Introduction to Dynamic Programming

The sequence alignment problem

Wilson Leung 08/2015

Outline

- Overview of the sequence alignment problem
- Calculate the optimal global alignment
- Characteristics of dynamic programming algorithms
- Calculate the optimal local alignment

Learning objectives

- Understand the theory behind sequence alignment
  - Become a **better informed user** of NCBI BLAST
- This presentation will not cover:
  - The BLAST algorithm
  - Parameter optimizations
  - Statistics for similarity searches (Karlin-Altschul theory)

Korf, I., Yandell, M., and Bedell, J. (2003). BLAST. O'Reilly Media, Inc.

Design goals

Drosophila melanogaster legless (lgs), mRNA
Sequence ID: ref|NM_143665.4| Length: 5357 Number of Matches: 6

Range 1: 1200 to 3209 GenBank Graphics Next Match Previous Match

Score	Expect	Identities	Gaps	Strand
2762 bits(3062)	0.0	1822/2016(90%)	6/2016(0%)	Plus/Minus

```
Query  3359  ATTACCAGCAGAGGACTGACCGAATGAGTCCAATTCCTTTGGTGATAGATGGGATAGAGG  3418
             || |||||||||||||||||||| || || |||||||||| |||||||| |||| ||
Sbjct  3209  ATCACCAGCAGAGGACTGACCGAAAGACTCAAATTCCTTTGGGGATAGATGAGATAAGGG  3150

Query  3419  GGTACTTGGGTTGCTGTTTAAGTTATGCGTAAGAACGCTTGATGATGCGGTATTTCTACT  3478
              ||||||||||||||| |||||||||||||||||||||||||||| ||| ||||||||||||
Sbjct  3149  GGTACTTGGGTTGCTGCTTAAGTTATGCGTAAGAACGCTTGACGATCCGGTATTTCTACT  3090
```

- Generate an alignment between **two sequences**
- Identify the "best" (**most parsimonious**) alignment
- Generate the best alignment "**quickly**"

Strategy #1: Visual inspection

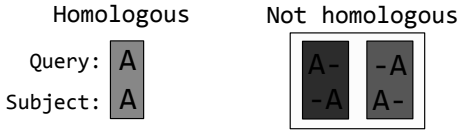
```
Query:  ATTACCAG
        || |||||
Subject: ATCACCAG
```

- Sequences must have **high percent identity**
- Applications:
  - PAM scoring matrix (align sequences with >= 85% identity)
  - Align mononucleotide runs during sequence improvement

Strategy #2: Enumerate all alignments

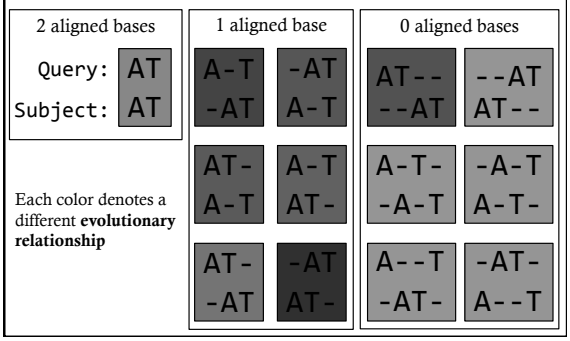
- Guaranteed to find the best alignment
- Does not scale
  - Combinatorial explosion
  - Two 300 bp sequences have **~10^{179}** possible alignments (Eddy 2004)
- **Brute-force** algorithm
  - Establish baseline performance and test cases
  - Identify patterns in the problem space

Apply the brute force algorithm to a single column of the alignment



- Three possible alignments for two 1 bp sequences
  - Query length (**M**) = 1; Subject length (**N**) =1
- Only two **biological interpretations**:
  - A in the query is **homologous** to A in the subject
  - A in the query is **not homologous** to A in the subject

Six possible relationships between the query and subject for M=2, N=2

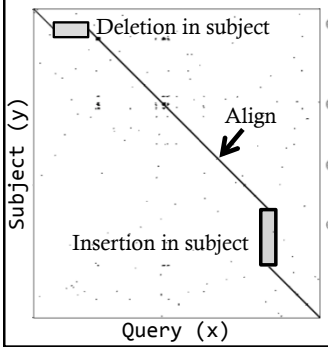


Each color denotes a different **evolutionary relationship**

Observations from the brute force alignment strategy

- Many of the possible alignments are **redundant**
  - Imply the same evolutionary relationship
- **Large number** of possible alignments
  - 13 possible alignments for sequences of length 2
- Can **ignore** many possible alignments
  - Many are suboptimal compared to the best alignment

