Lecture Title and Date

Sequence Comparison, 02/10

Objectives of the Lecture

By the end of this lecture, students should be able to:

1. Understand the role of gap penalties in sequence alignment.
2. Incorporate gap penalties into dynamic programming and compute the sum matrix with gap penalties.
3. Understand the role of similarity (substitution) matrices in scoring sequence similarities.
4. Describe how the similarity matrices are constructed.
5. Compare the difference between global and local alignments.
6. Understand key modifications for local alignment and adapt them to global alignment to compute local alignment.

Key Concepts and Definitions

- **Gap Penalties**: a negative score assigned to penalize gaps introduced between characters when aligning two sequences.
- **Dynamic Programming for Alignment**: a computational technique that optimally aligns two sequences by breaking the problem into smaller, overlapping subproblems, solving them iteratively, and constructing the best alignment using a defined scoring scheme.
- **Similarity (Substitution) Matrices**: a table that scores the frequency at which one amino acid is likely to be replaced by another during evolution, guiding alignment decisions based on evolutionary substitution probabilities.
- **Global Alignment**: an alignment method that aligns the entire length of two sequences, comparing every character from beginning to end while considering match scores, mismatch penalties, and gap penalties to optimize alignment.
- **Local Alignment**: an alignment method that identifies the most similar regions within two sequences, aligning only high-scoring subsequences while allowing gaps and ignoring less similar regions.

Main Content/Topics

Gap penalties (fixed or affine)

- Definition: Penalties by introducing gaps(i.e. Not going from i,j to i-1, j-1)
- Formula: GAP = a + bN
  * a = cost of opening a gap
  * b = cost of extending gap by one (affine)
  * N = length of gap

- Examples:
  (1) **Linear Model**: Fixed penalty per gap (e.g., a= 0.5, b= 0, GAP = 0.5, regardless of the length).
  (2) **Affine Model**: Separate costs for opening (a) and extending (b) gaps (e.g., a= 0.5, b= 0.1, GAP = 0.5 + 0.1b)

  Example:
  ATGCAAAAT
  ATG-AAAAT 0.5
  ATG--AAAT 0.5 + (1)(0.1) =0.6
  ATG---AAT 0.5 + (2)(0.1) =0.7

(Global) Sequence Alignment via Dynamic Programming (Needleman-Wunsch)

1. Dot Plot Creation: Binary similarity matrix (1 for matches, 0 otherwise).
2. Sum Matrix Calculation **with GAP**:
   Score = current cell value + max(diagonal, row-gap, column-gap).

$$S(i,j) = \text{match/mismatch} + \max \begin{cases} S(i-1, j-1) \\ \max_k \left(S(i-k, j) - \text{GAP}\right) \\ \max_l \left(S(i, j-l) - \text{GAP}\right) \end{cases}$$

3. Traceback: Identify the alignment path from the highest score
4. Results: Get output alignment

Example: Aligning a 4-mer

	C	R	P	M
C	1			
R		1		
B				
P			1	

	C	R	P	M
C	1			
R		2	0	0
B	1	1	0	0
P	0	0	1	0

	C	R	P	M
C	3	1	0	0
R	1	2	0	0
B	1	1	0	0
P	0	0	1	0

	C	R	P	M
C	3	1	0	0
R	1	2	0	0
B	1	1	0	0
P	0	0	1	0

Similarity (Substitution) Matrices

Common ones

- PAM Matrices:
  - PAM70: For closely related sequences (e.g., aligning orthologs like hemoglobin α/β).
  - PAM250: For distantly related sequences (e.g., cytochrome C across mammals).
  PAM70 vs. PAM250 yield different evolutionary inferences
- BLOSUM62: Standard for local alignment (e.g., conserved blocks in globins).

