Proc. Natl. Acad. Sci. USA
Vol. 93, pp. 12098–12103, October 1996
Applied Mathematics

Multiple DNA and protein sequence alignment based on segment-to-segment comparison

(sequence similarity/partial alignments/fragment comparison/dynamic programming/functional-site identification)

BURKHARD MORGENSTERN†, ANDREAS DRESS‡, AND THOMAS WERNER†

†GSF, National Research Center for Environment and Health, Institute of Mammalian Genetics, Ingolstädter Landstrasse 1, 85764 Neuherberg, Germany; and ‡Fakultät für Mathematik, Universität Bielefeld, 33501 Bielefeld, Postfach 100131, Germany

Communicated by Manfred Eigen, Max-Planck-Institut für Biophysikalische Chemie, Göttingen-Nikolausberg, Germany, August 6, 1996 (received for review March 15, 1996)

ABSTRACT **In this paper, a new way to think about, and to construct, pairwise as well as multiple alignments of DNA and protein sequences is proposed. Rather than forcing alignments to either align single residues or to introduce gaps by defining an alignment as a path running right from the source up to the sink in the associated dot-matrix diagram, we propose to consider alignments as *consistent equivalence relations* defined on the set of all positions occurring in all sequences under consideration. We also propose constructing alignments from whole *segments* exhibiting highly significant overall similarity rather than by aligning individual residues. Consequently, we present an alignment algorithm that (*i*) is based on segment-to-segment comparison instead of the commonly used residue-to-residue comparison and which (*ii*) avoids the well-known difficulties concerning the choice of appropriate gap penalties: gaps are not treated explicitly, but remain as those parts of the sequences that do not belong to any of the aligned segments. Finally, we discuss the application of our algorithm to two test examples and compare it with commonly used alignment methods. As a first example, we aligned a set of 11 DNA sequences coding for functional helix-loop-helix proteins. Though the sequences show only low overall similarity, our program correctly aligned all of the 11 functional sites, which was a unique result among the methods tested. As a by-product, the reading frames of the sequences were identified. Next, we aligned a set of ribonuclease H proteins and compared our results with alignments produced by other programs as reported by McClure *et al.* [McClure, M. A., Vasi, T. K. & Fitch, W. M. (1994) *Mol. Biol. Evol.* 11, 571–592]. Our program was one of the best scoring programs. However, in contrast to other methods, our protein alignments are independent of user-defined parameters.**

1. Introduction

In 1970, Needleman and Wunsch (1) published an algorithm for the alignment of two protein sequences. This algorithm aligns sequences by maximizing a score which is calculated by summing over the weights associated to matches and subtracting a penalty for each gap inserted during the alignment. Virtually all present-day alignment algorithms are based—one way or the other—on the Needleman–Wunsch algorithm.

Although this algorithm produces reasonable alignments if the compared sequences are closely related, it has some well-known shortcomings: it is hardly capable of detecting similar regions of sequences with a small overall similarity, and the resulting alignments depend sensitively on a set of user-defined parameters, especially the gap penalty.

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

In addition, even though in theory the Needleman–Wunsch algorithm can be extended quite easily to produce score-optimal alignments of more than two sequences, the complexity of the algorithm grows exponentially with the number of sequences (1, 2), prohibiting extension of the algorithm to more than three sequences.

Some authors therefore have proposed to restrict the search space of the multiple alignment problem by using reasonable heuristics (3). In this way, as many as eight protein sequences, each containing several hundred amino acid residues, have been aligned simultaneously.

Another practical solution for multiple alignment is to use successive pairwise alignments of single sequences or clusters of sequences, instead of simultaneous alignment of all sequences under consideration. However, in addition to other shortcomings of this procedure, its results depend sensitively on the order of the pairwise alignments which in turn is solely determined from alignments of only two sequences at a time.

In this paper, we propose a new concept for sequence alignment, which differs fundamentally from Needleman–Wunsch based algorithms and also from the local alignment procedures proposed by Argos (4), Argos and Vingron (5), Waterman and Jones (6), and Johnson and Doolittle (7).

Instead of comparing individual residues, we use gap-free segment-to-segment alignments with variable segment length. In particular, our algorithm does not employ any gap penalties.

