Guided Sequence Alignment

Abdullah N. Arslan
University of Vermont, USA

INTRODUCTION

Sequence alignment is one of the most fundamental problems in computational biology. Ordinarily, the problem aims to align symbols of given sequences in a way to optimize similarity score. This score is computed using a given scoring matrix that assigns a score to every pair of symbols in an alignment. The expectation is that scoring matrices perform well for alignments of all sequences. However, it has been shown that this is not always true although scoring matrices are derived from known similarities. Biological sequences share common sequence structures that are signatures of common functions, or evolutionary relatedness. The alignment process should be guided by constraining the desired alignments to contain these structures even though this does not always yield optimal scores. Changes in biological sequences occur over the course of millions of years, and in ways, and orders we do not completely know. Sequence alignment has become a dynamic area where new knowledge is acquired, new common structures are extracted from sequences, and these yield more sophisticated alignment methods, which in turn yield more knowledge. This feedback loop is essential for this inherently difficult task.

The ordinary definition of sequence alignment does not always reveal biologically accurate similarities. To overcome this, there have been attempts that redefined sequence similarity. Huang (1994) proposed an optimization problem in which close matches are rewarded more favorably than the same number of isolated matches. Zhang, Berman & Miller (1998) proposed an algorithm that finds alignments free of low scoring regions. Arslan, Eğecioğlu, & Pevzner (2001) proposed length-normalized local sequence alignment for which the objective is to find subsequences that yield maximum length-normalized score where the length-normalized score of a given alignment is its score divided by sum of subsequence-lengths involved in the alignment. This can be considered as a context-dependent sequence alignment where a high degree of local similarity defines a context. Arslan, Eğecioğlu, & Pevzner (2001) presented a fractional programming algorithm for the resulting problem. Although these attempts are important, some biologically meaningful alignments can contain *motif*s whose inclusions are not guaranteed in the alignments returned by these methods. Our emphasis in this chapter is on methods that guide sequence alignment by requiring desired alignments to contain given common structures identified in sequences (motifs).

BACKGROUND

Given two strings S_1, and S_2, the *pairwise sequence alignment* can be described as a writing scheme such that we use a two-row-matrix in which the first row is used for the symbols of S_1, and the second row is used for those of S_2, and each symbol of one string can be aligned to (i.e. it appears on the same column with) a symbol of the other string, or the blank symbol ′-′. A matrix obtained this way is called an alignment matrix. No column can be entirely composed of blank symbols. Each column has a weight. The score of an alignment is the total score of the columns in the corresponding alignment matrix. Fig. 1 illustrates an example alignment between two strings ACCGCCAGT and TGTTCACGT.

The following is the Needleman-Wunsch global alignment formulation (Durbin et al., 1998) that, for two given strings $S_1[1] \ldots S_1[n]$ and $S_2[1] \ldots S_2[m]$, computes

Figure 1. An example alignment with five matches

$$\begin{array}{ccccccccccc}
A & C & C & G & - & C & C & A & - & G & T \\
 & & & | & & & | & | & & | & | \\
T & - & - & G & T & T & C & A & C & G & T
\end{array}$$

$$H_{i,j} = \max\{\, H_{i-1,j}+\gamma(S_1[i],\text{'-'}),\; H_{i-1,j-1}+\gamma(S_1[i],S_2[j]),\; H_{i,j-1}+\gamma(\text{'-'},S_2[j]) \,\} \qquad (1)$$

for all $i, j, 1 \leq i \leq n, 1 \leq j \leq m$, with the boundary values $H_{0,0} = 0$, $H_{0,j} = H_{0,j-1} + \gamma(\text{'-'},S_2[j])$, and $H_{i,0} = H_{i-1,0} + \gamma(S1[i], \text{'-'})$ where γ is a given score function. Then $H_{n,m}$ is the maximum global alignment score between S_1 and S_2. For the strings in Fig. 1, if $\gamma(x,y)$=1 when $x=y$, and 0 otherwise, then the maximum alignment score *is* $H_{9,9}=5$. The figure shows an optimal alignment matrix with 5 matches each indicated by a vertical line segment. The well-known Smith-Waterman local alignment algorithm (Durbin et al., 1998) modifies Equation (1) by adding 0 as a new max-term. This means that a new local alignment can start at any position in the alignment matrix if a positive score cannot be obtained by local alignments that start before this position. For the example strings in Fig. 1, if the score of a match is +1, and each of all other scores is -1, then the optimum local alignment score is 3, and it is obtained between the suffixes CAGT, and CACGT of the two strings, respectively.

The definition of pairwise sequence alignment for a pair of sequences can be generalized to the *multiple sequence alignment* problem (Durbin et al., 1998). A multiple sequence alignment of *k* sequences involves a *k*-row alignment matrix, and there are various scoring schemes (e.g. sum of pairwise distances) assigning a weight to each column.

Similarity measures based on computed scores only does not always reveal biologically relevant similarities (Comet & Henry, 2002). Some important local similarities can be overshadowed by other alignments (Zhang, Berman & Miller, 1998). A biologically meaningful alignment should include a region in it where common sequence structures (if they exist) are aligned together although this would not always yield higher scores. It has also been noted that biologists favor integrating their knowledge about common patterns, or structures into the alignment process to obtain biologically more meaningful similarities (Tang et al., 2002; Comet & Henry, 2002). For example, when comparing two protein sequences it may be important to take into account a common specific or putative structure which can be described as a subsequence. This gave rise to a number of constrained sequence alignment problems. Tang et al. (2002) introduced the *constrained multiple sequence alignment* (CMSA) problem where the constraint for the desired alignment(s) is inclusion of a given subsequence. This problem and its variations have been studied in the literature, and different algorithms have been proposed (e.g. He, Arslan, & Ling, 2006; Chin et al., 2003).

Arslan and Eğecioğlu (2005) suggested that the constraint could be inclusion of a subsequence within a given edit distance (number of edit operations to change one string to another). They presented an algorithm for the resulting constrained problem. This was a step toward allowing in the constraint patterns that may slightly differ in each sequence.

Arslan (2007) introduced the *regular expression constrained sequence alignment* (RECSA) problem in which alignments are required to contain a given common sequence described by a given regular expression, and he presented an algorithm for it.

MAIN FOCUS

Biologists prefer to incorporate their knowledge into the alignment process by guiding alignments to contain known sequence structures. We focus on motifs that are described as a subsequence, or a regular expression, and their use in guiding sequence alignment.

Subsequence Motif

The constraint for the alignments sought can be inclusion of a given pattern string as a subsequence. A motivation for this case comes from the alignment of RNase (a special group of enzymes) sequences. Such sequences are all known to contain "HKH" as a substring. Therefore, it is natural to expect that in an alignment of RNase sequences, each of the symbols in "HKH" should be aligned in the same column, i.e. an alignment sought satisfies the constraint described by the sequence "HKH". The alignment shown in Fig. 1 satisfies the constraint for the subsequence pattern "CAGT".

Chin et al. (2003) present an algorithm for the constrained multiple sequence alignment (CMSA) problem. Let $S_1, S_2, ..., S_n$ be given *n* sequences to be aligned, and let $P[1..r]$ be a given pattern constraining the alignments. The algorithm modifies the dynamic-programming solution of the ordinary multiple sequence alignment (MSA). It adds a new dimension of size $r+1$ such that each position k, $0 \leq k \leq r$, on this dimension can be considered as a layer that corresponds to the ordi-