JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 12, Number 7, 2005

Pp. 978–1001

Correcting BLAST e-Values for Low-Complexity Segments

ITAI SHARON,[1] AARON BIRKLAND,[2] KUAN CHANG,[3] RAN EL-YANIV,[1] and GOLAN YONA[2]

ABSTRACT

The statistical estimates of BLAST and PSI-BLAST are of extreme importance to determine the biological relevance of sequence matches. While being very effective in evaluating most matches, these estimates usually overestimate the significance of matches in the presence of low complexity segments. In this paper, we present a model, based on divergence measures and statistics of the alignment structure, that corrects BLAST e-values for low complexity sequences without filtering or excluding them and generates scores that are more effective in distinguishing true similarities from chance similarities. We evaluate our method and compare it to other known methods using the Gene Ontology (GO) knowledge resource as a benchmark. Various performance measures, including ROC analysis, indicate that the new model improves upon the state of the art. The program is available at *biozon.org/ftp/* and *www.cs.technion.ac.il/~itaish/lowcomp/*.

Key words: AU1

1. INTRODUCTION

Low complexity sequences, also known as "simple sequences," are abundant in proteins. These compositionally biased sequences are commonly seen in homopolypeptides and short-period tandem repeats and are frequent in structural proteins such as collagens and cell-wall proteins. A study by Wootton and Federhen (1993) shows that about half of the proteins in SwissProt contain at least one such region. Another study (Golding, 1999) also shows that low complexity sequences are the most frequent segments in *Saccharomyces cerevisiae* proteins. Many studies analyzed the amino acid distribution in low complexity sequences, concluding that amino acids such as alanine (A), serine (S), proline (P), and glycine (G) are very frequent within these sequences while cysteine (C) and tryptophan (W) are very rare (Promponas *et al.*, 2000; Romero *et al.*, 2001; Alba *et al.*, 2002) (the complete statistics is given in Table 1). However, not much is known about the potential function of these sequences. T1

A major problem with low-complexity sequences arises in sequence homology searches. Because of the repetitive nature of these sequences, one might detect many high-scoring similarities that are biologically meaningless. The statistical theory that was developed to estimate the significance of sequence matches

[1]Department of Computer Science, Technion, Haifa, Israel.
[2]Department of Computer Science, Cornell University, Ithaca, NY.
[3]Department of Pathology and Immunology, Washington University School of Medicine, St. Louis, MO.

TABLE 1. AMINO ACID COMPOSITION IN LOW-COMPLEXITY SEGMENTS[a]

	Frequency		
Amino acid	*Low complexity*	*Overall*	*Ratio*
A	0.119	0.076	1.57
S	0.104	0.073	1.42
P	0.072	0.051	1.41
G	0.092	0.070	1.31
L	0.119	0.096	1.24
E	0.067	0.062	1.08
T	0.054	0.057	0.95
Q	0.038	0.040	0.95
R	0.050	0.053	0.94
K	0.048	0.056	0.86
V	0.055	0.065	0.85
I	0.040	0.057	0.70
D	0.035	0.050	0.70
N	0.029	0.043	0.67
F	0.024	0.041	0.59
H	0.012	0.023	0.52
C	0.008	0.017	0.47
M	0.011	0.024	0.46
Y	0.013	0.031	0.42
W	0.004	0.014	0.29

[a]For each amino acid we report its frequency in low-complexity segments, its overall frequency in protein sequences, and the ratio between the two. Amino acids are sorted based on the ratio. The segments were determined using SEG with the default options. More than 1,200,000 segments were analyzed varying in length between 3 and 4,339 residues.

(Karlin and Altschul, 1990; Dembo *et al.*, 1994a) fails to provide accurate estimates in this case. This theory assumes that the compositions of the compared sequences are similar to the overall composition of amino acids in the database (Karlin and Altschul, 1990), which is not true for sequences of low-complexity. As a result, the statistical estimates (as in BLAST p-value and e-value) tend to overestimate the significance of matches to proteins with unusual amino acid composition, reporting meaningless similarities as significant. Moreover, even for related sequences, these repetitive residues might cause misalignments.

To improve sequence similarity searches, several algorithms have been developed to handle low complexity sequences. The first algorithms to study these segments were XNU (Claverie *et al.*, 1993) and SEG (Wootton and Federhen, 1993). These algorithms consist of two basic steps: detecting low complexity segments and masking them before sequences are actually aligned. XNU measures whether the maximal segment pair (MSP) score of self-alignment reaches a certain threshold. If the score is above the threshold, the region is defined as a low complexity segment and is masked. Unlike XNU, SEG uses entropy to measure sequence complexity. It first scans through the sequences using a fixed sized sliding window to detect raw low complexity segments (low entropy). Then it refines the low complexity segments through an optimization routine. SEG has become well known since BLAST started applying SEG by default to filter out low complexity regions in protein sequences. However, not all low-complexity segments are filtered with this method, and the method is somewhat sensitive to the choice of the parameters such as the window size and the low-entropy threshold (see examples in Section 3.1.1). On the other hand, potentially useful information is lost due to this masking process. These low complexity fragments may play an important role in determining the structure and function of proteins (Wootton, 1994), and many relationships of biological significance can be missed if only sequences that pass the filter are to be considered (Yona and Levitt, 2000a).

Approaches that attempt to reduce information loss were recently introduced. For example, CAST masks only one type of the biased residues in low complexity segments. The type of the residues to be masked is determined by aligning the query amino acid sequence against the 20 different homopolypeptides (Promponas *et al.*, 2000). The SIMPLE algorithm (Alba *et al.*, 2002) uses the ratio of the average simplicity score of the sequence against that of its shuffled sequences to determine whether it is a low complexity sequence. If the ratio is above 1, the sequence is considered to be of low complexity. Karplus *et al.* (2003) introduced a method that computes the significance of a match by subtracting from the BLAST score the similarity score of a match with the reversed sequence. This preserves higher order correlations that are missed when using methods such as SEG. A related permutation approach is the *zscore*-based method. Given an alignment between a query sequence and a database sequence, a background population of "random" alignments is generated by permuting the database sequence and aligning the query sequence with each one of the permuted sequences.[1] The significance is estimated from the mean μ and the standard deviation σ of the background distribution in terms of the zscore $\frac{S-\mu}{\sigma}$. Permutation methods have the advantage that they preserve the context without making any assumptions on the source. Specifically, the sample space is sampled without replacements (i.e., finite source). However, they are very slow as they require tens of samples to generate the background distribution from which a zscore can be estimated. FN1

The most recent and effective method to handle low-complexity segments is the one implemented in the recent versions of BLAST (Schaffer *et al.*, 2001). The method (called *composition-based statistics*) rescales the statistical parameters based on the compositions of the sequences compared. This method proved to be quite effective in eliminating many chance similarities that are due to unusual sequence compositions (Schaffer *et al.*, 2001). However, the method does not work well in all cases, and low-complexity segments might still cause a problem. Furthermore, the composition-based statistics introduces some unnatural artifacts; most notably is the ranking of the query sequence in cases when the query sequence is also part of the database searched. Since the different parameters result in different statistical estimates that are not monotonic in the raw scores, the query sequence might not be the first hit reported, ranked sometimes below many other matches with less similar sequences, or totally missed in other cases (see Section 2.2).

Here we present a new approach for reducing BLAST false positive hits caused by low complexity sequences in a database search. Instead of focusing on detecting and masking low complexity segments, we keep the sequences intact and try to validate sequence alignments based on the statistics of the *alignment structure*. Our method uses prealigned sequences as input and re-estimates the probability of observing the alignment based on the amino acid composition of the protein sequences compared. The output is approximated e-values that serve to reflect the true significance of a match when the compared sequences are of low-complexity. We also introduce two filters of low-complexity segments based on statistical divergence measures.

This paper is organized as follows. We start with a brief overview of issues related to statistical significance of sequence alignments (Section 2). We then present the main components of our method (Section 3). We test our method and compare it to other methods using the GO database as a reference in Section 4 and finish with concluding remarks (Section 5).

2. BACKGROUND AND DEFINITIONS

2.1. The statistical significance of ungapped sequence alignments

The statistics of ungapped similarity scores has been studied extensively since the early 90's. One of the most important results in this field is the characterization of the distribution of local similarity scores without gaps. This distribution was shown to follow the *e*xtreme value distribution (Karlin and Altschul, 1990; Dembo and Karlin, 1991; Dembo *et al.*, 1994b). Specifically, $\mathcal{S}$, the local similarity score of two

[1] Alternatively, one can shuffle just the aligned residues and rescore the alignment. However, statistical estimates based on this method tend to overestimate the significance since the alignments with the random sequences are not optimized.

random sequences of lengths n and m, is distributed as an extreme value distribution with

$$p = Prob(\mathcal{S} \geq x) \sim 1 - \exp(-e^{-\lambda \cdot (x-u)}), \tag{1}$$

where $u = \frac{\ln Kmn}{\lambda}$ and λ and K are parameters that are estimated from the background distribution of amino acids and the scoring matrix (Karlin and Altschul, 1990).

The probability p is computed in the context of comparing *two* random sequences. It should be adjusted when multiple comparisons are performed (e.g., when searching a database with D sequences). Denote by p-**match** a match between two sequences that has a pvalue $\leq p$ (i.e., its score $\geq \mathcal{S}$). The *expectation value* (e-value) is the expected number of distinct p-matches (segment pairs) that would obtain a score $\geq \mathcal{S}$ by chance in a database search:

$$E = E(\text{number of } p\text{-matches}) = Dp. \tag{2}$$

Since not all database sequences have the same probability of sharing a similar region with the query sequence, D should be replaced with the effective size of the database. If the query sequence is of length n, the (pairwise) alignment of interest involves a database segment of length m, and the database has a total of N amino acids, then D should be replaced with N/m.

It should be noted that the theory of sequence alignment was established rigorously only for *ungapped* sequence alignments. However, several studies suggested that the scores of local *gapped* alignments are also distributed according to the extreme value distribution (Smith *et al.*, 1985; Arratia and Waterman, 1994; Waterman and Vingron, 1994), though some correction factors may be required (Altschul and Gish, 1996).

2.2. BLAST parameters, rescaling, and composition-based statistics

The parameters λ and K determine the exact characteristic of the extreme-value distribution for a given population of amino acid sequences. Specifically, given the distribution $\mathcal{P}_0$ (the overall background distribution of amino acids in the database), λ is obtained by solving the equation

$$\sum_{a,b} p_a p_b e^{\lambda \cdot s(a,b)} = 1, \tag{3}$$

where $s(a, b)$ is the similarity score of amino acids a, b and p_x is the probability of amino acid x. The equation is solvable using numerical methods such as Newton's method. The second parameter K is given by a geometrically convergent series which depends only on p_a and $s(a, b)$ (Karlin and Altschul, 1990).

