Alignment of Next-Generation Sequencing Data

Nadia Lanman
HPC for Life Sciences
2019

PURDUE UNIVERSITY
CENTER FOR CANCER RESEARCH

What is sequence alignment?

- A way of arranging sequences of DNA, RNA, or protein to identify regions of similarity
  - Similarity may be a consequence of functional, structural, or evolutionary relationships between sequences
  - In the case of NextGen sequencing, alignment identifies where fragments which were sequenced are derived from (e.g. which gene or transcript)
- Two types of alignment: local and global

```
FGFR1_10 ------------------------------------------------------------------------------------------------------------------------------------
FGFR1_12 ------------------------------------------------------------------------------------------------------------------------------------
FGFR1_15 MWSWKCLLFWAVLVTATLCTARPSPTLPEQ-------------------------------------------------------------------------DALPSSEDDDDDDDSSSEEKETDNTKPN--P     59
 FGFR1_6 MWSWKCLLFWAVLVTATLCTARPSPTLPEQ-------------------------------------------------------------------------DALPSSEDDDDDDDSSSEEKETDNTKPNRMP     61
 FGFR1_8 MWSWKCLLFWAVLVTATLCTARPSPTLPEQ-------------------------------------------------------------------------DALPSSEDDDDDDDSSSEEKETDNTKPNRMP     61
FGFR1_14 MWSWKCLLFWAVLVTATLCTARPSPTLPEQAQPWGAPVEVESFLVHPGDLLQLRCRLRDDVQSINWLRDGVQLAESNRTRITGEEVEVQDSVPADSGLYACVTSSPSGSDTTYFSVNVSDALPSSEDDDDDDDSSSEEKETDNTKPN--P    148
 FGFR1_4 MWSWKCLLFWAVLVTATLCTARPSPTLPEQAQPWGAPVEVESFLVHPGDLLQLRCRLRDDVQSINWLRDGVQLAESNRTRITGEEVEVQDSVPADSGLYACVTSSPSGSDTTYFSVNVSDALPSSEDDDDDDDSSSEEKETDNTKPNRMP    150
   FGFR1 MWSWKCLLFWAVLVTATLCTARPSPTLPEQAQPWGAPVEVESFLVHPGDLLQLRCRLRDDVQSINWLRDGVQLAESNRTRITGEEVEVQDSVPADSGLYACVTSSPSGSDTTYFSVNVSDALPSSEDDDDDDDSSSEEKETDNTKPNRMP    150
FGFR1_11 ------------------------------------------------------------------------------------------------------------------------------------
FGFR1_13 ------------------------------------------------------------------------------------------------------------------------------------
 FGFR1_7 MWSWKCLLFWAVLVTATLCTARPSPTLPEQ-------------------------------------------------------------------------DALPSSEDDDDDDDSSSEEKETDNTKPNRMP     61
 FGFR1_9 MWSWKCLLFWAVLVTATLCTARPSPTLPEQ-------------------------------------------------------------------------DALPSSEDDDDDDDSSSEEKETDNTKPNRMP     61
FGFR1_17 MWSWKCLLFWAVLVTATLCTARPSPTLPEQ-------------------------------------------------------------------------DALPSSEDDDDDDDSSSEEKETDNTKPNRMP     61
FGFR1_18 MWSWKCLLFWAVLVTATLCTARPSPTLPEQ-------------------------------------------------------------------------DALPSSEDDDDDDDSSSEEKETDNTKPN--P     59
 FGFR1_2 MWSWKCLLFWAVLVTATLCTARPSPTLPEQAQPWGAPVEVESFLVHPGDLLQLRCRLRDDVQSINWLRDGVQLAESNRTRITGEEVEVQDSVPADSGLYACVTSSPSGSDTTYFSVNVSDALPSSEDDDDDDDSSSEEKETDNTKPNRMP    150
 FGFR1_5 MWSWKCLLFWAVLVTATLCTARPSPTLPEQAQPWGAPVEVESFLVHPGDLLQLRCRLRDDVQSINWLRDGVQLAESNRTRITGEEVEVQDSVPADSGLYACVTSSPSGSDTTYFSVNVSDALPSSEDDDDDDDSSSEEKETDNTKPNRMP    150
 FGFR1_3 MWSWKCLLFWAVLVTATLCTARPSPTLPEQA------------------------------------------------------------------------------------------------     31
FGFR1_16 MWSWKCLLFWAVLVTATLCTARPSPTLPEQAQ-----------------------------------------------------------------------------------------------     32
   ruler 1.......10........20........30........40........50........60........70........80........90.......100.......110.......120.......130.......140.......150
```

http://www-personal.umich.edu/~lpt/fgf/fgfrseq.htm

Global vs Local Alignment

- Global aligners try to align all provided sequence end to end
- Local aligners try to find regions of similarity *within* each provided sequence (match your query with a substring of your subject/target)

Local Alignment

```
Target Sequence
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
             |||| |||||| |||||||||||||||
Query Sequence 5' TACTCACGGATGAGGTACTTTAGAGGC 3'
```

Global Alignment

gap mismatch

```
Target Sequence
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
   |||||||||||    |||||||  ||||||||||||||| |||||||
5' ACTACTAGATT----ACGGATC--GTACTTTAGAGGCTAGCAACCA 3'
Query Sequence
```

