Sequence analysis

Statistical evaluation and comparison of a pairwise alignment algorithm that a priori assigns the number of gaps rather than employing gap penalties

Yasuyuki Nozaki and Matthew Bellgard*
Centre for Bioinformatics and Biological Computing, Murdoch University, Murdoch, WA 6150, Australia

Received on July 11, 2004; revised on November 17, 2004; accepted on November 29, 2004
Advance Access publication December 10, 2004

ABSTRACT

Motivation: Although pairwise sequence alignment is essential in comparative genomics, it has proven difficult to precisely determine the gap penalties for a given pair of sequences. A common practice is to employ default penalty values. However, there are a number of problems associated with using gap penalties. First, alignment results can vary depending on the gap penalties, making it difficult to explore appropriate parameters. Second, the statistical significance of an alignment score is typically based on a theoretical model of non-gapped alignments, which may be misleading. Finally, there is no way to control the number of gaps for a given pair of sequences, even if the number of gaps is known in advance.

Results: In this paper, we develop and evaluate the performance of an alignment technique that allows the researcher to assign a priori set of the number of allowable gaps, rather than using gap penalties. We compare this approach with the Smith–Waterman and Needleman–Wunsch techniques on a set of structurally aligned protein sequences. We demonstrate that this approach outperforms the other techniques, especially for short sequences (56–133 residues) with low similarity (<25%). Further, by employing a statistical measure, we show that it can be used to assess the quality of the alignment in relation to the true alignment with the associated optimal number of gaps.

Availability: The implementation of the described methods SANK_AL is available at http://cbbc.murdoch.edu.au/
Contact: matthew@cbbc.murdoch.edu.au

INTRODUCTION

Pairwise sequence alignment is one of the most fundamental tools for comparing DNA and protein sequences. It establishes the basis for the interpretation of evolutionary and functional relationships between gene sequences and species. Through their seminal work in 1970, Needleman and Wunsch (1970) provided the catalyst to numerous other profound contributions to sequence alignment. To date, the most successful and widely understood algorithm is the Smith–Waterman technique (Smith and Waterman, 1981), which can detect similar segments in genomic sequences with maximal score if the compared sequences are closely related. More recently, there have been a number of modifications to the Smith–Waterman technique as well as novel approaches that are based on structural information (Geourjon et al., 2001; Wallqvist et al., 2000), profile information (Durbin et al., 1998; Gough et al., 2001) and evolutionary models of insertions and deletions (Knudsen and Miyamoto, 2003; Irie et al., 1999).

Syntonymous with these alignment techniques are gap penalties that must be prescribed in order to obtain optimal or near optimal alignments. However, it has proven difficult to precisely determine the penalties that should be selected for a given pair of sequences. There are a number of unavoidable problems when employing gap penalties. Firstly, sequence alignment results vary depending on the penalties associated with the initiation and extension of gaps, which attempt to model insertion and/or deletion events that have occurred in either or both sequences through the course of their evolution. Biologists often find it difficult to determine the gap parameters in order to obtain the true ‘biological’ alignment. It is well known that artificially assigned default gap parameters are not selective and/or sensitive for all types of alignments (Arslan et al., 2001; Apostolico and Giancarlo, 1998; Morgenstern, 1999; Pearson, 1995; Roytb erg, 1998). Secondly, the alignments with gaps do not always produce correct E- and P-value statistics. These statistics are based on a solid theoretical foundation for local alignments without gaps (Dembo et al., 1994; Karlin and Altschul, 1990). According to numerous computational experiments and some analytic results (Altschul and Gish, 1996; Altschul et al., 1990; Arratia and Waterman, 1994; Collins et al., 1988; Mott, 1992; Smith et al., 1985; Pearson, 1988; Waterman and Vingron, 1994a,b), it has been suggested that the formula for calculating statistics of ungapped alignments should be applied to gapped alignment as well. However, the statistics of gapped alignments do not exactly fit the theoretical distribution (extreme value distribution). Therefore, very small E- and P-values may not be reliable. Finally, there is no way to control the number of gaps to be assigned for a given pair of sequences even if the number of gaps is known in advance. This problem has been highlighted previously (Bellgard et al., 2003; Elleman, 1978; Sankoff and Cedergren, 1973; Zhu et al., 1997).