These results (Equations (1) and (3)) hold for a specific class of scoring matrices (negative mean) and are subject to the restriction that the amino acid composition of the two sequences that are compared are not too dissimilar. Assuming that both sequences are drawn from the background distribution, the amino acid composition of both should resemble the background distribution of amino acids. Without this restriction, this equation overestimates the probability of similarity scores, and indeed, this is observed in protein sequences with unusual compositions.

In theory, one way to correct the e-value in the presence of low-complexity sequences would be to recompute the parameters based on the composition of the specific database sequence. However, this simple solution cannot be applied in practice since the analytical solutions for λ and K were established only for ungapped alignments, while BLAST in its current format generates gapped alignments. To overcome this problem, gapped-BLAST uses precomputed parameters. The parameters are precomputed based on simulations with a variety of scoring matrices and gap penalties, but for fixed sequence composition (the background distribution of amino acids in the database) (Altschul and Gish, 1996). However, this might lead to substantially inaccurate estimates when the query or library sequences have unusual sequence compositions (i.e., λ differs markedly from the precomputed one). Because of the exponential dependency on λ in Equation (1), a change in the value of that parameter might result in a many-orders-of-magnitude change in the significance of a given similarity score. Recomputing the parameters for each pair of sequences using simulations is currently impractical, since this is a costly and slow process, and a few approaches were developed for rapid correction of the statistical estimates for gapped alignments (Mott and Tribe, 1999; Mott, 2000).

Query:

```
>swissprot: (P40273) Histone H1.M6.1.
MSDAAVPPKKASPKKAAAKKASPKKSAARKTAAKKTAKKPAVRKPAAKKRAAPKKKPAAA
KKPAAKKAPKKAVKKAPKKK
```

Top match (composition-based statistics):

```
>trembl: (Q9RU01) Glucose-6-phosphate 1-dehydrogenase (EC 1.1.1.49) (G6PD).
         Length = 590

 Score = 39.7 bits (91), Expect = 0.008
 Identities = 31/74 (41%), Positives = 41/74 (54%), Gaps = 3/74 (4%)

Query: 8  PKKASPKKAAAKKASPKKSAARKTAAKKTAKKPAVRKPAAKKRAAPKKKPAAAKKPAAK- 66
          PKK+SPKK+  +KA  K+SAA+  AAK T +     + A K   A    PA ++K A K
Sbjct: 7  PKKSSPKKSGPEKALAKESAAQGEAAKATRQATQQTEAAKKVGVAQPGAPAQSRKAARKS 66

Query: 67 --KAPKKAVKKAPK 78
            + PK A   APK
Sbjct: 67 RQRVPKHAGDNAPK 80
```

FIG. 1. Effect of SEG and composition-based statistics on a BLAST search. We compared the following SwissProt entry (P40273 Histone H1) against a large nonredundant sequence database that contains more than 900,000 sequence entries (including the query sequence). When composition-based statistics was disabled, about 450 matches were detected with evalue < 0.001 of which more than half were Histones or Histone-like proteins. We repeated the search with the SEG flag on, and the search reported zero matches. When the composition-based statistics was enabled, no matches were detected with evalue < 0.001 and only two were detected with evalue < 0.1 (the top scoring match is displayed above). The query sequence was missing from the list and only one Histone was reported (with evalue 1.4).

The solution employed by the newer versions of BLAST is to rescale the *scoring matrix* for gapped alignments based on a factor that is computed from the statistical parameters of ungapped alignments (the method is called *composition-based statistics*). Specifically, given a query sequence and a database sequence, the program computes the parameters λ_u (for ungapped alignments with the background distribution) and λ'_u (for ungapped alignments with the specific compositions of the query and database sequences, solving analytically Equation (3)), to derive the factor $r = \lambda'_u/\lambda_u$. This factor is used to rescale the substitution scores and the sequences are realigned using the re-scaled scoring matrix. The parameter λ_g (for gapped alignments with the background distribution) is then used to estimate the significance of the new similarity scores. Thus, composition-based statistics effectively changes the relative scale of the scoring matrix and the gap penalties for each pair of sequences compared. To save time, the new e-values are computed only for matches that pass the threshold in the first iteration.

While being quite effective in eliminating spurious meaningless hits, this method can also affect the sensitivity of the algorithm, and in many cases significant matches are eliminated. Moreover, it can create situations in which the query sequence is no longer the most significant match or it is eliminated altogether from the match list (see Fig. 1). This behavior is due to the fact that the rescaling method essentially uses different parameters for each pair, resulting in a nonmonotonic transformation. The problem is especially pronounced when the iterative PSI-BLAST algorithm is used (where the query sequence might be eliminated from the list after a few iterations). This counter-intuitive behavior poses a serious problem in large scale analysis of the protein universe.

F1

3. METHODS

Here we present a new heuristic approach for computing the significance of alignments in the presence of low-complexity sequences, through *postnormalization*. Our method does not change the alignment. Rather,

it reevaluates its significance based on the *alignment structure*. I.e., the method reestimates the probability that a specific match between two proteins can occur by chance, *given their alignment*.

This approach requires prealigned protein sequences from BLAST (without the application of filters or composition-based statistics). Our algorithm starts by evaluating whether the alignment is composed mostly of low-complexity segments and whether the sequences have diverged from their *common source* to the extent that their alignment is deemed suspicious. To this end we use information theoretic measures as described in Section 3.1. If the alignment is marked "suspicious" we invoke our postnormalization procedure.

The postnormalization procedure works in two steps. First, we apply the *factor method*. This method eliminates almost all alignments that are due to low-complexity sequences and are biologically insignificant (Section 3.2.1). Next, we apply the *segment-profile method* in an attempt to rescue matches that are based on low-complexity segments and were eliminated by our factor method, but are still biologically meaningful. The overall procedure can be summarized as follows:

For each BLAST alignment with $\mathsf{evalue} < T$, determine whether the alignment is suspicious. If it is, then do the following:

- Apply the factor method to compute a new e-value evalue_{factor}.
- Based on the segment profile, compute $\mathsf{evalue}_{profile}$.
- Let the final e-value be the minimum over $\{\mathsf{evalue}_{factor}, \mathsf{evalue}_{profile}\}$.

The threshold T is set to 0.1 in our experiments. We now turn to describe each element in detail.

3.1. Detection of suspicious alignments

3.1.1. Fast detection of low-complexity segments. BLAST e-values are quite reliable if the aligned sequences are of high complexity. Therefore, there is no need to automatically correct the statistical estimates even if the sequences compared have low-complexity segments. Only if the *aligned subsequences* are of low-complexity should a correction be introduced. We start by deciding if an aligned segment is of low complexity. If it is, the alignment is deemed suspicious, and we invoke our post normalization method.

To decide if the sequences are of low-complexity, one can use SEG (Wootton and Federhen, 1993). While SEG is quite effective in determining low-complexity segments, it is sensitive to the choice of parameters. SEG uses four parameters, including the window size and the low (trigger) complexity. However, if the average entropy does not fall below the threshold value, the filter is not triggered, and the segment is considered of high-complexity. Two such examples are given in Fig. 2. **F2**

Here we propose a different method for detecting low-complexity segments. Given a (sub)sequence **a**, our method first computes the composition of **a** to generate the distribution P_a of amino acid in **a**. We then compute the Jensen–Shannon (JS) divergence (Lin, 1991) between P_a and P_0, the background distribution of amino acids. If this divergence (distance) exceeds a certain threshold, D_1, then we say that the sequence is of low-complexity. Formally, given two (empirical) probability distributions **p** and **q**, their α-JS divergence is defined as

$$D_\alpha^{JS}[\mathbf{p}||\mathbf{q}] = \alpha D^{KL}[\mathbf{p}||\mathbf{r}] + (1-\alpha) D^{KL}[\mathbf{q}||\mathbf{r}],$$

where $D^{KL}[\mathbf{p}||\mathbf{q}] = \sum_i p_i \log_2 \frac{p_i}{q_i}$ is the Kullback–Leibler (KL) divergence (Kullback, 1959) and $\mathbf{r} = \alpha\mathbf{p} + (1-\alpha)\mathbf{q}$ can be interpreted as the most likely common source distribution of both distributions **p** and **q** (El-Yaniv *et al.*, 1997). The parameter α is the prior weight ($0 \le \alpha \le 1$), set here to $1/2$. This measure is symmetric and bounded (unlike the KL divergence) and ranges between 0 and 1. It has been shown recently that $\sqrt{D_\alpha^{JS}[\cdot||\cdot]}$ is a metric (Fuglede and Topse, 2004).

To set the threshold D_1 we computed the JS-distances of low complexity and of high-complexity segments (as assigned by SEG) from the background distribution. A total of more than 1,200,000 low-complexity segments and 2,100,000 high-complexity segments were analyzed. The distributions of distances are shown in Fig. 3. Note that the distributions are well separated and suggest a natural threshold of about 0.3. However, since the segments were defined by SEG and since SEG may miss some low-complexity segments, we also considered other thresholds in our experiments. **F3**

The proposed method allows for fast detection of low-complexity segments. Given an alignment, we compute the distribution of amino acids in the alignment, for both the query and the database sequence, and

```
>gb: (AF007575) ES/130-related protein [Homo sapiens]
VPMVVVPPVGAKGNTPATGTTQGKKAEGTQNQSKKAEGAPNQGRKAEGTPNQGKKTEGTP
NQGKKAEGTPNQGKKAEGTPNQGKKAEGAQNQGKKVDTTPNQGKKVEGAPTQGRKAEGAQ
NQAKKVEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGGQNQGKKTEEAQ
KQGKKAEGAQIQGKKNEGAQTQGKKAEGAQNQGKKNEGAQTQGKKAEGAQTQGKKADGAQ
NQGKKAEGAQNQGKKAEGAQNQGKKAEGAQNQGKKADGAQNQGKKAEGAQNQGKKAEGAQ
NQGTKAEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGAQ
NQGKKAEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGAQ
NQGKKVEGAQNQGKKAEGAQNQGKKAEGAQNQGKKAEGAQNQGQKGEGAQNQGKKTEGAQ
GKKAERSPNQGKKGEGAPIQGKKADSVANQGTKVEGITNQGKKAEGSPSEGKKAEGSPNQ
GKKADAAANQGKKTESASVQ

>gb: AF247172\_1 (AF247172) RP4-alkaline phosphatase hybrid protein
PVTKARTPEMPLQSKTSTNNFGGMMPGGDESTKISKTSTNNFGGMMPGGDESTKISKTST
NNFGGMMPGGDESTKISKTSTNNFGGMMPGGDESTKISKTSTNNFGGMMPGGDESTKISK
TSTNNFGGMMPGGDESTKISKTSTNNFGGMMPGGDESTKISKTSTNNFGGMMPGGDESTK
ISKTSTNNFGGMMPGGDESTKISKTSTNNFGGMMPGGDESTKISKTSTNNFGGMMPGGDE
STKISKTSLEVLENR
```

FIG. 2. Low complexity sequences that are missed by SEG. The entropy of the first is 3.06 while the entropy of the background distribution is 4.2. The entropy of the second is 3.47. When running SEG with the default options the sequences are left intact.

their JS-distances from the background distribution. If the distance (of one or both) is above the threshold, we trigger the correction using our postnormalization method described in Section 3.2.

