

e-mail: ecrocke@cs.washington.edu

A hybrid scoring metric for protein multiple alignment

Emily Rocke `ecrocke@cs.washington.edu`

University of Washington

The date of receipt and acceptance will be inserted by the editor

Abstract Previous algorithms for motif discovery and protein alignment have used a variety of scoring metrics, each specialized to find certain types of similarity in preference to others. Here we present a novel scoring metric that combines the relative entropy score with a sensitivity to amino acid similarities, producing a score that is highly sensitive to the types of weakly-conserved patterns that are typically seen in proteins. We use several algorithms to investigate the performance of the hybrid score compared to existing scoring metrics. We conclude that the hybrid is more sensitive than previous protein scoring metrics, both in the initial detection of a weakly conserved region of similarity, and given such a similarity, in the detection of weakly-conserved instances.

1 Introduction and previous work

Of the computations used to analyze the vast quantity of biological sequence data becoming available, a significant number involve detection or analysis of similarities between different DNA and protein sequences. This may take the form of sequence alignment (arranging two or more sequences so that corresponding residues are placed in alignment), pattern matching (searching a database for approximate instances of a given pattern), or pattern discovery (searching a database for a set of substrings that are similar to one another). In any of these cases, however, some score metric must be established, to guide the progress of the algorithm and to measure whether the final result is interesting enough to report.

1.1 Two-sequence metrics

In the case of evaluating a match between two sequences, the most popular metric is that used by the well-known dynamic programming algorithm generally called Smith-Waterman or Needleman-Wunsch alignment [15,18]. The value of aligning any pair of residues is independent of position, depending only on the two residues, with the weight often determined by empirical evidence as with the PAM [3] or BLOSUM [16,8] matrices. In addition, there are user-determined penalties for introducing a gap in a sequence and for extending it. The metric maximizes the sum of this score over all positions. For the purposes of this introduction, this metric will be abbreviated as the *sos/ag* (sum-of-similarities/ affine gap) metric.

Because of its position-independence, this metric is very tractable. It lends itself to efficient dynamic programming, each pairing of residues may be given a different score (as with PAM or BLOSUM) without harming complexity, and the correspondence between scores and *p*-values has largely been solved (e.g. [10,4,11]). The same position independence also causes some of the flaws commonly observed in Smith-Waterman

alignments: for example, serious over-penalization of long gaps, and poor alignment accuracy when the well-conserved segments of the sequences are short compared to the overall sequence lengths.

Most software treating two-sequence alignment accepts these properties of the sum-of-similarities/ affine gap metric, working on faster algorithms to arrive at a high score under the fixed metric rather than concentrating on the metric itself. For example, BLAST uses short exact-sequence matches to find candidate matches, but evaluates the candidates on the sos/ag score against the query sequence.

1.1.1 Segment-to-segment score An exception is the alignment program Dialign, which uses an alternative two-sequence metric called a *segment-to-segment* comparison [14, 13]. The Dialign algorithm for two-sequence alignment uses dynamic programming to string together shorter ungapped matches. The ungapped matches are initially given a sum-of-similarities score, with a fixed value of 1 for each match and 0 for each mismatch, and then converted into a p -value before the second dynamic programming step.

This metric addresses the above problems with the Smith-Waterman algorithm; there is no explicit gap penalty, so that long gaps are not over-penalized, and well-conserved regions can be strung together despite long regions of dissimilarity.

However, the segment-to-segment metric has two limitations that restrict it to certain contexts of use. One is that the two sequences must be enough conserved that sufficiently long segments are preserved without gaps. In practice, for long sequences, this is often the case even if they are considerably diverged. A more important restriction is that, unlike the sos/ag metric, which can be used for either global alignment or local match-discovery, this metric is primarily useful for aligning the entire length of two sequences.

1.2 Multiple sequence metrics

When scoring an alignment of several sequences, the Smith-Waterman alignment takes time proportional to the product of the sequence lengths. Because it takes few sequences before this computation is prohibitively expensive, a straightforward multiple sequence alignment is usually impossible.

