

From Analysis of Protein Structural Alignments Toward a Novel Approach to Align Protein Sequences

Shamil R. Sunyaev,^{1,2} Gennady A. Bogopolsky,¹ Natalia V. Oleynikova,^{3,4} Peter K. Vlasov,^{1,3} Alexei V. Finkelstein,⁵ Mikhail A. Roytberg^{4*}

¹*Institute of Molecular Biology, Russian Academy of Sciences, Moscow, Russia*

²*European Molecular Biology Laboratory (EMBL), Heidelberg, Germany*

³*Moscow Institute of Physics and Technology, Dolgoprudny, Moscow Region, Russia*

⁴*Institute of Mathematical Problems in Biology, Russian Academy of Sciences, Pushchino, Moscow Region, Russia*

⁵*Institute of Protein Research, Russian Academy of Sciences, Pushchino, Moscow Region, Russia*

ABSTRACT Alignment of protein sequences is a key step in most computational methods for prediction of protein function and homology-based modeling of three-dimensional (3D)-structure. We investigated correspondence between “gold standard” alignments of 3D protein structures and the sequence alignments produced by the Smith–Waterman algorithm, currently the most sensitive method for pair-wise alignment of sequences. The results of this analysis enabled development of a novel method to align a pair of protein sequences. The comparison of the Smith–Waterman and structure alignments focused on their inner structure and especially on the continuous ungapped alignment segments, “islands” between gaps. Approximately one third of the islands in the gold standard alignments have negative or low positive score, and their recognition is below the sensitivity limit of the Smith–Waterman algorithm. From the alignment accuracy perspective, the time spent by the algorithm while working in these unalignable regions is unnecessary. We considered features of the standard similarity scoring function responsible for this phenomenon and suggested an alternative hierarchical algorithm, which explicitly addresses high scoring regions. This algorithm is considerably faster than the Smith–Waterman algorithm, whereas resulting alignments are in average of the same quality with respect to the gold standard. This finding shows that the decrease of alignment accuracy is not necessarily a price for the computational efficiency. *Proteins* 2004; 54:569–582. © 2003 Wiley-Liss, Inc.

Key words: sequence alignment; 3D structure alignment; substitution score matrix

INTRODUCTION

Alignment of two protein sequences is an old and probably the most classic problem in computational biology.^{1–4} Direct alignment of three-dimensional (3D) structures is now also possible, although in a more limited number of cases.^{5–10} Sequence alignment is the core of numerous applications in sequence analysis (e.g., in functional annotation of genes and proteins,¹¹ in protein domain analysis,¹² and in homology modeling of protein

3D structure).¹³ Many sophisticated computational methods in molecular biology (e.g., multiple alignments,^{14,15} profile analysis,¹⁶ and threading¹⁷) use a pair-wise sequence alignment as a subprocedure.

Ideally, an algorithmically produced alignment of two protein sequences coincides with their evolutionary alignment. The latter alignment can be treated as one reproducing the result of the evolutionary history of homologous protein sequences (i.e., the aligned sites of given proteins are assumed to correspond to the same site of their common ancestor).¹⁸ Although the true evolutionary alignment is always unknown, an accurate alignment of 3D structures can serve as its reasonable approximation because strong stabilizing selection acts against structural changes, so that protein structural features remain constant, whereas amino acid sequences diverge.¹⁹ Therefore, alignments based on superposition of 3D structures can serve as the “gold standard” (GS) for pair-wise sequence alignments.

The aim of this study was to reveal features of structural alignments, which can be used to construct more efficient and/or accurate alignment algorithm than classic Smith–Waterman (SW) algorithm.² The alignment method, which is a result from the study, is presented in Results and Materials and Methods.

Abbreviations used: 3D, three-dimensional structure; GS, gold standard alignment of 3D structures; SW, Smith–Waterman alignment of sequences; HSP, high scoring pair; SDP, sparse dynamic programming; FSSP, fold classification based on structure–structure alignment of proteins.

Grant sponsor: INTAS; Grant number: 99-01476; Grant sponsor: Netherlands Organization for Scientific Research (NWO); Grant sponsor: RBRF; Grant numbers: 01-04-48400, 02-07-90412, and 03-04-49369; Grant sponsor: Russian Ministry of Science; Grant sponsor: International Research Scholar's Award to A.V.F. from the Howard Hughes Medical Institute.

S.R. Sunyaev's present address is Genetics Division, Department of Medicine, Brigham and Women's Hospital and Harvard Medical School, Thorn 914, 20 Shattuck Street, Boston, MA 02115.

*Correspondence to: Mikhail A. Roytberg, Computational Biology Group, Institute of Mathematical Problems in Biology, Russian Academy of Sciences, 142290, Pushchino, Moscow Region, Russia. E-mail: royberg@impb.psn.ru

Received 24 December 2002; Accepted 28 April 2003

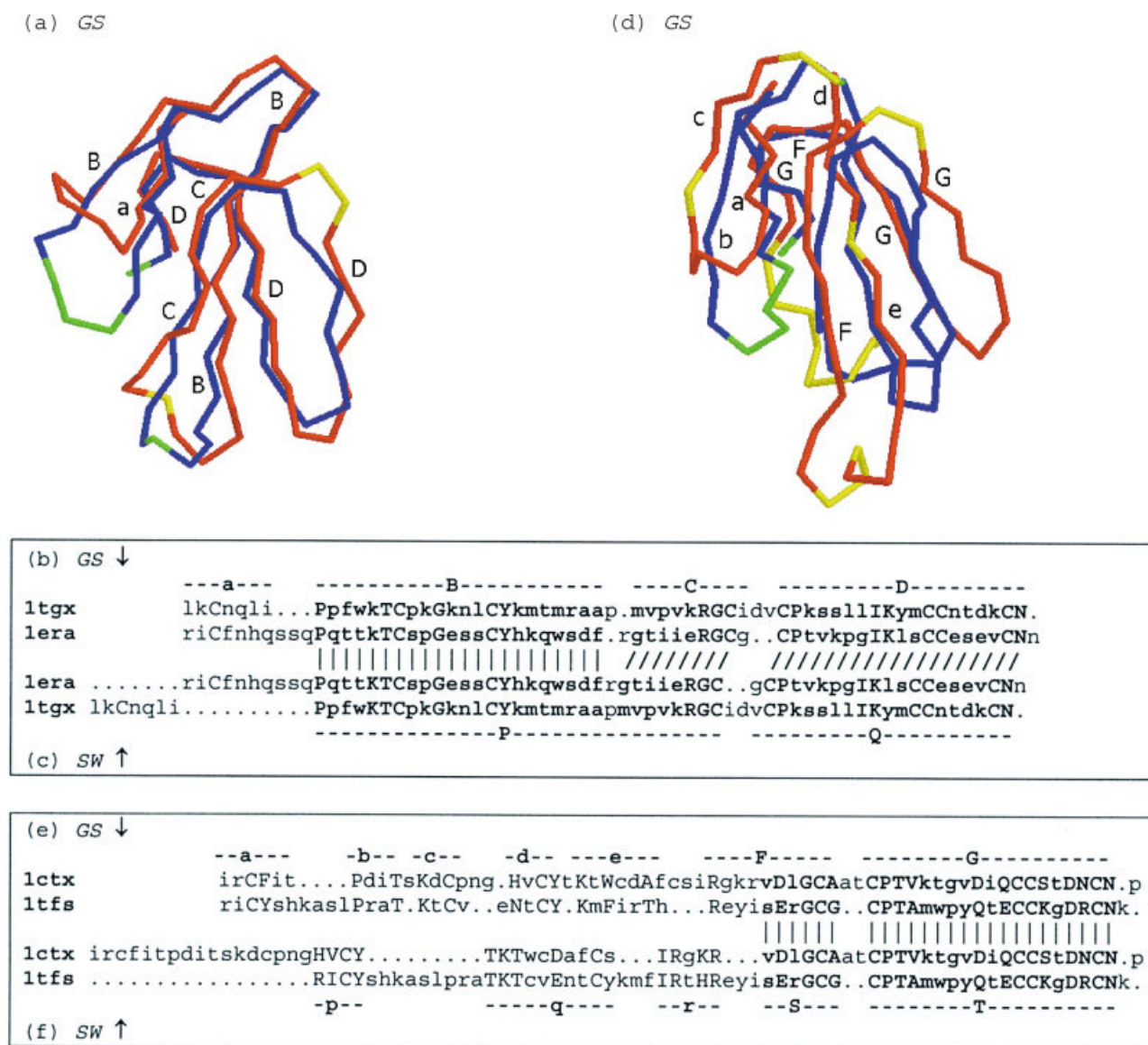


Fig. 1. Comparison of structural gold standard (GS) and Smith–Waterman (SW) alignments. **a–c**: Alignments of cardiotoxin (PDB code: 1ttx) versus erabutoxin B (PDB code: 1era). (a) Structural alignment in 3D space. Aligned regions are colored in red in 1ttx and in blue in 1era, nonaligned regions are in yellow in 1ttx and in green in 1era. Deviations in C_{α} positions of aligned residues are usually within 3 Å. However, at the ends of aligned regions, some deviations can be as great as 9 Å. (b) GS alignment of 1ttx and 1era sequences, that is, the sequence alignment induced by the above alignment of 3D structures. (c) SW sequence alignment. Capital characters in (b) and (c) show the exact matches. The GS alignment has 57 pairs of superimposed positions, forming four islands. These islands are marked with a, B, C, and D in (a) and with –a–, –B–, –C–, and –D– in (b) (the small letter refers to the GS island which has no analog in SW alignment); the islands are separated with gaps. The SW alignment has 51 pairs of superimposed positions, forming two islands (marked with –P– and –Q–). Short lines between the GS and SW alignments connect sites identically superimposed in both alignments. These 49 sites are shown in bold characters. Accuracy of the SW alignment of 1ttx and 1era, compared to the GS alignment, is $49/57 \approx 85.9\%$. Here 57 is the number of superimposed positions in the GS alignment, and 49 of them are equally superimposed in the SW alignment. Confidence of the SW alignment of 1ttx and 1era, is $49/51 \approx 96.1\%$. **d–f**: Alignments of α -cobratoxin (PDB code: 1ctx) versus toxin FS2 (PDB code: 1tfs). Here the similarity between the proteins, and therefore the similarity between their GS and SW alignments, is much lower than that in previous example. Notation is analogous to that in panels (a)–(c).

