A BLAST from the past: revisiting BLAST's E-value

Yang Young Lu[1], William Stafford Noble[2,3], and Uri Keich*[4]

[1]Cheriton School of Computer Science, University of Waterloo
[2]Department of Genome Sciences, University of Washington
[3]Paul G. Allen School of Computer Science and Engineering, University of Washington
[4]School of Mathematics and Statistics F07, University of Sydney

Abstract

The Basic Alignment Search Tool, BLAST, is an indispensable component of the genomic research toolbox. After it was introduced in 1990, BLAST quickly established itself as the canonical tool for sequence similarity search in large part thanks to its meaningful statistical analysis of its output. Specifically, BLAST reports the E-value of each reported alignment, which is defined as the expected number of alignments that will score at least as high as the observed alignment score, assuming that the query and the database sequences are randomly generated.

In this work we critically reevaluate BLAST's E-values, showing that they can be at times significantly conservative while at others much too liberal. As an alternative, we offer an approach that is based on generating a small sample from the null distribution of random optimal alignments, and asking whether our original optimal alignment score is consistent with this sample. In contrast with BLAST's E-values, our approach offers a valid statistical significance analysis, in the sense that it did not deliver inflated significance estimates in any of our extensive experiments. Moreover, although our method is slightly conservative, it is often significantly less so than the BLAST E-value. Indeed, in cases where BLAST's analysis is valid (i.e., not too liberal), our approach seems to deliver a greater number of correct alignments. A significant advantage of our approach is that it works with any reasonable choice of substitution matrix and gap penalties, avoiding BLAST's requirement to pre-compute statistics for a limited combinations of matrices and penalties.

Keywords: pairwise sequence similarity, similarity search, BLAST, E-value, FWER

*Correspondence: uri.keich@sydney.edu.au

1 Introduction

The Basic Alignment Search Tool, BLAST, is a cornerstone of genomics research. It enables researchers to search large sequence databases for similar subsequences, facilitating the identification of homologous and orthologous genes, proteins and conserved domains in both inter- and intra-species analyses. As such, BLAST has been instrumental in a wide range of biological research areas, from understanding the evolutionary relationships between species to predicting protein structure and function to designing new drugs. BLAST's popularity is borne out by the number of citations it has received: the original BLAST paper [1] has over 107,000 citations, and the followup paper [2] has over 83,000 citations (Google Scholar, 10/2023).

BLAST was not the first similarity search tool; however, it quickly became the de facto standard thanks to two key advantages it had over FASTA [17], its key competitor at the time. The first was that BLAST ran a lot faster than FASTA. The second was that BLAST provided the user with meaningful statistical analysis of its output.

BLAST evaluates the significance of a reported local alignment based on the assumption that the distribution of the score of an optimal local alignment, or in BLAST's terminology, a high-scoring segment pair (HSP), is a Gumbel distribution. This assumption is well grounded in theoretical asymptotic results that cover ungapped alignments [12, 9]. In practice, the parameters of the Gumbel distribution must be pre-computed, based on simulations using specific substitution matrix and gap parameters. The extension of this significance estimation approach to the more common case of gapped alignments is largely based on empirical evidence [3] and some limited analytic results [4].

In practice, the Gumbel-based p-value BLAST computes for an HSP must be adjusted to control for multiple hypothesis testing: although the p-value provides valuable information about an alignment between the query and a specific database sequence, in a typical application of BLAST we seek to align the query against a database containing many sequences. Accounting for this multiple hypothesis testing scenario is challenging because the database typically contains many closely related sequences. Therefore, standard methods that rely on an independence assumption are not valid here.

The solution chosen for BLAST was to express the significance in terms of the expected number of alignments, or the *E-value*. BLAST's E-value is defined as the expected number of alignments that will score at least as high as the observed alignment score, assuming that the query and the database sequences are randomly generated by an independent and identically distributed (iid) process. This calculation relies on the observation that the number of local alignments between two randomly drawn sequences that score above a given value follows a Poisson distribution [13, 12]. This distribution allows BLAST to compute the expected number of random alignments between the query and the database sequence that score above a specified threshold, and summing these expectations over the entire database yields the E-value (details in Section 2.1). Overall, BLAST's E-value offers a very simple way to adjust the Gumbel-based p-value of a single HSP, so that it accounts for all the different possible alignments that the database might offer the query. On the other hand, the E-value represents a significant departure from the classical statistical approach to the problem of multiple hypothesis testing.

We are now over 30 years past the initial release of BLAST, and in view of the significant advances in computing power we sought to reevaluate the E-value approach. Via extensive simulated draws from the null we show that, while generally reasonable, BLAST's E-values can at times be overly conservative, while at others, arguably more alarmingly, they can be too liberal, i.e., BLAST is inflating the significance of the reported alignments. For example, we noted an E-value of 0.05 or smaller reported in over 10% of random alignments.

We therefore offer an alternative significance analysis that relies on generating a sample of size m (we used $m = 50$) from the distribution of the maximal alignment score (Figure 1). We then compute a p-value, assessing how unlikely it is that our original maximal alignment score came from the same null sample, assuming all scores were generated by a Gumbel distribution. This computation is done by a novel adaptation of the rationale behind the independent two-sample t-test to our setting.

