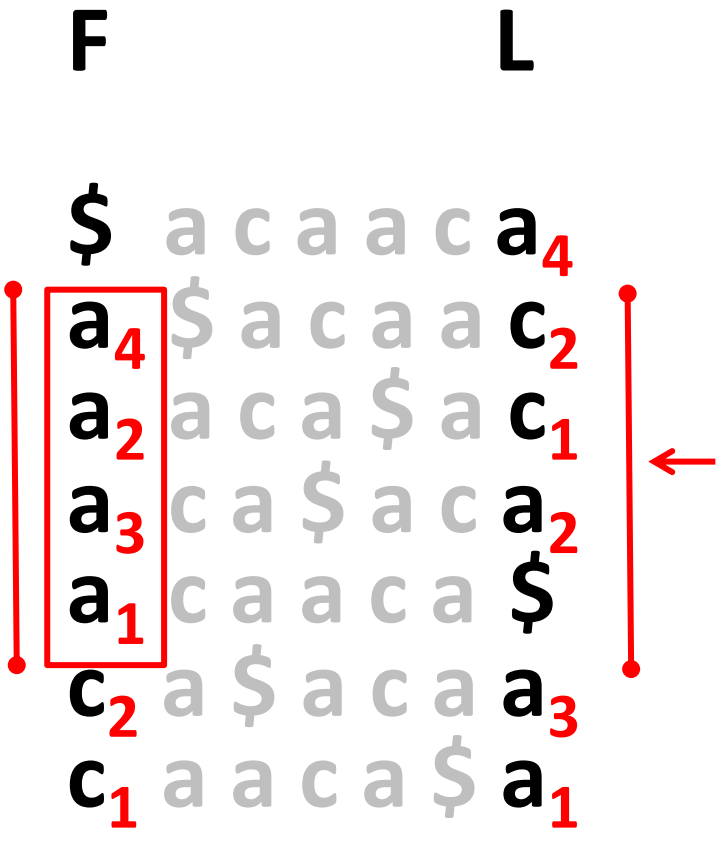
read: **a c a**

F		L
$	a c a a c	a_4
a_4	$ a c a a	c_2
a_2	a c a $ a	c_1
a_3	c a $ a c	a_2
a_1	c a a c a	$
c_2	a $ a c a	a_3
c_1	a a c a $	a_1

