3 Extension of global alignment

3.1 Homology at the sequence level

Constructing alignments can be useful to understand homology among different species. Finding homologies is important to reveal a common evolutionary ancestor.

Evolution and homology

All species are derived from a common ancestor at some point during the course of evolution.

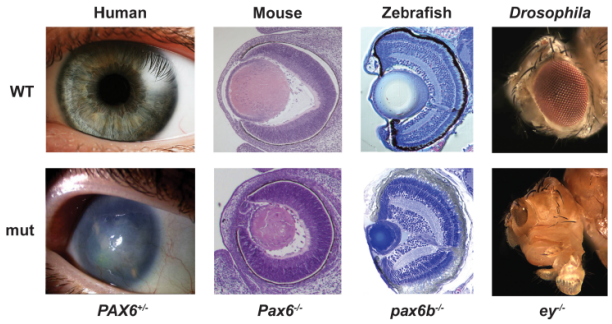

Figure 3.1: PAX6 alterations result in similar changes to eye morphology
(source: Washington et al, doi: 10.1371/journal.pbio.1000247 via Wikimedia Commons)

Homologous and analogous

It is useful to check similarity at the molecular level because there are cases that analogous structures may not indicate homologous.

Figure 3.2: Homologous and analogous structures
(source: John Romanes, 1892, Darwin and after Darwin via Wikimedia Commons)

Sequence homology

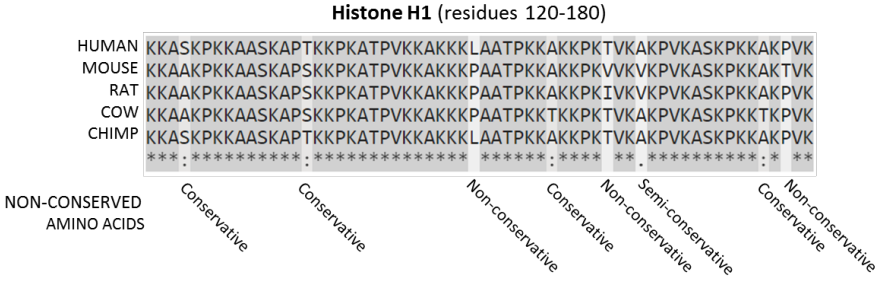

Figure 3.3: Multiple sequence alignment of histone sequences
(source: Shafee, Wikimedia Commons)

Evolution at the sequence level

Sequence differences in DNA

- Substitution (a mismatch in alignment)
- Insertion (a gap in alignment)
- Deletion (a gap in alignment)
- Inversion

Sources of variations

- Mutation
- Recombination
- Insertional mutagenesis
- ...

A mutation of the third nucleotide in a codon often does not affect which amino acid is synthesized.

- GCU → Ala (Alanine)
- GCC → Ala (Alanine)
- GCA → Ala (Alanine)
- GCG → Ala (Alanine)

An amino acid can be replaced by a different amino acid that has similar properties in some cases.

- AUU, AUC, AUA → Ile (Isoleucine)
- CUU, CUC, CUA → Leu (Leucine)

Extension of global alignment with DP

- Score matrix
  DNA, RNA, and protein
- Gap penalty
  Linear, affine, and constant

3.2 Introduction of score matrix

We will expand our simple scoring scheme to score matrices. This expansion allows us to solve general alignment problems with DNA, RNA, and protein sequences.

Extension of a scoring scheme to a score matrix

The matrix below is equivalent with match: 1 and mismatch: 0.

	a	b
a	1	0
b	0	1

Example of a DNA score matrix

The matrix below is equivalent with match: 5 and mismatch: -4.

	A	T	G	C
A	5	-4	-4	-4
T		5	-4	-4
G			5	-4
C				5

Applications of score matrix

Score matrices are more flexible than the simple scoring scheme. For instance, they can be used for the following cases.

- DNA pairs
- RNA pairs
- Similarity of protein sequences by amino acid properties

DNA pairs (Watson-Crick pairs)

A thymine pairs with an adenine, and a cytosine pairs with a guanine.