Regions of low similarity, often obscuring alignments based on classical Needleman–Wunsch algorithms, are excluded from our alignment. This allows detection and correct alignment of short similar regions in very long sequences of low overall similarity by our alignment.

Finally, we discuss applications of our algorithm to two test examples and compare the results with alignments produced by commonly used alignment methods.

We compared our program with several other methods in two examples. Our program was the only one of those methods that correctly aligned a set of 11 genomic DNA sequences with very limited similarity. We also applied our method to a set of ribonuclease H proteins and compared our results with a study published by McClure *et al.* (8). Here, our program was among the best scoring methods. Moreover, in contrast to other methods, our protein-alignment algorithm is independent of user-defined parameters.

To us, this seems to suggest that our proposal is well suited to produce biologically plausible alignments and that further efforts to improve its performance and to explore its potential may be justified. In addition, it should also be worthwhile to compare our approach with similarly structured proposals as suggested, for instance, by Taylor (9) or V. Brendel (personal communication, Dagstuhl Conference on Molecular Bioinformatics, 1995).

2. Basic Algorithm for the Alignment of Two DNA or Protein Sequences

In this section, we describe an algorithm for the alignment of two nucleic acid or protein sequences. The main idea of our

approach is to base alignments on comparing whole *segments* of the sequences instead of comparing single residues only. We require the segments that are to be compared have the same length, and we do not allow any gaps within them. Since such pairs of segments appear as *diagonals* in the so called *dot matrix*, we will refer to them by this term, and we call our method DIALIGN for diagonal alignment.

We also define an *alignment* to be a *consistent equivalence relation*, defined on the set of all positions of all sequences involved. Here, "consistent" means simply that the overall order of the positions in each sequence is respected. If only two sequences are to be compared, consistency holds if for any two diagonals D_1 and D_2 belonging to the alignment either $D_1 \ll D_2$ or $D_2 \ll D_1$ holds, where "$D_1 \ll D_2$" means that the positions aligned to each other in D_1 precede those aligned in D_2 in both of the respective sequences.

2.1. Weighting the Significance of Diagonals. We have to introduce a measure that enables us to assess the *significance* of a diagonal—i.e., the similarity of the respective segments—to select a suitable set of diagonals.

Let D be a fixed diagonal, let l be the length of D, and let m denote the number of matches contained in D. By $P(l,m)$, we denote the probability of any given diagonal of length l to contain at least m matches. This probability is given by

$$P(l,m) = \sum_{i=m}^{l} \binom{l}{i} p^i (1-p)^{l-i}, \qquad [1]$$

where p is the probability of a single point in the dot matrix to represent a match. For simplicity, one may assume uniform distribution—i.e., one may assume $p = 0.25$ for nucleic acid sequences and $p = 0.05$ for proteins. Next, motivated by traditions established in statistical mechanics and information theory, we consider[§] the negative logarithm

$$E(l,m) := -\ln (Pl,m) \qquad [2]$$

of the probability of a diagonal of length l to contain at least m matches, and we define the *weight* of the diagonal D by

$$w(D) := \begin{cases} E(l,m), & \text{if } E(l,m) > T \\ 0 & \text{otherwise} \end{cases}, \qquad [3]$$

where T is a user-defined threshold introduced to limit the number of diagonals under consideration and to retain only those diagonals that have a very good chance to be biologically relevant. If protein sequences or short DNA sequences are to be aligned, a threshold is not needed. In this case our algorithm is completely independent of user-defined parameters.

The weight $w(D)$ is *high* if a random-diagonal of length l is *unlikely* to contain as many as m matches. So, a high weight of a diagonal D indicates that the similarity of the corresponding segments is not expected to have come about by chance alone and, hence, is likely to be caused by those evolutionary processes one wants to uncover by comparative sequence analysis.

One important feature of our approach is that, in contrast to segment comparisons and "word searches" in conventional alignment algorithms (see for instance refs. 4–7), it enables us to compare the significance of diagonals of different lengths: formula **3** assigns high weights to shorter diagonals if they have a high rate of matches, as well as to longer diagonals with a lower rate of matches provided the diagonals are long enough.

[§]In the following, adopting standard conventions from present mathematics, we use the symbol ":=" rather than "=" in every formula in which the left-hand term is *defined* by the right-hand term, rather than found or claimed to be equal by some kind of reasoning.