Strategy #3: Dot plot

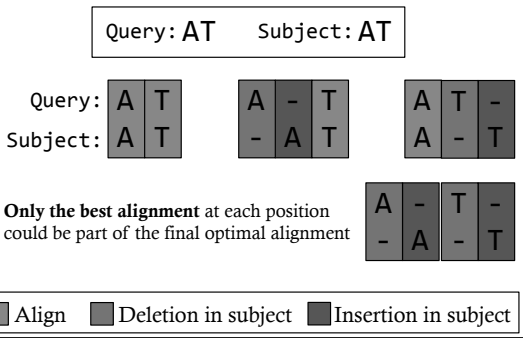


- Cell position (`i,j`):
  - i = Query position (x-axis)
  - j = Subject position (y-axis)
- **Draw a dot** at (`i,j`) if the two bases are **identical**
- **Connect the dots** to make a line (alignment)
- Level of noise depends on repeat density
  - Use **longer words** and higher cutoff scores to reduce noise

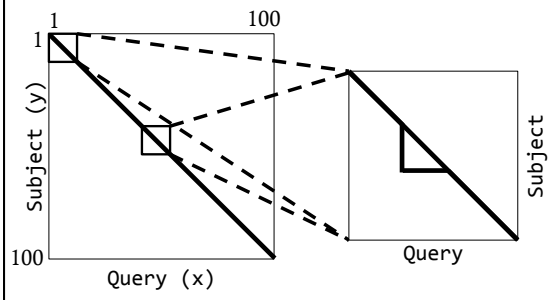
Assessment of the three sequence alignment strategies

- Infeasible to examine all possible alignments
  - Need to reduce the search space
- Only a small subset of alignments are "interesting"
  - Many alignments are redundant
- Connect the dots in the dot plot to create an alignment
  - Consider the **cumulative levels of similarity**

The optimal alignment is composed of **smaller optimal alignments**

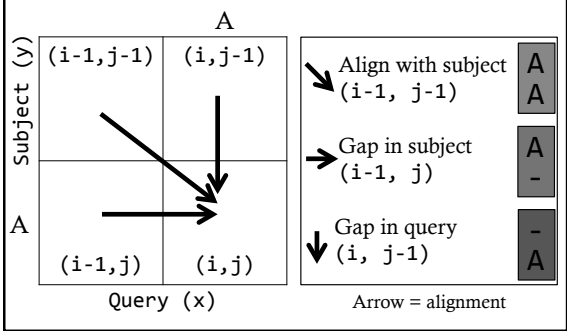


Partition the alignment problem into smaller subproblems



Assume the query and subject sequences are the same

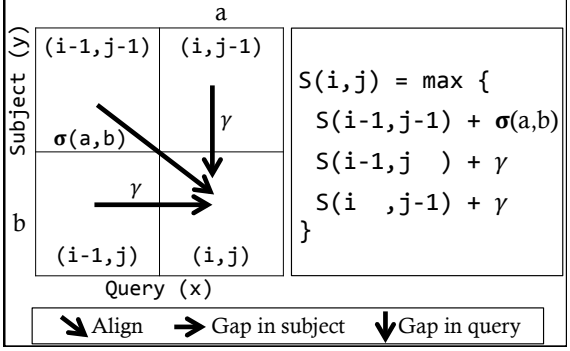
Three different ways to reach cell (i,j) in the alignment matrix



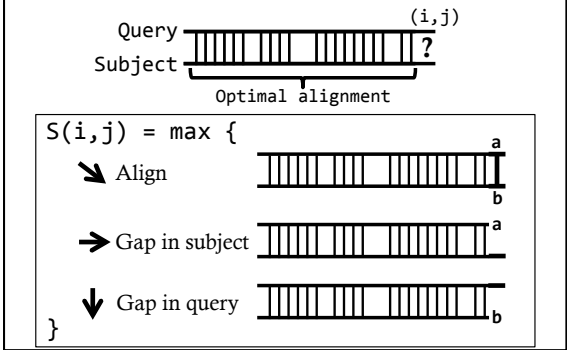
Construct a scoring system to measure **similarity** between two sequences

- Scoring system for the aligned state: **σ**
  - **σ**(a, b) = Score for aligning **a** in query with **b** in subject
    - **σ**(A, A) = **Bonus** for aligning **A** in query with **A** in subject
    - **σ**(A, T) = **Penalty** for aligning **A** in query with **T** in subject
- Penalty for adding a gap: γ
- More sophisticated scoring systems take transitions, transversions, affine gap penalty into account
  - Pearson WR. Selecting the Right Similarity-Scoring Matrix. Curr Protoc Bioinformatics. 2013;43:3.5.1-3.5.9.