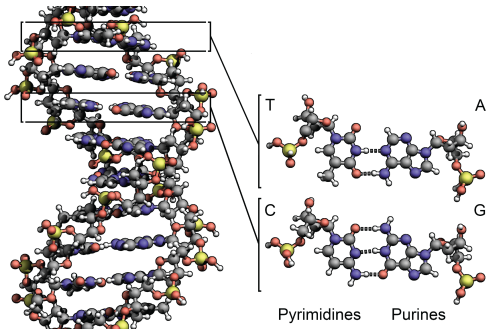

Figure 3.4: Watson-Crick pairs (source: Zephyris, Wikimedia Commons)

Example of score matrix for DNA pairs

The matrix reflects the differences of hydrogen bonds.

	A	T	G	C
A	-3	4	-3	-3
T		-3	-3	-3
G			-3	5
C				-3

Example of DP for DNA pairs

You can use DP to find a DNA alignment with Watson-Crick pairs. For instance, the DP table below is used to solve the optimal alignment for two DNA sequences: q = AC and d = GT with gap penalty g = 4.

DP table:

q/d		G	T
	0	-4	-8
A	-4	-3	0
C	-8	1	-3

Alignment:

```
q: AC-
d: -GT
```

RNA pairs

A single stand of RNA can form a 3D structure that has a biological function. The secondary structure of RNA is a two-dimensional representation of the structure.

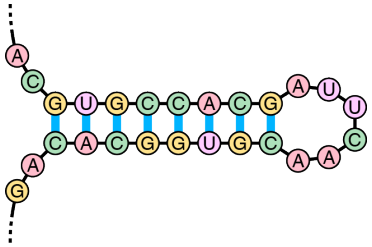

Figure 3.5: RNA stem-loop (source: Sakurambo, Wikimedia Commons)

Wobble pairs

Wobble pairs are not canonical Watson-Crick pairs, but they can still form hydrogen bonds.

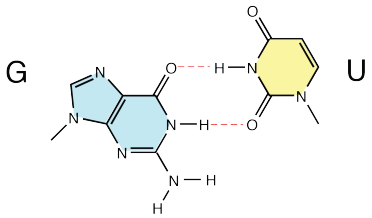

Figure 3.6: GU wobble pairs
(modified from the original version by Fdardel, Wikimedia Commons)

Example of score matrix for RNA pairs

The matrix takes GU wobbles into consideration.

	A	U	G	C
A	-3	5	-3	-3
U		-3	2	-3
G			-3	5
C				-3

Example of DP for RNA pairs

You can form the following DP table for two RNA sequences: q = AU and d = UGA with gap penalty g = 9.

DP table:

q/d		U	G	A
	0	-9	-18	-27
A	-9	5	-4	-11
U	-18	-4	7	1

Alignment:

```
q: A-U
d: UGA
```

Similarity of protein sequences

Amino acids can be categorized into several groups by their properties. Proteins alignments often need to take these properties into consideration.

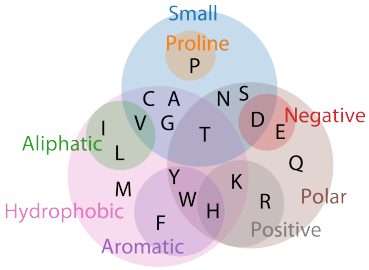

Figure 3.7: Venn diagram of amino acid properties

Example of a protein score matrix

It can be used to compare the similarity between two protein sequences.

	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**
A	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
R	3	17	4	3	2	5	3	2	6	3	2	9	4	1	4	4	3	7	2	2
N	4	4	6	7	2	5	6	4	6	3	2	5	3	2	4	5	4	2	3	3
D	5	4	8	11	1	7	10	5	6	3	2	5	3	1	4	5	5	1	2	3
C	2	1	1	1	52	1	1	2	2	2	1	1	1	1	2	3	2	1	4	2
Q	3	5	5	6	1	10	7	3	7	2	3	5	3	1	4	3	3	1	2	3
E	5	4	7	11	1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
G	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
H	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
I	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
L	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
K	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
M	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
F	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
P	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
S	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
T	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
W	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
Y	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
V	7	4	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	72	4	17

Table 3.1: Mutation probability matrix for the evolutionary distance of 250 PAMs (in percentage) (Chapter 22: A model of evolutionary change in proteins, Dayhoff and Schwartz, Atlas of Protein Sequence and Structure, 1978)

Exercise 3.1

1. Use the DNA score matrix below with g = 10 and find the optimal alignment for q = TG and d = TCG.

	A	T	G	C
A	5	-4	-4	-4
T		5	-4	-4
G			5	-4
C				5

2. The 250 PAM mutation matrix above can not directly be used for global alignments. Explain what kind of matrix you need for calculating alignment scores.