2.2. Computing Maximum Alignments. If the diagonals $D_1,\ldots,D_k$ form a consistent set of diagonals, the *score* of this set is defined to be the sum $\Sigma_{i=1}^{k} w(D_i)$.

A consistent set of diagonals with maximum score is termed a *maximum alignment*.

A *maximum alignment* of two sequences $X = (X_1,\ldots,X_{L_1})$ and $Y = (Y_1,\ldots,Y_{L_2})$ of length L_1 and L_2, respectively, can be calculated by an algorithm similar to the conventional *dynamic programming* algorithms. First, one determines, for every pair of positions (i,j) with $1 \le i \le L_1$ and $1 \le j \le L_2$, all integers $k \ge 0$ with $k \le \min(i-1, j-1)$ for which the diagonal $(X_{i-k}, Y_{j-k}; \ldots; X_i, Y_j)$ from $(i-k, j-k)$ to (i,j) has a positive weight. Next, for every pair (i,j) as above, one defines a value "*score*(i,j)," which is the score of a maximum alignment of the *prefixes* $(X_1,\ldots,X_i)$ and $(Y_1,\ldots,Y_j)$. Furthermore, if this maximum alignment consists of the diagonals $D_1,\ldots,D_k$ with, say, $D_1 \ll D_2 \ll \ldots \ll D_k$, the data of the last diagonal D_k have to be retained. We express this by defining $prec(i,j) := D_k$, where in case there are several alignments with maximal score, we choose D_k to be the right-most diagonal of all diagonals in question. This convention is reflected by the apparent asymmetry in Definition **8** below.

For each diagonal $D = (X_{i-k}, Y_{j-k}; \ldots; X_i, Y_j)$ with a positive weight, the maximum sum of weights accumulated up to and including D is denoted by "$\sigma(D)$" and the diagonal "preceding" D by "$\pi(D)$."

Obviously, the values of σ and π satisfy the relations

$$\sigma(D) = score(i-k-1, j-k-1) + w(D) \qquad [4]$$

and

$$\pi(D) = prec(i-k-1, j-k-1). \qquad [5]$$

Using these notations, we can compute the values of *score* recursively by

$$score(i,j) = \max\{score(i,j-1), score(i-1,j), \sigma(D_{i,j})\} \qquad [6]$$

where $D_{i,j}$ is any—say, the longest—diagonal ending at the point (i,j) that satisfies

$$\sigma(D_{i,j}) = \max\{\sigma(D) : D \text{ ends at the point } (i,j)\}, \qquad [7]$$

while *prec* may be defined by, say,

$$prec(i,j) := \begin{cases} prec(i,j-1) & \text{if } score(i,j) = score(i,j-1), \\ prec(i-1,j) & \text{if } score(i,j-1) < score(i,j) \\ & \quad = score(i-1,j), \\ D_{i,j} & \text{if } score(i,j-1), score(i-1,j) \\ & \quad < score(i,j) = \sigma(D_{i,j}). \end{cases} \qquad [8]$$

Straightforward dynamic programming can now be used to calculate all these values, which then allow us to find a maximum alignment by a simple backtracking process: We set $D_1 := \text{prec}(L_1, L_2)$ and $D_{i+1} := \pi(D_i)$ as long as $\pi(D_i)$ is defined.

Clearly, this algorithm works independently of how the weights of diagonals are defined and, hence, allows experimenting with different choices not only of T, but also with other ways of defining w (see, for instance, Section 5 below).

3. Multiple Alignment

In this section, we discuss an extension of the above described algorithm to N sequences, where $N > 2$. A direct extension of the algorithm would use N-dimensional diagonals—i.e., N-

tuples of segments instead of the pairs of segments we have used for the alignment of two sequences.

However, this would increase the computational complexity of the algorithm exponentially relative to the number of sequences to be aligned. Aside from these practical difficulties and, much more important, we do not want to restrict our attention to similarities that occur consistently in all of the N sequences. A motif occurring in two sequences should not be assumed *a priori* to occur in all the other sequences too.

We therefore prefer another approach and construct the multiple alignment from "two-dimensional" diagonals. As before, we try to select a consistent set of diagonals with a maximal sum of weights. However, now diagonals originate from all of the $\frac{1}{2}N(N-1)$ possible pairwise sequence comparisons.

