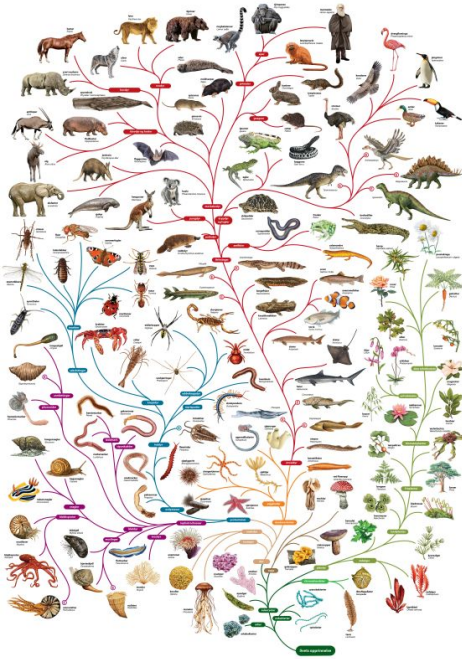


COMP90014

Algorithms for Bioinformatics

Week 2B - Sequence Alignment and Mapping II

Sequence Alignment and Mapping II



Local, Semi-Global alignment (previous lecture)

Scoring/Substitution Matrices

Gap penalties

Read mapping

Seed-extend (short reads)

Seed-chain-align (long reads)

BLAST

Academic Integrity Statement
DUE TOMORROW

Pairwise Alignment

Levenshtein distance

Forms the basis for the remainder of lecture

Global alignment

Local alignment

Semi-global alignment

These add a few important variations:

Penalties rather than adding edits

Penalties are different for mismatch vs gap

The arrows are stored

(directions we took to calculate each cell)

SPLATTERING -> PATTERN

Global	SPLATTERING -P-ATTE--
Local	ATTER ATTER
Semi-global	PLATTERIN P-ATTER-N

Global Alignment

Algorithm: Needleman Wunsch

GCACTGA-
GC-CTGAT

Algorithm 2: Needleman-Wunsch(A, B)

```

 $g \leftarrow \text{GapPenalty};$ 
for  $i = 0$  to  $\text{length}(A)$  do
   $S(i, 0) \leftarrow g \times i;$ 

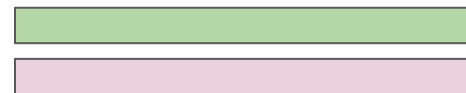
for  $j = 0$  to  $\text{length}(B)$  do
   $S(0, j) \leftarrow g \times j;$ 

for  $i = 1$  to  $\text{length}(A)$  do
  for  $j = 1$  to  $\text{length}(B)$  do
     $\text{Match} \leftarrow S(i-1, j-1) + \text{Scoring}(A_i, B_j);$ 
     $\text{Insert} \leftarrow S(i, j-1) + g;$ 
     $\text{Delete} \leftarrow S(i-1, j) + g;$ 
     $S(i, j) \leftarrow \max(\text{Match}, \text{Insert}, \text{Delete});$ 

return  $S(\text{length}(A), \text{length}(B))$ 

```

Seq A Seq B



Seq A

	start	G	C	A	C	T	G	A
start	0	-2	-4	-6	-8	-10	-12	-14
G	-2	1	-1	-3	-5	-7	-9	-11
C	-4	-1	2	0	-2	-4	-6	-8
C	-6	-3	0	1	1	-1	-3	-5
T	-8	-5	-2	-1	0	2	0	-2
G	-10	-7	-4	-3	-2	0	3	1
A	-12	-9	-6	-3	-4	-2	1	4
T	-14	-11	-8	-5	-4	-3	-1	2

Seq B

del ► (pointing to the cell at row C, column C)

ins ► (pointing to the cell at row T, column T)

Semi-global Alignment

Alignment of complete sequences, where offset is not penalised

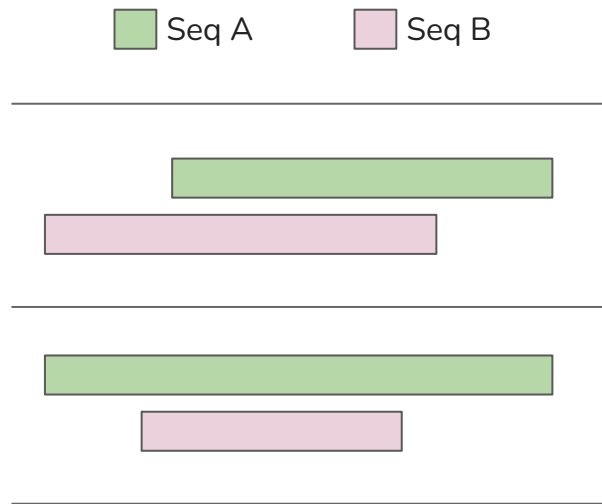
All of Seq A

All of Seq B

Best when sequences are expected to have similar overlapping region

eg. Read overlaps in OLC assembly

Returns the full alignment, [clipped by best offset](#)



Semi-global Alignment

Alignment of complete sequences, where offset is not penalised

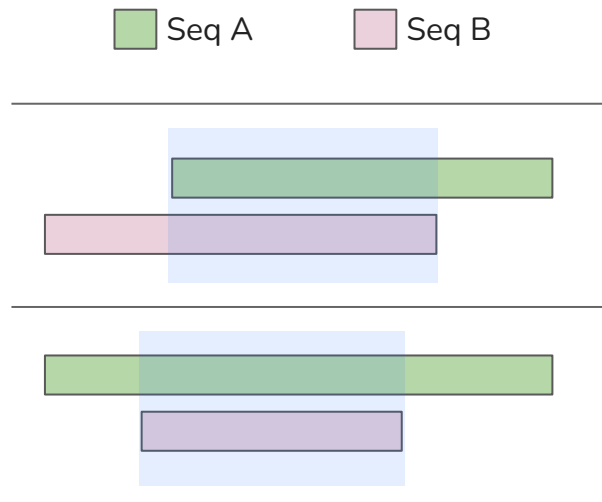
All of Seq A

All of Seq B

Best when sequences are expected to have similar overlapping region

eg. Read overlaps in OLC assembly

Returns the full alignment, [clipped by best offset](#)



Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start								
	G								
	C								
	C								
	T								
	G								
	A								
	T								

del ►

ins ►

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column: Why?

Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

Q

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0							
	C	0							
	del ▶								
	C	0							
	T	0							
	G	0							
	A	0							
ins ▶									
T	0								

del ►

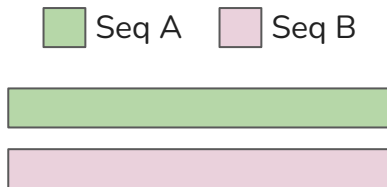
ins ►

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column: Why?



Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

Seq A

	start	G	C	A	C	T	G	A
start	0	0	0	0	0	0	0	0
G	0							
C	0							
C	0							
T	0							
G	0							
A	0							
T	0							

Seq B

del ►

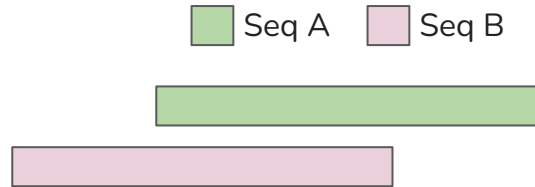
ins ►

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column: Why?



Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

Seq A

	start	G	C	A	C	T	G	A
start	0	0	0	0	0	0	0	0
G	0							
C	0							
C	0							
T	0							
G	0							
A	0							
T	0							

Seq B

del ► (pink arrow pointing to the 'C' row in Seq A)

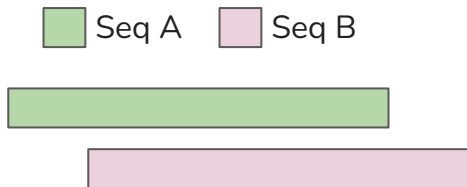
ins ► (pink arrow pointing to the 'T' row in Seq B)

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column: Why?



Scoring		(Gaps: -2)			
		C	T	A	G
C		+1	-1	-1	-1
T		-1	+1	-1	-1
A		-1	-1	+1	-1
G		-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0							
	C	0							
	C	0							
	T	0							
	G	0							
	A	0							
	T	0							

del ►

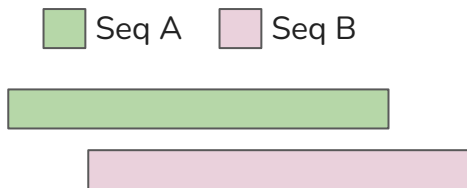
ins ►

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column: Why?



Do not want to penalise beginning shifts

Scoring		(Gaps: -2)			
		C	T	A	G
C		+1	-1	-1	-1
T		-1	+1	-1	-1
A		-1	-1	+1	-1
G		-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0							
	C	0							
	C	0							
	T	0							
	G	0							
	A	0							
	T	0							

del ▶

ins ▶

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column

Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0							
	C	0							
	C	0							
	T	0							
	G	0							
	A	0							
	T	0							

del ►

ins ►

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column

Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	1	-1	-1	-1	-1	1	-1
	del ► C	0	-1	2	0	0	-1	-1	0
	C	0	-1	0	1	1	-1	-2	-2
	T	0	-1	-2	-1	0	2	0	-2
	G	0	1	-1	-3	-1	0	3	1
	A	0	-1	0	0	-2	-2	1	4
	ins ► T	0	-1	-2	-1	-1	-1	-1	2

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column

-> the return score:

Max(bottom row), or Max(right column)

Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	1	-1	-1	-1	-1	1	-1
	C	0	-1	2	0	0	-1	-1	0
	C	0	-1	0	1	1	-1	-2	-2
	T	0	-1	-2	-1	0	2	0	-2
	G	0	1	-1	-3	-1	0	3	1
	A	0	-1	0	0	-2	-2	1	4
	ins ► T	0	-1	-2	-1	-1	-1	-1	2

del ►

ins ►

Semi-global Alignment

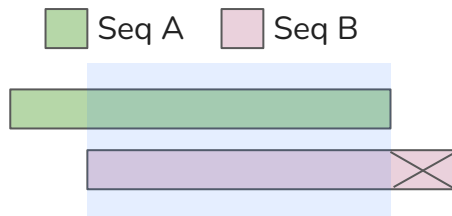
Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column

-> the return score:

Max(bottom row), or Max(right column)



Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	1	-1	-1	-1	-1	1	-1
	C	0	-1	2	0	0	-1	-1	0
	C	0	-1	0	1	1	-1	-2	-2
	T	0	-1	-2	-1	0	2	0	-2
	G	0	1	-1	-3	-1	0	3	1
	A	0	-1	0	0	-2	-2	1	4
	T	0	-1	-2	-1	-1	-1	-1	2

del ▶

ins ▶

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column

-> the return score:

Max(bottom row), or Max(right column)

-> the backtracking:

Starts at max cell

Ends when hit top row, or left column

GCACTGA

GC-CTGA

Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	1	-1	-1	-1	-1	1	-1
	C	0	-1	2	0	0	-1	-1	0
	C	0	-1	0	1	1	-1	-2	-2
	T	0	-1	-2	-1	0	2	0	-2
	G	0	1	-1	-3	-1	0	3	1
	A	0	-1	0	0	-2	-2	1	4
	T	0	-1	-2	-1	-1	-1	-1	2

del ►

ins ►

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column

-> the return score:

Max(bottom row), or Max(right column)

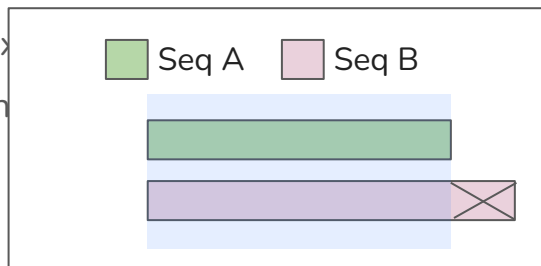
-> the backtracking:

