JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 6, Number 2, 1999
Mary Ann Liebert, Inc.
Pp. 219–235

Minimal-Risk Scoring Matrices for Sequence Analysis

THOMAS D. WU, CRAIG G. NEVILL-MANNING,* and DOUGLAS L. BRUTLAG

ABSTRACT

We introduce a minimal-risk method for estimating the frequencies of amino acids at conserved positions in a protein family. Our method, called minimal-risk estimation, finds the optimal weighting between a set of observed amino acid counts and a set of pseudofrequencies, which represent prior information about the frequencies. We compute the optimal weighting by minimizing the expected distance between the estimated frequencies and the true population frequencies, measured by either a squared-error or a relative-entropy metric. Our method accounts for the source of the pseudofrequencies, which arise either from the background distribution of amino acids or from applying a substitution matrix to the observed data. Our frequency estimates therefore depend on the size and composition of the observed data as well as the source of the pseudofrequencies. We convert our frequency estimates into minimal-risk scoring matrices for sequence analysis. A large-scale cross-validation study, involving 48 variants of seven methods, shows that the best performing method is minimal-risk estimation using the squared-error metric. Our method is implemented in the package EMATRIX, which is available on the Internet at http://motif.stanford.edu/ematrix.

Key words: frequency estimates, Hidden Markov models, position-specific scoring matrices, profiles, protein families, pseudocounts, sequence analysis

1. INTRODUCTION

MODERN RESEARCH in molecular biology depends heavily upon computer-based techniques to analyze and characterize new sequences. For example, one well-established technique is similarity search, exemplified by FASTA (Pearson and Lipman, 1988) or BLAST (Altschul *et al.*, 1990), which compares an entire query sequence against target sequences in a database. More recently, sequence analysis techniques have been developed based on pattern matching. In this approach, targets are relatively short patterns that represent conserved regions, or alignment blocks, that characterize a particular protein family. Therefore, a query sequence may be identified by matching one of its segments against a database of alignment blocks. Examples of such databases include BLOCKS (Henikoff and Henikoff, 1991), PRINTS (Attwood and Beck, 1994), and PIMA (Worley *et al.*, 1995).

However, for the purposes of matching in sequence analysis, we must represent each alignment block as a symbolic or numerical pattern. Symbolic patterns, which we will not cover in this paper, consist of regular expressions or discrete motifs. Examples of motif databases are PROSITE (Bairoch, 1992) and IDENTIFY (Nevill-Manning *et al.*, 1990). Numerical patterns on the other hand, consist of a matrix of probability estimates or scores. Each position in the alignment block is represented by a column of frequencies or scores in the

---

Department of Biochemistry, Stanford University School of Medicine, Stanford, California.
*Current address: Department of Computer Science, Rutgers University, Piscataway, NJ.

matrix. There are several examples of numerical representations, including weight matrices (Stormo and Hartzell, 1989; Staden, 1990), profiles (Gribskov *et al.*, 1987), position-specific scoring matrices (Henikoff, 1996), hidden Markov models (Krogh *et al.*, 1994; Eddy, 1996), and Gibbs sampling (Lawrence *et al.*, 1993). Numerical representations are closely related, and one form may often be converted into another. Therefore, we will use the term "scoring matrix" to denote a generic numerical representation, although our results generalize to other numerical representations.

The basic problem faced by most numerical methods is how to convert a vector of observed amino acid counts into a vector of frequencies or a vector of scores. The set of observed counts is finite and almost always contains zero counts for one or more amino acids. However, zero frequencies are undesirable for sequence analysis, because they may exclude true but unusual members of a given family. Therefore, we must introduce some methodology to augment the finite set of observed counts and to estimate the true population frequencies. Essentially, the problem is one of frequency estimation, and the predominant method has been to add pseudocounts, or artificial counts, to the observed counts.

However, the Bayesian theory underlying the pseudocount approach leaves open the question of how many pseudocounts to add, or equivalently, how much weight to assign to the prior information. This open question has spawned many different methods for combining pseudocounts with observed counts. Existing methods are based largely on intuitive or empirical grounds, which appears to suggest that no optimal method exists. In this paper, however, we show how optimality may be introduced to solve the problem.

