ALIGNING SEQUENCES

THE AUTHOR

1. Aligning two sequences

We will refer to any member of the set $\{A, C, G, T, -\}$ as a character and to any subset an alphabet. Throughout this paper σ denotes a sequence over the alphabet $\{A, C, G, T\}$. Sequences can be expanded by inserting the character at $-$ anywhere in the sequence. This includes inserting $-$ at the very beginning and at the very end. Expanded sequences are denoted by μ.

As an example, given the sequence $ACCTGAC$, the following are expanded sequences by inserting the character $-$: $A-CCTGAC$, and $-ACCTGAC$.

An alignment of two sequences σ^1 and σ^2 is to expand them by inclusion of $-$ so that the expanded sequences are of same length and are in 1-1 correspondence. We will momentarily precisely define this notion, but first let us look at an example. Let $\sigma^1 = ACTGT$ and $\sigma^2 = CACCCG$ be two sequences. Let $\mu^1 = -ACT-GT$, and $\mu^2 = CACCCG-$ be the corresponding expanded sequences. Now μ^1 and μ^2 can be put in 1-1 correspondence:

$$\begin{array}{ccccccc} - & A & C & T & - & G & T \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ C & A & C & C & C & G & - \end{array},$$

hence, we have aligned the two sequences.

Definition 1.1. *A **substitution**, denoted by S, is an assignment of a character from the set $\{A, C, G, T\}$ to a character of the set $\{A, C, G, T\}$, i.e., it is an ordered pair in $\{A, C, G, T\} \times \{A, C, G, T\}$.*

Thus, for example, in the previous correspondence $\mu^1 \leftrightarrow \mu^2$, the second, third, fourth, and sixth assignments (read from left to right) are **substitutions,** the fourth substitution is a *mismatch*; others are identities.

Definition 1.2. *An insertion, denoted by I, is an assignment $- \to \{A, C, G, T\}$, i.e, an ordered pair in $\{-\} \times \{A, C, G, T\}$.*

Thus, for example, in $\mu^1 \leftrightarrow \mu^2$, the first and fifth assignments are **insertions.**

Definition 1.3. *A deletion, denoted by D, is an assignment of a character from $\{A, C, G, T\}$ to $-$, i.e, an ordered pair in $\{A, C, G, T\} \times \{-\}$.*

Thus, for example, the seventh assignment in $\mu^1 \leftrightarrow \mu^2$, is a **deletion.**

Definition 1.4. *An **edit string** is a finite sequence of operations S, D, I.*

Each edit string induce two expansions of a given sequence. One, literally an insertion operation, is to insert the character $-$ where an I is found in the string. In order to execute all the edits and edit every member of the original sequence, the length of the sequence must be the sum of the number of S and the number of D in the edit sequence.

The other is a deletion operation formed by inserting $-$ where a D is found (It is rather ironic that *deletion operation* is formed to expand a sequence, but note the name was used simply because D are used in someway to expand the sequence). In this case we tacitly assume that the sum of number of S and number of I in the edit string is length of the original sequence.

Now we are ready to define an alignment precisely.

Definition 1.5. *An alignment of an ordered pair of sequences (σ^1, σ^2) is an edit string that induces an insertion expansion μ^1 on σ^1 and a deletion expansion μ^2 on σ^2 such that μ^1 is in one-to-one correspondence with μ^2.*

Let n and m be the lengths of σ^1 and σ^2. From the discussion preceding the definition, it is clear that an edit string is an alignment iff

$$\#S + \#D = n, \quad \#S + \#I = m.$$

Example 1.1. *Let $\sigma^1 = ACGTAGC$ and $\sigma^2 = ACCGAGACC$. Then the edit string $SSISISSIDS$ is an alignment of σ^1 and σ^2 because*

$$\begin{array}{rcccccccccc} & S & S & I & S & I & S & S & I & D & S \\ \mu^1 = & A & C & - & G & - & T & A & - & G & C \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \mu^2 = & A & C & C & G & A & G & A & C & - & C \end{array}$$

2. Alignment graph

Alignment of sequences can be made into a problem of determining paths in directed graphs.