In the analytical part of the study, we compared alignments produced by the well-established SW method with GS structural alignments (Fig. 1). The set of GS alignments consisted of nearly 600 protein domain pairs. The GS alignments were extracted from multiple 3D structure alignments of protein domains, given in BALiBase²⁰ (www-igbmc.u-strasbg.fr/BioInfo/BALiBASE2/). The results of our analysis have been further verified by using the FSSP

database (www2.ebi.ac.uk/dali/fssp/fssp.html) to eliminate a possible data set-related bias. The average level of similarity between algorithmic and structural alignments are considered below as a measure of accuracy of a sequence-aligning algorithm.^{21–23} Accuracy of the sequence-aligning algorithm is critical for homology-based modeling of 3D structures and for other applications.

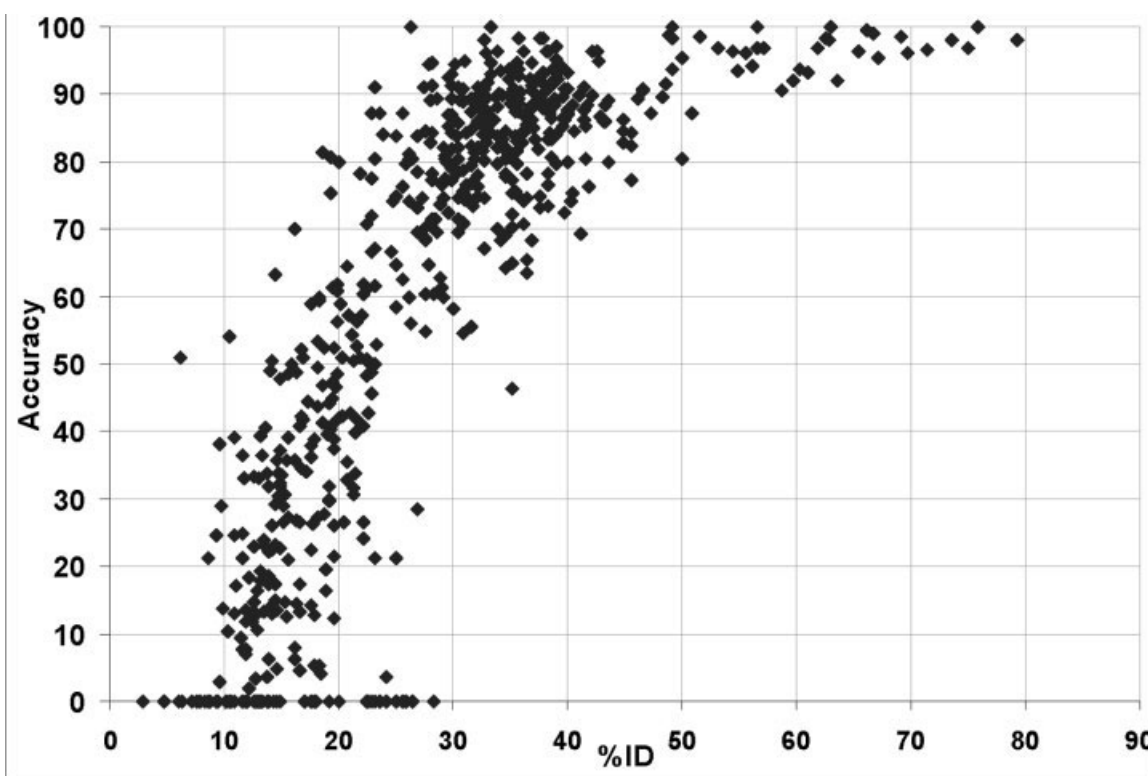


Fig. 2. Accuracy of 583 pair-wise SW alignments with respect to the structural GS. Each point of the scatter plot corresponds to one pair of proteins. X-axis (%ID) shows the identity of aligned sequences, computed as the number of identical residue matches in the GS alignment, divided by the length of the shorter sequence. Y-axis (Accuracy) shows the accuracy of the SW alignment defined by Eq. 1. All SW calculations in this and other figures and tables have been performed by using Gonnet250²⁹ substitution matrix. The scatter plot presents the results obtained for gap penalties in linear domain (10, 0.5). Log domain penalties give qualitatively similar plot (see Materials and Methods).

Following earlier works by other groups,^{21,22} we define alignment accuracy as the number I of positions *Identically* superimposed in the algorithmic and the GS alignment divided by the total number G of positions in the GS alignment:

$$\text{Alignment_accuracy} = I/G \quad (1)$$

It is known that alignment accuracy strongly depends on identity of aligned sequences (defined here as the number of exact residue matches in GS alignment divided by the length of the shorter sequence). Figure 2 (analogous to Fig. 3 in Ref. 21) shows this dependence for our data set. One can see that if sequence identity exceeds 31–40%, Smith–Waterman alignment is almost precise in the vast majority of cases. This sequence identity range approximately agrees with the range where homology can be predicted with a high degree of confidence from the sequence information alone.^{24–26} If identity of two sequences is <10%, the SW algorithm is unable to reconstruct the GS alignment even in its small part. However, in the region of 10–30% sequence identity, the Smith–Waterman alignments show a very wide range of accuracy values; sequence pairs with the same identity level display very different “alignabilities.” This finding suggests that in this “twilight zone” alignment, accuracy depends on internal properties of GS alignments to be reconstructed, that is, not only on the

sequence identity of the compared proteins, but presumably on distribution of similar and dissimilar sequence regions along the GS alignment. The analysis of internal structure of GS alignments is essential for understanding of the underlying evolutionary process and for the development of new alignment methodologies. Properties of GS alignments were previously out of the focus of studies aiming at development of novel sequence comparison techniques. Thus, we investigate the internal structure of both structural and Smith–Waterman alignments, statistical properties of their ungapped segments (“islands” between two gaps) and high and low scoring fragments of these islands. (Note that the metaphor island was used differently in Altschul et al., Nucl. Acid Res. 2001;29:351–361, namely, to describe some 2D areas in the alignment graph.)

The algorithm based on the results of this investigation explicitly identifies regions of sufficiently high similarity (anchors), finds the optimal pathway through these anchors, and then specifies alignment in the regions between the anchors. The idea to start the alignment procedure from the search for high-similarity ungapped regions is definitely not new and was implemented in several software tools (e.g., BLAST [<http://www.ncbi.nlm.nih.gov/BLAST/>], FASTA [<http://fasta.bioch.virginia.edu/fasta/>]). However, this idea was considered as a way to increase