1.2.1 Sum-of-similarity equivalents There are several common multiple-sequence equivalents of the sos/ag score that can still be used for a variety of heuristic algorithms. These methods use the same pairwise similarities between residues as the sos/ag score, for example from a PAM matrix or from fixed match and mismatch scores, and combine these pairwise comparisons in various ways.

One method, the sum-of-pairs score, adds together the pairwise similarity between each pair of sequences at each position, for a total of t^2 comparisons per column, if t is the number of sequences in the alignment. A similar approach chooses a single sequence as the template, and scores each position by taking the sum of similarities between each sequence and the template at that position. The template sequence may be one of the sequences being aligned, their consensus sequence, or another sequence that was determined externally. For example, MultiPipMaker [6], a DNA multiple alignment program, takes the first input sequence to be privileged and pairwise aligns the other sequences against it. In this case, the metric being optimized is the sum of similarities to the lead sequence.

Both metrics allow some of the same flexibility as the pairwise score: match scores can be weighted according to an arbitrary matrix and the score is easy to compute for most algorithms. However, in practice neither measure is very sensitive to biologically realistic patterns.

1.2.2 Phylogeny-based score One improvement to the sum-of-pairs score involves first placing the sequences to be aligned in a phylogenetic relationship, and then using the sos/ag or another pairwise metric to greedily align the closest unconnected nodes. ClustalW, for example, first creates a phylogenetic tree using pairwise alignment distances, and then applies a sophisticated scoring metric during the hierarchical alignment. The pairwise metric is sensitive to secondary structure and evolutionary distance, giving the algorithm greater sensitivity.

This kind of metric is appropriate when a phylogenetic tree relates the candidate sequences, i.e., when aligning whole sequences (rather than pattern discovery or pattern matching) and when the sequences are closely enough related to determine an accurate phylogenetic relationship. In most contexts, a phylogenetic viewpoint is not useful for motif finding, except if the putative motif location is restricted to a family of phylogenetically related proteins (as in [1]).

In addition, even for those situations where a phylogeny applies, some other metric is usually necessary in the preprocessing stage to construct the tree. An algorithm like ClustalW could benefit from a more sensitive alignment algorithm during tree construction, to avoid mistakes in the phylogeny which will propagate to impact the accuracy of the alignment.

1.2.3 Segment-to-segment score Dialign’s “segment-to-segment” score extends to multiple dimensions by using the same p -value metric as in the two-sequence case on ungapped pairwise matches, adding weight to a pairwise match if it is also matched in other sequences. The algorithm then uses a greedy heuristic to take the most significant ungapped matches first.

The metric relies on there being enough significant ungapped matches to string together a correct alignment, which will be true only when the sequences are closely related. In section 7.1 we briefly discuss a method of extending the Dialign algorithm, using the metric proposed here as a subcomponent.

1.2.4 Relative entropy Another popular multiple sequence scoring metric, and the one this paper’s metric builds from, is the relative entropy score, as used by the Gibbs sampling algorithm introduced in [12]. This score rewards positions of the putative pattern that look very different from the background frequency by multiplying the ratio, for each residue, of the frequency of the residue at that position of the pattern to the frequency of that residue in general. The logarithm of the ratio is generally taken to keep the scores smaller in magnitude. The score is computed independently at each position, and the log ratios added together (or, equivalently, the ratios multiplied) to produce the score for the total pattern.

Mathematically, the score is expressed as

$$\sum_{\alpha \in \Sigma} p_{\alpha} \log\left(\frac{p_{\alpha}}{b_{\alpha}}\right),$$

where Σ is the alphabet, p_{α} is the fraction of the column made up of letter α , and b_{α} is the background frequency or prior probability, usually measured as the fraction of the database made up of letter α .

2 Problem

2.1 Tradeoffs

Any scoring metric for protein multiple alignment has advantages and disadvantages. Some of the difference between scoring functions in use comes from different ideas of how to represent the underlying biology and the likelihoods of different evolutionary events. For example, the decision of whether to use a PAM or BLOSUM scoring matrix falls into this category.