3.1.2. Divergence from the common source. Even if the segments compared are of low-complexity, the alignment might still be a biologically meaningful alignment. We apply a second filter to determine if an alignment is suspicious or not. For each alignment we collect the set of identities and similar amino acids. These are the *anchor residues* of the alignment. The distribution of anchor residues can be considered as the distribution of the common source of the sequences compared. If the JS-distance between the query sequence and the library sequence (*using the distribution of anchor residues as the the common source*) is small, then we can infer that the two sequences are likely to emerge from the same common ancestor. The common source induces a probability distribution on all possible types over the 20 amino acids. If the type is not preserved in the compared sequences and the distance from the common source exceeds a threshold D_2, then it is postulated that the similarity is either a chance similarity or a similarity that is due to low-complexity sequences.

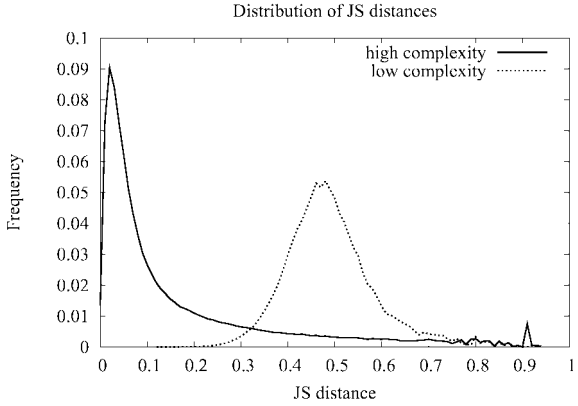

FIG. 3. Distribution of Jensen–Shannon (JS) distances for low-complexity and high-complexity segments.

When deriving the distribution of the common source from the anchor residues, each identity of amino acid a adds one to the counts of a, while similarity between a and b adds 0.5 to the counts of a and b each. For example, the common source of the following alignment is estimated from the anchor residues M, L, and V with counts 1, 1.5, and 0.5, respectively.

Query	$\rightarrow$	M	K	L	L	T
		M		$+$	L	
Subject	$\rightarrow$	M	Y	V	L	A

Due to undersampling, the empirical distributions might assign zero probability to some amino acids (as in the example above). To circumvent that, we augment the real counts with pseudo-counts, following the method that was introduced by Henikoff and Henikoff (1996), and the final probabilities are adjusted accordingly.

3.2. Correcting e-values by postnormalization

Given a suspicious alignment, our postnormalization procedure starts by applying the factor method. This method recomputes the probability that a given alignment was generated by chance, based on the *alignment structure*, as defined in Section 3.2.1. To determine if a match (that is based on low-complexity segments) was unjustifiably eliminated, we apply a second procedure as described in Section 3.3.

It should be noted that the approximated e-values our methods report are heuristically derived and do not necessarily correspond to exact probabilities or e-values. However, as our tests indicate (Section 4), these scores are more effective for recognizing related sequences than the original e-values reported by BLAST.

3.2.1. The factor method. Consider a hypothetical alignment between a query sequence and a subject (database) sequence, such as the short alignment displayed above. We divide the *matching sequence* between the two aligned sequences (the middle line) into four match-types: *exact*, *similar* (conservative substitution), *neutral* (neutral substitution), and *dissimilar* (mismatch). The definitions of the match-types rely on the scoring matrix used. Let $s(x, y)$ be the similarity score of amino acids x and y according to some scoring matrix, such as BLOSUM62 (Henikoff and Henikoff, 1992). Then, the match-type is defined as

- exact if $x = y$ (denoted by $=$),
- similar if $x \neq y$ and $s(x, y) > 0$ (denoted by $+$),
- neutral if $x \neq y$ and $s(x, y) = 0$ (denoted by $*$), and
- dissimilar if $x \neq y$ and $s(x, y) < 0$ (denoted by $-$).

For example, leucine (L) and valine (V) are assigned positive similarity scores according to BLOSUM62 and are generally considered similar biochemically. Under that representation, the alignment above is expressed as follows:

Query	$\rightarrow$	M	K	L	L	T
		$=$	$-$	$+$	$=$	$*$
Subject	$\rightarrow$	M	Y	V	L	A

We denote the match-type by m (where $m \in \{=, +, -, *\}$). A *match* is simply the pair (x, m) where x is the amino acid (in the query sequence) and m is the match-type (with the amino acid of the database sequence at that position). For example, the second match in the alignment displayed above is the pair $(K, -)$. The set of matches in a given alignment $\mathbf{A}$ is referred to as the **alignment structure**. I.e.,

$$STRUCT(\mathbf{A}) = \{(x_1, m_1), (x_2, m_2), \ldots, (x_n, m_n)\}$$

where n is the length of the alignment (ignoring gaps). Note that here we ignore the order of matches (extensions of this model account also for the order; see subsequent sections). Finally, we define the

binary relation $\mathcal{R}$ between amino acids x, y, such that $\mathcal{R}(x, y) = m$, where m is determined based on the similarity score $s(x, y)$. The *equivalence set*[2] $S_{x,m}$ is the set of all amino acids y such that $\mathcal{R}(x, y) = m$. FN2

The BLAST e-value is computed assuming that the sequence is drawn from a fixed background distribution. Our assumption is that the actual distribution of amino acids in the database (subject) sequence affects the likelihood of observing a given alignment. If the alignment is more likely to happen by chance when the subject distribution is used, then the e-value should be increased, as the similarity is less significant than reported. Similarly, if the alignment is less likely to happen by chance when considering the subject distribution, then the e-value should be reduced, as the similarity is more significant than reported (although that hardly ever happens).

To compute the likelihood of a **specific** alignment **A**, given a distribution **Q** of amino acids, we assume that the amino acids are sampled at random with replacements from that distribution and compute the probability (as induced by **Q**) to observe the exact, similar, neutral, and dissimilar matches in the alignment (gaps are ignored). I.e., if a position i in the alignment is of match-type m and the query residue at that position is a, then the probability of observing a match of that type is the sum of probabilities over all amino acids in the equivalence set $S_{a,m}$, that is, all amino acids b for which $\mathcal{R}(a, b) = m$:

$$Prob(\mathbf{A}_i|\mathbf{Q}) = Prob(S_{a,m}|\mathbf{Q}) = \sum_{b\ s.t.\ \mathcal{R}(a,b)=m} \mathbf{Q}(b),$$

where $\mathcal{R}$ is the binary relation defined above. For example, $Prob(S_{a,+}|\mathbf{Q})$ is the probability (according to **Q**) to generate an amino acid that is similar to a. Assuming independence, the probability of the entire alignment structure is simply the product of the probabilities of all the positions along the alignment

$$Prob(\mathbf{A}|\mathbf{Q}) \triangleq Prob(STRUCT(\mathbf{A})|\mathbf{Q}) = \prod_i Prob(\mathbf{A}_i|\mathbf{Q}).$$

In the example above, with **Q** being the subject distribution of *MYVLA*,

$$\begin{aligned} Prob(\mathbf{A}|\mathbf{Q}) &= Prob(S_{M,=}|\mathbf{Q})Prob(S_{K,-}|\mathbf{Q})Prob(S_{L,+}|\mathbf{Q})Prob(S_{L,=}|\mathbf{Q})Prob(S_{T,*}|\mathbf{Q}) \\ &= 1/5 \times 1 \times 2/5 \times 1/5 \times 2/5 = 6.4 \cdot 10^{-3}. \end{aligned}$$

The proposed correction factor F, for the BLAST evalue, is defined as the likelihood ratio

$$F = \frac{Prob(\mathbf{A}|\mathbf{Q})}{Prob(\mathbf{A}|\mathcal{P}_0)},$$

where $Prob(\mathbf{A}|\mathbf{Q})$ is the probability of forming the alignment (the matching sequence) using the context of the subject sequence (the database sequence), and $Prob(\mathbf{A}|\mathcal{P}_0)$ is defined as the probability of forming the matching sequence as induced by the background amino acid distribution. The final evalue_{factor} is defined as $\text{evalue}_{blast} \times F$ where evalue_{blast} is the original e-value reported by BLAST. If the factor is larger than one, i.e., the probability to observe the alignment by chance is larger when considering the subject distribution, then the correction factor will increase the e-value (reduce significant) as expected. In the example above, $Prob(\mathbf{A}|\mathbf{Q}) = 6.4 \cdot 10^{-3}$ while

$$\begin{aligned} Prob(\mathbf{A}|\mathcal{P}_0) &= Prob(S_{M,=}|\mathcal{P}_0)Prob(S_{K,-}|\mathcal{P}_0)Prob(S_{L,+}|\mathcal{P}_0)Prob(S_{L,=}|\mathcal{P}_0)Prob(S_{T,*}|\mathcal{P}_0) \\ &= 3.93 \cdot 10^{-5}. \end{aligned}$$

Therefore, the correction factor is $F = 162.85$. For long alignments of low-complexity sequences, this factor can be many magnitudes of order larger. Note that this approach somewhat resembles the rescaling technique that underlies the composition-based statistics method, but it is derived from different principles and it does not modify the scoring function or the statistical parameters; nor does it require realignment.

[2] Although the equivalence sets depend on the scoring matrix used and its scale, the fundamental properties of the alignment, as summarized in the statistics of match types, are expected to remain roughly the same. Therefore, our final statistical estimates are not expected to change drastically with the scoring matrix.

```
Query: 6   TPAKTLEKETTTNPTKKPTPKTTERGTSTSQSTVLDTTISKHTIQQQSLYSTTPENTPNS 65
           TP +T  +  T  PT+ PTP  TE  T T   T   T     T       + TP  TP
Sbjct: 641 TPTETPTETPTETPTETPTPTPTETPTETPTETTTPTPTETPTETPTETPTETPTETPTE 700

Query: 66  TQTPAASEPSTSNST 80
           T+TP  +E  T   T
Sbjct: 701 TETPTPTETPTPTET 715
```

FIG. 4. An alignment between two low-complexity sequences. We compared TrEMBL AAL60315 (Attachment glycoprotein G) with TrEMBL Q23853 (HYPOTHETICAL 98.7 kDa PROTEIN). BLAST assigns an evalue of 2.0e−05 to this alignment while the factor method assigns an evalue of 4.3e+19.

A concrete example is provided in Fig. 4. This is the alignment reported by BLAST between TrEMBL AAL60315 and TrEMBL Q23853. This alignment does not seem to have any biological significance, as it is based almost purely on the repetitive amino acids T and P. BLAST assigns an evalue of 2.0e−05 to this alignment. The factor method assigns an evalue 4.3e+19. While overly high (due to the sampling without replacements), it eliminates the similarity from the list of significant matches, as desired. **F4**

3.2.2. Finite versus infinite sources. The underlying assumption of the factor method, that the source is infinite, is clearly unrealistic (note that the same assumption is employed in the statistical estimates of BLAST). However, changing the setup such that sampling is done without replacements (finite sample) introduces other problems; specifically, one has to update $\mathbf{Q}$ and recompute $Prob(S_{a,m}|\mathbf{Q})$ after every drawing, which slows down the computation. Moreover, edge effects lead to other undesirable properties. The assumption that the source is fixed (infinite source) is reasonable in practice since the same assumption is applied to both terms that make up the factor F, based on the source distribution $Prob(\mathbf{A}|\mathbf{Q})$ and the background distribution $Prob(\mathbf{A}|\mathcal{P}_0)$. Only if the probabilities are different will we observe different values for $Prob(\mathbf{A})$ and a factor that is $\neq 1$.