Alignment Example

Raw sequences:
A G A T G and G A T T G

2 matches, 0 gaps

```
A G A T G
      | |
G A T T G
```

4 matches, 1 insertion

```
A G A - T G .
  | |   | |
. G A T T G
```

4 matches, 1 insertion

```
A G A T - G .
  | | |   |
. G A T T G
```

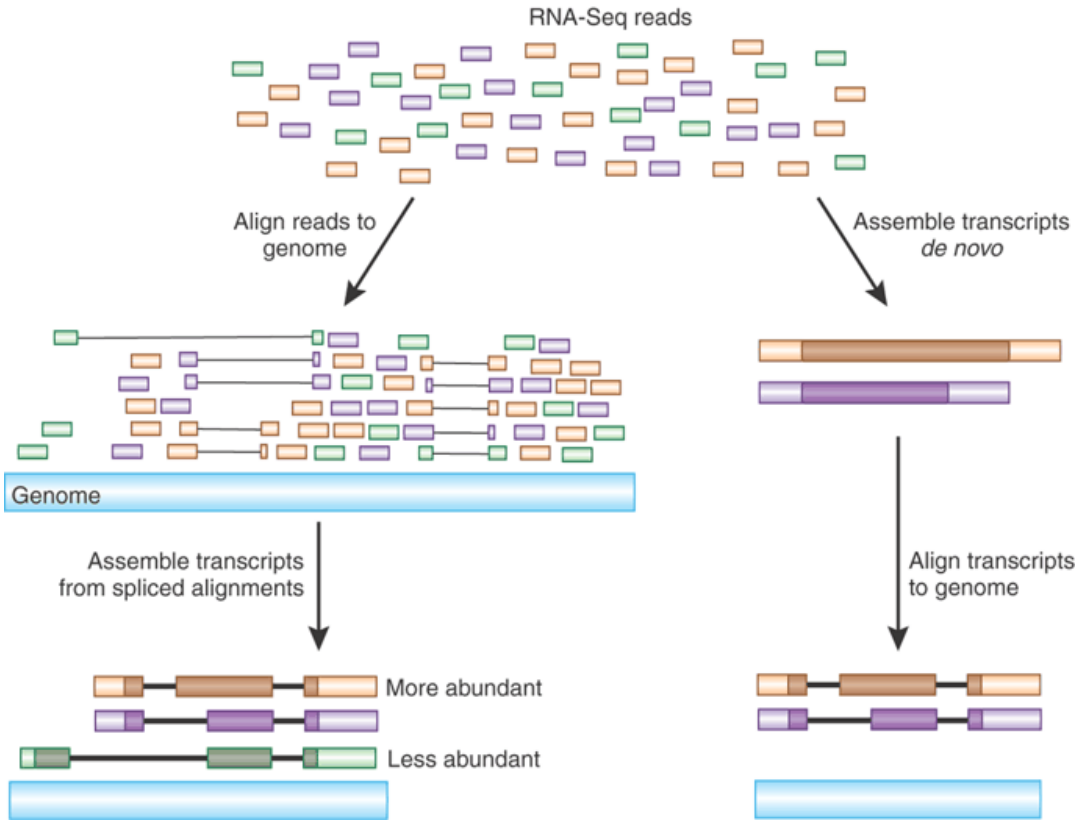
3 matches, 2 end gaps

```
A G A T G .
  | | |
. G A T T G
```

NGS read alignment

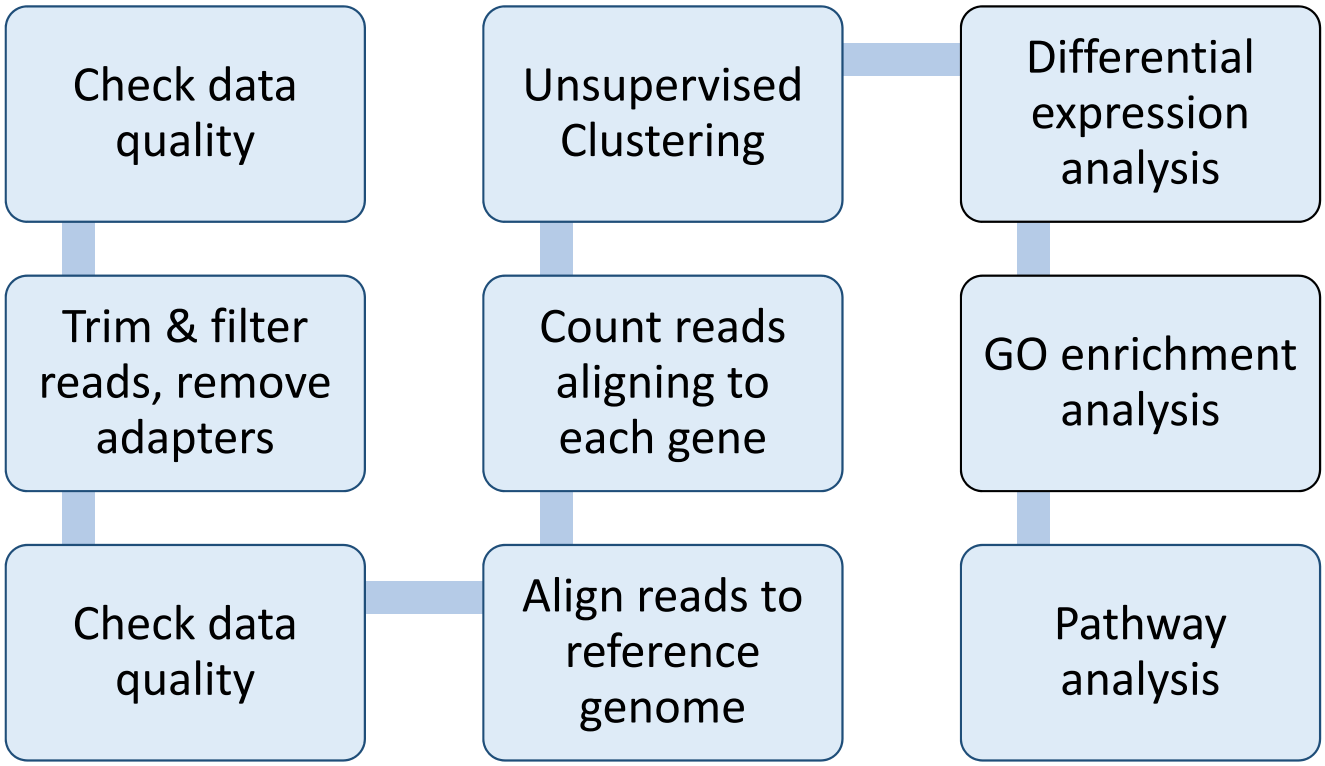
- Allows us to determine where sequence fragments ("reads") came from

- Quantification allows us to address relevant questions
  - How do samples differ from the reference genome
  - Which genes or isoforms are differentially expressed



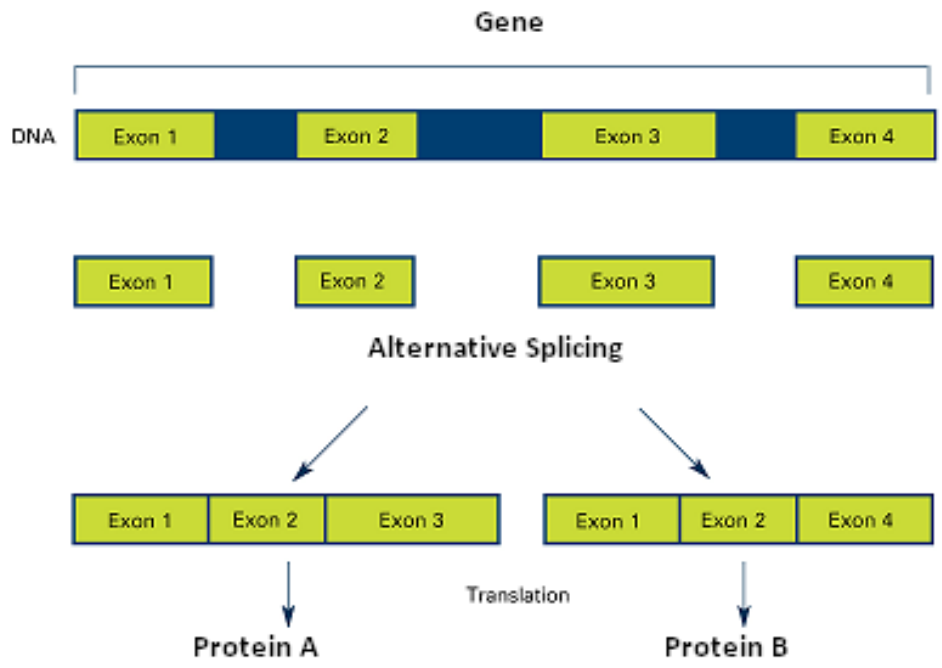
Haas *et al*, 2010, *Nature*.