Specifically, we define optimality using a criterion called risk, which is the expected distance between the estimated frequencies and the true population frequencies. Our approach, called *minimal-risk estimation* computes an optimal weighting between the prior information and the observed counts. In short, it computes the optimal number of pseudocounts to add. The resulting frequency estimates may then be converted to scoring matrices in the usual log-odds manner, resulting in *minimal-risk scoring matrices*, or MRSMs.

Our method models the source of the pseudofrequencies, and provides different formulas for different sources. Pseudofrequencies represent domain knowledge either about the background distribution of amino acids (e.g., that leucine is generally more common than cysteine) or about similarities and dissimilarities between amino acids (e.g., that isoleucine and valine are chemically similar and likely to substitute for each other). We call these two sources *background pseudofrequencies* and *substitution pseudofrequencies*, respectively. Estimation using background and substitution pseudofrequencies is also called *Bayesian* and *empirical Bayesian* estimation. Whereas background pseudocounts represent prior knowledge that is the same for all possible observations, substitution pseudocounts depend on the observed data, and they have therefore been called *data-dependent pseudocounts* by Tatusov *et al.* (1994).

Another consideration in our method is the measurement of risk, which is essentially a distance measurement. We consider two metrics, one based on squared-error distance and one based on relative-entropy distance. Thus, we present a total of four solutions in this paper, corresponding to the two metrics for risk and the two sources of pseudofrequencies. However, we show empirically that the best solution appears to combine the squared-error metric with substitution pseudofrequencies.

Our minimal-risk technique is grounded in statistical decision theory, and follows closely a statistical approach developed for estimating cell frequencies in contingency tables (Bishop *et al.*, 1975). However, the solution for contingency tables corresponds to only one of our cases: a squared-error metric with background pseudofrequencies. Accordingly, in this paper, we extend statistical decision theory to handle the relative-entropy metric and substitution pseudofrequencies, which are fundamental in computational biology.

In contrast with existing approaches, which have been developed largely through empirical trial and error, our theoretical approach attempts to achieve optimal accuracy in a wide range of situations. Our method does not require training on an existing database, nor is it fine-tuned to perform well for a particular substitution matrix. In this paper, we show that our approach does indeed yield superior results. We examine 48 different variations of seven methods for constructing scoring matrices, including Dirichlet-mixture (Brown *et al.*, 1993; Sjölander *et al.*, 1996) and average-score (Gribskov *et al.*, 1987) methods, using a large cross-validation test. Our experiment shows that the most accurate prediction method is provided by minimal-risk scoring matrices.

2. METHODS

2.1. Pseudocount estimation

In the estimation problem, we are given a vector $\mathbf{C}$ of observed counts. For our problem, which involves amino acids, this vector is of length 20. We assume that $\mathbf{C}$ is generated by true population frequencies $\mathbf{p}$, which are hidden from us and cannot be known. We wish to produce a vector $\mathbf{p}^*$ of frequency estimates, which should approximate $\mathbf{p}$ as well as possible.

In the pseudocount approach, we introduce domain knowledge in the form of a vector $\boldsymbol{\lambda}$ of pseudofrequencies. As we discussed previously, we may use either background pseudofrequencies, represented by $\mathbf{q}$, or substitution pseudofrequencies, represented by $\mathbf{r}$. The observed counts and pseudofrequencies may be combined by adding them in some proportion. The number of observed counts is $N = \sum_{j=1}^{20} C_j$. The number of pseudocounts is B, a quantity that must be specified by a particular pseudocount method. The result is the pseudocount frequency estimate

$$p_j^*(B, \boldsymbol{\lambda}) = \frac{C_j}{N+B} + \frac{B\lambda_j}{N+B}, \qquad j = 1, \ldots, 20 \tag{1}$$

Equivalently, we may express the problem in terms of weights α and β:

$$p_j^*(\beta, \boldsymbol{\lambda}) = \alpha \hat{p}_j + \beta\lambda_j, \qquad \alpha + \beta = 1 \tag{2}$$

In this representation, we normalize the observed counts to give observed frequencies $\hat{\mathbf{p}}$, where $\hat{p}_j = C_j/N$. The two representations are essentially equivalent, as expressed by the relationship

$$\alpha = \frac{N}{N+B} \qquad \beta = \frac{B}{N+B} \tag{3}$$

However, in this paper, we will find the weight formulation in Equation 2 to be more suitable than the count formulations in Equation.