3.3. Rescuing significant matches

Although most matches between low complexity sequences are chance similarities, once in a while a significant similarity might come along. The correction we introduce eliminates false matches due to low-complexity segments. However, since it uses only first-order statistics, while ignoring the order of amino acids, it might also eliminate these significant matches. Since the sequences are repetitive, the positions are not i.i.d., and the estimates might result in extremely high factors (recall that these similarities will be completely missed with tools like SEG, since the sequences are filtered out).

Our goal is to introduce a fast correction that will "rescue" these significant matches. The main observation is that true matches (as opposed to chance similarities) of low complexity segments are composed of relatively long continuous subalignments of identical or similar matches. Therefore, if we can characterize the distribution of segments of identical and similar matches in an alignment, we can re-estimate the significance of the alignment.

We are interested in the probability to observe this alignment by chance. Theoretical results on the statistics of local sequence alignment *scores* fail to provide accurate estimates for low-complexity sequences. Therefore, once again we focus on the alignment structure as characterized by the collection of similar and identical segments that make up the core of the alignment. To approximate the probability to observe a given alignment by chance we use the *segment-profile* model, described next (two other approaches that are based on different models are described in Appendix A). The parameters of the models are estimated from a large collection of insignificant alignments.[3] In the experiments described below, more than 18,000,000 alignments were collected and analyzed. Each of these alignments has evalue > 1 and therefore, is most likely to reflect chance similarity. We refer to this set as the *training set*. **FN3**

[3]These are not random alignments; the alignments are optimized, to maximize the similarity score between the aligned sequences.

```
Query: 8  PKKASPKKAAAKKASPKKSAARKTAAKKTAKKPAVRKPAAKKRAAPKKKPAAAKKPAAK- 66
          PKK+SPKK+  +KA  K+SAA+  AAK T +     + A K   A    PA ++K A K
          ---------  ---  ------  --- - -     - - -   -    -- --- - -
              9       3     6      3  1 1     1 1 1   1    2   3  1 1
Sbjct: 7  PKKSSPKKSGPEKALAKESAAQGEAAKATRQATQQTEAAKKVGVAQPGAPAQSRKAARKS 66


Query: 67 --KAPKKAVKKAPK 78
            + PK A   APK
            - -- -   ---
            1 2  1    3
Sbjct: 67 RQRVPKHAGDNAPK 80
```

FIG. 5. The segment-profile. The similar segments are marked by dashes, and the numbers underneath indicate their lengths. In the following alignment, there are ten 1-segments, two 2-segments, four 3-segments, one 6-segment and one 9-segment.

3.3.1. The segment-profile model. This model characterizes the alignment structure through its set of similar segments. Our approach resembles the approach that was introduced by Karlin and Altschul (1993) to assess the significance of a match that is composed of several ungapped alignments. However, unlike their approach that characterizes a match by the set of scores of the ungapped alignments, the segment-profile approach focuses on the lengths of the similar segments that make up the alignment. Formally, we are given an alignment **A** of length L (excluding gaps). A *similar segment* is defined as a continuous sequence of only positive matches (conservative substitutions including identities). Each similar segment is maximal in the sense that it cannot be extended to the left or the right, or it won't be a similar segment. We define a *k-segment* to be a similar segment of length k (as opposed to an arbitrary continuous segment of k paired residues) and denote by m_k the number of k-segments that are observed in a given alignment (see Fig. 5). The segment-profile of an alignment is defined as the set $\{m_1, m_2, m_3, \dots\}$. The same approach can be applied with identical segments (where only identities are considered) instead of similar segments. F5

To estimate the probability to observe a k-segment by chance, we consider all the alignments in the training set as if they were part of one long alignment. Let N be the total number of residues paired with one another (with gaps excluded) in the training set and n_k the total number of *observed* k-segments. An upper bound on the number of k-segments in the training set is given by $n = (N+1)/(k+1)$, where the 1 is added to force separators. We approximate[4] the probability that a segment of length k is a k-segment by n_k/n. FN4

Given an alignment **A** of length L, the maximal possible number of k-segments is denoted by $n_{kL} = (L+1)/(k+1)$, and we can use the binomial distribution with parameter $p_k = n_k/n$ to estimate the probability to observe exactly m k-segments,

$$Prob_k(m) = \binom{n_{kL}}{m} p_k^m (1-p_k)^{n_{kL}-m}.$$

Given an alignment with m_k k-segments, the pvalue of the alignment just based on m_k is

$$\text{pvalue}(m_k) = Prob_k(m > m_k) = \sum_{m=m_k+1}^{n_{kL}} Prob_k(m).$$

[4]The proposed method is a heuristic and is based on coarse estimates of probabilities. Specifically, N is kept fixed although once we eliminate all 1-segments the sample space for 2-segments is smaller, and so on. However, the same approach (of fixing the sample space) is applied when using these parameters to estimate the significance of a given alignment with a specific segment-profile.

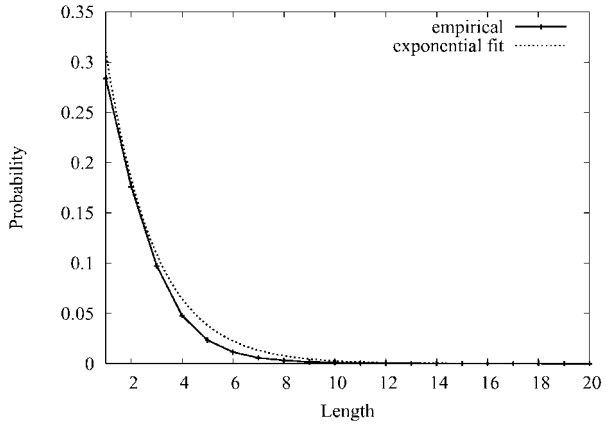

FIG. 6. The characteristic parameter of the binomial distribution. For each k, this is the probability that a random segment of length k is a k-segment (similar segment of length k).

Given the complete segment profile of the alignment $\{m_1, m_2, m_3, \dots\}$, the total pvalue is approximated by

$$\text{pvalue}(\mathbf{A}) = \prod_k \text{pvalue}(m_k).$$

Finally, the evalue is obtained by applying the Bonferonni correction, $\text{evalue}_{profile} = \text{pvalue} \cdot N$, where N is the number of sequences in the database searched. Although simplistic, this approach successfully rescues many biologically significant matches (see next sections).

The parameters $\{p_k\}$ are estimated from the training set as described above. To estimate the parameter p_k for arbitrarily large k, we extrapolate p_k for $k > 20$ as in Fig. 6. This function can be modeled with an exponential distribution of the form $f(x) = a \cdot \exp(-ax)$ with $a \approx 1$. F6

3.4. Combining methods

The final e-value returned by our postnormalization procedure is defined as the minimum over two different estimates, evalue_{factor} and $\text{evalue}_{profile}$. Each method detects a different subset of significant alignments, and by combining the results of all procedures we can rescue a larger set of significant and biologically meaningful alignments. We also tested two other methods, based on the similarity-profile and the longest match, as described in Appendix A. However, our tests indicate that for all practical purposes, it is sufficient to use the factor method and the segment-profile method (these two methods account together for about 97% of all alignments that are reported with evalue < 0.1, after correction).

4. RESULTS AND PERFORMANCE EVALUATION

The hardest part of this study proved to be the evaluation. While obtaining a set of low-complexity queries is easy, verifying the search results and assessing their quality was a much more difficult task. To find the most effective method, one has to evaluate the accuracy and sensitivity of each method in detecting significant and meaningful matches. While the definition of "significance" is method dependent and subjected to the exact computational procedure, the definition of "meaningful" should be objective and should hold regardless of the method used. In general, there are many databases of domain and protein families that one can use to evaluate sequence-based search methods, such as Prints (Attwood *et al.*, 1999), Pfam (Bateman *et al.*, 1999), ProDom (Corpet *et al.*, 1999), and others. However, these databases ignore, for the most part, low-complexity sequences (for example, Pfam uses SEG to exclude low-complexity segments). Moreover, almost all existing databases focus on domains rather than on complete protein sequences, thus further complicating the assessment.

As the standard of truth and our reference classification we chose the GO database (Ashburner *et al.*, 2000). The goal of our tests is to find out whether a suggested method succeeds in assigning lower e-values to the library sequences that are truly related to the query sequence. This is done by checking whether the library sequences share mutual GO-terms with the query sequence as is explained in the next section. It should be noted that while the GO database is an excellent source of biological knowledge (that was manually curated for the most part), it is not ideal for performance evaluation. Usually the GO database is used to check if a group of objects is enriched with similar GO terms, more than is expected by chance (Wren and Garner, 2004; Cora *et al.*, 2004). However, performance evaluation in our case requires much more accurate measures. We describe our methodology in the next sections.

4.1. Evaluating biological relatedness using the GO database

Roughly speaking, a GO-term represents some mutual property of all proteins sharing it. GO terms are organized in an a-cyclic graph, in which all paths begin at the same root node and end in one of the graph's leaves or inner nodes. A node's parent represents a property that is more general than the node's property. We define the level or the *depth* of a node as the shortest distance from the root node. The amount of knowledge available on a protein will determine the depth of its GO-terms; this implies that proteins that were more closely researched are likely to be assigned to a lower GO-term in the graph. Unlike a tree form of a graph, it is possible to have more than one path leading from the root to a node; this implies that inner or leaf nodes may have more than one parent. Also, a protein may be assigned more than one GO-term, each one on a different branch of the graph (the different branches represent different groups of properties). At the top of the hierarchy, there is a single root node (gene ontology). This node branches into three main categories: biological process, cellular component, and molecular function. Each one of these second degree terms denote a major class of GO terms. Since we are interested in functional similarity, we restrict our analysis to GO terms that are child terms (directly or indirectly) of molecular function.

4.1.1. The common level. By definition, each pair of proteins must share some mutual ancestor; if the two proteins are not biologically related in any way, they will probably share the root GO-term, or one of the second degree GO-terms (biological process, cellular component, and molecular function). The *common level* of two proteins is defined as the level of their deepest common GO-term. To compute the common level of two proteins, we first compile the complete list of GO terms associated with each one them, starting from the leaf nodes and adding to the list all the parent nodes. These two lists are then compared to find the deepest common GO-term.

