Marco Pagni
is staff member at the Swiss Institute of Bioinformatics. His research interests include software development and handling of databases of protein domains.

C. Victor Jongeneel
is Director of the Office of Information Technology of the Ludwig Institute and of the Swiss Institute of Bioinformatics.

Making sense of score statistics for sequence alignments

Marco Pagni and C. Victor Jongeneel

Date received (in revised form): 9th November 2000

Abstract

The search for similarity between two biological sequences lies at the core of many applications in bioinformatics. This paper aims to highlight a few of the principles that should be kept in mind when evaluating the statistical significance of alignments between sequences. The extreme value distribution is first introduced, which in most cases describes the distribution of alignment scores between a query and a database. The effects of the similarity matrix and gap penalty values on the score distribution are then examined, and it is shown that the alignment statistics can undergo an abrupt phase transition. A few types of random sequence databases used in the estimation of statistical significance are presented, and the statistics employed by the BLAST, FASTA and PRSS programs are compared. Finally the different strategies used to assess the statistical significance of the matches produced by profiles and hidden Markov models are presented.

Keywords: *statistical model, E-value, phase transition, pairwise alignment, profile, hidden Markov model*

INTRODUCTION

The assessment of the similarity between two sequences (or character strings) is one of the corner-stones of bioinformatics, as it is often indicative of true homology, and can thus be used to assign a structure or a biological function. The topic has received much attention, and many algorithms and software recipes have been and are still being developed. Its major application is probably to search a database using a sequence as the query. The use of profiles or hidden Markov models as queries is also becoming increasingly popular. All these similarity-search tools produce lists of matched sequences that are aligned with the query, either locally or semi-globally. Every match is assigned a numerical *score*, whose value is determined by a *similarity matrix* and by *gap opening and extension penalties* or by a *position-specific scoring matrix* (profile), and only matches that give a score greater than some threshold are reported. Note that the precise methods (algorithms) used to compute an alignment producing a maximal score are outside the scope of this paper.

The matched sequences reported by search programs can be classified as *true positives* and *false positives* (the sequences missed by the program are the *negatives*). A true positive is a sequence that shares similarity with the query because both have evolved (diverged) from a common ancestral sequence, and is thus a true *homologue*. (Similarity can also be sometimes attributed to evolutive convergence.) A sequence is regarded as a false positive if the observed similarity is attributable to *chance*. It must be stressed that only biological arguments can let one decide whether a sequence should be regarded as a true or false positive. Nevertheless, a statistical analysis based on sound principles can help in the decision, because some matches are more likely to have been produced by chance than others.

Today, the most frequently used statistical estimator is the *E*-value. It is the number of matches with a score equal to or greater than a given score that are expected to occur by chance. In other words, the *E*-value provides an estimation of the number of expected false positives

Dr Victor Jongeneel,
Director,
Ludwig Institute of Cancer Research, Chemin des Boveresses, 155, Epalinges CH-1066, Switzerland

Tel: +41 21 692 5994
Fax: +41 21 692 5945
E-mail: VictorJongeneel@isrec.unil.ch

above a given threshold. The *E*-value depends on the size of the searched database, as the number of false positives above a fixed threshold must increase proportionally to the *size of the database*. The total number of sequences and the total number of residues are the most frequently used figures to determine this size.

Similarity and homology

Bench biologists performing searches against sequence databases are usually basing their decision to accept a particular similarity as indicating homology on the basis of the *E*-value returned by the search algorithm. The central problem we want to address here is how *E*-values are computed, and more importantly for biologists under which conditions they can be trusted. There is more than one way to calculate an *E*-value, and different methods can produce values that differ by several orders of magnitude. Understanding the origins of such discrepancies is crucial when dealing with applications in which a computer and not a biologist makes the decision. A trivial example is the determination of the lower score threshold used when reporting matches after a similarity search in a database. Other applications requiring better accuracy are iterative search methods such as profile building or PSI-BLAST. In these methods a key to success is that the threshold value is determined (automatically) so that a maximum number of true positives is recorded but essentially all false positives are rejected.

***E*-values**

the *extreme value distribution*

Finally, this paper also aims to give a critical perspective on the conditions under which *E*-values can be trusted. Indeed, the procedures for the annotation of the exponentially growing number of DNA sequences are more and more automated. The quality of these annotations is a matter of concern and relies heavily on the (automated) decision to produce a particular annotation for a particular sequence. Most often, such decisions rely on some 'test', which ideally should be an *E*-value based on a realistic random sequence model.

THE EXTREME VALUE DISTRIBUTION

The scores produced by similarity-search algorithms usually do not follow a normal (Gaussian) distribution. The *statistics of extremes* worked out by Gumbel[1] provide a more appropriate theoretical framework to describe them. The *extreme value distribution* (EVD), which is also known as the Gumbel distribution, plays a key role in the theory, and thus will be presented here in some detail. Olsen *et al.*[2] introduce the EVD as follows:

> 'Given a set of independent and identically distributed random variables $x_1, x_2, \ldots,$ with a distribution which decays reasonably fast for large values of the x_i's, e.g., $P\{x_i > x\} \propto exp(-\alpha x^{\gamma})$ for $x \to \infty$ and $\alpha, \gamma > 0$, the distribution of the random variables
>
> $$X_n \equiv \max\{x_1, \ldots, x_n\}$$
>
> for large n is known to obey the Gumbel distribution
>
> $$P\{X_n > x\} = 1 - \exp(-\kappa e^{-\lambda x}). \quad (1)$$
>
> The distribution is universal, in that its shape does not depend on the specifics of the distribution of the x_i's: It is, first of all, completely independent of the details of the distribution of the x_i's at small values. Moreover, the shape of (1) does not depend on the values of the parameters such as α, γ and n. The latter only enter (1) through the values of λ and κ, the only parameters of the Gumbel distribution.'

An alternative formulation of (1) is often employed and makes use of the parameter $\mu = \ln \kappa/\lambda$

$$P\{S > x\} = 1 - \exp(-e^{-\lambda(x-\mu)}) \quad (2)$$

The associated probability density function $p(x)$, defined such as $P\{S > x\} = \int_x^{\infty} p(u)\,du$, is

$$p(x) = \lambda \exp(-\lambda(x - \mu) - e^{-\lambda(x-\mu)}) \quad (3)$$

The major feature of this density function is that it is skewed, ie it is not

symmetrical, and its left tail is 'steeper' than its right tail (Figure 1). The parameters μ and λ are related to the location of the maximum of $p(x)$ and to the 'narrowness' of the peak, respectively. They should not be confused with the mean and the standard deviation of the distribution.

Z-scores

As can be observed in Figure 1, the right tail of the EVD is decreasing almost linearly when plotted in a semi-logarithmic coordinate system. Therefore the right tail of the EVD is satisfactorily approximated with a decreasing exponential

$$p(x) \simeq P\{S > x\} \simeq e^{-\lambda(x-\mu)};\ \mu \ll x \quad (4)$$

random sequences

for sufficiently large scores. This is a particularly useful result because it can be applied to the situation that is encountered after a similarity search in a database: one is only interested in the very small fraction of top scoring matches that are the least likely to have been produced by chance. In practical applications, approximation (4) is almost always used for the computation of *E*-values, instead of one of the preceding exact relationships. (This also caused confusion in the notation used by some authors.)