There is an alignment program, DIALIGN 2, which does not employ any gap penalties (Morgenstern, 1999). This approach uses segment-to-segment comparison rather than residue-to-residue comparison and alignments are composed from gap-free segment pairs. However, DIALIGN 2 requires user-defined parameters to determine the minimum size of segment and the significance between
two segments’ pairs and one must therefore specify appropriate parameters.

In a previous study, an approach that constrains the maximal number of allowable gaps was developed (Bellgard et al., 2003). This implementation, which is referred to as ‘gap mapping’, is a dynamic programming approach that employs an extra dimension corresponding to the number of gap constraints. Thus, for an a priori defined of the maximum allowable number of gaps \( k \), an alignment is obtained with the best alignment score with at most \( k \) gaps. Changing the number of allowable gaps can give a different alignment. Therefore, a novel way to explore optimal and near optimal alignments was developed (Bellgard et al., 2003). A search of the literature (and M. Waterman, personal communication) revealed that this implementation was related closely to an approach proposed by Sankoff (1972). Surprisingly, this approach has received little or no attention. For this reason, we have referred to our implementation as the SANK_AL approach.

In the present study, the SANK_AL implementation is evaluated and compared to both the Needleman–Wunsch algorithm (Needleman and Wunsch, 1970) as a global alignment technique and the Smith–Waterman algorithm (Smith and Waterman, 1981) as a local alignment technique. A statistical significance score \( P \)-value is developed based on a Monte Carlo procedure (Lipman and Pearson, 1985; Gilks et al., 1995) and used as a measure of the alignment for a given number of gaps. The sequences used for comparison are pairs of sequences taken from structurally aligned BALIBASE for a range of protein sequence length and percentage similarity (Thompson et al., 1999a,b). We show that in the majority of cases, the SANK_AL achieves biologically meaningful alignments over the other two techniques without the need to explore gap penalty parameter space. As SANK_AL produces more than one alignment (one alignment corresponds to a given number of maximal allowable gaps), each alignment produces its own score and associated statistical significance. Interestingly, the SANK_AL technique, as we shall show, in the majority of cases, associates the correct biological alignment to a minimal \( P \)-value for a given number of allowable gaps. This form of self-evaluation appears to be intuitive and novel in that it allows for the assessment of the most appropriate alignment out of the range of possible alignments. SANK_AL is available at http://cbbc.murdoch.edu.au/

**METHODS**

**Outline of SANK_AL algorithm**

The SANK_AL algorithm implementation developed independently is in fact a modification to the algorithm proposed by Sankoff (1972). The difference between the SANK_AL implementation and the original Sankoff algorithm is that SANK_AL can apply not only to a given pair of DNA sequences but also to amino acid sequences as well as provide global alignments for them. The SANK_AL algorithm is described briefly below. Let \( \{a_i\} \) (where \( i = 1, 2, \ldots, m \)) and \( \{b_j\} \) (where \( j = 1, 2, \ldots, n \)) be two sequences to be aligned and \( k \) be a number of maximal allowable gaps. To match a sequence \( a_i \) against another sequence \( b_j \), three by \( n \) by \( (k + 1) \) matrices \( P \), \( Q \) and \( R \) are constructed. The three matrices correspond to optimal alignments ending \( (a_i, b_j), (\cdot, a_i) \) and \( (a_i, \cdot) \), respectively, where ‘\( \cdot \)’ is a gap character. The matrix element in \( P \), \( Q \) and \( R \) at position \( s,t,g \) denotes the best score obtainable by matching \( a_i \) (where \( i = 1, 2, \ldots, s \) and \( b_j \) (where \( j = 1, 2, \ldots, t \)) including \( g \) gap regions. \( P[i][j][g], Q[i][j][g] \) and \( R[i][j][g] \) are initialized as follows:

\[
\begin{align*}
P[i][1][g] &= \begin{cases} s(a_i, b_1) & g = 0 \\ -\infty & g \neq 0 \end{cases} \\
Q[1][j][g] &= \begin{cases} 0 & g = 0 \\ -\infty & g \neq 2 \end{cases} \\
R[1][j][g] &= \begin{cases} 0 & g = 0 \\ -\infty & g \neq 2 \end{cases}
\end{align*}
\]

for all \( 0 \leq g \leq k, \) and

\[
P[i][j][g] = \begin{cases} s(a_i, b_j) & g = 1 \\ -\infty & g \neq 1 \end{cases}
\]

\[
Q[i][j][g] = \begin{cases} 0 & g = 2 \\ -\infty & g \neq 2 \end{cases}
\]

\[
R[i][j][g] = \begin{cases} s(a_i, b_j) & g = 1 \\ -\infty & g \neq 2 \end{cases}
\]

for all \( 0 \leq g \leq k \) and \( 2 \leq j \leq n, \) and

\[
P[i][1][g] = \begin{cases} s(a_i, b_1) & g = 1 \\ -\infty & g \neq 1 \end{cases}
\]

\[
Q[i][1][g] = \begin{cases} 0 & g = 2 \\ -\infty & g \neq 2 \end{cases}
\]

\[
R[i][1][g] = \begin{cases} s(a_i, b_1) & g = 1 \\ -\infty & g \neq 2 \end{cases}
\]

for all \( 0 \leq g \leq k \) and \( 2 \leq i \leq m, \) where \( s(a_i, b_j) \) is a score in a substitution matrix corresponding to \( a_i \) and \( b_j \). The element \( -\infty \) implies impossible alignments for a given \( i, j \) and \( g \). For instance, \( P[1][3][2] \) should be \( -\infty \), because it is impossible to align \( a_1 \) and \( b_2 b_3 \) satisfying both constraints allowing two gap regions and ending \( (a_1, b_3) \). The element 0 implies that such alignments have no residue-to-residue pair. For instance, \( R[2][1][3] \) is derived only from the case of alignment \( a_1 - a_2 \) and \( -b_1 - \), therefore 0 is assigned for it. Then inner elements of matrices are calculated recursively by

\[
P[i][j][g] = \max(P[i−1][j−1][g]+s(a_i, b_j), P[i−1][j−1][g], Q[i−1][j−1][g]+s(a_i, b_j), R[i−1][j−1][g]+s(a_i, b_j)),
\]

\[
Q[i][j][g] = \max(P[i][j−1][g]+s(a_i, b_j), P[i−1][j−1][g]), Q[i−1][j][g]+s(a_i, b_j), R[i−1][j−1][g]+s(a_i, b_j)),
\]

\[
R[i][j][g] = \max(P[i−1][j−1][g], Q[i−1][j−1][g], R[i−1][j−1][g]+s(a_i, b_j)),
\]

for all \( 1 \leq i \leq m, 1 \leq j \leq n \) and \( 0 \leq g \leq k \).
To get an optimal alignment for a pair of sequences \( \{a_i\} \) (where \( i = 1, 2, \ldots, m \)) and \( \{b_j\} \) (where \( j = 1, 2, \ldots, n \)) with \( g \) gap regions, first, the highest value \( C \) among \( P[m][n][g] \), \( Q[m][m][g] \) and \( R[m][n][g] \) is selected. The matrix containing \( C \) determines whether the last pair of alignment has one of the following patterns \((a_m, b_n), (-', b_n) \) or \((a_m, '-')\). Then, just as in Needleman–Wunsch algorithm, matrices are traced back to find a path of elements that contribute the highest value \( C \). Finally, the alignment is constructed by assigning \((a_i, b_j), (-', b_j) \) or \((a_i, '-')\) according to the matrices of elements in the path. Similarly, for the other number of gap regions \( g' \), an alignment can be obtained as well by finding a path of elements contributing to the highest value \( C' \) among \( P[m][n'][g'] \), \( Q[m'][m][g'] \) and \( R[m'][n][g'] \).