Our initial set of tests was based on the common level. Our assumption was that the deeper the shared GO-term, the more biologically significant the result is. However, while the common level is a good indicator of a valid relationship, it does not perfectly correlate with functional similarity. Indeed, it has been shown by Lord *et al.* (2003) that different GO terms at the same level are not equally significant. For example, 160,636 proteins in our dataset (see Section 4.2) can be associated with the GO term "binding" at level 3, while only 6 are associated with "chaperone regulator activity" at the same level. In other words, the common level does not necessarily designate the level of the functional similarity and does not translate well to a structured hierarchy.

4.1.2. The semantic similarity. To account for the variability in the size of GO families, we adopt an approach related to the one described by Lord *et al.* (2003) that attempts to quantify the *semantic similarity* based on the least frequent common GO term. Here we associate a significance measure with each GO term that is the probability that two protein selected at random will be associated with that GO term. Given a GO term g, we first count how many proteins are associated with it (or any of its children), denoted by N_{cg}. The probability is then given by

$$p_g = \frac{N_{cg}(N_{cg} - 1)}{N(N - 1)},$$

where N is the total number of proteins that are associated with GO terms. The *GO significance* of the semantic similarity of two proteins sharing a GO term g is defined as p_g. If two proteins share more than

one GO term, then the significance of their semantic similarity is defined as the least probability over all common GO terms.[5]

FN5

We exclude GO terms whose probability to occur at random is more than 0.05 (21 GO terms, each one associated with 30,000 proteins or more) and GO terms that occur only once (572 GO terms). We also eliminate the GO term 'molecular_function unknown' that is associated with 11675 proteins. All together, 3425 GO terms are considered, ranging in size from 2 to 29,424 proteins (for GO term "monovalent inorganic cation transporter activity").

Finally, for each protein we compile a list of semantically similar proteins (based on GO annotations), and order the list based on the significance of the common GO term. The list can be analyzed at different significance thresholds. This parameter is tuned in Section 4.2.2.

4.1.3. Matching sequence space and semantic space. While the semantic similarity is a better measure of functional similarity than the common level, it still does not make the GO data suitable for evaluation. GO data is partial, and since it is derived from multiple sources, it is not necessarily coherent. Consequently, in many cases proteins that are very similar based on sequence are not necessarily associated with the same set of GO terms. This can pose a big problem when evaluating the validity of the results, as true relationships can be wrongly labeled as false relations. The problem is especially pronounced when the query protein is semantically similar to only a few proteins according to GO (the *relevant semantic space*), but is similar (based on sequence) to many proteins (*relevant sequence space*) most of which are associated with other, unrelated GO terms. A similar mismatch problem occurs when the relevant sequence space is very small, but the relevant semantic space is very large. In an attempt to minimize false assignments and produce an effective set of queries for evaluation, we processed the queries to maximize the correlation between their semantic space and their sequence space. We define a *viable query* as a query for which the relevant semantic space is not more than double the size of the relevant sequence space (according to BLAST) and is not less than half the size of the sequence space. This criterion can be applied at different GO significance levels. Each significance level defines a different relevant semantic space. Naturally, the size of the semantic space decreases, as the significance level increases.

4.2. Datasets

4.2.1. Database. We use a composite nonredundant (NR) database that contains about 933,000 sequence entries compiled from multiple sources such as SwissProt, PIR, Genbank, and others (the database is available at *biozon.org*). GO terms were collected from multiple sources, downloaded from the GO consortium website. Out of the 933,000 proteins in our dataset, 493,260 can be associated with GO terms. In this study, we focus only on the subset of GO terms that describe molecular function. A total of 431,753 proteins are associated with GO terms that belong to that category.

4.2.2. Queries. An initial random set of 10,000 low-complexity queries was selected from the NR database. The selection criterion was that at least 15 sequence residues were filtered when using SEG (assuming that a smaller low-complexity segment will have only a minor effect on the e-value). Many of these sequences might still be robust to the existence of low-complexity segments; therefore, the set was reduced to a set of 3,461 queries for which a significant drop (at least 50%) in the number of matches was observed when the sequence was filtered with SEG. Of these, 2,115 can be associated with GO terms that describe molecular function. As discussed in Section 4.1.3, the relevant semantic space of a query does not necessarily match its relevant sequence space. For each GO significance level, we identified the set of viable queries (see Section 4.1.3). The procedure was repeated for significance levels 0.005, 0.001, 0.0001, 0.00001, 0.000001, and 0.0000001. The maximal number of viable queries was obtained at a level of 0.00001 with 264 queries, and therefore this significance level was selected as the *default level* for

[5]It should be noted that this approach is not perfect either, as a larger set of proteins that is associated with a specific GO term does not necessarily mean a less significant functional similarity, but can be merely a consequence of a statistical bias that is often observed as some protein families are more studied than others. However, it is a reasonable compromise, considering other existing approaches.

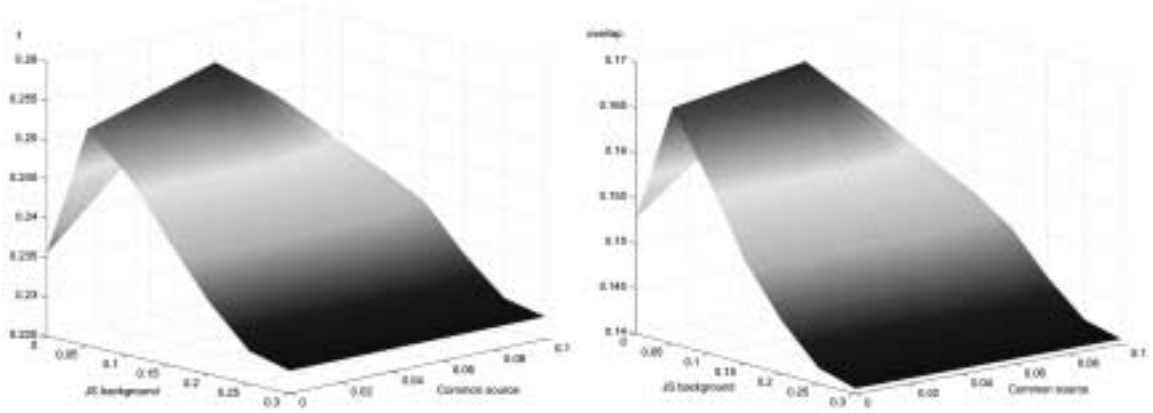

FIG. 7. Parameter optimization. A total of 28 sets were tested and evaluated using the harmonic average (**left**) and the overlap measure (**right**). See text for details.

subsequent analysis.[6] Finally, the 264 queries are divided randomly into a training set and a test set, each with 132 queries. **FN6**

4.3. Parameter optimization

Our method uses two parameters, D_1, the maximal distance from the background distribution above which a segment is deemed a low-complexity segment (Section 3.1.1) and D_2, the maximal distance from the common source, above which an alignment is considered suspicious (Section 3.1.2). If an alignment fails to pass these two filters, then the postnormalization method is triggered.

To select the best set of parameters, we ran our procedure on the training set with different values for D_1 and D_2. A total of 28 sets of parameters were tested, with D_1 ranging from 0 to 0.3 and D_2 ranging from 0 to 0.1. Each set of results was evaluated using the two performance measures described in Section 4.5. With both measures, the best set is $D_1 = 0.05$ and $D_2 = 0.05$ (Fig. 7). **F7**

4.4. Results

We compared the performance of our method to the performance of three other methods: BLAST, BLAST with SEG, and the latest version of BLAST that uses composition-based statistics.[7] Each one of the queries in the test set was compared against the NR database using BLAST, and the results were reevaluated using our postnormalization method with (POST+T) and without (POST) applying the filters described in Section 3.1. We also ran BLAST with the SEG filter on (SEG) and with composition-based statistics (COMP). For each method the match list was pruned such that only matches with evalue < 0.1 were retained. The results of these five procedures (BLAST, SEG, COMP, POST, and POST+T) were then evaluated using the metrics described below. **FN7**

4.4.1. Accuracy. To evaluate the accuracy of each method, we computed for each query the number of significant matches that also share common GO terms with the query sequence (denoted by N_c). The accuracy is defined as that number divided by N_g, the number of significant matches that are associated with any GO term

$$accuracy = N_c/N_g.$$

[6]The right significance level can vary from one query to another, but to simplify the analysis we focused on one level of significance at a time.

[7]Note that in the current version of NCBI's BLAST, the composition-based statistics is coupled to SEG, and SEG is applied to all database proteins (although not to the query sequence) by default whenever the former is invoked. Hence, it was difficult to test the performance of BLAST using composition-based statistics alone.

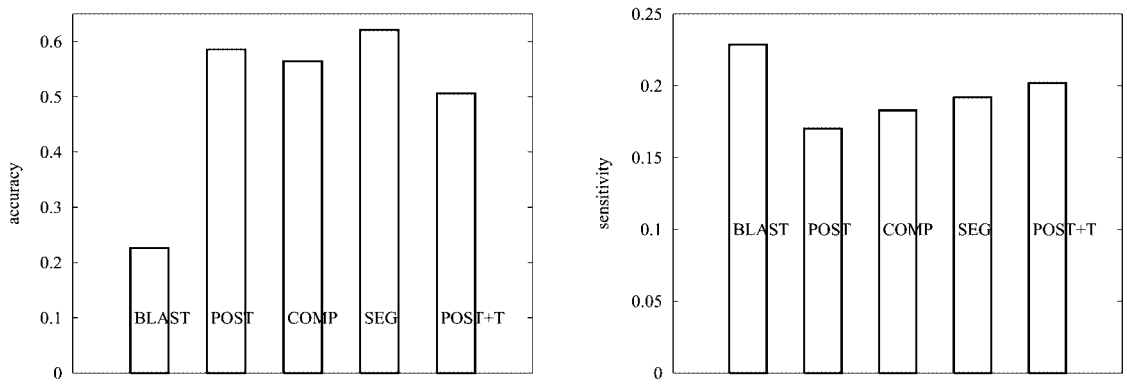

FIG. 8. (**a**) Accuracy of significant sequence matches. (**b**) Sensitivity of methods compared.

We repeated this procedure for each method and averaged the results over the set of 132 queries in our test set. The results of this procedure are given in Fig. 8a. As the graph indicates, all methods perform much better than the original BLAST program. As these methods filter many of the chance similarities, clearly their accuracy increases. In general, the more conservative a method is, the more accurate the results are expected to be. Indeed, the results are consistent with that postulation, and we note that SEG (that simply removes most of the low-complexity segments) is more accurate than COMP. Our postnormalization (POST) has high accuracy; however, when thresholds are used (POST+T), then the method is triggered only when the distances from the common source or the background distribution exceed the threshold. Consequently, the accuracy decreases. F8

4.4.2. Sensitivity. While the accuracy plot indicates the quality of the results, they might be misleading. As the number of hits reported decreases, the quality is expected to increase, especially if only very significant hits are reported. However, this can also affect substantially the *sensitivity*, as many meaningful hits might be missed as well. It is the balance between the two that determines the effectiveness of a method.