Starts at max

Ends when h

GCACTGA

GC-CTGA



Example to the right

Scoring		(Gaps: -2)		
	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	1	-1	-1	-1	-1	1	-1
	C	0	-1	2	0	0	-1	-1	0
	C	0	-1	0	1	1	-1	-2	-2
	T	0	-1	-2	-1	0	2	0	-2
	G	0	1	-1	-3	-1	0	3	1
	A	0	-1	0	0	-2	-2	1	4
	T	0	-1	-2	-1	-1	-1	-1	2

del ►

ins ►

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column

-> the return score:

Max(bottom row), or Max(right column)

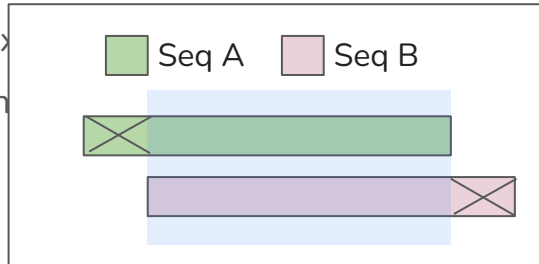
-> the backtracking:

Starts at max

Ends when h

GCACTGA

GC-CTGA



Hit top row

Scoring		(Gaps: -2)			
		C	T	A	G
C		+1	-1	-1	-1
T		-1	+1	-1	-1
A		-1	-1	+1	-1
G		-1	-1	-1	+1

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	1	-1	-1	-1	-1	1	-1
	C	0	-1	2	0	0	-1	-1	0
	C	0	-1	0	1	1	-1	-2	-2
	T	0	-1	-2	-1	0	2	0	-2
	G	0	1	-1	-3	-1	0	3	1
	A	0	-1	0	0	-2	-2	1	4
	T	0	-1	-2	-1	-1	-1	-1	2

del ►

ins ►

Semi-global Alignment

Algorithm: Needleman Wunsch (variant)

Same as Needleman-Wunsch except:

-> no offset penalties for top row, left column

-> the return score:

Max(bottom row), or Max(right column)

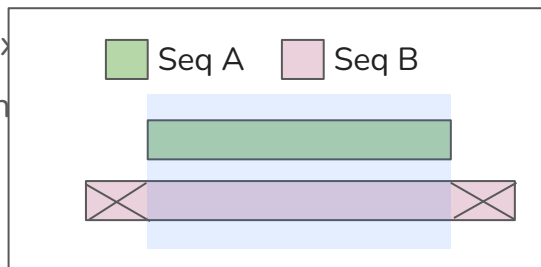
-> the backtracking:

Starts at max

Ends when h

GCACTGA

GC-CTGA



Hit first column

Scoring (Gaps: -2)

	C	T	A	G
C	+1	-1	-1	-1
T	-1	+1	-1	-1
A	-1	-1	+1	-1
G	-1	-1	-1	+1

Seq A

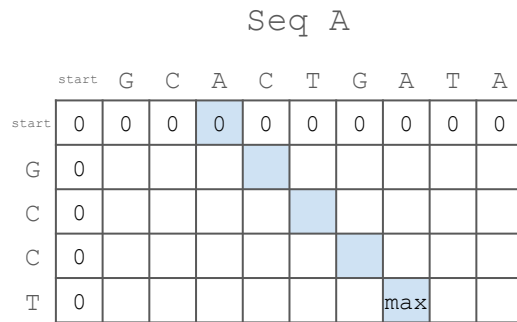
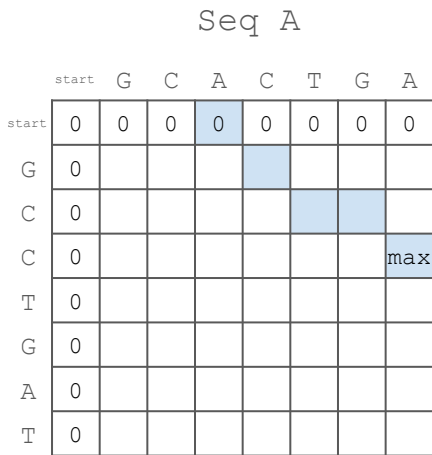
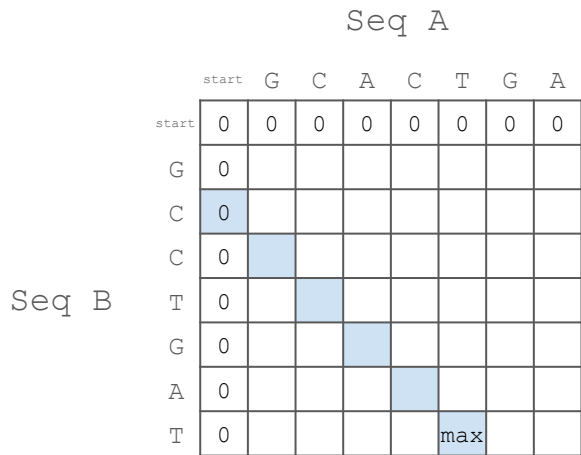
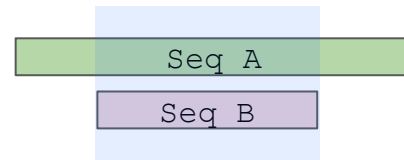
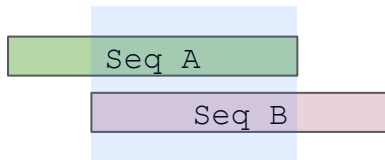
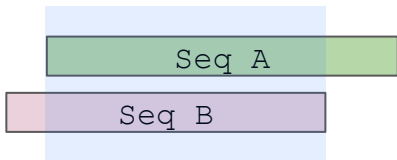
	start	G	C	A	C	T	G	A
start	0	0	0	0	0	0	0	0
G	0	1	-1	-1	-1	-1	1	-1
C	0	-1	2	0	0	-1	-1	0
C	0	-1	0	1	1	-1	-2	-2
T	0	-1	-2	-1	0	2	0	-2
G	0	1	-1	-3	-1	0	3	1
A	0	-1	0	0	-2	-2	1	4
T	0	-1	-2	-1	-1	-1	-1	2

del ►

Seq B

ins ►

Semi-global Alignment



Local Alignment

Region of best local similarity

Some of Seq A

Some of Seq B

Best when sequences are dissimilar, but contain regions of similarity

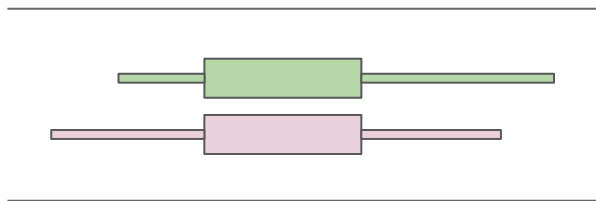
eg. BLAST: gene homology

Best region dictated by penalty scores

Returns the alignment in [highest scoring region](#)

Algorithm: Smith Waterman

Seq A Seq B



Gene homology between rat and human

Parts of the gene will be similar
(due to evolutionary viability)
(active sites, specific domains)

Parts of the gene will be dissimilar
(those which are more permissive to mutation)

Local Alignment

Region of best local similarity

Some of Seq A

Some of Seq B

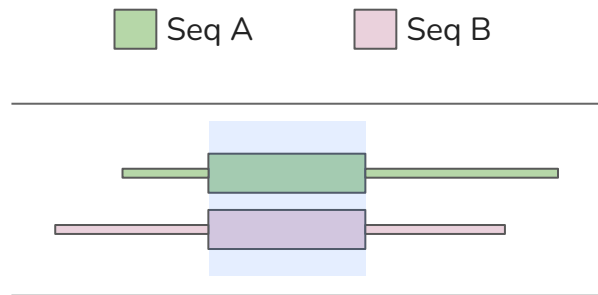
Best when sequences are dissimilar, but contain regions of similarity

eg. BLAST: gene homology

Best region dictated by penalty scores

Returns the alignment in [highest scoring region](#)

Algorithm: Smith Waterman



Gene homology between rat and human

Parts of the gene will be similar
(due to evolutionary viability)
(active sites, specific domains)

Parts of the gene will be dissimilar
(those which are more permissive to mutation)

Local Alignment

Algorithm: Smith Waterman

Same as Needleman-Wunsch except:

-> First row and first column set to 0

-> Negative score set to 0

-> the return score: $\max(S)$

-> the backtracking:

Starts at $\max(S)$

Ends when hit score of zero

Scoring (Gaps: -2)

	C	T	A	G
C	+2	-1	-1	-1
T	-1	+2	-1	-1
A	-1	-1	+2	-1
G	-1	-1	-1	+2

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	2	0	0	0	0	2	0
	C	0	0	4	2	2	0	0	0
	C	0	0	2	3	4	2	0	0
	T	0	0	0	1	2	6	4	2
	G	0	2	0	0	0	4	8	6
	A	0	0	1	0	0	2	6	10
	T	0	0	0	0	0	2	4	8

del ►

ins ►

Local Alignment

Algorithm: Smith Waterman

Same as Needleman-Wunsch except:

-> First row and first column set to 0

-> Negative score set to 0

-> the return score: $\max(S)$

-> the backtracking:

Starts at $\max(S)$

Ends when hit score of zero

Scoring (Gaps: -2)

	C	T	A	G
C	+2	-1	-1	-1
T	-1	+2	-1	-1
A	-1	-1	+2	-1
G	-1	-1	-1	+2

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	2	0	0	0	0	2	0
	C	0	0	4	2	2	0	0	0
	C	0	0	2	3	4	2	0	0
	T	0	0	0	1	2	6	4	2
	G	0	2	0	0	0	4	8	6
	A	0	0	1	0	0	2	6	10
	T	0	0	0	0	0	2	4	8

del ►

ins ►

Local Alignment

Algorithm: Smith Waterman

Same as Needleman-Wunsch except:

-> First row and first column set to 0

-> Negative score set to 0

-> the return score: $\max(S)$

-> the backtracking:

Starts at $\max(S)$

Ends when hit score of zero

Scoring (Gaps: -2)

	C	T	A	G
C	+2	-1	-1	-1
T	-1	+2	-1	-1
A	-1	-1	+2	-1
G	-1	-1	-1	+2

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	2	0	0	0	0	2	0
	C	0	0	4	2	2	0	0	0
	C	0	0	2	3	4	2	0	0
	T	0	0	0	1	2	6	4	2
	G	0	2	0	0	0	4	8	6
	A	0	0	1	0	0	2	6	10
	T	0	0	0	0	0	2	4	8

del ►

ins ►

Local Alignment

Algorithm: Smith Waterman

Same as Needleman-Wunsch except:

-> First row and first column set to 0

-> Negative score set to 0

-> the return score: $\max(S)$

-> the backtracking:

Starts at $\max(S)$

Ends when hit score of zero

Scoring (Gaps: -2)

	C	T	A	G
C	+2	-1	-1	-1
T	-1	+2	-1	-1
A	-1	-1	+2	-1
G	-1	-1	-1	+2

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	2	0	0	0	0	2	0
	C	0	0	4	2	2	0	0	0
	C	0	0	2	3	4	2	0	0
	T	0	0	0	1	2	6	4	2
	G	0	2	0	0	0	4	8	6
	A	0	0	1	0	0	2	6	10
	T	0	0	0	0	0	2	4	8

del ►

ins ►

Local Alignment

Algorithm: Smith Waterman

Same as Needleman-Wunsch except:

-> First row and first column set to 0

-> Negative score set to 0

-> the return score: $\max(S)$

-> the backtracking:

Starts at $\max(S)$

Ends when hit score of zero

GCACTGA

GC-CTGA

Scoring (Gaps: -2)

	C	T	A	G
C	+2	-1	-1	-1
T	-1	+2	-1	-1
A	-1	-1	+2	-1
G	-1	-1	-1	+2

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	2	0	0	0	0	2	0
	C	0	0	4	2	2	0	0	0
	del ▶	C	0	0	2	3	4	2	0
	T	0	0	0	1	2	6	4	2
	G	0	2	0	0	0	4	8	6
	A	0	0	1	0	0	2	6	10
	ins ▶	T	0	0	0	0	0	2	4

del ►

ins ►

Local Alignment

Algorithm: Smith Waterman

Same as Needleman-Wunsch except:

-> First row and first column set to 0

-> Negative score set to 0

-> the return score: $\max(S)$

-> the backtracking:

Starts at $\max(S)$

Ends when hit score of zero

ACTGA

CCTGA

Scoring (Gaps: -2)

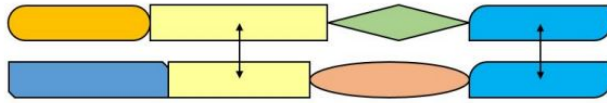
	C	T	A	G
C	+2	-1	-1	-1
T	-1	+2	-1	-1
A	-1	-1	+2	-1
G	-1	-1	-1	+2

		Seq A							
		start	G	C	A	C	T	G	A
Seq B	start	0	0	0	0	0	0	0	0
	G	0	2	0	0	0	0	2	0
	C	0	0	4	0	2	0	0	0
	C	0	0	2	3	4	2	0	0
	T	0	0	0	1	2	6	4	2
	G	0	2	0	0	0	4	8	6
	A	0	0	1	0	0	2	6	10
	T	0	0	0	0	0	2	4	8

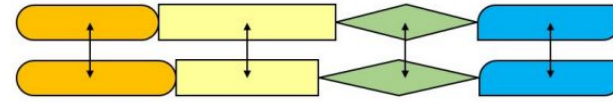
del ►

ins ►

Local vs. Global



Local Alignment



Global Alignment

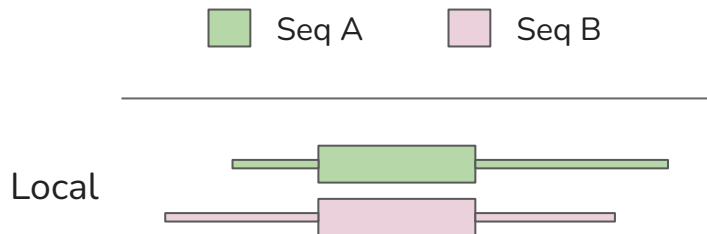
	Smith-Waterman algorithm	Needleman-Wunsch algorithm
Initialization	First row and first column are set to 0	First row and first column are subject to gap penalty
Scoring	Negative score is set to 0	Score can be negative
Traceback	Begin with the highest score, end when 0 is encountered	Begin with the cell at the lower right of the matrix, end at top left cell
Complexity	$O(m \times n)$	$O(m \times n)$

Pairwise Alignment

Local Alignment

Finds region of best local similarity

$O(m \times n)$

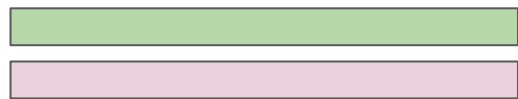


Global Alignment

End-to-end alignment of two sequences

$O(m \times n)$

Global

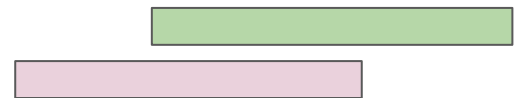


Semi-Global Alignment

Alignment of complete sequences, where offset is not penalised

$O(m \times n)$

Semi-global



Does this scale to large datasets?

Alignment: Scoring / Substitution Matrices

Scoring Alignments

- The optimal alignment depends on our scoring system

- Matches (reward)
- Mismatches (penalty)
- Starting/Extending Gaps (penalty)

- Simple match/mismatch score

- Matches: +1
- Mismatches/Gaps: -1

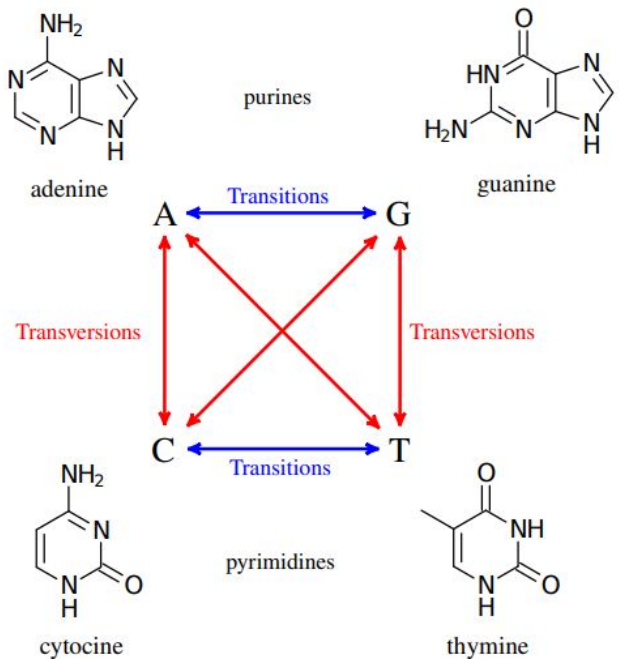
- Generalize to a substitution matrix

- Assign a score to each pair of characters
- $N \times N$ Symmetric matrix
 - $N=4$ Nucleic acids
 - $N=20$ Proteins
- $M(i, j)$ cost/reward to change from i to j

	C	T	A	G
C	1	-1	-1	-1
T	-1	1	-1	-1
A	-1	-1	1	-1
G	-1	-1	-1	1

Why can't we just use a constant score for mismatches?

这是因为DNA序列中的不同类型的错配——转换 (transitions) 和颠换 (transversions) ——具有不同的生物学意义和发生频率。



- two types of DNA substitution mutations:
 - transitions (bases with similar shape)
 - transversions (different number of rings)
- transition mutations occur more frequently
- transitions are less likely to result in amino acid substitutions (often synonyms)
- we can capture this in a substitution matrix

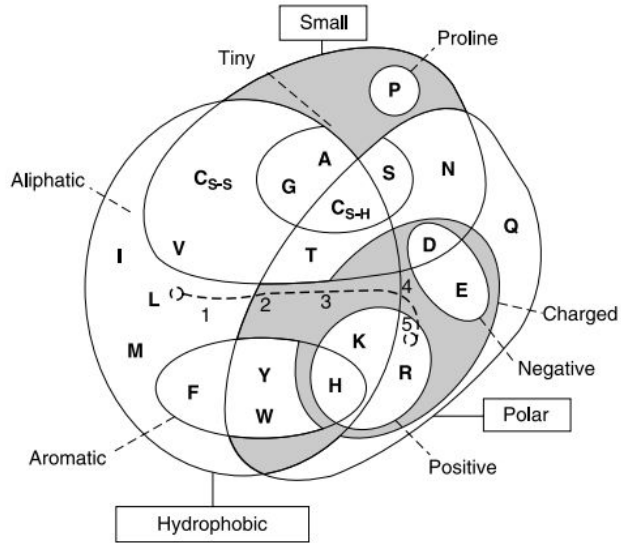
	C	T	A	G
C	2	1	-1	-1
T	1	2	-1	-1
A	-1	-1	2	1
G	-1	-1	1	2

转换是指同类核苷酸（嘌呤或嘧啶）之间的替换，例如腺嘌呤（A）和鸟嘌呤（G）之间，或者胞嘧啶（C）和胸腺嘧啶（T）之间的替换。
颠换是指不同类核苷酸之间的替换，例如嘌呤和嘧啶之间的任意组合。

Krishnavedala via [Wikimedia Commons](#)

同一核苷酸的匹配得分为2。
转换（A↔G或C↔T）的得分1
颠换的得分为-1，表明其不受青睐。

Protein substitution matrix



- protein substitution matrices are more complex than DNA scoring matrices
- proteins are composed of 20 different amino acids
- varied physicochemical properties
 - compare a D to E with a D to W
- scoring matrices reflect:
 - chemical similarity
 - observed mutation frequencies
 - how protein sequences evolve

蛋白质替换矩阵比DNA评分矩阵更复杂，因为蛋白质由20种不同的氨基酸组成，每种氨基酸具有独特的物理和化学性质。氨基酸的多样性物理化学性质让替换矩阵更为复杂。例如，比较酸性氨基酸天冬氨酸（D）和谷氨酸（E）的替换可能比将天冬氨酸与芳香氨基酸色氨酸（W）替换的评分更为合理，因为前者在化学性质上更为相似。蛋白质替换矩阵反映了化学相似性、观测到的突变频率以及蛋白质序列如何进化。这意味着替换矩阵中的分数是基于特定氨基酸对在自然进化过程中发生突变的频率以及这些氨基酸在结构和功能上的相似性。

Protein substitution matrices

*How often is one amino acid substituted
for another in related proteins?*

PAM

Point Accepted Mutation

BLOSUM

Blocks Substitution Matrix

图片列出了两种不同类型的替换矩阵：

PAM (Point Accepted Mutation)：这是一种基于一定数量的氨基酸改变所发生的进化距离来估算的矩阵。PAM模型是基于相对较小的进化变化，适用于比较近缘序列。

BLOSUM (Blocks Substitution Matrix)：这种矩阵是根据实际序列的区块 (blocks) 对序列进行对比，来推断氨基酸替换的频率。BLOSUM矩阵用于更远亲缘关系的序列比对，因为它是基于实际的氨基酸替换频率，而不是预估的进化距离。

这两种矩阵都是用来评估在蛋白质比对时，一个特定的氨基酸替换发生的可能性，从而帮助科学家理解蛋白质如何进化，以及哪些替换是生物学上可接受的。

PAM1 matrix

- 🌐 Margaret Dayhoff in *Atlas of protein sequence and structure* (1978)
- 🌐 the first widely-used amino acid substitution matrix

1	2	3	4	5	6	7	8	9
A	C	G	C	T	A	F	K	I
G	C	G	C	T	A	F	K	I
A	C	G	C	T	A	F	K	L
G	C	G	C	T	G	F	K	I
G	C	G	C	T	L	F	K	I
A	S	G	C	T	A	F	K	L
A	C	A	C	T	A	F	K	L

- 🌐 derived from global alignments of closely related sequences

- 71 groups of protein sequences
- minimum 85% identity
- functional proteins
- 1572 amino-acid changes/mutations

- 🌐 evolutionary model

- assumes symmetry: $A \rightarrow B = B \rightarrow A$
- assumes substitutions observed over short periods of time can be extrapolated to long periods of time (mathematically)

Extrapolating PAM matrices to longer distances

PAM (Point Accepted Mutation) 矩阵从短进化距离外推至长进化距离的方法

- Margaret Dayhoff in *Atlas of protein sequence and structure* (1978)
- Family of matrices:
PAM1, PAM80, PAM120, PAM250
- *evolutionary interval is the time taken for n mutations to occur per 100 amino acids*
- the number represents the evolutionary distance between the sequences
- higher numbers denote greater distances
- PAM matrices for larger evolutionary distances are extrapolated from PAM1
- probabilities are calculated by matrix multiplication, e.g.
$$M_2 = M_1 \times M_1$$
$$M_{250} = M_1^{250}$$
- PAM250 is the substitution matrix calculated from M_{250}

Limitations of PAM matrices

[illegible]

- inferred from a small dataset with $\geq 85\%$ identity
- mainly small globular proteins
- doesn't account for different evolutionary rates between conserved and non-conserved regions

BLOSUM (BLOcks SUBstitution Matrix)

	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
A	0	1	0	4																	A
G	-3	0	-2	0	6																G
P	-3	-1	-1	-1	-2	7															P
D	-3	0	-1	-2	-1	-1	6														D
E	-4	0	-1	-1	-2	-1	2	5													E
Q	-3	0	-1	-1	-2	-1	0	2	5												Q
N	-3	1	0	-2	0	-2	1	0	0	6											N
H	-3	-1	-2	-2	-2	-2	-1	0	0	1	8										H
R	-3	-1	-1	-1	-2	-2	-2	0	1	0	0	5									R
K	-3	0	-1	-1	-2	-1	-1	1	1	0	-1	2	5								K
M	-1	-1	-1	-1	-3	-2	-3	-2	0	-2	-2	-1	-1	5							M
I	-1	-2	-1	-1	-4	-3	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-1	-4	-3	-4	-3	-2	-3	-3	-2	-2	2	2	4					L
V	-1	-2	0	0	-3	-2	-3	-2	-2	-3	-3	-3	-2	1	3	1	4				V
W	-2	-3	-2	-3	-2	-4	-4	-3	-2	-4	-2	-3	-3	-1	-3	-2	-3	11			W
Y	-2	-2	-2	-2	-3	-3	-3	-2	-1	-2	2	-2	-2	-1	-1	-1	-1	2	7		Y
F	-2	-2	-2	-2	-3	-4	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	1	3	6	F
	C	S	T	A	G	P	D	E	Q	N	H	R	K	M	I	L	V	W	Y	F	