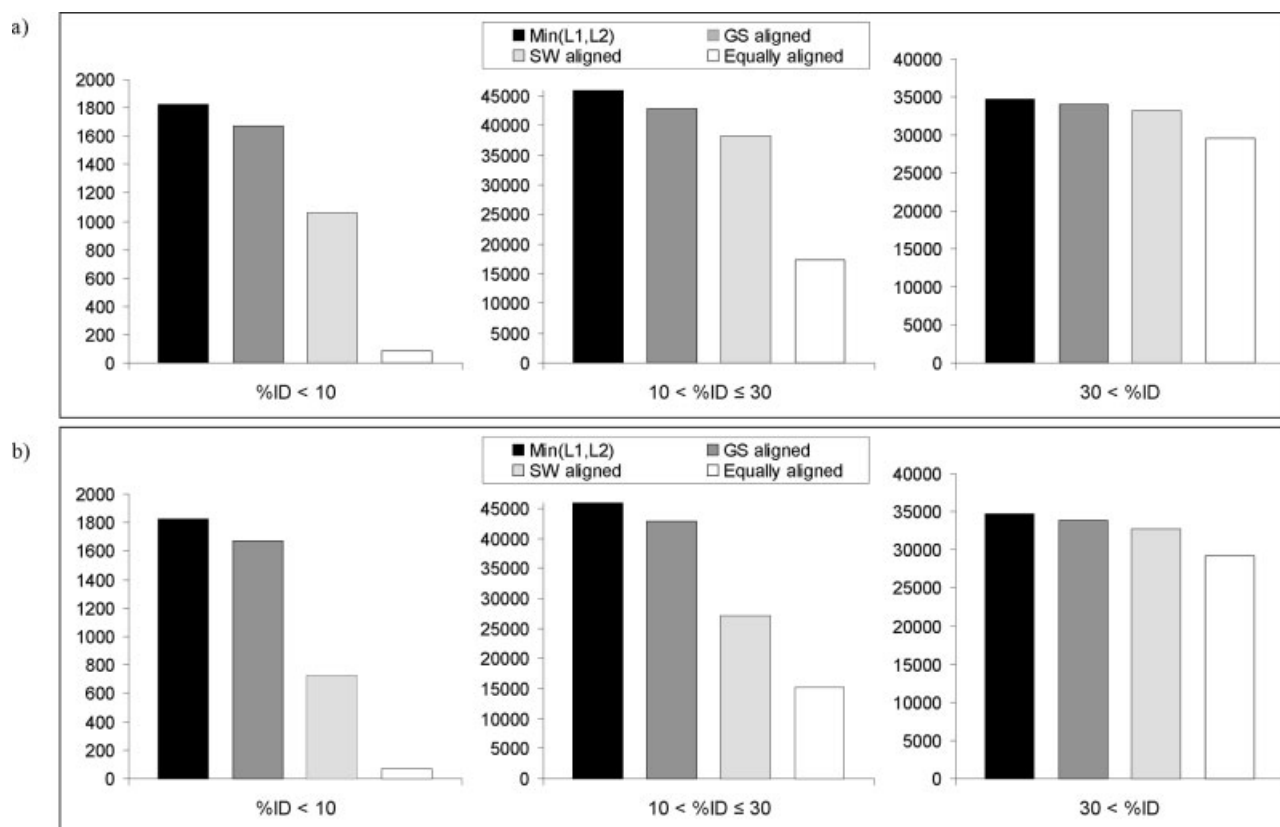


Fig. 3. Correspondence between GS and SW alignments for protein pairs of different sequence identity. The two gap parameter settings (see Materials and Methods) have been used for the SW alignments. **a:** Linear domain setting: gap opening penalty 10, gap extension penalty 0.5. **b:** Log domain setting: gap opening 15, gap extension 1. Columns represent four total characteristics of the aligned protein pairs. Black column: maximal possible number of aligned positions, that is, total sum of $L = \min(L_1, L_2)$, where L_1 and L_2 are the lengths of individual sequences. Dark-gray column: number of aligned positions in the GS alignments. Light-gray column: number of aligned positions in the SW alignments. White column: number of positions, identically aligned in the GS and SW alignments. The data are given separately for three different ranges of sequence identity (<10%, between 10 and 30%, >30%).

computational speed of alignment techniques inevitably associated with the loss of alignment accuracy. Our observations suggest the way to improve computational speed without sacrificing alignment accuracy and confidence compared to the SW algorithm.

MATERIALS AND METHODS

GS alignments were obtained from the BALiBase (Release 2) database²⁰ of multiple alignments. Only 3D structure-based alignments of homologous proteins were considered. This resulted in the data set of 583 pair-wise sequence alignments.

Smith–Waterman alignments were produced by the standard routine (C code was obtained from [http://fasta.bioch.virginia.edu/pub/fasta/], compiled for a Windows NT PC).

The following parameters were found to give the most accurate alignment (i.e., the highest average coverage accuracy): substitution matrix Gonnet250 (this is in agreement with Refs. 21 and 25), although the difference with the results obtained with BLOSUM62 and several other matrices is relatively small). Two sets of gap opening and extension penalties are used. The values of 10 (opening)

and 0.5 (extension) penalties were found to give the highest alignment accuracy. This is in good agreement with Refs. 21 and 42. However, it is important to note that these gap penalty values correspond to the linear domain²⁷; therefore, they are not applicable for domain identification in multidomain proteins and for database homology searches. To ensure this does not affect the results of the presented study, we have recomputed all alignments with the logarithmic²⁷ domain penalties 15 (opening) and 1 (extension) used in database searches. In Figures 3(a), (b) and 6 (see Results and Discussion) we present results corresponding to both linear and logarithmic domain alignments.

The program implementation of the newly developed algorithm (see Results and Appendix) available at “ftp://genetics.bwh.harvard.edu/Sunyaev/saadi/” or by request from the authors.

RESULTS

In the first part of this section, we present results of the comparative analysis of GS and SW alignments. In the second part, we describe and test a novel alignment algorithm based on the results of the analysis.

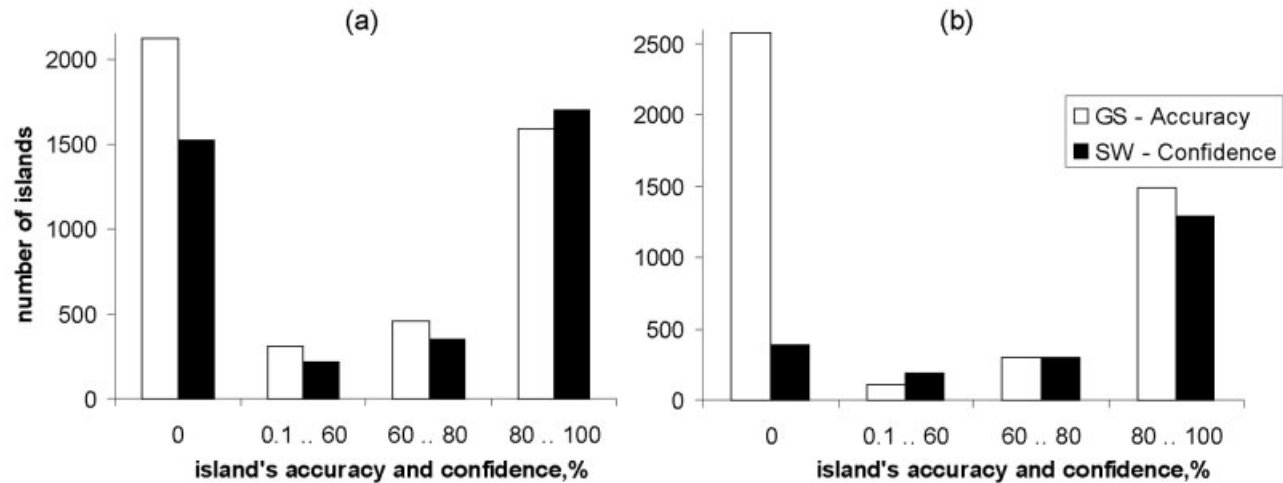


Fig. 4. Number of islands of a given accuracy and confidence. White bars correspond to the alignment accuracy of GS islands, and black bars correspond to the alignment confidence of algorithmically determined islands. We define the accuracy and the confidence of islands analogously to Eqs. 1 and 2. The accuracy of GS island is defined as the number of residues of the GS island, correctly aligned by the algorithm, and normalized by the length of the GS island. Confidence of an algorithmically determined island is defined as the number of residues of the island, correctly (as in the GS alignment) reconstructed by the algorithm, and normalized by the length of the island in the algorithmic alignment. **a,b:** Linear and logarithmic gap penalty domains, respectively (see Materials and Methods).

Analysis of the GS and SW Alignments

The scatter plot presented in Figure 2 poses the following questions.

1. What part of the typical algorithmic alignment is correct (i.e., is the typical sequence alignment much longer than its correct fraction)?
2. How are correctly aligned positions distributed along the alignment?
3. Are there some confident parts of the alignment even if it is mostly wrong?

It is obvious (e.g., cf. Sauder et al.²⁸) that the answer to the first question strongly depends on the choice of gap penalties, especially on the choice between linear and logarithmic gap penalty domain²⁷ [Figs. 3(a) and (b)]. The increase in gap penalties slightly reduces the total length of the correct part of the alignment, substantially reduces its incorrect part, and thereby strongly increases its confidence.

Formally, the notion of alignment confidence can be defined analogously to the notion of accuracy (Eq. 1):

$$\text{Alignment_confidence} = I/A \quad (2)$$

where I is again the number of residues *Identically* aligned in algorithmic and GS alignments and A is the total number of aligned positions in the *Algorithmic* alignment. For example, in Figure 1(c) confidence = $49/51 \approx 96.1\%$. In terms of Sauder et al.,²⁸ alignment_identity corresponds to the f_D measure of alignment quality, and alignment_confidence corresponds to the f_M measure.

To answer questions 2 and 3, we have to investigate the inner structure of structural and algorithmic alignments. It is natural to study the distribution of correctly aligned positions with respect to ungapped segments of GS and

SW alignments. Any sequence alignment can be presented as a chain of alternating gaps and ungapped segments of superimposed (aligned) amino acid residues. The latter form “islands” between the gaps [see Figs. 1(b) and (c)]. The goal of a sequence alignment algorithm is to single out and to superimpose the ungapped segments of both sequences to achieve the maximal possible score, taking into account both the substitution scores and the gap penalties.

As seen in Figure 4, a GS island can be either almost perfectly recognized by the SW algorithm or almost completely lost, whereas partial recognition of GS islands is infrequent. Similarly, semicorrect SW islands comprise a minor part of all SW islands. Further analysis (results are not shown) suggests that this is essentially true for all ranges of protein sequence identity. However, the all-or-nothing situation is not valid for the overall alignment accuracy and confidence. For example, in linear parameter set (cf. Fig. 2), there are 69 SW alignments with alignment accuracy 0% and 249 alignments with accuracy 80% or higher (229 of them correspond to protein identity > 30%), whereas 264 alignments have accuracy between 0% and 80%.