Standard Differential Expression Analysis



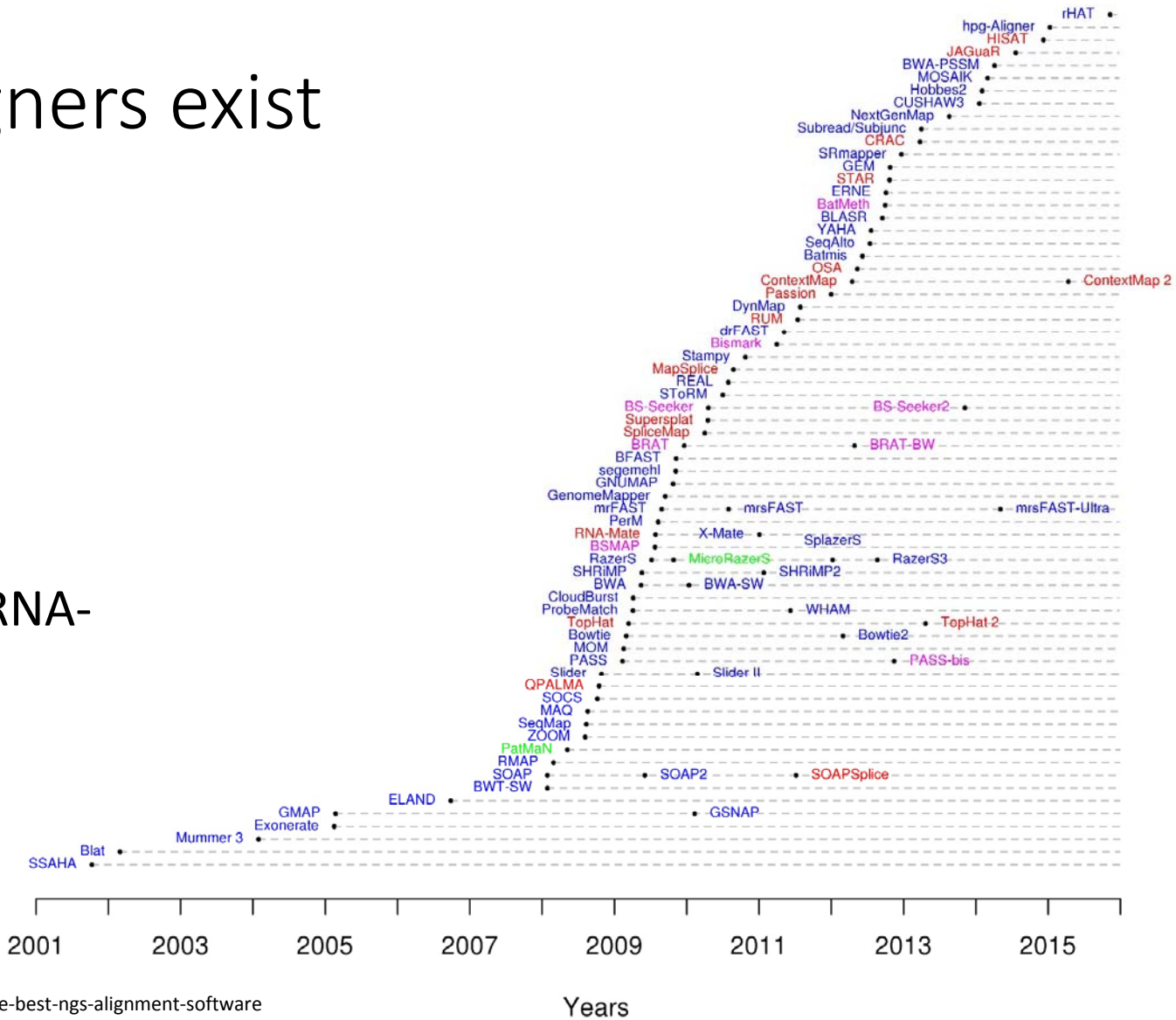
Challenges of NGS Read Alignment

- Alternative splicing
- Often must align billions of reads
- Reads will have some mismatches = an approximate matching problem
  - Sequencing errors
  - True biological variation
- Eukaryotic genome are rich in repetitive regions`
- New methods had to be developed which were specific to NGS sequence alignment
- Multi-mapped reads



More than 90 aligners exist

- Bowtie2
- BWA
- STAR
- Tophat2
- Pseudo Aligners for RNA-seq quantification
  - Kalliso
  - Salmon
  - Sailfish



https://www.ecseq.com/support/ngs/what-is-the-best-ngs-alignment-software

How to align a huge amount of data?

Two general ideas

1. Filtering approaches – exclude a large region of the reference where no approximate match can be found
   - Pidgeonhole lemma or *q*-gram lemma
2. Indexing – involve preprocessing the reference, set of reads, or both. Queries can be conducted without scanning the entire reference genome.
   - String indices commonly used:
     - Suffix array
       - Uses more memory but very fast to query
     - Enhanced suffix array
     - FM-index (a data structure based on the Burrows-Wheeler transform)
       - Make very good use of memory and is also quite fast

Burrows-Wheeler Aligners

- Used with FM-index (Ferragina & Manzini, 2000) allows efficient finding of substring matches within compressed text
- Sub-linear
- Lower memory footprint, fast execution.
- Rearranges a character string into runs of similar characters
  - Makes the string very easy to compress if it has runs of repeated characters – very useful for DNA strings!
- Is reversible

Burrows, Michael; Wheeler, David J. (1994), A block sorting lossless data compression algorithm, Technical Report 124, Digital Equipment Corporation

Burrows-Wheeler Aligners

- Burrows-Wheeler Transform encodes data so it is easier to compress
- Burrows-Wheeler transform of the word BANANA

Transformation				
Input	All Rotations	Sorting All Rows in Alphabetical Order by their first letters	Taking Last Column	Output Last Column
^BANANA|	^BANANA| |^BANANA A|^BANAN NA|^BANA ANA|^BAN NANA|^BA ANANA|^B BANANA|^	ANANA|^B ANA|^BAN A|^BANAN BANANA|^ NANA|^BA NA|^BANA ^BANANA| |^BANANA	ANANA|^B ANA|^BAN A|^BANAN BANANA|^ NANA|^BA NA|^BANA ^BANANA| |^BANANA	BNN^AA|A

BWA

- Very fast, can do gapped alignments
  - bio-bwa.sourceforge.net
- Can be run without seeding and then will find all matches within a given edit distance
- Long read aligner (>200 bp) within BWA is also fast
- Actively maintained and has strong user community

Li and Durbin, 2009, Bioinformatics
Li and Durbin, 2010, Bioinformatics

BWA

- Burrows-Wheeler transform algorithm with FM-index using suffix arrays.
  - Need to create a genome index
- BWA can map low-divergent sequences against a large reference genome, such as the human genome.
- It consists of three algorithms:
  - BWA-backtrack (Illumina sequence reads up to 100bp)
  - BWA-SW (more sensitive when alignment gaps are frequent)
  - BWA-MEM (maximum exact matches)
- BWA SW and MEM can map longer sequences (70bp to Mbp) and share similar features such as long-read support and split alignment, but BWA-MEM, which is the latest, is generally recommended for high-quality queries as it is faster and more accurate.
- BWA-MEM also has better performance than BWA-backtrack for 70-100bp Illumina reads.