To measure the sensitivity of each method, we run a procedure similar to the one described above. For each query, we computed the number of significant matches that also share common GO terms with the query sequence (N_c) and divide it by the total number of database sequences that share common GO terms with the query (N_{cg})

$$sensitivity = \frac{N_c}{N_{cg}}.$$

That number is averaged over all queries. The results for all five methods are given in Fig. 8b. As the graph indicates, the trend is now reversed, and all methods perform worse than the original BLAST program. Surprisingly, SEG performs quite well while being also the most accurate (Fig. 8). Of all correction methods, the threshold-triggered postnormalization method (POST+T) performs the best.

4.5. Measures of overall performance

Both the accuracy and the sensitivity are incomplete measures of performance. In the ideal situation, we would like to maximize both; however, usually there is a tradeoff between the two and one would like to balance them somehow. There is no single measure that is widely accepted as the "right" measure. We report the results for three popular measures that account for both; the harmonic average, the overlap, and the ROC measure.

4.5.1. The harmonic average. A common measure of performance is the harmonic average of the accuracy (precision) and sensitivity (recall), defined as

$$f = 2 * \frac{accuracy * sensitivity}{accuracy + sensitivity}.$$

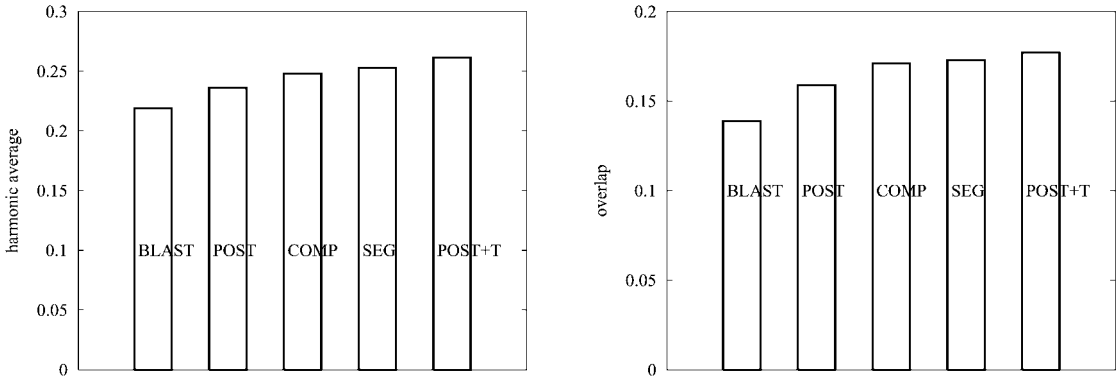

FIG. 9. Measures of overall performance. (**a**) Harmonic average. (**b**) Overlap.

This measure is often used in information retrieval to estimate the "break-even point" between the recall and precision thus giving an estimate of the "best possible" compromise between these two measures (Shaw *et al.*, 1997). The results over the test set are shown in Fig. 9a. As the graph indicates, the best performance is obtained with the POST+T method. A close second is SEG. BLAST comes last, as expected. **F9**

4.5.2. The overlap measure. Ideally, one would like to maximize the number of truly related sequences that are detected (true positives), while minimizing the number of true sequences missed (false negatives) as well as the number of unrelated sequences that are reported (false positives). One measure that combines all these elements is the overlap measure. Formally, if A is the group of truly related sequences (based on GO annotations) and B is the set of sequences that are reported as significantly similar (and are associated with any GO term), then the overlap is defined as

$$Overlap = \frac{|A \cap B|}{|A \cup B|}.$$

The results are shown in Fig. 9b and are in good agreement with the previous results (using the harmonic average).

4.5.3. The ROC measure. Our last and most important performance evaluation is ROC analysis (Hanley and McNeil, 1982). The ROC measure is the area under the curve that plots the number of positives detected versus the number of negatives, in a sorted list of results. If in the sorted list the separation between the two populations is perfect, then the area under the curve is maximized, and is minimized in the reverse situation. The larger the area under the curve, the better the method. To generate the ROC curve for each method, we pool together the results for all queries in the test set and order them based on their e-value. Next we determine for each match if it is a true positive or a false positive based on the common GO terms (considering only terms that are more significant than the default GO significance level, as described in Section 4.2.2). The results are shown in Fig. 10. All methods perform better than the original BLAST, but the differences between SEG and COMP are marginal. The best performance is obtained with POST+T. **F10**

4.6. Examples

Of the 3,461 queries in our dataset, SEG returns zero matches for 44 queries, and COMP returns zero matches for 19, when searched against the NR database. When using the POST+T method, none of the queries end up with zero matches. In this section, we give a few concrete examples of alignments between low-complexity sequences. The first example is SwissProt query P02734 (antifreeze peptide 4 precursor). This protein has a very high content of alanine (A). When searched against our NR database, BLAST reports 1,691 unique matches (multiple matches with the same library sequence are counted as one) with evalue < 0.1, most of which are purely due to repetitive alanines. SEG eliminates all matches.

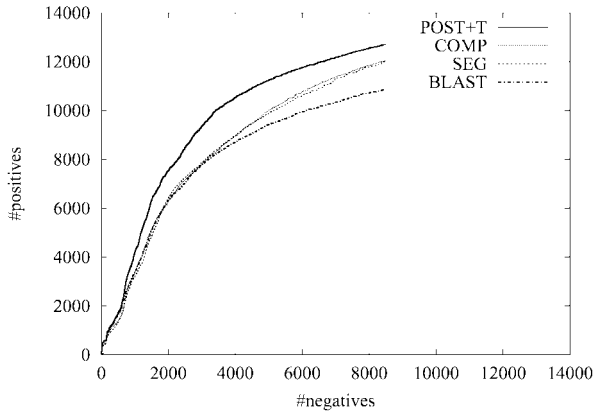

FIG. 10. ROC curves.

```
Query: 1  MRITEANPDPDAKAVPAAAAPSTASDAAAAAAAATAATAAAAAAAATAATAAAAAAAATAATA 60
          +RITEANPDP AKA PAA A   A+ AAAA A TA+ AAAAAAATAA AA AAA TAA A
Sbjct: 18 LRITEANPDPAAKAAPAAVADPAAA-AAAAVADTASDAAAAAAATAAAAAKAAADTAAAA 76

Query: 61 AKAAALTAANAAAAAAATA 79
          AKAAA TAA AA AAAATA
Sbjct: 77 AKAAADTAAAAAEAAAATA 95
```

FIG. 11. Alignment of SwissProt P02734 and P09031.

```
Query: 87  KSAKKLRSSNADAEEKLDRSHDKSDRGHDKSDRSHEK-LDRGHDKSDRGHDKS---DRDR 142
           KS ++ R  + D E   +RSHD++     K DR H +  DR  D+ +RG D+     RDR
Sbjct: 303 KSRERPRERDRD-ERTRERSHDRTRERDSKEDRHHHRDRDRTRDR-ERGRDRERDHGRDR 360

Query: 143 ERGYDKVDRERERDR----ERDRDRGYDKADREEGKERRHHRREELAPYPKSKKAVSRKD 198
           +R  D+ DR+R+RDR    ERDRDRG+D+  RE G++R   R  E A + + +  +  +D
Sbjct: 361 DRERDRRDRDRDRDRGRDYERDRDRGHDR-HRERGRDR--ERDYERASHERDRGHMHERD 417

Query: 199 EELDPMDP 206
            E    +P
Sbjct: 418 AEFANGEP 425
```

FIG. 12. Alignment of TrEMBL O60828 and TrEMBL Q94LP1.

COMP reports only one match, with evalue of 0.002 (the similarity of the query with itself is eliminated as well). POST+T keeps 218 of the original matches reported by BLAST (one example is given in Fig. 11), with many of the top matches being other antifreeze proteins. F11

The second example is the alignment of TrEMBL O60828 (polyglutamine binding protein) with TrEMBL Q94LP1 (putative U1 small nuclear ribonucleoprotein). The alignment is displayed in Fig. 12. BLAST F12 assigns evalue of 5e−11 to this alignment. Both proteins are marked as suspicious (JS-divergence from the background distribution exceeds the threshold, with values of 0.120 and 0.223 for the query and the library sequence), and the JS-divergence from the common source is relatively high (0.088). Consequently, the postnormalization method is triggered. The factor method assigns an evalue of 2.5e+18. However, the segment-profile model assigns a still significant evalue of 4.3e−04. This similarity is missed with COMP,

```
Query: 5   IILALFAVAAASAMPNYPPPPPKPYHAPPPPPHHAHPPPPPPPPAHYGHHAHPAPAPVVH 64
           ++L    + A +  P  PPPPPKPYHAPPPPP+HA PP   P P H          PVVH
Sbjct: 3   LVLFTAVIGALADYPAPPPPPPKPYHAPPPPPYHA-PPHHAPAPLH----------PVVH 51

Query: 65  TYPVHAPHAKCGANLLVGCAPSVAHAPCVPVHGHGHGYPAPAPHY 109
           TYPV AP AKCGANLLVGCAPSVAH PCVPVH H    P P  HY
Sbjct: 52  TYPVKAPAAKCGANLLVGCAPSVAHVPCVPVHPH----PPPPAHY 92
```

FIG. 13. Alignment of TrEMBL O01362 and PIR B48831.

```
Query: 2   TLAAFSLAVLALLLAPGPTNTLMAVAGASHGLARVLRLVPAELAGYLLTVVPLALAGAGL 61
           TL   S  V  +L+ PGPTNTL+  +G   G+ R   LV AE  GY++ +         L
Sbjct: 6   TLLKMSFYVSLVLIMPGPTNTLLLSSGLKVGVRRTRHLVMAEALGYVIAISLWGFFLCSL 65

Query: 62  MARAPGAALLLKLAAALWVMVLAVRLWCSAAA--GAEGFAIGAGRIFVTTALNPKALIFG 119
            A  P     +KL ++++++ LAV++W  + A    E   +G   +FVTT +NPKAL+F
Sbjct: 66  AASRPWLLDAIKLLSSVYILYLAVKMWTKSRALQHVEAGPVGFRDVFVTTLMNPKALLFA 125

Query: 120 LVLLPAP---SPAEIAARLLVFCALVVAVALLWGGFGA-LSHARAGEARAGGLRRIASGW 175
             L P     S A  A  + VF  ++  + + W   G  L+  RA       L R A+
Sbjct: 126 STLFPLEAFRSAAYFAWAMAVFLIVLAPIGIGWSYLGVLLTSRRAWAPHTPKLLRGAALV 185

Query: 176 LALVSVSLLLGAL 188
           L + S +L+   L
Sbjct: 186 LLMFSGTLMFSIL 198
```

FIG. 14. Alignment of TrEMBL O68026 and TrEMBL Q8XXM5.

while SEG reports a much shorter alignment (due to filtering) with an evalue of 0.040. Not much is known about these two proteins, and therefore it is hard to evaluate the biological significance of the match, however, the signal suggests possible homology.