- Henikoff & Henikoff, *Amino acid substitution matrices from protein blocks* (1992).
[10.1073/pnas.89.22.10915](https://doi.org/10.1073/pnas.89.22.10915)
- alignments of 500 distantly related protein families
- scores derived from frequencies of substitutions in blocks of ungapped local alignments
- BLOSUMx is based on sequences that share at least x% identity
 - e.g. BLOSUM62 was constructed from aligned sequences sharing no more than 62 % identity

这是一个用于评分生物序列对比中氨基酸替换的矩阵，特别用于比较蛋白质序列。矩阵中的每个分数表示了两种特定氨基酸发生替换的相对可能性。正分数代表比随机替换更有可能发生的替换，而负分数代表不太可能发生的替换。

BLOSUM矩阵是从大量的蛋白质家族中的无间隙局部对齐序列块中得出的。这意味着它们分析了在不考虑序列插入或缺失的情况下，在蛋白质结构的相应位置上发生的替换。

PAM vs. BLOSUM

BLOSUM80

PAM1

BLOSUM62

PAM120

BLOSUM45

PAM250

Less divergent



More divergent

- For closely related proteins: lower PAM matrices or higher BLOSUMs
- For distantly related proteins higher PAM matrices or lower BLOSUMs
- BLOSUM62 is commonly used for database searching (BLAST default)

- In general:**
 - BLOSUMs perform well for local similarity searches
 - PAM matrices perform well for global alignments
- BLOSUMs are calculated from observed frequencies
- higher PAM matrices are extrapolated mathematically

具体应用：

对于密切相关的蛋白质，推荐使用较低的PAM矩阵或较高的BLOSUM矩阵。
对于关系较远的蛋白质，推荐使用较高的PAM矩阵或较低的BLOSUM矩阵。

计算方法：

BLOSUM矩阵是基于实际观察到的替换频率来计算的。

PAM矩阵是基于较短的进化时间内的替换事件，并通过数学方法外推到较长的时间。

BLOSUM80和PAM1用于比较较少分歧（即更密切相关）的蛋白质。

BLOSUM62是介于两者之间，适用于一般的数据库搜索，并且是BLAST（Basic Local Alignment Search Tool）的默认矩阵。

BLOSUM45和PAM250用于比较更多分歧（即距离更远的关系）的蛋白质。

Alignment: Gap Penalties

Why penalize gaps?

为什么在蛋白质或核酸序列比对时需要对空位 (gaps) 进行惩罚

空位不惩罚：如果比对过程中引入空位不进行任何惩罚，将导致误导性的比对结果。这是因为比对算法可能会产生许多空位，来轻易增加匹配的数量，但这并不代表两个序列在功能或进化上的真实关系。

- allowing gaps with no cost results in misleading alignments

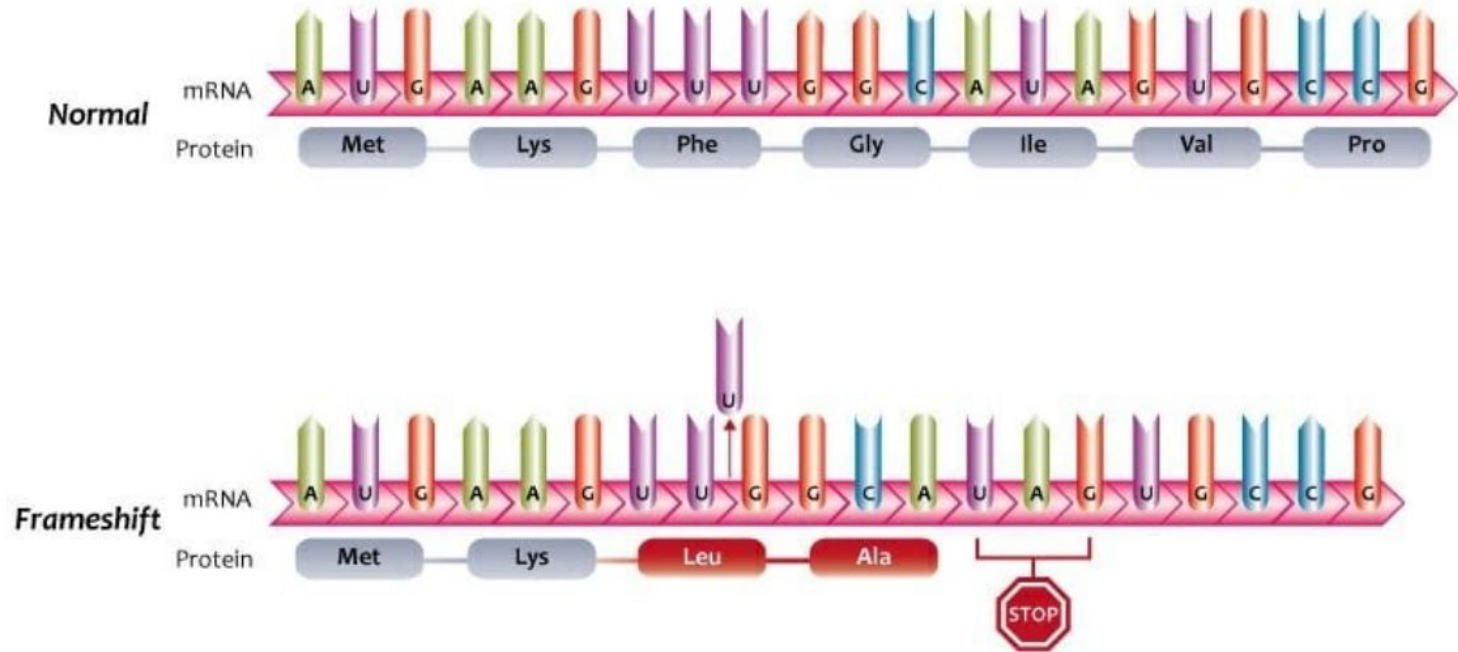
V	D	S	-	C	Y
V	E	S	L	C	Y

optimal alignment: **maximizes** the number of matches, and **minimizes** the number of gaps

- tradeoff: adding gaps reduces mismatches
- penalising gaps heavily forces alignments to have fewer gaps

增加空位减少错配：引入空位可以减少错配数量，错配是指不同序列中相对应位置上不同的氨基酸（或核苷酸）。在某些情况下，这可能反映了序列在进化过程中的插入或缺失事件。
重度惩罚空位：如果对空位给予重度惩罚，比对算法会倾向于减少空位数量，这可能导致更少的错配。但是，过分的惩罚又可能忽略了真实的生物学事件，如插入和缺失。

Why penalize gaps?



Adapted from Campbell NA (ed). Biology, 2nd ed, 1990.

How do we penalise gaps?

Negative score:

- 🌐 Gap penalty should be several times greater than the mismatch penalty
 - Proteins: an insertion/deletion could interrupt the entire polymer chain
 - DNA: shift the reading frame

A naïve approach:

- 🌐 fixed penalty (σ) for every gap/indel
 - $-\sigma$ for one indel
 - -2σ for two consecutive indels
 - -3σ for three consecutive indels
- 🌐 what is the problem with that?

V	D	S	-	C	Y
V	E	S	L	C	Y

Fixed gap penalty

Alignment 1:

```
ATGTAGTGTATAGTACATGCA
ATGTAG-----TACATGCA
```

- an indel of length k is more likely to occur as a single event than as k events each of length 1
- i.e. alignment 1 is a better representation of homology

Alignment 2:

```
ATGTAGTGTATAGTACATGCA
ATGTA--G--TA---CATGCA
```

- a fixed gap penalty would give the same score for both
- treat gap initiation and gap extension differently

```
V D S - C Y
V E S L C Y
```

Affine gap penalties

affine: **linear** (in this context).

i.e. the penalty grows at the same rate as the length of the gap.

- ① large penalty for opening a gap
- ① much smaller penalty for gap extension
- ① an indel of length k has a penalty $W(k)$
 - $W(k) = -(\rho + \sigma \times k)$
 - ρ : penalty to open a gap
 - σ : penalty to extend a gap
- ① e.g. used by BLAST

affine: 在这里指的是线性的，即空位 (gap) 的惩罚随着空位长度的增加而线性增长。

large penalty for opening a gap: 这意味着开始一个空位时会受到较大的惩罚。

much smaller penalty for gap extension: 一旦空位开始，其延伸所受的惩罚要比开始时小得多。

an indel of length k has a penalty $W(k)$: 这里介绍了惩罚函数 $W(k)$ ，用于计算长度为 k 的插入 (insertion) 或缺失 (deletion) 的惩罚。该函数定义为：

(rho): 是开始一个空位的惩罚。

(sigma): 是延伸一个空位的惩罚。

例如，这种惩罚模型被 BLAST 使用。BLAST (Basic Local Alignment Search Tool) 是一个广泛使用的工具，用于进行生物序列比对。

这种仿射空位惩罚模型是基于生物学上的考虑，因为一次性的较大缺失或插入事件比多次的小缺失或插入更可能发生。因此，该模型通过对第一个空位赋予较大的惩罚而对随后的空位延伸赋予较小的惩罚，来更真实地反映这种情况。这样的空位惩罚方式有助于避免序列比对时因为过多的小空位而引起的分数惩罚累积，导致对比对质量的错误评估。

Read Mapping: Seed-Extend

Recap

Kmers

Subsequences of length K

Used extensively in bioinformatics

Indexing

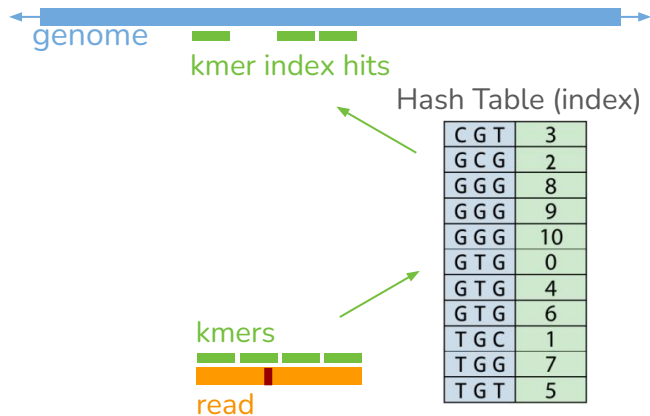
Method to quickly identify matching regions of two sequences

Key: Value store
Kmers: Occurrences

Alignment

Finding optimal match between two sequences

Considers matches, mismatches, gaps



	start	G	C	A	C	T	G	A
start	0	-2	-4	-6	-8	-10	-12	-14
G	-2	1	-1	-3	-5	-7	-9	-11
C	-4	-1	2	0	-2	-4	-6	-8
C	-6	-3	0	1	1	-1	-3	-5
T	-8	-5	-2	-1	0	2	0	-2
G	-10	-7	-4	-3	-2	0	3	1
A	-12	-9	-6	-3	-4	-2	1	4
T	-14	-11	-8	-5	-4	-3	-1	2

Alignment

GCACTGA-
||.||||.
GC-CTGAT

即如何将测序得到的短DNA片段（称为reads）映射到人类参考基因组上。

Read Mapping: Seed-Extend

半全局对齐指的是一种序列对齐方式，其中一个序列（read）必须完全对齐，而另一个序列（参考基因组）可以在两端有不对齐的部分。

Mapping reads to human reference genome

Ideally, we could use [semi-global](#) alignment

Is this feasible for average dataset?