BWT mappers: Example

reference: a c a a c a

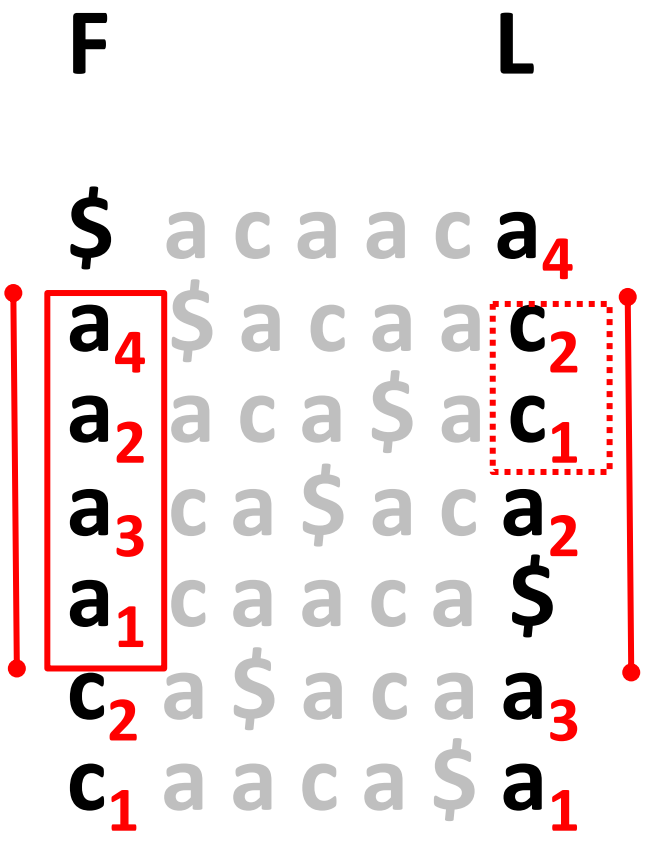
read: a c a



BWT mappers: Example

reference: a c a a c a

read: a c a



BWT mappers: Example

reference: a c a a c a

read: a c a

F						L
$	a	c	a	a	c	a_4
a_4	$	a	c	a	a	c_2
a_2	a	c	a	$	a	c_1
a_3	c	a	$	a	c	a_2
a_1	c	a	a	c	a	$
c_2	a	$	a	c	a	a_3
c_1	a	a	c	a	$	a_1

