Statistical Science
1994, Vol. 9, No. 3, 367–381

Sequence Comparison Significance and Poisson Approximation

Michael S. Waterman and Martin Vingron

Abstract. The Chen–Stein method of Poisson approximation has been used to establish theorems about comparison of two DNA or protein sequences. The most useful result for sequence alignment applies to alignment scoring with no gaps. However, there has not been a valid method to assign statistical significance to alignment scores with gaps. In this paper we extend Poisson approximation techniques using the Aldous clumping heuristic to a practical method of estimating statistical significance.

Key words and phrases: Poisson approximation, clumping, dynamic programming sequence comparison, sequence alignment, DNA, protein.

1. INTRODUCTION

Since the invention of rapid gene sequencing techniques in the mid-1970's, new genetic sequences from a wide range of organisms have been determined. The DNA sequences have been placed into internationally available databases since about 1984. Figure 1 gives the growth of the DNA sequence data in GenBank, the database funded by NIH. The DNA Data Bank of Japan (DDBJ), the EMBL Data Library, and GenBank are collaborating and virtually equivalent databases. The doubling time for the DNA data is approximately 2 years. In addition there are protein sequence databases where the amino acid sequences of genes are stored. These sequence databases are an important resource for biological sciences. New sequences are quickly entered, making the databases very dynamic. Not only are the sequences themselves stored but basic biological information about the sequences and relevant references are included as well.

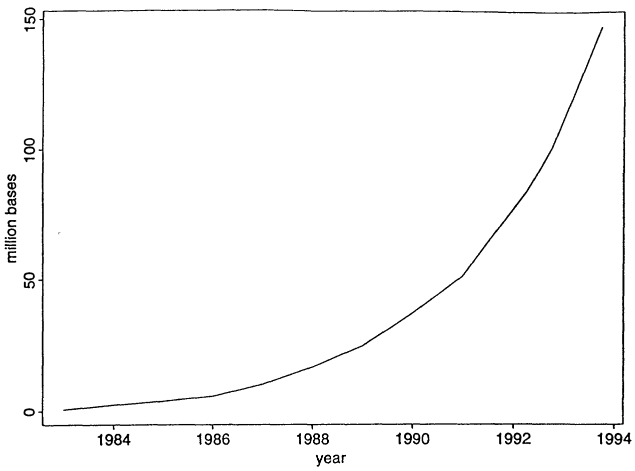

FIG. 1. *DNA database size.*

All new DNA or protein sequences are compared to the appropriate sequence databases to find sequences that are "close" in a sense to be made precise later. These searches have become central to the practice of modern molecular biology, and they are based on ideas from evolution. The evolutionary process usually proceeds by utilizing existing genes. If mutation of a current gene gives a selective advantage, then that mutation has an increased chance of being fixed in the population. Thus all or part of a protein sequence in one organism might appear as all or part of another protein sequence in the same or another organism; knowing this relationship between a new sequence and an already studied sequence can give valuable clues as to the function of the new sequence. Many important discoveries have resulted from sequence database searches. A similarity between the human cancer related viral *v-sis oncogene* product and platelet-derived growth factor (Doolittle et al., 1983) gave valuable insight into how the cancer was regulated. The sequence similarity was great with a stretch of 50–60 identical amino

Michael S. Waterman is Professor of Mathematics and of Biological Sciences; Departments of Mathematics and Molecular Biology, University of Southern California, Los Angeles, California 90089-1113. Martin Vingron is a postdoctoral fellow at Gesellschaft für Mathematik und Datenverarbeitung, Institut I1, Schoss Birlinghoven, D-53757 St. Augustin, Germany.

acids. Other similarities are just as important but less dramatic at the sequence level. Cystic fibrosis is a recessive genetic disease carried by about $\frac{1}{25}$ of the Caucasian population. Recently the gene of the most prevalent allele for the disease was cloned and sequenced (Riordan et al., 1989). A database search showed that the gene product is similar to a family of related protein sequences that bind ATP and are involved in the transport of small hydrophilic molecules across the cytoplasmic membrane. While the similarity was weaker, it allowed a structure and function of the gene product to be proposed.

A nice, well-studied example is the family of hemoglobin sequences, which are used to illustrate the ideas developed in this paper. Hemoglobin is a protein of red blood cells that binds oxygen. This molecule is very important as larger organisms (animals) cannot obtain oxygen simply by diffusion from the air. A similar molecule is found in all vertebrates and in many invertebrates. The most primitive globin is a protein of about 150 amino acids and is utilized in insects, worms and some fish. In higher (more recently evolved) organisms there are two kinds of globins that apparently came from gene mutations and duplications. The two globins, α and β, appear in a complex of four globin sequences, two α-globins and two β-globins, that comprise the hemoglobin molecule in higher vertebrates. In addition, there is apparently another evolutionary sequence of events leading to the γ-globin used in embryos and to the δ-globin found only in adult primates. There is even a hemoglobin-like protein expressed in plants. Thus this is a well-studied and varied family of proteins that can test our ability to understand the results of database searches.

The outline of the paper is as follows. Sequence comparison will be reviewed. The dynamic programming comparison algorithms in Section 2 are motivated by the biology just discussed. Each comparison results in a score that is the basis of determining possible similarity. Then the known results for assigning statistical significance to the comparison scores are discussed in Section 3. The statistical distribution of scores depends critically on certain parameters of the algorithm. Some of the most useful results are motivated by the Chen–Stein method of Poisson approximation. In Section 4 this method is extended by the Aldous clumping heuristic to a practical method of estimating statistical significance for the most useful part of the algorithm parameter space. The model is tested on simulated data in Section 5. A numerical method is presented in Section 6 to estimate the two parameters of Poisson approximation, and the quality of approximation is studied. The technique is applied to a database search using a globin sequence. Finally, in Section 8 data from a database search are used to test the model and to improve the parameter estimates.

2. ALGORITHMS

In this section we present the basic dynamic programming algorithms used to compare genetic sequences. See Waterman (1984) for a review. Two nondynamic programming algorithms, FASTA (Lipman and Pearson, 1985; Wilbur and Lipman, 1983) and BLAST (Altschul et al., 1990), for rapid database searches are very well known and widely used. Both these algorithms are faster than the quadratic algorithms presented below, and both can be considered heuristics for the comparison score we compute here using dynamic programming (Pearson, 1991). Thus the statistical methods we present can be used for these rapid search techniques, and in the case of BLAST are already an integral part of the algorithm.

Let us set the stage. Given are two sequences $\mathbf{x} = x_1x_2\cdots x_n$ and $\mathbf{y} = y_1y_2\cdots y_m$ over a finite alphabet. For DNA the alphabet has 4 letters; for protein sequences it has 20 letters. Later the letters will become random; for now they are deterministic. There is a scoring function $s(x, y)$ for aligning letter x with letter y. Not only do letters change (y is "substituted" for x) but they are inserted or deleted (an indel). For example, let

$$s(x, y) = \begin{cases} +1, & x = y, \\ -\mu, & x \neq y, \end{cases}$$

and score indels by $-\delta$. The alignment

```
ATAGC
AAGCC
```

scores $2 - 3\mu$. The score can be changed with appropriate indels:

```
ATAGC-
A-AGCC
```

scores $4 - 2\delta$. Which is preferable depends on the value of (μ, δ). Our problem is to compute the global alignment score $S(\mathbf{x}, \mathbf{y})$ = maximum alignment score over all alignments of $\mathbf{x}$ and $\mathbf{y}$. Alignment score can be computed from specifying the k aligned letters $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ and $1 \leq j_1 < j_2 < \cdots < j_k \leq m$, for $k \geq 0$, so there are a total of

$$\sum_{k\geq 0} \binom{n}{k}\binom{m}{k} = \binom{n+m}{k}$$

alignments. In the alignments above, $k = 5$ and 4, respectively. See Figure 2. Instead of maximizing

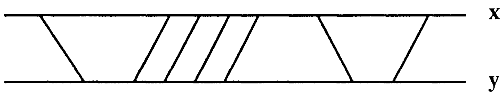

FIG. 2. *Global alignment with $k = 7$.*

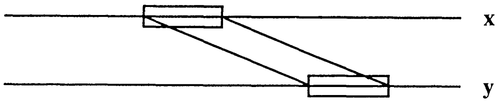

FIG. 3. *Local alignment of intervals I and J.*

over this exponential number of alignments, $S(\mathbf{x}, \mathbf{y})$ can be computed in $O(nm)$ steps.