However, another factor goes into the selection of a scoring metric: for any given algorithm, some scoring metrics dovetail well with the algorithm, allowing efficiency, and others are difficult to realize. For example, although one may believe that the biology is better represented by an exponential falloff in increase of gap penalty with gap length, we typically use an affine gap penalty to perform Smith-Waterman alignment, precisely so that the score may be computed using an efficient dynamic program.

Another example of this phenomenon is the great prevalence of pattern discovery algorithms that find only ungapped patterns. Although biological reality includes many patterns whose instances vary in length, permission of gapped instances throws a confounding factor into many motif discovery algorithms and scoring schema.

A third example, discussed below, is sensitivity to amino acid similarities in protein alignment.

2.2 Amino acid alignment

Protein sequences are made up of amino acids, some subsets of which are highly chemically similar, while others are chemically very different. Given two related proteins or protein segments, similar amino acids are more likely to have replaced one another than dissimilar ones.

The alignment of well-conserved protein regions does not absolutely require a sensitivity to this fact. For example, the Dialign algorithm for sequence alignment, and the Gibbs sampler for motif discovery, perform quite well in practice while only regarding exact matches between amino acids, as long as the conservation is sufficiently high.

However, there is compelling evidence that matching can be more intelligent if it takes amino acid similarity into account. Indeed, nearly any two-sequence alignment program will use amino acid similarities, in the form of the PAM or BLOSUM matrices, since the results improve significantly in their biological realism.

The main reason the same unanimity is not seen for multiple sequence alignment is that, for many multiple sequence metrics with otherwise desirable properties, it is unclear how to include amino acid similarities.

2.3 Benefits of relative entropy

The relative entropy score is an example of a metric that has not typically been considered in the context of gapped alignment or of including amino acid similarities. Although both restrictions would seem to limit the metric in the protein patterns it can find, the metric continues to be popular, particularly in the context of Gibbs sampling. It should be noted here that, although Gibbs sampling and relative entropy were proposed to be used together in [12], the Gibbs sampling algorithm is trivially adaptable to any ungapped multiple-sequence metric.

The use of the relative entropy score, rather than, for example, a sum-of-pairs score which would allow accounting for amino acid similarities, is explained by its high sensitivity to subtle patterns. A great advantage of the metric is that it serves as an approximate measure of the probability of an alignment, based on the priors predicted by an independent random background model. In practical terms, it means that the relative entropy score can recognize more subtle patterns than a score that counts differences, as long as the subtle pattern is significantly different from the background.

As just one example, if half the instances of a motif have one residue α_i at position t , and the other half have residue α_j at that position, then the relative entropy score rewards column t highly as a percentage of maximum score. The sum-of-pairs (or other similarity counting) score is far slower to recognize the significance of this pattern. Although this is slightly ameliorated if residues α_i and α_j are considered to be related (a high matching value in the similarity matrix), the relative entropy score is still better at recognizing such patterns.

3 Approach

The new method proposed in this paper is designed to combine the strengths of the relative entropy score with a sensitivity to amino acid similarities. In addition, as in our previous proposal for gapped Gibbs sampling ([17]), the relative entropy score is modified to allow for a gapped alignment.

In essence, the proposed scoring metric adds weight to a residue if similar residues are in the same column. In order to describe the score more thoroughly, we begin with a description of the adaptation which allows the relative entropy score to deal with gaps, then describe how we allow sensitivity to amino acid similarities.

3.1 Adaptation to gaps

The adaptation to permit gaps is little changed from that described in [17], in which we propose to extend Gibbs sampling to the gapped alignment case. The most important addition to the regular relative entropy

score is that in every column, in place of each blank, we substitute a fractional occurrence of every residue α in proportion to its background frequency b_α . The score is then computed as above, adding these fractional residues ($b_\alpha \times$ the number of blanks in the column) to the tally of p_α .

For example, suppose the alphabet consisted of letters A, B, C, and D, all occurring equally often in the database, and that the aligned column to be scored is “AA-ABA,” where “-” represents a blank in the alignment. Then p_A would be 4.25 for the relative entropy computation, p_B would be 1.25, and C and D (neither of which appears explicitly in the column) would have $p_C = p_D = 0.25$.