How to get the similarity matrics

- Manually align protein structures/sequences
- Analyze amino acid substitutions at conserved structural positions
- Log-odds Calculation:

$$S(aa_1, aa_2) = \log_2 \left(\frac{\text{freq}_{\text{observed}}(aa_1 \leftrightarrow aa_2)}{\text{freq}_{\text{expected}}(aa_1) \cdot \text{freq}_{\text{expected}}(aa_2)} \right)$$

Example:

A-R pair observed 10× less than random expectation

Explanation:

+ value: More frequent than random (conserved)

0: Random occurrence

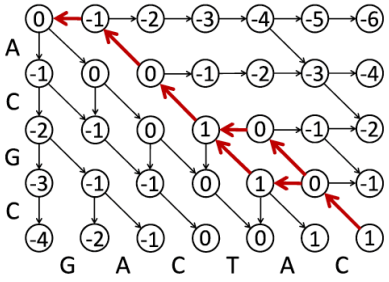
- value: Less frequent than random (disfavored)

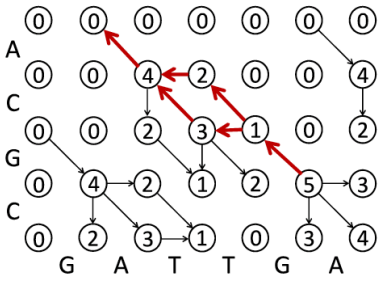
Global vs. Local Alignment

Needleman-Wunsch (Global): Aligns entire sequences.
Smith-Waterman(Local): Finds optimal subsequences (e.g., identifying conserved domains).
Key modifications for local alignment:

- Using negative mismatch scores
- Zeros as the minimum score in the matrix
- Find the best score anywhere in the matrix (not just row or column)

These modifications allow for searching the high score subsequences, which are not penalized for their global effects, don't include areas of poor match, and can occur anywhere.






```
-ACG-C      -AC-GC          AC-G      A-CG
 ||   |  or  ||   |          |   |  or  |   |
GACTAC      GACTAC          GATTGA    GATTGA
```

Global : Match: +1 Mismatch: 0 Gap: -1

Local : Match: +4 Mismatch: -1 Gap: -2

Discussion/Comments

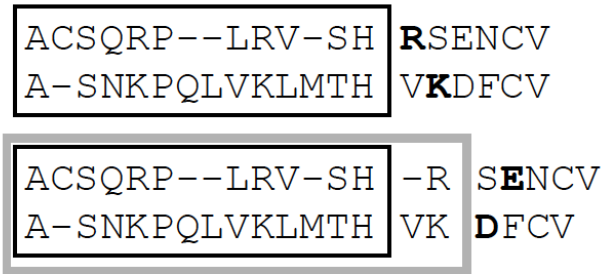
Key idea in dynamic programming

The core principle of dynamic programming (DP) in sequence alignment is step-wise optimality:

- The best alignment ending at positions *i* (Sequence 1) and *j* (Sequence 2) is derived by adding the score for aligning residues *i* and *j* to the optimal alignment of all previous residues (up to *i-1* and *j-1*).
- Once a partial alignment is computed (e.g., aligning residues up to *i-1* and *j-1*), it remains fixed. Subsequent steps build on this foundation without revisiting earlier decisions.

  Example:

  If aligning residue R (Sequence 1) to K (Sequence 2) retroactively changes the best alignment of prior residues (e.g., N-terminal regions), it breaks DP's assumption of independence between steps. This could lead to suboptimal or inconsistent global alignments.



Suboptimal Alignments

- Multiple traceback paths exist (e.g., two alignments with scores = 8).

Evolutionary Distance & Matrix Selection

- Different matrices are appropriate at different evolutionary distances

Required Reading

- Smith, T., & Waterman. (1981). Journal of Molecular Biology, 147(1), 195–197. Identification of common molecular subsequences. https://doi.org/10.1016/0022-2836(81)90087-5 (http://www.gersteinlab.org/courses/452/10-spring/pdf/sw.pdf) (Just Algorithm Section)

Identification of common molecular subsequences provides a general overview of existing homology algorithms and a mathematical proof for what is now known to be the

Smith-Waterman local alignment algorithm. Although the proof requires some in-depth understanding to comprehend, the sample matrix provided nicely demonstrates the process of traceback under local alignment.