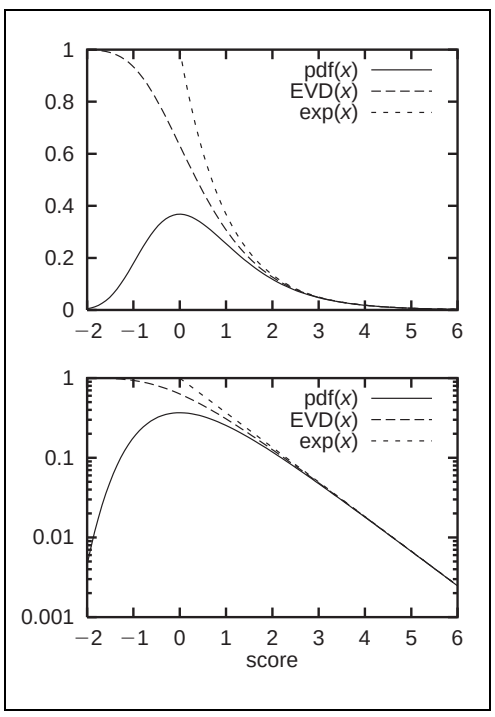

Figure 1: The extreme value or Gumbel's distribution. The integral representation of the distribution EVD(x) (2), its associated probability density function pdf(x) as defined by (3) and the exponential approximation of the right tail of the distribution exp(x) (4), computed with $\mu = 0$ and $\lambda = 1$ are represented on a linear scale (top panel) or a logarithmic one (bottom panel)

An older yet still commonly used 'statistical estimator' is the *Z-score*, which is the number of standard deviations that separates an observed score from the average score of all true negatives matches. Conversion of a *Z*-score into an *E*-value is possible if and only if the shape of the underlying distribution of the scores is known, as for example in eqn (12). Hence, a given *Z*-score can be regarded as quite improbable if the underlying distribution is Gaussian and quite probable if the distribution is an EVD, because the right tail of the latter is more 'extended' on the right than the one of the former (see Figure 2). Scores from the comparison of 10,000 pairs of random sequences of length 1,000 were obtained using the Smith–Waterman algorithm, a BLOSUM62 similarity matrix and two sets of gap penalties. The frequency distribution of the scores presented in the top graph was obtained with penalties open = 12/extend = 1; the full line is an EVD (3) and the dotted line is a normal (Gaussian) distribution fitted to the data. The bottom graph presents the results obtained by the same procedure when the gap opening penalty was lowered to 4. *Z*-scores have been replaced by *E*-values in most recent applications.

RANDOM SEQUENCES

Databases of 'random' sequences are commonly used to derive the distributions of alignment scores that can be attributed to chance rather than to homology. The statistician's favourite model of a *random sequence* is the concatenation of a fixed number of independently drawn letters from a given alphabet. The occurrences of the letters 'A', 'C', 'G', 'T' of the DNA alphabet are often taken as equiprobable. In the case of proteins, the frequencies of amino acids are usually chosen to be different, in order to mimic real biological sequences. For example, in the SWISS-PROT database, the most common amino acid is leucine

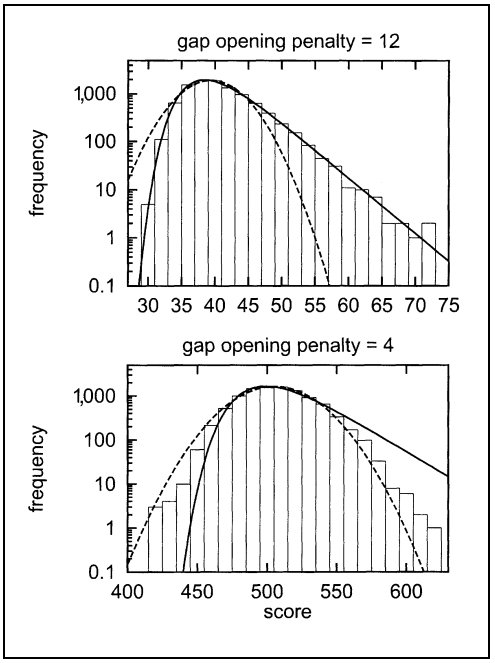

Figure 2: The EVD is an adequate model for the distribution of the maximal scores of local alignment of random sequences, in some cases (top graph) and not in others (bottom graph)

randomised databases

with a relative frequency of 9.3 per cent, followed by alanine (7.6 per cent) and serine (7.2 per cent). The rarest amino acids are tryptophan (1.3 per cent) and cysteine (1.7 per cent). Truly random sequences generated in this way are relatively tractable algebraically and allow for the analytical investigation of (some) aspects of the statistics of pairwise sequence alignment (see Karlin and Altschul[3] or Bundschuh and Hwa[4] for examples).

With the help of a pseudo-random number generator and a table of frequencies, a collection of random sequences is easily generated. Scores obtained by searching such a collection, provided it is sufficiently large, can be extrapolated to the properties of an ideal random sequence. This computational approach has the benefit of simplicity, but its experimental character should never be forgotten when interpreting the results, ie one should refrain from generalising results obtained with such methodology.

An alternative numerical procedure consists of searching a randomised database (see below) derived from real sequences to produce lists of matches that can be attributed to chance only. The relevant statistical parameters are then derived from the score distribution of these matches. There are several possible ways to *shuffle* sequences; some preserve the observed distribution of sequence lengths in the database, others preserve the composition of the individual sequences in the database. Although largely empirical, such strategies are quite efficient in practice because they produce data that are more realistic from a biological point of view.

A collection of randomised databases[5] is maintained by the Swiss Institute of Bioinformatics, all of which are derived from release 34 of the SWISS-PROT database (sprot34), containing 59,021 sequences and 21,299,624 residues. The most useful ones are the following:

- **scramble** preserves the distribution of the sequence lengths of sprot34, with randomly generated sequences of amino acids obeying the average amino acid frequencies of sprot34.

- **permutation** is obtained by random permutation of the amino acids in every sequence of sprot34. The length and composition of each individual sequence are preserved.

- **window20** is obtained by random permutation of the amino acids in adjacent windows with a width of 20 residues. The length and the composition of each individual sequence are preserved. Low-complexity regions and other local compositional biases are maintained.

- **reversed** is obtained by reversing the order of the amino acids in each sequence of sprot34. The length and the composition of each individual sequence are preserved. In addition to low-complexity regions, transmembrane segments and repetitive

regions such as coiled-coil domains are also maintained.

properties of randomised databases

The following example will illustrate how the use of randomised databases derived from actual protein sequences can mimic more accurately the properties of these sequences than can 'synthetic' random databases. Figure 3 presents the frequency distribution of cysteines in the proteins of release 34 of the SWISS-PROT database and of the **scramble** database (top graph). The cysteine content of individual proteins has an average value of 1.7 per cent, but the distribution is more uneven in real sequences than in the random sequences where residues were independently drawn (scramble). Moreover, the cysteine residues often appear clustered in real protein sequences: the bottom graph of Figure 3 presents the distribution of the maximal number of cysteines found in 20 adjacent amino acids. This distribution is bimodal for the real database, a property that is not preserved in **scramble**, but is preserved in **reversed** and in **window20**. Many of these cysteine-rich regions correspond to extracellular domains. Figure 4 presents an example of such a domain, the calcium-binding EGF (epidermal growth factor) domain. A database search using a sequence or profile representing the EGF domain as a query would thus generate more accurate *E*-values if it had been calibrated using the **reverse** or **window20** databases than if it assumed a random distribution of residues.

