Assessing distant homology between an aligned family and a proposed member through accurate sequence alignment*

Weiqing Zhang† John D. Kececioglu‡ E. Will Taylor§

August 17, 1997; revised January 15, 1998

Abstract A major challenge in computational biology is the identification of evolutionarily related macromolecules in cases of distant homology, where pairwise sequence similarity may be low, or even insignificant. Methods for the quantitative assessment of such distant relationships must go beyond simple pairwise sequence comparison, and exploit all the information available in multiple alignments of related sequences. Evaluation of the statistical significance of a pairwise alignment score by random shuffling is an established approach, which we extend here to the more general case of evaluating the similarity between a query sequence and a prealigned family of macromolecules whose multiple alignment may contain gaps. The method involves (1) the optimal alignment of the query sequence against the prealigned family, for which we develop a new algorithm that accurately takes into account the structure of gaps in the alignment, followed by (2) repeated shuffling of the query sequence and calculation of a shuffling statistic to assess the significance of the query-family homology, for which we consider four different measures. We apply the method to several data sets of interest, and compare its sensitivity to the traditional method of pairwise sequence comparison. Overall, the significance score for the query-family homology assessed by the new method exceeded the median significance score assessed by the traditional pairwise method by 2.2 standard deviations on average. This suggests that the method is consistently more sensitive for assessing distant homology than the conventional approach based on pairwise sequence comparison.

Keywords Sequence comparison, statistical significance, sum of pairs alignment, affine gap costs

1 Introduction

Sequence comparison and alignment are among the most important tools of computational molecular biology. They are fundamental for studies of structure-function relationships and

*Technical Report 97-3, Department of Computer Science, The University of Georgia, August 1997. Submitted to *Journal of Molecular Biology*.

†Department of Medicinal Chemistry, University of Georgia, Athens, GA 30602. Email: `wzhang@rx.uga.edu`

‡Department of Computer Science, The University of Georgia, Athens, GA 30602-7404. Email: `kece@cs.uga.edu`. Research supported by a National Science Foundation CAREER Award, Grant DBI-9722339.

§Computational Center for Molecular Structure and Design, The University of Georgia, Athens, GA 30602. Email: `wtaylor@rx.uga.edu`

the evolution of proteins and nucleic acids. Sequence alignments are normally used to identify the regions of similarity and dissimilarity within sets of related sequences, e.g., as a basis for computing molecular phylogenies. Identification of regions of sequence similarity can help the biologist to identify conserved residues that are essential either for function (e.g., active site and catalytic amino acids), for maintaining the required secondary structure, or for long-distant interactions involved in stabilization of the 3-dimensional structure.

Thus, an accurate sequence alignment is also an essential requirement for 3-dimensional structure prediction by protein homology modeling. A problem with that approach is that genuinely homologous proteins can have low or even insignificant pairwise sequence similarity, even though their overall 3-dimensional structures may be essentially the same. Such proteins can be described as "distant homologues," which can include proteins related by a particularly ancient divergence, as well as cases where at least one of the sequences is subject to an unusually high mutation rate. The latter applies to many proteins encoded by RNA viruses, where lack of polymerase "proofreading" capability combined with very high replication rates can lead to rapid sequence divergence.

With the advent of various genome sequencing projects, as databases grow exponentially, the problem of testing an hypothesis of distant homology is becoming increasingly common. Database searches with query sequence probes can turn up intriguing "hits," but those are often based upon a local sequence similarity, and overall sequence identity may be low; this provides a basis for the hypothesis of homology, but offers little firm evidence to support it. Aside from the obvious cases of functional similarity based upon biochemical studies, or structural similarity confirmed by X-ray crystallography, NMR, or theoretical structure-based approaches such as threading or inverse folding [12, 13, 14], there are few available options for the quantitative assessment of an homology hypothesis based purely upon primary sequence data.

The two major sequence-based approaches that have been widely used to support or assess a proposed case of homology are,

- Database searching with the query sequence as a probe, where the significance of the similarity score of the query to a given "hit" is expressed as a "Z score" relative to the average similarity score for the entire database, where the Z score represents the number of standard deviations above the database average [32].
- Pairwise sequence alignment of the putative homologous segments of the query and the hit sequences, where the significance of the alignment score of the query to the hit is expressed as a Z score relative to the average alignment score over a large set of randomly shuffled sequences of identical composition; in this case the Z score represents the number of standard deviations above the average score expected for a random sequence with the same composition [32].

Both these methods are of limited usefulness in cases of borderline or distant homology. The most widely-used database searching algorithms are designed for speed, and are most useful for finding matches to local regions of high similarity.

Pairwise alignment with random shuffling has been widely used for significance testing, in part because the method has been implemented in various readily available commercial programs, such as the Bestfit and Gap routines in the GCG package [35]. Although it is a classic method for assessing the potential significance of sequence similarities [6, 7, 18], pairwise alignment with random shuffling is inherently limited in cases of borderline or distant homology, which by definition imply that low pairwise sequence similarity is present.

The solution to this problem lies in the comparison of a query sequence to the complete family of sequences, since there is additional information in a multiple alignment of the family that can not be detected by comparison to a single member alone. For example, a query sequence that has high similarity to several highly conserved regions in a multiple alignment, which are usually called sequence motifs [22, 3, 34], is conceptually more significant than a query sequence that has a comparable degree of similarity to individual member sequence, but with matches randomly distributed across various locations in a multiple alignment.

Although it is a widely used strategy, running successive pairwise alignments of the query against all the individual sequences of a multiple alignment cannot overcome this limitation, because it is not possible to correlate where the matches or similarities occur in the different pairwise alignments.

In such cases, if the "hit" to the query sequence corresponds to an entire family of sequences that can be simultaneously aligned, what is needed is *a method that can take into account the existence of conserved and unconserved regions present in a multiple alignment containing gaps*. Such a method should be capable of taking as input the existing multiple alignment of the family and producing as output a rigorously optimized alignment of the query sequence against the aligned family, yielding an alignment score that can then be compared to the average alignment score for randomly shuffled sequences of identical composition as the query. Such an approach could be seen as a direct extension of the method of pairwise alignment with random shuffling, except that the query and permuted variants are each optimally aligned against an existing multiple alignment that may contain internal gaps, instead of against an isolated sequence.

In this paper, we report this new approach and its application to several data sets of interest, with a comparison of its sensitivity to the traditional method of pairwise sequence comparison. The algorithm for optimal alignment of a sequence vs. a fixed multiple alignment is outlined here, and is described in detail in a separate publication [24].

2 Computational methods

We first describe the new algorithm for aligning a query sequence to a prealigned sequence family that accurately takes into account gaps using the technique of dynamic programming. (For a survey on dynamic-programming sequence-alignment algorithms, see [33].)

2.1 Aligning a sequence to a family

The input to the algorithm is the query sequence A and the fixed multiple alignment B of the sequence family. The output is an optimal alignment of the elements of A against the columns of B, where alignments are scored using the sum-of-pairs scoring function [4] with linear gap costs [15]. More precisely, an alignment of A to B is assigned a score by summing over all rows of B the score of the induced pairwise alignment between A and the given row of B, where pairwise alignments are scored with gaps penalized by a general linear function of their length. We seek such an alignment of optimum score.[1] In the remainder we call the input alignment B a *block* and we refer to this problem as *sequence-block alignment*. Note

[1]Note that this differs from general sum-of-pairs multiple alignment [25, 20] (which is intractable [39]) in that the alignment on the rows of B is fixed.

that in the literature the term "block" is most often used to refer to an ungapped multiple alignment. In our use, a block can contain gaps.

Our computational work [24] develop several algorithms for sequence-block alignment.[2] The problem of optimal sequence-block alignment under exact gap counts is difficult, and prior to our work no algorithm that is efficient in the worst-case was known. Due to this difficulty, simplifications of exact gap counts are usually considered, most often the "quasi-natural gap counts" introduced by Altschul [1]. When inaccurate, these gap counts over-estimate the exact counts, hence we call these gap counts *pessimistic*. Our simplest algorithm for sequence-block alignment computes an optimal alignment under pessimistic gap counts in $O(kmn)$ time[3] where k is the number of sequences in block B, n is the length of B, and m is the length of sequence A. Our most complex algorithm computes an optimal sequence-block alignment under *exact* gap counts in $O(kmn + mn \log n)$ time. This may be contrasted with Gotoh's work [16] on general block-block alignment, for which no polynomial worst-case time bound is known under exact gap counts. We describe here our simple algorithm for pessimistic gap counts; more details are in [24].