We rely on evaluating our diagonals by their respective weights in contrast to the conventional alignment algorithms which maximize a sum of single matches, penalizing isolated matches only in terms of sophisticated gap penalties. Consequently, diagonals are sorted according to their weights irrespective of the initial pairwise sequence comparison they originate from.

To reduce the total number of diagonals under consideration, we may restrict our attention to only those diagonals which belong to one of the pairwise maximum alignments; however, this is not a prerequisite for the algorithm and is used only to enhance its efficiency. So, in a first step of our multiple alignment algorithm, all $\frac{1}{2}N(N-1)$ pairwise maximum alignments are constructed by the algorithm described in the previous section. The resulting set of diagonals is denoted by $\mathcal{D}$.

In a second step, the set $\mathcal{D}$ is sorted according to the weights of the diagonals in a *greedy* way. Diagonals are incorporated one by one into the multiple alignment starting with the diagonal of maximum weight, provided they are *consistent* with the diagonals already incorporated (our concept of *consistency* is discussed in Section 4). Diagonals *not* consistent with the growing set of consistent diagonals are rejected. The result of this selection process is a consistent set of diagonals—i.e., a multiple alignment A.

Diagonals with high weights are most likely to be biologically relevant, and, hence, are added first to the alignment. Thus, they establish a *frame* into which the diagonals with lower weights have to fit to be included into the multiple alignment.

Next, we complete the alignment A using an iterative procedure: Those parts of the sequences that are not yet aligned by the alignment A are realigned by the above described procedure, considering now of course only those diagonals which are consistent with the existing alignment A.

Again, the resulting diagonals are sorted according to their weights and included into the alignment A as described above. This procedure is repeated as often as possible—i.e., as long as there are any diagonals of positive weight which can be added to the existing multiple alignment A.

Obviously, many other variants of extending our pairwise alignment procedure to more than two sequences are imaginable, and they should be investigated in the future (see also below). Yet, at present, the variant we have described above appears to be fully satisfactory: it is efficient and it appears to produce biologically meaningful results.

4. Alignments and Consistency

One crucial concept of the multiple alignment algorithm described in the last section is our concept of *consistency*. We had to decide whether or not a diagonal is consistent with the diagonals already incorporated into the alignment. To define consistency properly, we have to introduce some mathematical formalism.

Consider a set of sequences $\{X^{(1)}, \ldots, X^{(N)}\}$ where $X^{(i)} = (X_1^{(i)}, \ldots, X_{L_i}^{(i)})$ is the ith sequence. The set

$$X := \{(i,l) | 1 \le i \le N, 1 \le l \le L_i\} \qquad [9]$$

of all positions of all sequences can be regarded as a partially ordered set $(X, \preceq)$: for any two positions x and y we define $x \preccurlyeq y$ if and only if (i) both positions belong to the same sequence and (ii) x precedes or equals y in the natural order of this sequence; that is, if and only if $x = (i,l)$ and $y = (i,k)$ for some $i \in \{1, \ldots, N\}$ and some $l,k \in \{1, \ldots, L_i\}$ with $l \le k$.

Obviously, any alignment can be regarded as an *equivalence relation* A on the set X: xAy means that position x is aligned (or coincides) with position y, though, of course, not every equivalence relation on the set X deserves to be called an *alignment*. An alignment A has to meet a certain *consistency* criterion. To define this criterion, we note that every relation R on the set X extends the relation "$\preceq$" to a (quasi-partial order) relation "$\preceq_R$" which is given by

$$\preceq_R := (R \cup \preceq)_t \qquad [10]$$

where for a relation S defined on X we denote by S_t its *transitive hull*, so two elements $x,y \in X$ are in relation S_t if and only if there exist elements $x_0, \ldots, x_k \in X$ with $x_0 = x$, $x_k = y$, and $X_{i-1} S x_i$ for all $i = 1, \ldots, k$.

Now we define: An equivalence relation A on the set X is called an *alignment* if all restrictions of the extended relation "$\preceq_A$" to the single sequences coincide with their respective natural order relation.