Define

$$S_{ij} = S(x_1x_2 \cdots x_i, y_1y_2 \cdots y_j),$$

with $S_{0,j} = -\delta j$ and $S_{i,0} = -\delta i$. Then the recursive step of the algorithm is

$$S_{i,j} = \max\{S_{i-1,j} - \delta, S_{i,j-1} - \delta, S_{i-1,j-1} + s(x_i, y_j)\}. \tag{1}$$

The score is found by $S_{n,m} = S(\mathbf{x}, \mathbf{y})$. The algorithm is derived from considering the three ways an alignment can end,

$$\begin{matrix} x_i & - & x_i \\ - & y_j & y_j \end{matrix},$$

and assuming that the other letters are optimally aligned. In Table 1 we present a small example of sequence alignment with

$$s(x, y) = \begin{cases} +1, & x = y, \\ -1, & x \neq y, \end{cases}$$

and $\delta = 1$. The two optimal alignments are shown by the two boxed patterns in the matrix in Table 1 and are

GATC--AATTCGCA
TATCTTAA--CGCC

and

GATCAATT--CGCA
TATC--TTAACGCC.

Such alignments where all letters from each sequence are in the alignment are called *global alignments*. Next we turn to *local alignments*, where intervals of $\mathbf{x}$ and $\mathbf{y}$ are optimally aligned. See Figure 3. Actually local alignment is

$$\left(\binom{n+1}{2} + 1\right)\left(\binom{m+1}{2} + 1\right)$$

global alignment problems since, for example, there are $\binom{n}{2} + n + 1$ intervals $x_ix_{i+1} \cdots x_j$ with $1 \le i \le j \le n$ including the empty interval. Our objective function is $H(\mathbf{x}, \mathbf{y})$:

$$\begin{aligned} H(\mathbf{x}, \mathbf{y}) &= \max\{0; S(x_ix_{i+1} \cdots x_j, y_ky_{k+1} \cdots y_l): \\ &\qquad 1 \le i \le j \le n,\ 1 \le k \le l \le m\} \\ &= \max\{0; S(I, J): I \subset \mathbf{x},\ J \subset \mathbf{y}\}. \end{aligned}$$

The quantity 0 comes from the empty intervals and getting $S(\emptyset, \emptyset) = 0$. There is a nice recursion for this quantity, too. Define

$$H_{i,j} = \max\{0; S(x_k \cdots x_i, y_l \cdots y_j): 1 \le k \le i,\ 1 \le l \le j\}$$

TABLE 1
Global alignment

	–	T	A	T	C	T	T	A	A	C	G	C	C
–	0	−1	−2	−3	−4	−5	−6	−7	−8	−9	−10	−11	−12
G	−1	−1	−2	−3	−4	−5	−6	−7	−8	−9	−8	−9	−10
A	−2	−2	0	−1	−2	−3	−4	−5	−6	−7	−8	−9	−10
T	−3	−1	−1	1	0	−1	−2	−3	−4	−5	−6	−7	−8
C	−4	−2	−2	0	2	1	0	−1	−2	−3	−4	−5	−6
A	−5	−3	−1	−1	1	1	0	1	0	−1	−2	−3	−4
A	−6	−4	−2	−2	0	0	0	1	2	1	0	−1	−2
T	−7	−5	−3	−1	−1	1	1	0	1	1	0	−1	−2
T	−8	−6	−4	−2	−2	0	2	1	0	0	0	−1	−2
C	−9	−7	−5	−3	−1	−1	1	1	0	1	0	1	0
G	−10	−8	−6	−4	−2	−2	0	0	0	0	2	1	0
C	−11	−9	−7	−5	−3	−3	−1	−1	−1	1	1	3	2
A	−12	−10	−8	−6	−4	−4	−2	0	0	0	0	2	2

TABLE 2

Local alignment (a) optimal alignments and (b) declumped matrix with second-best alignment

(a)

	-	T	A	T	C	T	T	A	A	C	G	C	C
-	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	1	0	0
A	0	0	$\boxed{1}$	0	0	0	0	1	1	0	0	0	0
T	0	1	0	$\boxed{2}$	1	1	1	0	0	0	0	0	0
C	0	0	0	1	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	0	0	1	0	1	1
A	0	0	1	0	$\boxed{2}$	2	1	$\boxed{2}$	1	0	0	0	0
A	0	0	1	0	$\boxed{1}$	1	1	2	$\boxed{3}$	2	1	0	0
T	0	1	0	2	1	$\boxed{2}$	2	1	$\boxed{2}$	2	1	0	0
T	0	1	0	1	1	2	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$	1	1	0	0
C	0	0	0	0	2	1	2	2	1	$\boxed{2}$	1	2	1
G	0	0	0	0	1	1	1	1	1	1	$\boxed{3}$	2	1
C	0	0	0	0	1	0	0	0	0	2	2	$\boxed{4}$	3
A	0	0	1	0	0	0	0	1	1	1	1	3	3

(b)

	-	T	A	T	C	T	T	A	A	C	G	C	C
-	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	1	0	0
A	0	0	0	0	0	0	0	1	1	0	0	0	0
T	0	1	0	0	0	1	1	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	1	0	1	1
A	0	0	1	0	0	0	0	0	1	0	0	0	0
A	0	0	$\boxed{1}$	0	0	0	0	1	0	0	0	0	0
T	0	1	0	$\boxed{2}$	1	1	1	0	0	0	0	0	0
T	0	1	0	1	1	2	2	1	0	0	0	0	0
C	0	0	0	0	2	1	1	1	0	0	0	1	1
G	0	0	0	0	1	1	0	0	0	0	0	0	0
C	0	0	0	0	1	0	0	0	0	1	0	0	1
A	0	0	1	0	0	0	0	1	1	0	0	0	0

to be the best score of any intervals ending at x_i and y_j or 0 if no such alignment scores positive. The algorithm begins with $H_{i,j} = 0$ if $i \cdot j = 0$. Then

$$H_{i,j} = \max\{H_{i-1,j} - \delta, H_{i,j-1} - \delta, H_{i-1,j-1} + s(i,j), 0\} \tag{2}$$

just as in (1) except for the initial conditions and 0 in the recursion. We find

$$H(\mathbf{x}, \mathbf{y}) = \max_{\substack{1 \le i \le u \\ 1 \le j \le v}} H_{i,j}.$$

The local algorithm is known as the Smith–Waterman algorithm (Smith and Waterman, 1981). See Table 2 for an example of this algorithm with $s(x, y) = 1, x = y, s(x, y) = -1, x \neq y$ and $\delta = 1$. The optimal alignments are

ATC--AATTCGC
ATCTTAA--CGC

and

ATCAATT--CGC
ATC--TTAACGC

The nonuniqueness of optimal alignments is typical. In fact around a high-scoring alignment there are many intersecting alignments that are optimal or near-optimal and differ in small details only. Sometimes it is of biological interest to debate these small details but usually we are only concerned if there are any other high-scoring alignments that are not too dependent on the first alignment. Motivated by this we define a *clump of alignments* to be the set of alignments sharing at least one pair of aligned letters with a given alignment. When calculating H order $(i, j, H_{i,j})$ by $\succ$ as follows: $(i, j, H_{i,j}) \succ (k, l, H_{k,l})$ if

$$\begin{aligned} & H_{i,j} > H_{k,l}, \\ \text{or} \quad & H_{i,j} = H_{k,l} \quad \text{and} \quad i + j < k + l, \\ \text{or} \quad & H_{i,j} = H_{k,l}, \quad i + j = k + l \text{ and } i < k. \end{aligned}$$

The optimal alignment first output ends at $(i, j, H_{i,j})$, which is largest under $\succ$. We pick one of the alignments ending at (i, j) with score $H_{i,j}$. Then we *declump* by removing the effect of all alignments in the clump; $H^*_{i,j}$ is the matrix computed by not allowing any aligned pair in the output alignment. Let (k, l) be the upper-left position of aligned letters in the alignment. Then

$$H^*_{i,j} = H_{i,j} \quad \text{when } i < k \text{ or } j < l,$$

and

$$H^*_{k,l} = \max\{0, H^*_{k-1,l} - \delta, H^*_{k,l-1} - \delta\}.$$

The remainder of the row $(k, j), l < j$, can be computed term by term until $H_{k,j} = H^*_{k,j}$. Then there is no need to continue, as the recursion will now always return $H_{k,j} = H^*_{k,j}$ for the rest of the row. Declumping continues by switching from row to column until the effects of the alignment clump are removed. This is much cheaper in time than just redoing the entire matrix. The K best clump scores can be found in this manner: $H_{(1)} \geq H_{(2)} \geq \cdots \geq H_{(K)}$

The results of the local alignment in Table 2 are now analyzed by the declumping algorithm. As before, $s(x, y) = +1$ if $x = y$; $s(x, y) = -1$ if $x \neq y$; and $\delta = 1$. The two optimal alignments are