BWT mappers: Example

reference: a c a a c a

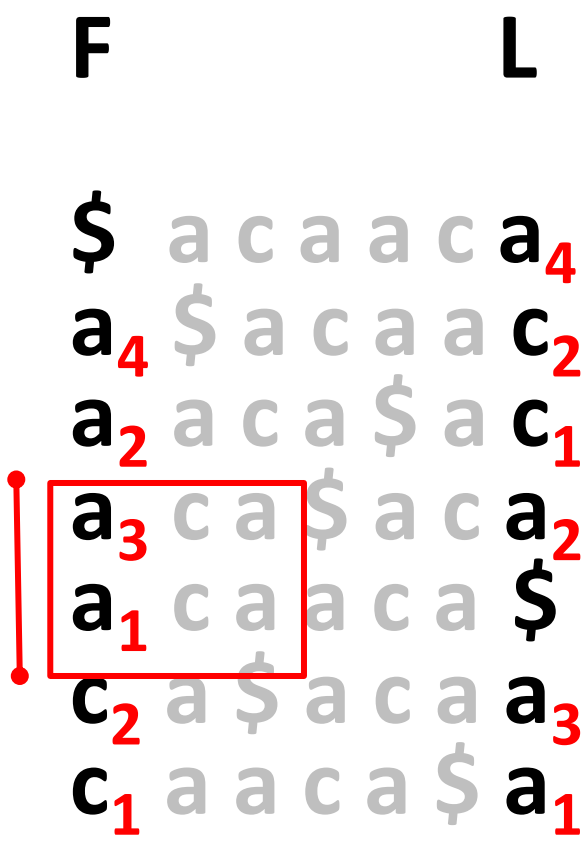
read: a c a

BWT mappers: Example

reference: **a c a a c a**

read: **a c a**

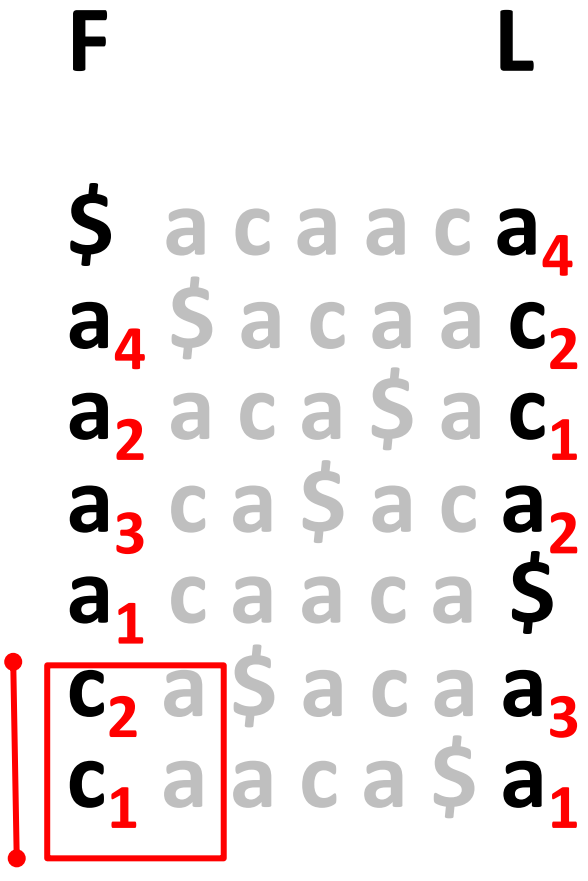
the read position is found



BWT mappers: Example

reference: a c a a c a

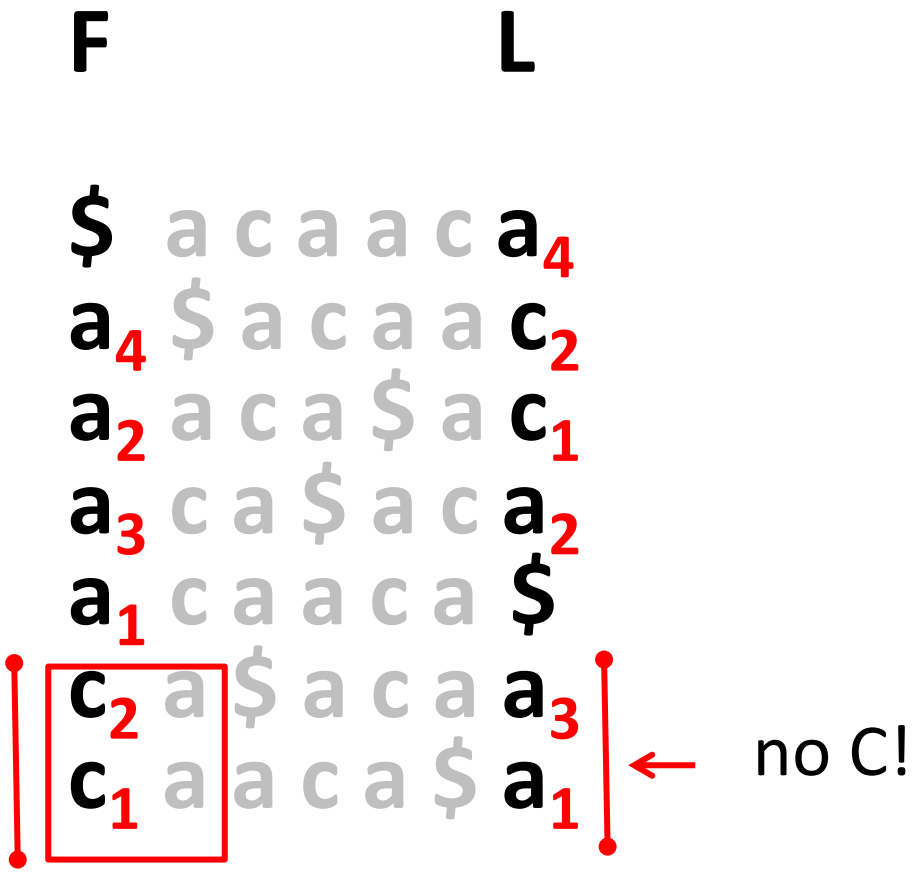
read: c c a



BWT mappers: Example

reference: a c a a c a

read: c c a



Choosing an Aligner

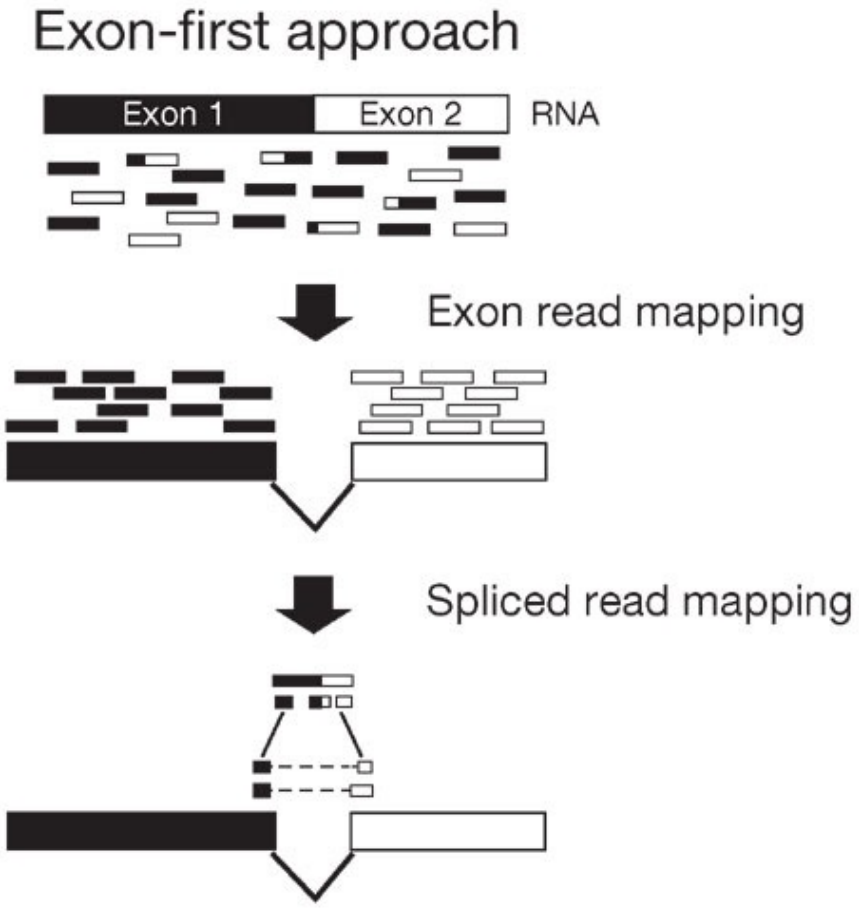
Compute resources vs sensitivity