The next example is of TrEMBL O01362 (vitelline membrane protein) with PIR B48831 (vitelline membrane protein homolog). BLAST assigns an evalue of 7.0e−30 (Fig. 13). The factor method assigns an evalue of 3.3e−08 while the segment-profile method assigns an evalue 8.2e−14. The divergence from the background distribution exceeds the threshold for both sequences (0.151 and 0.130), and they are marked as suspicious. However, the divergence from the common source is very small (0.028), and the original BLAST evalue is kept. COMP mises this similarity, while SEG reports an alignment which is one third of the alignment shown, again due to filtering. **F13**

Our fourth example is TrEMBL O68026 (hypothetical protein) with TrEMBL Q8XXM5 (probable transmembrane protein). BLAST assigns evalue of 5e−17 to this alignment (Fig. 14). The sequences do not pass the JS-filters and the postnormalization method is triggered. While the segment-profile model assigns an insignificant evalue of 0.972, the factor method assigns a significant evalue of 9.8e−13. A slightly shorter similarity (by 30 residues) is reported with COMP (with evalue of 2e−9), while SEG reports a much shorter alignment (and filtered for the most part) with evalue 0.006. Again, there is a strong signal that suggests homology. **F14**

Our last example is an interesting similarity between TrEMBL O68026 and TrEMBL Q9KQQ9 (both hypothetical proteins). BLAST assigns evalue of 2.0e−06 (Fig. 15). This similarity is missed by COMP and SEG. The JS-divergence from the common source is border-case (0.047) and therefore is still considered potentially meaningful. Furthermore, the factor method corrects the evalue to 4.6e−04, thus maintaining its significance. It is unclear what is the biological significance of this relationship, but both proteins are also similar to other transporter transmembrane proteins. **F15**

```
Query: 15  LAPGPTNTLMAVAGASHGLARVLRLVPAELAGYLLTVVPLALAGAGLMARAPGAALLLKL 74
           + PG   TL    G S G  R L ++  EL G  L  V   +  A +M R P    L K+
Sbjct: 18  ITPGMNMTLALTLGMSVGYRRTLWMMVGELLGVALVAVSAVVGIAAVMLRYPDIFTLFKI 77

Query: 75  AAALWVMVLAVRLWCSAAA------------GAEGFAIGAGRIFVTTALNPKALIFGLVL 122
             A +++ L V++W S               G++   +  G  FVT   NPK   F + L
Sbjct: 78  VGASYLVYLGVQMWRSRGKLAINIEQENTYQGSDWGLLVQG--FVTAIANPKGWAFMVSL 135

Query: 123 LP--APSPAEIAARLLVFCALV-----VAVALLWGGFGALSHARAGEARAGGLRRIASGW 175
           LP        +A +L V  A++     ++++L   G   L    +       L RIA
Sbjct: 136 LPPFIDQSLSLAPQLTVLVAIILLSEFISMSLYATGGKGLKRLLSQAHHVRLLNRIAGSL 195

Query: 176 LALVSVSL 183
           +A V + L
Sbjct: 196 MAGVGIWL 203
```

FIG. 15. Alignment of TrEMBL O68026 and TrEMBL Q9KQQ9.

5. DISCUSSION

Low complexity segments are abundant in protein sequences. These segments pose a problem for sequence comparison algorithms since their skewed composition leads to high scoring and statistically significant matches that are biologically insignificant. Current approaches to deal with this problem handle low complexity segments either through filtering or the use of composition-based statistics.

In this paper, we described a new heuristic method for re-estimating the significance of a match between the query sequence and a database sequence based on the alignment structure and information theoretic divergence measures. Our method is based on higher-order statistics that is more expressive than the first-order statistics of the sequence. The method was developed while trying to preserve the statistical framework that is used in BLAST and provide meaningful statistical estimates in the form of an e-value. We evaluated the results using the GO database as a reference set, and much attention was given to design sensible protocols of evaluation.

Our tests show that the existing methods are very successful in filtering spurious matches and already improve over BLAST. However, they sometimes suffer from certain artifacts (see Section 2.2). Our challenge was to improve over the existing methods. The tests indicate that our algorithm eliminates the excess of false positives that are otherwise abundant with low-complexity sequences, while performing well (in terms of overall performance measures, such as the harmonic average and ROC analysis) compared to the other methods. Moreover, our method does not generate counter-intuitive situations like the one observed with composition-based statistics (Section 2.2); nor does it filter the sequences as SEG while missing useful information. The proposed method is fast and can be applied to BLAST output files or can be integrated as part of the BLAST program.

Although our new estimates of the p-value might still deviate from the exact p-value, we believe that our approximation is more useful than discarding these similarities overall. Since one usually would like to know only whether the similarity is significant, the exact value is less important, and even deviations on the order of several orders of magnitude are acceptable. Note that our method will not work for low-complexity sequences that have diverged extensively through evolution, as it is almost impossible to discern those from chance similarities (nor will they be discovered by all other methods). However, our method already discovers many matches that are otherwise missed. The discovery of interesting low-complexity similarities opens the possibility of exploring less-studied proteins that were neglected thus far.

Currently our approach is using two parameters, which are the thresholds that trigger the postnormalization method (see Section 3.1). These thresholds were optimized over an independent training set. However, as is the nature of many thresholded decisions, they might be suboptimal since certain instances that are very close to the threshold might be wrongly classified. A better approach would be to use soft decisions, for example by using regression models or Bayesian analysis (and we are currently pursuing this direction).

Another drawback is that our method is composed of a number of elements which are based on different principles. One would wish for a single coherent method that captures the functionality of all the elements we propose here, and we believe this is an important open question that concerns about 20% of the current protein space. Despite these weaknesses, this paper presents new approaches to tackle this problem, and the overall method makes significant advances over the state of the art.

APPENDIX A

In addition to the segment-profile approach we have attempted two other methods to rescue significant matches between low-complexity sequences.

The similarity-profile criterion

The similarity-profile approach characterizes an alignment structure in terms of the set of similar segments in the alignment, but in a way different from the segment-profile approach. Formally, we are given an alignment $\mathbf{A}$ of length L (excluding gaps). Let $m_k(\mathbf{A})$ denote the number of similar segments of length k and $f_k(\mathbf{A})$, the fraction of the alignment that is in similar segments of length k, i.e., $f_k(\mathbf{A}) = \frac{k \cdot m_k(\mathbf{A})}{L}$. We refer to the set of fractions $\{f_k(\mathbf{A})\}_1^L$ as the *similarity profile* of the alignment. Similarly, we define an *identical segment* as a contiguous sequence of identities. The similarity profile is reduced to the *identity profile* when only identities are considered.

To characterize the similarity profiles in chance similarities, we analyzed the alignments in our training set. For each alignment we determined the similarity profile, and the distribution of fractions f_k was computed for each k, based on the complete set of alignments. The distributions $\hat{p}_k$ and cumulative distributions $\hat{P}_k$ for $k = 1, \ldots, 6$ are plotted in Fig. 16. As expected, the probability to observe $f_k \geq C$ decreases as k increases (very similar results are obtained when using the identity profile of the alignments). F16

Note that the distributions appear to follow the extreme-value distribution. This is not surprising since the dynamic programming algorithm that computes the alignments maximizes the similarity score by virtually maximizing the number of aligned residues that are assigned a positive score (i.e., positive matches). Although the positive matches are not guaranteed to form continuous segments, more often than not they do; thus, the algorithm practically maximizes the number and length of similar segments. As such, it also maximizes the fraction of the alignment that is in similar segments, and the extreme value distribution follows.

Given the true cumulative distributions P_k, one can compute, for each k, the probability to observe in a chance similarity a fraction f_k that is higher than $f_k(\mathbf{A})$, i.e., $P_k(f_k \geq f_k(\mathbf{A}))$. The true distributions are estimated from the empirical distributions by curve fitting (see Fig. 16). We approximate the significance (p-value) of the given alignment by assuming that $f_1, f_2, \ldots, f_L$ are independent random variables, distributed according to $p_1, p_2, \ldots, p_k$, i.e.,

$$P \triangleq pvalue(\mathbf{A}) = P_1(f_1 \geq f_1(\mathbf{A})) \cdot P_2(f_2 \geq f_2(\mathbf{A})) \cdots P_L(f_L \geq f_L(\mathbf{A})).$$

Note that $P_k(f_k \geq f_k(\mathbf{A})) = 1$ if $f_k(\mathbf{A}) = 0$. An alignment with a similarity profile such that $f_k(\mathbf{A}) = 0 \ \forall k > k_0$ will tend to have higher p-value if k_0 is small (1 to 4). An unrelated match will have most of its similarity profile in segments of length 1 or 2, resulting in a fairly high p-value. However, if a significant part of the alignment is in long similar segments (i.e., $f_k(\mathbf{A}) > 0$ for some k large), then the p-value of that alignment will be small, as it is quite unlikely to observe that in chance alignments. Finally, the e-value is obtained by applying the Bonferonni correction, $E = P \cdot N$, where N is the number of sequences in the database searched (the number of tests), which is on the order of 10^6 in current sequence databases.

It should be noted that segments of length > 6 are counted with segments of length 6, since the data is insufficient to generate reliable statistics for $k > 6$ in chance similarities. Extrapolation of parameters was not feasible either as no clear trend was obvious from the parameters for lower values of k. This leads to conservative estimates that tend to underestimate significance. However, this method can still detect similarities that are missed by the factor method.