ATCTTAA--CGC
ATC--AATTCGC

and

```
ATC--TTAACGC
ATCAATT--CGC.
```

There are several next-best alignments of score 2. The first is shown in Table 2, with the entries changed by declumping the best alignments outlined. The first local alignment after declumping is also outlined:

```
AT
AT.
```

In biology indels larger than one letter often appear, and they are likely to be the result of one event rather than the sum of one-letter events. Thus it is desirable to score such indels as one event. Computational efficiency can be obtained for the gap penalty $-g(k) = -\alpha - \beta k$ for an indel of k letters. The first letter costs $\alpha + \beta$, and each succeeding letter costs β. Three recursions are required: E, F and H. Set $E_{i,j} = F_{i,j} = H_{i,j} = 0$ if $i \cdot j = 0$. Then the recursion due to Gotoh (1982) is

$$\begin{aligned} E_{i,j} &= \max\{H_{i,j-1} - (\alpha+\beta), E_{i,j-1} - \beta, 0\}, \\ F_{i,j} &= \max\{H_{i-1,j} - (\alpha+\beta), F_{i-1,j} - \beta, 0\}, \\ H_{i,j} &= \max\{H_{i-1,j-1} + s(x_i, y_j), E_{i,j}, F_{i,j}, 0\}. \end{aligned}$$

For declumping, all three matrices must be recomputed, stopping in a row or column when all these agree with the earlier matrices.

3. DISTRIBUTIONAL RESULTS

Now take random sequences $\mathbf{X} = X_1X_2\cdots X_n$, $\mathbf{Y} = Y_1Y_2\cdots Y_n$, where X_i and Y_j are iid. Given a scoring scheme

$$s(x, y) = \begin{cases} 1, \\ -\mu \end{cases}$$

with $\mu \geq 0$ and $\delta \geq 0$, our interest is in the random variable $H(\mathbf{X}, \mathbf{Y})$. When we do database searches with a new sequence $\mathbf{X}$ there are tens of thousands of database sequences $\mathbf{Y}$, so with no biological similarity we will see large-deviations behavior. Before moving to a study of what scores should surprise us, it is instructive to ask about the growth of score with sequence length. This is motivated by the variety of sequence lengths in the database.

First we consider global alignment scores. Let

$$S_n = S(X_1\cdots X_n, Y_1\cdots Y_n)$$

and observe that

$$S_{n+m} \geq S_n + S(X_{n+1}\cdots X_{n+m}, Y_{n+1}\cdots Y_{n+m});$$

moreover, $S(X_{n+1}\cdots X_{n+m}, Y_{n+1}\cdots Y_{n+m})$ equals S_m in distribution. The theory of subadditive sequences implies that the following limit exists:

$$(3) \qquad a(\mu, \delta) = \lim_{n\to\infty} \frac{\mathbb{E}(S_n)}{n} = \sup_{n\geq 1} \frac{\mathbb{E}(S_n)}{n}.$$

In fact Kingman's subadditive ergodic theorem (Kingman, 1973) applies, giving

$$(4) \qquad a(\mu, \delta) = \lim_{n\to\infty} \frac{S_n}{n} \quad \text{a.s. and in } L_1.$$

When $\mu = \infty$ and $\delta = 0$, S_n is the length of the longest common subsequence of $X_1X_2\cdots X_n$ and $Y_1Y_2\cdots Y_n$ and $a(\infty, 0)$ is the Chvátal–Sankoff (1975) constant. Unfortunately, even for $P(X_i = 0) = 1 - P(X_i = 1) \in (0, 1)$ the constant is unknown.

Now $S_n \leq H(X_1\cdots X_n, Y_1\cdots Y_n) = H_n$, so that

$$\frac{S_n}{n} \leq \frac{H_n}{n} \leq \frac{n}{n} = 1,$$

and the asymptotic growth of H_n is "caught" between $a(\mu, \delta)n$ and n. In fact it is not too surprising and can be proved that, when $a(\mu, \delta) > 0$,

$$\mathbb{P}\left(\lim_{n\to\infty} \frac{H_n}{n} = a(\mu, \delta)\right) \to 1.$$

Moreover, when $a(\mu, \delta) < 0$ it can be proved that H_n grows like a constant time $\log(n)$. When $a(\mu, \delta) < 0$, there is a constant b such that, for all $\varepsilon > 0$,

$$(5) \qquad \mathbb{P}\left((1-\varepsilon)b < \frac{H_n}{\log(n)} < (2+\varepsilon)\,b\right) \to 1,$$

and $H_n/\log(n) \to 2b$ is conjectured to hold. Having divided the growth of H_n into linear and logarithmic regions, it should also be noted that $\{(\mu, \delta): a(\mu, \delta) = 0\}$ defines a line in $[0, \infty]^2$ separating $\{a < 0\}$ from $\{a > 0\}$. Of course $(\infty, 0)$ and $(0, 0) \in \{a > 0\}$ while $(\infty, \infty) \in \{a < 0\}$. Thus there is a phase transition between linear and logarithmic growth in $[0, \infty]^2$. A graph of the parameter space with the phase transition curve appears in Figure 4. These results appear in Arratia and Waterman (1994).

Moving back to biological motivation for a moment, recall that we wish to find aligning intervals that have more similarity than random sequences. It is not productive to use $(\mu, \delta) \in \{a > 0\}$ since even if there are such intervals they will be surrounded by or even obscured by alignments that have only random sequences aligned. In the linear region, it is not penalized just to wait for another well aligned pair because the penalty for poorly aligned pairs and indels is too low. Thus we are motivated to use logarithmic penalties. For a discussion of the effects of

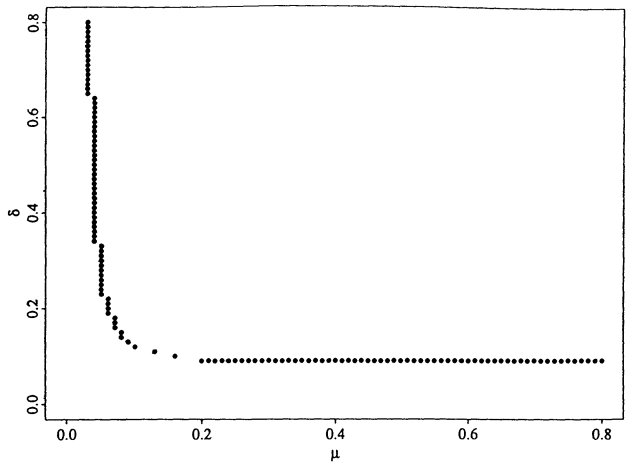

FIG. 4. *Phase transition curve.*

parameter choice on biological sequence alignment see Vingron and Waterman (1994).

Just where does the logarithmic region come from? A simple heuristic can be given and later made rigorous. Take $\mu = \delta = \infty$ and let

$$p = \mathbb{P}(X = Y).$$

Then $Z_{i,j} = 1$ if a match of length k starts at X_i and Y_i $(Z_{i,j} = \mathbb{I}\{X_i X_{i+1} \cdots X_{i+k-1} = Y_j Y_{j+1} \cdots Y_{j+k-1}\})$ and, neglecting end effects,

$$\mathbb{E}\left(\sum_{i,j} Z_{ij} \right) = n^2 p^k.$$

If the longest match occurs about once, take

$$1 = n^2 p^k$$

and solve for k, obtaining

$$k = 2 \log_{1/p}(n).$$

Aldous has formulated the *Poisson clumping heuristic* which we use as the basis of our calculation of alignment p-values. The heuristic is employed in cases where occurrences happen in clumps and where the distribution of the number of clumps is approximately Poisson. The approximation is to locate the clumps according to a Poisson process and then assign iid clump sizes to the locations. Aldous formalizes this approximating process as a mosaic process.

Our use of Poisson approximation is not rigorous; for the full range of parameters where we believe and provide evidence that Poisson approximation holds, we are unable to give a theorem. Special cases have been proved, however, and to this end we give two theorems on Poisson approximation by the Chen–Stein method as they appear in Arratia, Goldstein and Gordon (1989). It is not the purpose of this paper to give careful proofs of all results, but these theorems allow easy proofs of some important results and moreover provide a nice guide for our intuition in other situations.