- > each read is ~ 100 bp
- > reference length ~ 3.2 billion bp
- > number of reads ~ 300 million (30x coverage)

Semi-global alignment time and space complexity

- > quadratic: $O(n \times m)$
- > single read: 300 billion operations
- > average dataset: 10^{20}

[Not feasible!](#)

We need a heuristic approach

先找到reads与参考基因组中一个小区域的粗略匹配（种子），然后扩展这个匹配到整个read。



Needle in haystack via wikipedia commons
(CC BY-SA 4.0)

Read Mapping: Seed-Extend

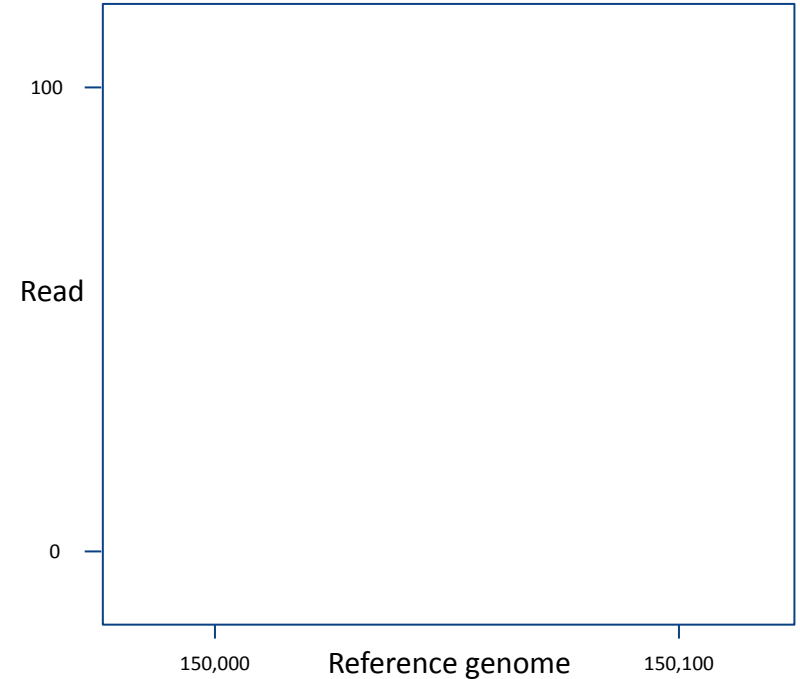
Seed-Extend

Combines [indexing](#) and [alignment](#)

Used for aligning short, accurate sequences

Process

Seed-extend strategy



Read Mapping: Seed-Extend

Seed-Extend

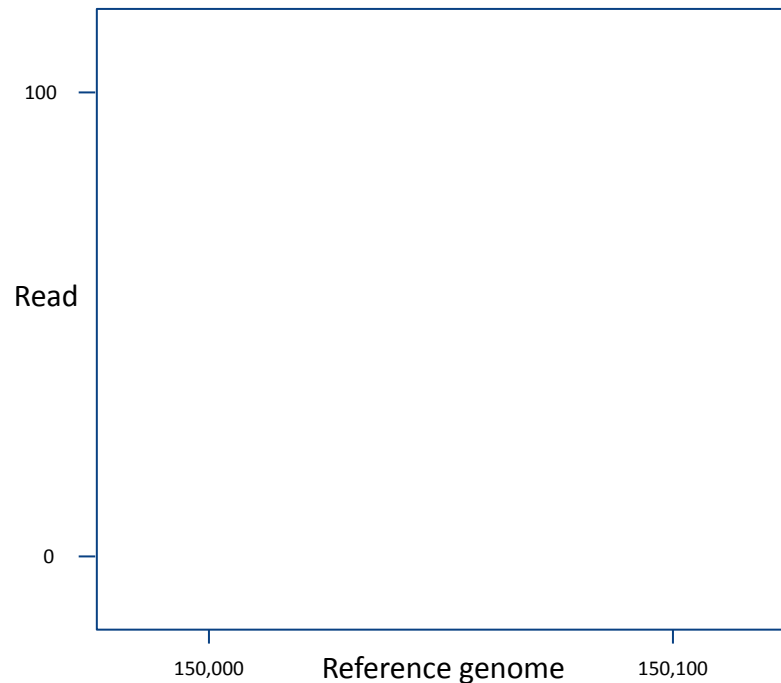
Combines [indexing](#) and [alignment](#)

Used for aligning short, accurate sequences

Process

Index the reference genome using kmers

Seed-extend strategy



Read Mapping: Seed-Extend

Seed-Extend

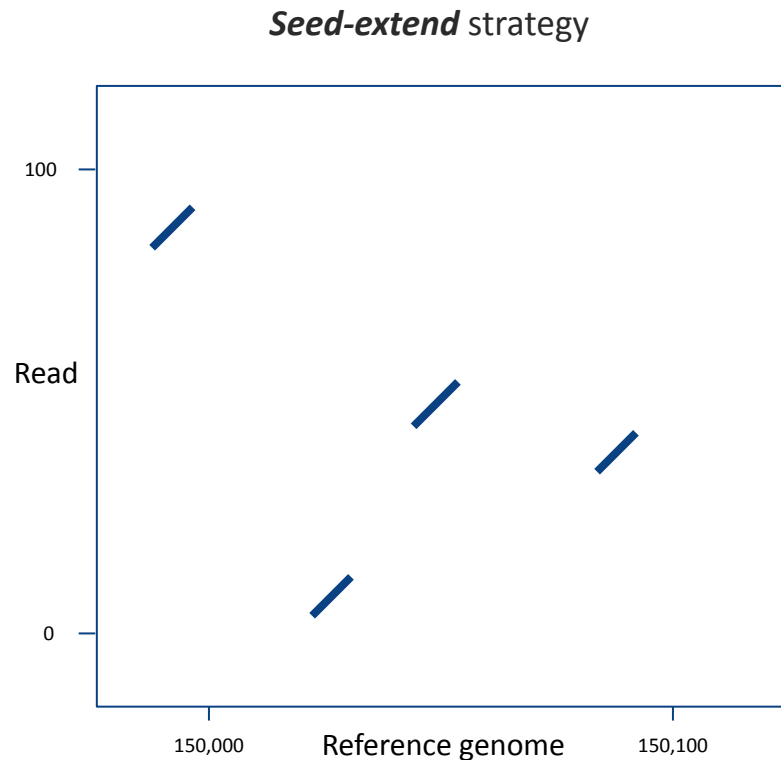
Combines [indexing](#) and [alignment](#)

Used for aligning short, accurate sequences

Process

Index the reference genome using kmers

Use the index to find “seed” matches for each read
(kmer hits between read and index)



Read Mapping: Seed-Extend

Seed-Extend

Combines **indexing** and **alignment**

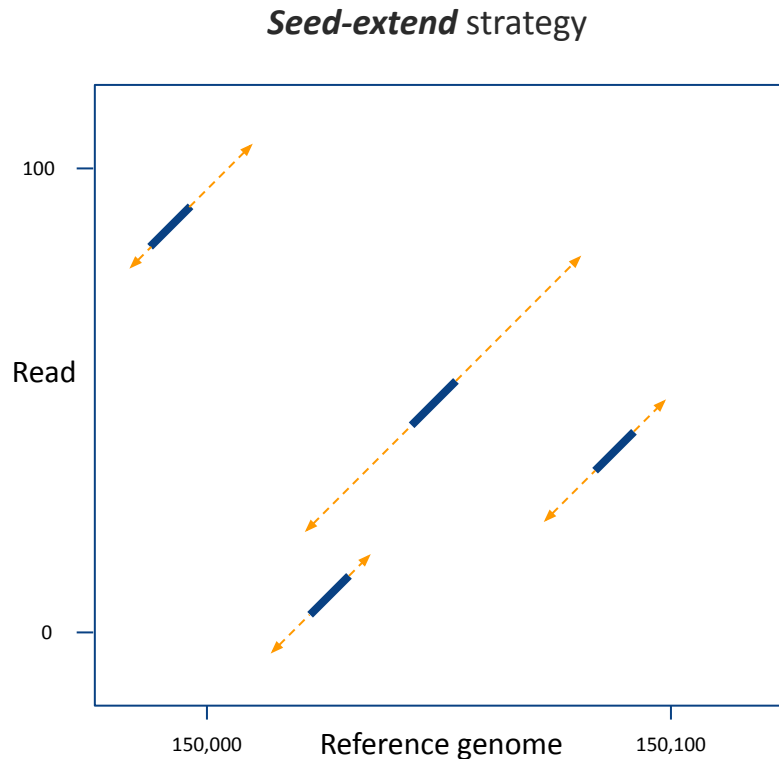
Used for aligning short, accurate sequences

Process

Index the reference genome using kmers

Use the index to find “seed” matches for each read
(kmer hits between read and index)

Extend the match ends using alignment
(Local alignment and / or gap-free alignment)



Read Mapping: Seed-Extend

Seed-Extend

Combines **indexing** and **alignment**

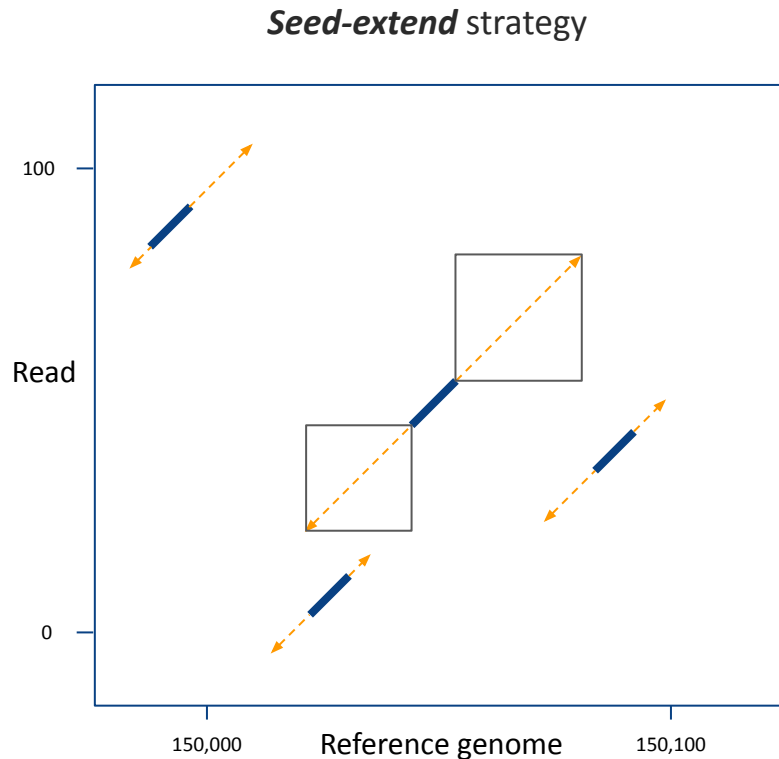
Used for aligning short, accurate sequences

Process

Index the reference genome using kmers

Use the index to find “seed” matches for each read
(kmer hits between read and index)

Extend the match ends using alignment
(Local alignment and / or gap-free alignment)



右侧有一个图表显示了Seed-extend策略的视觉表示。
Read: 表示一个单独的DNA read。 Reference genome: 表示参考基因组的一小段, 数值范围从150,000到150,100。 5' alignment和3' alignment: 用不同颜色的箭头标出了read的5'端和3'端如何与参考基因组对齐的位置。 Kmer match: 粉红色的虚线表示在read和参考基因组之间找到的kmer匹配。

Read Mapping: Seed-Extend

Seed-Extend

Combines **indexing** and **alignment**

Used for aligning short, accurate sequences

Process

Index the reference genome using kmers

Use the index to find “seed” matches for each read
(kmer hits between read and index)

Extend the match ends using alignment
(Local alignment and / or gap-free alignment)

Return the **best location**
(5' alignment + kmer match + 3' alignment)

整个幻灯片传达的信息是, Seed-Extend策略通过索引参考基因组和使用有效的对齐技术来克服直接对齐所有可能的位置所需的计算量, 从而使read映射过程更加可行。通过这种方式, 可以快速地定位read在参考基因组中的最佳匹配位置。



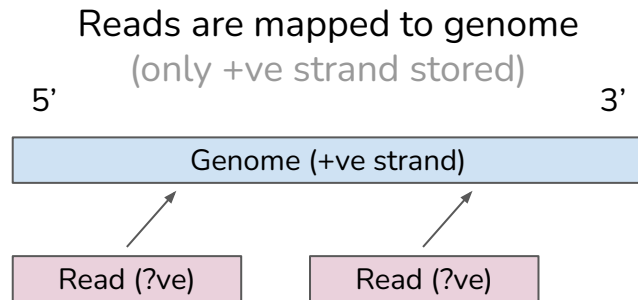
Read Mapping: Seed-Extend

What about reverse strand?