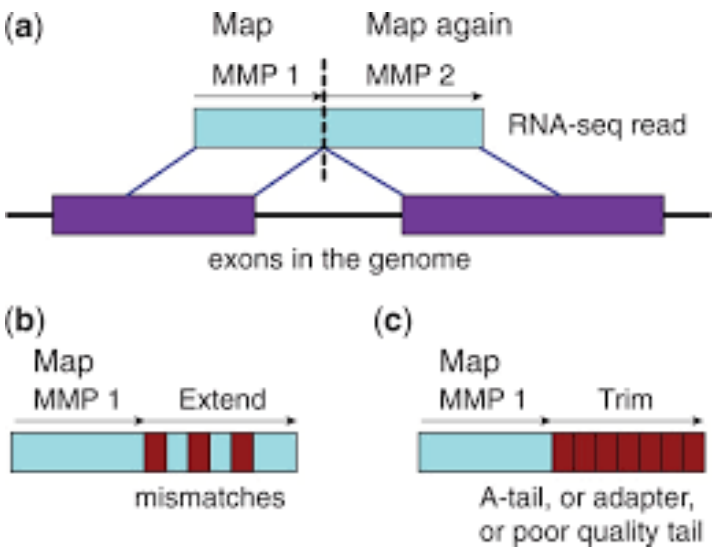
Bowtie2

- Bowtie (now Bowtie2) is part of a suite of tools for analyzing RNA-seq data (Bowtie, Tophat, Cufflinks, CummeRbund)
  - http://bowtie-bio.sourceforge.net
- Bowtie2 is a Burrows-Wheeler Transform (BWT) aligner and handles reads longer than 50 nt.
  - Need to prepare a genome index
- Given a reference and a set of reads, this method reports at least one good local alignment for each read if one exists.
- Bowtie (now Bowtie2) is faster than BWA for some alignments but is sometimes less sensitive than BWA
  - Can view all sorts of arguments for one or the other on SeqAnswers.com

Langmead and Salzberg, 2012, Nat. Methods

STAR

- Splicing Transcripts Alignment to a Reference
- Two steps: Seed searching and clustering/stitching/scoring (find MMP -maximal mappable prefix using Suffix Arrays)
- Fast splice aware aligner, high memory (RAM) footprint
- Can detect chimeric transcripts
- Generate indices using a reference genome fasta, and annotation gtf or gff from Ensembl/UCSC.



Dobin et al., 2013 Bioinformatics

Alignment Concepts and Terminology

- Edit distance

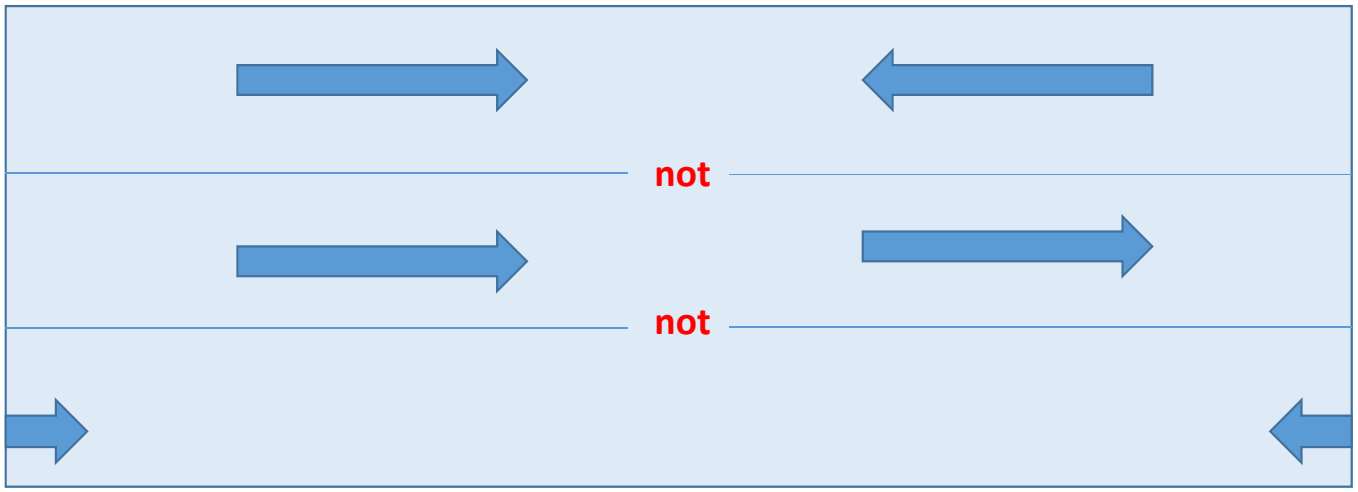
```
ATCGACCGCGCTAA-TATTAGTC...
  CGACGGCGCTAACTATTA
```

edit distance = 2

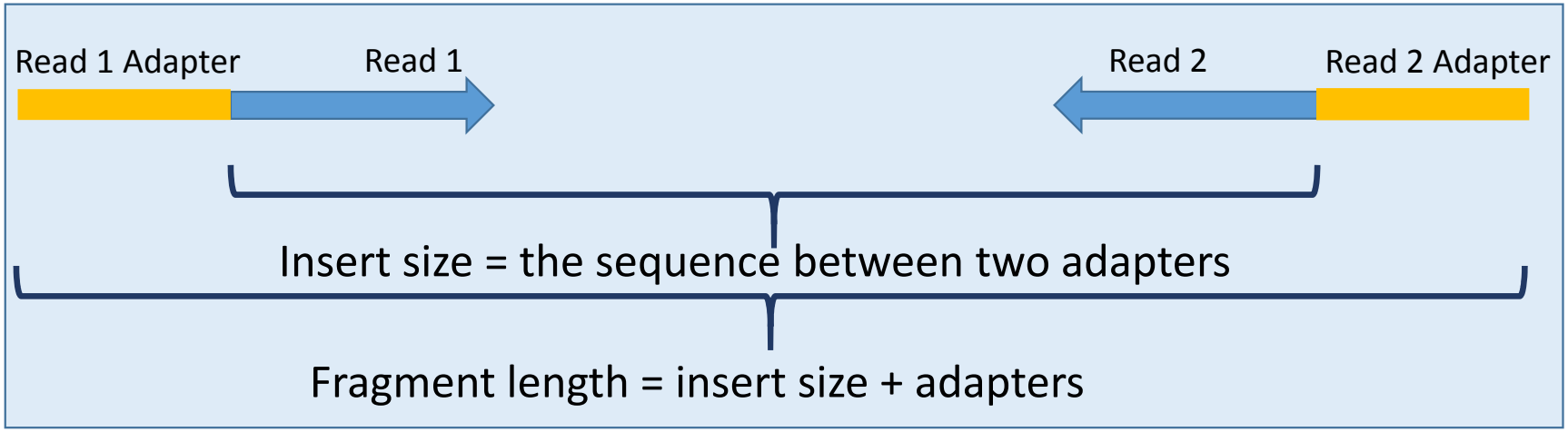
- Mapping quality: the confidence that the read is correctly mapped to the genomic coordinates