The above results show that alignment accuracy for a given protein pair can be described in terms of “lost” (having nothing in common with GS alignment) and “found” (having at least one correctly aligned position) islands. The accuracy is determined by two factors: properties of islands in the GS alignment of proteins and by the scores of possible sequence alignments paths alternative to the GS islands.²⁹ As usual, we define the total substitution score (or simply score) of an island as

$$\text{Score} = \sum s(a_i, b_i), \quad (3)$$

where summation is carried out over alignment positions; a_i and b_i form i th pair of aligned amino acids; $s(a_i, b_i)$ is the

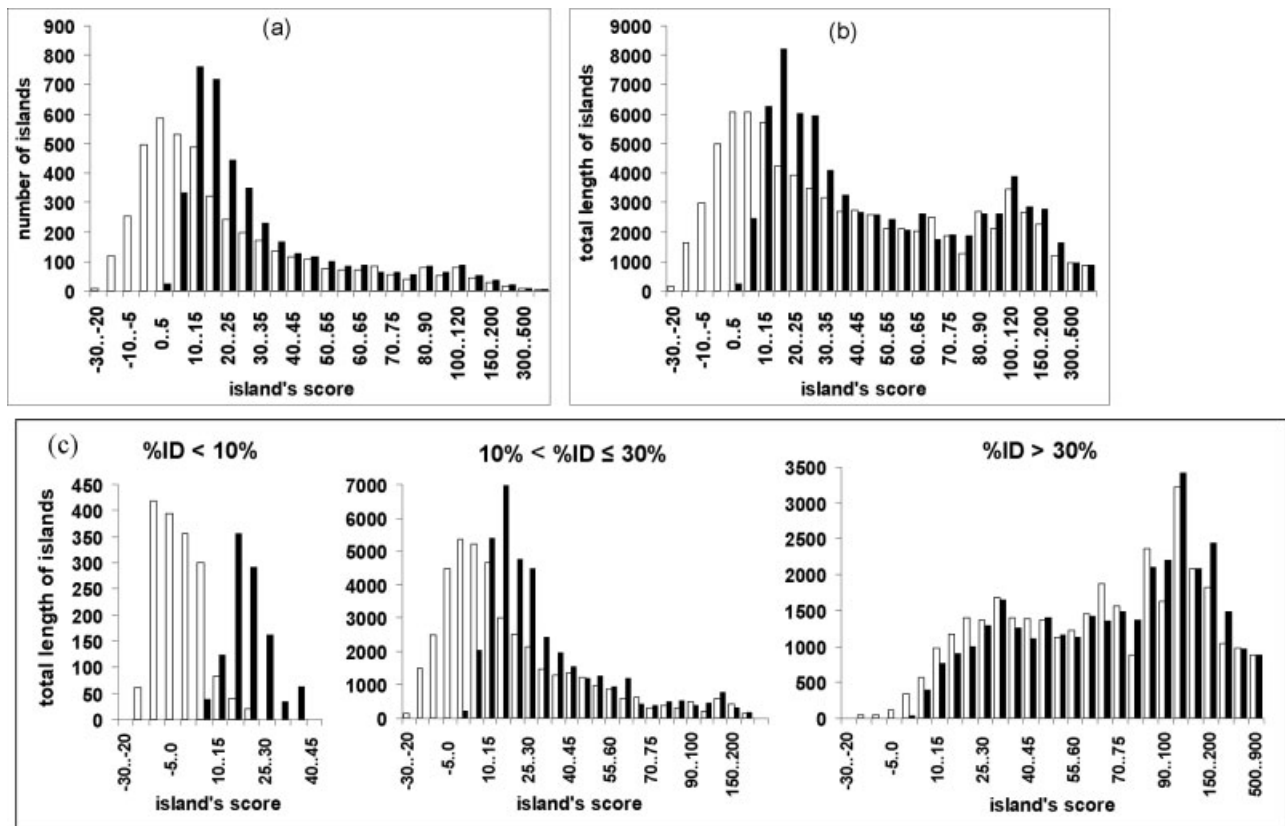


Fig. 5. **a:** Number of islands in the GS alignments (white) and in the Smith–Waterman alignments (black), having their scores within a given range. **b:** Total lengths of the islands having their scores within a given range. For example, 588 of the GS islands have their scores between 0 and 5, and the total length of these islands is 6077 amino acids. In contrast, only 22 of SW islands with total length of 232 amino acids have their scores in this range. The SW alignments are obtained with the linear domain setting (see Materials and Methods). The results for log domain setting are essentially the same. **c:** The same as (b) but for different ranges of protein sequence identity separately.

substitution score for this amino acid pair. If the score of a GS island is low, it has a poor chance to be restored by the Smith–Waterman algorithm with given scoring matrix. The inability of the algorithm to restore such an island can follow for two reasons. If the gap penalty is relatively high, it is not advantageous for the algorithm to introduce separate gap(s) to capture this island [see island “a” in Fig. 1(b)]. If the gap penalty is low, an alternative higher-scored way of alignment might exist because of purely statistical reasons [Figs. 1(e) and (f)]. With a given gap penalty, the first case is typical of the alignment of sufficiently close homologs. The second case is typical of the alignment of remote homologs.

Histograms of scores of islands in the structural GS alignments and in the Smith–Waterman sequence alignments are presented in Figure 5(a). They show a surprisingly large number of the GS islands of low and moreover negative score. In contrast, the algorithmic SW alignments almost do not contain low scoring islands. Note in Figure 5(b) that the total length and number of weak islands in the gold standard alignments are large. GS islands scoring <5 comprise 32% of all islands in the BaliBase database and cover 20% of all GS alignments. These weak islands have almost no chance to be restored by any sequence

alignment algorithm based on a given substitution-scoring matrix.

Table I shows that the lost and found islands are distinguished mostly by their score rather than by their length.

The last question posed in the beginning of this section is answered in Figure 6. Even if the alignment corresponds to the twilight zone of Figure 2 (pairs in 10–30% sequence identity range), it is possible to assess the confidence level to alignment islands, although the levels are slightly different for different sets of parameters. For the twilight zone alignments, in linear domain, only 4% of SW islands with the score > 40 have nothing in common with the GS alignment; only 15% of them contain less than two thirds of correctly aligned positions. In logarithmic domain, 9% of SW islands with a score > 40 have nothing in common with GS alignment; 25% of them contain less than two thirds of correctly aligned positions.

Island’s confidence correlates with the overall confidence of the SW alignment (see Table II, data shown for the linear domain). Four hundred eleven SW alignments have at least one island with the score exceeding 40. Three hundred thirty-eight of them have alignment confidence > 60%, and 73 alignments have alignment confidence < 60%.

TABLE I. Numbers of Lost and Found Islands Depending on Their Score and Length

	1–5		6–10		11–15		16–20		21–25		26–30		31–35		Island
															Length
30 .. 35			23	0	48	0	40	0	19	0	8	1	13	0	
25 .. 30	2	0	29	1	55	2	46	5	18	2	11	1	13	2	
20 .. 25	17	2	46	3	54	10	36	4	26	5	14	3	7	1	
15 .. 20	20	7	89	22	60	22	39	6	18	4	9	6	3	3	
10 .. 15	20	64	89	69	57	50	50	24	11	13	12	2	3	3	
5 .. 10	15	79	46	138	34	99	13	43	11	15	7	8	4	1	
0 .. 5	6	92	18	217	13	129	14	45	0	16	1	9	0	2	
–5 .. 0	6	92	3	189	5	110	3	51	3	13	1	1	0	0	
–10 .. –5	0	23	0	100	3	68	0	29	0	12	0	4	0	4	
–15 .. –10	0	6	1	21	0	27	0	22	0	9	0	2			
–20 .. –15			0	7	0	5	0	8	0	4	1	0			
–25 .. –20					0	1	0	2	0	1					
–30 .. –25									0	1					
Island	F	L	F	L	F	L	F	L	F	L	F	L	F	L	
Score															

Lost GS islands are in subcolumns *L*, and found GS islands in subcolumns *F*. The data for islands with the score < 35 and the length < 35 are only shown. All islands with score > 35 are found. The number of islands with the score < 35 and the length > 35 (as well as of those with the score below –30) is negligible. The shadowed cells are those where the number of lost islands exceed the number of found islands.

Among the alignments with the confidence > 60%, all islands have positive island confidence and only 3% contain less than two thirds of correctly aligned positions. In contrast, 14% of islands in the alignments with lower confidence have nothing in common with GS alignment, and 31% of them contain less than two thirds of correctly aligned positions.

Figures 4 and 5 show that alignment accuracy is determined by the total length of lost GS islands. In turn, the chance for a GS island to be found correctly is determined almost solely by its score (see Table I).

Kernels of the GS Islands

Generally, low scoring islands can be subdivided into two classes: islands that lack any region of significantly positive score (No algorithm using given substitution matrix is able to detect these islands.) and islands having a kernel of significantly positive score (compared to the gap-opening penalty in use). Low scoring islands of both types are mostly lost by the SW algorithm. However, in principle, a proper sequence alignment algorithm can identify the high scoring kernels.