In this algorithm, the number of calculations is obviously proportional to the size of matrices, in other words, the total running time of this SANK_AL algorithm is \( O(mnk) \). Note that this SANK_AL algorithm does not make use of any gap penalties or any other specific information in the process of alignment. Moreover, this algorithm can find subsequences under the given number of gap regions, in other words, satisfying insertion and/or deletion constraints.

**Dataset of structure alignments**

To assess the quality of our alignment algorithm, we require a standard for the ‘true’ alignment. This is usually derived from alignments of protein structures. Several comparative programs of protein alignment algorithms (Sauder et al., 2000; Eldonsson, 2002) use SCOP, a protein family database (Andreeva et al., 2004). However, as SCOP does not have alignment information, it is necessary to make ‘true’ alignments from it by using structural alignment programs such as CE (Shindyalov and Bourne, 1998) and DALI (Holm and Sander, 1999). It is acknowledged that sequence alignments produced by two structure alignment programs may differ substantially when the sequence identity between them is low (Feng and Sippl, 1996; Godzik, 1996).

For that reason, in this study, we selected BAliBASE which is manually constructed and is not intended for any specific alignment program. It should be noted that BAliBASE has a potential bias toward global alignment programs since aligned sequences are trimmed down at the boundaries of the alignment to the core block only (Karplus and Hu, 2001). However, it has been used as a standard multiple alignment sequence set to evaluate alignment algorithms in recent studies (Thompson et al., 1999b; Karplus and Hu, 2001; Notredame et al., 2000). BAliBASE is currently available at http://www-igbmc.u-strasbg.fr/BioInfo/BAliBASE2/. BAliBASE is classified into eight hierarchical reference sets by the length and similarity of the sequences in the core blocks and by the presence of insertions, N/C-terminal extensions and repeats. In this paper, Reference 1 which contains 82 multiple alignment sequences was used. Reference 1 is divided into nine groups depending on the length of the sequences: short (between 56 and 133 residues), medium (182–317 residues) and long (329–995 residues); and the identity of sequences: <25, 20–40 and >35%. For the analysis of pairwise alignment, the first and second sequences from each multiple alignment were chosen. Our test data, therefore, results in the comparison of 7–12 pairs of sequences in each group.

**Comparison of SANK_AL with other techniques**

In order to show the performance of our SANK_AL algorithm, SANK_AL (Bellgard et al., 2003) was compared with other alignment methods. As our program produces global alignments and focuses on detailed analysis of alignment for the two given sequences, we chose the Needleman–Wunsch algorithm for comparison. We also chose the Smith–Waterman algorithm for a local alignment comparison. Like SANK_AL, these two algorithms can be used generically as they are not intended for specific tasks such as fold recognition or gene finding. The Needleman–Wunsch and the Smith–Waterman algorithms were carried out through a local copy of EMBoss (Rice et al., 2000). The substitution matrix for evaluating the alignments used BLOSUM50, BLOSUM62 and PAM250 and the gap penalties employed a range of parameters for initiation (9, 10, 11, 12 and 13) and extension (0.5, 1, 2, 3 and 4). Therefore, 25 combinations of gap penalties for each substitution matrix have been tested.

To measure alignment accuracy compared to the BAliBASE structural alignment, we chose to use the \( f_{21} \) score (Sauder et al., 2000). The \( f_{21} \) is defined to be the percentage of paired residues matched against the standard alignment. Although many other measures for assessing the significance of alignments are proposed (Levitt and Gerstein, 1998; Cristobal et al., 2001), the \( f_{21} \) score is the simplest and most intuitive method and it has been used in many papers (Sauder et al., 2000; Blake and Cohen, 2001).