Define the total variation distance between two random variables U and V by

$$\|\mathcal{L}(U) - \mathcal{L}(V)\| = 2 \sup_A |\mathbb{P}(U \in A) - \mathbb{P}(V \in A)|,$$

where the sup is over all subsets of the reals. The bookkeeping is done in the following way. There is a finite or countable index set I. For each $\alpha \in I$, U_α is a Bernoulli random variable and

$$p_\alpha = \mathbb{P}(U_\alpha = 1) > 0.$$

Set

$$W = \sum_{\alpha \in I} U_\alpha$$

and assume

$$\lambda = \mathbb{E}(W) = \sum_{\alpha \in I} p_\alpha \in (0, \infty).$$

Take Z to be Poisson(λ). For each $\alpha \in I$ we have a set $B_\alpha \subset I$ with $\alpha \in B_\alpha$ to be a set of indices where $\beta \notin B_\alpha$ implies U_α and U_β are independent. Then define

$$b_1 = \sum_{\alpha \in I} \sum_{\beta \in B_\alpha} p_\alpha p_\beta$$

and

$$b_2 = \sum_{\alpha \in I} \sum_{\alpha \neq \beta \in B_\alpha} p_{\alpha\beta},$$

where

$$p_{\alpha\beta} = \mathbb{E}(U_\alpha U_\beta).$$

The following theorem gives error bounds on the approximation of W by a Poisson random variable Z.

THEOREM 1. *Let $W = \sum_{\alpha \in I} U_\alpha$ be the number of occurrences, and let Z be Poisson with $0 < \lambda = \mathbb{E}W = \mathbb{E}Z < \infty$. Then*

$$\|\mathcal{L}(W) - \mathcal{L}(Z)\| \le 2(b_1 + b_2)\left(\frac{1 - e^{-\lambda}}{\lambda}\right) \le 2(b_1 + b_2)$$

and

$$|\mathbb{P}(W = 0) - e^{-\lambda}| \le (b_1 + b_2)\left(\frac{1 - e^{-\lambda}}{\lambda}\right) \le (b_1 + b_2).$$

There is a process version of the theorem that proves to be very useful in our application.

THEOREM 2. *Let* $\mathbf{Z} = \{Z_\alpha\}_{\alpha \in I}$ *be an independent Poisson process with* Z_α *of mean* p_α. *The total variation distance between* $\mathbf{Z}$ *and* $\mathbf{U} = \{U_\alpha\}_{\alpha \in I}$ *satisfies*

$$\|\mathcal{L}(\mathbf{Z}) - \mathcal{L}(\mathbf{U})\| \le 4(b_1 + b_2).$$

The length of the longest head run R_n in n independent coin tosses is closely related to the longest matching between two sequences. First we set a test length t so that runs of length t occur with small probability. While it seems that counting these runs should give a Poisson random variable, the clumping or overlapping of the runs makes this idea fail. Given a run of length t, there is a geometric number of runs in a clump. Instead we count the clumps by counting the leftmost t-run in each clump. The iid sequence $V_1, V_2, \dots$ has $p = \mathbb{P}(V_i = 1) = 1 - \mathbb{P}(V_i = 0) \in (0, 1)$. The index set is $\{1, 2, \dots, n\}$. The leftmost run of length t has indicator

$$U_1 = V_1 V_2 \cdots V_t,$$

and for $\alpha > 1$ we declump by requiring $V_{\alpha-1} = 0$:

$$U_\alpha = (1 - V_{\alpha-1}) V_\alpha V_{\alpha+1} \cdots V_{\alpha+t-1}.$$

As above, the sum of indicators is

$$W = \sum_{\alpha=1}^{n-t+1} U_\alpha.$$

Note that

$$\{R_n < t\} = \{W = 0\}$$

and

$$\lambda = \lambda_n(t) = \mathbb{E}(W) = p^t\{(n - t)(1 - p) + 1\}.$$

So if we obtain a Poisson $\lambda_n(t)$ approximation for W,

$$\mathbb{P}(R_n \ge t) \approx 1 - \exp[-\lambda_n(t)].$$

To obtain bounds for the approximation set $B_\alpha = \{\beta \in I: |\alpha - \beta| \le t\}$. It follows that $b_1 < \lambda^2(2t+1)/n + 2\lambda p^t$ and $b_2 = 0$. To have an interesting approximation, we need λ bounded away from 0 and ∞, which holds if and only if $t - \log_{1/p}(n)$ is bounded. In this case $b_1 = O(\log(n)/n) \to 0$ as $n \to \infty$.

Extending this result to sequence matching is done by setting $I = \{(i, j): 1 \le i \le n,\ 1 \le j \le m\}$. The declumped random variables $U_\alpha = U_{(i,j)}$ are defined by

$$U_{i,j} = \mathbb{I}\{X_i X_{i+1} \cdots X_{i+t-1} = Y_j Y_{j+1} \cdots Y_{j+t-1}\},$$

if i or $j = 0$, and otherwise by

$$U_{ij} = \mathbb{I}\{X_{i-1} \ne Y_{j-1} \text{ and } X_i X_{i+1} \cdots X_{i+t-1} = Y_j Y_{j+1} \cdots Y_{j+t-1}\}.$$

The dependence set for $\alpha = (i, j) \in I$ is

$$B_\alpha = \{(i', j') \in I: |i - i'| \le t \text{ or } |j - j'| \le t\}.$$

For this situation,

$$b_1 < (n - t + 1)(m - t + 1)(2t + 1)^2 p^t$$

and

$$\lambda = \lambda_{n,m}(t) = p^t\{(n+m-2t-1) + (n-t)(m-t)(1-p)\}.$$

Therefore $t = \log_{1/p}(nm(1 - p)) + c$ will keep λ between 0 and ∞. When $n = m$, λ is approximately p^c. Therefore we can derive bounds on

$$\left|\mathbb{P}(H_{n,m} < t) - \exp(-\lambda_{n,m}(t))\right|,$$

providing a Poisson approximation for pure matching between two sequences of lengths n and m.

Several extensions of these results have been made. For the longest match between two random sequences where the fraction of identities is $\beta > p$, see Arratia, Gordon and Waterman (1990). There, no indels are allowed, only mismatches. The ballot theorem is used for a much more complicated dependence structure. Neuhauser (1994) extends those results to cover a fraction of indels. While these are mathematically nontrivial results, they fall far short of covering our general random variable H_n.

The most useful analytical result in this area covers the case $\delta = \infty$ so that there are no indels. The scoring function must satisfy $\mathbb{E}(s(X, Y)) < 0$ for random letters X and Y. A widely used scoring function on amino acids that fulfills the criterion is the PAM250 matrix (Dayhoff, Barker and Hunt, 1983). Let $p \in (0, 1)$ be the largest root of

$$1 - \mathbb{E}(\lambda^{-s(X,Y)}) = 0.$$

Then

$$\lim \frac{H_n}{\log_{1/p}(n^2)} \to 1,$$

with probability 1. This was proved by Arratia, Morris and Waterman (1988) and generalized for more general scoring by Karlin and Altschul (1990), who presented the following Poisson approximation. Let $t = \log_{1/p}(nm) + c$. Then

$$\mathbb{P}(H(\mathbf{X}, \mathbf{Y}) > t = \log_{1/p}(nm) + c) \approx 1 - \exp(-\gamma nmp^t), \tag{6}$$

where γ is found by numerical solution of an equation.

4. THE MODEL

Above, reasons were given to restrict attention to algorithm parameters in the logarithmic region. In the logarithmic region, the expected score per letter is negative, and positive-scoring local alignments are rare events. Certainly, positive-scoring local alignments occur in clumps, and we even have an algorithm to declump. In coin tossing we take the leftmost run of length t and no runs overlapping that clump of t-runs. In sequence matching we take an alignment ending at (i, j) and no alignments intersecting that alignment. Neuhauser (1994) follows this approach of Waterman and Eggert (1987) when studying Poisson approximation of alignments with indels. Our model is to follow the Aldous clumping heuristic (Aldous, 1989). Alignment clumps are laid down by a Poisson process, and each alignment clump is assigned an independent clump size. The number of clumps with scores larger than a test value $t = (\text{center} + c)$ has a Poisson distribution with mean $\lambda_{n,m}(t)$. We will apply this in the form

$$\begin{aligned}&\mathbb{P}(\text{at least one score exceeds } t)\\&\quad = 1 - \mathbb{P}(\text{no score exceeds } t)\\&\quad = 1 - e^{-\lambda}.\end{aligned}$$

To relate this to alignment, set

$$W(t) = \text{number of alignment clumps of score greater than or equal to } t.$$

$W(t)$ can be calculated by applying the declumping algorithm until $H_{(i)} < t$.

Our goal is to show that equation (6) fits well in the entire logarithmic region. We will estimate the parameters γ and p in that equation, where $t = \text{center} + c = \log_{1/p}(mn) + c$. In carrying that formula over to the logarithmic region, we implicitly make several assumptions. We want to identify and then test the following three assumptions:

(A1) $W(t)$ is approximately Poisson distributed [with mean $\lambda_{n,m}(t) = \mathbb{E}(W(t))$].