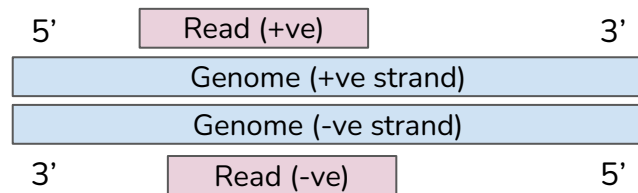
Genome: Double Stranded DNA (except some viruses)

DNA Seq. Reads: Can originate from +ve or -ve strand

Reference genome: Only the +ve strand



But reads originate from both
strands of genome!



Read Mapping: Seed-Extend

What about reverse strand?

Genome: Double Stranded DNA (except some viruses)

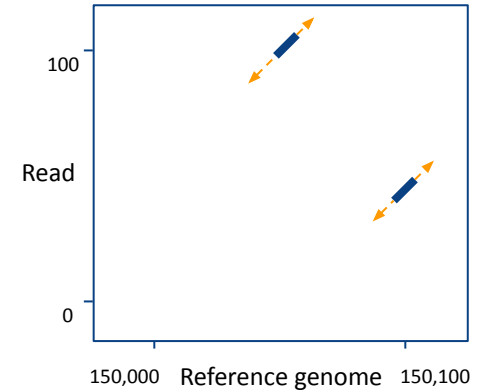
DNA Seq. Reads: Can originate from +ve or -ve strand

Reference genome: Only the +ve strand

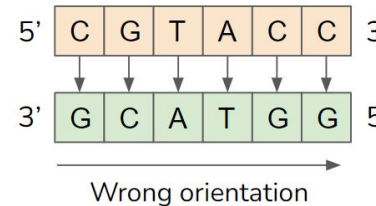
For each read (sequence to align)

1. Do Seed-Extend (original - forward orientation).
2. Flip it (reverse complement).

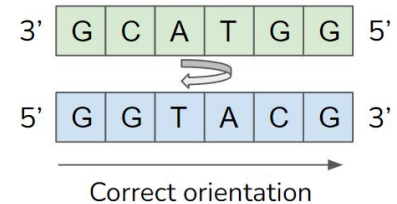
Read in Fwd (original) Orientation



Calculate Complement



Reverse



Read Mapping: Seed-Extend

What about reverse strand?

Genome: Double Stranded DNA (except some viruses)

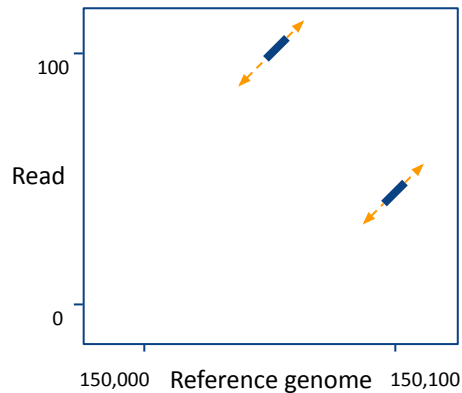
DNA Seq. Reads: Can originate from +ve or -ve strand

Reference genome: Only the +ve strand

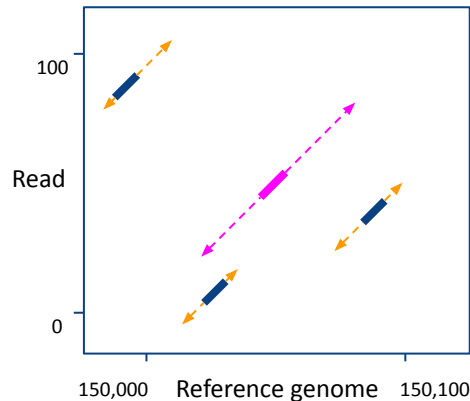
For each read (sequence to align)

1. Do Seed-Extend (original - forward orientation).
2. Flip it (reverse complement).
3. Do Seed-Extend (flipped - reverse orientation).
4. Take the best alignment from Fwd & Rev orientation.
5. If it's Rev, report seq as originating from the -ve strand .

Read in Fwd (original) Orientation



Read in Rev (flipped) Orientation



Read Mapping: Seed-Chain-Align

Read Mapping: Seed-Chain-Align

Aligning long, noisy sequences

Long-read data has unique properties due to sequencing technology

Read Mapping: Seed-Chain-Align

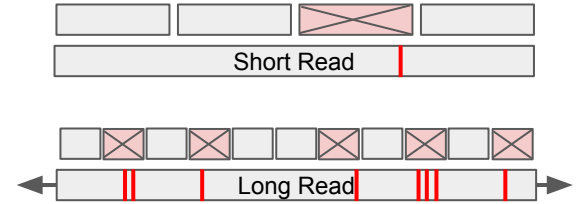
Aligning long, noisy sequences

Long-read data has unique properties due to sequencing technology

Error rate

- > Long-read data more noisy
- > ~1 error every 10 bp (improved by 2023)
- > Kmer size needs to be lower

Use of different K for short reads and long reads



Read Mapping: Seed-Chain-Align

Aligning long, noisy sequences

Long-read data has unique properties due to sequencing technology

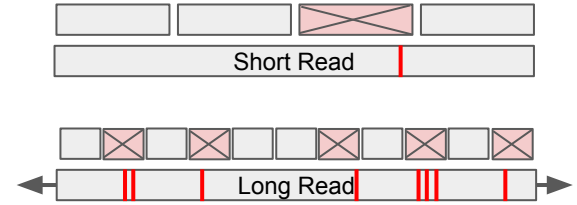
Error rate

- > Long-read data more noisy
- > ~1 error every 10 bp (improved by 2023)
- > Kmer size needs to be lower

Read length

- > Length ~10kb
- > Many kmers -> seeds for a single read (thousands)
- > Alignments for each seed: very expensive

Use of different K for short reads and long reads



Specificity vs Sensitivity when varying K

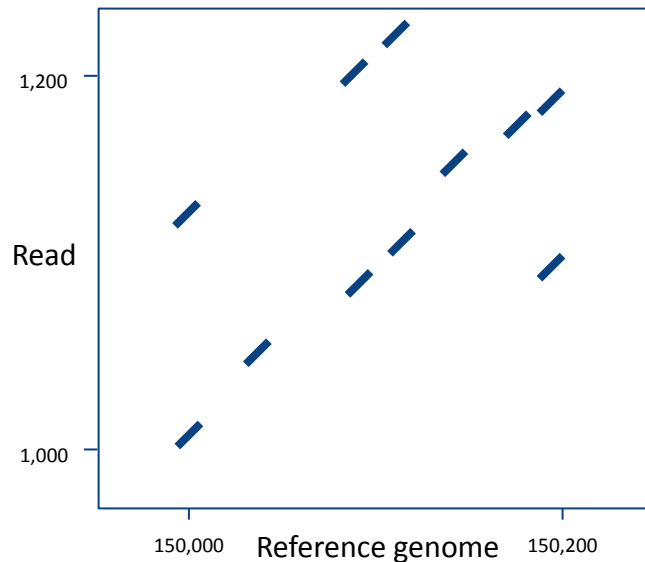


Read Mapping: Seed-Chain-Align

Can we use this to our advantage?

For a given read, many kmers will match the genome index

Expect the relative kmer positions to be similar in the read and genome



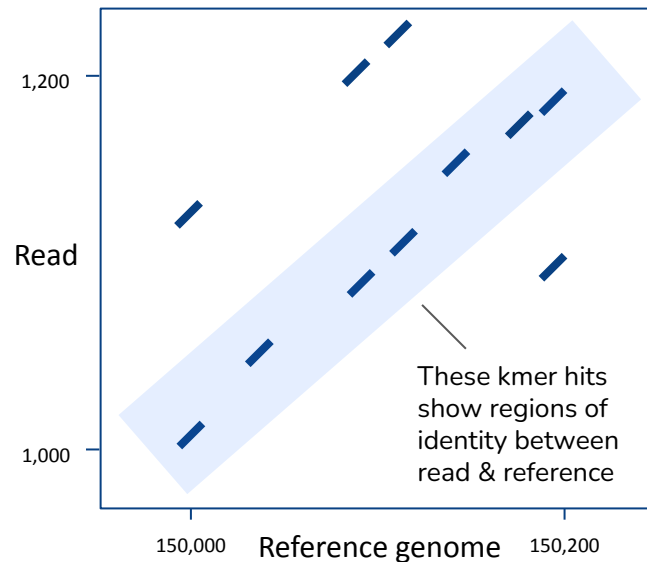
Read Mapping: Seed-Chain-Align

Can we use this to our advantage?

For a given read, many kmers will match the genome index

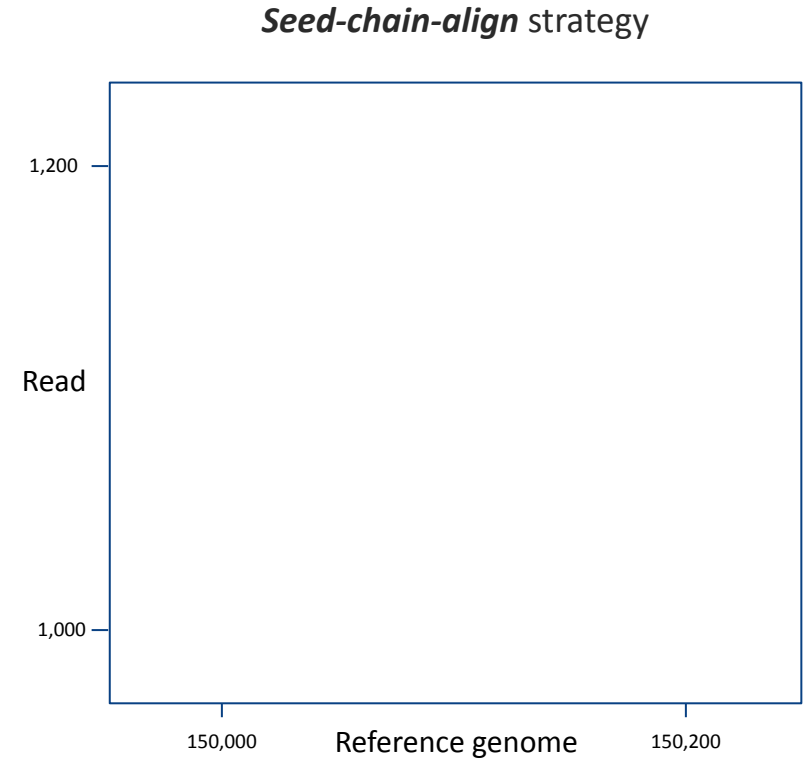
Expect the relative kmer positions to be similar in the read and genome

Chaining



Read Mapping: Seed-Chain-Align

Seed-Chain-Align

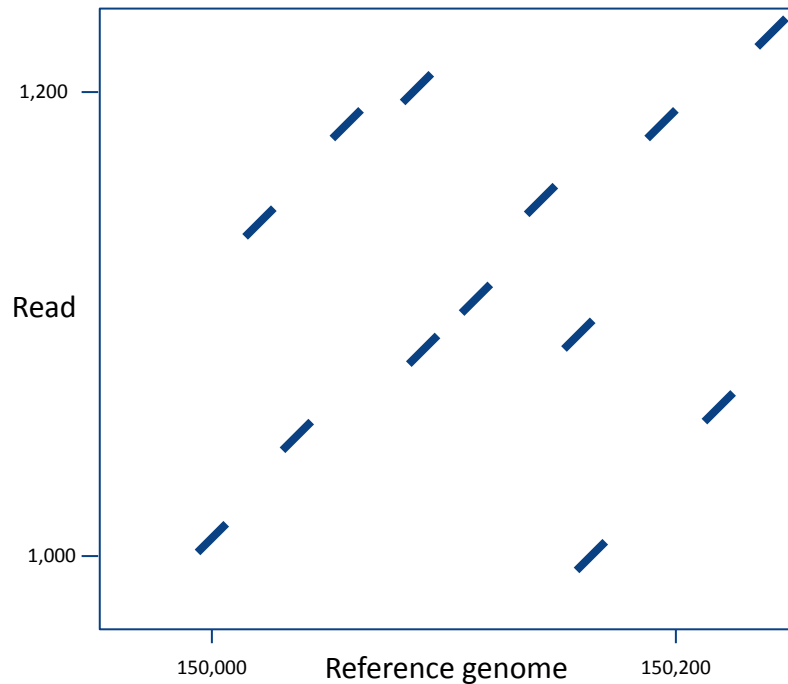


Read Mapping: Seed-Chain-Align

Seed-Chain-Align

1. Break the read into k-mers and look up their genomic positions in the index to find seeds
Note: don't need to extract every possible kmer from the read – can jump a little in between.
Common to extract a kmer every few – tens of base pairs.