The rationale behind this approach is that, with no remaining knowledge about a “missing” residue, i.e. gap, in an alignment, its possible values before deletion include any residue, and for lack of other information the best prior on this shape is the background frequencies.

More practically, this technique avoids the thorny problem of placing some kind of prior distribution on a gap. It has the pleasant side effect of penalizing gaps to a minor extent, although not sufficiently strongly to avoid the necessity for an additional gap penalty. A user-determined penalty, simply multiplied by the length of each gap, is subtracted from the combined score, although an affine gap penalty could also be used.

3.2 Amino acid similarities

Just as the idea of “fractional” residues is used to deal with gaps, a similar concept allows an accommodation of amino acid similarities. Suppose that for any pair of residues α_i and α_j , ($i \neq j$), we have a weight $w_{i,j}$ that measures how often α_j is substituted for α_i in related sequences. The value $w_{i,j}$ in some way reflects to what extent the residue α_j *might actually have been* residue α_i before a substitution.

Then let $w_{i,j}$ be the “credit” that α_i gets for each pairing of α_i with α_j . This credit is multiplied by the fraction of possible pairings which are in fact a pairing of α_i with α_j . Finally, this credit is given to α_i as if α_i itself occurred those extra fractional times:

$$p_{\alpha_i}^* = p_{\alpha_i} + \sum_{j \neq i} (p_{\alpha_i} \times p_{\alpha_j} \times w_{i,j}).$$

The new measures p_α^* are then used in the relative entropy calculation in place of the p_α . Note that whereas before, with fractional residues only to fill the blanks in a column, $\sum_{\alpha \in \Sigma} p_\alpha$ must add to exactly 1, the new sum $\sum_{\alpha \in \Sigma} p_\alpha^*$ may be greater than 1, although less than 2. This is a peculiarity of the method, but a benign one; the property that the p_α sum to 1 is not crucial to the metric’s use in measuring sequence similarity.

4 Implementation

4.1 Scoring function

The scoring function is implemented as described above. For the weights $w_{i,j}$, we use the intermediate values from the computation of the BLOSUM62 matrix [9] to give an empirical value for the proportion of pairs involving α_i that use α_j . This is a good estimate of $w_{i,j}$, the frequency with which α_j substitutes for α_i . However, any scoring system on amino acids could be substituted modularly in this function. A PAM matrix could be used, if preferred, or more interestingly a system such as Wu and Brutlag’s ([19]), where here one would weight a pair of amino acids based on whether (and perhaps how strongly) they participate in a substitution group together.

The gap penalty used in all experiments is $1.5 \times$ (gap length). This is applied after normalizing the relative entropy variant to an approximate z -score (mean and standard deviation determined via random sampling). As is so often the case, this gap penalty is determined through trial, error, and aesthetic, with a dash of whim. However, since all scores are normalized similarly, this gap penalty at least behaves consistently with all scoring functions tested and appears to remain appropriate at various pattern sizes and lengths.

4.2 Data

The tests of pattern discovery are run on 11 instances of the dihydrofolate reductase signature (Accession PS00075), the set which was available from PDBsum ([5]) as occurring in PDB entries. This motif was selected by eye from a small set of candidates as being a reasonable length, varying mildly in length (from 23 to 24 AA) but not so greatly that an ungapped method could not conceivably find them, and well-conserved but not so tightly as to make all scoring methods highly successful, which would make a distinction between them difficult.

The tests of pattern matching were run on 100 instances of the C2H2-type zinc finger motif (Accession PS00028), plus an additional 12 instances used to specify the pattern, all downloaded from SwissPROT ([7]). These instances were chosen entirely at random from the 3599 instances available at the time. The zinc finger motif was particularly felicitous for this test because instances are so plentiful in the database that test cases illustrating performance are easy to construct, and because while the average motif is sufficiently strong for most relative entropy variants to find it (although never the sum-of-pairs score), there are a large number of anomalous instances. A good test is to see how often a scoring function can detect these without introducing too many false positives. This variety of anomalies includes gaps, insertions, and residue changes from the regular expression pattern describing the pattern. Several hundred of the C2H2 motifs cited do not follow the nominal regular expression (C.2,4C.3[LIVMFYWC].8H.3,5H) for the C2H2 zinc finger.