(A2) $\mathbb{E}(W(t)) = \widehat{\gamma} p^t$.

(A3) $\widehat{\gamma} = \gamma mn$.

These three assumptions are sufficient to approximate the significance of alignments with gaps. Assumption (A1) is used to estimate the significance of optimal and suboptimal scores; (A2) allows us to interpolate the mean of the Poisson in the tail where simulations rarely yield sufficient data to estimate $\mathbb{E}(W(t))$. Assumptions (A1) and (A2) together mean that the empirical distribution function of optimal scores less than t is approximated by $\exp(-\widehat{\gamma} p^t)$; (A3) allows us to normalize scores for sequences of different length from only knowing γ and p.

5. TESTING THE MODEL

All our tests rely on simulated alignment scores obtained for sequences with iid letters for some given letter distribution. In the remainder of the paper we will study protein sequences with the alphabet of 20 amino acids. For our protein sequence simulations we will use the amino acid distribution of McCaldon and Argos (1988). The scoring matrix and gap penalties are chosen in the logarithmic region. To test (A1), we collected scores of suboptimal alignments for many (between 1,000 and 10,000) alignments of sequences of length $n = m = 900$ with PAM250 and $g(k) = 12 + 3k$. Then, for given threshold t, the number of clumps that score above t is counted. The quality of the approximation depends strongly on the threshold chosen. Generally the higher the threshold, the better is the empirical distribution approximated by a Poisson. Figure 5 shows data for different thresholds. Those thresholds for which the approximations look very good are at approximately the $t > 80$ level and higher in the distribution function, as can be seen from the first value in the bar diagrams. The high quality of the Poisson approximation for large t is exactly what we need to assign the statistical significance of large scores.

We use $m = n = 900$ in our simulations while protein sequences are often smaller than this. For parameter estimation discussed in the next section, repeated declumping is necessary. The larger values of m and n allow enough "area" for declumping to give us a large number of nonoverlapping alignments. We tested other pairs of lengths, such as $m = n = 400$ and $m = 200, n = 800$. The results are essentially

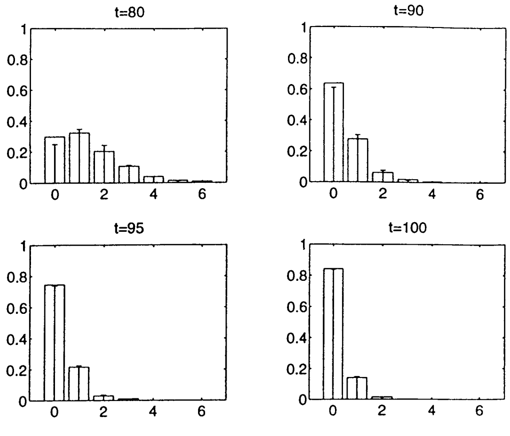

FIG. 5. *Poisson approximation.*

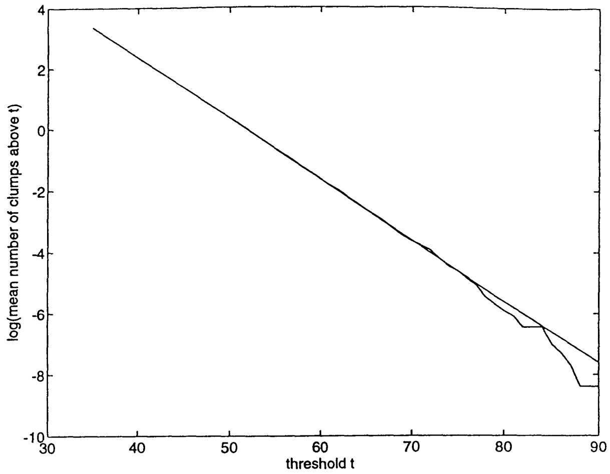

FIG. 6. *Logarithm of estimated mean number of clumps exceeding t.*

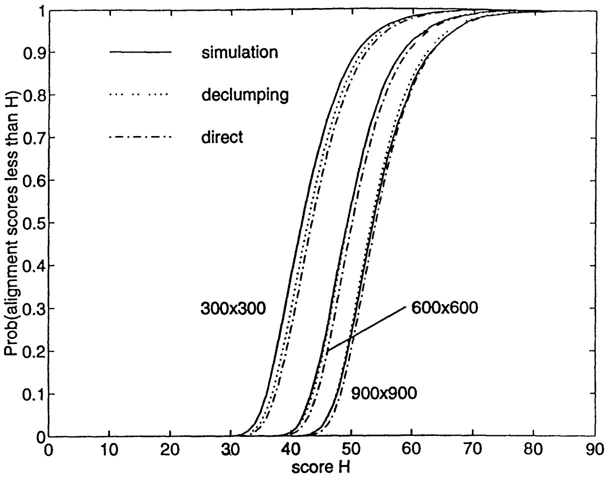

FIG. 7. *Comparison of fits for estimates made from $n = 900$ length sequences.*

the same. As computation time is proportional to mn, it is best for our declumping estimates to minimize boundary effects with $m = n$.

Testing (A2) was done by accumulating suboptimal solutions from 10,000 comparisons and counting the number above a threshold t with PAM250 and $g(k) = 12 + 3k$. The logarithm of the resulting curve is shown in Figure 6. The regression line is based on the interval [35, 60]. For t larger than 75 the data are sparse.

Assumption (A3) implies that $\log(\lambda(t))$ derived from sequences of different length should give parallel lines with the parameter settings of Figure 5, PAM250 and $g(k) = 12+3k$. While this holds in general, boundary effects seem to be responsible for certain limitations in the validity of this assumption. For example, the slopes (of the logarithm shown in Figure 6) derived from the simulations of $n = m = 900$ long sequence pairs and from $n = m = 1{,}000$ long sequence pairs are -0.1995 and -0.1972, respectively. For short sequences of length 300, it is -0.2043. It is thus easy to normalize the parameters derived from sequence pairs of lengths 900 each to a 600-by-600 comparison. However, applying these parameters to short sequences will lead to significant error. Figure 7 illustrates the effect of normalization for different lengths on the estimation of statistical significance.

6. PARAMETER ESTIMATION

By testing the validity of our assumptions we have presented two different ways of estimating γ and p. These methods estimate the parameters assuming a certain scoring scheme [$s(x, y)$ and a gap penalty function] and a certain letter distribution. For our examples we use PAM250 and $g(k) = 12 + 3k$.

The obvious method is to apply the algorithm in equation (2) many times to statistically independent sequences and to calculate the empirical distribution function of optimal alignment scores, that is, the fraction of alignments with score less than t. The Poisson clumping heuristic suggests that the probability for an alignment to score less than or equal to t is given by $\exp(-\gamma mnp^t)$. After appropriate transformation [$\log(-\log(\text{data}))$], the empirical distribution function is expected to form a straight line. In fact, linear regression gives a correlation coefficient above 0.99. It is then straightforward to estimate the parameters γ and p, and we call this method of derivation *direct estimation*.

Yet the true power of the theory sketched above comes to bear in the second method, which we call *declumping estimation*. Instead of many optimal alignments, from a few comparisons we calculate $H_{(1)}, H_{(2)}, \ldots, H_{(N)}$ using the declumping algorithm described above. The crucial observation is that the mean λ of the Poisson can be estimated from this data set as the average number of $H_{(i)}$ exceeding a threshold t. Based on the theory, these data can be fitted by a function of the form γmnp^t. Simulations show that plotting the empirical data on a logarithmic scale leads to an almost perfect straight line (Figure 6). Estimation of γ and p is then straightforward. As we will demonstrate, both approaches provide almost equally good estimates of statistical significance, thus supporting, by their agreement, the assumptions on which they are based.

For the direct estimation we usually run 1,000 sequence alignments before deriving γ and p. Given that each alignment takes time quadratic in the sequence length, this may take a very long time. The declumping estimation on the other hand can be done from 10 comparisons, collecting approximately 300 suboptimal solutions for each pair. For sequences of length 900 this computation can be done in 1.5 minutes on a Sun SPARC 10. This brings the estimation of the parameters for alignment significance into the realm of interactive computing. Declumping estimation, however, does not produce reliable results when done on short sequences. This probably is due to the fact that a small comparison matrix will soon be exhausted when taking out too many clumps, and independence between the clumps will be lost.

7. TESTING THE APPROXIMATION

Using the parameters derived by either of the above methods, we predict the distribution function of optimal alignment scores. To test the quality of the approximations given by our two methods, we derive the empirical distribution function from extensive simulations. First we tested the no-gap case. There the agreement between the empirical distribution function and either direct or declumping simulation is extremely good. There is hardly any difference throughout the range of the distribution function (not only in the tail). Both in terms of the agreement of the parameters and in overall approximation quality, our results are essentially the same as those obtained analytically by Karlin and Altschul (1990).