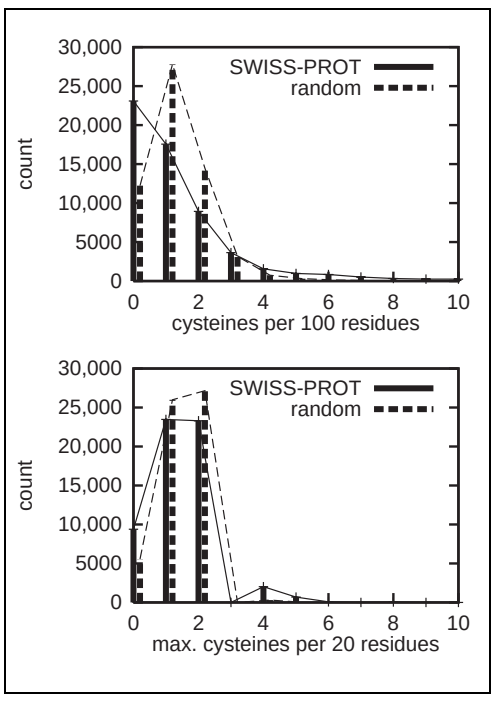

Figure 3: Distribution of cysteines in SWISS-PROT and in a random database

LOCAL ALIGNMENTS WITHOUT GAPS

ungapped alignments

Consider two sequences of length m and n made of letters from an alphabet of size Z, drawn with probabilities p_x; $x = 1..Z$; $\sum_{i=1}^{Z} p_i = 1$. Consider a substitution matrix $\{s_{i,j}\}$ that provides a score for aligning a letter i with a letter j and which is symmetric $s_{i,j} = s_{j,i}$. Provided that the expected score contributed by a pair of random residues is lower than zero (also assuming that some pairs of amino acids have positive contributions), ie

$$\sum_{i,j=1}^{Z} p_i p_j s_{i,j} < 0 \tag{5}$$

it has been demonstrated[3] that the distribution of the scores of gap-less pairwise alignments of two random sequences tends towards an EVD for sufficiently large sequence lengths m and n. It was also shown that the EVD parameter λ is given as the implicit solution of

$$\sum_{i,j=1}^{Z} p_i p_j \, \mathrm{e}^{s_{i,j}\lambda} = 1 \tag{6}$$

and μ is given by

$$\mu = \frac{\ln Kmn}{\lambda} \tag{7}$$

where K is a constant, with a value depending on $\{s_{i,j}\}$ and p_x.

If expression (5) does not hold, long alignments will be favoured simply because they are long, and will thus be biologically meaningless.

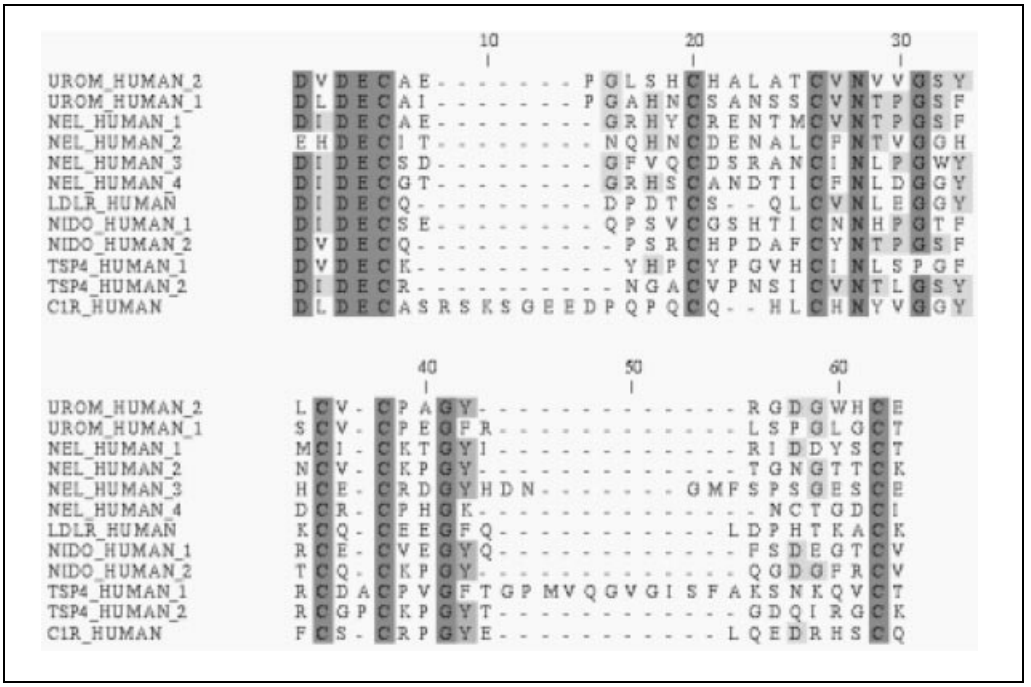

Figure 4: The calcium-binding EGF domain is made of cysteine-rich subsequences. A few subsequences matched by the Prosite profile EGF_CA_2 were aligned and the conserved residues are highlighted

size of the search space

The product $n \cdot m$ is known as the *size of the search space*. Hence, the above result can be restated: the EVD is applicable to gap-free alignments provided the search space is sufficiently large. This is the main reason why the original implementation of BLAST produced only gap-less alignments – there was a strong theoretical justification for the derivation of *E*-values. Unfortunately, there is no comparable analytical result available for the case which is actually the most useful to the biologist: alignments with gaps.

gapped alignments

LOCAL ALIGNMENTS WITH GAPS

When local alignments of random sequences containing gaps are considered, it is often assumed that the EVD is applicable to the distribution of the maximal scores. This is true under some circumstances but not others.[6,7] Figure 2 illustrates this point by providing two examples, one in which the EVD is a valid model and one in which the EVD is clearly inappropriate, because the scores are more likely to be normally distributed.

Is it possible to test the conditions under which the EVD can be applied 'safely' or, in other words, is there an expression analogous to (5) but for gapped alignments? There is still no definite answer to this question. However, the situation is not totally desperate and the theoretical understanding of the problem is currently progressing (see Bundschuh and Hwa[4] and Mott and Tribe[8] for recent advances). The next sections will present observations made from numerical simulations, instead of theoretical considerations.