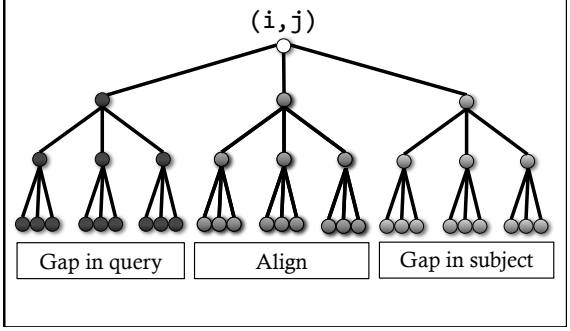
Recursive definition for the optimal cumulative alignment score S(i,j)



Determine **the best way** to reach cell (i,j) if it were part of the optimal alignment



Use the maximum score at each cell to eliminate entire branch of suboptimal alignments



Cumulative score S(i,j) encapsulates the alignment decisions up to position (i,j)

- All potential optimal alignments that go through cell (i,j) have the same ancestry
  - Re-use the cumulative alignment score (**memoization**)
- Gaps are described by the cumulative score
  - Do not affect the coordinates of the alignment matrix
- Do not know the optimal alignment until we complete the entire alignment matrix
  - Optimal alignment has the **highest cumulative score**

Needleman-Wunsch algorithm (global alignment)

(Query length: M; Subject length: N)

- Construct a **(M+1) x (N+1)** matrix
  - Extra column and row = gaps at the beginning of the alignment
- Fill in the cells in the first row and first column with the cumulative gap costs
- Calculate the **maximum score** for subsequent cells (i,j)
  - Keep track of the decision that leads to the maximum score (S)

$$S(i,j) = \max \begin{cases} S(i-1,j-1) + \sigma(a,b) \\ S(i-1,j\ \ ) + \gamma \\ S(i\ \ ,j-1) + \gamma \end{cases}$$

Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol. 1970 Mar;48(3):443-53.

Initialize the alignment matrix

(Match = +5; Mismatch = -2; Gap = **-6**)

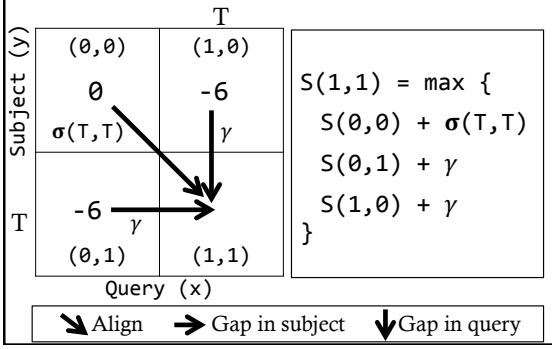
		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	-6	-12	-18	-24	-30	-36	-42	-48
1	T	-6								
2	T	-12								
3	C	-18								
4	A	-24								
5	T	-30								
6	A	-36								

Subject

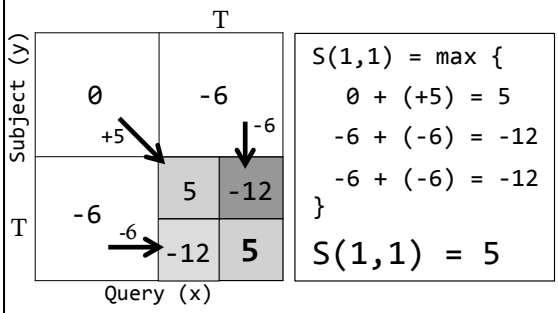
Query

(Eddy, 2004)

Calculate the possible scores for the cell at position (1,1)