The classic technique for solving sequence alignment problems is dynamic programming, and we use this technique to solve the sequence-block alignment problem. To specify our dynamic programming algorithm, we present the system of recurrence equations that the algorithm solves. This completely specifies the algorithm for a reader familiar with sequence alignment [40].

In the recurrences, we denote the characters in sequence A by $a_1 a_2 \cdots a_m$, and we treat block $B = (b_{ij})$ as a $k \times n$ matrix. An element b_{ij} of block B is either an amino acid or the *null character* ϵ. When displaying an alignment, the null character ϵ is usually denoted by a dash '-'.

To describe the alignment scoring function, we use δ to denote the matrix of amino acid substitution scores (such as the PAM [6] or BLOSUM [21] matrices), and we write $\delta(a, b)$ for the cost of substituting amino acid a by amino acid b. (Thus δ contains amino acid *dissimilarity* scores.) For the gap costs we use γ to denote the gap start-up cost and λ to denote the gap extension cost. Thus a gap of length $x > 0$ has cost $\gamma + \lambda x$. Together, δ, γ, and λ specify the alignment scoring function.

To conveniently express whether an element b_{ij} of block B is the null character, we define

$$\beta_{ij} \;:=\; \begin{cases} 1, & b_{ij} \neq \epsilon \text{ and } j > 0; \\ 0, & b_{ij} = \epsilon \text{ and } j > 0; \\ 1, & j = 0. \end{cases}$$

(To correctly handle the boundary conditions of the recurrences, this treats block B as starting with an artificial 0th column of non-null characters.) The number of non-null characters in column j of B is then

$$c(j) \;:=\; \sum_{1 \le i \le k} \beta_{ij}.$$

[2] Our algorithmic paper [24] also treats the block-block [16], sequence-profile [19], and profile-profile [17] variants as well.

[3] The notation $O(f(x))$ describes a quantity that for all large x is upper bounded by a constant multiple of $f(x)$.

To count gap start-ups we use a function $g(a_1, a_2, b_1, b_2)$ given in tabular form below. Each of the four arguments of g takes on the values 0 or 1 only. Essentially, a_1a_2 represents whether the last two entries of the row for A in the alignment are non-null characters, and b_1b_2 represents whether the last two entries of a given row for B in the alignment are non-null characters. In the table for g below, rows are indexed by the pair a_1a_2, and columns by the pair b_1b_2. With this indexing the function is symmetric.

	00	01	10	11
00	0	**1**	0	0
01	**1**	0	1	0
10	0	1	0	1
11	0	0	1	0

The two entries in boldface distinguish the gap counts as pessimistic; for optimistic gap counts these two entries are zero. Finally, to simplify the recurrences we define the four quantities,

$$
\begin{aligned}
g_{HH}(j) &:= \sum_{1 \le i \le k} g(0, 0, \beta_{ij-1}, \beta_{ij}), \\
g_{HV}(j) &:= \sum_{1 \le i \le k} g(0, 1, \beta_{ij}, 0), \\
g_{HD}(j) &:= \sum_{1 \le i \le k} g(0, 1, \beta_{ij-1}, \beta_{ij}), \\
g_{DD}(j) &:= \sum_{1 \le i \le k} g(1, 1, \beta_{ij-1}, \beta_{ij}).
\end{aligned}
$$

In the above, $g_{xy}(j)$ counts the number of gap start-ups created by following an x-edge by a y-edge involving column j of B in the standard dynamic programming graph, where x and y indicate whether the edge direction is horizontal, vertical, or diagonal. (See below.)

We can now write down the recurrences for the value of an optimal alignment. Let

- $H(i, j)$ be the value of an optimal alignment of the first i characters of A with the first j columns of B, given that it ends by inserting column j of B (which corresponds to a *horizontal* edge in the standard dynamic programming graph),
- $V(i, j)$ be the value of the same alignment, given that it ends by deleting character i of A (which corresponds to a *vertical* edge), and
- $D(i, j)$ be the value of the same alignment, given that it ends by substuting character i with the characters in column j of B (which corresponds to a *diagonal* edge).

Then the value of the optimal alignment of A and B (the alignment score) is

$$
s(A, B) = \min\{H(m, n), V(m, n), D(m, n)\}.
$$

The recurrences for H, V, and D are as follows. For $i > 0$ and $j > 0$,

$$
H(i, j) = \min \left\{ \begin{array}{l} H(i, j-1) + \gamma\, g_{HH}(j), \\ V(i, j-1) + \gamma\, c(j), \\ D(i, j-1) + \gamma\, c(j) \end{array} \right\} + \lambda\, c(j),
$$

$$V(i,j) \;=\; \min \left\{ \begin{array}{l} H(i-1,j) \;+\; \gamma\, g_{HV}(j), \\ V(i-1,j), \\ D(i-1,j) \;+\; \gamma\, c(j) \end{array} \right\} + \lambda\, k,$$

$$D(i,j) \;=\; \min \left\{ \begin{array}{l} H(i-1,j-1) \;+\; \gamma\, g_{HD}(j), \\ V(i-1,j-1), \\ D(i-1,j-1) \;+\; \gamma\, g_{DD}(j) \end{array} \right\} + \sum_{1 \le x \le k} \delta(a_i, b_{xj}).$$

The complete boundary conditions for the recurrences are given below. (These boundary conditions count gaps exactly.) We use $d(j)$ to denote the number of non-null characters in column j of B that are the leftmost non-null character in their row.

For $i = 0$ and $j > 0$,

$$\begin{aligned} H(0,j) &= H(0,j-1) \;+\; \lambda\, c(j) \;+\; \gamma\, d(j), \\ V(0,j) &= H(0,j) \;+\; \gamma\, k, \\ D(0,j) &= V(0,j). \end{aligned}$$

For $i > 0$ and $j = 0$,

$$\begin{aligned} V(i,0) &= V(i-1,0) \;+\; \lambda\, k, \\ H(i,0) &= V(i,0) \;+\; \gamma\, c(1), \\ D(i,0) &= H(i,0). \end{aligned}$$

For $i = 0$ and $j = 0$,

$$\begin{aligned} V(0,0) &= \gamma\, k, \\ H(0,0) &= 0, \\ D(0,0) &= 0. \end{aligned}$$

The algorithm evaluates the above recurrences, storing the computed values in three $(m+1) \times (n+1)$ tables that correspond to quantities H, V, and D above. Once the solution value $s(A,B) = \min\{D(m,n), H(m,n), V(m,n)\}$ is computed, the optimal alignment is recovered by tracing back through the tables. The complete algorithm to find the optimal sequence-block alignment with pessimistic gap counts takes $O(kmn)$ time and uses $O(mn)$ space.

The experimental results we report are for alignments obtained by this algorithm. We note that the space can be reduced to $O(\min\{m,n\})$ when the sequences are extremely long, at a cost of roughly doubling the execution time, using standard techniques for linear-space alignment [30]. We also note that when the number of sequences in block B is large, the time can be reduced to $O(smn)$ by preprocessing B, where s is the size of the alphabet (in this context $s = 20$). We also mention that the recurrences may be modified to find an alignment with no charge for overhanging ends, and to find a local alignment over substrings of A and substrings of B rather than a global alignment of all of A with all of B. Both are achieved by transforming into a similarity formulation and modifying the recurrences slightly [40].

As mentioned earlier, we have developed an algorithm that computes an optimal sequence-block alignment under *exact* gap counts in $O(kmn + mn \log n)$ time [24] by applying techniques from pairwise alignment with convex gap costs [29, 11]. As this algorithm is much more involved, and will give differing results only for input blocks with an uncommonly high density of gaps, we do not describe it here.

2.2 Assessing statistical significance

To assess the homology of the protein sequence A to the protein family B, we follow the established procedure [7, 26] of comparing the score of an optimal alignment between A and B to the score of an optimal alignment between a random sequence $\widetilde{A}$ and B, where $\widetilde{A}$ is constrained to have the same amino acid composition as A [10, 2]. The homology between A and B is deemed statistically significant if the deviation of the optimal sequence-block alignment score $s(A, B)$ from the expected value of $s(\widetilde{A}, B)$ is sufficiently high. A threshold τ for the number of standard deviations is chosen, and a sequence A whose score differs from the expected value by at least τ standard deviations is considered a candidate for being homologous to family B.

