Algorithmic Bioinformatics DD2450, spring 2010, Lecture 3

Lecturer Jens Lagergren
Several current and previous students
will be acknowledged in a separate document.

April 18, 2010

Chapter 1

Pairwise sequence alignment cont.

1.1 Introduction

1.2 Global Alignment with Affine Gap Penalty

1.2.1 Slippage and Affine Gap

In alignments (DNA, RNA, and proteins), missing nucleotides often occur in consecutively groups called gaps. This is because, evolution frequently deletes or inserts "entire substrings" as a unit, as opposed to deleting or inserting individual nucleotides. When observing a gap it is natural to ask whether several consecutive nucleotides have been removed in one of sequence or if several new nucleotides have been added consecutively to the other sequences. In fact it is impossible for us to tell the difference between these two cases and therefore they are both refereed to as indels.

One mechanism that can introduce gaps in alignments is so called slippage in which DNA polymerase makes a mistake during the replication of DNA by not traversing a loop or by traversing it twice, see Figure 1.1.

Until now we have assumed that $s(A,-) = s(T,-) = s(C,-) = s(G,-) = p$, for some penalty p. In global alignment with affine gap penalty, we change our model slightly to cover slippage (and similar effects). We introduce two new penalties; one penalty for opening a new gap (cost d), and one penalty for extending an existing gap (cost e). Normally, the penalty for opening a new gap opening is greater than extending an existing gap. The two penalties are regarded as negative values, hence we have ($d < e < 0$), and this type of gap cost is called *affine*.

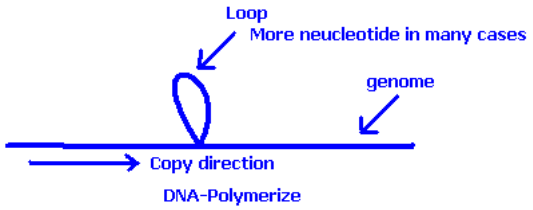

Figure 1.1: Illustration of slippage. Sometimes a loop occurs in the sequence. Here it is possible that the loop is copied exactly once during mutations, which results no change in the sequence. It is possible that the loop is not copied at all which results in a deletion and that the loop is copied two which results in duplication.

Example Given the sequences `GTAG` and `GTTTC` a possible alignment is:

```
GT--AG
GTTT-C
```

The first, upper sequence contains one gap opening and one gap extension while the second sequence contains one gap opening. Using affine gap cost the score for this alignment is:

$$\sigma = s(\mathtt{G},\mathtt{G}) + s(\mathtt{T},\mathtt{T}) + d + e + d + s(\mathtt{G},\mathtt{C})$$

Note that the alignment in the example is considered to have two gaps, not a single 3-position wide gap. However, it is worth mentioning that some algorithms for local aligning in fact considers a series of consecutive positions with indels to be one large gap even if all blank symbols do not appear in the same sequence.

1.2.2 Definitions

Input: Two homologous sequences $X = x_1, \ldots, x_m$ and $Y = y_1, \ldots, y_n$.

Output: The alignment score γ(X,Y) (and possibly the corresponding alignment).

Definition 1. *Let*

1. $g(i,j)$ = *optimal score of* X^i *and* Y^j *when* x_i *and* y_j *are matched*
2. $g_x(i,j)$ = *optimal score of* X^i *and* Y^j *when the last position in* X^i *is matched with a blank symbol*
3. $g_y(i,j)$ = *optimal score of* X^i *and* Y^j *when the last position in* Y^j *is matched with a blank symbol*

1.2.3 Dynamic programming procedure

The global aligning with affine gap penalty is also a recursive procedure that can be solved efficiently by means of dynamic programming. With similar reasoning as earlier the recursions for the new matrices defined above can be derived as:

$$g(i,j) = \max \begin{cases} g(i-1,j-1) + s(x_i, y_j) \\ g_x(i-1,j-1) + s(x_i, y_j) \\ g_y(i-1,j-1) + s(x_i, y_j) \end{cases}$$

$$g_x(i,j) = \max \begin{cases} g(i-1,j) + d \\ g_y(i-1,j) + d \\ g_x(i-1,j) + e \end{cases}$$

$$g_y(i,j) = \max \begin{cases} g(i,j-1) + d \\ g_x(i,j-1) + d \\ g_y(i,j-1) + e \end{cases}$$