Compared with BLAST's E-values, our approach has several advantages and one glaring disadvantage. On the positive side, first, our significance analysis is expressed in terms of classical statistical approach to the problem of multiple hypothesis testing: we compute a p-value, and we control the canonical family-wise error rate (FWER). Specifically, if we report all alignments whose p-value is ≤ 0.05, then the probability that even one random alignment to the query is reported is ≤ 0.05. Second, our approach does not rely on

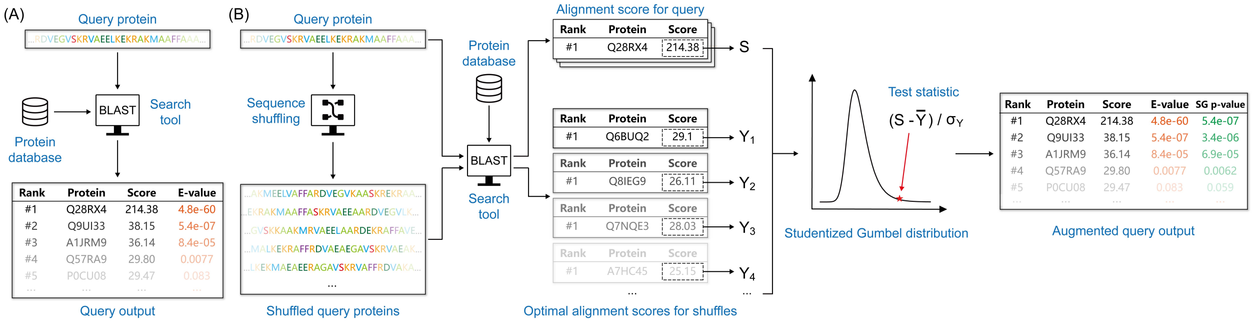
(A)
Query protein
...RDVEGVSKRVAEELKEKRAKMAAFFAAA...
BLAST
Search tool
Protein database
Rank Protein Score E-value
#1 Q28RX4 214.38 4.8e-60
#2 Q9UI33 38.15 5.4e-07
#3 A1JRM9 36.14 8.4e-05
#4 Q57RA9 29.80 0.0077
#5 P0CU08 29.47 0.083
...
Query output
(B)
Query protein
...RDVEGVSKRVAEELKEKRAKMAAFFAAA...
Sequence shuffling
...AKMEELVAFFARDVEGVKAASKREKRAA...
...EKRAKMAAFFASKRVAEEAARDVEGVLK...
...GVSKKAAKMRVAEELAARDEKRAFFAVE...
...MALKEKRAFFRDVAEAEGAVSKRVAEAK...
...LKEKMAEAEERAGAVSKRVAFFRDVAKA...
...
Shuffled query proteins
Protein database
BLAST
Search tool
Alignment score for query
Rank Protein Score
#1 Q28RX4 214.38
S
Rank Protein Score
#1 Q6BUQ2 29.1
Y_1
Rank Protein Score
#1 Q8IEG9 26.11
Y_2
Rank Protein Score
#1 Q7NQE3 28.03
Y_3
Rank Protein Score
#1 A7HC45 25.15
Y_4
...
Optimal alignment scores for shuffles
Test statistic
$(S - \bar{Y}) / \sigma_Y$
Studentized Gumbel distribution
Rank Protein Score E-value SG p-value
#1 Q28RX4 214.38 4.8e-60 5.4e-07
#2 Q9UI33 38.15 5.4e-07 3.4e-06
#3 A1JRM9 36.14 8.4e-05 6.9e-05
#4 Q57RA9 29.80 0.0077 0.0062
#5 P0CU08 29.47 0.083 0.059
...
Augmented query output

Figure 1: **The workflow of (A) BLAST and (B) the Studentized Gumbel (SG) p-value calculation.**

any pre-computed parameters, so it is applicable anywhere the null sample generation is feasible. Third, we show that in those cases where BLAST's E-value is statistically valid (i.e., it is not liberal) we deliver more statistical power (i.e., a greater number of interesting alignments). On the negative side our approach imposes a significant runtime penalty (a factor of m) and would not have been viable back when BLAST was introduced. On other hand, today, with our vastly improved computing power, the extra computational expense can be easily justified for many applications of BLAST.

In addition to comparing with BLAST, we include BLAST's competitor, FASTA , in our analyses. We find that, although FASTA's E-value is defined and computed differently than BLAST's, much of the above still holds for FASTA as well. The one exception is that unlike BLAST, FASTA also does not rely on pre-computed parameters, and hence its significance analysis is at least as widely applicable as our approach.

Finally, it is important to note that when we refer to BLAST in this paper we specifically mean `blastp`, the version that is applicable to amino acid sequences. Accordingly, all our analyses were done in the context of databases of sequences of proteins or protein domains. This was partly due to the heavy runtime penalty our approach incurs.

2 Background

2.1 Calculating the BLAST E-value

BLAST evaluates the significance of a reported local alignment based on the assumption that the distribution of the score of an optimal HSP is a Gumbel distribution. Specifically, the p-value of an HSP with score s, i.e., the probability of finding an HSP with score $\geq s$, in a randomly generated query of length n and a database sequence of length l can be approximated by

$$1 - \exp\left(-Knle^{-\lambda s}\right), \tag{1}$$

where K (related to location) and λ (rate, or 1/scale) are parameters of the Gumbel distribution. In practice, these parameter values are pre-computed from extensive simulations using specific substitution matrix and gap parameters.