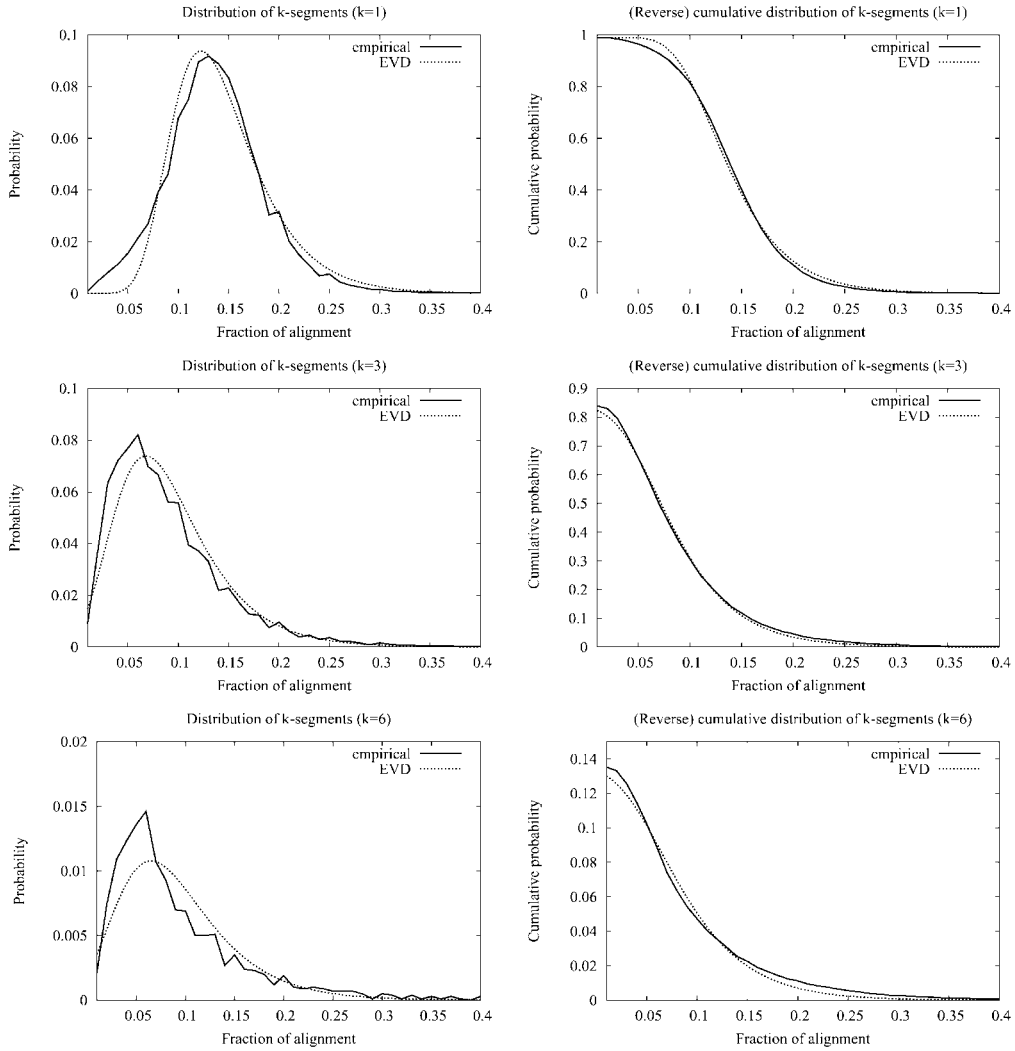

FIG. 16. Distributions of similar segments in chance similarities. Density and cumulative functions of f_k for $k = 1, 3, 6$ (data for $k > 6$ is sparse and noisy and therefore was not considered in our model). For example, in about 9% of chance alignments, roughly 12% of the aligned residues are in similar segments of length 1 (corresponds to the peak in the top left figure), while very few alignments have more than 35% of the aligned residues in similar segments of length 1 (where the tail of the distribution meets the x-coordinate). To model these distributions we used the extreme-value distribution, for $f_k > 0$.

The longest match criterion

The similarity-profile approach described above uses statistics of the alignment as a whole. However, the assumptions made (independence, ignoring other positions in the alignment in assessing the significance of the whole alignment) may not hold and the empirical distributions may not be easily characterized for long segments. Our second criterion focuses on the longest similar (or identical) segment. As before, the assumption is that the typical longest similar segment in a chance similarity is relatively short, while true similarities are characterized by relatively long identical and similar segment(s). Given an alignment of length L and a maximal similar segment of length k, the p-value is estimated based on the probability to observe such a segment by chance. It has been shown that the longest similar segment (sequence of positive matches) in alignments of random sequences is distributed as the extreme value distribution (Arratia *et al.*, 1986).

Indeed, the empirical distribution that was derived from the alignments in the training set follows nicely the extreme value distribution as is shown in Fig. 17. Similar results are obtained when the alignment is **F17**

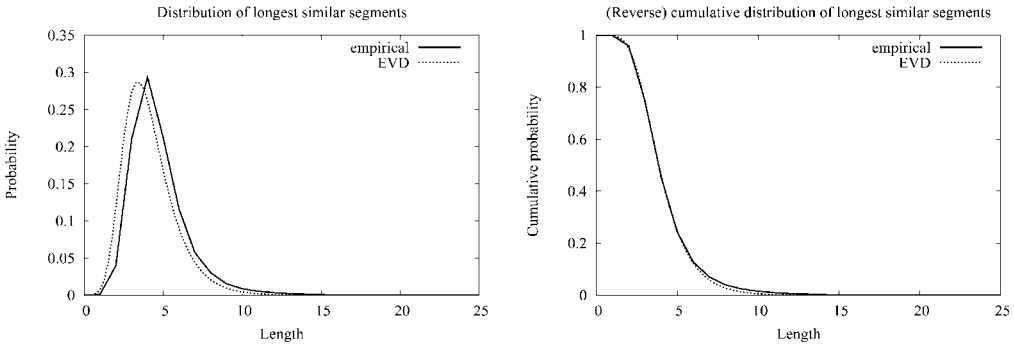

FIG. 17. Distribution of the longest similar segment in chance similarities.

characterized by the longest identical segment (the longest segment of identities).

Given an alignment **A**, we compute the length of the longest similar segment and the p-value of observing that length by chance, using the background distribution in Fig. 17. As before, the e-value is obtained by applying the Bonferonni correction.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under Grant No. 0133311 to Golan Yona.

REFERENCES

Alba, M.M., Laskowski, R.A., and Hancock, J.M. 2002. Detecting cryptically simple protein sequences using the SIMPLE algorithm. *Bioinformatics* 18, 672–678.

Altschul, S.F., and Gish, W. 1996. Local alignment statistics. *Methods Enzymol.* 266, 460–480.

Arratia, R., Gordon, L., and Waterman, M.S. 1986. An extreme value theory for sequence matching. *Ann. Stat.* 14, 971–993.

Arratia, R., and Waterman, M.S. 1994. A phase transition for the score in matching random sequences allowing deletions. *Ann. Appl. Prob.* 4, 200–225.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., and Sherlock, G. 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25, 25–29.

Attwood, T.K., Flower, D.R., Lewis, A.P., Mabey, J.E., Morgan, S.R., Scordis, P., Selley, J., and Wright, W. 1999. PRINTS prepares for the new millennium. *Nucl. Acids Res.* 27, 220–225.

Bateman, A., Birney, E., Durbin, R., Eddy, S.R., Finn R.D., and Sonnhammer E.L. 1999. Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucl. Acids Res.* 27, 260–262.

Claverie, J.M., and States, D.J. 1993. Information enhancement methods for large scale sequence analysis. *Comput. Chem.* 17, 191–201.

Cora, D., Di Cunto, F., Provero, P., Silengo, L., and Caselle, M. 2004. Computational identification of transcription factor binding sites by functional analysis of sets of genes sharing overrep-resented upstream motifs. *BMC Bioinformatics* 5, 57.

Corpet, F., Gouzy, J., and Kahn, D. 1999. Recent improvements of the ProDom database of protein domain families. *Nucl. Acids Res.* 27, 263–267.

Dembo, A., and Karlin, S. 1991. Strong limit theorems of empirical functionals for large exceedances of partial sums of i.i.d variables. *Ann. Prob.* 19, 1737–1755.

Dembo, A., Karlin, S., and Zeitouni, O. 1994a. Critical phenomena for sequence matching with scoring. *Ann. Prob.* 22, 1993–2021.

Dembo, A., Karlin, S., and Zeitouni, O. 1994b. Limit distribution of maximal non-aligned two-sequence segmental score. *Ann. Prob.* 22, 2022–2039.

El-Yaniv, R., Fine, S., and Tishby, N. 1997. Agnostic classification of Markovian sequences. *Advances in Neural Information Processing Systems* 10, 465–471.

Fuglede, B., and Topse, F. 2004. Jensen–Shannon divergence and Hilbert space embedding. *IEEE Int. Sym. Information Theory*.

Golding, G.B. 1999. Simple sequence is abundant in eukaryotic proteins. *Protein Sci.* 8, 1358–1361.

Hanley, J.A., and McNeil, B.J. 1982. The meaning and use of the area under the receiver operating characteristic (ROC) curve. *Radiology* 143, 29–36.

Henikoff, S., and Henikoff, J.G. 1992. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* 89, 10915–10919.

Henikoff, J.G., and Henikoff, S. 1996. Using substitution probabilities to improve position-specific scoring matrices. *Comp. Appl. Biosci.* 12(2), 135–143.

Karlin, S., and Altschul, S.F. 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA* 87, 2264–2268.

Karlin, S., and Altschul, S.F. 1993. Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc. Natl. Acad. Sci. USA* 90, 5873–5877.

Karplus, K., Karchin, R., and Hughey, R. 2003. Calibrating E-values for hidden Markov models with reverse-sequence null models. **AU2**

Kullback, S. 1959. *Information Theory and Statistics*, John Wiley, New York.

Lin, J. 1991. Divergence measures based on the Shannon entropy. *IEEE Trans. Info. Theory* 37(1), 145–151.

Lord, P.W., Stevens, R.D., Brass, A., and Goble, C.A. 2003. Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation. *Bioinformatics* 19, 1275–1283.

Mott, R. 2000. Accurate formula for P-values of gapped local sequence and profile alignments. *J. Mol. Biol.* 300, 649–659.

Mott, R., and Tribe, R. 1999. Approximate statistics of gapped alignments. *J. Comp. Biol.* 6, 91–112.

Promponas, V.J., Enright, A.J., Tsoka, S.T., Kreil, D.P., Leroy, C., Hamodrakas, S., Sander, C., and Ouzounis, C.A. 2000. CAST: An iterative algorithm for the complexity analysis of sequence tracts. *Bioinformatics* 16, 915–922.

Romero, P., Obradovic, Z., Li, X., Garner, E.C., Brown, C.J., and Dunker, A.K. 2001. Sequence complexity of disordered protein. *Proteins* 42, 38–48.

Schaffer, A.A., Aravind, L., Madden, T.L., Shavirin, S., Spouge, J.L., Wolf, Y.I., Koonin, E.V., and Altschul, S.F. 2001. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucl. Acids Res.* 29, 2994–3005.

Shaw, W.M., Burgin, R., and Howell, P. 1997. Performance standards and evaluations in IR test collections: Cluster-based retrieval models. *Information Processing and Management* 33, 1–14.

Smith, T.F., Waterman, M.S., and Burks, C. 1985. The statistical distribution of nucleic acid similarities. *Nucl. Acids. Res.* 13, 645–656.

Waterman, M.S., and Vingron, M. 1994. Rapid and accurate estimates of statistical significance for sequence data base searches. *Proc. Natl. Acad. Sci. USA* 91, 4625–4628.

Wootton, J.C. 1994. Sequences with 'unusual' amino acid compositions. *Curr. Opin. Struct. Biol.* 4, 413–421.

Wootton, J.C., and Federhen, S. 1993. Statistics of local complexity in amino acid sequences and sequence databases. *Comp. Chem.* 17, 149–163.

Wren, J.D., and Garner, H.R. 2004. Shared relationship analysis: Ranking set cohesion and commonalities within a literature-derived relationship network. *Bioinformatics* 20, 191–198.

Yona, G., and Levitt, M. 2000a. A unified sequence-structure classification of proteins: Combining sequence and structure in a map of protein space. *Proc. RECOMB 2000*, 308–317.

Address correspondence to: **AU3**
Golan Yona
Department of Computer Science
Cornell University
Ithaca, NY

E-mail: golan@cs.cornell.edu

Author
Right running head okay as shown (short title)?

AU1
Please provide Key words.

AU2
Publication?

AU3
Please provide street address.