For Figure 7, parameters were derived from sequences of length 900 each. The rightmost group of three distribution functions shows declumping and direct estimation compared to the empirical distribution function. Notice that approximating the empirical distribution function by direct estimation amounts to fitting a double exponential. This in itself is not a test of the method. For the declumping estimation, however, the parameters for the approximation are derived from totally different data, and all three curves agree remarkably well. The other groups of curves in Figure 7 illustrate the quality of the normalization for length and thus prove our point with respect to both declumping and direct estimation. We normalized the parameters derived from a 900 × 900 comparison to approximate 600 × 600 and 300 × 300 comparisons. Only in the latter case is there some deviation between the empirical distribution function and the approximation.

These Poisson approximation methods also provide the approximate distributions of the suboptimal scores $H_{(1)} \geq H_{(2)} \geq \cdots \geq H_{(k)} \geq \cdots$ (Goldstein and Waterman, 1992):

$$\mathbb{P}(H_{(k)} \leq t) = \exp(-\gamma mnp^t) \sum_{j=0}^{k-1} \frac{(\gamma mnp^t)^j}{j!}.$$

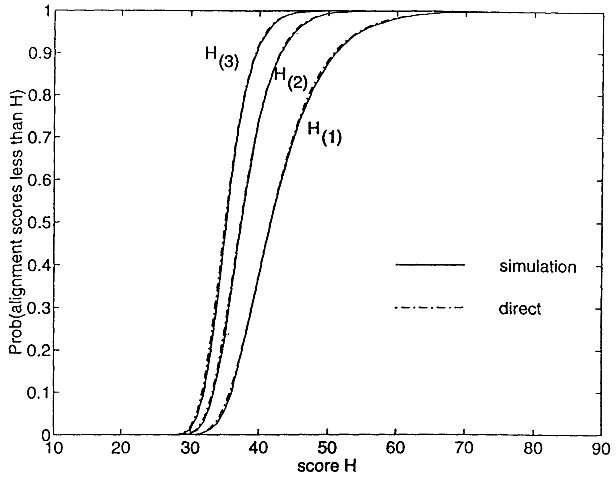

FIG. 8. *Estimates of the distribution function of the optimal $H_{(1)}$ and suboptimal ($H_{(2)}$ and $H_{(3)}$) scores.*

Figure 8 demonstrates the quality of approximating the significance of suboptimal solutions; γ and p were derived by direct simulation for sequences of length 300. The empirical distribution is based on 4,000 comparisons of random sequences, collecting the best, second-best and third-best solutions. The precision of the approximation is extraordinary.

Our assumptions are guided by the idea that in the logarithmic region the statistical behavior is essentially similar to the no-gap case. We therefore also want to show how the method fails in the linear region. It was pointed out before that direct estimation without length normalization is nothing but fitting an empirical distribution function with a double exponential, which may be deceptively easy. It is therefore not surprising to find that, even in the linear region, one can approximate an empirical distribution function using direct estimation. However, applying the length normalization, the approximation fails totally. Similarly, attempting declumping estimation in the linear regions, one quickly finds that the logarithm of the number of clumps above a threshold does not form a straight line, and it is impossible to fit a mean of the form $\hat{\gamma} p^t$.

8. DATABASE SEARCHES

Several new challenges arise when a query sequence is used to search a database. There are of course a wide variety of sequence lengths, families of closely related sequences and even duplicated sequences. Certainly, real protein sequences do not have iid letters. The model has been fitted using sets of sequences of identical length and with iid letters. The tests were made with other such sets, sometimes with length changed. While this is encouraging, it remains to test the model on real protein databases.

To remove the effects of duplicate sequences we use Newat, a protein database put together about 1986 by R. Doolittle (1981) to have one representative from each protein family. Most duplicate sequences have thus been removed, and the database has $N = 1{,}358$ sequences.

The most optimistic approach to our problem is to estimate (γ, p) from simulation of random sequences of length $n = 900$. However, when we compare our query sequence of length m with the N database sequences, we get score H_i for a sequence of length n_i, $i = 1, \ldots, N$. These H_i are not identical. Recall that

$$\mathbb{P}(H_i \le t) = \exp\big(-\gamma m n_i p^t\big),$$

so that the distribution functions of the various scores are not identical. Later this will be looked at again, but now just perform the probability integral transform:

$$T_i = \exp\big(-\gamma n_i m p^{H_i}\big)$$

and T_i is distributed as $U(0, 1)$. Ordering $T_{(1)} \ge T_{(2)} \ge \cdots \ge T_{(N)}$, we note that $\mathbb{E}(T_{(i)}) = i/(N+1)$. Denoting H_i^* as the score H in

$$T_{(i)} = \exp\big(-\gamma n_{(i)} m p^{H_i^*}\big),$$

we "expect"

$$\left(\frac{i}{N+1}, \exp\big(-\gamma n_{(i)} m p^{H_i^*}\big)\right)$$

to fall on an approximately straight line. To increase resolution we take a log-log transformation moving $\log n m_i$ to the left-hand side:

$$\left(-\log\left(-\log\left(\frac{i}{N+1}\right)\right)+\log n_{(i)}m, -\log\gamma - H_i^* \log p\right).$$

To illustrate this transformation, we take the declumping estimates from Figure 7 applied to iid sequences of length $n = 300$, there shown in cdf form, and apply this log-log transformation. The results are presented in Figure 9. In Figure 10, the data points come from the log-log transformation applied to scores obtained by comparing human alpha hemoglobin to the Newat database. In Figure 10a, γ and p are estimated from length $n = 900$ sequences by direct estimation, and in Figure 10b they are estimated by declumping estimation. Ideally they would cluster around the solid line drawn at 45°. Some outliers have been removed to make the difference more striking. The slope p looks about right and γ is too large. This gives conservative p-value estimates for a real database, which are actually quite good.

The observation that protein sequences do not have iid letters leads us to simulate sequences with the same first-order Markov statistics as the database sequences. In Figure 11 we see that the difference between our (p, γ) fit and the data is almost identical to that in Figure 10. The lack of sensitivity of score distribution on biological sequence higher-order dependencies was noted early (Smith, Burks and Waterman, 1985). There is an effect but it is numerically insignificant here.

This returns us to the central question about the above lack of fit. The length $900 = m = n$ sequences used to estimate p and γ are far longer than those used on most of our comparisons. Recall the Poisson mean $\lambda = \gamma m n_i p^t$. If we interpret $\gamma m n_i$ to be the area that clumps can be placed in, then it is plausible that shorter sequences have an effective area smaller than the factor γ would indicate. To test this idea, we simulated sequences of length 142 (that of alpha hemoglobin) and 350 (about the median database sequence length). The improved fit is shown in Figure 12.

Recall that in the Introduction the globin family of proteins was introduced. We used significance estimates derived from the last estimates of γ and p discussed above to evaluate the output of a PIR1 database search done with the sequence of human α hemoglobin. There were 25 sequences unrelated to globins that ranked higher in score than leghemoglobin, a distantly related plant globin. When instead the ranking is done according to estimated

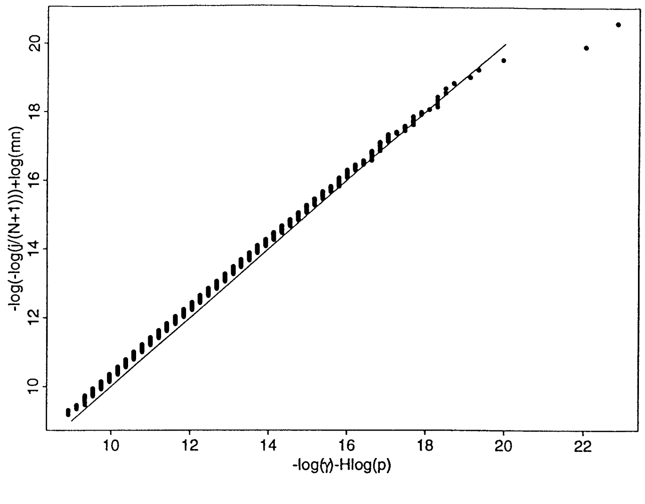

FIG. 9. *Declumping estimation* ($n = 900$) *fit to length* $n = 300$ *sequences.*