Feature	Hash table index tools	BWT tools
Speed	Slower	Faster
Memory	Higher	Lower
Sensitivity	Higher	Lower

- For RNA-Seq Applications, BWT approach is mainly used

Spliced mappers: Exon first

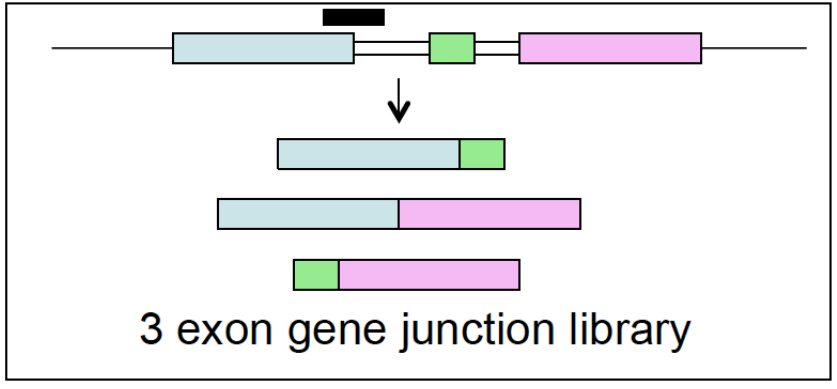
- Exonic reads are aligned first, the remaining reads are divided into smaller pieces and then mapped to the genome
- Fast, require less computational resources
- Bias towards un-spliced alignments
- MapSplice, SpliceMap, TopHat