Background pseudofrequencies $\mathbf{q}$ may be obtained from the distribution of amino acids in a large sequence database. Such pseudofrequencies are independent of the observed counts. In contrast, substitution pseudofrequencies $\mathbf{r}$ depend very much on the observed data. Similarity information is represented by a substitution matrix $\mathbf{M}$, such as the PAM (Dayhoff *et al.*, 1978), MDM (Jones *et al.*, 1992), Gonnet (1992), or BLOSUM (Henikoff and Henikoff, 1992) series of matrices. Entry M_{jk} in the matrix represents the conditional probability of seeing amino acid j given amino acid k. Therefore, we compute substitution pseudofrequencies from the observed frequencies $\hat{\mathbf{p}}$ by matrix multiplication:

$$r_j = \sum_{k=1}^{20} M_{jk}\hat{p}_k = \sum_{k=1}^{20} M_{jk}C_k/N \tag{4}$$

The end result is to give higher pseudofrequencies to those amino acids similar to the observed ones, and lower frequencies to dissimilar amino acids.

2.2. *Minimal-risk estimation*

In minimal-risk estimation, our primary goal is to derive a relationship between the true frequencies $\mathbf{p}$, which are constant for a particular problem, and the optimal weight β^*, where the asterisk denotes optimality. Although the true frequencies are actually unknown to the estimator, we wish nevertheless to derive the ideal relationship based on the true $\mathbf{p}$. The vector of observed counts is a set of random variables. Each vector $\mathbf{C}$ occurs with probability according to a multinomial model:

$$\Pr(\mathbf{C}) = \binom{N}{C_1 C_2 \cdots C_{20}} p_1^{C_1} p_2^{C_2} \cdots p_{20}^{C_{20}} = \frac{\Gamma(N+1)}{\prod_{j=1}^{20} \Gamma(C_j+1)} \prod_{j=1}^{20} p_j^{C_j} \tag{5}$$

Let us view the situation geometrically, as in Fig. 1A. The position of the true frequencies $\mathbf{p}$ is fixed. The position of the observed frequencies $\hat{\mathbf{p}}$ is random, according to the multinomial model. In the pseudocount method, each vector of observed counts has a corresponding a vector $\boldsymbol{\lambda}$ of pseudofrequencies. For background pseudofrequencies, this position is constant; for substitution pseudofrequencies, it varies with the observed data and is therefore also random. The pseudocount method constrains the solution to lie along the line that connects $\hat{\mathbf{p}}$ and $\boldsymbol{\lambda}$. For any particular instance of $\hat{\mathbf{p}}$ and $\boldsymbol{\lambda}$, we advocate choosing the parameter value that minimizes the distance, or loss, to the true frequencies $\mathbf{p}$. That is, we should minimize

$$L(\mathbf{p}^*, \mathbf{p}) = \|\mathbf{p}^* - \mathbf{p}\|^2 = \begin{cases} \displaystyle\sum_{j=1}^{20} \left(p_j^* - p_j\right)^2 & \text{[Squared error]} \\ \displaystyle\sum_{j=1}^{20} p_j^* \log\left(\frac{p_j^*}{p_j}\right) & \text{[Relative entropy]} \end{cases} \tag{6}$$