Phase transitions

The expected score of a local alignment depends on the length of the two compared random sequences and, intuitively, one can guess that the longer the sequences, the greater the score expectation. Figure 5 presents the results of two numerical simulations performed to demonstrate how the average score $\bar{x}$ varies with the length of the random sequences (see the figure legend for details of the simulation). The curve obtained with an expensive gap opening penalty was fitted quite successfully with a logarithmic function

phase transitions

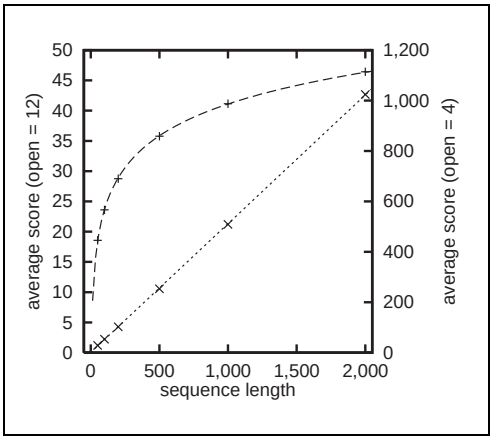

Figure 5: The average score of local alignments between random sequences increases with sequence length: this relationship can be modelled with a linear function in some cases and with a logarithmic function in others. Scores from the comparison of six collections of 10,000 pairs of random sequences of increasing lengths were obtained using the Smith–Waterman algorithm, a BLOSUM62 similarity matrix and two sets of gap penalties. The average score was determined for each set of scores and plotted as a function of sequence length. The data obtained with penalties open = 12/extend = 1 were fitted with a logarithmic model (8); the data obtained by the same procedure after lowering the gap penalty to 4 were fitted with a linear model (9)

EVD and logarithmic score/length relationship

$$\bar{x} = c_A \ln c_B l \tag{8}$$

where c_A, c_B are *ad hoc* adjustable parameters. The curve obtained with a cheap gap opening penalty was fitted, also quite successfully, with a linear function

$$\bar{x} = c_C l \tag{9}$$

with c_C another adjustable parameter. However, alignments that were produced in the latter case are certainly without value for a biologist: gaps are inserted so frequently that the resulting alignments lack any meaning. This is another situation where long alignments are favoured simply because they are long. The two examples of score distributions in Figure 2 actually correspond to two of the points of Figure 5, ie those for sequences of length 1,000. Accordingly, the EVD appears associated with the logarithmic curve, while the more symmetrical distribution of the scores is associated with the linear curve. Credit is due to Waterman *et al.*[9] to have first pointed out these features while studying local alignments of DNA sequences with a linear gap cost model. These authors also observed that this system was unlikely to be observed in an intermediate state. Indeed, the linear and logarithmic curves correspond to two distinct *phases* and the change from one to the other occurred abruptly as a function of the gap penalty. This was compared to various physical systems in which *phase transitions* have been described.

Phase transitions can be revealed as follows: consider the residual square sums $\mathrm{RSS}_{\mathrm{lin}}$ obtained from the least square fit of the linear model to the data, and $\mathrm{RSS}_{\mathrm{log}}$ the corresponding value for the logarithmic model. The phase θ in which the system is found can readily be estimated as the ratio

$$\theta = \frac{\mathrm{RSS}_{\mathrm{lin}}}{\mathrm{RSS}_{\mathrm{log}} + \mathrm{RSS}_{\mathrm{lin}}},\ 0 \leqslant \theta \leqslant 1 \tag{10}$$

which approaches zero in the linear phase and one in the logarithmic phase. The computational recipe used to produce the curves of Figure 5 was repeated for intermediate values of the gap opening penalty. Both phase models (8, 9) were fitted to each curve. The value of θ was determined using (10) and plotted against the gap opening penalty (Figure 6). In this way phase transitions can be clearly visualised: the system is in an intermediate phase for only a very restricted range of gap penalties.

Many programs used for local similarity searches take as parameters the similarity matrix and the gap opening and extension penalties. Among these programs, BLASTP (NCBI version 2.0.9) offers only a limited subset of allowed parameter combinations. For example considering the above example of a BLOSUM62 similarity matrix and a gap extension penalty of 1, BLASTP allows only *gap existence* penalties of 10, 11 and 12 which

BLASTP parameters

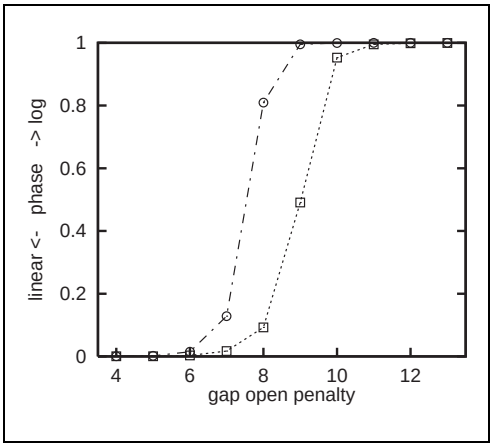

Figure 6: A *phase transition* from the linear phase to the logarithmic phase is observed when the gap opening penalty is progressively increased. Given a similarity matrix (BLOSUM62), a gap extension penalty (1) and different gap opening penalties and random sequences of various length, we produced curves similar to those presented in Figure 5). The *shape* θ of each curve was characterised by using eqn (10), and this factor was plotted as a function of the gap penalty used. The circles represent data points obtained using random sequences produced with the amino acid frequencies of SWISS-PROT; the squares represent data obtained with a database in which the frequency of proline was increased by 10 per cent

correspond to gap opening penalties of 11, 12 and 13 respectively. By examining Figure 6 one can expect that the local alignments obtained with these combinations of parameters are produced in the logarithmic phase.

One might wonder why a gap opening penalty lower than 11 is not allowed by the BLASTP program. The answer is probably that the random-sequence composition used for BLAST calibration might be significantly different from the composition faced by the biologist in a real application. A second curve is presented in Figure 6 to illustrate this point. It was obtained by modifying the amino acid frequencies of the random sequences: the frequency of proline was increased by 10 per cent while preserving the relative frequencies of the other amino acids. This apparently small alteration of amino acid frequencies was sufficient to shift the phase transition by more than one 'unit' of gap opening penalty. Hence BLASTP simply does not allow the user to perform similarity searches with parameters close to the phase transition limits, thus tolerating sequences whose compositions depart from the calibration model. (The masking of low-complexity regions is another 'trick' that is used to prevent the similarity search algorithm from producing spurious alignments.)

Mott and Tribe[8] have recently shown that the behaviour of gapped alignments is essentially governed by a single parameter, α, which depends on the scoring scheme and on the sequence composition. The value of α is a very useful predictor of the position of the transition point between logarithmic and linear behaviour.

effective search space

The effective search space

The *size of the search space*, ie the product of the two sequence lengths n and m, can be interpreted as the number of pairs of coordinates from which maximal scoring local alignments can start. However, because any alignment has a non-negligible length, a high scoring match is unlikely to start from the last positions of a sequence. Hence, the size of the search space defined by the product $n \cdot m$ is overestimated. Altschul and Gish[10] proposed to correct this size by introducing an additional value L, the typical length of an alignment, and to express the corrected size of the search space by

$$(n - L)\cdot(m - L) \qquad (11)$$

length of an alignment

One can expect L to increase logarithmically with the length of the sequences, just as the average score increases in the logarithmic phase (cf. eqn (8)). Hence the correction will have a greater effect on short sequences. This is probably why sufficiently long sequences are required to obtain an EVD in the analytical demonstration, while no such correction of the size of the search space was envisioned.

Of course, a major practical problem is to determine the value of L. A recipe is given in Altschul and Gish[10] which makes use of an additional parameter, the *relative entropy H (in nats)* or *the expected score, per aligned residues, of an optimal random subalignment*. A value for H can be obtained analytically for alignments without gaps. For gapped alignments, L can be derived numerically from a collection of empirically determined match lengths. (Note that the definition of match length is ambiguous in the Smith–Waterman algorithm because the maximal score can correspond to several alignments, which may well have different lengths. Hence, statistics derived from match lengths actually depend on the details of the implementation of the backtracking algorithm, which is an undesirable property.)