Seed-chain-align strategy

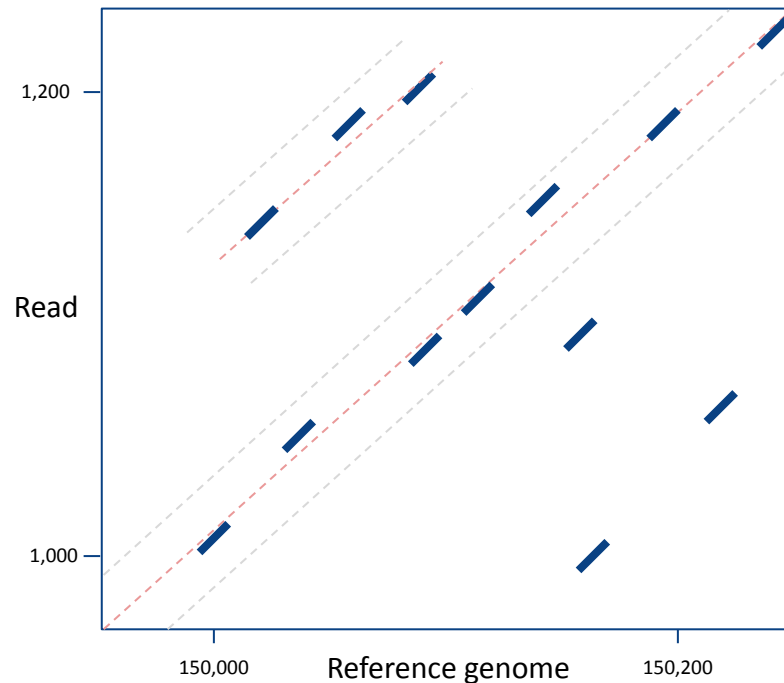


Read Mapping: Seed-Chain-Align

Seed-Chain-Align

1. Break the read into k-mers and look up their genomic positions in the index to find seeds
Note: don't need to extract every possible kmer from the read – can jump a little in between.
Common to extract a kmer every few – tens of base pairs.
2. Identify **colinear chains**

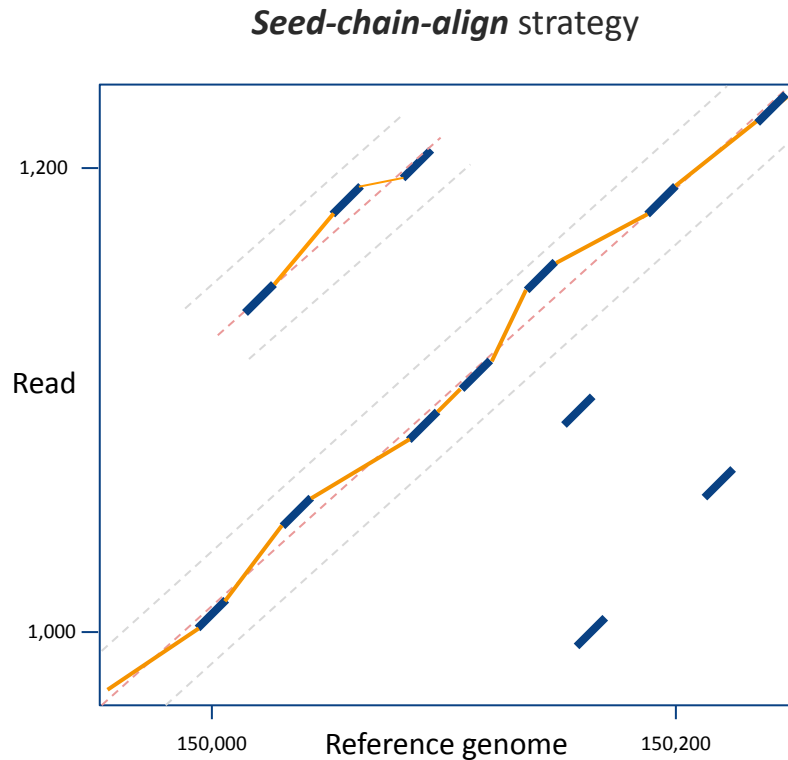
Seed-chain-align strategy



Read Mapping: Seed-Chain-Align

Seed-Chain-Align

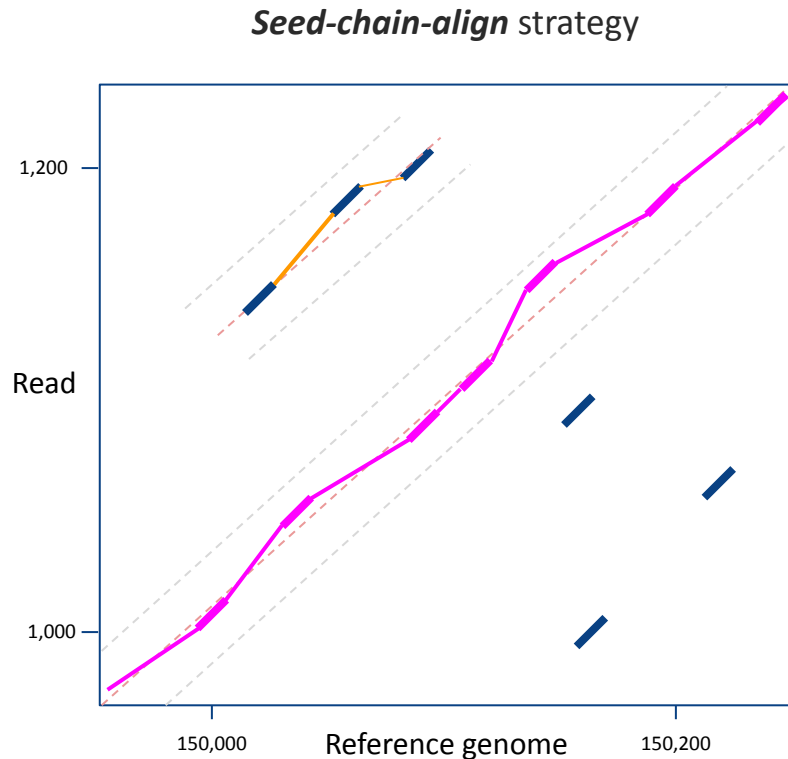
1. Break the read into k-mers and look up their genomic positions in the index to find seeds
Note: don't need to extract every possible kmer from the read – can jump a little in between.
Common to extract a kmer every few – tens of base pairs.
2. Identify **colinear chains**
3. For each: base-level alignments to fill gaps



Read Mapping: Seed-Chain-Align

Seed-Chain-Align

1. Break the read into k-mers and look up their genomic positions in the index to find seeds
Note: don't need to extract every possible kmer from the read – can jump a little in between.
Common to extract a kmer every few – tens of base pairs.
2. Identify **colinear chains**
3. For each: base-level alignments to fill gaps
4. Return the best location
(gap-filled chain with highest score)



Read Mapping: Seed-Chain-Align

Seed-Chain-Align

1. Break the read into k-mers and look up their genomic positions in the index to find seeds
Note: don't need to extract every possible kmer from the read – can jump a little in between.

Common to extract a kmer every few – to base pairs.

2. Identify *colin*

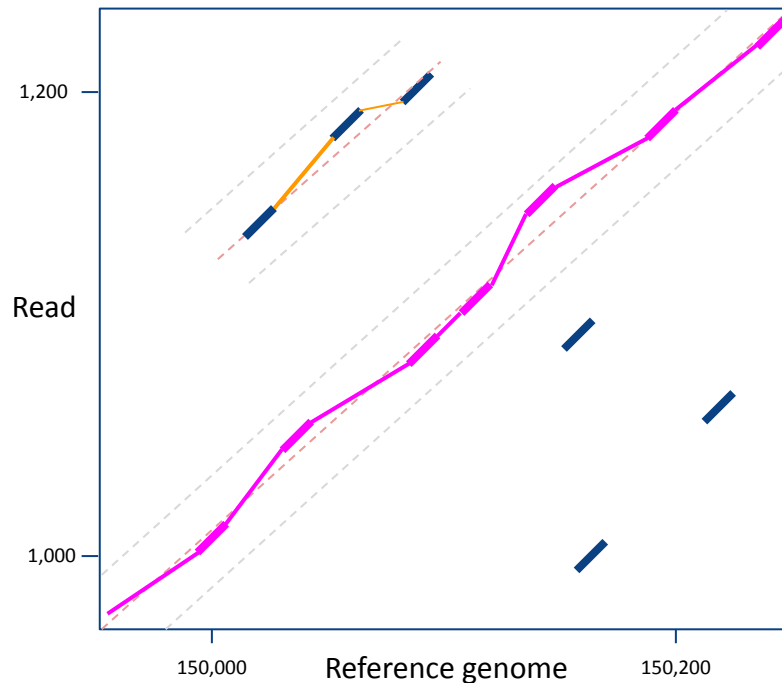
3. Set alignments to fill gaps

Return the best location

(gap-filled chain with highest score)

Expanded upon in week 3

Seed-chain-align strategy



BLAST

BLAST

Basic Local Alignment Search Tool (BLAST) - Extreme efficiency heuristics!