To make this procedure more precise, we first review how it is traditionally carried out for sequence-sequence alignment. We then explain how the procedure conventionally has been applied in the more general setting of comparing a sequence to an aligned family, and we contrast the conventional practice with a natural generalization of the traditional method that takes advantage of the algorithm for sequence-block alignment.

After describing these various approaches for estimating the significance of sequence-family homology, we end the section by explaining why the approaches based on sequence-sequence alignment give lower estimates than the approach based on sequence-block alignment, and by describing how to compute the shuffled sequence for the significance tests.

2.2.1 Standard shuffling statistic

Traditionally the significance of homology between two sequences A and B is tested as follows. Let $\widetilde{A}_1$, $\widetilde{A}_2$, ..., $\widetilde{A}_r$ be a sample of r independent random sequences chosen uniformly over all sequences having the same character frequency as A, and let τ be an agreed-upon standard deviation threshold. For a sample of real numbers a_1, a_2, ..., a_n, denote the sample *average* by

$$\operatorname*{avg}_{1 \le i \le n} \{a_i\} \;:=\; \frac{1}{n} \sum_{1 \le i \le n} a_i,$$

and the sample *standard deviation* by

$$\operatorname*{dev}_{1 \le i \le n} \{a_i\} \;:=\; \sqrt{\frac{1}{n} \sum_{1 \le i \le n} a_i^2 \;-\; \left(\operatorname*{avg}_{1 \le i \le n} \{a_i\}\right)^2}.$$

Then the *shuffling statistic* for the homology of A to B (often called the "Z score" in the literature[4]) is

$$Z_r \;:=\; \frac{|s(A, B) \;-\; \mu|}{\sigma},$$

[4] We prefer the term "shuffling statistic" to "Z score," as the quantity is computed by randomly shuffling, or permuting, the input sequence, and we prefer to not use a term that is based on notation.

where

$$\begin{aligned}\mu &:= \operatorname*{avg}_{1\le i\le r}\left\{ s(\widetilde{A}_i, B)\right\},\\ \sigma &:= \operatorname*{dev}_{1\le i\le r}\left\{ s(\widetilde{A}_i, B)\right\},\end{aligned}$$

and $s(x, y)$ is the score of the optimal sequence-sequence alignment between x and y.

If for sufficiently large r (i.e. an acceptably large number of randomizations),

$$Z_r \;\ge\; \tau,$$

then the shuffling statistic is considered to support the homology of A and B. In practice, $\tau = 4$ is often chosen; in our experiments of Section 3, we take $r = 100$.

2.2.2 Generalizing to sequence-block alignment

We now describe four ways to define the shuffling statistic between a sequence and an aligned family. The first three are appropriate when only an algorithm for sequence-sequence alignment is available, while the last exploits an algorithm for true sequence-block alignment. Each is successively a more faithful approximation to the last one, which is a natural generalization of the standard shuffling statistic to sequence-block alignment.

Given sequence A and block B, again let $\widetilde{A}_1$, $\widetilde{A}_2$, ..., be a series of random shufflings of A, and let B_1, B_2, ..., B_k be the family of sequences in B. (Formally B_i is the ith row of B with null characters removed.) We now write $s(A, B)$ for the score of an optimal *sequence-block* alignment.[5]

We give four definitions of the shuffling statistic for A and B.

(1) The *conventional pairwise* shuffling statistic, which we denote by Z_r^C, takes the *average* of the traditional shuffling statistic between A and each member of B. When one can only compute a shuffling statistic between two sequences, this is the simplest approach, and is the one conventionally taken in the literature. For a family with k members, one runs a traditional shuffling statistic program [32] separately on each member, and takes the average of the k pairwise shuffling statistics that are output.

Formally,

$$Z_r^C \;:=\; \operatorname*{avg}_{1\le i\le k}\left\{ \frac{|s(A, B_i) \;-\; \mu_i|}{\sigma_i} \right\}, \tag{1}$$

where

$$\begin{aligned}\mu_i &:= \operatorname*{avg}_{1\le j\le r}\left\{ s\Big(\widetilde{A}_{(i-1)r+j}, B_i\Big)\right\},\\ \sigma_i &:= \operatorname*{dev}_{1\le j\le r}\left\{ s\Big(\widetilde{A}_{(i-1)r+j}, B_i\Big)\right\}.\end{aligned}$$

Note that this performs r randomizations of A for each member of B, for a total of kr shufflings.

[5] Note that with this new interpretation, $s(A, B_i)$ is still the score of a sequence-sequence alignment.

This definition of a sequence-block shuffling statistic has little formal basis. First, it is unclear whether an average of shuffling statistics is a meaningful quantity. (For instance, such an average is itself not a shuffling statistic.) Second, there are a multitude of ways to reduce the k pairwise shuffling statistics to a single number, and any one of them could just as defensibly be used. In Section 3, for example, we consider taking the median, minimum, and maximum of the k shuffling statistics, as well as the mean. All gave consistently low estimates for the significance of homology between the sequence and the family in experiments with distant sequences.

Modifying the conventional approach so that it does not take an average of shuffling statistics leads to the following definition.

(2) The *modified pairwise* shuffling statistic, which we denote by Z_r^M, has the same form as the standard shuffling statistic, in contrast to the conventional approach. It differs from the standard statistic by replacing the alignment score with an average score, where the average is taken over the pairwise alignments between A and the members of B. This average is compared to the same random scores used by the conventional approach; instead of taking an average of shuffling statistics, however, the shuffling statistic of an average is computed.

Formally,

$$Z_r^M \;:=\; \frac{\left| \operatorname*{avg}_{1\le i\le k} \Big\{ s(A, B_i) \Big\} \;-\; \mu \right|}{\sigma}, \tag{2}$$

where

$$\begin{aligned} \mu \quad &:= \quad \operatorname*{avg}_{\substack{1\le i\le k \\ 1\le j\le r}} \left\{ s\left(\widetilde{A}_{(i-1)r+j}, B_i\right) \right\}, \\ \sigma \quad &:= \quad \operatorname*{dev}_{\substack{1\le i\le k \\ 1\le j\le r}} \left\{ s\left(\widetilde{A}_{(i-1)r+j}, B_i\right) \right\}. \end{aligned}$$

In our experiments of Section 3, this definition, which follows the form of a shuffling statistic, yielded even lower estimates than the conventional approach.

(3) The *summed pairwise* shuffling statistic, which we denote by Z_r^S, also has the form of the traditional statistic, but (1) replaces the alignment score by the sum of the pairwise alignment scores between A and the members of B, and (2) compares this score to the same sum evaluated on random shufflings of A.

Formally,

$$Z_r^S \;:=\; \frac{\left| \sum_{1\le i\le k} s(A, B_i) \;-\; \mu \right|}{\sigma}, \tag{3}$$

where

$$\mu \quad := \quad \operatorname*{avg}_{1\le j\le r} \left\{ \sum_{1\le i\le k} s(\widetilde{A}_j, B_i) \right\},$$

$$\sigma \quad := \quad \operatorname*{dev}_{1 \le j \le r} \left\{ \sum_{1 \le i \le k} s(\widetilde{A}_j, B_i) \right\}.$$

Notice that in contrast to the conventional and modified approaches, the summed pairwise statistic only requires r shufflings of A.

Of the approaches that use only sequence-sequence alignment, the summed approach is closest in form to the final one we present next, and in our experiments with distant sequences, it gave much higher estimates for the significance of homology than either the conventional or modified approach.

(4) The *blockwise* shuffling statistic, which we denote by Z_r^B, generalizes the traditional sequence-sequence shuffling statistic by simply replacing every score with a sequence-block alignment score.

Formally,

$$Z_r^B \quad := \quad \frac{|s(A,B) \,-\, \mu|}{\sigma}, \tag{4}$$

where

$$\mu \quad := \quad \operatorname*{avg}_{1 \le j \le r} \left\{ s(\widetilde{A}_j, B) \right\},$$

$$\sigma \quad := \quad \operatorname*{dev}_{1 \le j \le r} \left\{ s(\widetilde{A}_j, B) \right\}.$$

Notice that this is identical to the traditional shuffling statistic when family B consists of one member, and also requires only r randomizations of A. In our experiments, it gave the highest estimates for the significance of homology.