Definition 2.1. *The* alignment graph *$\mathcal{G}_{n,m}$ is the directed graph on the set of nodes $\{0, 1, \ldots, n\} \times \{0, 1, \ldots, m\}$ and three classes of directed edges horizontal, vertical, and diagonal defined below:*

Horizontal: *These directed edges* $[(i, j) \to (i, j+1)], \quad 0 \le i \le n,\ 0 \le j \le m-1.$

Vertical: *These are the edges* $[(i, j) \to (i+1, j)] \quad 0 \le i \le n-1,\ 0 \le j \le m.$

Diagonal: *These are the edges* $[(i,\,j) \to (i+1, j+1)], \quad 0 \le i \le n-1,\ 0 \le j \le m-1$.

2.1. **Identification of sequences on the graph.** To align two sequences σ^1 and σ^2 of length n and m, the left most edges of the zeroth column of $\mathcal{G}_{n,\,m}$ are identified with the terms of the sequence σ^1, i.e., $\sigma^1_1 \leftrightarrow [(0,\,0) \to (1,\,0)], \ldots \sigma^1_n \leftrightarrow [(n-1,\,0) \to (n,\,0)]$. Likewise, the m directed edges of the zeroth row of $\mathcal{G}_{n,m}$ are identified with the elements of σ^2. Notice we go with rows and columns rather than x and y coordinates.

2.2. **Interpretation of S, I, and D on the graph.** Insertions I are interpreted as the directed horizontal edges, more precisely, $[(i,\,j) \to (i,\,j+1)],\ 1 \le i \le n-1,\ 0 \le j \le m-1$ is the insertion of $-$ between the ith and $i+1$ characters of the first sequence σ^1. That $-$ corresponds to σ^2_j which was earlier identified with the edge $[(0,\,j),\,(0,j+1)]$. Segments $[(0,\,j) \to (i,\,j+1)],\ \ 0 \le j \le m-1$ are the insertions $-$ at the very beginning of σ^1 and assigning that to σ^2_j. Similarly interpret the insertions along the very lower edges.

Deletions D are interpreted as directed vertical segments. More precisely, the directed vertical edge $[(i,\,j) \to (i+1,\,j)],\ 0 \le i \le n-1,\ 0 \le j \le m$ is the assignment of σ^1_{i+1} (which was identified with the segment $[(i,\,0),\,(i+1,\,0)]$) to $-$.

Directed diagonals $[(i,\,j) \to (i+1,\,j+1)],\ 1 \le i \le n,\ 1 \le j \le m$ are *substitutions* S. Thus the substitution $[(i,\,j) \to (i+1,\,j+1)],\ 0 \le i \le n-1,\ 0 \le j \le m-1$ means the assignment $\sigma^1_{i+1} \to \sigma^2_{j+1}$.

Thus every edit string corresponds to a unique path from (0, 0) to $(n,\,m)$ and conversely. In other words aligning two sequences is equivalent to determining a (continuous) path from (0, 0) to $(n,\,m)$.

An immediate consequence of the identification is the following recurrence relation.

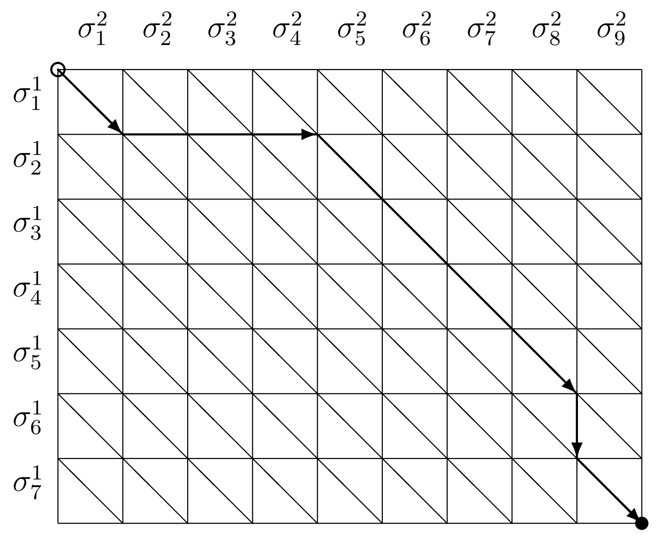
Corollary 2.1. *Let* $a_{n,\,m},\ n \ge 1,\ m \ge 1$ *be the total number of paths from (0, 0) to* $(n,\,m)$. *Then*

$$a_{n,\,m} = a_{n-1,\,m} + a_{n,\,m-1} + a_{n-1,m-1}, \quad a_{0,\,0} = 1,$$

and we have set $a_{n,\,-1} = 0$ *and* $a_{-1,\,m} = 0$.

