

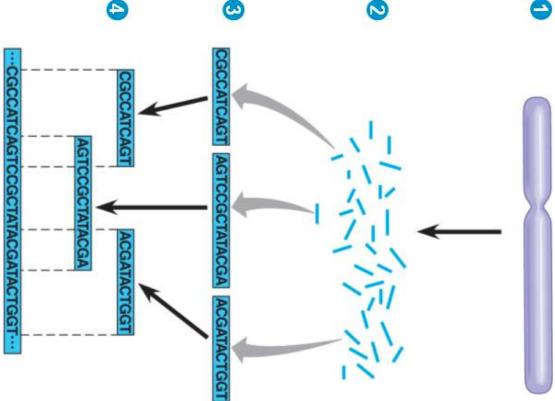
# Short Read Alignment

Mapping Reads to a Reference

Brandi Cantarel, Ph.D. & Daehwan Kim, Ph.D.  
BICF  
05/2019

- Introduction to Mapping
- Short Read Aligners
- DNA vs RNA
- Alignment Quality
- Pitfalls and Improvements

## Whole Genome Shotgun



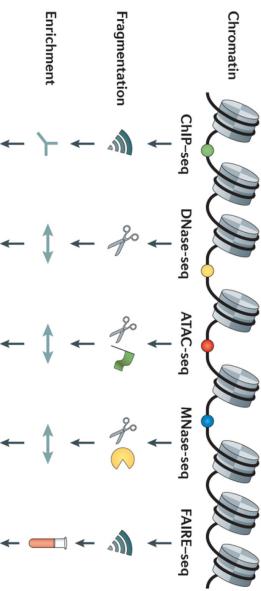
1979	Margaret Dayhoff compiled one of the first protein substitution matrices
1981	Temple Smith and Michael Waterman proposed an optimal alignment algorithm for local alignments
1985	William Pearson and David Lipman implement FASTA
1990	Temple Smith and Michael Waterman propose a faster alignment tool to identify high identity matches without GAPS
1992	Randall Smith and Temple Smith implement pattern-induced multiple-sequence alignment (PUMA)
1995	Sean Eddy implements multiple sequence alignments using hidden Markov Models (HMMER)
1996	Warren Gish, branches the development of BLAST with WU-BLAST
1997	Gapped BLAST and PSI-BLAST
2009	Bowtie, BWA, and TopHat Released
2012	STAR Released
2013	HISAT Released
2015	HISAT2 Released

## History of Sequence Similarity

# Genetic Variation

## Utility of Mapping

- DNASeq
  - Identify Variation
- RNASeq
  - Estimate the abundance of transcripts and genes
  - ChromatinSeq
    - Determine the structure of DNA (open, close, bound to proteins, etc)



## Chromatin Seq



	Wild-Type:	Substitution	Insertion	Deletion	Indel
Mutant:	AACGGCCCTGTAAC	AACGGCCCTGTAAC	AACGGCCCTGTAAC	AACGGCCCTGTAAC	AACGGCCCTGTAAC
	AACGGCCCTAGAAC	AACGGCCCTAGAAC	AACGGCCCTAGAAC	AACGGCCCTAGAAC	AACGGCCCTAGAAC
Wild-Type:	-	-	-	-	-
Mutant:	-	-	-	-	-
	-	-	-	-	-
	Wild-Type:	Duplication	Insertion	Translocation	
Mutant:	-	-	-	-	-
	-	-	-	-	-
	Wild-Type:	Duplication	Inversion	Translocation	
Mutant:	-	-	-	-	-
	-	-	-	-	-
	Wild-Type:	Individual 1: AACGGCCCTGTAAC	Individual 7: AACGGCCCTGTAAC	Individual 8: AACGGCCCTGTAAC	
	Individual 2: AACGGCCCTGTAAC	Individual 3: AACGGCCCTGTAAC	Individual 9: AACGGCCCTGTAAC	Individual 10: AACGGCCCTGTAAC	
	Individual 4: AACGGCCCTGTAAC	Individual 5: AACGGCCCTGTAAC	Individual 6: AACGGCCCTGTAAC	Individual 11: AACGGCCCTGTAAC	
	Individual 12: AACGGCCCTGTAAC	Individual 13: AACGGCCCTGTAAC	Individual 14: AACGGCCCTGTAAC	Individual 15: AACGGCCCTGTAAC	

Cantosso JG, Andersen MR, Hergard MJ, Sonnenburg N. Analysis of genetic variation and potential applications in genome-scale mutation modeling. Front Bioeng Biotechnol. 2015;13:1. doi: 10.3389/fbioe.2015.00013. eCollection 2015. Review. PMID: 25763369 | PubMed Central PMCID: PMC432917