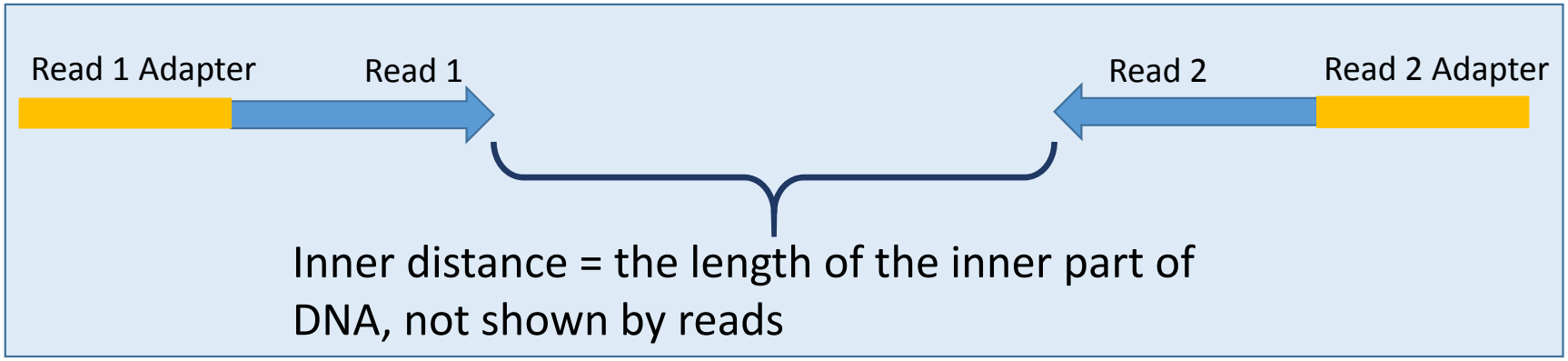
  Probability mapping is incorrect = $10^{-MQ/10}$

- Proper pairs



Alignment Concepts and Terminology





Note: when you align RNA reads to a DNA genome, set the insert size fairly liberally to allow for introns. Ex 200,000 for the human genome

Insert size

Find insert size and mean and median

```
bamtools stats -i foo.bam –insert
```

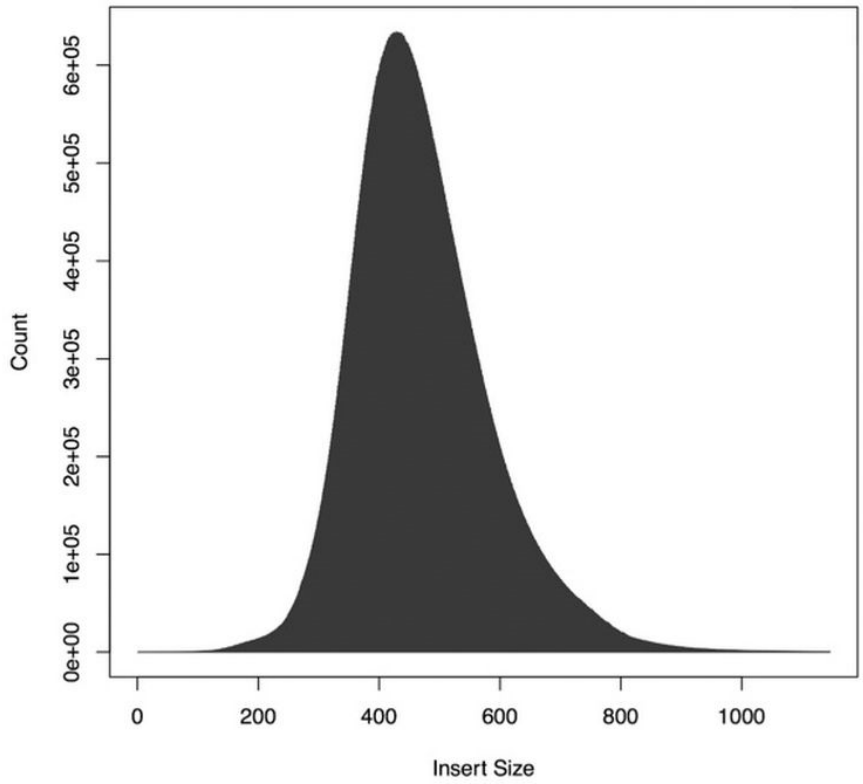
Or

```
samtools stats --insert-size foo.bam
```

Or

```
bbmap.sh ref=reference.fasta in1=read1.fq
in2=read2.fq ihist=ihist_mapping.txt
out=mapped.sam maxindel=200000
```

Or one of the many other options available (just Google).....



Alignment Concepts and Terminology

Multimapped reads: reads that align *equally well* to more than one reference location

-How multimapped reads are handled depends on parameter settings selected, program, application, and how many time reads map to multiple places

Duplicate reads: reads that arise from the same library fragment

-How duplicates are handled also depends on parameter settings selected, program, application
-Can arise during library prep (PCR duplicates) or during colony formation (optical duplicates)

Mappability

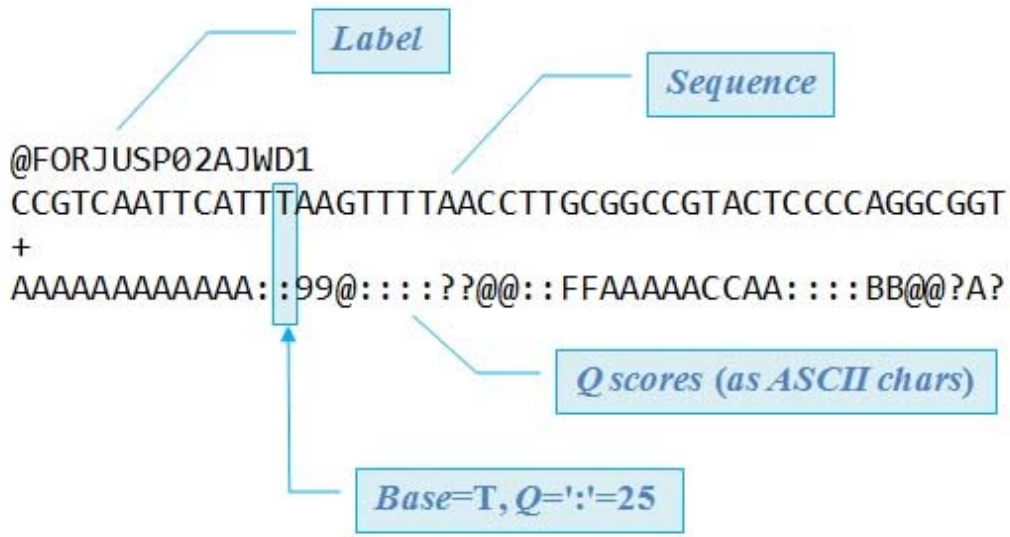
- Not all of the genome is 'available' for mapping when reads are aligned to the unmasked genome.
- Uniqueness: This is a direct measure of sequence uniqueness throughout the reference genome.