We next discuss why pairwise statistics tend to give lower significance estimates than the blockwise statistic for distantly-related query sequences.

2.2.3 Explicating the difference between the blockwise and pairwise statistics

Intuitively one might expect that when pairwise similarity between the sequence and individual family members is low, pairwise shuffling statistics might tend to give lower estimates for the significance of homology than the blockwise shuffling statistic, and our experiments of Section 3 support this intuition. Of the pairwise statistics, the summed statistic Z^S is closest in form to the blockwise statistic Z^B; we now consider why the significance estimate Z^S might be less than Z^B.

To see why this might be the case, consider the following thought-experiment abstracted in Figure 1. In this scenario, both the proposed member A and the known members of family B contain a region of small variation, here idealized by the patterns `CX` and `XC`, that is anchored on either side by highly conserved regions, here idealized by `W`. Within the varying region, the majority of the members of B contain `CX`, but a small minority contain the variant `XC`. Furthermore, this region contains an element, represented by `C`, whose position varies across the members, but is usually conserved in protein sequences. Because this element is usually conserved, the substitution matrix δ that scores alignments favors aligning these elements. The `C` elements are not aligned in family B, however, because doing so would force either misalignment of the highly conserved region W, or introduction of a gap to align both the C elements and the flanking W elements.[6]

A	$\cdots$ W CX W $\cdots$
B_1	$\cdots$ W CX W $\cdots$
B_2	$\cdots$ W CX W $\cdots$
	$\cdots$ W CX W $\cdots$
	$\cdots$ W CX W $\cdots$
$\vdots$	$\vdots$
	$\cdots$ W CX W $\cdots$
	$\cdots$ W XC W $\cdots$
B_k	$\cdots$ W XC W $\cdots$

Figure 1 Hypothetical example for blockwise versus pairwise shuffling statistics.

Now consider the alignment scores that result when shuffling A for the summed pairwise statistic. Once A is shuffled, the flanking W elements will be at arbitrary positions, and will no longer constrain the alignment of the C element in $\widetilde{A}$. Hence if the C element in $\widetilde{A}$ falls sufficiently near either of the two positions of the C element in B, the pairwise alignments of $\widetilde{A}$ to members of B that contain CX will align the C to the first position, while the pairwise alignments to members containing XC will align the C to the second position. Thus the C element will contribute favorably to the pairwise alignment scores as long as it falls near either of the two positions of the C element in B. Furthermore, when the C in $\widetilde{A}$ is permitted to align to the first position in some members and the second position in other members, this effectively lines up all C elements in B through the C element in $\widetilde{A}$, in effect altering the multiple alignment of B to achieve a more favorable summed score with the shuffled $\widetilde{A}$.

Consider, on the other hand, the situation with the blockwise statistic. In the sequence-block alignment of the unshuffled A to B, the C element in A will be aligned to the first position in B since more members of B contain the pattern CX. In the sequence-block alignment of the shuffled $\widetilde{A}$ to B, the C element in $\widetilde{A}$ can only be aligned to *one* of the two positions of the C element in B as the multiple alignment of B is fixed. Since few members of B contain a C element at the second position, the C element will contribute a value close to its contribution for the unshuffled A only when it falls near the first position. The event of falling near the *first* position is rarer than falling near the first *or* second position, so in fewer shufflings of A will the C element contribute a value close to its contribution for the unshuffled score. Thus with the blockwise statistic, the shuffled score will deviate from the unshuffled score more often than with the summed pairwise statistic, and the estimate of the significance of homology obtained from the blockwise statistic will tend to exceed the estimate obtained from the summed pairwise statistic.

[6]The C elements might also not be aligned because of known structural information.

2.2.4 Generating a shuffled sequence

The actual generation of a random sequence $\widetilde{A}$ is done as follows. To randomly select a sequence uniformly from the set of sequences with the same character frequency as A, we generate a random permutation of the elements of A by the following procedure. We write $A[i]$ for the ith character of A, and assume that a procedure Random(i, n) is available that generates a random integer uniformly in the range $[i, n]$.

procedure Permute(A, n) **begin**
  for $i := 1$ to n **do**
    Swap $A[i]$ with $A[\text{Random}(i, n)]$.
end

Assuming each call to Random takes $O(1)$ time, this generates a permutation of the elements of A uniformly over the $n!$ possible permutations in $O(n)$ time. To see that this selects a sequence uniformly over the set of sequences with the same character frequency as A, note that the number of permutations of elements of A that correspond to any fixed sequence $\widetilde{A}$ is the same, so every possible $\widetilde{A}$ is output with equal probability.

It is worth mentioning that it is also possible to uniformly sample over all sequences having the same frequency of substrings of length 2, 3, or any fixed length, in $O(n)$ time, as shown by Kandel, Matia, Unger and Winkler [23], though this is significantly more involved.

2.3 Evaluating the methods

Three sets of multiple sequence alignments of known proten families (hemoglobins, glutathione peroxidases and bacterial cellulose binding domains) were used to test our approach, each involving the comparison of a single query sequence with the prealigned family. In these examples, the sequences in the families each represent a set of known homologues, and the query represents a suspected or hypothetical member of that set. In addition, as a control experiment to assess the accuracy of this method, several comparisons involving query sequences that are presumably unrelated to these families were also conducted.

The first step in this approach is the input or generation of the multiple alignment of the sequences to which the query is to be compared. These may be aligned using any existing program for multiple sequence alignment, or a published alignment may be reproduced precisely for this purpose. It is worth emphasizing that *any* multiple alignment can be used as input, including alignments that have undergone extensive manual refinements. (This is an important point, because most multiple alignment experts believe such manual adjustments are necessary, given the limitations of current multiple alignment algorithms.) Several of our tests are based on published examples from other research groups, from which the multiple alignments were scanned and converted to text in order to permit a precise comparison to the published results.

Then, using the sequence-block alignment algorithm, optimal alignments of each of the query sequences with the respective target multiple alignments were produced for all three of the protein families considered, and randomly shuffled versions of the query sequences were generated and aligned for the significance calculation, as described in Section 2. The computed blockwise shuffling statistic Z^B was then compared to the pairwise shuffling statistics Z^C, Z^M, and Z^S computed for the query sequences vs. the individual sequences in the respective families.

3 Results

Results of five analyses used to test the program by aligning related and unrelated queries with a given family are shown in Figures 2-4 and Tables 1-3, and summarized in Tables 4-5, and are now individually described. We note that when we quote a shuffling statistic, or "Z score," we follow the convention of stating it in units of standard deviation (abbreviated SD). The reader should keep in mind, however, that this number cannot be interpreted as a percentile rank of the optimal alignment, or "p value," since the distribution of alignment scores for randomly shuffled queries is not known. In particular, shuffling statistics should not be translated into percentile ranks by *assuming* a normal distribution. The actual distribution of random alignment scores is a subject that clearly merits further study.

Example 1 Endoglucanase TreEgl II vs. bacterial cellulose binding domains

The alignment generated by sequence-block alignment (Figure 2) of the TreEglII sequence vs. bacterial cellulose binding domains (CBD) is identical to the alignment published by Meinke et al. [28], except for the placement of a single gap, indicated by '#' above the alignment.

The statistical significance of the TreEglII-CBD similarity, reported by Meinke et al. [28] as Z scores computed by what we call the "conventional pairwise" method, are shown in Table 1. These pairwise Z scores are fairly low, falling in the range of 1.4 to 3.3 SD, with a median of 2.9 SD, which puts the authors in a weak position to argue that TreEglII may belong to CBDs family and share a similar functionality, based on sequence alignment alone. As reported in Table 4, the blockwise shuffling statistic of 4.0 SD for TreEglII vs. the CBD alignment demonstrates that the significance of the sequence similarity is considerably higher than the pairwise calculations suggest. The distribution of the computed blockwise random scores is plotted in Figure 5.