Controlling for multiple hypotheses is accomplished by reporting the E-value. Returning to local alignments between a query and a single database sequence, we note that it was shown that X_s, the number of alignments between two randomly drawn such sequences that score above s, has a Poisson distribution [13, 12]. This was theoretically established in the ungapped case and empirically observed for the gapped case as well. Recall that for a Poisson(μ) random variable X we have, $P(X > 0) = 1 - e^{-\mu}$, and note that $X_s > 0$ if and only if there exists an alignment that score $\geq s$. It follows from (1) that

$$1 - e^{-\mu_s} = P(X_s > 0) \approx 1 - \exp\left(-Knle^{-\lambda s}\right).$$

That is, μ_s, the expected number of random alignments between the query and the database sequence that score $\geq s$, is approximated by

$$Knle^{-\lambda s}. \tag{2}$$

Expectation is additive, so if we add these expectations over all N database sequences of lengths $l_1, l_2, \ldots, l_N$, then we get that the expected number of random alignments between the query and the entire database that score $\geq s$ is approximately

$$\sum_{1}^{N} Knl_i e^{-\lambda s} = KnLe^{-\lambda s}, \tag{3}$$

where $L = \sum_1^N l_i$ is the total length of the database. This is essentially BLAST's E-value, except that BLAST employs a sophisticated edge effect correction to account for the fact that an alignment cannot start arbitrarily close to the end of a sequence. Specifically, the nl term in (2) is replaced by a complicated term that factors in that edge effect. To compute the final E-value we note that (3) can be derived from (2) by multiplying by L/l, so BLAST employs the same adjustment to obtain the overall, database-wide E-value from the edge-corrected, sequence-specific E-value.

2.2 Calculating the FASTA E-value

FASTA also reports its significance estimates via the E-value; however, there is a subtle difference between its goal and that of BLAST, which impacts how the two methods evaluate the significance of their respective results. Specifically, BLAST aims to report all sufficiently high-scoring local alignments between the query and the database, and it evaluates each one by computing the expected number of random alignments of the same or higher score. FASTA, on the other hand, aims to find all *database sequences* that are sufficiently similar to the query. "Similar" here means that the two sequences share a sufficiently significant local alignment, but FASTA typically only reports the top alignment between the two sequences. Accordingly, it assesses the significance of the similarity by computing an E-value, which it defines as the expected number of random database sequences that will have the same or higher similarity score.

FASTA also takes a different approach to BLAST in how it computes its E-value. Briefly, its default E-value computation (called "REGRESS1") first regresses all observed similarity scores against the log of the database sequence length, in order to find the mean and variance of the null sequence similarity score as a function of the database sequence length. Here, the underlying assumption is that the vast majority of the database sequences offer a random match to the query. FASTA employs some heuristics to take out of the estimation process the few sequences that might be truly related to the query. The Gumbel distribution is then fitted to the normalized similarity scores, and the fitted distribution is used to estimate the probability of seeing the observed normalized sequence similarity score between the query and a single random database sequence.

To overcome the unknown dependence structure in the database, FASTA converts the single sequence p-value to a database-corrected E-value similarly to BLAST. In FASTA's case this is done by multiplying the p-value by the number of sequences in the database, which coincides with the Bonferroni correction for multiple testing: one for each database sequence. Importantly, unlike BLAST, FASTA's significance analysis does not depend on pre-computed parameters, so it can be applied to any combination of a substitution matrix and gap penalties, making it more widely applicable than BLAST.

3 The Studentized-Gumbel (SG) p-value

3.1 Approach

Consider a canonical hypothesis testing problem, where you are given an observation X and you want to test the null hypothesis that it came from, say, a $N(\mu, \sigma^2)$ distribution. If the parameters (μ, σ^2) are known, then you standardize X by computing its so-called Z-value, $Z = (x - \mu)/\sigma$, and compute a p-value based on the fact that $Z \sim N(0, 1)$.

Suppose next that μ and σ are unknown but that you can generate a small sample $Y_1 \ldots, Y_m$ from the null distribution. If that null is again a $N(\mu, \sigma^2)$ distribution then you can use $(\bar{Y}, \sigma_Y^2)$, the sample mean and variance of Y, in lieu of the unknown (μ, σ^2). Specifically, your test statistic is now the studentized value of X, defined as $T = (X - \bar{Y})/\sigma_Y$. You can next assign a p-value to X by realizing that, up to a constant of $\sqrt{1 + 1/m}$, T has a t-distribution with $m - 1$ degrees of freedom. Indeed, $T/\sqrt{1 + 1/m}$ coincides with the statistic of the independent two-sample t-test comparing the X and Y samples (of sizes 1 and m).

We are interested in testing whether S, the score of the optimal local alignment across the entire database, is an observation from the null. Here we define the null as the distribution of the score of the optimal alignment to a random shuffling of the given query. Ideally, we would have searched all possible permutations of the query against the database, noting the optimal match to each of the shuffled queries. This would have allowed us to precisely characterize the null, which is analogous to knowing (μ, σ^2) above. Of course, this approach is not practical even for very short queries, so instead we opt, as in the unknown (μ, σ^2) case above, to generate a small sample from the null. Specifically, we search each of m random shuffles of the query (we used $m = 50$ here) against the database, noting the corresponding scores of the m optimal alignments $Y_1, \ldots, Y_m$.

We next define our statistic again through studentizing the observed optimal value S: $T = (S - \bar{Y})/\sigma_Y$. To assign a p-value to T we need to know its distribution under the null, i.e., when S is also drawn from the null. This is possible if the null distribution depends only on a location and scale/rate parameters, as it does in the normal null example.

Fortunately, the Gumbel cumulative distribution function (CDF), $F(x) = \exp(-e^{-\lambda(x-\mu)})$, is defined only in terms of its rate λ and location μ. Moreover, BLAST's entire approach is predicated on the Gumbel approximation to the distribution the optimal alignment score between two random sequences. Intuitively, it makes sense that this Gumbel approximation would extend to S, which further maximizes the score over all database sequences: it is easy to see that a maximum of independent Gumbel random variables (RVs) with the same rate λ, is again a Gumbel RV with the same λ, and BLAST assumes that λ depends only on the substitution matrix and the gap penalties. The left column of Figure 2 shows those fits for a couple of examples that we highlight, where in each case we fitted the Gumbel distribution to a sample of size 1e6 from the null distribution of S.