3.3 Extension of gap penalties

Types of gap penalties

Three types of gap penalties can be considered when creating an alignment. They treat a gap penalty differently depending on the gap length.

- Linear
- Affine
- Constant

Gap penalty notation

- g: single gap penalty
- l: length of a gap
- g_l: gap penalty of length l
- g_{open}: initial gap penalty
- g_{extend}: extended gap penalty

Linear gap penalty

It is the same as our simple scoring scheme. It treats a gap with multiple blanks as a result of several mutations. A gap of length l can be calculated as: $g_l = g \times l$.

Example of a gap of length 2

```
q: ACCCGT
d: AC--GT
```

The score of the gap (only the gap part) is 10 when $g = 5$.

Affine gap penalty

It treats a gap with multiple blanks as a result of a single mutation. A gap with length l can be calculated as: $g_l = g_{open} + (l-1) \times g_{extend}$.

Example of a gap of length 2

```
q: ACCCGT
d: AC--GT
```

The score of the gap (only the gap part) is 5.5 when g_{open} and g_{extend} are 5 and 0.5 respectively.

Constant gap penalty

It is similar to the affine gap penalty, but the score is independent form the gap length. A gap with length l can be calculated as: $g_l = g$

Example of a gap of length 2

```
q: ACCCGT
d: AC--GT
```

The score of the gap (only the gap part) for the alignment above is 5 when $g = 5$.

Exercise 3.2

Calculate all three types of gap penalties for the gap in alignment 1 & 2.

- g: 5
- g_{open}: 5
- g_{extend}: 0.5

```
Alignment 1
    q: CCCGG
    d: CC-CG
```

```
Alignment 2
    q: CCCGG
    d: C---G
```

3.4 Affine gap penalties with a single DP table

DP for general gap penalty

We need to modify DP so that extra cells are checked to find the optimal score of a cell.

Cell update rule of general gap penalty

$$H_{i,j} = \max \left[H_{i-1,j-1} + R_{q_i d_j}, \quad \max_{1 \le l \le j} (H_{i,j-l} - g_l), \quad \max_{1 \le l \le i} (H_{i-l,j} - g_l) \right]$$

Example of cell update

Sequences:

```
q: AG, d: ACG
```

Scoring scheme:

$g_{open} = 1$
$g_{extend} = 0.1$
$R_{ab} = 1$ for a = b
$R_{ab} = 0$ for a $\neq$ b

Update $H_{2,1}$

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	1			
T	-1.1	0			

- vertical: $max(1-1, \quad -1-1-0.1) = 0$
- horizontal: $-1.1-1 = -2.1$
- diagonal: $-1-0 = -1$

Update $H_{1,2}$

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	1	0		
T	-1.1	0			

- vertical: $-1.1-1 = -2.1$
- horizontal: $max(1-1, \quad -1-1-0.1) = 0$
- diagonal: $-1-0 = -1$

Update $H_{1,3}$

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	1	0	**-0.1**	
T	-1.1	0	1		

- vertical: $-1.2-1 = -2.2$
- horizontal: $max(0-1, \quad 1-1-0.1, \quad -1-1-0.1-0.1) = -0.1$
- diagonal: $-1.1-0 = -1.1$

Exercise 3.3

Complete the DP table below.

Sequences:

`q: AT, d: ACTT`

Scoring scheme:

$g_{open} = 1$
$g_{extend} = 0.1$
$R_{ab} = 1$ for a = b
$R_{ab} = 0$ for a $\neq$ b

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	1	0	-0.1	
T	-1.1	0	1		

3.5 Affine gap penalties with three DP tables

DP can effectively solve affine gap penalties with three tables.