Example 2 A putative selenoprotein encoded in HIV-1 vs. glutathione peroxidases

Taylor et al. [38] recently identified a potential virally-encoded selenoprotein in human immunodeficiency virus type 1 (a hypothetical *env-fs* gene in HIV-1) as a probable homologue of glutathione peroxidase, the prototypical mammalian selenoprotein. Taking the sequence encoded overlapping the HIV-1 *env* gp41 region as the query vs. an alignment of glutathione peroxidases, the significance of the similarity is computed as 5.2 SD by blockwise shuffling, vs. 2.5 SD median score by conventional pairwise comparisons (range 1.9 to 4.2; Table 2(a)). The alignment between *env-fs* and glutathione peroxidases generated by sequence-block alignment has already been published and is not reproduced here [38]. The distribution of the computed blockwise random scores is plotted in Figure 6.

Example 3 A novel open reading frame of hepatitis C virus vs. glutathione peroxidases

We have identified another virally-encoded glutathione peroxidase module in hepatitis C virus, which like the HIV-1 example above, is also encoded in a novel ORF in the -1 reading frame overlapping a known coding region, NS-4; thus we call this hypothetical gene of HCV NS4-fs. As shown in Table 2(b) and in the alignment of Figure 3, the significance of the similarity is computed as 6.2 SD by blockwise shuffling, vs. 3.8 SD median score by conventional pairwise comparisons (range 1.6 to 6.4; Table 2(b)). The distribution of the computed blockwise random scores is plotted in Figure 7.

Example 4 Myoglobin vs. hemoglobins

The example of myoglobin vs. a family of hemoglobins is shown in Figure 4; because of their similar structure and function, these are believed to have evolved from a common ancestor, but share low percent identity of amino acid sequence between these different members of the family [5]. The alignment of myoglobin vs. hemoglobins generated by sequence-block alignment (Figure 4) exactly reproduces the previously published alignment. The pairwise Z scores for myoglobin listed in Table 3 and summarized in Table 4 were calculated using our own software, because they were not calculated in the original paper. In this case also, significance scores are considerably improved by blockwise comparison, which gave a shuffling statistic of 10.8 SD (Table 4), vs. 8.1 SD median score for the conventional pairwise comparison (range 7.2 to 9.2; Table 3). The distribution of the computed blockwise random scores is plotted in Figure 8.

Example 5 A control experiment

We carried out six control tests of the blockwise shuffling statistic by aligning a query sequence against the alignments of presumably unrelated protein families. The results are shown in Table 5. All significance scores are below 1 SD, indicating that this method does not generate false positive significance scores when a query sequence does not share similarity with a given family.

4 Discussion

When new genomic sequences are obtained, the first question is usually whether or not the novel sequence has any similarity to existing sequences in any well-known family, which can be determined by database searching. For relating a sequence of interest to sequences in such a specific family, the general approach is then to align the query sequence with the sequences in the target set, by either pairwise or multiple alignment. However, while methods exist for the statistical assessment of similarities between pairs of sequences, no comparable method has been previously developed for comparisons of a probe sequence to a set of representative sequences of a given class. This is highly desirable, however, because multiple sequence alignments provide more information than a pairwise alignment, and are widely used for identifying conserved functional domains, predicting secondary and tertiary structure, estimating evolutionary distances, and in general evaluating evolutionary relationships between the sequence of interest and members of the given family. The advantage of multiple sequence alignment has long been known [9] when pairwise similarity suggests homology but is not sufficiently strong to be considered significant on its own. In brief, the shuffling statistic computed for a query vs. an aligned family of sequences should give a more reliable assessment of significance of similarity than that computed from pairwise comparisons alone.

The results of applying the method to the sample data sets confirm our expectation that significance scores computed by the traditional pairwise shuffling procedure can underestimate the true significance of the sequence similarity between a sequence and family of sequences.

```
TreEglIIa   GAYCIVDIHNYARWNGGIIGQGGPTNAQFTSL..WSQLASKYASQSRV...WFGIMNEPH
TreEglIIb   GAYCIVDIHNYARWNGGIIGQGGPTNAQFTSL..WSQLASKYASQSRV...WFGIMNEPH
CfiCenA     APGCRVDYAVTNQWPGGFGANVTITNLG.DPVSSWK.LDWTYTAGQRIQQLWNGTASTNG
CfiCex      PAGCQVLWGV.NQWNTGFTANVTVKNTSSAPVDGWT.LTFSFPSGQQVTQAWSSTVTQSG
CfiCenB     TPSCTVVYS.TNSWNVGFTGSVKITNTGTTPL.TWT.LGFAFPSGQQVTQGWSATWSQTG
CflClfX     TGSCKVEYNASS.WNTGFTASVRVTNTGTTALNGWT.LTFPFANGQTVQQGWSADWSQSG
PflEglA     GGNC..QYVVTNQWNNGFTAVIRVRNNGSSAINRWS.VNWSYSDGSRITNSWNANVTGNN
PflEglB     AAVC..EYRVTNEWGSGFTASIRITNNGSSTINGWS.VSWNYTDGSRVTSSWNAGLSGAN
PflXynA     TATC..SYNITNEWNTGYTGDITITNRGSSAINGWS.VNWQYAT.NRLSSSWNANVSGSN
BfiEndl     GALK.AEYTI.NNWGSGYQVLIKVKNDSASRVDGWT.LKISKSEV.KIDSSWCVNIAEEG
MbiCelA     GRACEATYALVNQWPGGFQAEVTVKNTGSSPINGWT.VQWTLPSGQSITQLWNGDLSTSG

                                #        #
TreEglIIa   ...DVNINTWAATVQEVVTAIRNAGATSQ.FISLPGNDWQSAGAFISDG
TreEglIIb   ...DVNINTWAATVQEVVTA.IRNAGATSQFISLPGNDWQSAGAFISDG
CfiCenA     GQVSVTSLPWNGSIPTGGTASFGFNGSWA.GSNPTPASFSLNGTTCTGT
CfiCex      SAVTVRNAPWNGSIPAGGTAQFGFNGSHT.GTNAAPTAFSLNGTPCTVG
CfiCenB     TTVTATGLSWNATLQPGQSTDIGFNGSHP.GTNTNPASFTVNGEVCG..
CflClfX     TTVTAKNAAWNGSLAAGQTVDIGFNGAHN.GTNNKPASFTLNGATCTVG
PflEglA     PY.AASALGWNANIQPGQTAEFGFQGTKGAGSRQVPA...VTGSVCQ..
PflEglB     PY.SATPVGWNTSIPIGSSVEFGVQGNNGSSRAQVPA...VTGAICGGQ
PflXynA     PY.SASNLSWNGNIQPGQSVSFGFQVNKNGGSAERPS...VGGSICSGS
BfiEndl     GYYVITPMSWNSSLEPSASVDFGIQGS...GS.IGTS...VNISVQ...
MbiCelA     SNVTVRNVSWNGNVPAGGSTSFGFLGS...GTGQLSSS.....ITCSAS
```

Figure 2 Alignment of the catalytic domain of *endoglucanase* II of *Trichoderma reesi* (TreEglII) vs. a family of bacterial cellulose binding domains (CBD). Our alignment, shown as TreEglIIa, is identical to the previously published alignment, shown as TreEglIIb [28], except for the placement of a single gap at the locations indicated by '#'. The parameters used are the PAM250 matrix, gap startup 3, and gap extension 0.5.

Table 1 Traditional shuffling statistics for pairwise comparison of TreEglII vs. each of the CBS sequences [28]. Parameters used are the PAM250 matrix, 100 randomizations, gap startup 3, and gap extension 0.7.