Additionally, the left columns of Figures S3 and S4 demonstrate that, consistent with our expectation, the Gumbel fit improves with increasing query length. Keeping in mind the stochastic nature of the sample, we provide in Figure S1A an example of what those probability plots should look like when the sample of 1e6 points is indeed generated from the Gumbel distribution. With that reference figure in mind, it is clear that the fit for the shortest sequence in Figure S3 is not perfect — a fact we will return to below.

3.2 Computing the Studentized Gumbel (SG) distribution

Recall that our test statistic is the studentized value of S, $T = (S - \bar{Y})/\sigma_Y$. When S and $Y_1, \ldots, Y_m$ are $N(\mu, \sigma^2)$ the normal-studentized RV T has (up to a constant) a t-distribution with $m-1$ degrees of freedom. When S and $Y_1, \ldots, Y_m$ are Gumbel with location μ and a rate λ, then again the distribution of T is invariant of μ and λ, and this distribution depends only on m. However, to the best of our knowledge, unlike in the normal case, this distribution does not have a name nor is it tabulated. We therefore refer to it as the Studentized-Gumbel(m) (SG$_m$) distribution. Figure S2A shows an estimate of the density of SG$_{50}$ using a sample of 1e7 points.

We used Monte Carlo simulations with importance sampling to estimate the CDF of SG$_{50}$ as follows. We divided the range of positive values T can attain between 0 and 45 into bins of size 0.001. The upper limit of 45 was arbitrarily chosen (more on that below), and the lower limit of 0 was chosen so as not to waste computational resources on negative values of T: if your best score is less than the mean of a null sample then it is probably not an interesting one.

We next drew N=1e10 samples of size $m = 50$ from the Gumbel distribution with $\mu = 0$ and scale $1/\lambda = 3$ (we could have used any λ, except this choice impacts the parameters used in the importance sampling). This gave us N independent draws $(\mu_1, \sigma_1), \ldots, (\mu_N, \sigma_N)$ of the sample mean $\bar{Y}$, and the sample standard deviation σ_Y of the null samples of $Y_1, \ldots, Y_m$. If we had drawn the values of S from the same Gumbel distribution, then we would have been rather limited in our ability to compute very small p-values (corresponding to very large values of S), so instead we used importance sampling, as explained next.

We drew N values of S from a Gumbel distribution with $\mu = 33$ and scale $1/\lambda = 15$. This significant shift in location allowed us to sample much higher values of S than had we used the same $\mu = 0$ that was used to generate the Y sample. At the same time we increased the scale (the reciprocal of the rate) so as to make sure we are effectively sampling S across a wide range of values. To account for the fact that S was not sampled from a Gumbel$(0, 1/3)$ we weighted each observed value s_i by its Radon-Nikodym derivative, which in this case is $f_{G(0,1/3)}(s)/f_{G(33,1/15)}(s)$, where $f_{G(\mu,\lambda)}(s) = \lambda \exp[-\lambda(s-\mu) - e^{-\lambda(s-\mu)}]$ is the Gumbel(μ, λ)