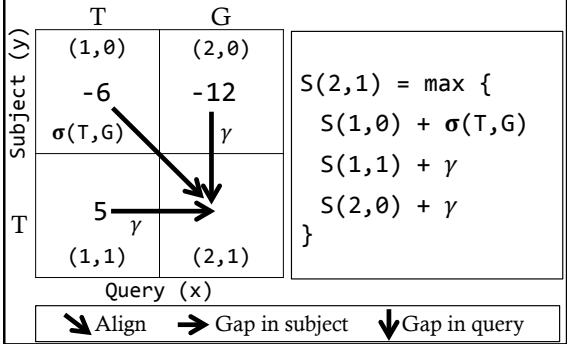


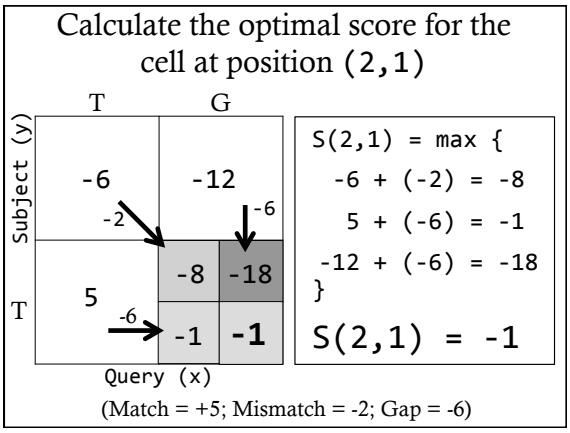
Calculate the optimal score for the cell at position (1,1)

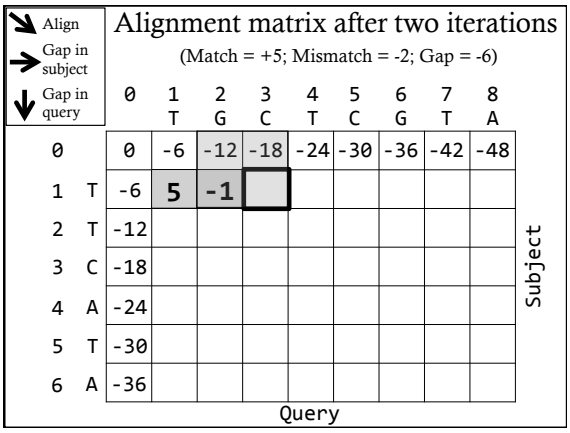


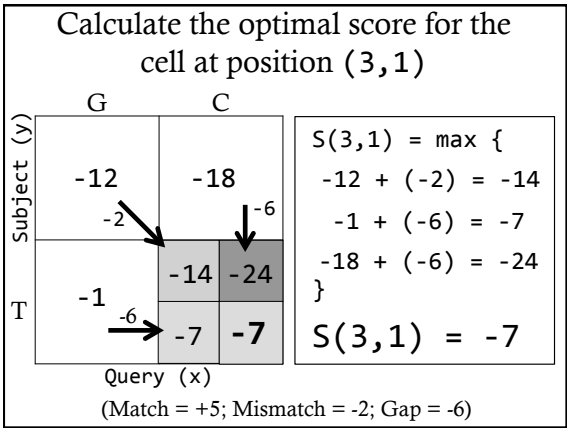
(Match = +5; Mismatch = -2; Gap = -6)

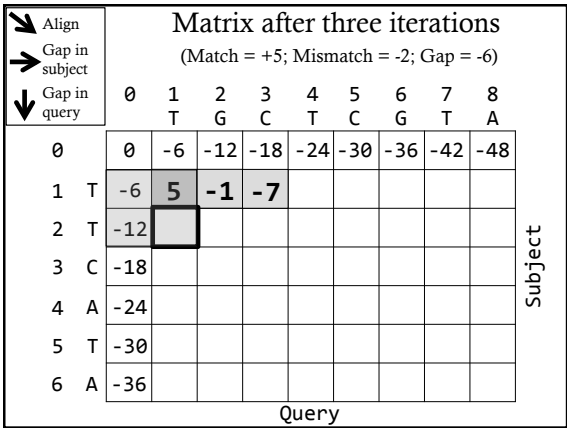
Calculate the possible scores for the cell at position (2,1)