Finding conserved sequences (eg. genes)

Query sequence -> Massive database

Somewhere between Seed-Extend and Seed-Chain-Align


Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance. [Learn more](#)

NEWS

BLAST Quick Start guides!
Need some help getting started with BLAST?
Thu, 22 Jun 2023
[More BLAST news...](#)

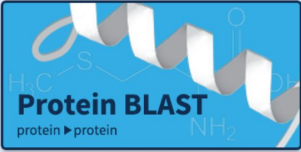
Web BLAST



Nucleotide BLAST
nucleotide ► nucleotide

blastx
translated nucleotide ► protein

tblastn
protein ► translated nucleotide



Protein BLAST
protein ► protein

BLAST Genomes

Search

Human Mouse Rat Microbes

BLAST

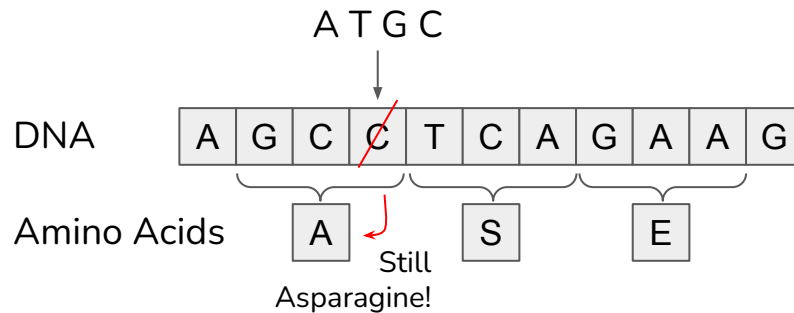
Extreme efficiency heuristics

BLAST

Extreme efficiency heuristics

1. DNA translated to protein seq for use

- > AA seq more conserved than DNA seq
- > Degeneracy / codon wobble
- > 3rd base in codon: multiple different nucleotides encode same AA
- > Eliminates meaningless DNA mismatches
- > More useful kmer hits
(mismatches, but coding seq undisturbed)



BLAST

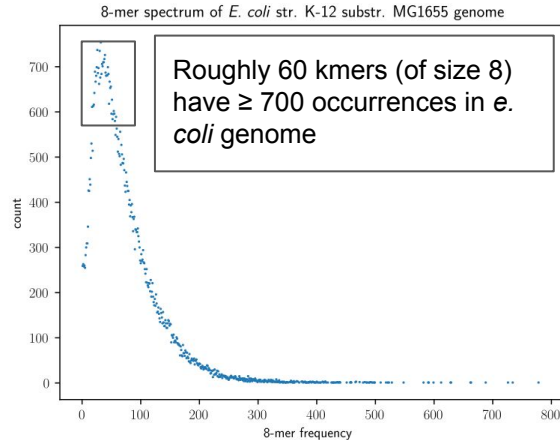
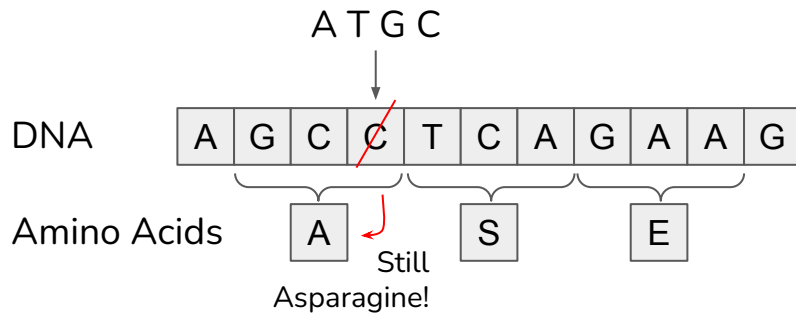
Extreme efficiency heuristics

1. DNA translated to protein seq for use

- > AA seq more conserved than DNA seq
- > Degeneracy / codon wobble
- > 3rd base in codon: multiple different nucleotides encode same AA
- > Eliminates meaningless DNA mismatches
- > More useful kmer hits
(mismatches, but coding seq undisturbed)

2. Low-complexity regions removed

- > Repetitive DNA
- > Functional elements generally not repetitive
- > Leads to uninformative kmer seeds (if retained)



Ytngargar via [wikipedia commons](#)

BLAST

Extreme efficiency heuristics

3. Kmers are fast, let's maximise their use

- > Generate kmer (word) 'neighbourhood' for query kmers
- > Referred to as
- > Allows mismatches in the seed step
(prev. only exact kmer matches in this lecture)
- > Similar to short ungapped alignment
- > Kmer matches must be above threshold score
(high scoring words)
- > Words in neighbourhood looked up in index

Generating Word Neighbourhood

Query: PQGEFG

Possible words		Neighbourhood
PQA = 12		PQG
PQV = 9		PEG
PEG = 15	Above Threshold	...
...		

BLAST

Extreme efficiency heuristics

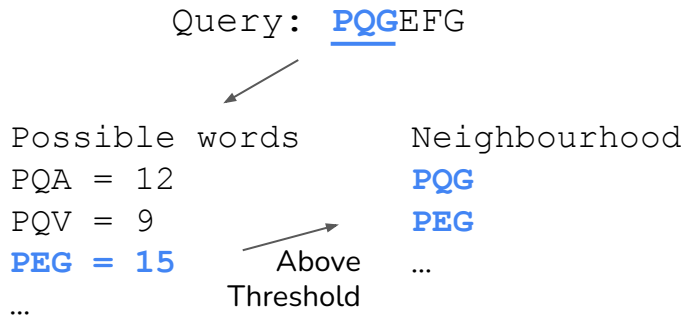
3. Kmers are fast, let's maximise their use

- > Generate kmer (word) 'neighbourhood' for query kmers
- > Referred to as
- > Allows mismatches in the seed step
(prev. only exact kmer matches in this lecture)
- > Similar to short ungapped alignment
- > Kmer matches must be above threshold score
(high scoring words)
- > Words in neighbourhood looked up in index

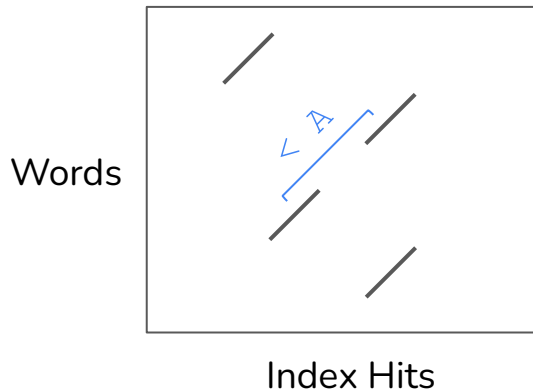
4. Identify nearby word hits on same diagonal

- > Similar to Chaining in Seed-Chain-Align
- > Word distance $< A$
- > Extends matching region to a:
"High-scoring Segment Pair (HSP)"

Generating Word Neighbourhood



Matching words on diagonal



BLAST

Extreme efficiency heuristics

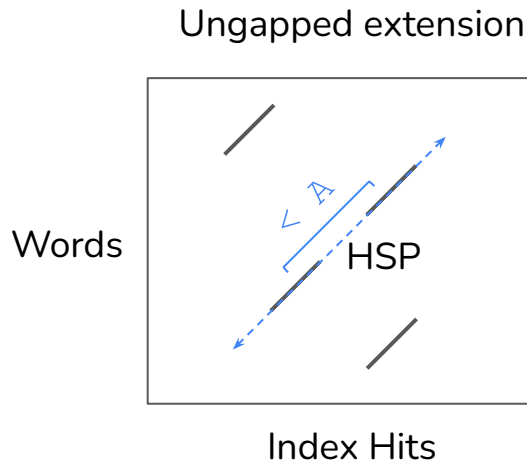
5. Ungapped extension

- > **Extend** HSP on either end
- > How far do we extend?
Stop when score < threshold (efficiency)
- > Rank HSPs by E-score & do cutoff

How likely this segment would appear in random sequence,
same size of our database?

Probability that this segment appears by simple chance.

Lower is better.



BLAST

Extreme efficiency heuristics

5. Ungapped extension

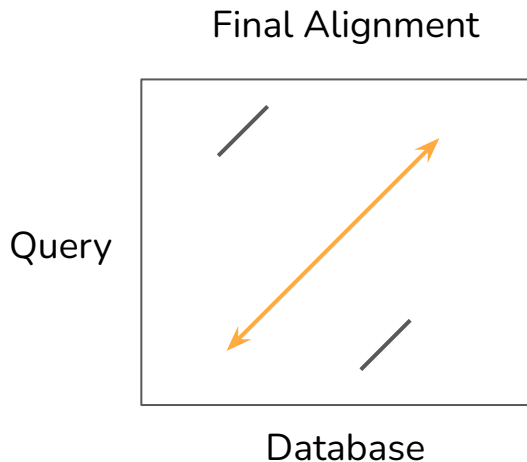
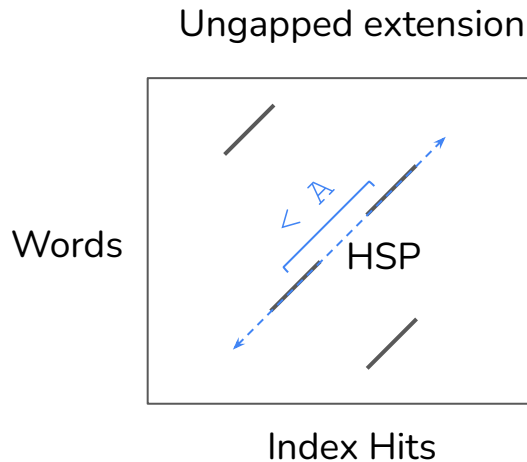
- > **Extend** HSP on either end
- > How far do we extend?
Stop when score < threshold (efficiency)
- > Rank HSPs by E-score & do cutoff

How likely this segment would appear in random sequence,
same size of our database?

Probability that this segment appears by simple chance.
Lower is better.

6. Gapped alignment

- > Local-alignment for HSP + end extension (early termination)
- > Recalculate E-score & report alignments greater than threshold



BLAST

Summary

Iterative approach

Ruthlessly & Continuously **reduces search space**

Very fast!

Clever thought applied to properties of genomic data.

The screenshot displays the NCBI BLAST web interface. At the top, the title "Basic Local Alignment Search Tool" is followed by a brief description of BLAST's function and a "Learn more" link. A sidebar on the right contains a "NEWS" section with a date and a link to "More BLAST news...". The main section, "Web BLAST", features three primary search options: "Nucleotide BLAST" (nucleotide to nucleotide), "blastx" (translated nucleotide to protein), and "tblastn" (protein to translated nucleotide). A "Protein BLAST" option (protein to protein) is also visible. At the bottom, the "BLAST Genomes" section includes a search bar with a placeholder text "Enter organism common name, scientific name, or tax id" and a "Search" button. Below the search bar, there are links for "Human", "Mouse", "Rat", and "Microbes".

Sequence Alignment

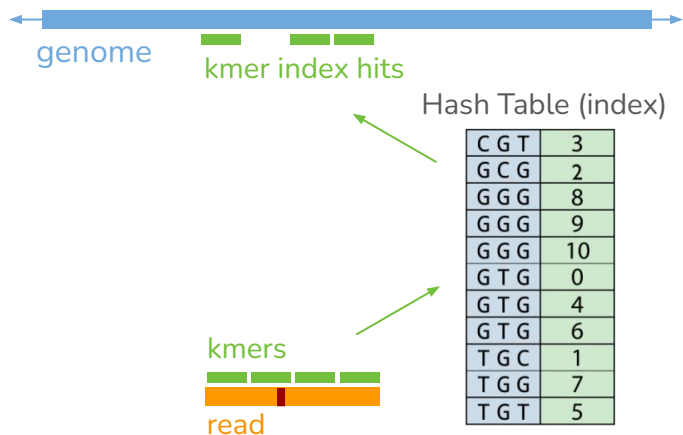
Kmers

breaking sequence into smaller pieces

Indexing

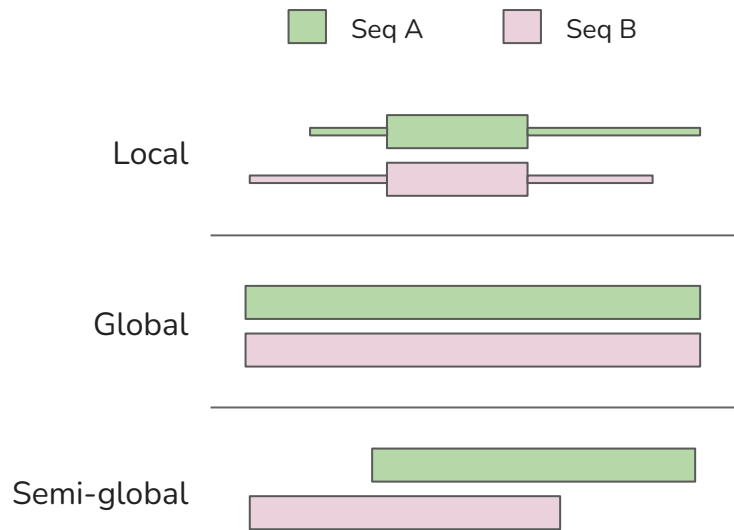
Use of kmers + hash tables

Fast lookup of subsequence matches



Alignment

Different variations on Levenshtein distance
...for different tasks



Thank you!

Don't forget your signed academic integrity statement
Due tomorrow!!

Today: Sequence Alignment II

Next time: Comparing Sequences I