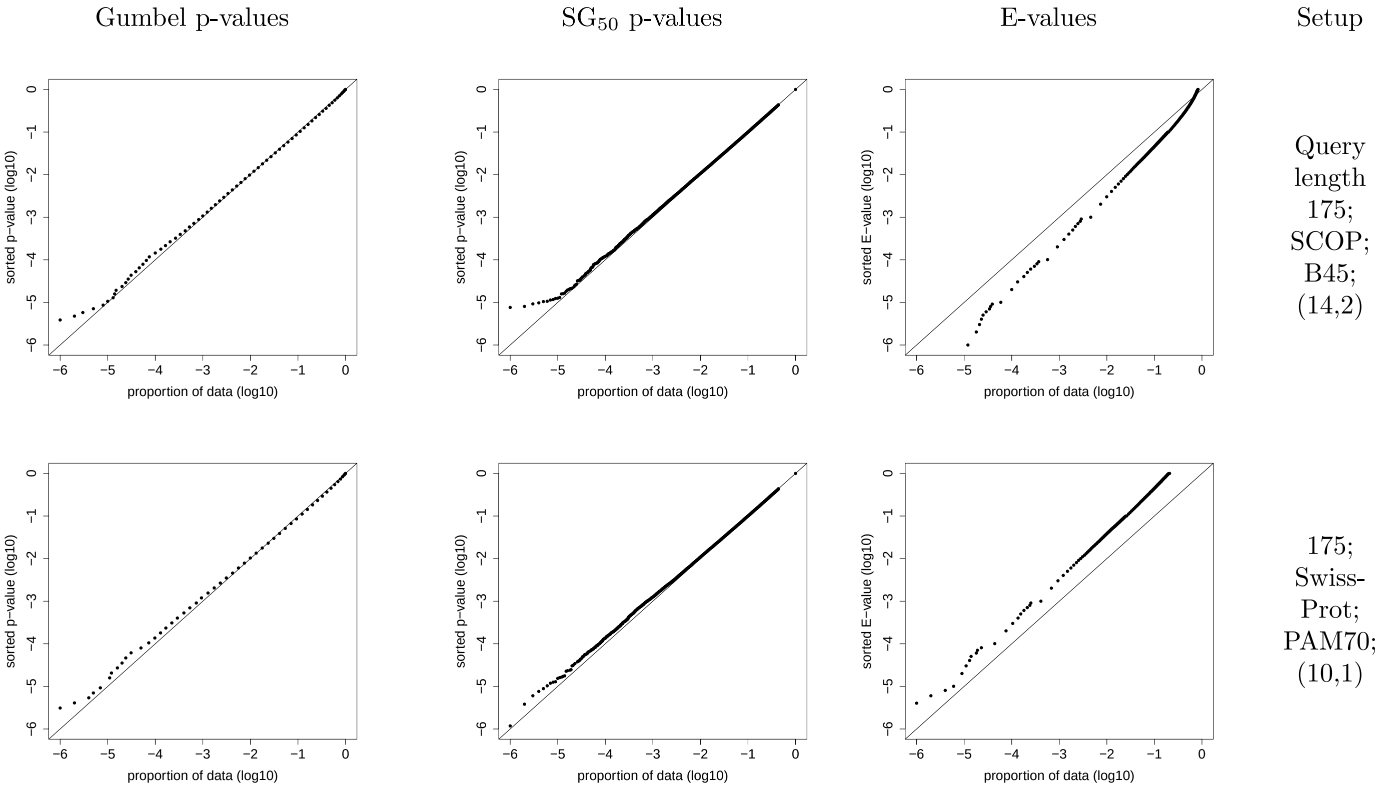

Figure 2: **BLAST E-value can be liberal as well as conservative**. The plots were made based on a sample generated by using BLAST to search 1e6 shuffles of a query of length 175 amino acids against the SCOP database with BLOSUM45 and its BLAST default gap penalties of (14,2). The bottom row is similarly based 1e6 BLAST searches against the Swiss-Prot database with PAM70 and its BLAST default gap penalties of (10,1). The left panel examines the fit of the optimal alignment score to the Gumbel, the middle one the validity of our SG$_{50}$ p-values, and the right panel examines the validity of the E-values (see Section 4.1 for details).

density. That weight was assigned to the bin that contained the studentized value of s_i: $t_i = (s_i - \mu_i)/\sigma_i$.

For each bin we then considered the weights $w_1, \ldots, w_n$ of all the studentized samples that fell in that bin or in bins corresponding to higher studentized values. If we let t denote the center of the considered bin then the right tail of the CDF at t is estimated as $G(t) = \sum_{i=1}^{n} w_i/N$. Additionally, we computed the sample standard deviation of those same weights, and the first bin for which the ratio of this standard deviation to $G(t)$, equivalently the coefficient of variation, was ≤ 0.01 defined our precision cutoff. That is, we did not try to estimate the tail probability, G of values greater than that cutoff. In practice we further truncated the cutoff of 34.599 to 34. Figure 3.2B shows the (log10) of the computed tail probabilities.

3.3 Computing the SG$_m$ p-values

Equipped with the right tail probabilities of the SG$_m$ distribution it is conceptually straightforward to compute the p-values of the observed scores $S_1, \ldots, S_j$ of aligning a given query to the database. Specifically, we shuffle the same query m times (here we use $m = 50$) and apply BLAST (or more generally, the search tool) to the m shuffled queries in exactly the same way it was applied to the original query.

Let $Y_1, \ldots, Y_m$ denote the m scores of the maximal alignments for each of the m shuffles. We use the sample moments of Y to studentize the observed scores and then look up the corresponding entries in the table of SG$_m$ tail probabilities we computed in the previous section. Note that any studentized value that exceeds the accuracy cutoff of 34 is assigned a p-value of $G(34)$=8.716e-15. Similarly, any value ≤ 0 is assigned a p-value of 1. An algorithmic description of this process is available in Algorithm 1.

Algorithm 1: The SG p-value algorithm

```
Data: A query protein $q$, a protein database $D$, a search algorithm $S(q, D)$ that returns a ranked list
      of alignment scores, a list of observed alignment scores $s = (s_1, \dots, s_k)$, a specified number of
      shuffles $m$, $T_p$ the two-column table of right tail probabilities of SG$_m$ computed in Section 3.2:
      first column is the bin center, the second is the tail probability.
Result: $PV = (p_1, \dots, p_k)$ - the list of SG$_m$ p-values of $(s_1, \dots, s_k)$
$Y \leftarrow []$;
for $i = 0;\ i < m;\ i++$ do
    $\tilde{q} \leftarrow$ shuffle($q$);
    Add $Y_i = \max(S(\tilde{q}, D))$ to $Y$;
end
$PV \leftarrow []$;
Calculate the mean $\bar{Y}$ and the standard deviation $\sigma_Y$ of $Y$;
for $j = 0;\ j < k;\ j{+}{+}$ do
    Calculate the test statistic $t_j = \frac{s_j - \bar{Y}}{\sigma_Y}$;
    if $t_j \leq 0$ then
        $p_j = 1$;
    end
    else if $t_j > m_x := \max T_p[, 1]$ then
        /* $m_x$ is the maximal bin recorded in the SG$_m$ table $T_p$                    */
        $p_j = T_p[m_x, 2]$;
        /* The minimum p-value recorded in $T_p$                                          */
    end
    else
        Find the bin $i_j$ in which $t_j$ falls and set $p_j = T_p[i_j, 2]$;
    end
    Add $p_j$ to $PV$;
end
```

3.4 Controlling the family-wise error rate (FWER)

We first describe a straightforward procedure that determines which of the k BLAST alignments of the given query to the database is reported as significant. Let $S_1, \dots, S_k$ be the corresponding scores of the local alignments returned by BLAST, and let α be the selected significance threshold (canonically $\alpha = 0.05$). Then,

1. Apply Algorithm 1 to obtain the SG$_m$ p-values $p_1, \dots, p_k$ of the corresponding alignment scores $S_1, \dots, S_k$.