We define a kernel score of an island as a maximal score of its fragment. For example, the island shown in Figure 7(a) has the kernel score of +11 (cf. with the notion of high scoring pair³⁰ HSP). The two-dimensional histogram given in Figure 7(b) presents the numbers of lost and found islands, depending on their scores and kernel scores. The data on the islands of significantly high kernel score, but low total scores are given in bold. A sequence alignment algorithm that addresses the kernels explicitly can, in principle, reveal kernels of these islands. This suggests that focusing on kernel regions only does not lead to the loss of the algorithm accuracy and even might increase it. At the same time, ungapped regions of a score larger than typically used gap opening penalty comprise a negligible part of the Needleman–Wunsch matrix.¹ Therefore, an algorithm that does not scan the complete matrix can significantly increase the computation speed.

Algorithm and Test Results

The above observation leads to the following algorithm schematically presented in Figure 8 (more detailed description of the algorithm is given in the Appendix):

1. Generate a set of ungapped high scoring segments (e.g., all HSPs⁴ with the score above specific threshold $T > 0$). At later steps of the algorithm, these segments will be used as “anchors” of the alignment procedure. Finally, some of these anchors will form island kernels of the alignment to be produced.

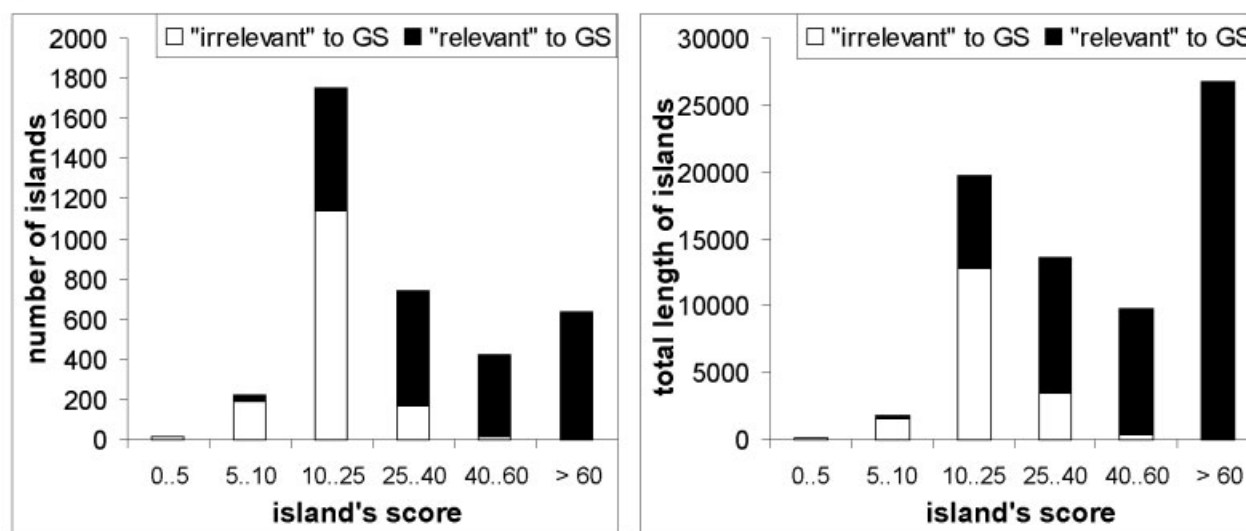
2. Find the optimal alignment path through the set of anchors (all elements of the Needleman–Wunsch matrix besides the anchors are set to zeros). The scoring function to be optimized differs from the traditional one in two aspects. First, it scores substitutions in anchor regions only. Second, instead of penalizing for number of gaps, the scoring function penalizes for number of anchors in the alignment path. In particular, we penalize the linkage between the anchors even if they belong to the same diagonal of the Needleman–Wunsch matrix (see Materials and Methods).

3. Specify the alignment path in the regions between the established anchors. These parts of the alignment have been left unspecified at the previous step of the algorithm.

The details of each step of the algorithm can be formulated in a number of ways. Special relation between the run time, accuracy, and confidence can be tuned to the desirable behavior, depending on exact algorithmic details and parameter values. The analysis of possible technical implementations and associated advantages and drawbacks is out of the scope of this manuscript. Table III presents the data on the characteristics of the algorithm version described in Materials and Methods.

Accuracy and confidence of the method have been tested through comparison with ~13,000 structural alignments extracted from BaliBase and FSSP databases. Results of these tests show that the novel method on average is equivalent to the SW algorithm both in accuracy and

a)



b) Same for %ID = 10-30%%

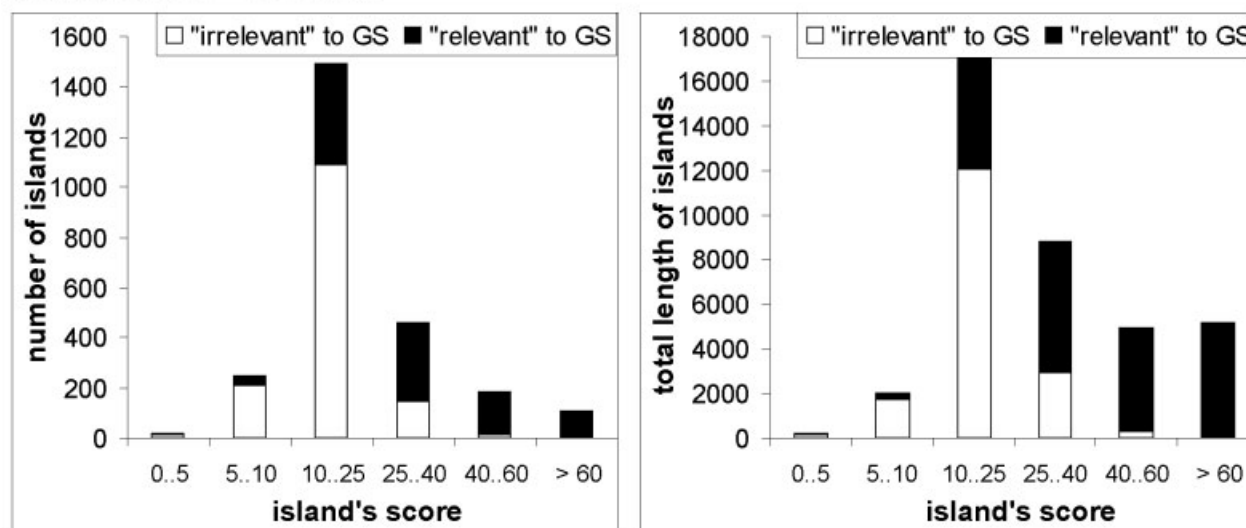


Fig. 6. Number and total length of irrelevant (white) and relevant (black)—to the gold standard alignment—islands from the Smith–Waterman alignments, having their scores within a given range. Irrelevant is the SW island with no single site corresponding to any GS island. Otherwise, the SW island is considered as relevant. Data are shown for linear domain gap penalty setting only. The results for log domain setting are qualitatively the same.

confidence of resulting alignments. As has been stated above, focusing on high scoring regions can significantly improve the computational speed of the algorithm. The current software has been compared with the standard (see Materials and Methods) implementation of the SW algorithm. Table III shows that the suggested method requires about twice a shorter computational time than the classic SW technique.

We have also compared (see Table IV) the accuracy and confidence of our algorithm with the BLAST algorithms [ftp://ftp.ncbi.nih.gov/blast/executables/blastz.exe]. The table shows that for the twilight zone protein pairs, our algorithm significantly outperforms BLAST both in confi-

dence and accuracy. However, BLAST requires much lower run time than the SW algorithm. Approximate estimates suggest that BLAST is about 15 times faster than SW algorithm.³¹

Another approach³² to speed up the Smith–Waterman algorithm was implemented recently by T. Rognes in the ParAlign program. The approach aims at reducing run time of the database search rather than individual alignments. First, it adapts the single instruction multiple data (SIMD) hardware technology to process in a parallel way the diagonals in the pathway matrix. Second, it uses a heuristic prealignment procedure to get a rough estimation of the protein similarity and thus to avoid the precise

TABLE II. Island's Confidence Correlates With the Overall Confidence of the SW Alignment†

	No. of alignments	No. of islands with score > 40	No. of islands with zero confidence	%	No. of islands with confidence < 2/3	%
a Alignments confidence ≤ 60%						
All	73	101	14	13.9	31	30.7
10–30%	69	97	13	13.4	29	29.9
b Alignments confidence > 60%						
All	338	870	0	0	26	3.0
10–30%	81	168	0	0	8	4.8

†The data shown in the table correspond to the linear domain of gap penalties. Data for logarithm domain are similar.