BLASTN statistics

BLAST statistics

The BLASTP and BLASTN programs are designed to be faster, by several orders of magnitude, than programs (such as SSEARCH) that implement the original Smith–Waterman algorithm. We will not deal here with the BLAST heuristics (see Altschul *et al.*[11,12]), but discuss topics that are relevant for the computation and statistical assessment of E-values. Because computational speed is crucial for these programs, the use of any CPU-intensive procedures for computing statistics is undesirable; only algebraic relationships and pre-tabulated numerical values are used. In practice BLASTP and BLASTN (NCBI version 2.0.9) currently make use of two different strategies to achieve a fast estimation of the E-values.

BLASTP statistics

BLASTN simply ignores gaps when computing E-values, ie the analytical solutions available to determine the values λ, K and L for alignments without gaps are applied to compute the E-value of gapped alignments. (The actual base compositions of the query and of the database are ignored in the calculation.) This is acceptable provided gaps are sufficiently rare. Indeed, the default behaviour of BLASTN is optimised for maximal speed, because many DNA databases are large, and matches that would involve many gaps are missed by the default heuristic anyway. The benefit of this simplification is that the values of λ, K and L can be calculated exactly and quickly (see eqns (6), (7)) for any combination of match/mismatch scores and gap penalties as long as relationship (5) holds. This, however, is not sufficient to ensure that the alignments are produced in the logarithmic phase (see above). It is quite easy to produce a BLASTN request that performs a similarity search whose results will be in the linear phase. For example, using *any* DNA sequence as query against *any* DNA database with match/mismatch scores of $10/-30$ and (relatively cheap) gap penalties of $-2/-1$ produces a long list of biologically and statistically meaningless alignments.

The parameter values used for the computation of the E-values are listed at the end of every BLASTN output (Figure 7). The way these are calculated is as follows: Given the raw score S obtained for an alignment and given λ, K and L, the raw score is first transformed into a *bit score* S_{bit} trough

$$S_{\text{bit}} = \frac{\lambda S - \ln_K}{\ln 2}$$

The E-value is derived using approximation (4):

$$E = \text{sss} \cdot 2^{-S_{\text{bit}}}$$

where sss is the size of the search space (see 11) defined here for the whole searched database

$$\text{sss} = (m - L)(n - N \cdot L)$$

where m is the length of the query in amino acids, n is the size of the database in amino acids, N is the number of sequences in the database and L the typical length of the maximal scoring alignment.

In contrast the BLASTP program takes gaps into account when computing E-values. Numerical values for λ and K were obtained for various combinations

```
Lambda K H
1.37 0.711 1.31 ← λ, K, H
Gapped
Lambda K H
1.37 0.711 1.31
Matrix: blastn matrix:1 -3
Gap Penalties: Existence: 5, Extension: 2
Number of Hits to DB: 3910993
Number of Sequences: 818659
Number of extensions: 3910993
Number of successful extensions: 287203
Number of sequences better than 10.0: 59
length of query: 1272 ← n
length of database: 4,977,035,752 ← m
effective HSP length: 22 ← L
effective length of query: 1250 ← n − L
effective length of database: 4,959,025,254 ← m − N · L
effective search space: 6198781567500 ← (n − L)(m − N · L)
effective search space used: 6198781567500
T: 0
A: 0
X1: 6 (11.9 bits)
X2: 0 ( 0.0 bits)
S1: 12 (24.3 bits)
S2: 20 (40.1 bits)
```

Figure 7: The last lines of a BLASTN output provide a listing of the parameters used to compute the *E*-values. The symbols used correspond to those defined in the text

FASTA statistics

of scoring matrices and gap penalties using the Smith–Waterman algorithm. These values are hard-coded in the BLAST program. The recipe that BLASTP uses to compute L is currently documented only in the source code and relies on sequence length and on the value of the gap-less K, ie L does not depend on the gap penalties. Every BLASTP output reports the parameter values used for the computation of the E-value; an annotated example is presented in Figure 8.

There are many other points that together make the BLAST programs so successful. To mention a few:

- The low-complexity filters DUST for nucleic acids and SEG for proteins are activated by default. It is usually not a good idea to turn them off, as low-complexity regions tend to produce lengthy meaningless alignments and, moreover, they interfere with the BLAST heuristic.
- The computation of the E-value outlined above only addresses the problem of the maximal scoring pairwise alignments. When multiple matches occur between two sequences, BLASTP and BLASTN report a single combined E-value. For details of the computation of this value see Altschul and Gish[10] and Anon.[13]
- One usually observes that the BLAST and Smith–Waterman algorithms produce identical scores and alignments, provided E-values are sufficiently small, no low-complexity region is involved, and the search parameters are identical.
- BLASTP is optimised for a few predefined combinations of parameters, as proposed on the NCBI BLAST web pages. It is generally good advice to respect these.

FASTA statistics

The FASTA heuristics[14] made it possible to search biological databases for gapped local alignments many years before this was possible with the BLAST programs. FASTA is a little slower than BLAST, and can produce only one local alignment per pair of compared sequences. On the other hand, FASTA accepts any combination of scoring matrix and gap penalties, and produces quite reliable E-values. The performance of FASTA and BLAST were compared in many papers, and several recent contributions on this topic also consider other algorithms such as PSI-BLAST.[15,16]

E-value computation by FASTA is based on the following principle (from Pearson[17]):

> 'Current protein and DNA sequence databases contain many tens of thousands of sequences, almost all of which are unrelated to an individual query sequence (. . .). Thus, every database search provides tens of thousands of scores from unrelated, effectively random, protein and DNA

```
Lambda K H
0.318 0.139 0.395 ← −K,H
Gapped
Lambda K H
0.270 0.0470 0.230 ← λ, K −
Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Details of the blast heuristics
Number of Hits to DB: 16939096
Number of Sequences: 83857 ← N
Number of extensions: 716750
Number of successful extensions: 1756
Number of sequences better than 10.0: 65
Number of HSP's better than 10.0 without gapping: 43
Number of HSP's successfully gapped in prelim test: 22
Number of HSP's that attempted gapping in prelim test: 1638
Number of HSP's gapped (non-prelim): 79
Effective search space calculation:
length of query: 292 ← n
length of database: 30,454,973 ← m
effective HSP length: 52 ← L
effective length of query: 240 ← n − L
effective length of database: 26,094,409 ← m − N · L
effective search space: 6262658160 ← (n − L)( m − N · L)
effective search space used: 6262658160
T: 11
A: 40
X1: 16 ( 7.3 bits)
X2: 38 (14.8 bits)
X3: 64 (24.9 bits)
S1: 41 (21.7 bits)
S2: 64 (29.3 bits)
```

Figure 8: The last lines of a BLASTP output provide a listing of the parameters used to compute the *E*-values. The symbols correspond to those defined in the main text. Note that two different values for *K* are used

sequences. For local similarity scores, these 'random' sequence scores are expected to follow the extreme-value distribution (. . .) with location and scale parameters that reflect the lengths and compositions of the query and library sequences and the scoring matrix and gap penalties used.'

The actual computation of *E*-values by FASTA is quite intricate; the following steps can be recognised in the process:

1. For any given query sequence, a large collection of scores is gathered during the search of the database. This is possible because of the details of the FASTA heuristic, which quickly produce an alignment score for any pair of compared sequences (this so-called *optimal* score can actually differ from the Smith–Waterman score). Note that the searched database must be sufficiently large and of a sufficiently heterogeneous composition to effectively produce a useful collection of scores.