The optimal global alignment score is now given by $\max\{g(m,n), g_x(m,n), g_y(m,n)\}$. The alignment that corresponds to the optimal score can be extracted in the same way as before using back-pointers.
A set of base cases is needed for each matrix for the computation of its first row and its first column. Since three matrices are computed, three sets of base cases are needed. These are:

$$\begin{array}{ll} g(0,0) = 0 & \\ g(i,0) = -\infty & 1 \leq i \leq m \\ g(0,j) = -\infty & 1 \leq j \leq n \end{array}$$

$$\begin{array}{ll} g_x(i,0) = -\infty & 0 \leq i \leq m \\ g_x(0,j) = d + (j-1)e & 1 \leq j \leq n \end{array}$$

$$\begin{array}{ll} g_y(i,0) = d + (i-1)e & 1 \leq i \leq m \\ g_y(0,j) = -\infty & 0 \leq j \leq n \end{array}$$

Algorithm complexity

Time: $O(nm)$.
Three $(n+1) \times (m+1)$-matrices are computed. A constant amount of work is needed for each matrix element.

Memory usage for the optimal alignment matrix using back-pointers: $O(mn)$.

Memory usage for the optimal score only: $O(min(m,n))$
Only the previous and the current row of each matrix are needed in memory when the corresponding alignment is unimportant.

1.3 Motivation for additive score

Using an additive score may at first not seem like a natural or even good choice. A motivation for why it is sound to use an additive score is therefore needed.

We make the following two simplifying assumptions:

- Sequences have no gaps
- The positions are independent

Let $X = x_1, \ldots, x_n$ and $Y = y_1, \ldots, y_n$ be two sequences of the same length. We have two different models for these sequences. The first model is the random model R, where X and Y are not homologous (i.e. no common ancestor to X and Y). Under this model the probability of observing an individual nucleotide is given by nucleotide probabilities q_A, q_T, q_C and q_G. The probability of observing X and Y under this model is:

$$Pr[X, Y|R] = \prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} q_{y_i}$$

The second model is the homology model H, where X and Y in fact have a common ancestor. Under this model the probability of obsering two nucleotides in the same position in X and Y is given by a pairwise probability: $p_{A,A}, p_{A,T}, \ldots, p_{G,G}$, respectively. The probability of obsering X and Y under this model is:

$$Pr[X, Y|H] = \prod_{i=1}^{n} p_{x_i,y_i}$$

Let us consider the ratio between these two likelihoods:

$$\frac{Pr[X, Y|H]}{Pr[X, Y|R]} = \frac{\prod_{i=1}^{n} p_{x_i,y_i}}{\prod_{i=1}^{n} q_{x_i} \prod_{i=1}^{n} q_{y_i}} = \prod_{i=1}^{n} \frac{p_{x_i,y_i}}{q_{x_i} q_{y_i}}$$

By taking the logarithm of this ratio we get:

$$\sum_{i=1}^{n} log \frac{p_{x_i,y_i}}{q_{x_i} q_{y_i}}$$

From this expression we conclude that additive score is in fact OK, if we define our similarity function s as

$$s(x, y) = log \frac{p_{x,y}}{q_x q_y}$$

Now the question arises how we can estimate $P_{NN'}$ and q_N. This is a bioinformatics problem rather than a mathematical problem. We have no mathematical solution for this. We can approximate or measure this by measuring real mutations and count by observation. For this purpose, we take the "so-called"

"trusted" alignments of pairs on the same distance, i.e. we take the aligned sequences that we can "trust", all on same distance, say the pairs of sequence that agree in 99% of all positions. Now, let

$$A_{NN'} = \#N, N' \text{mutation pairs}$$

$$B_N = \#N \text{in the sequences}$$

$$P_{NN'} = \frac{A_{NN'}}{\sum_{NN'} A_{NN'}}$$

$$q_N = \frac{B_N}{\sum_{N'} B_{N'}}$$

Then we extrapolate to other distances. Example of such matrices are PAM, BLOSSOM, etc.