4.3 Input

The input to each program consists of the above motifs with added distractor sequence of unrelated protein, selected using the SWISS-PROT random entry option¹. Using protein sequence relieves the necessity of providing a background model for random generation, and gives more realistic results. The reason to use unrelated sequence rather than the sequence surrounding the motif instances is that these surrounding protein sequences are likely to be related to each other, and so the algorithm will discover true patterns at many points in the sequence, making it hard to distinguish false positives from true positives.

The distractor sequence used for DHFR is 10,360 random AA, which is 40 times the aggregate length of the 11 planted DHFR motifs. The distractor sequence for C2H2 is 23,620 AA, which is 10 times the aggregate length of the 100 planted zinc finger motifs.

The twelve pattern-specifying zinc finger instances are pre-aligned with some human intervention to insure that the critical cysteine and histidine residues are aligned correctly. For the runs involving the ungapped algorithm, a different pattern specification must be used, in which the instances are aligned ungapped. For this alignment, the inner cysteine and histidine of each instance are aligned but the outer ones vary in position.

4.4 Pattern discovery

To test the effectiveness of the combined scoring function in detecting novel patterns, we use a hill-climbing search heuristic similar to a less randomized version of the Lawrence et al. [12] Gibbs sampling approach. This algorithm differs from typical Gibbs sampling for motifs in two aspects. The first is that it performs a dynamic programming match between the fixed pattern and the database, as described in [17], in order to permit gapped alignment of instances. The other important difference is that, while the instance it discards is chosen randomly, it selects the best of its choices instead of sampling from a probability distribution determined by the score. While not gaining the theoretical benefits that come with sampling (see, e.g., [2]), this algorithm's behavior is easier to characterize, and in practice, given repeated random restarts, it samples effectively from the high end of the set of scores with rapid convergence.

¹ <http://us.expasy.org/prot/get-random-entry.html>

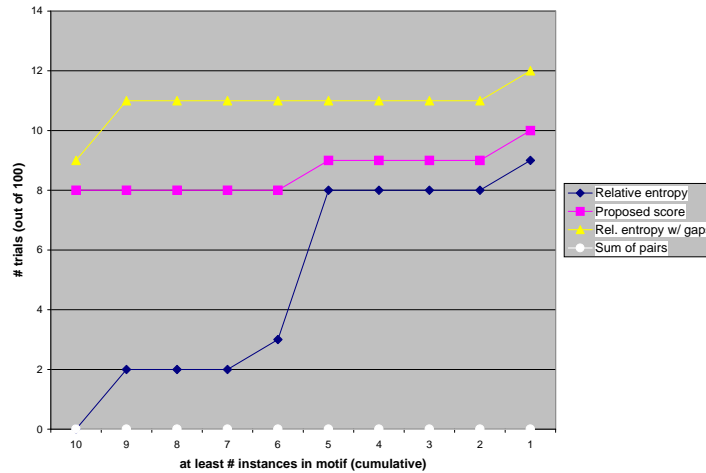


Fig. 1 Out of 100 hill-climbing runs, how many runs ended in instances of the planted pattern. Along the x -axis, the number of instances for “success” decreases; for the leftmost data, all 10 proposed instances must be motif instances.

As a measure of each score’s usefulness at finding novel motifs, we use the frequency with which a random restart results in discovering the planted motif.

The size parameters on the runs of the motif instructed it to find 10 motifs of size 20. This does not mesh exactly with the size (14) or number of instances (11) of the motif, in order to simulate a real-life usage, where one may not know precisely how many instances or what length motif to search for, but instead might choose round numbers at a reasonable order of magnitude for the problem.