Name	Z_{100}
CfiCenA	3.27
CfiCex	2.02
CfiCenB	3.17
CflClfX	1.95
PflEglA	3.27
PflEglB	3.17
PflXynA	2.25
PfuEnd1	1.40
MbiCelA	2.90

```
              #
HCV      VQVASPCGLLGEAHVELH..QRDTVSSRLVHPAWKPRNSI...TDGIHS
S05317   ENVASLCGTTIRDYTEMNDLQKRLG...LVVLGF.PCNQFGHQVYGARW
S18912   VNVATYCGLTA.QYPELNALQEELKPYGLVVLGF.PCNQFGKQEPGDNK
A47367   VNVATYCGLTI.QYPELNALQEDLKPFGLVILGF.PCNEFGKQEPGDNL
Jq0476   VNVASYCGLTG.QYIELNALQEELAPFGLVILGF.PCNQFGKQEPGENS
A55086   VNVASYCGLTD.QYLELNALQEELGPFGLVILGF.PSNQFGKQEPGENS
Jx0176   VNVASYCGLTD.QYLELNALQEELGPFGLVILGF.PCNQFGKQEPGENS
Jx0280   VNVASYCGLTG.QYVELNALQEELEPFGLVILGF.PCNQFGKQEPGENS
B40464   VNVASFCGLTA.TYPELNTLQEELRPFNVSVLGF.PCNQFGKQEPGKNS
A45207   ENVASLCGTTTRDFTQLNELQCRF.PRRLVVLGF.PCNQFGHQENCQNE
S21119   ENVASLCGTTTRDYTEMNDLQKRLGPRGLVVLGF.PCNQFGHQENGKNE
S06304   ENVASLCGTTVRDYTQMNELQRRLGPRGLVVLGF.PCNQFGHQENAKNE
S03723   ENVASLCGTTVRDYTQMNELQERLGPRALVVLGF.PCNQFGHQENAKNE
A53395   TNVASQCGKTEVNYTQLVDLHARYAECGLRILAF.PCNQFGRQEPGSDA
S20501   VNVASQCGLTNSNYTDLTEIYKKYKDQGLEILAF.PCNQFGGQEPGSIE
```

Figure 3 Hypothetical hepatitis C virus *NS4-fs* sequence aligned with various glutathione peroxidases of multiple alignment, aligned using the `pileup` program in GCG [35]. '`C`' is either cysteine or selenocysteine at the position shown by '`#`'. The parameters used are the Blosum62 matrix, gap startup 5, and gap extension 1.

Table 2 Traditional shuffling statistics for pairwise comparison between various glutathione peroxidases (whose names are listed by PIR accession number) and (a) HIV-1 *env-fs* gene [38] (parameters used are Blosum62 matrix, 100 randomizations, gap startup 5, and gap extension 2.5); (b) HCV *NS4-fs* gene (parameters used are the Blosum62 matrix, 100 randomizations, gap startup 5, and gap extension 1).

Name	(a) HIV-1 *env-fs* Z_{100}	(b) HCV *NS4-fs* Z_{100}
S05317	2.09	1.96
S18912	2.64	4.30
A47367	2.63	3.54
Jq0476	3.25	6.35
A55086	3.99	4.58
Jx0176	3.62	4.23
Jx0280	2.85	6.16
B40464	4.17	3.95
A45207	1.98	1.85
S21119	1.96	2.42
S06304	2.30	1.59
S03723	2.43	1.92
A53395	1.90	3.45
S02501	1.99	3.96

```
Myoglobin       G.LSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASE
Hemoglobin a    V.LSPADKTNVKAAWGKVGAHAGQYGAEALERMFLSFPTTKTYFPHF.DLSH.....GSA
Hemoglobin b    VHLTPEEKSAVTALWGKVN..VDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP
Hemoglobin g    GHFTEEDKATITSLWGKVN..VEDAGGETLGRLLVVYPWTQRFFDSFGNLSSASAIMGNP
Hemoglobin d    VHLTPEEKTAVNALWGKVN..VDAVGGEALGRLLVVYPWTGRFFESFGDLSSPDAVMGNP
Hemoglobin e    VHFTAEEKAAVTSLWSKMN..VEEAGGEALGRLLVVYPWTQRFFDSFGNLSSPSAILGNP

Myoglobin       DLKKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKH
Hemoglobin a    QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL
Hemoglobin b    KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF
Hemoglobin g    KVKAHGKKVLTSLGDAIKHLDDLKGTFAQLSELHCDKLHVDPENFKLLGNVLVTVLAIHF
Hemoglobin d    KVKAHGKKVLGAFSDGLAHLDNLKGTFSQLSELHCDKLHVDPENFRLLGNVLVCVLARNF
Hemoglobin e    KVKAHGKKVLTSFGDAIKNMDNLKPAFAKLSELHCDKLHVDPENFKLLGNVMVIILATHF

Myoglobin       PPDFGADAQGAMNKALELFRKDMASNYKELGFQG
Hemoglobin a    PAEFTPAVHASLDKFLASVSTVLTSKYR......
Hemoglobin b    GKEFTPPVQAAYQKVVAGVANALAHKYH......
Hemoglobin g    GKEFTPQVQASWQKMVTGVASALSSRYH......
Hemoglobin d    GKEFTPEMQAAYQKVVAGVANALAHKYH......
Hemoglobin e    GKEFTPEVQAAWQKLVSAVAIALAHKYH......
```

Figure 4 Alignment of myoglobin vs. a prealigned hemoglobin sequences. The sequence-block alignment is identical to the previously published alignment [5]. The parameters used are the Blosum62 matrix, gap startup 7 and gap extension 1.

Table 3 Traditional shuffling statistics for pairwise comparison of myoglobin vs. hemoglobins. The parameters used are the Blosum62 matrix, 100 randomizations, gap startup 7, and gap extension 1.

Name	Z_{100}
Hemoglobin α	8.18
Hemoglobin β	7.17
Hemoglobin γ	9.17
Hemoglobin δ	7.74
Hemoglobin ϵ	8.13

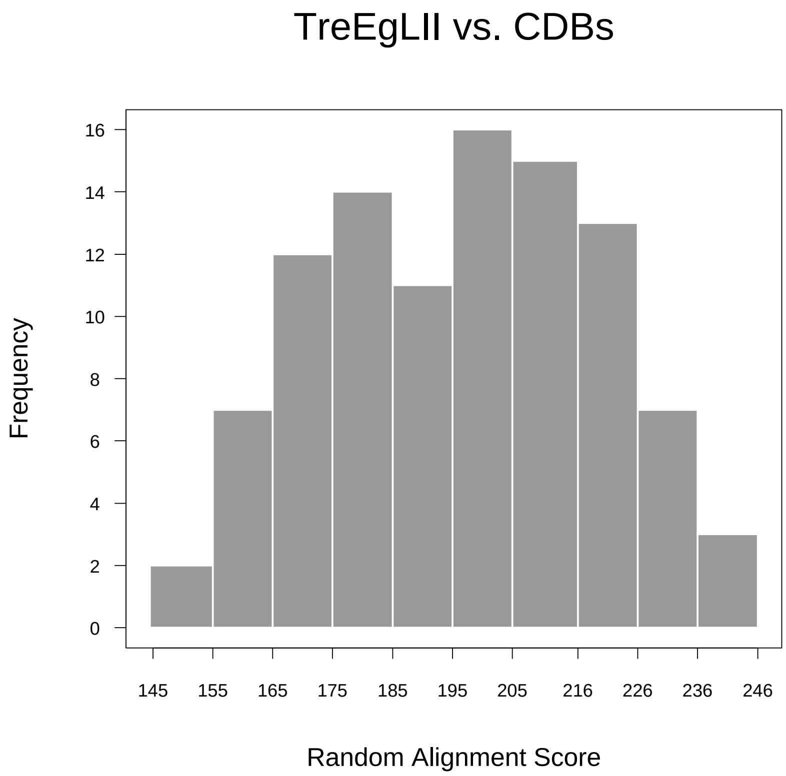

Figure 5 Histogram of sequence-block alignment scores for 100 randomizations of TreEglII against the CBD family. Parameters used for aligning the shuffled TreEglII are identical to those given in Table 1. The alignment score of the unshuffled TreEglII vs. the CBD family is 106, the mean score is 196, and the standard deviation is 22.7, which gives a shuffling statistic of $Z^B_{100} = 3.96$.