2. Report as significant any alignment with p-value $p_i \leq \alpha$.

We next argue that, if we assume that our SG p-values are valid, that is, that under the null hypothesis $P[\text{p-value}(S) \leq \alpha] \leq \alpha$, then our procedure controls the FWER among the reported local alignments to the given query: the probability that even one random/null-generated alignment is reported is $\leq \alpha$.

Note first that because we are only concerned about reporting random alignments, we can assume without loss of generality the worst case scenario that all alignments to the query are null generated, or equivalently, that the query is randomly shuffled. Let q^m_α denote our $1 - \alpha$ quantile of the SG$_m$ distribution, $S = \max S_i$, and $\bar{Y}$ and σ_Y denote the sample moments of the randomly drawn $Y_1, \ldots, Y_m$ in Algorithm 1. Then

$$\begin{aligned} P(\exists i : p_i \leq \alpha) &= P\left(\exists i : \frac{S_i - \bar{Y}}{\sigma_Y} \geq q_\alpha^m\right) \\ &= P\left(\frac{S - \bar{Y}}{\sigma_Y} \geq q_\alpha^m\right) \\ &= P(\text{p-value}(S) \leq \alpha) \leq \alpha, \end{aligned}$$

where the last inequality follows by our assumption that the SG p-values are valid. We conclude this section with two comments:

1. The above discussion is predicated on our SG p-values being valid. First note that, ignoring the sampling errors when estimating the SG_m distribution (Section 3.2), if the null distribution of the maximal alignment score S is indeed a Gumbel then our p-values are valid. Our simulated draws show that, just as in the case of BLAST, this is a reasonable assumption for longer queries (e.g., the left columns of Figures S3 and S4). Moreover, as we will see below, even when the fit to the Gumbel is clearly not perfect (e.g., the shortest queries in the above figures), in practice our SG p-values are conservatively biased, and hence valid, in those cases.

2. While we mostly referred to BLAST throughout this section, its content applies just as well to other alignment/similarity search tools including FASTA, SSEARCH (A Smith-Waterman alignment algorithm implemented in the FASTA program package [16]) and AB-BLAST (a rebranded version of WU-BLAST), as long as the SG p-values are valid — a point which we address empirically below.

4 Methods

To compare our SG p-values with E-values we used two types of experiments. The first was designed to test the validity of the reported values; the second compares their statistical power, asking how effectively they can help us reject the null when we should.

4.1 Analyzing the validity of the significance measures

We empirically studied the validity of the SG_{50} p-values and the reported E-values in multiple setups, where we varied the search engine, the query that was shuffled, the database composition, the substitution matrix and the gap penalties. For each such setup we generated a sample of 1 million draws from the null distribution by running the search tool with that many shuffles of the query and recording S, the maximal alignment score, as well as E, the minimal alignment E-value for each run. The following options were considered for each category.

- Search engine: NCBI BLAST (version 2.11.0), AB-BLAST (version 3.0, which we ran with the `-kap` option because its default sum statistic is not compatible with our p-value approach), FASTA, and SSEARCH (both version 36). The tools were run using their default settings except where explicitly stated otherwise.

- Query: the shuffled query was one of four different proteins that were selected at random from the UniProt database (UniProt ID: Q88D80, Q9NSN8, Q9D8T0, and P17654) subject to having lengths of 45, 90, 175, and 350 amino acids. Each query has a distinct length, so below we only refer to the length of the query.

- Database: the human-annotated SCOP database [15] (release 2022-06-29, consisting of 35,644 family-level representative domain sequences), the Swiss-Prot database [5] (release 2023_01, consisting of 481,450 manually annotated and non-redundant protein sequence), and the ASTRAL40 database [7] (version 2.08, a subset of the SCOP database containing 15,178 domain sequences, each with less than 40% identity to the others).

- Substitution matrices: BLOSUM45 (B45), BLOSUM62 (B62), and BLOSUM80 (B80) of [10], PAM70 of [8], and the non-standard PFASUM60 of [14].

- Gap penalties: largely the default for the chosen substitution matrix, e.g., BLAST's default for BLOSUM62 is (11,1): 11 to open a gap and 1 to extend it.

For example, the bottom row of Figure 2 is based on data generated by using BLAST to search 1e6 shuffles of a query of length 175 amino acids against the Swiss-Prot database [5], with PAM70 and its BLAST default gap penalties of (10,1). The top row was derived using a similar sample of 1e6 from the null, but this time BLAST was used to search the same shuffled queries against the SCOP database [15] with BLOSUM45 and its BLAST default gap penalties of (14,2).

We next used these null samples to examine the validity of the SG_{50} p-values and the E-values using 3-panel wide figures as explained below.

- We examined how well the null distribution of S fits the Gumbel distribution by looking at the frequency of p-values, computed using the Gumbel distribution with parameters estimated via maximum-likelihood estimation from the sample (left column of the 3-panel figures). Note that the closer the points are to the diagonal line the better the fit is (Figure S1A provides a reference point here).

- we computed a SG_{50} p-value for each of the 1m optimal alignment scores and then examined the frequency of the reported p-values (middle column panels): when the curve goes below the line, particularly for points to the right of ≈ -4.5, it indicates the p-values might not be valid, whereas points significantly above the line indicate a conservative bias.