Garber M. et al. 2011. Nature Methods 8(6):469-477

Mapping of RNA-seq reads

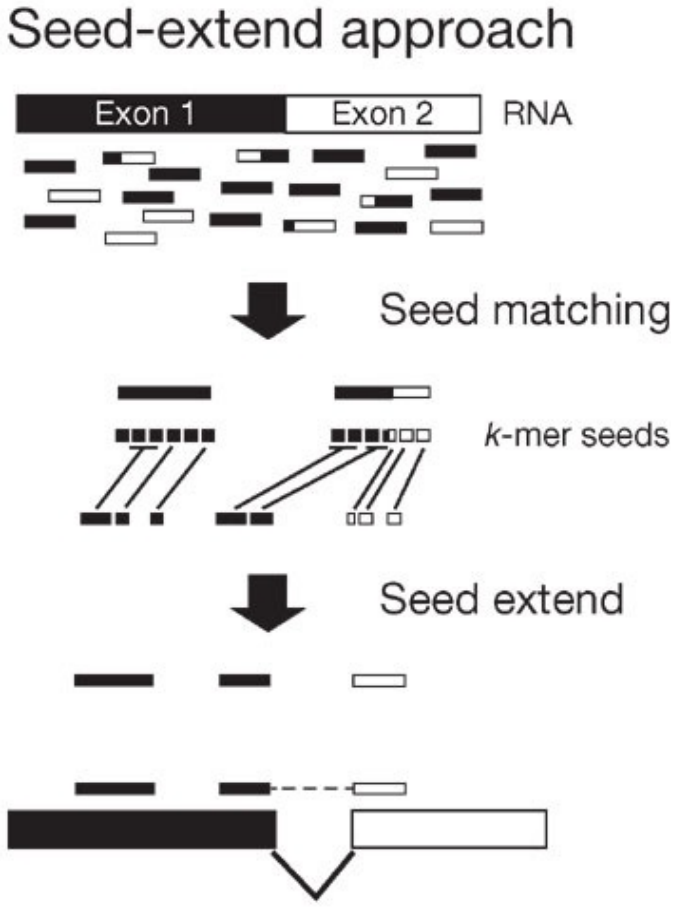
- Reference (fasta format)
  - Genome
  - Transcriptome
  - Junction library



- When mapped to the genome, RNA-Seq reads can span junctions
  - Spliced alignments
  - Example: Mouse retina 60 nt paired reads (41 of 91 Mio map to junctions)

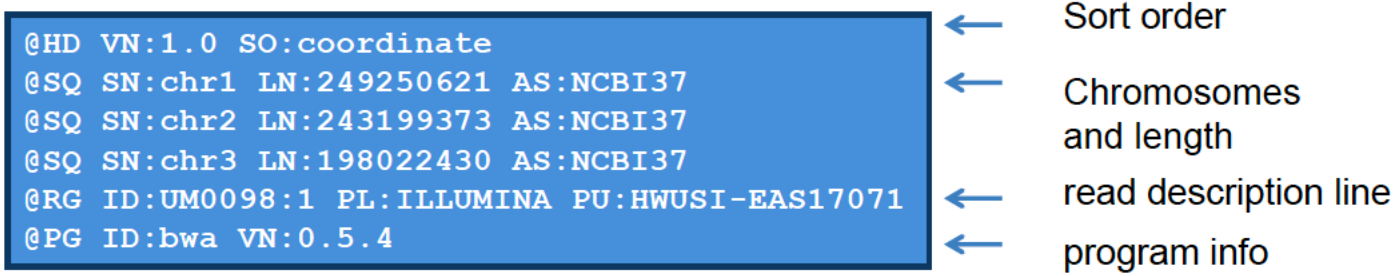
Spliced mappers: Seed extend

- Each read is divided into k-mers which are mapped to the genome using table lookup. Mapped k-mers are extended into larger alignments which may include gaps flanked by splice sites.
- Little bias, best placement of each read
- More tolerant to sequence differences
- GSNAP, PALMA



Garber M. et al. 2011. Nature Methods 8(6):469-477

SAM/BAM Header



SAM Format Read information

Field	Description	Alignment
QNAME	Query template name	HWI-ST1359:47:H1178ADXX:2:2201:20747:42719
FLAG	bitwise flag	99
RNAME	Reference name	chr13
POS	Reference position	28540518
MAPQ	Mapping quality	60
CIGAR	CIGAR string	101M
MRNM/RNEXT	Reference next read/mate	=
MPOS/PNEXT	Position next/read mate	28540656
ISIZE/TLEN	Template length	239

SAM Format Read Information

Field	Description	Alignment
SEQ	Sequence	CCTCAAACCCACTCCAGGCTGCCATT GGTACTCGCCCCTTTTTACAGATGAG GAAATGGAGAATCAGACCGGGTCACG CAGATAGTATCAGGCGGGGTTGG
QUAL	Base qualities	7783>446=;@6;B;@B56/>=7>;8 <;?@0>A>=;;7A==@@9BB<;497> 565179==2864:>C=083&&42;37 69<8==>=;=>>D=768==8:79
TAGs	Tags	X0:i:1 X1:i:0 AM:i:37 NM:i:0 SM:i:37 XM:i:0 XO:i:0 MQ:i:60 XT:A:U