2. All potentially significant scores must be removed from the whole collection. Several procedures were compared by Pearson[17] to prune outliers. In the current implementation (FASTA 2.0 and above), scores are assigned into bins of a histogram based on the length of the database sequence, and the trimming is performed within each bin. An average score and standard error is then computed for each bin.

3. The expected random score $\bar{x}$ as a function of the logarithm of the length of the sequences in the database $\ln l$ is described with a linear model

   $$\bar{x}(l) = a + b \ln l$$

   where the values of the parameters a and b are obtained through a weighted linear regression with the histogram data. This empirical model is highly reminiscent of (8). Steps 2 and 3 are repeated iteratively to produce a more reliable estimation for the parameters.

4. Each potentially significant score is converted into a *length-corrected Z-score*

   $$Z(x,\ l) = \frac{x - \bar{x}(l)}{\text{sdev}}$$

   where sdev is the average of the standard deviation of the random scores.

5. The Z-score is eventually converted into an E-value under the assumption that the distribution of the random scores is an EVD, ie

$$E = N \cdot [1 - \exp(-e^{-1.282 \cdot Z - 0.5772})] \quad (12)$$

where N is the number of sequences in the database.

PRSS statistics

PRSS vs BLASTP

Given two protein sequences, a similarity matrix, and a combination of gap-opening and gap-extension penalties, the program PRSS[18] computes the score of the maximal scoring alignment and an *E*-value using the Smith–Waterman algorithm. The *E*-value is derived from a collection of maximal scores obtained from alignments between one of the sequences and random permutations of the other. The parameters of an extreme value distribution are derived from these scores and used to estimate the probability of the observed real score. The main benefit of this approach is obvious: the statistics are based on amino acid frequencies that are taken from the sequences being actually compared, which might significantly depart from the composition of a 'standard' database.

PRSS can also use more sophisticated shuffling methods for the generation of random sequences: for example, residues can be randomly permuted with in a local window of 20 amino acids' width. This preserves the regional variation of the amino acid composition along the sequences. The following example, which features the use of this option, demonstrates that PRSS may be better at detecting significant matches than BLASTP.

The protein P75020 of TrEMBL is annotated as being similar to a chitinase of *Bacillus circulans*. This is almost certainly wrong, as the chitinase active site is absent from this sequence. On the other hand, two well-known protein domains are present, a discoidin domain[19] (which resembles the coagulation factor 5/8 type C domain), and a fibronectin type III domain.[20] The latter is also present in the *B. circulans* chitinase, and this caused the erroneous annotation. The SWISS-PROT database was searched with P75020 as the query, using BLASTP with a BLOSUM45 substitution matrix and gap penalties of −14/−2. These settings are suitable for the detection of remote homologies, but they are not *recommended* according to the NCBI web pages. One can suspect that the alignment algorithm operates pretty close to the phase transition with these settings. The output of the BLASTP search was analysed in detail. The default score threshold was used and only the maximal score for each pair of sequences was kept (multiple matches were ignored!). The 120 matched protein sequences were classified in one of the following three categories:

- **Low complexity**: the protein is present in the BLASTP list of matches when the SEG filter is turned off, and is absent when the filter is turned on; ie the database sequence matches a low-complexity region of the query.
- **True positive**: is arbitrarily defined on the basis of the presence of a discoidin or fibronectin type III domain, as predicted by the Prosite profile DS_DOMAIN or by the Pfam HMM FN3, respectively.[21–23]
- **False positive**: the protein does not fall into one of the above categories.

Figure 9, top panel, presents the *E*-values of the 120 best matches computed by BLASTP, with the SEG filter turned off. The bottom panel shows the *E*-values produced by PRSS (500 shufflings with a window width of 20 amino acids) for the same matches and in the same ranking. Considering the repetition of the *E*-value into the three categories defined above, which are represented by distinct symbols in Figure 9, it is clear that PRSS performs much better than BLASTP in separating the true positives from the false positives. Moreover, while sequences with low-complexity regions appear to pollute the BLASTP output list, most of them are (correctly) placed with the false positives by PRSS. Only two proteins with a low

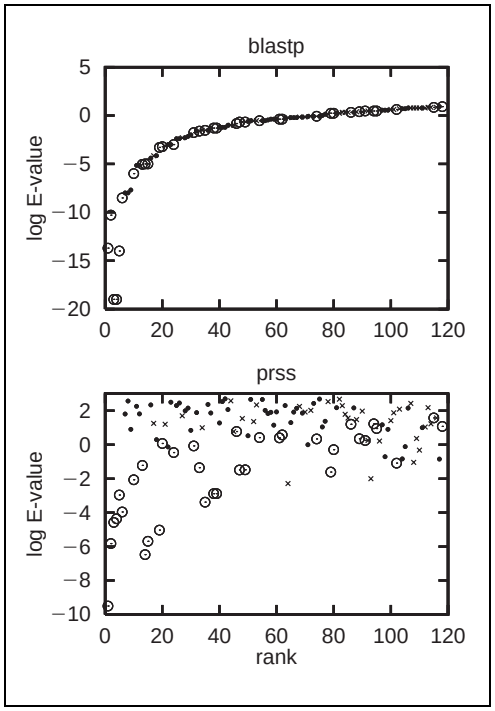

Figure 9: The *E*-values of the matches produced by a BLASTP search can be re-evaluated using PRSS. This significantly improves the classification of the matches into true or false positives. The SWISS-PROT database was searched with TrEMBL entry P75020 as a query using a BLOSUM45 similarity matrix and gap penalties of −14/−2. The 120 best matches were classified into three categories: *low complexity*, where the sequence is only reported when the SEG filter is turned off (cross); *true positive*, where the sequence is matched by either the Prosite profile DS_DOMAIN or the Pfam HMM FN3 (open circle); *false positive*, where the sequence does not fit in either of the previous categories (small closed circle). The top graph presents the *E*-values as computed by the BLASTP program and plotted against the rank at which the corresponding match appear in the BLASTP listing; the bottom graph presents the *E*-values computed by PRSS using 500 shufflings and a window width of 20 amino acids, in the same order

hidden Markov models

HMM and profile calibration

complexity received a relatively low *E*-value; on closer examination, both of these proteins proved to be true positives.

Note, however, that the generation of shuffled sequences followed by Smith–Waterman alignments (the strategy used by PRSS) is extremely expensive in terms of CPU usage. Therefore, a standard strategy for database searching involves the collection of potentially significant matches with BLAST, followed by an examination of individual pairs using PRSS.

SEMI-GLOBAL ALIGNMENTS

Pfam hidden Markov models (HMMs) and Prosite profiles are two examples of *descriptors* that are used to predict domains in protein sequences. To search for the occurrence of a domain one can use, for example, the programs HMMSEARCH of Sean Eddy's HMMER 2.0 package for Pfam HMMs or the program PFSEARCH of Philipp Bucher's PFTOOLS package for Prosite profiles. Similarity searches with these tools differ in many respects from the previously considered local similarity searches. A first difference is that the scoring system is position-dependent, ie the score rewards or penalties for a given amino acids vary along the descriptor. Another difference is that one usually searches for matches that are local on the sequences but global on the descriptor. Protein domains are often viewed as protein building blocks, or structural units, and as such should be matched globally. This alignment method is sometimes called *semi-global* and there is no theoretical framework to account for the statistics of the score distributions under these conditions. It is assumed that the distribution of the scores is skewed and resembles an EVD, but when more detailed analyses are performed, it appears that the EVD model may not always be appropriate.

