Aligning Sequences

3° For the sake of simplicity, this discussion will deal only with DNA sequences; it is important to note that bioinformaticians use RNA sequences and protein sequences perhaps more often than DNA sequences. Let us define what we mean by a DNA sequence.

> **Definition (DNA sequence).** A DNA sequences is a finite string of symbols from the alphabet $\{A, T, C, G\}$. Each string represents a molecular chain of nucleotides.

The value of aligning sequences comes from the ability to tell how related, molecularly, two strands of DNA are. In turn, this relationships tells us how evolutionary related two species are. This is immensely important as it is a mathematically and molecularly sound method of describing the evolutionary relationship between species. Previously, speciation was determined by macroscopic physical traits (such a presence of a tail)–a method prone to error.

4° There are many ways to align two sequences and the choice of method depends on what we wish to find: do we want the longest common subsequence, or do we want the alignment that gives the best overall fit? The following algorithm, when given two sequences, will return the two aligned sequences that give the longest common subsequence.

Algorithm 1 (Longest Common Subsequence). We wish to produce an algorithm that, given two sequences from the alphabet

$$\aleph = \{\epsilon, A, T, G, C\},$$

produces the longest common subsequence. Let v be a sequence of length m and let w be a sequence of length n. Let Σ and Φ be $(m+1) \times (n+1)$ matrices. Let the elements of the first row and column of Σ have value 0, and let the elements of the first row and column of Φ have value $\varnothing$. Let $v(i)$ refer to the i^{th} element of v, and let $w(j)$ refer the j^{th} element of w where $v(0)$ and $v(0)$ are ϵ.

1♭ We begin at $\Sigma(1,1)$ and $\Phi(1,1)$. If $v(1) = w(1)$, then

$$\Sigma(1,1) = \Sigma(0,0) + 1 = 1.$$

Otherwise, if $v(1) \neq w(1)$, then

$$\Sigma(1,1) = max \begin{cases} \Sigma(0,0) \\ \Sigma(1,0) \\ \Sigma(0,1) \end{cases} = 0.$$

Similarly, if $v(1) = w(1)$,

$$\Phi(1,1) = \nwarrow .$$

Otherwise, if $v(1) \neq w(1)$, then

$$\Phi(1,1) = \uparrow .$$

2♭ We proceed row by row. If $v(i) = w(j)$, then $\Sigma(i,j) = \Sigma(i-1, j-1) + 1$. Otherwise, if $v(i) \neq w(j)$, then

$$\Sigma(i,j) = max \begin{cases} \Sigma(i-1, j-1) \\ \Sigma(i, j-1) \\ \Sigma(i-1, j). \end{cases}$$

The cases for Φ are more complex.

A♭ If $v(i) = w(j)$, then $\Phi(i, j) = \nwarrow$.

B♭ If $v(i) \neq w(j)$ and $\Phi(i, j-1) = \nwarrow$, then $\Phi(i, j) = \leftarrow$.

C♭ If $v(i) \neq w(j)$, $\Phi(i, j-1) \neq \nwarrow$, and $\Sigma(i, j) = \Sigma(i, j-1) > \Sigma(i-1, j)$, then $\Phi(i, j) = \leftarrow$.

D♭ If $v(i) \neq w(j)$, $\Phi(i, j-1) \neq \nwarrow$, and $\Sigma(i, j) = \Sigma(i-1, j) > \Sigma(i, j-1)$, then $\Phi(i, j) = \uparrow$.

3♭ Examine Φ. Beginning at the southeastern-most element, we follow the path the arrows form until we reach $\varnothing$. Removing all arrows that are not on this path, we create the matrix Π.

4♭ Let σ be our path represented as a sequence with elements from the alphabet $\{\varnothing, \nwarrow, \uparrow, \leftarrow\}$. To create σ, we take the elements from *northwest* to *southeast.* We wish to derive from σ our aligned sequences, call them α and β corresponding to v and w, respectively. Let $\sigma(x)$ correspond to the x^{th} element of σ, and similarly y for both α and β. Let (i, j) correspond to the position of each arrow in the path in Π. Require that $\sigma(0) = \varnothing$.

A♭ If $\Pi(0, 0) = \varnothing$, then do nothing. If $\Pi(1, 0) = \varnothing$, then $\alpha(y) = v(1)$ and $\beta(y) =$ ‘-’. If $\Pi(0, 1) = \varnothing$, then $\alpha(y) =$ ’-’ and $\beta(y) = w(1)$.

B♭ We proceed through our path arrow by arrow. If $\sigma(x) = \nwarrow$, then $\alpha(y) = v(i)$ and $\beta(y) = w(j)$.

C♭ If $\sigma(x) = \uparrow$, then $\alpha(y) = v(i)$ and $\beta(y) =$ ’-’.

D♭ If $\sigma(y) = \leftarrow$, then $\alpha(y) =$ ’-’ and $\beta(y) = w(j)$.