4.5 Pattern matching

Comparing the effectiveness of this metric to others for pattern matching is easier, since it requires no additional algorithm. Given a pattern, the scoring functions alone induce a score for each candidate sequence. We compare the number of true positives (identified zinc fingers) and false positives (high-scoring regions that are not the planted zinc finger motifs) at various recognition score thresholds.

In this case, the pattern being searched for is precisely the length of the (aligned) predetermined zinc fingers.

5 Results

5.1 Pattern discovery

The results for pattern discovery are shown in Figure 1. The difference between gapped relative entropy and the proposed score (gapped relative entropy with pairwise comparison weights) is minor: 8 versus 9 runs out of 100 found all 10 instances.

Note on the other hand that the ungapped relative entropy score never found as many as 10 of the 11 motif instances, and only in 3% of the runs found more than half of the instances. In real data, this could

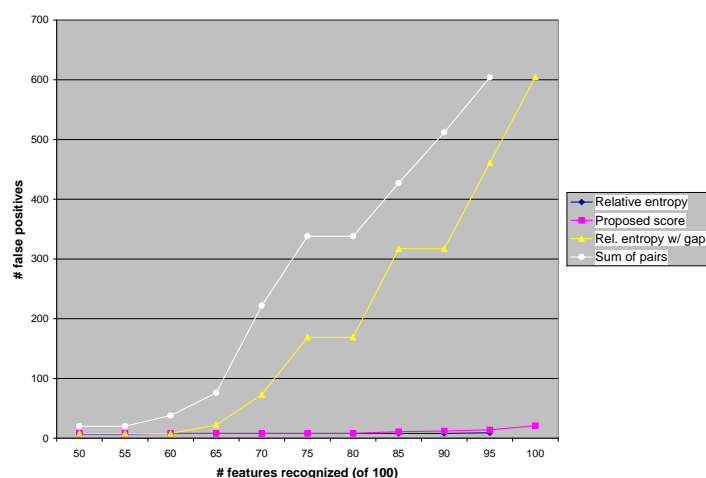


Fig. 2 At various score thresholds, how many of 100 zinc fingers were recognized (x-axis) versus how many other locations were falsely identified as zinc fingers (y-axis).

translate to a 3-4x increase in running time, or more importantly, to subtler patterns that the gapped scores barely find being permanently hidden to this metric.

The sum-of-pairs score is clearly poorly suited to this kind of heuristic sampling approach. This is likely because its score increases slowly as a pattern begins to appear, in contrast with a relative entropy score which rapidly recognizes a developing pattern and moves to add more instances.

5.2 Pattern matching

Figure 2 shows the tradeoff between false positives and true positives at different score thresholds. In this case, it is the *ungapped* relative entropy metric which performs comparably to the proposed metric. The ungapped metric has slightly fewer false positives than the proposed metric. However, there is 1 out of the 100 zinc fingers which the ungapped score was unable to recognize, even after raising the threshold until 600 or more false positives were introduced, because of gaps in the inner portion.

The reason for the surprising performance of ungapped relative entropy is that, although the C2H2 pattern is variably sized, the center portion (a cysteine, 8 residues, and a histidine) is tightly conserved and usually, although not always, ungapped. Since a completely preserved column speaks loudly to a relative entropy score, the ungapped score in this case is relatively quick to recognize most instances.

Again, in this case, the sum of pairs score performs poorly, but the interesting thing here is the dramatic reversal of position of the gapped and ungapped relative entropy scores from the previous graph.

5.3 Repeatability

Both results shown appear to be robust, the relative orderings and magnitudes holding up to repeated experiments and minor changes to the data (such as regenerating the distractor sequence).

6 Discussion

The results demonstrate that adding amino acid similarities to a relative entropy metric is a powerful tool for recognizing protein motifs and motif instances. Simpler metrics that use amino acid similarities, such as a sum-of-pairs metric, are not viable; the power of the relative entropy metric for multiple sequence alignment and comparison is important to preserve.

At first, it might appear unpromising that the proposed metric was edged out in each experiment by a previous metric, in one case the common relative entropy score and in the other the score simply modified for gaps.