**Statistics of alignments**

The \( f_{21} \) score can assess the likely biological significance of the similarity although it is also necessary to know whether the alignments themselves are statistically significant or not. In this study, statistical significance was estimated by using the Monte Carlo techniques first proposed by Lipman and Pearson (1985) and examined by Comet et al. (1999) and Bacro and Comet (2001). This method is known to be less dependent on the sequence lengths and compositions (Comet et al., 1999; Bacro and Comet, 2001). For any two given sequences \( a \) and \( b \), this method compares one of the sequences \( a \) with the randomly shuffled second sequence \( b^* \) many times. The shuffled sequences share the same amino acid composition and length with the initial sequence. For an empirical mean \( \hat{\mu} \) and standard deviation \( \hat{\sigma} \) of the alignment score \( S(a, b^*) \) estimated by a number of shuffled sequences, the Z-value is defined as

\[
Z(a, b) = \frac{S(a, b) - \hat{\mu}}{\hat{\sigma}}
\]

where \( S(a, b) \) is an alignment score of \( a \) and \( b \). However, Comet et al. (1999) reported shuffling the first sequence \( Z(b, a) \) can lead to Z-values that are markedly different from those obtained by shuffling the second sequence \( Z(a, b) \). Thus, a conservative Z-value \( Z'(a, b) \) was used by following the formula suggested by Comet et al. (1999) as

\[
Z'(a, b) = \min[Z(a, b), Z(b, a)]
\]

To obtain a converged Z-value, the number of shuffling times is recommended from 100 to 1000 (Comet et al., 1999; Aude and Louis, 2002); therefore, we have used 1000 times. Then the probability (P-value) observed at least Z-value \( Z \), \( P(Z > z) \), was calculated by

\[
P(Z > z) = 1 - \exp(-e^{-\pi/(\sqrt{6}z)})
\]

where \( \gamma \) is a Euler constant. This P-value is based on the fact that the Z-value should follow the extreme value distribution, more precisely, the Grumbel distribution. In fact, interestingly, the histogram of Z-values was well fitted with its distribution for random sequences...
(a) P00520 and P04637 (short, <25% identity)

\[ f_D = 0.38 \log(P\text{-value}) = -2.02 \]

NLFVALDVFASGNTLSSITGKELRVLGLYNHNG----EWECAQTNNQGQWPSNYITPVN
 ...........................................:.................||.............:|:|.|.|.|
KGVIALMDYEPQNDDELPMKEGDCMTI1HREDDEDEIEWWWARLNDKGEYVPRNLGLYP

\[ f_D = 0.87 \log(P\text{-value}) = -4.23 \]

NLFVALYDFVASGDNTLSITKGEKLRVLGYNHNG----EWECAQTNNQGQWPSNYITPVN
 ...........................................:.................||.............:|:|.|.|.|
KGVIALMDYEPQNDDELPMKEGDCMTI1HREDDEDEIEWWWARLNDKGEYVPRNLGLYP

\[ f_D = 0.78 \log(P\text{-value}) = -3.43 \]

NLFVALYDFVASGDNTLSITKGEKLRVLGYNHNG----EWECAQTNNQGQWPSNYITPVN
 ...........................................:.................||.............:|:|.|.|.|
KGVIALMDYEPQNDDELPMKEGDCMTI1HREDDEDEIEWWWARLNDKGEYVPRNLGLYP

\[ f_D = 0.78 \log(P\text{-value}) = -3.19 \]

NLFVALYDFVASGDNTLSITKGEKLRVLGYNHNG----EWECAQTNNQGQWPSNYITPVN
 ...........................................:.................||.............:|:|.|.|.|
KGVIALMDYEPQNDDELPMKEGDCMTI1HREDDEDEIEWWWARLNDKGEYVPRNLGLYP

(b) P32081 and P15277 (short, >35% identity)

\[ f_D = 0.27 \log(P\text{-value}) = -2.84 \]

MLEGKVKWFNSEKGFGFIEVEGQDDVFVHFSAIQGEQGTKLTEEQMSFEIVEGNGRPGQAANVTKEA
 ...........................................:.................||.............:|:|.|.|.|
MSGKMTGIVKWFNDAGKFQGFTPDGDGKMEVHFSAIQNQGEYKSLDEEQKSFTIESGAKGPAAGNVTSL

\[ f_D = 0.62 \log(P\text{-value}) = -8.87 \]