- We examined the frequency of the reported minimal E-values. Because an E-value should be greater than its associated p-value, an E-value of say 0.01 or less should not appear in significantly more than 1% of the null samples. Thus, dips below the diagonal in the right column panels indicate a potential liberal bias: the reported E-values unduly inflate the significance of the alignments. A conservative bias is more difficult to quantify because the E-value is an overestimate of a p-value, however points substantially above the line indicate potential conservative bias which would translate to reduced power.

- Finally, for the canonical significance cutoffs of $\alpha \in \{0.05, 0.01, 0.001, 0.0001\}$ we looked at the frequency with which the SG_{50} p-values / E-values were $\leq \alpha$. Specifically, if that frequency was $> \alpha$, then we conducted a binomial test to see if this liberal tendency is statistically significant.

4.2 Comparing the power of the E-values and SG_{50} p-values

To compare the statistical power of the significance measures we performed homology search experiments based on the ASTRAL40 database [7] (version 2.08). ASTRAL40 is widely recommended as the gold standard for evaluating homology search performance [6, 18].

The proteins in the ASTRAL40 database are hierarchically organized into classes, folds, superfamilies, and families. To assess the sensitivity of the homology search in this experiment we made the common assumption that two proteins from the same superfamily are homologous (positive).

We performed a homology search for each protein within the ASTRAL40 database against itself, employing the same versions of BLAST, AB-BLAST, FASTA and SSEARCH noted in Section 4.1. Where possible, each search engine was applied with three different substitution matrices: BLOSUM62 (11,1), PAM70 (10,1), and the non-standard PFASUM60 (15,1). The gap penalties for BLOSUM62 and PAM70 are BLAST's default ones, and the penalty for PFASUM60 was recommended by its authors [14].

Power was compared in terms of the number of positive alignments that were reported at the cutoffs of 0.01 and 0.05. Based on our validation experiment reported below, we did not include AB-BLAST's E-values in this analysis because they are clearly invalid (Figure S7). Similarly, BLAST was not applied with PFASUM60 because it is a non-standard matrix.

5 Results

5.1 The E-values can be too liberal while the SG_{50} p-values appear to be valid

Our validation experiments, and specifically the right columns of our 3-panel figures, show that all the different flavors of E-values are often either too conservative or too liberal. Starting with BLAST, we see in the top right column of Figure 2 that BLAST can substantially inflate the significance of the alignments: 11% of the alignments have an E-value ≤ 0.05 (a p-value of 0 according to a 1-sided binomial test), and 2.7% have an E-value ≤ 0.01 (a p-value of 0). The bottom right panel of that figure shows the flip side where only 1.3% of the alignments have an E-value ≤ 0.5. While the magnitude of the significance inflation in Figure 2 is rather extreme, Supplementary figures S3, S4, S5, S6, and S8 all demonstrates that the problems that Figure 2 demonstrate are fairly common.

Moving on to AB-BLAST we were surprised to see that its reported E-values are clearly invalid. Specifically, Figure S7 shows that for a query length of 45 the reported E-values are too liberal for moderately small E-values (e.g., 6.1% of the alignments have an E-value ≤ 0.05; p-value of 0), but they become very conservative for smaller E-values (e.g., none of the 1m alignments has an E-value ≤ 0.0001). The situation is even worse for a query of length 90 where, for example, 35% of the alignments have an E-value ≤ 0.05 (p-value of 0). Note that we found similar problems when using AB-BLAST's default sum statistic. For this reason we chose to exclude AB-BLAST's E-values when comparing the power of the significance measures.

FASTA and SSEARCH rely on the same method when computing their E-values. While generally they tend to be overly conservative, we found examples of both tools inflating the significance. Indeed, the right column of Figure S9 shows that SSEARCH is very conservative for a query of length 45 (e.g., only 0.01% of the alignments have an E-value ≤ 0.001), while for a query of length 350 it is too liberal (e.g., 6.5% of the alignment have an E-value ≤ 0.05; p-value of 0). A similar trend, though less pronounced, can be observed for FASTA in Figure S10, where for example for a query of length 350, 6% of the alignments have an E-value ≤ 0.05 (p-value of 0).

At the same time, the SG_{50} p-values are much more consistent than the E-values in the same experiments. For shorter query lengths those p-values are somewhat conservative, but still typically significantly less so than the E-values. For the longer queries, together with the improved fit between the Gumbel and the null distribution of the maximal alignment score, the SG_{50} p-values improve considerably, while still remaining valid. On the latter point we note that, in contrast with the E-values, the fraction of p-values that were $\leq \alpha$ rarely exceeded α (for $\alpha \in \{0.05, 0.01, 0.001, 0.0001\}$), and none of those cases where it did was determined to be significant at the 0.05 cutoff of the same 1-sided binomial test that was applied in the case of the E-values.

5.2 The SG_{50} p-values discover more homologous sequences where the E-values are not too liberal

The SG_{50} p-values rely on an auxiliary null sample, which could have theoretically compromised their statistical power. Thus, it is particularly reassuring to see that in our power comparisons they typically detect more homologous sequences than the E-values do. Specifically, examining Table 1 we find that the number of reported homologs using the p-value cutoff is typically larger than the corresponding number reported using the E-value cutoff. For example, using SSEARCH and PFASUM60 (15,1) we report 145,288 homologs using the E-value cutoff of 0.01 compared with 149,736 using the E-value cutoff of 0.01 (a 3% increase.) The one exception is when using BLAST with BLOSUM62 (11,1), where it reports more homologs using the E-value cutoff criterion (140,488 vs. 136,921 or 2.6% more). However, looking at Figure S8 it is also clear that BLAST inflates the significance of some of its alignments in this setup (e.g., for a query length of 175, 6.1% of the null alignments have an E-value ≤ 0.05; a p-value of 0), so any advantage the E-value has should be taken with a grain of salt.