If A is an alignment of the partially ordered set X, then for every position $x \in X$ and every $i \in \{1, \ldots, N\}$, there is a lower bound $\underline{b}_A(x,i)$ and an upper bound $\bar{b}_A(x,i)$, so that x can be consistently aligned with the pair (i,k) if and only if

$$\underline{b}_A(x,i) \le k \le \bar{b}_A(x,i) \qquad [11]$$

holds [clearly, if $x = (i,l)$, for some $l \in \{1, \ldots, L_i\}$, then

$$\underline{b}_A(x,i) = \bar{b}_A(x,i) = l].$$

To decide whether a diagonal D is consistent with a given alignment A, we have to know the values $b_A(x,i)$ and $\bar{b}_A(x,i)$ for any $x \in X$ and $i \in \{1, \ldots, N\}$. So, for every diagonal that is consistent with A and therefore suitable to be included into the multiple alignment, we have to calculate the values $\underline{b}_{\tilde{A}}(x,i)$ and $\bar{b}_{\tilde{A}}(x,i)$ for the alignment $\tilde{A}$ generated by A and the newly included diagonal D.

Let D consist of the pairs $(x_1,y_1), \ldots, (x_k,y_k)$ with $x_1 \preceq \ldots \preceq x_k$ and $y_1 \preceq \ldots \preceq y_k$. If we define $c_A(x,i)$ to be 1 if xAy holds for some position y of the ith sequence and to be 0 otherwise, it can be shown that the values $\underline{b}_{\tilde{A}}(x,i)$ and $\bar{b}_{\tilde{A}}(x,i)$ are given as follows (with $p \in \{1, \ldots, k\}$ always assumed to be chosen appropriately):

$$\underline{b}_{\tilde{A}}(x,i) = \begin{cases} \max[\underline{b}_A(x,i), \underline{b}_A(y_p,i)], & \text{if } xAx_p, \\ \max[\underline{b}_A(x,i), \underline{b}_A(x_p,i)], & \text{if } xAy_p, \\ \max[\underline{b}_A(x,i), \underline{b}_A(y_p,i) + c_A(y_p,i)], & \text{if } x \succ_A x_p \wedge \\ & \quad x \not\succcurlyeq_A x_q \ \forall\, q > p, \\ \max[\underline{b}_A(x,i) \underline{b}_A(x_p,i) + c_A(x_p,i)], & \text{if } x \succ_A y_p \wedge \\ & \quad x \not\succcurlyeq_A y_q \ \forall\, q > p, \\ \underline{b}_A(x,i) & \text{else,} \end{cases}$$

$$\bar{b}_{\mathcal{A}}(x,i) = \begin{cases} \min[\bar{b}_A(x,i),\bar{b}_A(y_p,i)], & \text{if } xAx_p, \\ \min[\bar{b}_a(x,i),\bar{b}_A(x_p,i)], & \text{if } xAy_p, \\ \min[\bar{b}_A(x,i),\bar{b}_A(y_p,i) - c_A(y_p,i)], & \text{if } x \prec_A x_p \wedge \\ & x \not\preccurlyeq_A x_q \;\forall\, q > p, \\ \min[\bar{b}_A(x,i)\bar{b}_A(x_p,i) - c_A(x_p,i)], & \text{if } x \prec_A y_p \wedge \\ & x \not\preccurlyeq_A y_q \;\forall\, q < p, \\ \bar{b}_A(x,i) & \text{else,} \end{cases}$$

5. Refinements of the Alignment Algorithm

5.1. The weight-oriented sorting of diagonals prior to construction of a consistent set of diagonals can lead in some cases to "biologically wrong" alignments. Consider, for example, a set of sequences containing a common motif. A biologically meaningful alignment would certainly bring together all instances of the motif in one single column.

However, if the conservation of the motif is low, it might happen that the motif is found in some of the pairwise alignments, but in others it is displaced by a diagonal $\tilde{D}$ not consistent with the "correct" diagonals.

If we are lucky, the weight of $\tilde{D}$ is smaller than the weights of the correctly aligned motifs in the other sequence pairs. In this case, $\tilde{D}$ will be rejected in the selection process described in Section 3 and replaced in the next iteration step by a correct diagonal.

However, if $w(\tilde{D})$ is too high, $\tilde{D}$ will be incorporated into the multiple alignment, and some of the correct diagonals will be rejected instead.