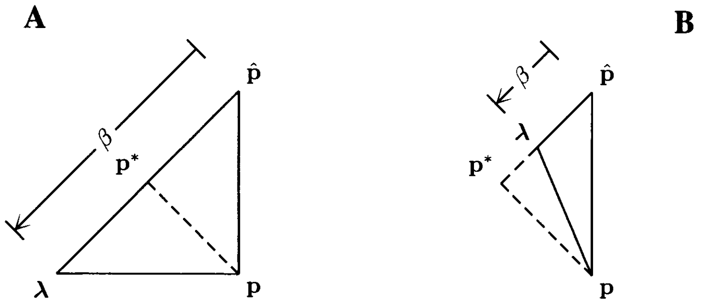

FIG. 1. Geometric interpretation of statistical decision theory. The goal is to find the value of $\mathbf{p}^*$ that minimizes the expected distance to the true frequencies $\mathbf{p}$. The estimate $\mathbf{p}^*$ is obtained by weighting the pseudofrequencies $\boldsymbol{\lambda}$ by β^* and the observed frequencies $\hat{\mathbf{p}}$ by $(1 - \beta^*)$. **(A)** Relationships when pseudofrequencies are distant from the observed frequencies. **(B)** Relationships when the pseudofrequencies are close to the observed frequencies; β^* may exceed 1 in order to minimize risk. (Figure 1A is adapted from Bishop *et al.*, 1975.)

Here, we have measured the distance using two metrics: squared error and relative entropy. We will develop and test a theory for each metric, although the squared-error metric will eventually prove to work better for sequence analysis.

The loss is a random variable that varies according to the distribution of observed counts and pseudofrequencies. Therefore, we can consider the expected loss, or risk, over all possible vectors of observed counts:

$$R = E[L(\mathbf{p}^*, \mathbf{p})] = \begin{cases} \sum_{j=1}^{20} E\left(p_j^* - p_j\right)^2 & \text{[Squared error]} \\ \sum_{j=1}^{20} E\left(p_j^* \log\left(\frac{p_j^*}{p_j}\right)\right) & \text{[Relative entropy]} \end{cases} \tag{7}$$

In this equation, we have used the fact that expectation is a linear operator, so the expectation of a sum is equal to the sum of expectations.

We may now establish a relationship between the true frequencies $\mathbf{p}$ and the optimal weight β^*. We advocate choosing the weight β^* to minimize risk, or expected loss. We have four cases to consider, corresponding to the two metrics for loss and the two sources of pseudofrequencies. Solutions for the squared-error metric, which we derive in the Appendix, can be expressed in closed form:

$$\beta^* = \begin{cases} \dfrac{1 - \sum_{j=1}^{20} p_j^2}{1 - \sum_{j=1}^{20} p_j^2 + N \sum_{j=1}^{20} (p_j - q_j)^2} & \text{[Background]} \\[2ex] \dfrac{1 - \sum_{j=1}^{20} [M_{jj} p_j + p_j (p_j - s_j)]}{1 + \sum_{j=1}^{20} \left[(N-1)(p_j - s_j)^2 - 2 M_{jj} p_j + \sum_{j=1}^{20} M_{jk}^2 p_k \right]} & \text{[Substitution]} \end{cases} \tag{8}$$

where

$$s_j = \sum_{k=1}^{20} M_{jk} p_k \tag{9}$$

In contrast, solutions for the relative-entropy must be solved numerically; details are given in the Appendix. The solution for background pseudofrequencies in Equation 8 is equivalent that obtained by Bishop *et al.* (1975), whereas the solution for substitution pseudofrequencies is a new result.

These relationships are ideal, because they depend on knowing the true frequencies $\mathbf{p}$, which are unknown. To obtain frequency estimates for a particular set of data, we make use of a plug-in statistical technique (Bishop *et al.*, 1975). We make an initial estimate for $\mathbf{p}$ as a starting point, and then use the optimal weighting to obtain a better estimate. For our initial estimate of $\mathbf{p}$, we choose $B = \sqrt{N}$ pseudocounts, or

$$\mathbf{p}' = \frac{\mathbf{C}}{N + \sqrt{N}} + \frac{\sqrt{N}\boldsymbol{\lambda}}{N + \sqrt{N}} \tag{10}$$

Our starting point is the estimate used in Gibbs sampling (Lawrence *et al.*, 1993), and has been found to work well empirically. Incidentally, this starting point corresponds to the minimax solution when the pseudofrequencies are uniform (Trybula, 1958); that is, $\lambda_j = 0.05$ for all j. Note that our initial estimate accounts for the size of the observed sample, but not its composition or the composition of the pseudofrequencies. Our method is able to improve upon the initial estimate by accounting for these factors.