MLEGKVKWFNSEKGFGFIEVEGQDDVFVHFSAIQGEQGTKLTEEQMSFEIVEGNGRPGQAANVTKEA
 ...........................................:.................||.............:|:|.|.|.|
MSGKMTGIVKWFNDAGKFQGFTPDGDGKMEVHFSAIQNQGEYKSLDEEQKSFTIESGAKGPAAGNVTSL

\[ f_D = 0.90 \log(P\text{-value}) = -15.05 \]

MLEGKVKWFNSEKGFGFIEV-EGQDDVFVHFSAIQGEQGTKLTEEQMSFEIVEGNGRPGQAANVTKEA
 ...........................................:.................||.............:|:|.|.|.|
MSGKMTGIVKWFNDAGKFQGFTPDGDGKMEVHFSAIQNQGEYKSLDEEQKSFTIESGAKGPAAGNVTSL

\[ f_D = 0.85 \log(P\text{-value}) = -13.91 \]

MLEGKVKWFNSEKGFGFIEVEGQDDVFVHFSAIQGEGGTKLTEEQMSFEIVEGNGRPGQAANVTKEA
 ...........................................:.................||.............:|:|.|.|.|
MSGKMTGIVKWFNDAGKFQGFTPDGDGKMEVHFSAIQNQGEYKSLDEEQKSFTIESGAKGPAAGNVTSL

\[ f_D = 0.80 \log(P\text{-value}) = -13.19 \]

MLEGKVKWFNSEKGFGFIEVEQDDVFVHFSAIQGEGGTKLTEEQMSFEIVEGNGRPGQAANVTKEA
 ...........................................:.................||.............:|:|.|.|.|
MSGKMTGIVKWFNDAGKFQGFTPDGDGKMEVHFSAIQNQGEYKSLDEEQKSFTIESGAKGPAAGNVTSL

Fig. 1. Two examples of SANK_AL alignments using the BLOSUM62 substitution matrix are shown (short sequence pair <25% identity and short sequence pair >35% identity). While the number of allowable gaps is changed from one to five, SANK_AL produces biologically meaningful optimal alignments for each.

RESULTS

For a given \( k \) gaps, SANK_AL produces the alignment with the best score with at most \( k \) gaps. Figure 1 shows an example of SANK_AL outputs allowing from one gap through to five gaps. It is worth noting that these alignments do not use any artificial gap penalties. Therefore, they are 'naturally' aligned using only a substitution matrix and a given maximal allowable gap.

We have compared our SANK_AL algorithm against the Needleman–Wunsch and Smith–Waterman algorithms using the BALiBASE database. In this comparison, SANK_AL performed sequence alignment allowing from 1 to 25 gaps. Figures 2 and 3 show the results of each algorithm using BLOSUM62, by measuring averages of the \( f_D \) score within each group. Note that SANK_AL produces 25 alignments for a given pair of sequences while the
Fig. 2. Comparison of SANK_AL and the Needleman–Wunsch algorithm. A range of 25 allowable gaps is used in SANK_AL. N&W1, N&W2 and N&W3 represent the top three averages of $f_D$ scores for each group among 25 gap penalty combinations. The averages of $f_D$ scores obtained from the default gap penalty are also plotted in this figure. These gap penalties are shown in the figure as insertion/extension. All comparisons are conducted against (a) long sequences, (b) medium sequences and (e) short sequences.

The other two programs produce only one. Therefore, we explored 25 gap penalty combinations for the other two programs and picked the top three averages of the $f_D$ score for each group among the 25 combinations of gap penalties. These are referred to as N&W1, N&W2 and N&W3 for the Needleman–Wunsch program in Figure 2, and S&W1, S&W2 and S&W3 for the Smith–Waterman program in Figure 3. Also, the averages of the $f_D$ score obtained from the default gap penalty (insertion = 10, extension = 0.5) are plotted in Figures 2 and 3. These gap penalty combinations are represented as initiation/extension in the box near the plots. The $f_D$ score of SANK_AL represents the highest $f_D$ among all 25 allowable gaps (SANK_AL).