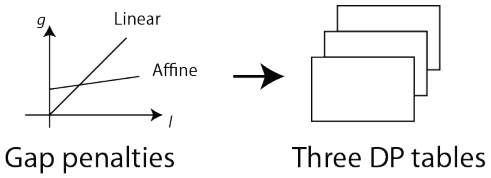

Figure 3.8: Affine gap penalties and three tables

Three DP tables

We need to modify DP so that extra cells are checked to find the optimal score of a cell.

- $E_{i,j}$: alignment ending with a gap extend (vertical)
- $F_{i,j}$: alignment ending with a gap extend (horizontal)
- $G_{i,j}$: alignment ending with a match/mismatch (diagonal)

Cell update rule of the three tables

$$E_{i,j} = max(E_{i-1,j} - g_{extend}, \quad F_{i-1,j} - g_{open}, \quad G_{i-1,j} - g_{open})$$
$$F_{i,j} = max(E_{i,j-1} - g_{open}, \quad F_{i,j-1} - g_{extend}, \quad G_{i,j-1} - g_{open})$$
$$G_{i,j} = max(E_{i-1,j-1} + R_{q_i d_j}, \quad F_{i-1,j-1} + R_{q_i d_j}, \quad G_{i-1,j-1} + R_{q_i d_j})$$

You can calculate H only in the last cell.

$$H_{m,n} = max(E_{m,n}, F_{m,n}, G_{m,n})$$

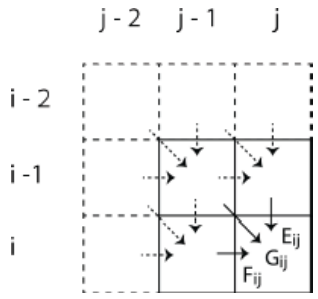

Figure 3.9: Update a cell with E, F, and G

Recurrence rules when i = 0 and j = 0

	$i > 1, j > 1$	$i = 1$	$j = 1$
$E_{i,j}$	$max\begin{cases} E_{i-1,j} - g_{extend} \\ F_{i-1,j} - g_{open} \\ G_{i-1,j} - g_{open} \end{cases}$	$max\begin{cases} E_{i-1,j} - g_{open} \\ F_{i-1,j} - g_{open} \\ G_{i-1,j} - g_{open} \end{cases}$	$max\begin{cases} E_{i-1,j} - g_{extend} \\ F_{i-1,j} - g_{open} \\ G_{i-1,j} - g_{open} \end{cases}$
$F_{i,j}$	$max\begin{cases} E_{i,j-1} - g_{open} \\ F_{i,j-1} - g_{extend} \\ G_{i,j-1} - g_{open} \end{cases}$	$max\begin{cases} E_{i,j-1} - g_{open} \\ F_{i,j-1} - g_{extend} \\ G_{i,j-1} - g_{open} \end{cases}$	$max\begin{cases} E_{i,j-1} - g_{open} \\ F_{i,j-1} - g_{open} \\ G_{i,j-1} - g_{open} \end{cases}$
$G_{i,j}$	$max\begin{cases} E_{i-1,j-1} + R_{q_i d_j} \\ F_{i-1,j-1} + R_{q_i d_j} \\ G_{i-1,j-1} + R_{q_i d_j} \end{cases}$	$max\begin{cases} E_{i-1,j-1} + R_{q_i d_j} \\ F_{i-1,j-1} + R_{q_i d_j} \\ G_{i-1,j-1} + R_{q_i d_j} \end{cases}$	$max\begin{cases} E_{i-1,j-1} + R_{q_i d_j} \\ F_{i-1,j-1} + R_{q_i d_j} \\ G_{i-1,j-1} + R_{q_i d_j} \end{cases}$

Example of updating DP tables with affine gaps

Sequences:

`q: AT, d: ACTT`

Scoring scheme:

$g_{open} = 1$
$g_{extend} = 0.1$
$R_{ab} = 1$ for a = b
$R_{ab} = 0$ for a ≠ b

Initialization

E

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1				
T	-1.1				

F

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1				
T	-1.1				

G

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1				
T	-1.1				

Update the first row

E

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	-2	-2.1	-2.2	-2.3
T	-1.1				

F

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	-2	0	-0.1	-0.2
T	-1.1				

G

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	1	-1	-1.1	-1.2
T	-1.1				

Update the second row

E

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	-2	-2.1	-2.2	-2.3
T	-1.1	0	-1	-1.1	-1.2