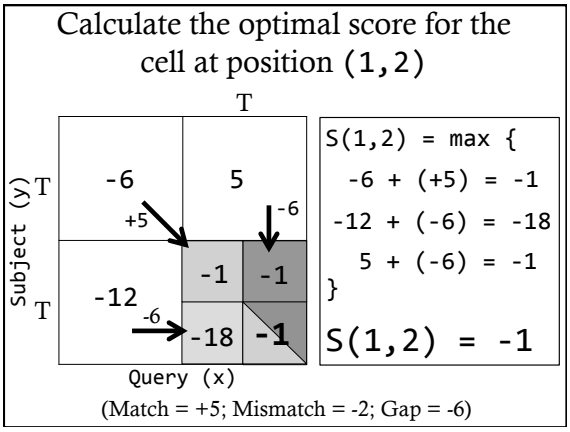


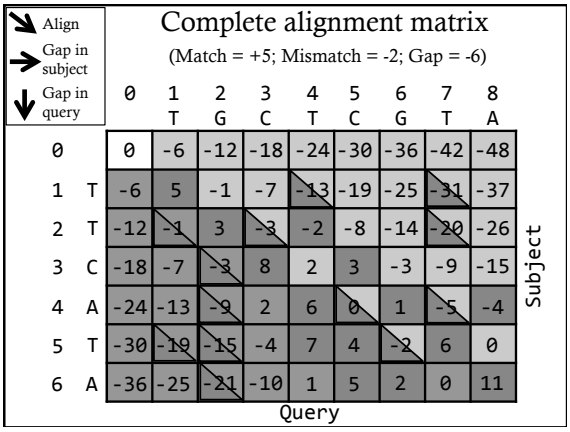

Calculate the optimal score for the cell at position (2,1)
T
G
Subject (y)
T
-6
-12
-2
-6
-8
-18
5
-6
-1
-1
Query (x)
S(2,1) = max {
-6 + (-2) = -8
5 + (-6) = -1
-12 + (-6) = -18
}
S(2,1) = -1
(Match = +5; Mismatch = -2; Gap = -6)


Align
Gap in subject
Gap in query
Alignment matrix after two iterations
(Match = +5; Mismatch = -2; Gap = -6)
0 1 2 3 4 5 6 7 8
T G C T C G T A
0 0 -6 -12 -18 -24 -30 -36 -42 -48
1 T -6 5 -1
2 T -12
3 C -18
4 A -24
5 T -30
6 A -36
Subject
Query


Calculate the optimal score for the cell at position (3,1)
G
C
Subject (y)
T
-12
-18
-2
-6
-14
-24
-1
-6
-7
-7
Query (x)
S(3,1) = max {
-12 + (-2) = -14
-1 + (-6) = -7
-18 + (-6) = -24
}
S(3,1) = -7
(Match = +5; Mismatch = -2; Gap = -6)

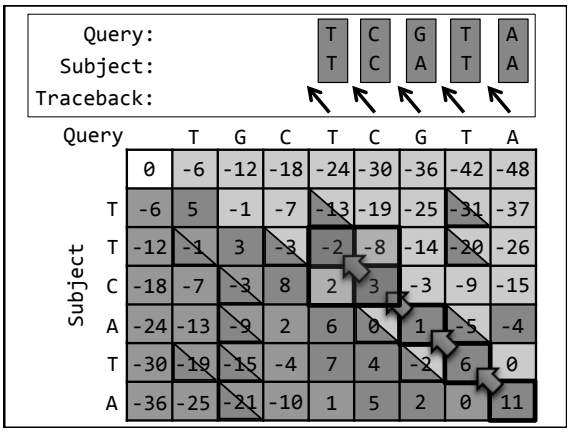

Align
Gap in subject
Gap in query
Matrix after three iterations
(Match = +5; Mismatch = -2; Gap = -6)
0 1 2 3 4 5 6 7 8
T G C T C G T A
0 0 -6 -12 -18 -24 -30 -36 -42 -48
1 T -6 5 -1 -7
2 T -12
3 C -18
4 A -24
5 T -30
6 A -36
Subject
Query


Calculate the optimal score for the cell at position (1,2)
T
Subject (y)
T
-6
5
+5
-6
-1
-1
-12
-6
-18
-1
Query (x)
S(1,2) = max {
-6 + (+5) = -1
-12 + (-6) = -18
5 + (-6) = -1
}
S(1,2) = -1
(Match = +5; Mismatch = -2; Gap = -6)