However, on the contrary, these results show compellingly that the extra power bestowed by recognizing beneficial amino acid combinations makes the score more robust to variation in the pattern being looked for and in the surrounding sequence. With the novel metric, one need not decide in advance whether one wants a motif which is, like the zinc finger, still clear to an ungapped metric, or whether the motif in question is, like DHFR, multimodal enough in size that only a gapped metric can easily recognize that the differently-sized subsets are part of the same motif.

In addition, the new metric's consistent high performance will become important in domains where neither previous metric performs well. That such domains exist seems likely from the amount of separation between good and bad metrics in the results. Although so far in our experiments no protein data has been seen to confound both the gapped and ungapped metrics, the space of possible data types is large and notoriously difficult to sample.

7 Future work

7.1 Sequence alignment

Because of the arbitrary nature of the choice of affine gaps, and the error factor the penalty introduces over long sequences, it makes as little sense to extend this metric directly to consider alignments of long, ill-conserved sequences. However, Dialign's segment-to-segment approach is an answer to this problem.

A reasonable way to align long, ill-conserved sequences, then, might be to discover short matches using the hill-climbing sampler and string them together, just as in [14,13], according to score and consistency of order.

An important future project is to implement this algorithm, in order to compare it to existing metrics and alignment software. If empirically successful, it would fill a compelling niche: the alignment of sequences sufficiently diverged that few short regions remain ungapped, making it impossible for a segment-to-segment algorithm using ungapped segments to find the correct alignment.

References

1. M. Blanchette, B. Schwikowski, and M. Tompa. An exact algorithm to identify motifs in orthologous sequences from multiple species. In *Proc. of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, pages 37–45, 2000.
2. K. S. Chan. Asymptotic behavior of the gibbs sampler. *J. Amer. Statist. Assoc.*, 88:320–326, 1993.
3. M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. In M. O. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, volume 5, suppl. 3, pages 345–352. Natl. Biomed. Res. Found., Washington, 1978.
4. A. Dembo and S. Karlin. Strong limit theorems of empirical functionals for large exceedances of partial sums of iid variables. *Annals of Probability*, 19(4):1737–1755, 1991.
5. R. Laskowski et. al. Pdbsum. <http://www.biochem.ucl.ac.uk/bsm/pdbsum/>, 2002.
6. Schwartz et al. Pipmaker—a web server for aligning two genomic dna sequences. *Genome Research*, 10:577–586, April 2000.
7. ExpASy. Swiss-prot. <http://www.us.expasy.org/sprot/>, 2002. hosted by the Swiss Insitute of Bioinformatics.

8. J. G. Henikoff and S. Henikoff. Using substitution probabilities to improve position-specific scoring matrices. *Comput. Appl. Biosci.*, 12(2):135–43, 1996.
9. S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89:10915–10919, 1992.
10. S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA*, 87:2264–2268, 1990.
11. S. Karlin and S. F. Altschul. Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc. Natl. Acad. Sci. USA*, 90:5873–5877, 1993.
12. C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: A gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.
13. B. Morgenstern. Dialign 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15:211–218, 1999.
14. B. Morgenstern, A. Dress, and T. Werner. Multiple dna and protein sequence alignment based on segment-to-segment comparison. *Proc. Natl. Acad. Sci. USA*, 93:12098–12103, 1996.
15. S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.
16. S. Pietrokovski, J. G. Henikoff, and S. Henikoff. The blocks database—a system for protein classification. *Nucl. Acids Res.*, 24(1):197–200, 1996.
17. E. Rocke and M. Tompa. An algorithm for finding novel gapped motifs in dna sequences. In *Proc. of the 2nd Annual International Conference on Computational Molecular Biology (RECOMB 1998)*, pages 228–233, March 1998.
18. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.
19. T. D. Wu and D. L. Brutlag. Discovering empirically conserved amino acid substitution groups in databases of protein families. In *Proc. of the 4th International Conference on Intelligent Systems for Molecular Biology (ISMB 1996)*, pages 230–240, 1996.