Hence, to favor similarities occurring in more than two sequences, we may introduce a second weight function w^* on the set $\mathcal{D}$, which reflects not only the significance of a diagonal D, but also the significance of all diagonals that have any overlap with D. Such a weight function w^* can be calculated as follows.

Consider two diagonals D_l and D_m. If altogether three sequences $S^{(i)}$, $S^{(j)}$, $S^{(k)}$ are involved in the diagonals D_l and D_m with, say, $S^{(j)}$ belonging to both diagonals D_l and D_m and if D_l and D_m have a common region in sequence $S^{(j)}$, then the corresponding positions of the other two sequences are connected *indirectly* by D_l and D_m. This way, a segment of $S^{(i)}$ is connected with a segment of $S^{(k)}$—i.e., we have a third diagonal D_n belonging to the sequences $S^{(i)}$ and $S^{(j)}$. In this situation, we define

$$\tilde{w}(D_l,D_m) := w(D_n). \qquad [12]$$

If diagonals D_l and D_m have no overlap or if they are identical, we define

$$\tilde{w}(D_l,D_m) := 0. \qquad [13]$$

Now, for any diagonal D, we may define its *overlap weight* $w^*(D)$ by

$$w^*(D) := w(D) + \sum_{E \in \mathcal{D}} \tilde{w}(D,E). \qquad [14]$$

Here, we expect our sequence family to consist of an unbiased sequence sample. If instead our family is expected to contain large subfamilies of closely related sequences biasing the overlap weights toward favoring diagonals from those subfamilies, one may use any of the correction methods discussed in the literature (10) to counterbalance this bias.

Anyway, instead of sorting diagonals according to their weights w before starting the selection process, we may now sort them according to their—uncorrected or corrected—overlap weights.

5.2. A further modification concerns the weight function w if protein sequences are to be aligned. Amino acids show different degrees of similarity, which necessitates the employment of more sophisticated similarity matrices.

To assess the significance of a diagonal D of length l, we may consider the sum s of the similarity values of the residue pairs aligned by D instead of the number of matches. We may then calculate the probability of any diagonal of length l to have at least the respective sum of similarity values, and continue as described above.

We believe that developing similarity matrices from sequence statistics directly related to our approach will further improve the performance of our algorithm, and we plan to develop such statistics in the near future (S. Perrey, J. Stoye, and A.D., unpublished data).

5.3. A similar modification of the weight function w can be applied if DNA sequences are to be aligned which are supposed to contain protein coding regions. In this case, it may be advisable to consider only segments whose length can be divided by three, and to translate them into peptides before comparing them. The weight of a DNA diagonal is then calculated as the weight of the corresponding peptide diagonal using the procedure described just above. As is shown in the next section, one may even identify reading frames using this approach.

6. Examples

6.1. DNA Alignment. We have applied our alignment method to a set of 11 genomic DNA sequences coding for basic helix-loop-helix DNA binding proteins reported by Dang *et al.* (11) (GenBank accession nos. X57435, X16106, S77532, M24405, X51990, Y00396, X55666, M33620, L12469, X51330, and X03719).

These proteins have a DNA binding site of about 30 amino acids which has been detected based on experimental evidence. They appear to be mostly unrelated outside this region, prohibiting detection of the binding site by common alignment algorithms.

We applied our alignment program directly to the genomic DNA sequences of those proteins. The lengths of the DNA sequences vary between 960 and 7225 bp and all sequences contain introns.

Diagonals for our alignment were calculated by systematic translation of the nucleic acid sequences using all possible reading frames as described in Section 5.3. The threshold T for the weights of the diagonals has been set to $T := 16$. This value proved to be useful for suppression of diagonals in random sequences of comparable length. Twenty of the 55 initial pairwise alignments correctly aligned the functional site of the sequences.

Diagonals produced by the pairwise alignments were sorted according to their (uncorrected) overlap weights w^* as described in Section 5.1. *All* of the 20 correct diagonals were incorporated into the resulting multiple alignment. Since these 20 correct diagonals connected (directly or indirectly) *all* of the 11 DNA binding sites with each other, one single column containing all 11 DNA binding sites emerged in the first iteration step of the alignment.

Moreover, the region where all of the 11 sequences were aligned at a time, was clearly defined. This region coincides with the functional site described in ref. 11.