As can be seen in Figures 2 and 3, in the majority of cases, the $f_D$ scores of SANK_AL alignments are the same or better than that of the other two algorithms. SANK_AL is particularly effective when a pair of sequences is short and the identity between them is low. In fact, 69.6, 56.8 and 55.1% of all SANK_AL alignments for short, medium and long sequences produce higher $f_D$ scores than those of the Needleman–Wunsch program. Similarly, 84.7, 78.0 and 71.2% of all SANK_AL alignments for short, medium and long sequences, respectively produce higher $f_D$ scores than the score of the Smith–Waterman program. Furthermore, note that there is no one gap penalty combination which can provide superior performance at all times. This highlights the difficulty in selecting appropriate gap penalties for a given pair of sequences. In fact, when the default gap penalty is employed for the Needleman–Wunsch and Smith–Waterman programs, 68.7 and 78.7% of all SANK_AL alignments outperform them, respectively. As we discuss, SANK_AL can find the optimal or near optimal number of gaps for a given pair of sequences. Consequently, in the majority of cases, SANK_AL makes biologically better alignments than the other two algorithms. Although the alignments using the BLOSUM62 are shown, other alignments using the BLOSUM50 and PAM250 matrices have similar characteristics (data not shown).

We also examined the probability of obtaining each alignment ($P$-value). Figure 4 plots the $f_D$ scores and logarithms of $P$-values while the number of allowable gaps changes from 1 to 25. As can be seen, the higher $f_D$ score and the lower $P$-value are almost correlated with each other. Actually, 73.1% of the number of gap regions, having the maximum $f_D$, correspond to the number of gap regions with the minimum $P$-value plus or minus two gaps. If the number of gap regions in alignments is less than the optimal gap, alignments
Fig. 3. Comparison of SANK_AL and the Smith–Waterman algorithm. A range of 25 allowable gaps is used in SANK_AL. S&W1, S&W2 and S&W3 represent the top three averages of \( f_D \) scores for each group among 25 gap penalty combinations. The averages of \( f_D \) scores obtained from the default gap penalty are also plotted in this figure. These gap penalties are shown as insertion/extension. All comparisons are conducted against (a) long sequences, (b) medium sequences and (c) short sequences.

must have unlikely aligned residue pairs and result in small alignment scores. As shuffled sequence alignments with a small number of gaps generally produce small alignment scores, the Z-values are also small. From the formula for \( P \)-value, smaller Z-values result in larger \( P \)-values. Similarly, if the number of gap regions in alignments is greater than the optimal gap, the alignment tends to contain many unnecessary gaps. Such alignments will be similar to shuffled sequence alignments when the number of gaps increases. Hence, the Z-values will be small. As a result, by using statistics of alignments, we can find an optimal or near optimal number of gaps effectively. However, on the other hand, 26.9% of the number of gap regions, having the maximum \( f_D \), do not correspond to the number of gap regions with the minimum \( P \)-value plus or minus two gaps. These can be classified into two different cases. One is that the alignment corresponding to the smallest \( P \)-value gives almost the same as that with the highest \( f_D \) value. In other words, the curve of \( f_D \) value is flat around the highest \( f_D \). In this case, even though the alignment obtained from the smallest \( P \)-value is not correlated to the best alignment with respect to \( f_D \) value, the alignment is still guaranteed to have biological features and the alignment is quite similar to the one corresponding to the highest \( f_D \) value. The other case is that \( P \)-value is consistently large (between 1 and \( 10^{-3} \)) while the number of allowable gaps changes. That is, the alignment obtained by SANK_AL is not regarded as a significant alignment for any number of gaps. Generally speaking, the \( f_D \) value of such alignment is quite low (<0.3). Accordingly, if one finds that \( P \)-values are consistently large while the number of gap changes, one should be wary of using this alignment. This \( P \)-value statistics allows us to avoid less meaningful alignments.