The good news with semi-global alignment methods is that it is possible to estimate in advance the composition of a subsequence that should produce a high scoring match: it must resemble the consensus sequence of the predictor. Therefore, even if this consensus sequence contains low-complexity regions, or if its amino acid composition is highly biased, statistics can be adapted for

equivalence of HMM and profiles

each particular case. In other words, HMMs and profiles can be *calibrated* and the relevant parameters can be incorporated as part of the descriptor (see details below and Figures 10 and 11).

The HTOP program, part of PFTOOLS, converts a Pfam HMM into a Prosite-formatted generalised profile. The conversion is fully conservative in the sense that the raw score produced by the HMM (provided the HMMSEARCH −null2 option is set) for a given alignment is identical to the score produced by the converted profile on the same alignment.[24] Hence, Pfam HMMs and generalised profiles are interconvertible objects as far as the raw score generation procedure is concerned. Nevertheless, the *E*-values computed by HMMSEARCH and PFSEARCH may sometimes be quite different.

HMMER 2.0 statistics and the Pfam database

The program HMMCALIBRATE (HMMER 2.0 package) generates a random database similar to scramble, in which amino acids have the frequencies found in SWISS-PROT and the sequence lengths are distributed with the same mean and standard deviation as observed in SWISS-PROT. Then, each of these random sequences is aligned with the HMM to calibrate and the highest score per sequence is collected. An EVD is fitted to the score frequencies, and the values of the two parameters λ and μ are obtained using a *maximum likelihood fitting* procedure (details are available in Eddy[25]). It should be noted that HMMCALIBRATE censors the data set by retaining only those that exceed some threshold. In other words, only the right tail of the EVD is fitted.

The statistical estimator used is the *E*-value $E(x, N)$ which is defined by

$$E(x, N) = N \cdot Pr\{S > x\} \quad (13)$$

where N is the number of sequences in the searched database.

The composition of the random sequences used for calibration can be quite different from the composition of the subsequences matched by a particular HMM. For this reason HMMSEARCH rescores each alignment during a post-processing step that takes into account possible biased composition in either the HMM or the target sequence (the *null2* option). The score that is eventually reported may thus be significantly different from the score that would have been produced without this post-processing step, as for example during the calibration of the HMM.

The Pfam database[21,22] is an annotated collection of HMMs. The Pfam collection is currently designed for the automated annotation of genome data. In that perspective, it is essential that all false positives are rejected, even at the cost of rejecting a few true positives. Several cutoffs are declared in the header of Pfam HMMs as can be seen in Figure 10. The numerical values of these cutoffs were 'manually' set by an expert, based on the inspection of the list of the matched proteins. This ensures that false positives

```
HMMER2.0
NAME  EGF
ACC   PF00008
DESC  EGF-like domain
LENG  45
ALPH  Amino
RF    no
CS    no
MAP   yes
COM   hmmbuild -F HMM.ann SEED.ann
COM   hmmcalibrate --seed 0 HMM.ann NSEQ 87
DATE  Sat Dec 18 01:42:35 1999
CKSUM 5531
GA    24.7 10.6 ← "gather" cutoffs
TC    24.7 10.6 ← "trusted" cutoffs
NC    24.6 39.8 ← "noise" cutoffs
XT    -8455 -4 -1000 -1000 -8455 -4 -8455 -4
NULT  -4 -8455
NULE   595 -1558 85 338 -294 453 -1158 197 249 902 -1085 \
       -142 -21 -313 45 531 201 384 -1998 -644
EVD    -37.898247 0.207810 ← μ, λ
...
```

Figure 10: Header of the Pfam hidden Markov model EGF with the EVD parameters on the last visible line. Note also the three lines that provide the cutoffs on a per protein (first number) or on a per match (second number) basis

```
ID   EGF_CA_2; MATRIX. (preliminary)
AC   PS50034; DT ? (CREATED).
DE   Calcium binding EGF domain.
MA   /GENERAL_SPEC: ALPHABET='ABCDEFGHIKLMNPQRSTVWYZ'; LENGTH=40;
MA   /DISJOINT: DEFINITION=PROTECT; N1=5; N2=36;
MA   /NORMALIZATION: MODE=1; FUNCTION=LINEAR; \
     R1=.9563; R2=.00781033; ← R1 = (ln A/n − λμ)/ln 10 and R2 = λ/ln 10.
     TEXT='NScore';
MA   /CUT_OFF: LEVEL=0; SCORE=965; N_SCORE=8.5; MODE=1;
MA   /CUT_OFF: LEVEL=-1; SCORE=709; N_SCORE=6.5; MODE=1;
...
```

Figure 11: Header of the Prosite profile EGF_CA_2. The EVD parameters are 'hidden' in the parameters R_1 and R_2 (A/n is the average length of a sequence in the searched database)

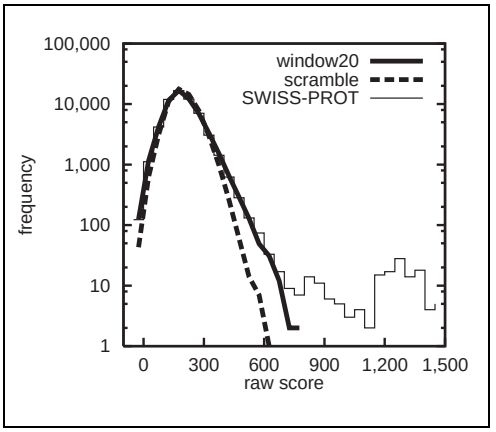

Figure 12: Calibration of a generalised profile. The SWISS-PROT database was searched with the Prosite profile EGF_CA_2 and the maximal scoring match was kept for each protein. The search was also performed on two randomised databases: scramble with the same distribution of protein lengths as SWISS-PROT, but with sequences made of independently drawn amino acids, and window20 obtained by random permutation in windows of 20 amino acids of all the SWISS-PROT proteins. This particular profile is actually calibrated using window20: a match with a score of 965 will receive an *E*-value of 1 if the searched database is made of $10^{8.5}$ residues (about ten times SWISS-PROT)

Pfam database

Prosite profiles

are efficiently rejected, but at the expense of the significance of the *E*-value. Therefore, the *E*-value produced by Pfam HMMs should not be trusted too much when remote homologies are searched for, as for example when searching for new genes containing a particular domain. The empirical calibration procedure described below is probably better suited for this task.

Generalised profile statistics

The calibration of Prosite profiles is empirical: a collection of scores is first obtained by searching a database of shuffled real sequences, most often window20 or reversed (see example in Figure 12). The best scoring matches, encompassing about 10 per cent of the total number of searched proteins, are fitted with a decreasing exponential, which is equivalent to the right tail approximation of the EVD (4). The size of the search space is defined by the number A of residues in the searched database. The *E*-value associated with a score, ie the number of matches with scores exceeding x in a database of size A residues, is given by

$$E(x,\ A) = A \cdot 10^{-R_1 - R_2 x} \qquad (14)$$

where R_1, R_2 are the two parameters that characterise the right tail of the distribution.