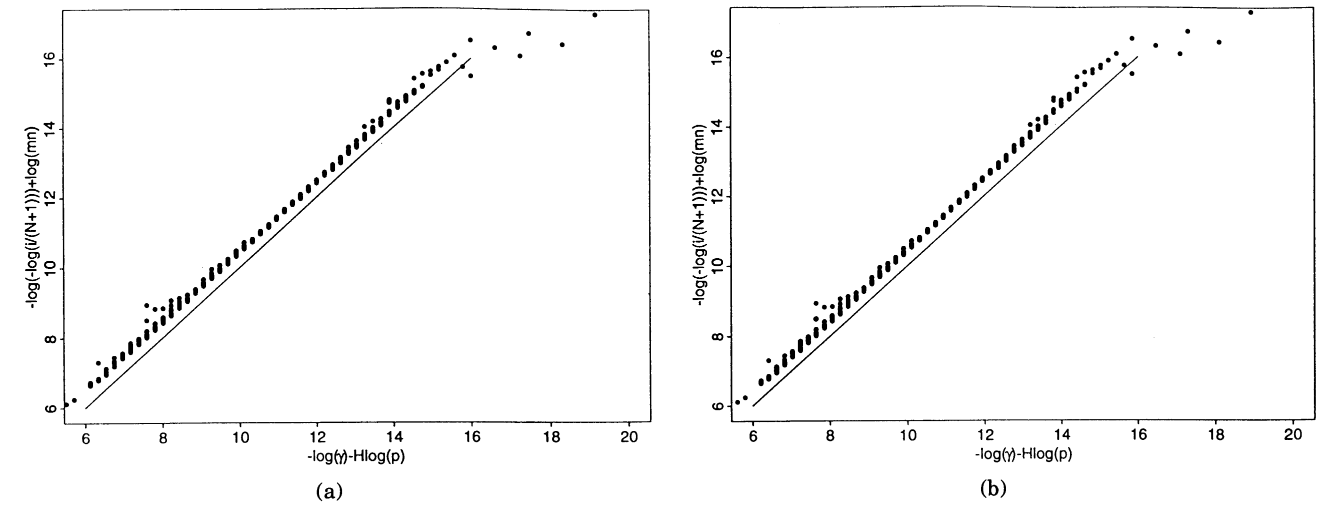

FIG. 10. *(a) Direct estimation with* $n = 900$; *(b) Declumping estimation with* $n = 900$.

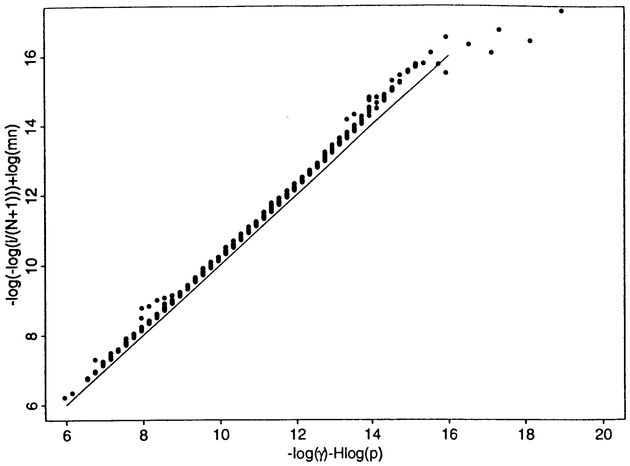

FIG. 11. *Estimation with Markov sequences.*

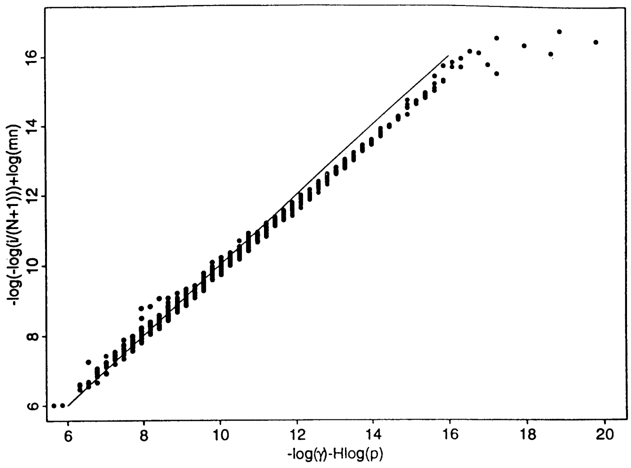

FIG. 12. *Estimation using* $m = 142$ *and* $n_i = 350$.

statistical significance which accounts for sequence lengths, only 10 nonglobins rank higher than leghemoglobins.

Notice that we have fit the distribution of database scores without looking at the data itself, but just using our model and the statistical (letter) composition of the database. It is worth looking at the problem of fitting γ and p from the results of a database search using maximum likelihood estimation (MLE). In Mott (1992) a four-parameter extreme value distribution is fit by MLE to the scores from a database search using a Smith–Waterman algorithm. The extreme value distribution was used earlier (Arratia, Gordon and Waterman, 1986) for sequence matching and is another side of Poisson approximation. Other early approaches to fitting database scores to estimate statistical significance appear in Smith, Burks and Waterman (1985); Coulson, Collins and Lyall (1987); and Collins and Coulson (1990).

Recall that, if the model is true,

$$\mathbb{P}(H_i \le t) = \exp(-\gamma m n_i p^t)$$

and has density function

$$f(t) = mn_i \log\left(\frac{1}{p}\right) p^t \exp(-\gamma m n_i p^t), \quad i = 1, \ldots, N.$$

We will assume the sequences are all independent. The likelihood of $H_1 \cdots H_N$ is

$$\begin{aligned}\mathbb{L} &= \prod_{i=1}^{N} \left(mn_i p^{H_i} \log\left(\frac{1}{p}\right) \exp(-\gamma m n_i p^{H_i}) \right) \\ &= \left(m \log\left(\frac{1}{p}\right)\right)^N \left(\prod_{i=1}^{N} n_i \right) \\ &\quad \cdot p^{\Sigma_{i=1}^N H_i} \exp\left(-\gamma m \sum_{i=1}^{N} n_i p^{H_i} \right)\end{aligned}$$

and

$$\log \mathbb{L} = \log(\gamma m)^N + N \log\log\left(\frac{1}{p}\right) + \sum_{i=1}^{N} \log n_i$$
$$+\left(\sum_{i=1}^{N} H_i\right)\log p - \gamma m \sum_{i=1}^{N} n_i p^{H_i}.$$

The derivative with respect to γ is

$$\frac{\partial \log \mathbb{L}}{\partial \gamma} = \frac{N}{\gamma} - m \sum_{i=1}^{N} n_i p^{H_i},$$

so $\partial \mathbb{L}/\partial\gamma = 0$ gives

$$\widehat{\gamma} = \frac{N}{m\sum_{i=1}^{N} n_i p^{H_i}}.$$

We will go on with the MLE, but let us apply this to the result of Figure 10. The value of p there looked good while γ was too large. Using that estimated p^*, we compute

$$\widehat{\gamma} = \frac{N}{m\sum_{i=1}^{N} n_i (p^*)^{H_i}}$$

and recompute the ordering $T_{(1)} \le \cdots \le T_{(N)}$, giving the results in Figure 13. This is very much better than both Figure 10 and Figure 12. Returning to MLE estimation,

$$\frac{\partial \log \mathbb{L}}{\partial p} = \frac{\sum_{i=1}^{N} H_i}{p} + \frac{N}{p\log p} - \gamma m \sum_{i=1}^{N} n_i \frac{H_i p^{H_i}}{p},$$

so $\partial \log \mathbb{L}/\partial p = 0$ implies

$$\sum_{i=1}^{N} H_i + \frac{N}{\log p} - \gamma m \sum_{i=1}^{N} n_i H_i p^{H_i} = 0, \tag{7}$$

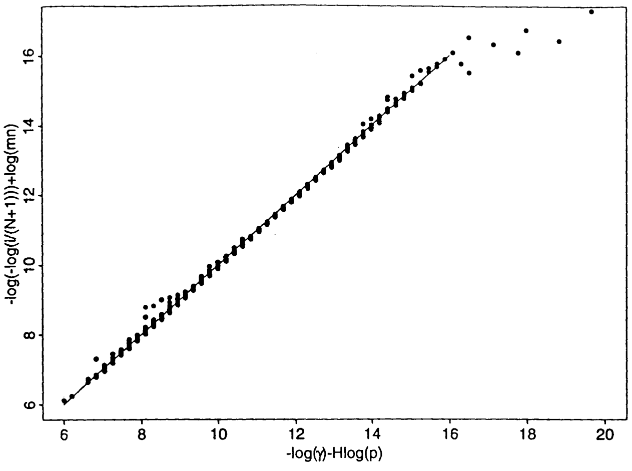

FIG. 13. *Correcting γ by MLE.*

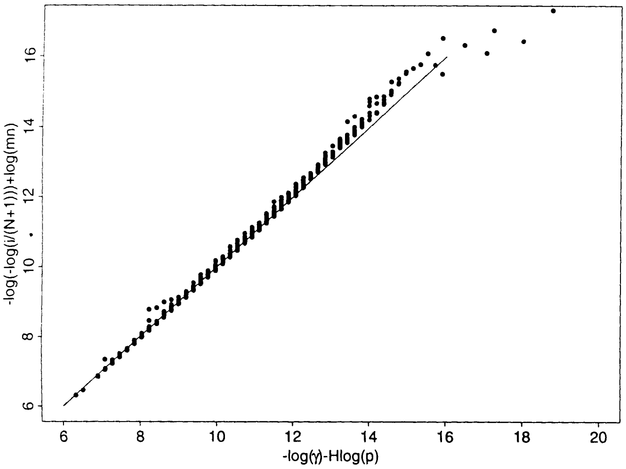

FIG. 14. *MLE estimated parameters.*

which with

$$\gamma - \frac{N}{\left(m\sum_{i=1}^{N} n_i p^{H_i}\right)} = 0 \tag{8}$$

comprise the MLE equations.

