Scoring Alignments

Genome 373
Genomic Informatics
Elhanan Borenstein
A quick review

- Course logistics
- Genomes (so many genomes)
- The computational bottleneck
Informatic Challenges: Examples

• Sequence comparison:
  – Find the best alignment of two sequences
  – Find the best match (alignment) of a given sequence in a large dataset of sequences
  – Find the best alignment of multiple sequences

• Motif and gene finding

• Relationship between sequences
  – Phylogeny

• Clustering and classification

• Many many many more ...
Motivation

- Why compare two protein or DNA sequences?
Motivation

• Why compare two protein or DNA sequences?
  – Determine whether they are descended from a common ancestor (homologous)
  – Infer a common function
  – Locate functional elements (motifs or domains)
  – Infer protein or RNA structure, if the structure of one of the sequences is known
  – Analyze sequence evolution
  – Infer the species from which a sequence originated
Informatic Challenges: Examples

• **Sequence comparison:**
  – Find the best alignment of two sequences
  – Find the best match (alignment) of a given sequence in a large dataset of sequences
  – Find the best alignment of multiple sequences

• Motif and gene finding

• Relationship between sequences
  – Phylogeny

• Clustering and classification

• Many many many more ...
Informatic Challenges: Examples

• Sequence comparison:
  ✓ Find the best alignment of two sequences
  ✓ Find the best match (alignment) of a given sequence in a large dataset of sequences
    – Find the best alignment of multiple sequences

• Motif and gene finding

• Relationship between sequences
  – Phylogeny

• Clustering and classification

• Many many many more ...
One of many commonly used tools that depend on sequence alignment.
Sequence Alignment

<table>
<thead>
<tr>
<th>G</th>
<th>A</th>
<th>A</th>
<th>T</th>
<th>T</th>
<th>C</th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>G</td>
<td>A</td>
<td></td>
<td>T</td>
<td>C</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| G | G | A | T | C | G | | | | | A |
Mission:
Find the best alignment between two sequences.
Mission: Find the best alignment between two sequences.

Find the best alignment of GAATC and CATAC:

GAATC | GAAT-C | -GAAT-C | GAAT-C  
CATA | C-ATAC | C-A-TAC | C-ATAC  
GAATC | GAAT-C | GA-ATC | GAAT-C  
CA-TAC | CA-TAC | CATA-C | CA-TAC  

(some of a very large number of possibilities)
Mission: Find the best alignment between two sequences. This is an optimization problem!

What do we need to solve this problem?
Mission: Find the best alignment between two sequences.

- A “search” algorithm for finding the alignment with the best score
- A method for scoring alignments
Scoring Principles

- Score each locus independently.
- The alignment score will be the sum of the scores in all loci.
- Perfect Matches will get a positive (good) score.
- What about mismatches?

GAATC
CATAC
Scoring Principles

• Score each locus independently.
• The alignment score will be the sum of the scores in all loci.
• Perfect Matches will get a positive (good) score.
• What about mismatches?

GAATC
CATAC

(transitions are typically about 2x as frequent as transversions in real sequences)
Scoring Aligned Bases

• A reasonable substitution matrix:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
</tr>
<tr>
<td>T</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
<td>10</td>
</tr>
</tbody>
</table>

What about gaps?

GAATC
CATAC

-5 + 10 + -5 + -5 + 10 = 5
What About Gaps?

- A reasonable substitution matrix:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
</tr>
<tr>
<td>T</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
<td>10</td>
</tr>
</tbody>
</table>

What do gaps mean?

What if gaps have no penalty?

What if gaps have no penalty?

\[ \text{GAAT} - \text{C} \]

\[ \text{CA} - \text{TAC} \]

\[-5 + 10 + ? + 10 + ? + 10 = ?\]
Scoring Gaps?

- **Linear** gap penalty: every gap receives a score of $d$:

\[
\begin{align*}
GAAT & - C \\
CA & - TAC
\end{align*}
\]

\[d = -4\]

\[-5 + 10 + -4 + 10 + -4 + 10 = 17\]
Scoring Gaps?