The plug-in technique acts as a regularizer, by improving our estimate of $\mathbf{p}^*$ from a well-motivated starting point. However, one might reason that the new estimate $\mathbf{p}^*$ could serve as a new starting point, and that we could plug in this value into our minimal-risk equations to obtain a revised estimate. Iteration in this fashion converges to a fixed point, which constitutes a self-consistent estimator (Tarpey and Flury, 1996). Self-consistent estimation is closely related to many statistical approaches, including expectation maximization (Dempster *et al.*, 1977).

However, a self-consistent estimator may not be suitable for our particular task, because data is often scarce. When the number of observed data points is small, an iterative approach can lead to progressive overfitting and poor estimates. The behavior of iterative estimation is demonstrated in Fig. 2, for both small and large data sets. When a small data set contains no clear pattern, β diverges progressively to 1, indicating that the observed data should be ignored and the background frequency should be used instead.

Although self-consistent estimation might seem reasonable for a single estimation problem, our experience shows that it can generate poor scoring matrices for sequence analysis. A scoring matrix requires solution of several estimation problems of the same size, because each position in the block has the same number of observations. For small blocks in which progressive overfitting occurs, self-consistency produces scoring matrices that are almost discrete: they ignore the observed data in most positions and allow only observed

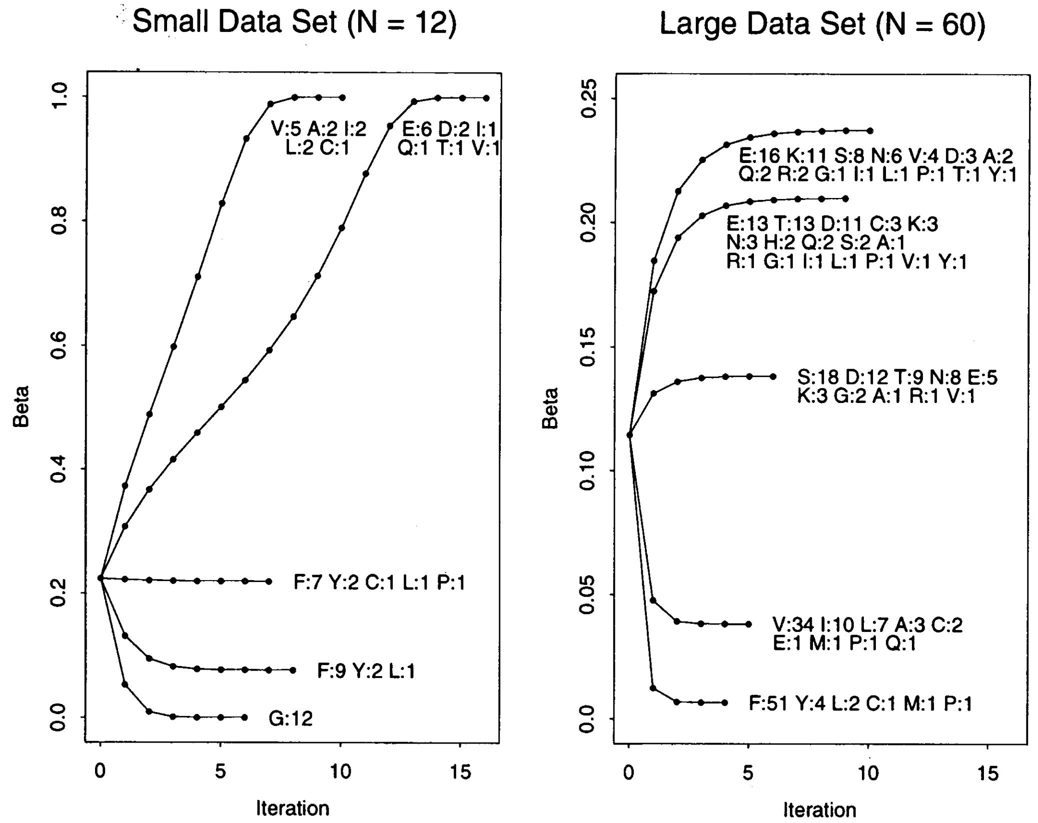

FIG. 2. Behavior of iterative minimal-risk estimation. The left graph shows the behavior for small data sets (12 observations); the right graph, on a different vertical scale, shows the behavior for large data sets (60 observations). Each graph contains several curves, each representing a particular set of observations, showing how the estimate for β changes with each iteration. The initial value of β at iteration 0 is the square-root estimate, $\sqrt{N}/(N + \sqrt{N})$. Iteration continues until the parameter β converges, with a tolerance of 0.0001. The method used was the unweighted squared-error method with background pseudofrequencies. Data sets were drawn from various columns of the first 12 and 60 sequences of block 30B (RNP_1) of the BLOCKS database. For small data sets, iterative estimation sometimes leads to progressive overfitting. For large data sets, a single step obtains an estimate that is close to the iterative estimate.

amino acids in other positions. Because the progressive overfitting problem occurs typically in small blocks, self-consistent estimates often generate scoring matrices that are unsuitable for sequence analysis. On the other hand, the single-step estimate not only avoids the overfitting problem for small blocks but also approaches the self-consistent estimate for large blocks, as Fig. 2 shows.