		Nonrepetitive sequence		Mappable sequence	
Organism	**Genome size (Mb)**	**Size (Mb)**	**Percentage**	**Size (Mb)**	**Percentage**
Caenorhabditis elegans	100.28	87.01	86.8%	93.26	93.0%
Drosophila melanogaster	168.74	117.45	69.6%	121.40	71.9%
Mus musculus	2,654.91	1,438.61	54.2%	2,150.57	81.0%
Homo sapiens	3,080.44	1,462.69	47.5%	2,451.96	79.6%

Rozowsky, (2009)

FASTQ Format

- Text files containing header, sequence, and quality information
- Quality information is in ACII format
- $Q = -10 \log_{10} p$



https://www.drive5.com/usearch/manual/fastq_files.html

Phred quality scores are logarithmically linked to error probabilities

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

BAM and SAM format

- SAM file is a tab-delimited text file that contains sequence alignment information
- BAM files are simply the binary version (compressed and indexed version)of SAM files → they are smaller
- Example:

Header lines (begin with "@")

```
@SQ     SN:chrM LN:16299
@SQ     SN:chrUn_random LN:5900358
@SQ     SN:chrX LN:166650296
@SQ     SN:chrX_random  LN:1785075
@SQ     SN:chrY LN:15902555
@SQ     SN:chrY_random  LN:58682461
HWI-EAS038:6:1:23:122#0 4       *       0       0       *       *       0       0       TAGCCTTGATGTTTACCTATTGTATCAAAGGGC       DJYMXLTPKDPOXYBBBBBBBBBBBBBBBBBBB
B
HWI-EAS038:6:1:25:283#0 0       chr14   27002726        0       33M     *       0       0       AGAGACCCAGGAAATTGAAGTCAGAGCAGTTAG       abaa_Z_`X]PM^BBBBBBBBBB
BBBBBBBBB       XT:A:R  NM:i:1  X0:i:3  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:10T22
HWI-EAS038:6:1:26:649#0 0       chr9    27884899        37      33M     *       0       0       CCTTTCTTTTGTCTACTCCTTTCCTCTTGGTAT       abbaabbbbbb```aZ\`a\aa[]
_QWaa`YXS       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:33
HWI-EAS038:6:1:30:918#0 16      chr17   95265601        0       33M     *       0       0       GTGTTTATCAGTCCCAAGGCCACTAGAGGCTTG       BBBBBBBBBBBBBB[[`\aaZaa
__aaaa`a`       XT:A:R  NM:i:2  X0:i:3  X1:i:0  XM:i:2  XO:i:0  XG:i:0  MD:Z:3G8T20
HWI-EAS038:6:1:32:1507#0        16      chr13   57505480        37      33M     *       0       0       CGGAGCTGGTGGTAGACATTGTGTGCTGCCTAG       \Z]W_[`]Z\[]^^Z^T
`bbab_[W\_]bb_W_b       XT:A:U  NM:i:0  X0:i:1  X1:i:0  XM:i:0  XO:i:0  XG:i:0  MD:Z:33
HWI-EAS038:6:1:32:298#0 4       *       0       0       *       *       0       0       TATAATAAAAATGACATTTTATTAAATACGCCT       `^aaa__\]^[SBBBBBBBBBBBBBBBBBBB
B
HWI-EAS038:6:1:32:1938#0        0       chr7    65636851        37      33M     *       0       0       TTTATATTTCTCCCCTTATCATTCCATTTTTTT       ]aa^^X\`YQ\Y[^U\Y
ZHMXXWZEVFQ][BBBB       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:31G1
HWI-EAS038:6:1:32:861#0 4       *       0       0       *       *       0       0       TGCATTCTAAGTTGGTTTAATATAAATCAACAT       ]bUSJGKHWuK_\BBBBBBBBBBBBBBBBBB
B
HWI-EAS038:6:1:32:1814#0        0       chr2    98506740        0       33M     *       0       0       CCACTTGACGACTTCAAAAATGACGAAATCACT       W^R^X`]Z]a]XZ]aZ
W]PYVV\YRW[SUZSST       XT:A:R  NM:i:1  X0:i:12 X1:i:44 XM:i:1  XO:i:0  XG:i:0  MD:Z:14G18
HWI-EAS038:6:1:34:2002#0        0       chr10   97252488        37      33M     *       0       0       CCTAGATTCCTTAGGGTATAAAAGGAGGAGAGC       _a`_ba]_Oa]aV[`a
OHDT^_BBBBBBBBBBB       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:29T3
HWI-EAS038:6:1:37:667#0 0       chrX    90652654        37      33M     *       0       0       CAAGTCCAAAAATTCCTTGAAAAATTTCACAAT       Y`_TOMPT^^_[PUWOJQLQQYW]
BBBBBBBBB       XT:A:U  NM:i:1  X0:i:1  X1:i:0  XM:i:1  XO:i:0  XG:i:0  MD:Z:19C13
HWI-EAS038:6:1:37:1236#0        4       *       0       0       *       *       0       0       ATGATTTCTTGTTGTGTATCACTATTCTAGGGG       _Q\LYBBBBBBBBBBBBBBBBBBB
BBBBBBBBB
HWI-EAS038:6:1:37:262#0 16      chr2    3386587 23      33M     *       0       0       TCTAGTACCCACATGGTGCAAGGAGAGAACCAA       BB]Z[LFTXX]TZYQRXHJUOISU\X]_[UO]
```

Alignment section

Understanding SAM flags

Decoding SAM flags

This utility makes it easy to identify what are the properties of a read based on its SAM flag value, or conversely, to find what the SAM Flag value would be for a given combination of properties.

To decode a given SAM flag value, just enter the number in the field below. The encoded properties will be listed under Summary below, to the right.

SAM Flag: 256 Explain

Switch to mate Toggle first in pair / second in pair

Find SAM flag by property:

To find out what the SAM flag value would be for a given combination of properties, tick the boxes for those that you'd like to include. The flag value will be shown in the SAM Flag field above.