DISCUSSION

The source code of SANK_AL was written in the ANSI C++ language. As described before, the computation of SANK_AL is \( O(mnk) \) for a pair of sequences of length \( m \) and \( n \) allowing up to \( k \) gap regions. In practice, for a pair of short (56–133 residues), medium (182–317 residues) and long (329–995 residues) sequences, it takes 0.02, 0.20, and 0.99 s on average, respectively, to find the optimal alignment after testing 25 allowable gaps, running on Linux operating system with Intel® Xeon™ 2.80 GHz and 2 GB RAM. This computation time is reasonable for the investigation of a few sequences of interest, even if a number of computations to calculate the \( P \)-value statistics are required. However, this method is not applicable to a large screening of databanks unlike BLAST (Altschul et al., 1990), FASTA (Pearson and Lipman, 1988) or Automat (Cantalloube et al., 1994). SANK_AL targets detailed and thorough
analysis of alignment. In our testing, the number of allowable gaps that led to the best alignment is likely to be <1 for a pair of short sequences (96% of all short pairs and 81% of all medium pairs are under 1) or for a pair of high-similarity sequences (96% of all >35% identity pairs and 87% of all 20–40% identity pairs are under 11). Therefore, the recommended maximal gaps is 15 for a short pair or high identity pair of sequences and 25 for other sequences.

In this paper, we detail a new alignment algorithm that does not employ gaps and propose a statistical evaluation that provides the basis for identifying biologically optimal alignments. To our knowledge, this is the first implementation of the algorithm. It is shown that the SANK_AL implementation for pairwise alignment (Bellgard et al., 2003) has a better performance compared with other mainstream approaches. In particular, it performs better than other approaches when the sequence identity is low. For such low identity pairs (often referred to as the ‘Twilight Zone’), it is known that almost all existing alignment programs do not perform well (Sauder et al., 2000; Elofsson, 2002). There are some alignment programs attempting to improve alignment quality in the ‘Twilight Zone’ (Geourjon et al., 2001; Blake and Cohen, 2001). Callebaut et al. (1997) for instance, developed a method that works through a different representation of protein sequences around their hydrophobic core structure and has proven very efficient for aligning low identity proteins. Practically, as there are many low identity pairs in the PDB (Gough et al., 2001), such algorithms may be more useful.

Another feature is that SANK_AL can provide many alignments corresponding to the maximal number of allowable gaps in the alignment. It is well known that optimal alignments in terms of maximal score may not be biologically optimal. For this reason, Vingron and Argos (1990) first proposed a suboptimal alignment algorithm that generates near optimal solutions deduced from the local reliability of alignment. Subsequently various suboptimal algorithms have been proposed (Zuker, 1991; Naor and Brutlag, 1994). However, these algorithms usually create a large number of alternative alignments and it is therefore difficult to discover the differences and/or similarities among alignments within the set of alternatives (Smoot et al., 2004). The SANK_AL algorithm makes it possible to find potential biologically optimal (or near optimal) alignments among the many alignments by calculating the statistics for each of these alignments. While it appears that there are many alignments to assess for a given pair of sequences, in practice, this is not the case. Biologists can select the appropriate alignment(s) using P-value statistics and/or using prior knowledge of the sequences, such as evolutionary information or the number of introns in coding DNA.

ACKNOWLEDGEMENTS

We wish to acknowledge Ross Taplin and Mark Reynolds for their comments and suggestions, and Hitachi Software Engineering Co., Ltd for supporting Y.N. to do this work.

REFERENCES


Fig. 4. Correlations between f_D score and P-value for a range of 25 allowable gaps using (a) long sequence pair, >35% identity (P43831 versus P47897), (b) medium sequence pair, 20–40% identity (P09211 versus P20136) and (c) short sequence pair, >35% identity (P02244 versus P27686).

SANK_AL: a new gap penalty-free alignment method

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/21/8/1421/250046 by guest on 24 December 2018