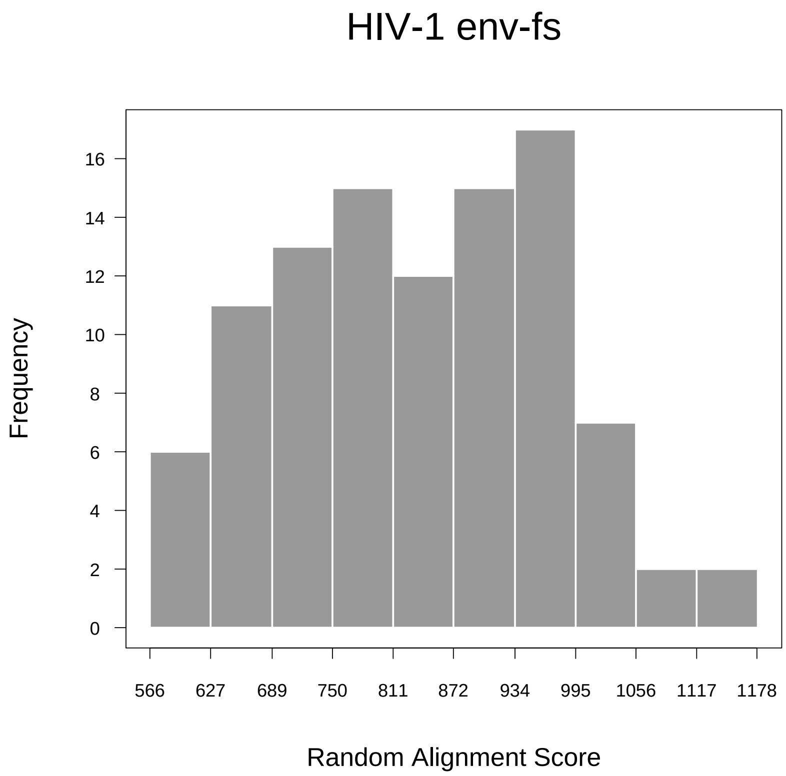

Figure 6 Histogram of sequence-block alignment scores for 100 randomizations of HIV-1 *env-fs* against the GPX family. Parameters used for aligning the shuffled HIV-1 *env-fs* are identical to those given in Table 2(a). The alignment score of the unshuffled HIV-1 *env-fs* vs. the GPX family is 135, the mean score is 835, and the standard deviation is 134, which gives a shuffling statistic of $Z^B_{100} = 5.22$.

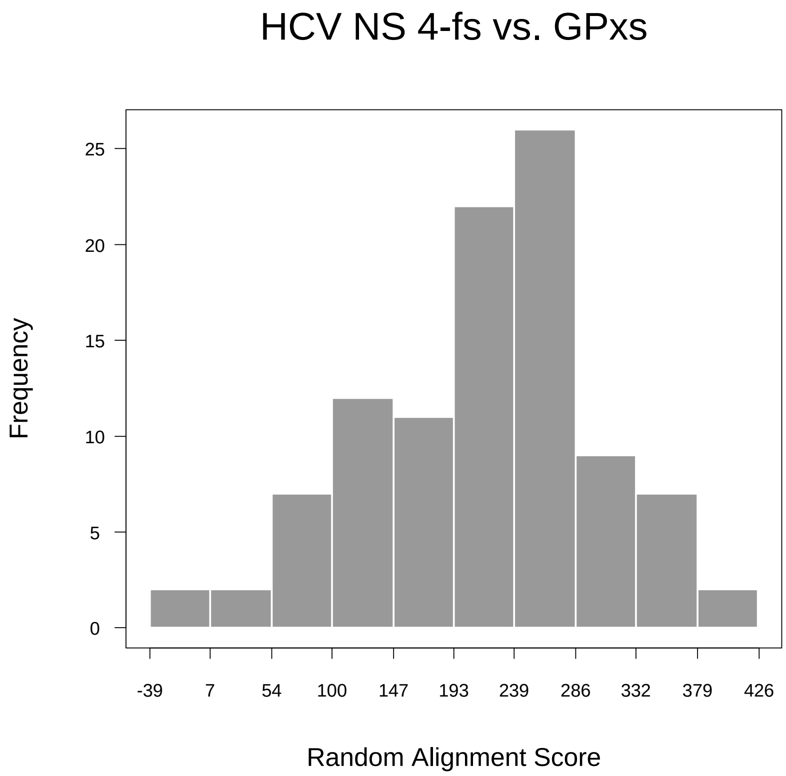

Figure 7 Histogram of sequence-block alignment scores for 100 randomizations of HCV *NS4-fs* against the GPX family. Parameters used for aligning the shuffled HCV *NS4-fs* are identical to those given in Table 2(b). The alignment score of the unshuffled HCV *NS4-fs* vs. the GPX family is -298, the mean score is 204, and standard deviation is 80.6, which gives a shuffling statistic of $Z^B_{100} = 6.23$.

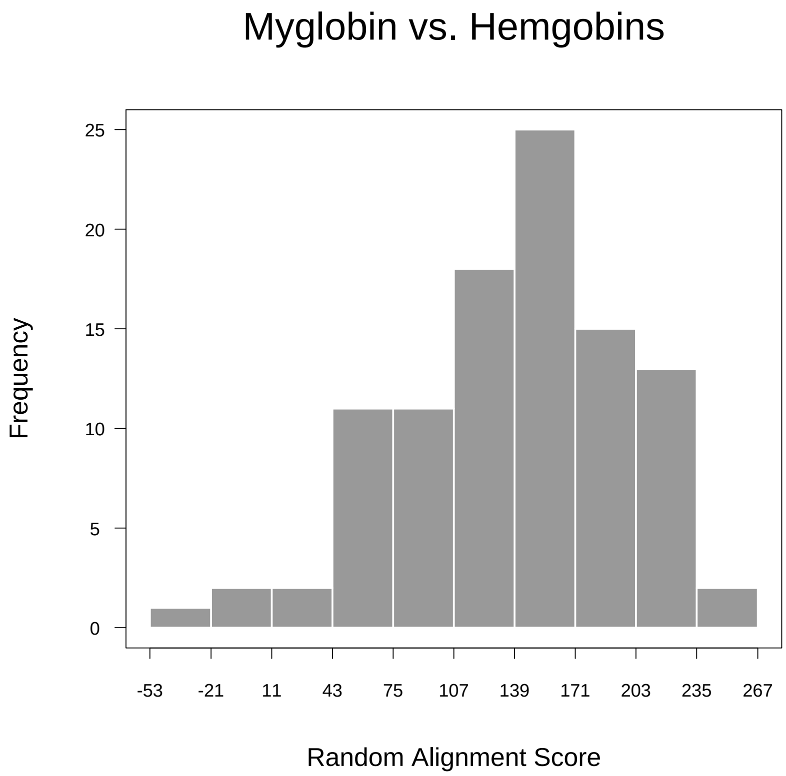

Figure 8 Histogram of sequence-block alignment scores for 100 randomizations of myoglobin against the hemoglobin family. The parameters used for aligning the shuffled myoglobin are identical to those given in Table 3. The alignment score of the unshuffled myoglobin vs. the hemoglobin family is -503, the mean score is 138, and the standard deviation is 59.5, which gives a shuffling statistic of $Z^B_{100} = 10.8$.

Table 4 Pairwise versus blockwise shuffling statistics for all families from Tables 1-3. For all experiments, the Z^C, Z^M, Z^S, and Z^B scores were obtained by 100 randomizations. (See Section 2.2 for definitions of the statistics Z^C, Z^M, Z^S, and Z^B.) In each row, the parameters used to compute the sequence-block alignments are identical to those in Tables 1-3.

	Pairwise						Blockwise
	Conventional						
	Min	Mean	Median	Max	Modified	Summed	
Query vs. family		Z^C_{100}			Z^M_{100}	Z^S_{100}	Z^B_{100}
TreEgl II vs. CBDs	1.40	2.60	2.99	3.27	2.16	3.27	3.98
HIV-1 *env-fs* vs. GPXs	1.90	2.70	2.53	4.17	2.58	4.17	5.22
HCV *NS4-fs* vs. GPXs	1.59	3.59	3.75	6.35	3.04	5.54	6.22
Myglobin vs. hemoglobins	7.17	8.08	8.13	9.17	7.99	8.74	10.8

Table 5 Control experiments on the blockwise shuffling statistic Z^B for nonhomologous query sequences vs. the three families from Tables 1-3. If the query sequence was longer than the alignment of the sequence family, the query was trimmed to the consensus length for the family. For all experiments, 100 randomizations were performed. The parameters for the myoglobin and HCV *NS4-fs* (S20502) experiments are the Blosum62 matrix, gap startup 6 and gap extension 6. The parameters for the TreEglII experiments are the PAM250 matrix, gap startup 3, and gap extension 0.5.