The two molecular sequences will be $\underset{\sim}{A} = a_1 a_2 \ldots a_n$ and $\underset{\sim}{B} = b_1 b_2 \ldots b_m$. A similarity $s(a,b)$ is given between sequence elements a and b. Deletions of length k are given weight W_k. To find pairs of segments with high degrees of similarity, we set up a matrix H. First set

$$H_{k0} = H_{0l} = 0 \text{ for } 0 \leq k \leq n \text{ and } 0 \leq l \leq m.$$

Preliminary values of H have the interpretation that H_{ij} is the maximum similarity of two segments *ending* in a_i and b_j, respectively. These values are obtained from the relationship

$$H_{ij} = \max\{H_{i-1,j-1} + s(a_i,b_j), \max_{k \geq 1}\{H_{i-k,j} - W_k\}, \max_{l \geq 1}\{H_{i,j-l} - W_l\}, 0\}, \quad (1)$$

$1 \leq i \leq n$ and $1 \leq j \leq m$.

The formula for H_{ij} follows by considering the possibilities for ending the segments at any a_i and b_j.

(1) If a_i and b_j are associated, the similarity is

$$H_{i-1,j-1} + s(a_i,b_j).$$

(2) If a_i is at the end of a deletion of length k, the similarity is

$$H_{i-k,j} - W_k.$$

(3) If b_j is at the end of a deletion of length l, the similarity is

$$H_{i-k,j} - W_l.$$

(4) Finally, a zero is included to prevent calculated negative similarity, indicating no similarity up to a_i and b_j.†

References ISL/ESL (if any)

This lecture primarily focuses on sequence comparison and bioinformatics algorithms, but there are statistical methods that are essential for understanding sequence alignment and scoring techniques. While these statistical methods might not be explicitly covered in the lecture, they are crucial when analyzing biological sequences in genomic studies. Here are some suggested references from **ISL** and **ESL** that may be useful:

ISL: *Chapter 5* on *Resampling Methods* is highly relevant when assessing sequence alignment accuracy. The chapter discusses **cross-validation** and the **bootstrap**, which are essential for evaluating scoring matrices such as PAM and BLOSUM. Since sequence comparison often involves statistical inference on substitution patterns, these methods provide a robust framework for understanding the reliability of sequence alignments.

ESLII: *Chapter 18* on *High-Dimensional Problems ($p \gg N$)* is particularly relevant when working with biological sequences and genomic data, where the number of features (positions in a sequence) far exceeds the number of samples. This chapter addresses overfitting, variance issues, and regularization techniques, which are crucial when designing **substitution matrices** (e.g., PAM, BLOSUM) and optimizing alignment algorithms.

Other suggested references for many of the key concepts

- Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48(3), 443-453.
    - Introduces **Needleman-Wunsch global alignment algorithm** using dynamic programming.
- https://bioboot.github.io/bimm143_W20/class-material/nw/
    - Interactive Demo of **Needleman-Wunsch global alignment algorithm.**
- Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids – Richard Durbin, Sean R. Eddy, Anders Krogh, Graeme Mitchison.
    - This book is a foundational reference for **sequence alignment**, substitution matrices, and dynamic programming.
- Algorithms on Strings, Trees, and Sequences – Dan Gusfield
    - A deeper dive into the algorithms behind **sequence comparison and bioinformatics applications**.
- Substitution matrices - Stephen F Altschul
    - In-depth explanation of **substitution matrices** and their use in alignment.
- Henikoff, S., & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *PNAS, 89(22), 10915-10919.*
    - Describes **BLOSUM similarity matrices**.
- Mount D. W. (2008). Using gaps and gap penalties to optimize pairwise sequence alignments. *CSH protocols*, *2008*, pdb.top40. https://doi.org/10.1101/pdb.top40
    - Introduces how to use **gap penalties** for alignment optimization.
- Gusfield D. Inexact Matching, Sequence Alignment, Dynamic Programming. In: *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press; 1997:209-211.
    - Chapter 3 on Inexact Alignment, Sequence Alignment, **Dynamic Programming.**
- Rosenberg, M. S. (Ed.). (2009). *Sequence Alignment: Methods, Models, Concepts, and Strategies* (1st ed.). University of California Press. http://www.jstor.org/stable/10.1525/j.ctt1pps7t
    - Chapter 1 on Sequence Alignment provides a great overview of **multiple, global, local, and pairwise alignment**, and **dynamic programming**.
- Nalbantoğlu Ö. U. (2014). Dynamic programming. *Methods in molecular biology (Clifton, N.J.)*, *1079*, 3–27. https://doi.org/10.1007/978-1-62703-646-7_1
    - Introduction to **pairwise alignment**.