## How to Make an Alignment

PMILGYWNVRGL  
PPYTIVYFPVRG

PMILGYWNVRGL  
PPYTIVYFPVRG

PM-ILGYWNVRGL  
PPYTIV-YFPVRG

PMILGYWNVRGL

### Local Alignment

AAPMILGYWNVRGLBB  
DDPPPYTIVYFPVRGCG

## Global Alignments

### Global Alignment

$A: \text{PMILGYWNVRGL}$   
 $B: \text{PYTIVYFPVRG-}$

```

Basis:
 $F_0 = d * j$ 
 $F_{i0} = d * i$ 
Recursion, based on the principle of optimality:
 $F_{ij} = \max(F_{i-1,j-1} + S(A_i, B_j), F_{i-1,j} + d, F_{i,j-1} + d)$ 

```

The pseudo-code for the algorithm to compute the  $F$  matrix therefore looks like this:

```

for i=0 to length(A)
    F(i,0) = d*i
for j=0 to length(B)
    F(0,j) = d*j
for i=1 to length(A)
    for j=1 to length(B)
        {
            Match = F(i-1,j-1) + S(Ai, Bj)
            Insert = F(i-1,j) + d
            Delete = F(i,j-1) + d
            Insert = max(Match, Insert, Delete)
        }
    }
}

```

## Local Alignments

### Local Alignment

$A: \text{AAPMILGYWNVRGLBB}$   
 $B: \text{DDPYTIVYFPVRGCC}$

A matrix  $H$  is built as follows:

$H(i,0) = 0, 0 \leq i \leq m$   
 $H(0,j) = 0, 0 \leq j \leq n$   
 if  $a_i = b_j$  then  $w(a_i, b_j) = w(\text{match})$  or if  $a_i^l = b_j$  then  $w(a_i, b_j) = w(\text{mismatch})$   
 $H(i,j) = \max \begin{cases} 0 & \\ H(i-1,j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1,j) + w(a_i, -) & \text{Deletion} \\ H(i,j-1) + w(-, b_j) & \text{Insertion} \end{cases}, 1 \leq i \leq m, 1 \leq j \leq n$

Where:

- $a, b = \text{Strings over the Alphabet } \Sigma$
- $m = \text{length}(a)$
- $n = \text{length}(b)$
- $w(c, d)$  is the maximum Similarity-Score between a suffix of  $a[1..j]$  and a suffix of  $b[1..l]$
- $w(c, d), c, d \in \Sigma \cup \{-\}$ ; '-' is the gap-scoring scheme

## Short Read Aligners

- Introduction to Mapping
  - Short Read Aligners
  - DNA vs RNA
  - Alignment Quality
  - Pitfalls and Improvements
- Short read aligners assume that the read came from “intact” from the reference
  - So the alignment is “global” from the read perspective and “local” from the reference perspective

# Short Read Aligners

# Short Read Aligners

Read    GATCGCAGAGCTCGGGCATAGCTAGCGC

Seed

AGCTATGATCGCAGAGCTCGGGCATAGCTAGCGCTAGAGGCTCGGATCGCAGAGTCGA  
Genome

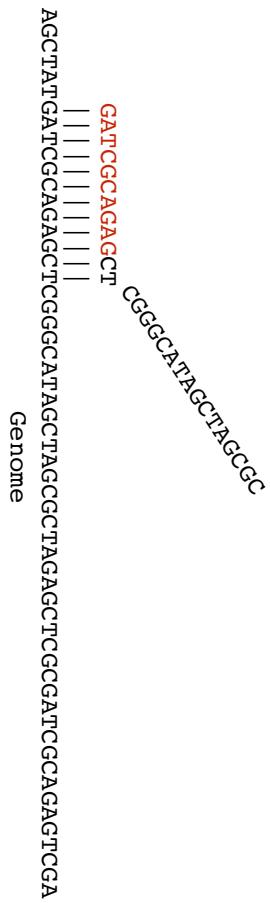
AGCTATGATCGCAGAGCTCGGGCATAGCTAGCGCTAGAGGCTCGGATCGCAGAGTCGA  
Genome

# Short Read Aligners

GATCGCAGAGCTCGGGCATAGCTAGCGC  
|||  
GATCGCAGAGCTCGGGCATAGCTAGCGC  
|||  
AGCTATGATCGCAGAGCTCGGGCATAGCTAGCGCTAGAGGCTCGGATCGCAGAGTCGA  
Genome