(a) Substitution**Scores:**

-3+1-2-4+1+7+1-1+3-4-4-1-2+1-3-4
v s f k d G d a i i n v q a i d
g a a w q G q i v g w y c t n l

Kernel**Score = +11****Total Island****Score = -14****(b)**

	< 0	0 .. 5	5 .. 10	10 .. 15	15.. 20	20.. 25	25.. 30	30.. 35	35+	Kernel score
> 35									655	1
30 .. 35								85 0	75 1	
25 .. 30							74 9	88 4	14 1	
20 .. 25						83 12	86 14	32 1	6 1	
15 .. 20					103 43	101 21	30 6	8 1	0 0	
10 .. 15				71 158	106 57	53 10	10 0	2 1	0 2	
5 .. 10			27 153	60 199	29 31	11 5	3 0			
0 .. 5		5 124	21 266	22 113	4 7	0 1				
-5 .. 0	0 16	8 230	9 174	2 37	2 2					
-10 .. -5	0 14	2 148	1 65	0 13	1 1					
-15 .. -10	1 6	0 53	0 23	0 5	0 0					
-20 .. -15	0 1	0 18	0 3	0 2	1 0					
-25 .. -20		0 2	0 1	0 1						
-30 .. -25		0 1		0 1						
Island Score	F L	F L	F L	F L	F L	F L	F L	F L	F L	F L

Fig. 7. **a:** Island in the GS alignment of two proteins (PDB codes 1ark and 1vie). The scores of residue substitutions are given according to the Gonnet250 matrix (values rounded to integers). This island has a negative score (−14); thus, it is lost by the SW alignment algorithm. However, this island contains a five-residue kernel fragment (underlined) with a positive score of +11. A sequence alignment algorithm using the Gonnet250 substitution matrix could find this kernel, in principle. **b:** Numbers of lost (subcolumns L) and found (subcolumns F) islands as a function of the island and kernel scores. Bold characters single out a region of islands with island scores < 10, but with kernel scores of ≥ 10. The empty cells contain zeros only.

alignment of nonsimilar proteins. The heuristic is based on the anchor paradigm, but it substantially differs from our approach. First, the heuristic gives only a rough estimation of the sequence similarity. The sequences with high heuristic score have to be subsequently realigned with the Smith–Waterman method. The necessity of realignment results from the features of the heuristic scoring function of the ParAlign (e.g., it uses only one HSP for each

diagonal). The SIMD technology can be implemented to expedite the most time-consuming step of our algorithm, the anchor's generation.

DISCUSSION

In this study, we investigated features of GS structural alignments in comparison with standard algorithmic alignments and proposed a new algorithmic solution for the

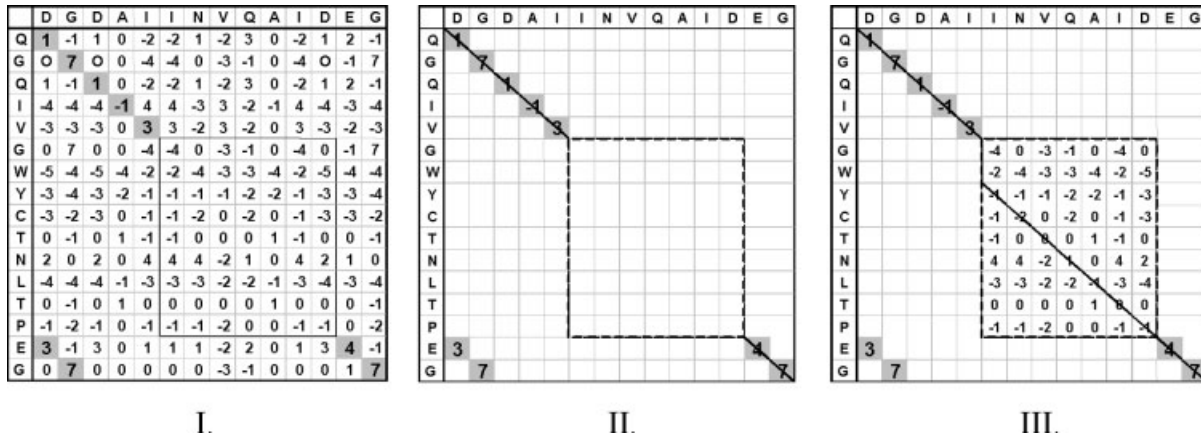


Fig. 8. The main steps of the suggested alignment algorithm. I: Identification of anchors. II: Creation of the optimal path through the anchors. III: Refinement of the alignment between the anchors.

TABLE III. Comparison of the Average Run Times, Accuracy, and Confidence of the Alignments, Produced by the SW Algorithm and the Newly Proposed Anchor Algorithm for the Protein Pairs From the BALiBase(a) and FSSP (b) Databases†

%ID	No. of pairs	Accuracy			Confidence			Run-time		
		Anchor	SW	An/SW	Anchor	SW	An/SW	Anchor	SW	An/SW
(a) 10–30%	298	36.1 ± 31.4	35.0 ± 32.1	1.03	49.6 ± 35.5	48.6 ± 37.1	1.02	397.5	731.3	0.54
>30%	253	83.2 ± 7.0	84.5 ± 6.6	0.98	89.1 ± 5.7	86.8 ± 6.6	1.03	158.8	393.4	0.40
All	583	39.8 ± 22.6	40.1 ± 24.7	0.99	54.5 ± 26.2	49.7 ± 27.5	1.10	275.7	552.3	0.50
(b) 10–30%	2605	47.7 ± 34.2	48.8 ± 33.4	0.98	59.3 ± 34.7	59.8 ± 31.7	0.99	174.9	316.6	0.55
>30%	10049	98.3 ± 1.5	98.5 ± 1.2	1.00	98.9 ± 1.1	98.7 ± 1.1	1.00	169.8	320.4	0.53
All	12671	76.5 ± 18.9	78.5 ± 18.7	0.97	84.5 ± 15	83.4 ± 15.4	1.01	170.8	319.5	0.53

†The data are given both for all protein pairs from the testing set and its subsets of the pairs of the twilight zone. For the Accuracy and Confidence columns, we give also standard deviation. The logarithmic sets of parameters (see Materials and Methods and Appendix) were taken both for Anchor and SW algorithms. To demonstrate that results do not depend on different size of considered protein families, the two-stage averaging procedure was applied for the “All” lines. First, the average values were calculated for each protein family separately. Then the averages of these values were obtained.

TABLE IV. Comparison of the Average Accuracy and Confidence of Alignments, Produced by the SW Algorithm, the Newly Proposed Anchor Algorithm and the BLAST Algorithm for the Protein Pairs Represented in the BALiBase†

%ID	No. of pairs	Accuracy			Confidence		
		Anchor	SW	BLAST	Anchor	SW	BLAST
10–30%	298	36.1 ± 31.4	35.0 ± 32.1	26.6 ± 20.5	49.6 ± 35.5	48.6 ± 37.1	44.6 ± 28.8
>30%	253	83.2 ± 7.0	84.5 ± 6.6	81.8 ± 8.1	89.1 ± 5.7	86.8 ± 6.6	87.0 ± 5.8
All	583	39.8 ± 22.6	40.1 ± 24.7	31.9 ± 23.8	54.5 ± 26.2	49.7 ± 27.5	47.8 ± 29.0

†The parameter sets for the Anchor and SW algorithm are same as in Table III. The default values of parameters were chosen for BLAST.

alignment problem, which follows from the analysis of 3D-based alignment.

Our results (Fig. 4) show that the relation between the GS and the algorithmic sequence alignments of a given protein pair can be expressed in terms of lost and found islands. Figure 5 shows that the GS islands with the score < 5 (given Gonnet250 matrix³³ in use) constitute a substantial part of GS alignments, and they have a negligible chance to be algorithmically reconstructed. The GS islands of a high score (generally higher than 25) are almost always found. However, the GS islands with scores in the range of 10–25 form a twilight zone. The chance of a particular GS island from a twilight zone to be identified depends on the presence of competing stretches of subse-

quent high scoring matches in the proteins to be aligned.²⁹ The analysis of high scoring alternative alignment paths is out of the focus of the current study; we concentrate on the analysis of properties of the GS islands.

The GS alignments often contain both high and low scoring islands. This finding suggests that the traditional substitution scoring functions of form $\text{Score} = \sum s(a_i, b_i)$ (see Eq. 3) assess many alignment regions inadequately. Equation 3 implies that all alignment positions are scored by a single substitution matrix or, in other words, the pattern of amino acid substitutions is identical in all sites. This assumption is well known to be inadequate. Usage of site-specific information incorporated into profiles¹⁶ (PSSMs) or hidden Markov models³⁴ (HMMs) greatly

TABLE V. Numbers and Lengths of Negatively Scored Islands in the Gold Standard Alignments From BALiBase for Different Substitution-Scoring Matrices Applied

Matrix	Blosum62	Gonnet250	BALiBase based
No. of negative islands	1374	936	799
Total length of negative islands	16833	10600	8674

Two first columns correspond to commonly used matrices Gonnet250 and BLOSUM62. The third column is obtained with the scoring matrices, computed from the BALiBase alignments. The latter matrices were computed for three categories of alignments (with sequence identity <25%, 25–50% and >50%) separately, according to the formulas given in Ref. 37. The BALiBase scores of the islands were calculated by using the matrix corresponding to the identity of compared protein sequences in the GS alignments.

improves alignment accuracy. However, these methods cannot be used without additional information on site-specific amino acid probabilities extracted from multiple alignments.

Although independence of the substitution matrix from the site-specific context is likely to be a major reason for the presence of numerous low scoring islands in structural alignments, it is definitely not the only reason. First, commonly used scoring matrices do not reflect exactly average statistics of amino acid substitutions in GS alignments. Table V shows that usage of matrices extracted directly from structural alignments (and restricted to specific sequence identity interval) significantly reduces the number and length of negatively scored GS islands. Analysis of low positively scored islands is difficult because of different scaling of these matrices.

The question arises of why commonly used matrices inadequately represent statistics of amino acid substitutions in GS alignments.