- [ ] read paired
- [ ] read mapped in proper pair
- [ ] read unmapped
- [ ] mate unmapped
- [ ] read reverse strand
- [ ] mate reverse strand
- [ ] first in pair
- [ ] second in pair
- [x] not primary alignment
- [ ] read fails platform/vendor quality checks
- [ ] read is PCR or optical duplicate
- [ ] supplementary alignment

Summary:

not primary alignment (0x100)

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

Processing SAM/BAM files

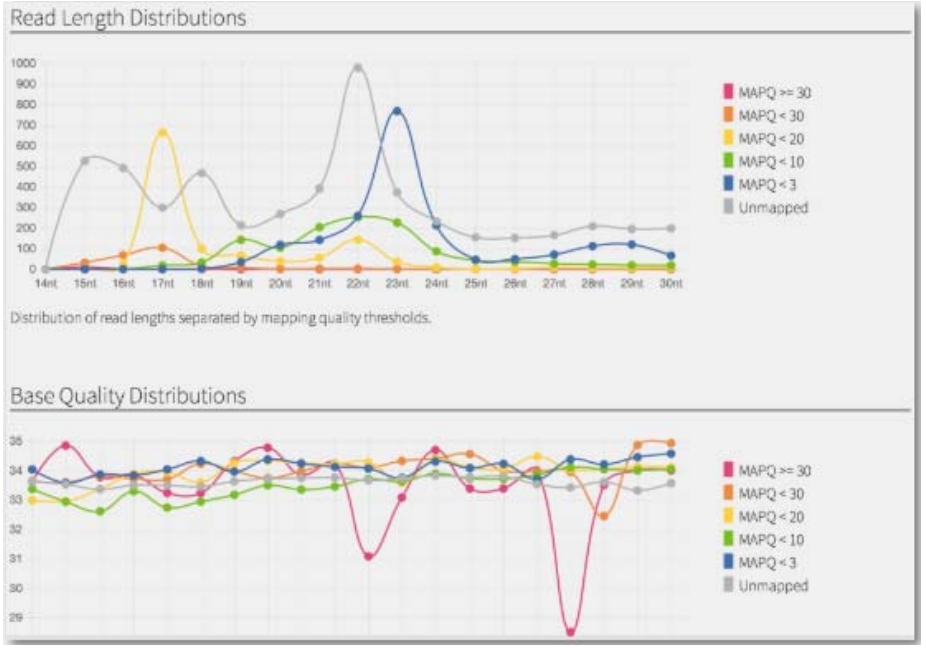
- SAMtools is a software suite which provides various utilities for manipulating alignments in SAM format
  - Sorting, merging, indexing, and generating alignments in a per-position format
  - import: SAM-to-BAM conversion
  - view: BAM-to-SAM conversion and sub alignment retrieval
  - sort: sorting alignment
  - merge: merging multiple sorted alignments
  - index: indexing sorted alignment
  - faidx: FASTA indexing and subsequence retrieval
  - tview: text alignment viewer
  - pileup: generating position-based output and consensus/indel calling
  - RSamTools package in Bioconductor allows similar functionality in R.

Processing SAM/BAM files

- Picard is a collection of Java-based command-line utilities that manipulate sequencing data and formats such as SAM/BAM/CRAM and VCF. It has a Java API (SAM-JDK) for creating new programs that read and write SAM files.
- Currently contains 86 different tools
- Well-supported and frequently utilized
- The mark duplicate function is particularly useful.

SAMstat for mapping QC

- SAMstat is a C program that plots nucleotide overrepresentation and other statistics in mapped and unmapped reads and helps understand the relationship between potential protocol biases and poor mapping.
- It reports statistics for unmapped, poorly and accurately mapped reads separately. This allows for identification of a variety of problems, such as remaining linker and adaptor sequences, causing poor mapping




```
samstat <file.sam>  <file.bam>  <file.fa>  <file.fq> ....
```

For each input file SAMStat will create a single html page named after the input file name plus a dot html suffix.

Lassmann et al., 2011, Bioinformatics.

Quantifying Genes

- Now that alignment has been performed, reads can be counted and a matrix created, quantifying genes and transcripts
- Many techniques exist
  - HTSeq-Count
  - FeatureCounts
  - RSEM
  - Pseudoaligners

RSEM

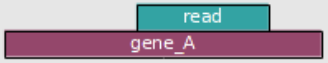
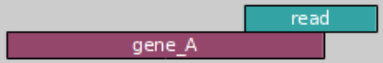
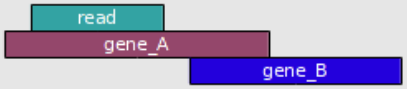
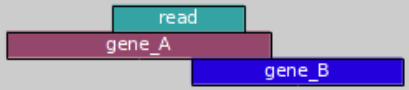
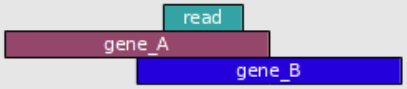
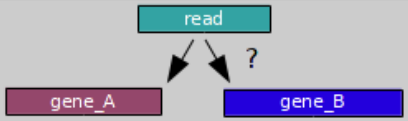
- RNA-seq Expectation Maximization
- RNA-seq gene and isoform quantification program
- Uses Bowtie, Bowtie2, or STAR to perform mapping
- Can take a *de novo* assembled transcriptome as input
- Uses EM (expectation-maximization) algorithm to estimate maximum likelihood expression levels (including dealing with multireads)
- Must index genome or transcriptome
- Outputs both gene and isoform quantification results

Li et al, BMC Bioinformatics, 2011

HTSeq-Count

- Input: SAM or BAM file and a GTF or GFF file with annotated gene models
- Output: counts for each gene the number of reads that overlap with its exons
- Only reads mapping unambiguously to a single gene are counted
- Reads aligned to multiple positions or overlapping with more than one gene are discarded
- Does not include "end" location of GTF files in the feature interval

Anders S, Pyl PT, Huber W. Bioinformatics. 2014.

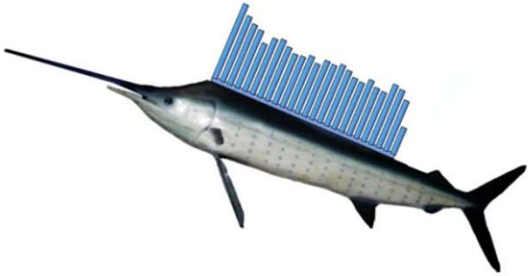
	union	intersection _strict	intersection _nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous (both genes with --nonunique all)	gene_A	gene_A
	ambiguous (both genes with --nonunique all)		
	alignment_not_unique (both genes with --nonunique all)		

featureCounts

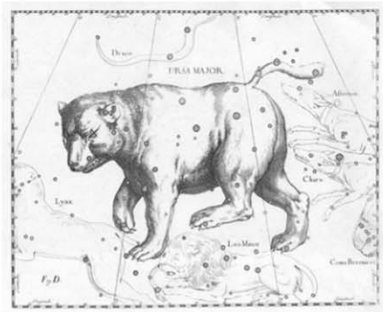
- A software program developed for counting reads to genomic features such as genes, exons, promoters, and genomic bins
- Quite similar to HTSeq-Count
- Takes as input SAM/BAM file and annotation file (GTF format or a SAF – Simplified Annotation Format)
- By default ignores reads mapping to more than one feature
- featureCounts is ~20 times faster than HTSeq-count
- With default settings is more liberal that HTSeq-count
- Feature Counts breaks the tie of ambiguous reads by assigning fragments to the feature that receives the highest number of reads from a pair (1 or 2) mapping to the feature
- Includes the GTF/GFF "end" location from feature intervals

Liao Y, Smyth GK and Shi W, Bioinformatics, 2014.