The proof is obvious once we look at the 3 nodes in $\mathcal{G}_{n,\,m}$ from which $(n,\,m)$ can be reached.

Example 2.1.

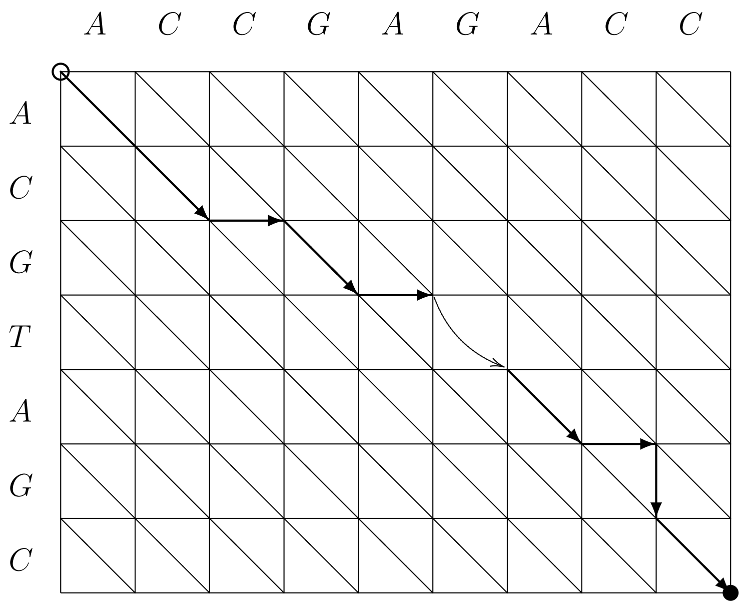

σ^2_1 σ^2_2 σ^2_3 σ^2_4 σ^2_5 σ^2_6 σ^2_7 σ^2_8 σ^2_9
σ^1_1
σ^1_2
σ^1_3
σ^1_4
σ^1_5
σ^1_6
σ^1_7

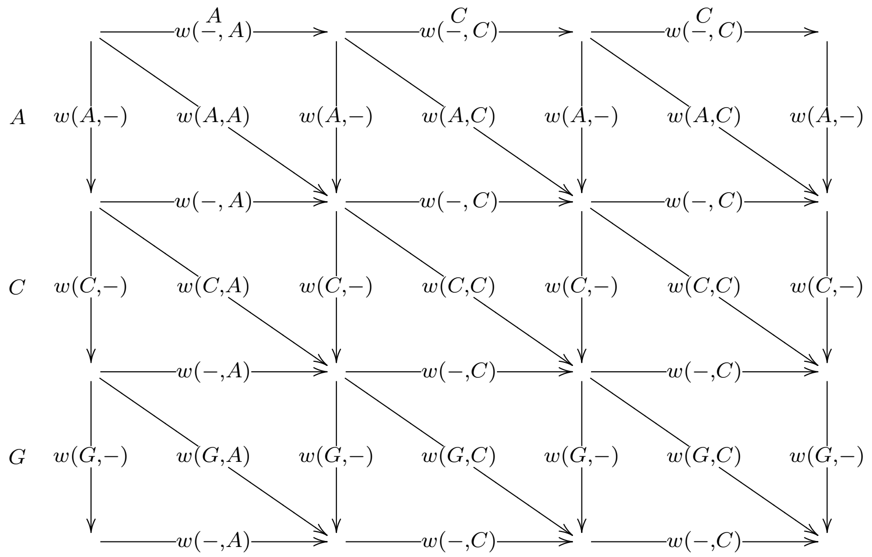
The above path is the alignment

$$\begin{array}{rcccccccccc} & S & I & I & I & S & S & S & S & D & S \\ \mu^1 = & \sigma^1_1 & - & - & - & \sigma^1_2 & \sigma^1_3 & \sigma^1_4 & \sigma^1_5 & \sigma^1_6 & \sigma^1_7 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \mu^2 = & \sigma^2_1 & \sigma^2_2 & \sigma^2_3 & \sigma^2_4 & \sigma^2_5 & \sigma^2_6 & \sigma^2_7 & \sigma^2_8 & - & \sigma^2_9 \end{array}$$