5° *An example.*

Let $v = \epsilon\texttt{ATAGGATC}$ and $w = \epsilon\texttt{ATCCGCT}$. Steps 1 and 2 produce Σ and Φ:

$$\Sigma = \begin{array}{c} \\ \epsilon \\ T \\ A \\ G \\ G \\ A \\ T \\ C \end{array} \begin{array}{c} \begin{array}{cccccccc} \epsilon & A & T & C & C & G & C & T \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 2 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\ 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 3 \\ 0 & 1 & 2 & 3 & 4 & 4 & 5 & 5 \end{pmatrix} \end{array},$$

$$\Phi = \begin{array}{c|cccccccc|} & \epsilon & A & T & C & C & G & C & T \\ \epsilon & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ T & \varnothing & \uparrow & \nwarrow & \leftarrow & \leftarrow & \leftarrow & \leftarrow & \nwarrow \\ A & \varnothing & \nwarrow & \leftarrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ G & \varnothing & \uparrow & \uparrow & \uparrow & \uparrow & \nwarrow & \leftarrow & \uparrow \\ G & \varnothing & \uparrow & \uparrow & \uparrow & \uparrow & \nwarrow & \leftarrow & \uparrow \\ A & \varnothing & \nwarrow & \leftarrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ T & \varnothing & \uparrow & \nwarrow & \leftarrow & \leftarrow & \uparrow & \uparrow & \nwarrow \\ C & \varnothing & \uparrow & \uparrow & \nwarrow & \nwarrow & \leftarrow & \nwarrow & \leftarrow \end{array}.$$

Step 3 produces Π:

$$\Pi = \begin{array}{c|cccccccc|} & \epsilon & A & T & C & C & G & C & T \\ \epsilon & & \varnothing & & & & & & \\ T & & & \nwarrow & \leftarrow & \leftarrow & & & \\ A & & & & & \uparrow & & & \\ G & & & & & \uparrow & & & \\ G & & & & & & \nwarrow & & \\ A & & & & & & \uparrow & & \\ T & & & & & & \uparrow & & \\ C & & & & & & & \nwarrow & \leftarrow \end{array}.$$

Step 4 produces σ. In this case, σ is as follows:

$$\sigma = \{\varnothing, \nwarrow, \leftarrow, \leftarrow, \uparrow, \uparrow, \nwarrow, \uparrow, \uparrow, \nwarrow, \leftarrow\}.$$

Then α and β are as follows:

```
- T - - A G G A T C -
A T G C - - G - - C T
```

Using pseudocode notation, the algorithm is more approachable:

Algorithm 2: Least Common Subsequence

```
for $i := 0$ to $n$
    $s_{i,0} = 0$
for $j := 1$ to $m$
    $s_{0,j} = 0$
for $i := 1$ to $n$
    for $j = 1$ to $m$
        $s_{i,j} = max \begin{cases} s_{i-1,j} \\ s_{i,j-1} \\ s_{i-1,j-1} + 1 \text{ if } v_i = w_j \end{cases}$
        $b_{i,j} = max \begin{cases} \uparrow \text{ if } s_{i,j} = s_{i-1,j} \\ \leftarrow \text{ if } s_{i,j} = s_{i,j-1} \\ \nwarrow \text{ if } s_{i,j} = s_{i-1,j-1} + 1 \end{cases}$
return $(s_{n,m}, b)$
```

Scoring Alignments

6° Why do we align sequences? We do so because, paired with an appropriate *scoring schema*, we can garner important information about the DNA, RNA, or protein chains being examined.

> **Definition (scoring schema).** A scoring schema assigns a value to a particular alignment based on how different two sequences are. Different scoring schemas reveal different characteristics of an alignment.

For example, let a simple scoring schema be as follows: for every matching character, add 1 to the score; for every '–', subtract $\frac{1}{2}$. Then for the alignment from 5°, we have

$$\text{score} = 1 \cdot (5) - \frac{1}{2} \cdot (4) = 3.$$

Note that in this schema, a higher score means more perfect matches and is hence more desirable. In other scoring schemas this may not be the case.

7° Of course, when aligning sequences in practice, one would use a scoring schema based on biological research that would be informative of true evolutionary relationships. The following excerpt describes a practical scoring schema for protein sequences:

> [The] best scoring matrices to compare two proteins depends on how similar [two] organisms are. Biologists get around this problem by first analyzing extremely similar proteins, for example, proteins that have, on average, only one mutation per 100 amino acids. Many proteins in human and chimpanzee fulfill this requirement. Such sequences are defined as being *one PAM unit diverged* and to a first approximation one can think of a PAM unit as the amount of time in which an "average" protein mutates 1% of its amino acids. The *PAM 1* scoring matrix is defined from many alignments of extremely similar proteins as follows.
>
> Given a set of base alignments, define $f(i,j)$ as the total number of times amino acids i and j are aligned against each other, divided by the total number of aligned positions. We also define $g(i,j)$ as $\frac{f(i,j)}{f(i)}$, where $f(i)$ is the frequency of amino acids i mutates into amino acid j with 1 PAM unit. The (i,j) entry of the *PAM 1* matrix is defined as $\delta(i,j) = \log \frac{f(i,j)}{f(i) \cdot f(j)} = \log \frac{g(i,j)}{f(j)}$ $(f(i) \cdot f(j)$ stands for the frequency of aligning amino acid i against amino acid j that one expects simply by chance). The *PAM n* matrix can be defined as the result of applying the PAM 1 matrix n times. If g is the 20×20 matrix of frequencies $g(i,j)$, then g^n (multiplying the matrix by itself n times) gives the probability that amino acid i mutates into amino acid j during n PAM units. The (i,j) entry of the PAM n matrix is defined as $\log \frac{g^n_{i,j}}{f(j)}$. (Pevzner, 179)

Obviously, when determining an appropriate scoring schema, one must account for biological factors that are directly relevant to evolution, such as rate of mutation.