Regardless, for each combination of threshold and a substitution matrix, our p-values deliver the largest number of discoveries among all combinations of search engines and significance measures. In particular, SSEARCH with PFASUM60 (15,1) and the SG_{50} p-value cutoffs deliver the largest number of discoveries. Oddly, when using PAM70 (10,1) SSEARCH performed worse than BLAST using the E-value, as well as the SG p-value cutoff criteria.

Search tool	Substitution matrix	The number of reported homologs			
	(Gap open/extension penalty)	E-value <0.01	SG p-value <0.01	E-value <0.05	SG p-value <0.05
AB-BLAST	BLOSUM62 (Q11R1)	N/A	138118	N/A	151914
BLAST	BLOSUM62 (Q11R1)	140488	136921	149846	147360
FASTA	BLOSUM62 (Q11R1)	128174	133185	139004	144120
SSEARCH	BLOSUM62 (Q11R1)	140448	142915	151321	154450
AB-BLAST	PAM70 (Q10R1)	N/A	95183	N/A	104378
BLAST	PAM70 (Q10R1)	113522	116094	119980	123767
FASTA	PAM70 (Q10R1)	88706	91648	97284	100848
SSEARCH	PAM70 (Q10R1)	93949	95242	102191	104163
AB-BLAST	PFASUM60 (Q15R1)	N/A	146834	N/A	159705
FASTA	PFASUM60 (Q15R1)	133351	137729	144578	148840
SSEARCH	PFASUM60 (Q15R1)	145288	149736	157231	162083

Table 1: **The number of ASTRAL40 homologous sequences reported at the given threshold**. See Section 4.2 for details.

6 Discussion

E-values have allowed BLAST to present scientists with a meaningful statistical evaluation of reported local alignments. In this paper we exposed some deficiencies in how the E-value are computed in practice. Specifically, we found that at times they are under-estimated, inflating the significance of the reported alignment, while at others they seemed to be over-estimated. We showed these problems are shared with other commonly used similarity tools that rely on E-values.

Aside from the issue of whether or not the E-values are computed accurately enough, we question whether this is the right notion of significance we should be using in this context. One of the arguments that NCBI makes in favor of using the E-values is that "it is easier to understand the difference between, for example, E-value of 5 and 10 than P-values of 0.993 and 0.99995" `https://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html`. However, this begs the question of whether one should be interested in alignments that score so low that a random alignment has a probability of 0.993 to score higher.

Moreover, suppose you found 40 alignments with an E-value of 10. In this case you would be tempted to think this is somehow significant, because you expected only 10 such alignments but you found 40. However, assigning significance to this 40 vs. 10 is a different question and might be simply a reflection of the redundancy in the database.

Instead, we promote here the use of the canonical approach of FWER control based on p-values that are adjusted to the multiple hypothesis problem at hand. Our p-values are empirically valid and are exact in the limiting Gumbel distribution case, while being more conservative with shorter query lengths. Still, we show that they deliver more power than when using E-values when the latter are nor overly liberal.

Our approach is flexible in terms of the substitution matrix and gap penalties it allows, although it is clear that unreasonable penalties or matrices could break the underlying Gumbel approximation. Also, we remind the user that our approach comes with a very hefty computational penalty.

Exploring the applicability of our approach to `blastn` and nucleotide databases is reserved for future work.

References

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[3] Stephen F Altschul and Warren Gish. [27] local alignment statistics. In *Methods in enzymology*, volume 266, pages 460–480. Elsevier, 1996.

[4] Richard Arratia and Michael S Waterman. A phase transition for the score in matching random sequences allowing deletions. *The Annals of Applied Probability*, pages 200–225, 1994.

[5] A Bairoch and R Apweiler. The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic Acids Research*, 28:45–8, 2000.

[6] S. E. Brenner, C. Chothia, and T. J. P. Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences*, 95(11):6073–6078, 1998.

[7] S. E. Brenner, P. Koehl, and M. Levitt. The ASTRAL compendium for sequence and structure analysis. *Nucleic Acids Research*, 28:254–256, 2000.

[8] M. Dayhoff, R. Schwartz, and B. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5:345–352, 1978.

[9] Amir Dembo, Samuel Karlin, and Ofer Zeitouni. Limit distribution of maximal non-aligned two-sequence segmental score. *The Annals of Probability*, pages 2022–2039, 1994.

[10] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89:10915–10919, 1992.

[11] M. Hess, F. Keul, M. Goesele, and K. Hamacher. Addressing inaccuracies in BLOSUM computation improves homology search performance. *BMC Bioinformatics*, 17:1–10, 2016.

[12] S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences of the United States of America*, 87:2264–2268, 1990.

[13] Samuel Karlin, Amir Dembo, and Tsutomu Kawabata. Statistical composition of high-scoring segments from molecular sequences. *The Annals of Statistics*, pages 571–581, 1990.

[14] F. Keul, M. Hess, M. Goesele, and K. Hamacher. PFASUM: a substitution matrix from Pfam structural alignments. *BMC Bioinformatics*, 18:1–14, 2017.

[15] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

[16] W. R. Pearson. Effective protein sequence comparison. *Methods in Enzymology*, 266:227–258, 1996.

[17] William R Pearson and David J Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988.

[18] M. P. Styczynski, K. L. Jensen, I. Rigoutsos, and G. Stephanopoulos. BLOSUM62 miscalculations improve search performance. *Nature Biotechnology*, 26(3):274–275, 2008.