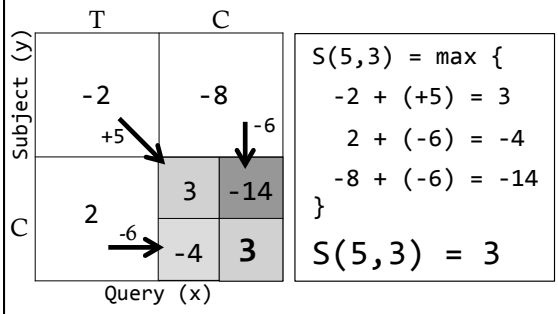

Align
Gap in subject
Gap in query
Complete alignment matrix
(Match = +5; Mismatch = -2; Gap = -6)
0 1 2 3 4 5 6 7 8
T G C T C G T A
0 0 -6 -12 -18 -24 -30 -36 -42 -48
1 T -6 5 -1 -7 -13 -19 -25 -31 -37
2 T -12 -1 3 -3 -2 -8 -14 -20 -26
3 C -18 -7 -3 8 2 3 -3 -9 -15
4 A -24 -13 -9 2 6 0 1 -5 -4
5 T -30 -19 -15 -4 7 4 -2 6 0
6 A -36 -25 -21 -10 1 5 2 0 11
Subject
Query

Use **traceback** to recover the optimal alignment

- Start from the cell within the last row and last column that has the highest score
- **Recall the step (color)** that leads to this optimal score
  - Report this step in the alignment output
  - **All the alignment decisions have already been made**
- Repeat until we reached the beginning of the sequence
- Two options if multiple paths produce the same score
  - Report only one of the paths (pick arbitrarily)
  - Report all paths with the optimal score

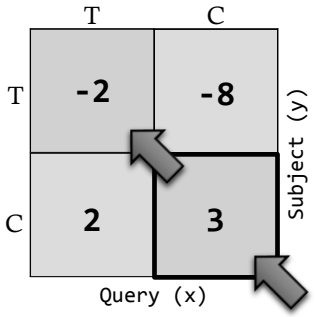


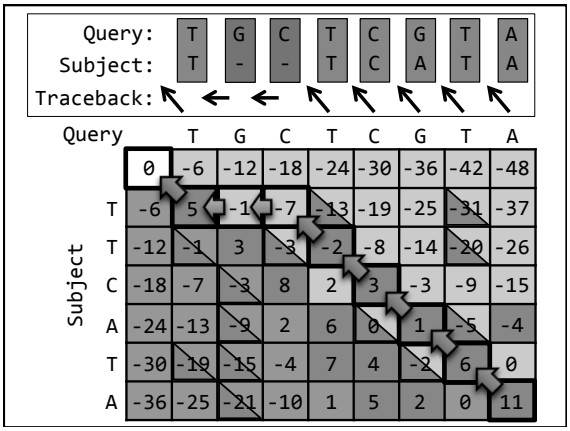
Calculate the optimal score for the cell at position (5,3)



(Match = +5; Mismatch = -2; Gap = -6)

Traceback must follow the steps that produce the optimal **cumulative** global alignment score





The Needleman-Wunsch algorithm is an example of a **dynamic programming** algorithm

- Problem must satisfy two criteria:
  - **Optimal substructure**
    - Optimal solution to the complete problem is composed of optimal solutions to the subproblems
  - **Overlapping problems**
    - **Re-use the results** for the subproblems (*e.g.*, lookup table)
- Many bioinformatics problems satisfy these criteria
  - Sequence alignment, gene prediction, RNA-folding

Bellman B. The theory of dynamic programming. Bulletin of the American Mathematical Society. 1954; 60(6):503–516

Smith-Waterman algorithm (local alignment)

(Query length: M; Subject length: N)

- Three changes to the Needleman-Wunsch algorithm:
  - The minimum score for a cell is **zero**
    - Initiate a new alignment when the cumulative score is negative
  - Begin traceback from the cell within **the entire matrix** that has the highest score
  - Terminate traceback when the score is zero

$$S(i,j) = \max \begin{cases} S(i-1,j-1) + \sigma(a,b) \\ S(i-1,j\ \ ) + \gamma \\ S(i\ \ ,j-1) + \gamma \\ 0 \end{cases}$$

Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol. 1981 Mar 25;147(1):195-7.