The matrices of Gonnet³³/PAM³⁵ series are based on the Markovian models of the process of amino acid substitutions. This approach relies on the neutral evolution theory¹⁸ and the concept of molecular clocks.¹⁸ There are two basic assumptions under this model: 1) divergence time can be estimated from amino acid sequence identity (which implicitly implies that evolutionary rate is identical for all sites in the sequence and for different sequences) and 2) probabilities of amino acid substitutions at a given divergence time depend solely on amino acid types. In other words, the pattern of natural selection does not vary among sites, proteins, and protein families. It is known that both assumptions are not valid in the strict sense for protein sequences.³⁶ This might constitute a reason for deviations of amino acid substitution statistics suggested by the scoring matrix from statistics observed in GS alignments.

The matrices of BLOSUM³⁷ series were computed for conserved ungapped blocks. Therefore, they reflect specific features of regions with a low evolutionary rate and may inadequately score the more divergent segments. This can

be an advantage in homology search application but can be a disadvantage if a complete and accurate sequence alignment is desirable (e.g., for homology-based 3D modeling). In addition, different conservative regions may also display distinct substitution statistics.

Recent studies³⁸ propose new models of amino acid sequence substitutions, which take into account differences in the evolutionary rates, site specificity, and differences in selective constraints. We did not address these models because they are not yet widely used for practical purposes and often rely on specific information additional to amino acid sequence.

Another implicit assumption underlying the scoring function (Eq. 3) is the assumption of independence of the alignment positions. Additive form of the scoring function implies that every amino acid pair is scored independently on its position along the alignment. To be more precise, total substitution score is defined as a logarithmic ratio of joint likelihoods in the multiplicative form.³⁹ Thus, the assumption of independence of positions is directly introduced into the additive form of substitution-scoring function.

Assessment of validity of the site independence assumption is presented in Table VI, which shows that the substitutions within the same ungapped segment (island) cannot be considered as independent. According to the contingency table χ^2 test,³⁹ the hypothesis of the independence of substitution score in islands has to be rejected with p value 5×10^{-17} . Table VI also shows that different protein families vary in the level of positional dependence of score.

Figure 9 illustrates this effect for three families by comparison of the original distribution of islands' scores with the distribution obtained for the same GS islands after random shuffling of alignment positions.

This result is complementary to the observation that many low scoring islands of structural GS alignments are inhomogeneous and contain internal high scoring kernels. This observation, together with the observation that the SW algorithm almost never identifies true islands when they have no significant kernel [Fig. 6(b)], shows that quick algorithms that explicitly address possible kernels (high similarity regions) should not necessarily have lower accuracy than the SW search.

Classic approaches such as the Wilbur–Lipman algorithm,⁴⁰ FASTA,³ and BLAST⁴ reduce the search space by focusing on high similarity regions and, therefore, have considerable increase in the computational speed. However, this strategy has been considered as an approximation of the true additive scoring function. FASTA and BLAST algorithms use a standard scoring around identified regions of high similarity. The Wilbur–Lipman method builds chains of high similarity kernels by using additive score over these kernels without a penalty. Regions of lower similarity are left unaligned by the method.

On the other hand, scoring functions designed to produce more adequate alignments^{41,42} were not tested on a large-scale basis and, therefore, probably did not attract the attention they would probably deserve. These methods

TABLE VI. Results of the χ^2 Test for Hypothesis of Independent Distribution of Substitution Scores Over the GS Islands

BaliBase family	Number of GS islands	<i>P</i> value
1havA	1642	$4.2 \cdot 10^{-7}$
1tgxA	568	$1.3 \cdot 10^{-5}$
1aboA	336	0.958
Kinase	280	$3.4 \cdot 10^{-9}$
1csy	256	0.003
1sbp	227	0.815
1pamA	165	0.388
1lvi	139	0.043
1ajsA	118	0.121
2pia	96	0.037
1cpt	92	0.089
2hsdA	82	$4.1 \cdot 10^{-4}$
3grs	81	0.459
1tvxA	76	0.394
1wit	69	0.010
1ped	66	0.025
1uky	57	0.310
4enl	53	0.099
1ubi	42	0.872
2trx	39	0.293
1r69	31	0.069
1idy	26	0.845
All BaliBase	4699	$5.1 \cdot 10^{-17}$

Family names as given in the BALiBase are listed in the first column (most of them are the PDB identifiers of one of the family members). The *p* value has a sense of probability that the hypothesis of independent distribution is valid. It is seen that this hypothesis is definitely not valid for some families (with $p < < 1$) as well as for the BaliBase as a whole. However, half of the families (with $p \approx 0.1-1$), including one fourth of the islands, do not show reliable deviations from the independence hypothesis.

differ from our approach mainly because they treat lower similarity regions between stronger matches essentially as gaps and penalize them accordingly. Thus, they do not allow these regions to be long and do not try to align them in any reasonable way after high similarity alignment elements have been found and the major alignment path has been determined.

Unlike methods described above, our algorithm takes a hierarchical approach that applies different scoring schemes to alignment regions according to degree of similarity. After high-similarity anchors have been constructed, it builds a major alignment backbone (chain of anchors) by using a scoring function that penalizes for number of anchors in the alignment. Each of the lower similarity regions left unaligned at the first step can be treated now as a region to be aligned. Because the search space in each of these yet unaligned regions is limited, they can be efficiently aligned by the global alignment dynamic programming (Needleman–Wunsch-like) with the standard additive scoring function and the standard affine gap penalty.

Two different strategies to build initial alignment backbone based on high-similarity anchors can be implemented

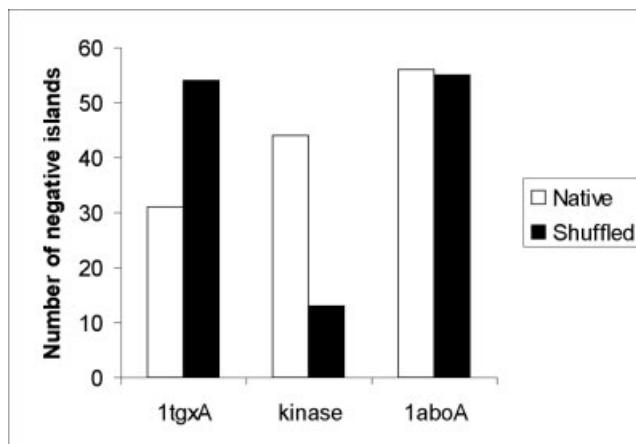


Fig. 9. Effect of random shuffling of the GS alignment columns on the number of negatively scored islands. The shuffling has been performed, separately for each protein alignment, by redistributing of the aligned amino acid pairs along the alignment (while the columns with gaps remain untouched). Results are presented for three BALiBase families: 1tgxA (cardiotoxins), kinase (protein kinases), and 1aboA (SH3 domains). White bars indicate the number of negative islands in the GS alignments, whereas black bars indicate the number of negative islands in the randomly shuffled alignments. The diagram shows that the shuffling does not change the number of negatively scored islands significantly in the case of 1aboA (which complements the insignificant *P*-value given in Table IV). For the kinase family, the number of negative islands considerably decreased after the shuffling. We note that this is a general case for families of low *p* values in Table IV. The 1tgxA family is an important exception. Almost every island in the alignments of this family contains one highly conserved cysteine residue. These aligned cysteine residues serve as a skeleton of the alignment. The shuffling often gathers several columns of aligned cysteines in one island. Correspondingly, the other islands lose their aligned conserved cysteines, and their scores become negative.

to use the reduction of the search space and gain computational speed. If the search space is large, one can use the sparse dynamic programming approach proposed by Epstein et al.⁴³ because its time complexity grows linearly with number of high scoring segments. Therefore, the sparse dynamic programming technique provides an efficient way to build an alignment path if the number of anchors is considerably high. However, in a smaller search space (e.g., for shorter protein sequences and greater anchor score thresholds), our method relies on the Wilbur–Lipman algorithm. Although asymptotically slower, this algorithm is faster in practice if the search space is not large and has been experimentally shown (data are not shown) to be preferable for protein of normal length.

Although the results presented in Table III report that our method is ~2 times faster than the SW search, two points are noteworthy. First, as any method that addresses only high-similarity alignment elements at the initial step, our method is potentially much faster as a database search tool. Therefore, the advantage over the SW algorithm in computational time exceeds the result of Table III if the database search problem is considered. Second, our method can be tuned to various regimens mostly by changing the anchor threshold. If a slight loss of accuracy is tolerated, the algorithm can be switched to a much faster behavior.

CONCLUSIONS

We investigated correspondence between GS alignments of 3D protein structures and sequence alignments produced by the Smith–Waterman algorithm. The comparison of the alignments is focused on their inner structure and specifically on the continuous ungapped alignment segments, which we call islands. Approximately one third of the islands in the GS alignments have negative or very low positive score (according to the commonly used scores), and recognition of these islands is below the sensitivity limit of the standard sequence-comparison algorithms. From the alignment accuracy perspective, the time spent by the algorithm while working in the regions where sequences cannot be aligned in principle is left without any profit. This finding inspired us to develop a novel hierarchical method to align a pair of protein sequences. At the first step, this method explicitly addresses the most similar fragments of compared sequences (the anchors). Furthermore, the method finds the optimal alignment pathway through the precalculated set of anchors and fills in (with Smith–Waterman-like method) the remaining comparatively short gaps between the anchors belonging to the optimal pathway. The resulting algorithm is considerably faster than the Smith–Waterman algorithm, whereas resulting alignments are on average of the same quality with respect to the GS. This finding shows that the decrease of alignment accuracy is not necessarily a price for the computational efficiency.