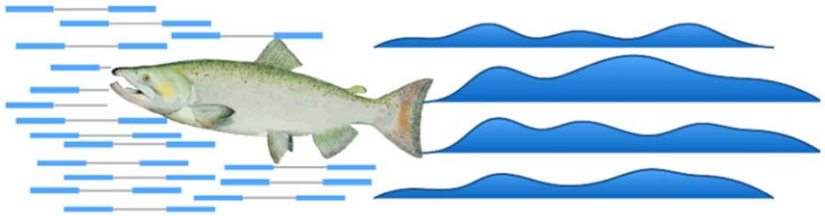
Alignment-free Quantification



Sailfish: 25x faster than anything that can before it. Accuracy is just as good.



Kalliso: 10x faster than Sailfish, more accurate.

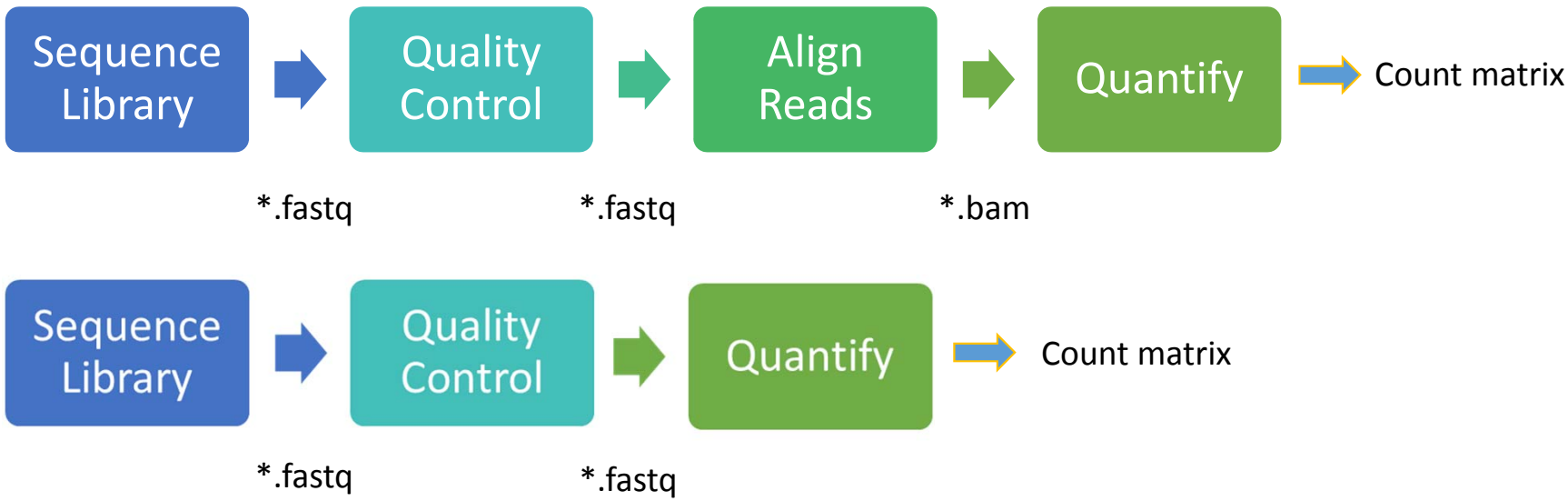


Salmon: Sailfish's successor. Borrows some techniques from Kallisto

Pseudoalignment

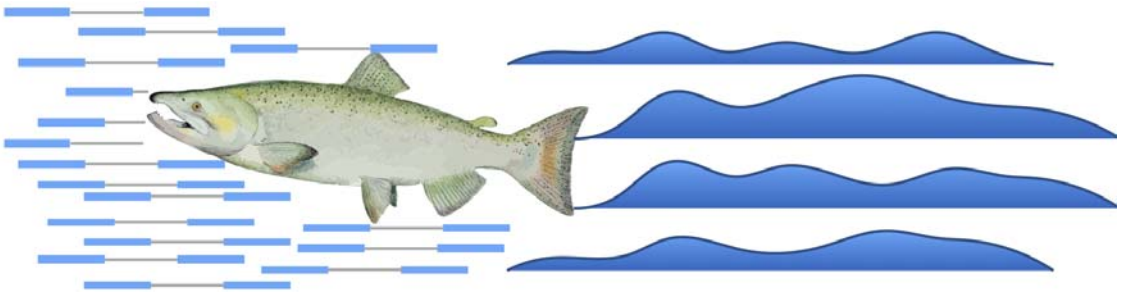
- Given a paired read, from which transcript could I have originated from?
- Not nucleotide sequence alignment
- It determines, for each read, not *where* in each transcript it aligns, but rather which transcripts it is compatible with.
- Very fast
  - The quantification of 78.6 million reads takes 14 minutes on a standard desktop using a single CPU core.
  - ~6 million reads quantified per minute

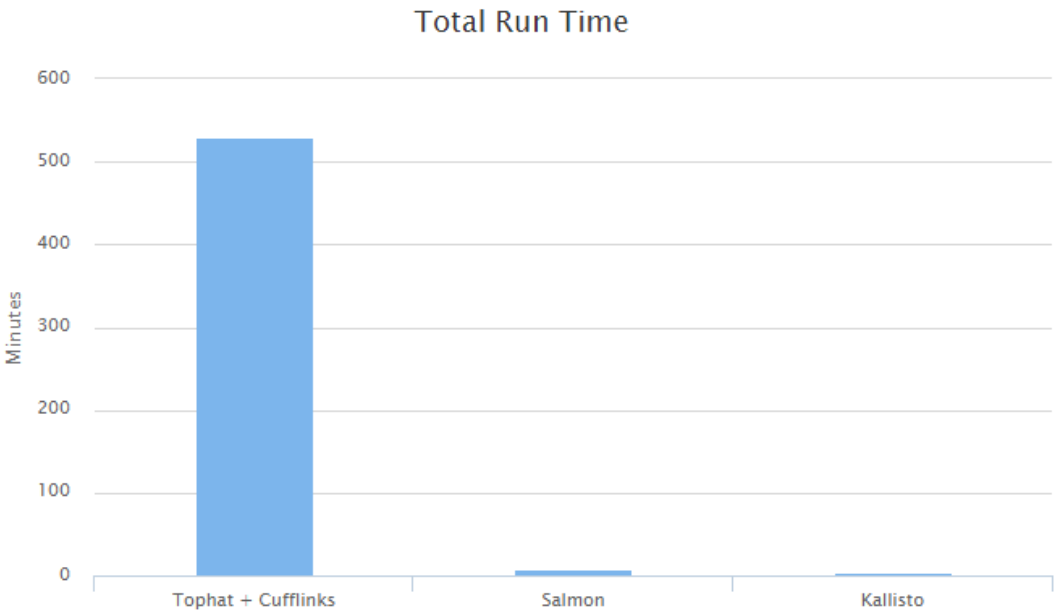
Why use a pseudoaligner?



Advantages:
-Pseudoalignment of reads preserves important information needed for quantification
-Fast
-Accurate

Salmon





- You used this in week 4 to generate a count matrix
- Input: fasta file with your reference sequence and fastq files of your reads
- Output: tab separated quantification file
- Two steps:
  1. Indexing
  2. Quantification

Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods*. 2017;14(4):417–419.