SAM Format FLAG Values

Flags

Flag	Description
0x0001	the read is paired in sequencing, no matter whether it is mapped in a pair
0x0002	the read is mapped in a proper pair (depends on the protocol, normally inferred during alignment) [1]
0x0004	the query sequence itself is unmapped
0x0008	the mate is unmapped [1]
0x0010	strand of the query (0 for forward; 1 for reverse strand)
0x0020	strand of the mate [1]
0x0040	the read is the first read in a pair [1,2]
0x0080	the read is the second read in a pair [1,2]
0x0100	the alignment is not primary (a read having split hits may have multiple primary alignment records)
0x0200	the read fails platform/vendor quality checks
0x0400	the read is either a PCR duplicate or an optical duplicate

https://broadinstitute.github.io/picard/explain-flags.html

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

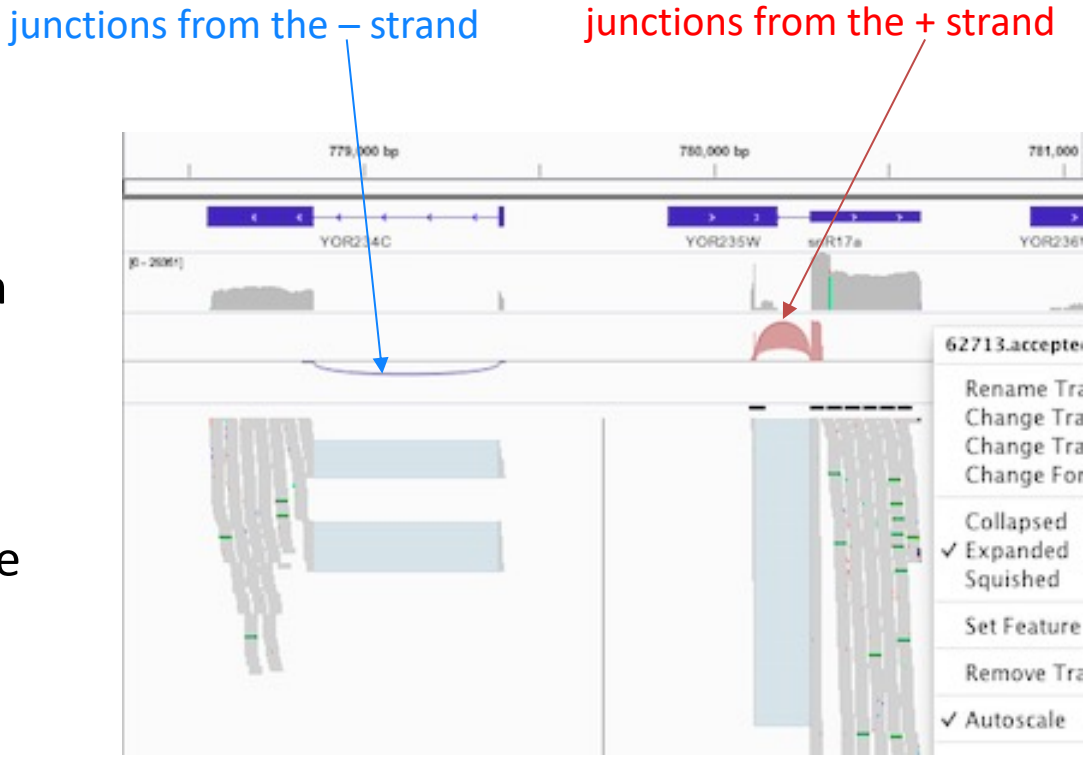
```
Match/mismatch,indels

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
```

Splice junction track

- Compute dynamically from alignment data (must be from spliced aligner)
- A splicing event is drawn when at least a single read splits across two exons
- The height and thickness of the arc are proportional to the coverage depth



Inversion, duplication, translocation

- Color alignment by read orientation