- **Linear** gap penalty: every gap receives a score of $d$:
  
  $\text{GAAT} - \text{C} \quad d=-4$
  
  $\text{CA} - \text{TAC}$
  
  $-5 + 10 + -4 + 10 + -4 + 10 = 17$

- **Affine** gap penalty: opening a gap receives a score of $d$; extending a gap receives a score of $e$:

  $\text{G} -- \text{AATC} \quad d=-4$
  
  $\text{CATA} -- \text{C} \quad e=-1$
  
  $-5 + -4 + -1 + 10 + -4 + -1 + 10 = 5$
Same Method Applies to AA

### BLOSUM62 Score Matrix

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

**regular 20 amino acids**

**ambiguity codes and stop**

Y mutates to V receives -1
M mutates to L receives 2
E gets deleted receives -10
G gets deleted receives -10
D matches D receives 6

**Total score = -13**
Mission: find the best alignment between two sequences.

- A “search” algorithm for finding the alignment with the best score
- A method for scoring alignments

**Substitution matrix:**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
</tr>
<tr>
<td>T</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
<td>10</td>
</tr>
</tbody>
</table>

**Gap penalty:**
- Linear gap penalty
- Affine gap penalty

- GAAT-C
- CA-TAC

\[ d = -4 \]

\[ -5 + 10 + -4 + 10 + -4 + 10 = 17 \]
Exhaustive search

- **Align the two sequences:** GAATC and CATAC

**Simple (exhaustive search) algorithm**

1) **Construct all possible alignments**

2) **Use the substitution matrix and gap penalty to score each alignment**

3) **Pick the alignment with the best score**
How many possibilities?

- **Align the two sequences: GAATC and CATAC**

```
GAATC
CATAC
-
GAAT-C
C-ATAC

GAATC-
CA-TAC

GAAT-C
C-ATAC

GAAT-C
C-ATAC

GAAT-C
C-ATAC

GAATC-
CA-TAC

GAAT-C
CA-TAC

GA-ATC
CATA-C

GAAT-C
CA-TAC
```

- How many different possible alignments of two sequences of length $n$ exist?
How many possibilities?

• Align the two sequences: GAATC and CATA

<table>
<thead>
<tr>
<th>Alignment</th>
<th>GAATC</th>
<th>GAAT-C</th>
<th>-GAAT-C</th>
<th>GAAT-C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATA</td>
<td>C-ATAC</td>
<td>C-A-TAC</td>
<td>C-ATAC</td>
<td>C-ATAC</td>
</tr>
<tr>
<td>GAATC-CATA</td>
<td>GAAT-C</td>
<td>GA-ATC</td>
<td>GAAT-C</td>
<td></td>
</tr>
<tr>
<td>GAATC-CA-TAC</td>
<td>GAAT-C</td>
<td>GAAT-C</td>
<td>CATA-C</td>
<td></td>
</tr>
<tr>
<td>GAAT-C-CA-TAC</td>
<td>GAAT-C</td>
<td>GAAT-C</td>
<td>CA-TAC</td>
<td></td>
</tr>
</tbody>
</table>

• How many different possible alignments of two sequences of length \( n \) exist?

| \( n \) | \( 2.5 \times 10^2 \) | \( 1.8 \times 10^5 \) | \( 1.4 \times 10^{11} \) | \( 1.2 \times 10^{17} \) | \( 1.1 \times 10^{23} \) |
Mission: Find the best alignment between two sequences.

- A “search” algorithm for finding the alignment with the best score
- Needleman–Wunsch Algorithm
- Dynamic programming

A method for scoring alignments

Substitution matrix:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
</tr>
<tr>
<td>T</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
<td>10</td>
</tr>
</tbody>
</table>

Gap penalty:
- Linear gap penalty
- Affine gap penalty

Alignment:
- GAAT–C
- CA–TAC

Distance, $d = -4$

Scoring:
-5 + 10 + -4 + 10 + -4 + 10 = 17
The Needleman–Wunsch Algorithm

- An algorithm for **global alignment** on two sequences

- A **Dynamic Programming (DP)** approach
  - Yes, it’s a weird name.
  - DP is closely related to recursion and to mathematical induction

- We can prove that the resulting score is optimal.
### DP matrix

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>G</td>
<td>A</td>
<td>A</td>
<td>T</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Substitution matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>-5</td>
<td>10</td>
<td>-5</td>
</tr>
<tr>
<td>T</td>
<td>-5</td>
<td>0</td>
<td>-5</td>
<td>10</td>
</tr>
</tbody>
</table>

Gap penalty: $d = -4$