Query vs. family	Z^B_{100}
Myoglobin vs. GPXs	0.09
Myoglobin vs. CBDs	0.99
HCV *NS4-fs* vs. hemoglobins	0.78
HCV *NS4-fs* vs. CBDs	0.11
TreEglII vs. GPXs	0.15
TreEglII vs. hemoglobins	0.24

In several examples of distant homology, the blockwise shuffling method establishes that the sequence similarity is substantially more significant than that predicted by the median of the conventional pairwise significance scores, which in two cases (Examples 1 and 2) were borderline (less than 3 SD). For closer relatives, exemplified by a myoglobin sequence vs. hemoglobins, significance scores were also improved, to 10.8 SD by blockwise comparison, vs. 8.1 SD median score for conventional pairwise comparison (range 7.2 to 9.2). Overall, significance scores assessed by the new blockwise method exceeded the median pairwise significance score by 2.2 SD on average. This suggests that the method is consistently more sensitive than pairwise sequence comparison, and may be particularly useful in evaluating a hypothesis of distant homology when pairwise similarity is borderline.

In summary, the sequence-block alignment algorithm presented here, which finds an optimal alignment of a query sequence vs. a prealigned multiple sequence set, can be used to rigorously evaluate the statistical significance of the similarity, by what we call the blockwise shuffling statistic. We believe this method will become a valuable tool in the field of computational biology.

Acknowledgements

Source code for software that computes optimal sequence-block alignments and calculates the blockwise shuffling statistic (written in C++ for the UNIX operating system) is available via anonymous `ftp` from `gene.rx.uga.edu/pub`.

References

[1] Altschul, S.F. "Gap costs for multiple sequence alignment." *Journal of Theoretical Biology* 138, 297–309, 1989.

[2] Altschul, S.F. and B.W. Erickson. "Significance of nucleotide sequence alignments: a method for random sequence permutations that preserves dinucleotide and codon usage." *Molecular Biology and Evolution* 2, 526–538, 1985.

[3] Bailey, T.L. and M. Gribskov. "Score distributions for simultaneous matching to multiple motifs." *Journal of Computational Biology* 4, 45–59, 1997.

[4] Carrillo, H. and D. Lipman. "The multiple sequence alignment problem in biology." *SIAM Journal on Applied Mathematics* 48, 1073–1082, 1988.

[5] Creighton, T.E. *Proteins*, 93–131, 1983.

[6] Dayhoff, M.O., R.M. Schwartz and B.C. Orcutt. "A model of evolutionary change in proteins." In *Atlas of Protein Sequence and Structure* 5:3, M.O. Dayhoff editor, 345–352, 1978.

[7] Doolittle, R.F. "Similar amino-acid sequences: chance or common ancestry." *Science* 214, 149–159, 1981.

[8] Feng, O.F., M.S. Johnson and R.F. Doolittle. "Aligning amino acid sequences: comparison of commonly used methods." *Journal of Molecular Evolution* 21, 112–125, 1985.

[9] Fitch, W.M. "Further improvements in the method of testing for evolutionary homology among proteins." *Journal of Molecular Biology* 49, 1–14, 1970.

[10] Fitch, W.M. "Random sequences." *Journal of Molecular Biology* 163, 171–176, 1983.

[11] Galil, Z. and R. Giancarlo. "Speeding up dynamic programming with applications to molecular biology." *Theoretical Computer Science* 64, 107–118, 1989.

[12] Godzik, A., A. Kolinski and J. Skolnick. "Topology fingerprint approach tto the inverse protein folding problem." *Journal of Molecular Biology* 227, 227–238, 1990.

[13] Godzik, A. and J. Skolnick. "Sequence structure matching in globular proteins: Application to supersecondary and tertiary structure determination." *Proceedings of the National Academy of Sciences USA* 89, 12098–12102, 1992.

[14] Godzik, A., A. Kolinksi and J. Skolnick. "De novo and inverse folding predictions of protein structure and dynamics." *Journal of Computer-Aided Molecular Design* 7, 397–438, 1993.

[15] Gotoh, O. "An improved algorithm for matching biological sequences." *Journal of Molecular Biology* 162, 705–708, 1982.

[16] Gotoh, O. "Optimal alignment between groups of sequences and its application to multiple sequence alignment." *Computer Applications in the Biosciences* 9:3, 361–370, 1993.

[17] Gotoh, O. "Further improvement in methods of group-to-group sequence alignment with generalized profile operations." *Computer Applications in the Biosciences* 10:4, 379–387, 1994.

[18] Gribskov M. and R.R. Burgess. "Sigma factors from *E. coli*, *B. subtilis*, phage SP01 and phage T4 are homologous proteins." *Nucleic Acids Research* 14, 6745–6763, 1986.

[19] Gribskov, M., A.D. McLachlan, and D. Eisenberg. "Profile analysis: detection of distantly related proteins." *Proceedings of the National Academy of Sciences USA* 84, 4355–4358, 1987.

[20] Gupta, S., J. Kececioglu and A. Schäffer. "Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment." *Journal of Computational Biology* 2:3, 459-472, 1995.

[21] Henikoff, S. and J.G. Henikoff. "Amino acid substitution matrices from protein blocks." *Proceedings of the National Academy of Sciences USA* 89, 10915–10919, 1992.

[22] Henikoff, S. and J.G. Henikoff. "Embedding strategies for effective use of information from multiple sequence alignments." *Protein Science* 6, 698–705, 1997.

[23] Kandel, D., Y. Matias, R. Unger and P. Winkler. "Shuffling biological sequences." *Discrete Applied Mathematics* 71, 171–185, 1996.

[24] Kececioglu, J.D. and W.Q. Zhang. "Aligning alignments." To appear in Proceedings of the 9th *Symposium on Combinatorial Pattern Matching*, July 1998.

[25] Lipman, D.G., S.F. Altschul and J.D. Kececioglu. "A tool for multiple sequence alignment." *Proceedings of the National Academy of Sciences USA* 86, 4412-4415, 1989.

[26] Lipman, D.J. and W.R. Pearson. "Rapid and sensitive protein similarity searches." *Science* 227, 1435–1441, 1985.

[27] Lipman, D.J. and W.J. Wilbur. "On the statistical significance of nucleic acid similarities." *Nucleic Acid Research* 12, 215–227, 1984.

[28] Meinke, A., N.R. Gilkes, D.G. Kilburn, R.C. Miller and R.A.J. Warren. "Bacterial cellulose-binding domain-like sequencs in eucaryotic polypeptides." *Protein Sequences and Data Analysis* 4, 349–353, 1991.

[29] Miller, W. and E.W. Myers. "Sequence comparison with concave weighting functions." *Bulletin of Mathematical Biology* 50, 97–120, 1988.

[30] Myers, E.W. and W. Miller. "Optimal alignments in linear space." *Computer Applications in the Biosciences* 4:1, 11–17, 1988.

[31] Needleman, S.B. and C.D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *Journal of Molecular Biology* 48, 443–453, 1970.

[32] Pearson, W.R. "Rapid and sensitive sequence comparison with FASTP and FASTA." *Methods in Enzymology* 183, 63–98, 1990.

[33] Pearson, W.R. and W. Miller. "Dynamic programming algorithms for biological sequence comparison." *Methods in Enzymology* 210, 575–601, 1992.

[34] Pietrokovski, S. "Searching databases of conserved sequence regions by aligning protein multiple-alignments." *Nucleic Acids Research* 24:19, 3836–3845, 1996.

[35] Program Manual for the Wisconsin Package, Version 8.1, September 1994. Genetics Computer Group, 575 Science Drive, Madison, WI, 53711.

[36] Sellers, P.H. "On the theory and computation of evolutionary distances." *SIAM Journal on Applied Mathematics* 26, 787–793, 1974.

[37] Smith, T.F. and M.S. Waterman. "The statistical distribution of automated DNA sequence analysis." *Journal of Molecular Evolution* 18, 38–46, 1981.

[38] Taylor, E.W., A. Bhat, R. Nadimpalli, W. Zhang and J.D. Kececioglu. "HIV-1 encodes a sequence overlapping env gp41 with highly significant similarity to selenium dependent glutathione peroxidases." *Journal of Acquired Immune Deficiency Human Retrovirology* 15:5, 393–394, 1997.

[39] Wang, L. and T. Jiang. "On the complexity of multiple sequence alignment." *Journal of Computational Biology* 1:4, 337–348, 1994.

[40] Waterman, M.S. *Introduction to Computational Biology.* Chapman and Hall, London, 1995.