As we had translated *all* "DNA diagonals" of the dot matrix into "peptide diagonals," all combinations of reading frames were possible. It is remarkable, that all of the 20 diagonals which connected the functional sites of the sequences were in the correct reading frame, though we did not presuppose any

Table 1. Alignment of 11 DNA sequences coding for basic helix-loop-helix proteins

Program (ref.)	Correctly aligned functional sites	Length restrictions, bp
DFALIGN (12)	0	2000
PILEUP (13)	2	2000
CLUSTAL (14)	2	None
GENALIGN (15)	0	None
DIALIGN	11	None

Comparison of our diagonal-alignment (DIALIGN) method with other DNA alignment programs. We have aligned a set of 11 genomic DNA sequences coding for basic helix-loop-helix proteins. The lengths of the DNA sequences vary between 960 and 7225 bp. Two programs were not able to align sequences of this length at all. In this case, we shortened the sequences to a length of 2000 bp by cutting off regions outside the functional site to allow comparison.

information concerning the (known) reading frames in our alignment.

Since the threshold T is the most crucial parameter of our algorithm, we tried to find out how the resulting alignments were influenced by the value of T. We therefore aligned the 11 basic helix-loop-helix sequences using different values for T. For $15.7 \leq T \leq 19.8$, the results were as described above—i.e., all 11 functional sites were (eventually) aligned correctly. With $T = 14.5$, as well as with $T = 21.5$, still nine sites were correctly aligned.

We have also applied four widely used alignment programs to the above set of sequences. None of them appeared to be capable of producing similarly satisfying results (see Table 1).

6.2. Protein Alignment. Recently McClure *et al.* (8) have published a systematic comparison of twelve commonly used protein alignment programs. We have applied our program to the sequences used in this study and found it among the three best scoring programs - together with the programs DFALIGN and CLUSTAL V. But in contrast to the other programs, our program is independent of user-defined parameters if protein sequences are to be aligned. The results of the three best scoring programs are shown in Table 2.

As a further test example, we used protein sequences from the Brookhaven Protein Data Bank. We compared our alignments with structure alignments performed by Lessel and Schomburg (16), which we used as "standard of truth." Though we used no information about the three-dimensional structures of the sequences, our algorithm was capable of correctly aligning regions of structural similarity (unpublished data).

Conclusions

If sequences show only limited regions of similarity, our algorithm aligns only those regions. In other words, it is a "local" alignment algorithm in the sense of Smith and Waterman (17). Yet, whereas the Smith–Waterman algorithm finds only the one single best local alignment, our algorithm can include several regions of similarity into the alignment (consistency provided). This is an important feature if, for example, coding regions of DNA sequences are separated by introns.

In accordance with other authors (12, 18, 19), we construct multiple alignments using iterative pairwise alignments to keep the computational complexity of the algorithm low.

Yet, in contrast to these approaches, pairwise alignments are not crucial for our procedure, since they are only used for preselection of diagonals to be included into the multiple alignment. In particular, our algorithm is independent of the order in which the pairwise alignments are constructed.

Assembling the multiple alignment from two-dimensional diagonals necessarily sacrifices information that could be obtained from higher dimensional ones. By introducing *overlap weights*, we favor those two-dimensional diagonals that are "projections" of a high scoring multidimensional diagonal. However, nonconsistent random overlaps of diagonals may increase overlap weights as well. Methods using consistent multidimensional overlaps, therefore, are desirable and should be developed in the future.

Our algorithm shows only little dependence on user-defined parameters. Because gaps are not treated explicitly, we avoid the well known difficulties of how to choose proper gap penalties (20). Restrictions of the length of the diagonals speed up the algorithm, but have no essential influence on the resulting alignments. The threshold T, on the other hand, does influence the results. However, since the purpose of T is to keep the number of random diagonals low, appropriate values for T can be obtained from alignments of random sequences. Moreover, in most of our test examples, we have found that T can be varied considerably without changing the results in an essential way and T is not at all required for alignments of short DNA sequences or protein sequences.