2.3. **Weighted paths.** As a means of distinguishing paths, we now assign weights to the edges of the graph, that is, a weight is assigned for each substitution, insertion, or deletion. The insertion $[(i,\,j) \rightarrow (i,\,j+1)]$ is assigned the weight $w(-,\,\sigma^2_{j+1})$. The deletion $[(i,\,j) \rightarrow (i+1,\,j)]$ is assigned the weight $w(\sigma^1_{i+1},\,-)$. The substitution $[(i-1,\,j-1) \rightarrow (i,\,j)]$ is assigned the weight $w(\sigma^1_i,\,\sigma^2_j)$. These weight matrices are given, below; the $(5,\,5)$ entry is left blank since a correspondence insertion to deletion $(-,\,-)$ is, of course, not used. The weight of a path is the total weight of edge of the path and it is viewed as the cost of the path.

Example 2.2. *The following illustrate the path corresponding to the edit sequence SSISISSIDS and the weights of the first few edges.*





The weight of the edit string (path) SSISISSIDS applied to σ^1 and σ^2 in example (1.1)

$$w(A,\, A) + w(C,\, C) + w(-,\, C) + w(G,\, G) + w(-,\, A) + w(T,\, G) +$$
$$w(A,\, A) + w(-,\, C) + w(G,\, -) + w(C,\, C).$$

The weights described can be put in the form of the weight matrix.

$$\begin{pmatrix} w_{(A,A)} & w_{(A,C)} & w_{(A,G)} & w_{(A,T)} & w_{(A,-)} \\ w_{(C,A)} & w_{(C,C)} & w_{(C,G)} & w_{(C,T)} & w_{(C,-)} \\ w_{(G,A)} & w_{(G,C)} & w_{(G,G)} & w_{(G,T)} & w_{(G,-)} \\ w_{(T,A)} & w_{(T,C)} & w_{(T,G)} & w_{(T,T)} & w_{(T,-)} \\ w_{(-,A)} & w_{(-,C)} & w_{(-,G)} & w_{(-,T)} & \end{pmatrix}$$

For simplicity, the diagonal entries are taken as zero. All deletions and insertions will be given the same weight. Thus the all fifth column entries and 5th row entries are the same. Finally all mismatch substitutions will have the same weight.

3. The algorithm

The algorithm provides the path(s) of minimal cost to reach all points on the graph from $(0,\, 0)$.

The concept of the algorithm is that given a path of total minimal weight, the minimal weight path to *any* point on the graph is the sub path of the path reaching the point.

The weight of any path lying wholly on the top row (zeroth row) is the sum of weights of the horizontal segments of the graph. The only path that can reach a point on the top row has to lie wholly on the top row. Similarly for the vertical path lying wholly on the left most column (zeroth column).

Now consider the three paths that can reach $(1,\, 1)$: S, ID, or DI. Pick the path with minimal (total) weight and color it in blue. Now consider the point $(1,\, 2)$. It can be reach from and only from $(1,\, 1)$, $(0,\, 1)$, or $(\, 0,\, 2)$. The minimal weight path all those 3 points is now known. Hence obtain the minimal path to $(1,\, 2)$ and color it blue. Continuing in this manner we will have the minimal path to every point.

Now we will formally write the algorithm. Here $M(i,\, j)$ is the total weight of along the minimal path of reaching i the row jth column position. First initialize the algorithm by defining $M(0,\, 0) = 0$. Then $M(0,\, j) = M(0,\, j-1) + w(-,\, \sigma^2_j), \quad 1 \leq j \leq m$ and $M(i,\, 0) = M(i-1,\, 0) + w(\sigma^1_i,\, -)$, $1 \leq i \leq n$.

The loop: for $1 \leq i \leq n,\ 1 \leq j \leq m$

$$M(i,\, j) = \min \begin{cases} M(i-1,\, j-1) + w(\sigma^1_i,\, \sigma^2_j) \\ M(i-1,\, j) + w(\sigma^1_i, -) \\ M(i,\, j-1) + w(-,\, \sigma^2_j) \end{cases}$$

The backtrack: Trace the blue optimal path from (n, m) to $(0, 0)$.