# Short Read Aligners

## SRA Features: Seeding

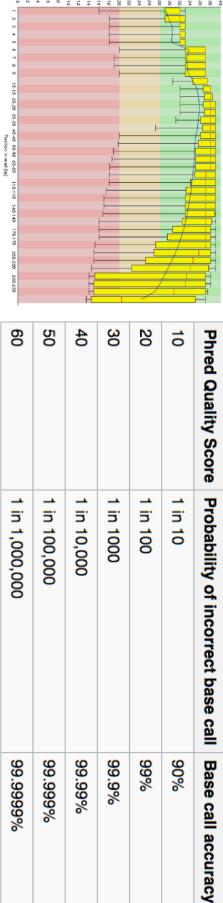


- Seeding represents the first few tens of base pairs of a read. The seed part of a read is expected to contain less erroneous characters due to the specifics of the NGS technologies. Therefore, the seeding property is mostly used to maximize performance and accuracy. The alignments are then extended from the seed.

## SRA Features: Base Quality

- Base quality scores provide a measure on correctness of each base in the read. The base quality score is assigned by a phred-like algorithm. The score Q is equal to  $-10 \log_{10}(e)$ , where e is the probability that the base is wrong. Some tools use the quality scores to decide mismatch locations. Others accept or reject the read based on the sum of the quality scores at mismatch positions.

Phred quality scores are logarithmically linked to error probabilities



## SRA Features: Gaps

- Existence of indels necessitates inserting or deleting nucleotides while mapping a sequence to a reference genome (gaps). The complexity of choosing a gap location increases with the read length. Therefore, some tools do not allow any gaps while others limit their locations and numbers.



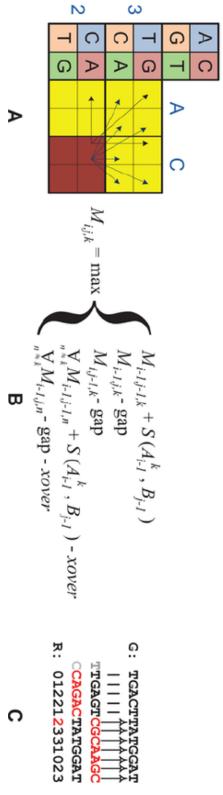
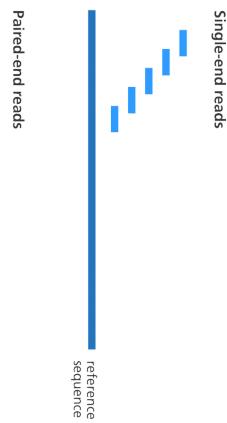
# SRA Features: Paired-End

- Paired-end reads result from sequencing both ends of a DNA molecule.

Mapping paired-end reads increases the confidence in the mapping locations due to having an estimation of the distance between the two ends.

Paired-end reads

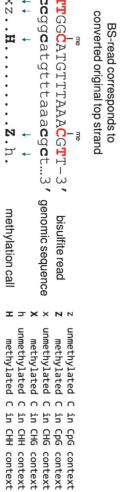
reference sequence



# SRA Features: Bisulphite

- Bisulphite treatment is a method used for the study of the methylation state of the DNA [3]. In bisulphite treated reads, each unmethylated cytosine is converted to uracil. Therefore, they require special handling in order not to misalign the reads.

**B**



# Short Read Aligners

- Color space read is a read type generated by SOLiD sequencers. In this technology, overlapping pairs of letters are read and given a number (color) out of four numbers [17]. The reads can be converted into bases, however, performing the mapping in the color space has advantages in terms of error detection.

**Table 1** Features supported by the tools

Bowtie	Bowtie2	BWA	SOAP2	MAQ	RIMAP	GSNAP	FANGS	Novaalign	mrFAST	mrFAST
Seed mm:	Up to 3	Up to 2	Any	Any	Count	Count	Count	QS	Count	Count
Non-seed mm:	QS	AS	Any	Count	Count	Count	Count	QS	Count	Count
Var. seed len:	> 5	> 5	Any	Count	> 28	Count	Count	QS	Count	Count
Mapping qual:	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gapped align.	Yes	Yes	PE	PE	Yes	Yes	Yes	Yes	Yes	Yes
Colorspace	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Splicing										
SNP tolerance										
Bisulfite reads										

PE-paired-end only, mm-mismatches, QS-base quality score, count-total count of mismatches in the read, AS-alignment score, and empty cells mean not supported.