F

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	-2	0	-0.1	-0.2
T	-1.1	-2.1	-1	0	0

G

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	1	-1	-1.1	-1.2
T	-1.1	-1	1	1	0.9

H

		A	C	T	T
A					
T					0.9

Update H

E

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	-2	-2.1	-2.2	-2.3
T	-1.1	0	-1	-1.1	-1.2

F

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	-2	0	-0.1	-0.2
T	-1.1	-2.1	-1	0	0

G

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	1	-1	-1.1	-1.2
T	-1.1	-1	1	1	0.9

H

		A	C	T	T
A					
T					0.9

Backtrack

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	-2	-2.1	-2.2	-2.3
T	-1.1	0	-1	-1.1	-1.2

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	-2	0	-0.1	-0.2
T	-1.1	-2.1	-1	0	0

G

		A	C	T	T
	0	-1	-1.1	-1.2	-1.3
A	-1	1	-1	-1.1	-1.2
T	-1.1	-1	1	1	0.9

H

		A	C	T	T
A					
T					0.9

Optimal alignment

```
q: A--T    Score: 0.9
d: ACTT
```

Constant gap penalty

DP with constant gap penalty can be solved in the same way as the affine gap penalty.

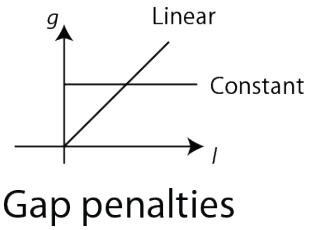

Figure 3.10: Constant gap penalty

3.6 Sequence distance

Distances can be also used to indicate the similarity of an alignment.

Edit distance

The Levenshtein distance is one of the most commonly used edit distances in computer science.

$$\begin{aligned} \text{Insertion}: \quad & AC \to AGC && (\varepsilon \to G) \\ \text{Deletion}: \quad & ATC \to AC && (T \to \varepsilon) \\ \text{Substitution}: \quad & AAA \to ATA && (A \to T) \end{aligned}$$

Scoring scheme for Levenshtein distance

- $R_{ab} = 0$ for a = b
- R_{ab} = -1 for a $\neq$ b
- g = 1

Distance from DP score

Given the best score T from DP, the edit distance d is $-T$.

Example of edit distance with DP

		A	T	C
	0	-1	-2	-3
A	-1	0	-1	-2
C	-2	-1	-1	-1

$T = -1$

$d = 1$

Metric space

The edit distance constitutes a metric space.

- $d_{xy} = 0$ for $x = y$
- $d_{xy} > 0$ for $x \neq y$
- $d_{xy} = d_{yx}$
- $d_{xy} \leq d_{xz} + d_{zy}$ for any z (the triangle inequality)

Mutation and distance

Mutations may occur several times on the same position.

Example of single mutations

$ACGT \rightarrow AGT \rightarrow ACT \rightarrow AGT \rightarrow AGCT$

Four mutations have occurred, but the edit distance is 2.

Distance per column

It indicates the number of mutations per column (nucleotide/amino acid).

$D = d/$(length of the longest sequence)

Correction of distance

The distance can be adjusted. Below is a simple correction approach for protein sequences.

$$K = -\ln\left(1 - D - 1/5D^2\right)$$

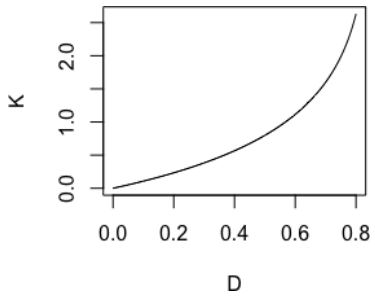

Figure 3.11: Correction of distance D

Example of distance correction

$D = 0.5$
$K = -\ln\left(1 - 0.5 - 1/5 \times 0.25\right) = -\ln(0.45) \approx 0.8$