Solving the MLE equations gives γ and p for Figure 14. While Figure 13 appears to be a better fit, the likelihood $\mathbb{L}$ for Figure 13 is $\exp\{-9, 971\}$ and that for Figure 14 is $\exp\{-9, 953\}$.

9. DISCUSSION

A common practice for assigning statistical significance is by simulation. A random sample of scores is created by comparing pairs of random sequences. After computing the mean and standard deviation of the sample, an alignment score from comparing biological sequences is reported as the number of standard deviations above the mean. Essentially the alignment score is normalized to give a z-value so that there is an assignment of significance using the normal distribution. This is incorrect. The tails of the normal converge to 0 very rapidly (quadratic in the exponent) in comparison to the distribution function we study (linear in the exponent). This means that the normal assumption will give p-values that are too small.

In our section on database searches we looked at the collection of p-values $1 - \exp(-\gamma m n_i p^{H_i})$, one p-value for each sequence comparison. Of course this is a test of N hypotheses, and in the larger context of the search no individual p-value is correct. Instead we feel it is appropriate to rank the importance of an alignment score by the p-values since matches with long sequences can yield larger scores simply due to sequence length. The other alternative is to consider the database one sequence of length $\sum_{i=1}^{N} n_i$ and compute a p-value for matching the search sequence with this long artificial sequence. This amounts to ranking by score size without considering sequence length n_i. This in fact is what Karlin and Altschul recommend in their treatment of the no-gap case. This practice is conservative but of less use in evaluating those important cases on the boundary of statistical significance. This list of matches ranked by individual p-values is often different from ranking by score and, we feel, more biologically informative.

We approached estimation of γ and p in two distinct ways. Both the direct and declumping estimates use simulated sequences to estimate γ and p. Then the estimated γ and p are applied to the results of a database search. In contrast, MLE uses the set of scores from a database search to obtain the estimates of the parameters γ and p. It is perhaps remarkable that these two approaches are in such good agreement.

The database Newat that we used has one representative of each sequence family, in contrast with the usual protein databanks. MLE could be degraded by having multiple members (not independent) of a family. Also, it is likely that within a database, sets of independent sequences exist with different values of p. Understanding of these topics could profit from further investigation.

10. PROGRAMS

Programs for local alignment and p-value estimation can be obtained by anonymous ftp from hto-e.usc.edu.

ACKNOWLEDGMENTS

This research was supported by grants from the National Institute of Health and the National Science Foundation. Most of the research was carried out when the second author was a postdoctoral fellow at the University of Southern California.

REFERENCES

ALDOUS, D. (1989). *Probability Approximations via the Poisson Clumping Heuristic*. Springer, New York.

ALTSCHUL, S. F., GISH, W., MILLER, W., MEYERS, E. W. and LIPMAN, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology* **215** 403.

ARRATIA, R., GOLDSTEIN, L. and GORDON, L. (1989). Two moments suffice for Poisson approximations: the Chen–Stein method. *Ann. Probab.* **17** 9–25.

ARRATIA, R., GORDON, L. and WATERMAN, M. S. (1986). An extreme value distribution for sequence matching. *Ann. Statist.* **14** 971–993.

ARRATIA, R., GORDON, L. and WATERMAN, M. S. (1990). The Erdös-Rényi Law in distribution, for coin tossing and sequence matching. *Ann. Statist.* **18** 539–570.

ARRATIA, R., MORRIS, P. and WATERMAN, M. S. (1988). Stochastic Scrabble: a law of large numbers for sequence matching with scores. *J. Appl. Probab.* **25** 106–119.

ARRATIA, R. and WATERMAN, M. S. (1994). A phase transition for the score in matching random sequences allowing deletions. *Ann. Appl. Probab.* **4** 200–225.

BARKER, W. C. and DAYHOFF, M. O. (1982). Viral SRC gene products are related to the catalytic chain of mammalian cAMP-dependent protein kinase. *Proc. Nat. Acad. Sci. U.S.A.* **79** 2836.

CHVÁTAL, V. and SANKOFF, D. (1975). Longest common subsequences of two random sequences. *J. Appl. Probab.* **12** 306–315.

COLLINS, J. F. and COULSON, A. F. W. (1990). Significance of protein sequence similarities. *Methods in Enzymology* **183** 474.

COULSON, A. F. W., COLLINS, J. F. and LYALL, A. (1987). Protein and nucleic acid sequence database searching: a suitable case for parallel processing. *Comput. J.* **30** 420.

DAYHOFF, M. O., BARKER, W. C. and HUNT, L. T. (1983). Establishing homologies in protein sequences. *Methods in Enzymology* **91** 524.

DOOLITTLE R. F. (1981). Similar amino acid sequences: chance or common ancestry? *Science* **214** 149–159.

DOOLITTLE, R. F., HUNKAPILLER, M. W., HOOD, L. E., DEVARE, S. G., ET AL. (1983). Simian sarcoma virus onc gene, v-sis, is derived from the gene (or genes) encoding a platelet-derived growth factor. *Science* **221** 275.

GOLDSTEIN, L. and WATERMAN, M. S. (1992). Poisson, compound Poisson and process approximations for testing statistical significance in sequence comparisons. *Bull. Math. Biol.* **54** 785–812.

GOTOH, O. (1982). An improved algorithm for matching biological sequences. *Journal of Molecular Biology* **162** 705.

KARLIN, S. and ALTSCHUL, S. F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Nat. Acad. Sci. U.S.A.* **87** 2264–2268.

KINGMAN, J. F. C. (1973). Subadditive ergodic theory. *Ann. Probab.* **1** 883–909.

LIPMAN, D. J. and PEARSON, W. R. (1985). Rapid and sensitive protein similarity searches. *Science* **227** 1435.

MCCALDON, P. and ARGOS, P. (1988). Oligopeptide biases in protein sequences and their use in predicting protein coding regions in nucleotide sequences. *PROTEINS: Structure, Function, and Genetics* **4** 99.

MOTT, R. F. (1992). Maximum likelihood estimation of the statistical distribution of Smith–Waterman local sequence similarities. *Bull. Math. Biol.* **54** 59.

NEUHAUSER, C. (1994). A Poisson approximation for sequence comparisons with insertions and deletions. *Ann. Statist.* **22**. To appear.

PEARSON, W. R. (1991). Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith–Waterman and FASTA algorithms. *Genomics* **11** 635.

RIORDAN, J. R., ROMMENS, J. M., KEREM, B., ALON, N., ET AL. (1989). Identification of the cystic fibrosis gene: cloning and characterization of complementary DNA. *Science* **245** 1066.

SMITH, T. F., BURKS, C. and WATERMAN, M. S. (1985). The statistical distribution of nucleic acid similarities. *Nucleic Acids Research* **13** 645–656.

SMITH, T. F. and WATERMAN, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology* **147** 195–197.

VINGRON, M. and WATERMAN, M. S. (1994). Sequence alignment and penalty choice. Review of concepts, case studies and implications. *Journal of Molecular Biology* **235** 1–12.

WATERMAN, M. S. (1984). General methods of sequence comparison. *Bull. Math. Biol.* **46** 473–500.

WATERMAN, M. S. and EGGERT, M. (1987). A new algorithm for best subsequence alignments with application to tRNA–rRNA comparisons. *Journal of Molecular Biology* **197** 723–728.

WILBUR, W. J. and LIPMAN, D. J. (1983). Rapid similarity searches of nucleic acid and protein data banks. *Proc. Nat. Acad. Sci. U.S.A.* **80** 726.