Global versus local alignments

- Global alignment
  - Optimal alignment along the entire length of two sequences
  - Compare protein sequences to identify orthologs
- Local alignment
  - Optimal alignment between parts of two sequences
  - Identify conserved domains within protein sequences
- Glocal (semi-global) alignment
  - Optimal global alignment for one sequence; optimal local alignment for the other sequence
  - Map a coding exon against a genomic sequence

Initialize the local alignment matrix

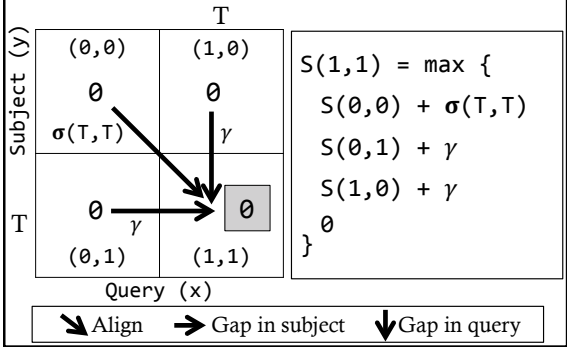
(Match = +5; Mismatch = -2; Gap = -6)

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	0	0	0	0	0	0	0	0
1	T	0								
2	T	0								
3	C	0								
4	A	0								
5	T	0								
6	A	0								

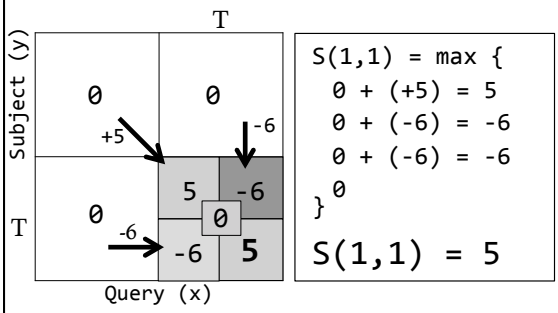
Subject

Query

Calculate the possible local alignment scores for the cell at position (1,1)



Calculate the optimal local alignment score for the cell at position (1,1)



(Match = +5; Mismatch = -2; Gap = -6)

Local alignment matrix

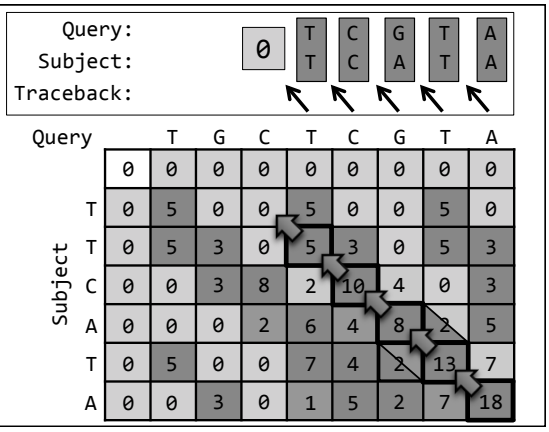
Align
Gap in subject
Gap in query

(Match = +5; Mismatch = -2; Gap = -6)

		0	1 T	2 G	3 C	4 T	5 C	6 G	7 T	8 A
0		0	0	0	0	0	0	0	0	0
1	T	0	5	0	0	5	0	0	5	0
2	T	0	5	3	0	5	3	0	5	3
3	C	0	0	3	8	2	10	4	0	3
4	A	0	0	0	2	6	4	8	2	5
5	T	0	5	0	0	7	4	2	13	7
6	A	0	0	3	0	1	5	2	7	18

Subject

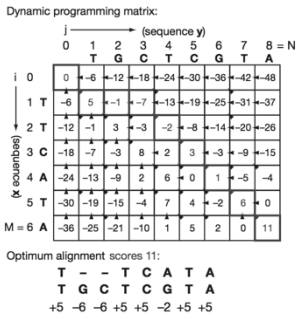
Query



Techniques to improve the performance of sequence alignment

- Time and space complexity: O(MN)
  - **Double the size** of the two sequences leads to a **four-fold increase** in the amount of time and space required
- Reduce memory requirement
  - Myers EW, Miller W. Optimal alignments in linear space. Comput Appl Biosci. 1988 Mar;4(1):11-7.
- Fill the matrix in parallel (SIMD, CUDA)
  - Farrar M. Striped Smith-Waterman speeds database searches six times over other SIMD implementations. Bioinformatics. 2007 Jan 15;23(2):156-61.
- Find **high-scoring** instead of the best alignment
  - Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990 Oct 5;215(3):403-10.

Questions?



Eddy SR. What is dynamic programming? Nat Biotechnol. 2004 Jul;22(7):909-10.