This definition of an *E*-value is distinct from the one used by Pfam (13) essentially because of the different definition of the size of the search space. In practical applications, these two *E*-values can be considered equivalent

$$E(x,\ N) \cong E(x,\ A)$$

and to be valid, their parameters for any x must satisfy

$$R_1 = \frac{\ln \dfrac{A}{N} - \lambda\mu}{\ln 10}$$

and

$$R_2 = \frac{\lambda}{\ln 10}$$

which reveals the relationships between the EVD characteristic parameters λ and μ and the Prosite profiles empirical parameters R_1 and R_2. Note that the ratio A/N is the average protein length in the

searched database. The Prosite profile definition of the size of the search space has one advantage, as it can be applied to DNA sequence profiles, where sequence length is a biologically meaningless parameter.

Prosite *E*-values

E-values do not appear explicitly in the output of PFSEARCH, but are replaced by so-called *normalised scores*. These result from an affine transformation of the raw score, namely $R_1 + R_2x$, and are comparable in spirit to the 'bit score' produced by the BLAST programs. The normalised score is independent of the size of the searched database and is related to the *E*-value by (14). For example, a normalised score of 9.0 correspond to an *E*-value of 1 if the searched database contains 10^9 residues or it corresponds to an *E*-value of 0.01 if the searched database contains 10^7 residues.

CONCLUSIONS

Database search algorithms are always walking a fine line between the desire to maximise sensitivity (ie the probability of detecting distant homologues) and the danger of generating meaningless alignments. It is clear that *E*-values produced under conditions where scores increase linearly with alignment size have no statistical value whatsoever. It is also clear that it is not too difficult to produce conditions where this occurs.

The programs currently used to search the (large) biological databases are under severe constraints of CPU cost. Because of these constraints, *E*-values are not calculated as efficiently as it may be possible using alternative, but CPU-expensive techniques. Accurate estimates of *E*-values for pairwise alignments may nevertheless be made possible by recent improvements in hardware (specialised processors) and software. The situation is better with the profiles and the hidden Markov models found in databases such as Prosite or Pfam. These have been carefully calibrated in advance (or endowed with an efficient system of cutoffs), before being added to the public database. This obviously contributes to the success encountered by these types of specialised predictors.

The principles presented here are far from constituting an exhaustive list of all the techniques used to produce *E*-values. We simply hope that this short review will have pointed out the underpinnings of the statistical evaluation of database search results, and can serve as both a friendly guide and a warning post to those inclined to explore further the boundaries of the search parameter space.

Acknowledgements
We wish to thank Philipp Bucher, Stéphane Vuilleumier and Thomas Junier for their helpful comments during the preparation of this manuscript. This work was supported by grant 31-49669.96 from the Swiss National Research Foundation.

References

1. Gumbel, E. J. (1958), 'Statistics of Extremes', Columbia University Press, New York.
2. Olsen, R., Bundschuh, R. and Hwa, T. (1999), 'Rapid assessment of extremal statistics for gapped local alignment', in 'Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology', AAAI Press, Menlo Park, CA, pp. 211–222.
3. 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*, Vol. 87, p. 2264.
4. Bundschuh, R. and Hwa, T. (1999), 'An analytical study of the phase transition line in local sequence alignment with gaps', in 'Recomb 99, Proceedings of the third Annual International Conference on Computational *Molecular Biology*, pp. 70–76.
5. ftp://ftp.isrec.isb-sib.ch/pub/databases/shuffled/
6. Mott, R. (1992), 'Maximum-likelihood estimation of the statistical distribution of Smith–Waterman and local similarity scores', *Bull. Math. Biol.*, Vol. 54, pp. 59–75.
7. Waterman, M. S. and Vingron, M. (1994), 'Rapid and accurate estimates of statistical significance for sequence database searches', *Proc. Natl Acad. Sci. USA*, Vol. 91, pp. 4625–4628.
8. Mott, R. and Tribe, R. (1999), 'Approximate statistics of gapped alignments', *J. Comp. Biol.*, Vol. 1, pp. 91–112.
9. Waterman, M. S., Gordon, L. and Arratia, R.

(1987), 'Phases transitions in sequence matches and nucleic acid structure', *Proc. Natl Acad. Sci. USA*, Vol. 84, pp. 1293–1243.

10. Altschul, S. F. and Gish, W. (1996), 'Local alignment statistics', *Meth. Enzymol.*, Vol. 266, pp. 460–480.
11. Altschul, S. F., Gish, W., Miller, W. *et al.* (1990), 'Basic local alignment search tool', *J. Mol. Biol.*, Vol. 215, pp. 403–410.
12. Altschul, S. F., Madden, T. L., Schäffer, A. A. *et al.* (1997), 'Gapped BLAST and PSI-BLAST: A new generation of protein database search programs', *Nucleic Acids Res.*, Vol. 25, pp. 3389–3402.
13. Anonymous (2000), 'Blast tutorial', http://www.ncbi.nlm.nih.gov/BLAST/tutorial/
14. Pearson, W. R. (1990), 'Rapid and sensitive sequence comparison with FASTP and FASTA', *Methods Enzymol.*, Vol. 183, pp. 63–98.
15. Brenner, S. E., Chothia, C. and Hubbard, T. J. P. (1998), 'Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships', *Proc. Natl Acad. Sci. USA*, Vol. 95, pp. 6073–6078.
16. Park, J., Karplus, K., Barrett, C. *et al.* (1998), 'Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods', *J. Mol. Biol.*, Vol. 284, pp. 1201–1210.
17. Pearson, W. R. (1998), 'Empirical statistical estimates for sequence similarity searches', *J. Mol. Biol.*, Vol. 276, pp. 71–84.
18. Pearson, W. R. (1996), 'Effective protein sequence comparison', *Methods Enzymol.*, Vol. 266, pp. 227–258.
19. Baumgartner, S., Hofmann, K., Chiquet-Ehrismann, R. and Bucher, P. (1998), 'The discoidin domain family revisited: New members from prokaryotes and a homology-based fold prediction', *Protein Sci.*, Vol. 7, pp. 1626–1631.
20. Kornblihtt, A. R., Umezawa, K., Vibe-Pedersen, K. and Baralle, F. E. (1985), 'Primary structure of human fibronectin: Differential splicing may generate at least 10 polypeptides from a single gene', *EMBO J.*, Vol. 4, pp. 1755–1759.
21. Bateman, A., Birney, E., Durbin, R. *et al.* (1999), 'Pfam 3.1: 1313 multiple alignments match the majority of proteins', *Nucleic Acids Res.*, Vol. 27, pp. 260–262.
22. Bateman, A., Birney, E., Durbin, R. *et al.* (2000), 'The Pfam protein families database', *Nucleic Acids Res.*, Vol. 28, pp. 263–266.
23. Hofmann, K., Bucher, P., Falquet, L. and Bairoch, A. (1999), 'The PROSITE database, its status in 1999', *Nucleic Acids Res.*, Vol. 27, pp. 215–219.
24. Bucher, P., Karplus, K., Moeri, N. and Hofmann, K. (1996), 'A flexible motif search technique based on generalised profiles', *Comput. Chem.*, Vol. 20, pp. 3–23.
25. Eddy, S. R. (1997), 'Maximum likelihood fitting of extreme value distributions', ftp://ftp.genetics.wustl.edu/pub/eddy/papers/evd.pdf