The examples demonstrate the capability of our algorithm to find functional sites even if the sequences under consideration

Table 2. Alignment of ribonuclease H sequences

Program and no. of sequences	Motif I	Motif II	Motif III	Motif IV	Parameters/comments
CLUSTAL V					
12	100	75	75	75	Defaults; parameters tweaked are pairwise:
10	100	70	70	80	indel: (1–8) and k-tuple (1–2); multiple
6	100	67	50	50	alignment: I (6–12) and E (2–10)
DFALIGN					
12	100	100	83	100	Begin weighting sequence 3 with value 3
10	100	60	70	100	Begin weighting sequence 4 with value 3
6	100	100	67	100	Begin weighting sequence 2 with value 2
DIALIGN					
12	92	83	92	92	No parameters
10	100	80	90	100	No parameters
6	100	100	83	100	No parameters

Comparison of the diagonal-alignment (DIALIGN) method with other protein-alignment programs. We used ribonuclease H sequences as test sequences and compared the alignments produced by our method with the results of other alignment methods as reported in McClure et al. (8). The results of the three best scoring programs are shown above. As in ref. 8, we aligned a set of 12 sequences, as well as subsets of 10 and 6 sequences. The sequences contain four motifs. The entries in the respective columns denote the percentage of correctly aligned motifs. In contrast to other methods, our program is independent of user-defined parameters if protein sequences are to be aligned. The results of the programs CLUSTAL V and DFALIGN are cited from McClure et al. (8).

only exhibit a low overall similarity. Therefore, our method seems to be an appropriate and sensitive tool to detect local functional similarities in sets of relative large sequences, and we believe that it justifies further efforts to improve its performance and to explore its potential. In addition, it should also be worthwhile to compare our approach with similarly structured proposals as suggested for instance by Taylor (9) or V. Brendel (personal communication).

We want to thank Kornelie Frech and Kerstin Quandt for continuous support during the program development. We want to thank both and Sören Perrey for critically reading the manuscript and for many helpful comments. Part of this work was supported by ring funding project GENUS 413-4001-01 IB 306 D (Förderschwerpunkt Bioinformatik) of the German Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie.

1. Needleman, S. B. & Wunsch, C. D. (1970) *J. Mol. Biol.* **48,** 443–458.
2. Murata, M., Richardson, J. S. & Sussman, J. L. (1985) *Proc. Natl. Acad. Sci. USA* **82,** 3073–3077.
3. Carrillo, H. & Lipman, D. L. (1988) *SIAM J. Appl. Math.* **48,** 1073–1082.
4. Argos, P. (1987) *J. Mol. Biol.* **193,** 385–396.
5. Argos, P. & Vingron, M. (1990) *Methods Enzymol.* **183,** 352–365.
6. Waterman, M. S. & Jones, R. (1990) *Methods Enzymol.* **183,** 221–237.
7. Johnson, M. S. & Doolittle, R. F. (1986) *J. Mol. Evol.* **23,** 267–278.
8. McClure, M. A., Vasi, T. K. & Fitch, W. M. (1994) *Mol. Biol. Evol.* **11,** 571–592.
9. Taylor, W. P. (1994) *J. Comp. Biol.* **1,** 297–310.
10. Vingron, M. & Sibbald, P. R. (1993) *Proc. Natl. Acad. Sci. USA* **90,** 8777–8781.
11. Dang, C. V., Dolde, C., Gillison, M. L. & Kato, G. J. (1992) *Proc. Natl. Acad. Sci. USA* **89,** 599–602.
12. Feng, D. F. & Doolittle, R. F. (1987) *J. Mol. Evol.* **25,** 351–360.
13. Higgins, D. G. & Sharp, P. M. (1989) *Comput. Appl. Biosci.* **5,** 151–153.
14. Higgins, D. G., Bleasby, A. J. & Fuchs, R. (1992) *Comput. Appl. Biosci.* **8,** 189–191.
15. Martinez, H. M. (1988) *Nucleic Acids Res.* **16,** 1683–1691.
16. Lessel, U. & Schomburg, D. (1994) *Protein Eng.* **7,** 1175–1187.
17. Smith, T. F. & Waterman, M. S. (1981) *Adv. Appl. Math.* **2,** 482–489.
18. Altschul, S. F. & Lipman, D. L. (1989) *SIAM J. Appl. Math.* **49,** 197–209.
19. Gotoh, O. (1993) *Comp. Appl. Biol. Sci.* **9,** 361–370.
20. Vingron, M. & Waterman, M. S. (1994) *J. Mol. Biol.* **235,** 1–12.