ACKNOWLEDGMENTS

We thank T. Gibson and W. Lathe III for useful discussions and careful reading of the manuscript. We also thank M.Yu.Lobanov for assistance in preparation of the figures and D.S. Rykunov for discussions. M.A.R. thanks I.M. Gelfand for stimulating communications. A.V.F. received the International Research Scholar's Award from the Howard Hughes Medical Institute.

REFERENCES

- Needleman S, Wunsch C. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 1970;48:443–453.
- Smith TF, Waterman MS. Identification of common molecular subsequences. *J Mol Biol* 1981;147:195–197.
- Lipman DJ, Pearson WR. Rapid and sensitive protein similarity searches. *Science* 1985;227:1435–1441.
- Altschul SF, Gish W, Miller W, Myers E, Lipman DJ. Basic local alignment search tool. *J Mol Biol* 1990;215:403–410.
- Russell RB, Barton GJ. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins* 1992;14:309–323.
- Holm L, Sander C. Protein structure comparison by alignment of distance matrices. *J Mol Biol* 1993;233:123–138.
- Gibrat JF, Madej T, Bryant SH. Surprising similarities in structure comparison. *Curr Opin Struct Biol* 1996;6:377–385.
- Koch I, Lengauer T, Wanke E. An algorithm for finding maximal common subtopologies in a set of protein structures. *J Comput Biol* 1996;3:289–306.
- Orengo CA, Taylor WR. SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol* 1996;266:617–635.
- Alexandrov NN. SARFing the PDB. *Protein Eng* 1996;9:727–732.
- Bork P, Koonin EV. Predicting functions from protein sequences—where are the bottlenecks? *Nat Genet* 1998;18:313–318.
- Bateman A, Birney E. Searching databases to find protein domain organization. *Adv Protein Chem* 2000;54:137–157.
- Sanchez R, Sali A. Comparative protein structure modeling. Introduction and practical examples with modeller. *Methods Mol Biol* 2000;143:97–129.
- Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res* 1994;22:4673–4680.
- Notredame C, Higgins DG, Heringa J. T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J Mol Biol* 2000;302:205–217.
- Gribskov M, Veretnik S. Identification of sequence pattern with profile analysis. *Methods Enzymol* 1996;266:198–212.
- David R, Korenberg MJ, Hunter IW. 3D-1D threading methods for protein fold recognition. *Pharmacogenomics* 2000;1:445–455.
- Li WH. Molecular evolution. Sunderland: Sinauer Associates; 1997.
- Doolittle RF. Similar amino acid sequences: chance or common ancestry? *Science* 1981;214:149–159.
- Thompson JD, Plewniak F, Poch O. BALiBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics* 1999;15:87–88.
- Vogt G, Etzold T, Argos P. An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited. *J Mol Biol* 1995;249:816–831.
- Domingues FS, Lackner P, Andreeva A, Sippl MJ. Structure-based evaluation of sequence comparison and fold recognition alignment accuracy. *J Mol Biol* 2000;297:1003–1013.
- Henikoff S, Henikoff JG. Amino acid substitution matrices. *Adv Protein Chem* 2000;54:73–97.
- Sander C, Schneider R. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins* 1991;9:56–68.
- Abagyan RA, Batalov S. Do aligned sequences share the same fold? *J Mol Biol* 1997;273:355–368.
- Rost B. Twilight zone of protein sequence alignments. *Protein Eng* 1999;12:85–94.
- Waterman MS. Introduction to computational biology. London-New York-Tokyo: Chapman & Hall; 1985.
- Sauser JM, Arthur JW, Dunbrack Jr. RL. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins* 2000;40:6–22.
- Saqi MA, Russell RB, Sternberg MJ. Misleading local sequence alignments: implications for comparative protein modelling. *Protein Eng* 1998;11:627–630.
- Altschul SF, Gish W. Local alignment statistics. *Methods Enzymol* 1996;266:460–480.
- Pearson WR. Protein sequence comparison and protein evolution. Tutorial-ISMB2000. Charlottesville, VA: University of Virginia; 2000. p 1–51.
- Rognes T. ParAlign: a parallel sequence alignment algorithm for rapid and sensitive database searches. *Nucleic Acids Res* 2001;29:1647–1652.
- Brenner SA, Cohen MA, Gonnet GH. Amino acid substitution during functionally constrained divergent evolution of protein sequences. *Protein Eng* 1994;7:1323–1332.
- Eddy SR. Profile hidden Markov models. *Bioinformatics* 1998;14:755–763.
- Dayhoff MO. Atlas of protein sequence and structure. Washington, DC: National Biomedical Research Foundation; 1979. p. 345–358.
- Grishin NV, Wolf YI, Koonin EV. From complete genomes to measures of substitution rate variability within and between proteins. *Genome Res* 2000;10:991–1000.
- Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA* 1992;89:10915–10919.
- Thorne JL. Models of protein sequence evolution and their applications. *Curr Opin Genet Dev* 2000;10:602–605.
- Kendall M, Stuart A. The advanced theory of statistics. New York: Charles Griffin and Co. Ltd.; 1979.
- Wilbur WJ, Lipman DJ. Rapid similarity searches of nucleic acid and protein data banks. *Proc Natl Acad Sci USA* 1983;80:726–730.
- Alexandrov NN, Luethy R. Alignment algorithm for homology modeling and threading. *Protein Sci* 1998;7:254–258.

42. Altschul SF. Generalized affine gap costs for protein sequence alignment. *Proteins* 1998;32:88–96.
43. Eppstein D, Galil Z, Giancarlo R, Italiano GF. Sparse dynamic programming. 1. Linear cost-functions. *J ACM* 1992;39:519–545.

APPENDIX: THE ANCHOR-BASED ALIGNMENT ALGORITHM

The proposed alignment algorithm (illustrated in Fig. 8) consists of the following steps:

Step 1. We generate a set of ungapped high-scoring segments (anchors). Anchor is an ungapped matching of equal-length fragments, $\{U[a, a+L] \text{ vs } V[b, b+L]\}$, of sequences U and V . These fragments meet the following conditions (cf. with BLAST HSPs):

- Anchor contains at least one seed pair $\{U[x, x+1] \text{ vs } V[y, y+1]\}$ with the score exceeding a cutoff C_{Seed}
- The anchor's score (i.e., the sum of the substitution scores $M(U[x], V[y])$ over the anchor) exceeds a cutoff C_{Anchor}
- The score of any continuous part of the anchor exceeds a cutoff C_{Min}
- The anchor is locally maximal, that is, 1) it is not a part of any other pair of segments $\{U[a', a'+L'] \text{ vs } V[b', b'+L']\}$ meeting conditions a–c and having greater or equal score, and 2) it does not include any continuous part having a greater score.

Step 1 starts with identification of seed pairs. Seeds are further expanded to obtain the anchors. This step is similar to procedures used in BLAST and FASTA. This is the most time-consuming step of our algorithm.

Step 2. We find the optimal block alignment path through the set of anchors. Block is a continuous part of an anchor. Block alignment is a chain of the blocks $\{B_1, \dots, B_N\}$, where B_i precedes B_{i+1} both along U and V sequences. The block alignment $\{B_1, \dots, B_N\}$ is optimal if it has maximal possible block score, which is defined as follows:

$$\begin{aligned} \text{Score}(B_1, \dots, B_N) = & \text{Score}(B_1) - \text{Link}(B_1, B_2) \\ & + \text{Score}(B_2) - \dots - \text{Link}(B_{N-1}, B_N) + \text{Score}(B_N) \end{aligned}$$

$\text{Score}(B_i)$ is the total score of matches along block B_i according to the given substitution matrix M . $\text{Link}(B_i, B_{i+1}) = \alpha + \beta \bullet |(y-x) - (y'-x')|$ is the linkage penalty for the blocks B_i and B_{i+1} , where α (linkage open penalty, LOP) and β (linkage elongation penalty, LEP) are analogs of the traditional gap opening (GOP) and gap elongation penalties (GEP), whereas x, y are the last residues of block B_i , and x', y' are the first residues of block B_{i+1} in sequences U and V , respectively. Note that we penalize links between the blocks even if the blocks are placed on the same diagonal.

Step 3. We specify the alignment path in regions between the blocks by connecting consecutive anchors via standard global dynamic programming. In case the resulting connecting path has a score below a given cut-off LST , the region is left unaligned. Our experiments show that usually this final step comprises only a small part of the total run time of our algorithm.

The data shown in Table III correspond to the following values of parameters: $C_{Seed} = 8$, $C_{Anchor} = C_{min} = 20$, $LOP = 15$, $LEP = 0.5$. The final step of the dynamic programming implied $GOP = 15$, $GEP = 1$ and $LST = -25$.

To find the optimal block alignment from the created set of anchors we have implemented two algorithms: the Wilbur–Lipman algorithm⁴⁰ and the sparse dynamic programming method⁴³ (SDP). These procedures produce the same alignments (given the same parameters and set of anchors), but they differ in the run time: the Wilbur–Lipman algorithm run time is proportional to K^2 , whereas the SDP run time is of order $K \cdot \log(L)$, where K is number of anchors and L is length of the shorter sequence. The Wilbur–Lipman procedure performs faster for small values of K because of much simpler technique in use, whereas the SPD method is more efficient in case of large K corresponding to longer sequences. Our experiments have shown that in realistic protein sequences the best run time can be achieved by the Wilbur–Lipman method.