Figure 2 also demonstrates how our procedure treats the amount of information in observed data. When the observed data has a strong pattern, such as the data set containing 51 counts of phenylalanine, our estimate for β is relatively small, signifying a relatively strong emphasis on the observed data. When the observed data has a weaker pattern, such as the data set containing 16 counts of glutamate and 11 counts of lysine, our estimate for β is relatively large, signifying a relatively strong emphasis on the background frequency. Our procedure also accounts for the size of the observed data, as seen by the generally lower values of β for large data sets compared with those for small data sets.

2.3. *Existing methods*

Several other methods have been proposed for choosing the single parameter in the pseudocount approach, including the constant, square-root, and step-function methods. The constant and step-function methods have been developed empirically, whereas the square-root method has a theoretical basis as a minimax solution, as mentioned in the previous section. These methods use the count formulation in Equation 1 and therefore select the number B of pseudocounts.

The constant method (Henikoff *et al.*, 1995) sets B to be a large constant number. Empirically, the value 50 has been found to work well. Therefore, when N is less than 50, pseudofrequencies will have greater weight; when N is greater than 50, the observed frequencies will have greater weight.

The square-root method (Lawrence *et al.*, 1993; Tatusov *et al.*, 1994) sets B to be equal to $\sqrt{N}$. In the square-root method, the weight of the pseudofrequencies is smaller than that of the observed frequencies for $N > 1$.

The step-function method (Henikoff and Henikoff, 1996) sets B to be proportional to the number of unique amino acid residues observed. In other words, if the count vector contains A non-zero values, then empirical results suggest setting B to be equal to $5A$. We refer to this method as the step-function method, because the value of B increases by step intervals of 5, from 5 through 100.

Note that the constant and square-root methods depend solely on the size N of the sample. In contrast, the step-function method depends also on the composition of the sample. The step-function method assigns greater weight count vectors that are concentrated in one or a few amino acids, and assigns less weight to count vectors that are distributed over several amino acids. Our minimal-risk solution also depends on the composition of the sample, but with more resolution than the step-function method. Our solution considers not just the number of different amino acids represented, but also their relative frequencies.

In addition to evaluating these three single-parameter methods, we also evaluate an estimation method based on a Dirichlet mixture model (Sjölander *et al.*, 1996). Dirichlet mixture models estimate frequencies by combining several component, or prior, distributions; nine components have been found to work well empirically. These prior distributions are obtained by training extensively on a large database of protein families. Given a new set of observed counts, the Dirichlet-mixture method combines each prior distribution according to the likelihood that it could generate the observed counts. Therefore, the Dirichlet-mixture method estimates multiple parameters, one for each component.

The methods we have described so far all estimate population frequencies, rather than scores per se. Ultimately, though, each column of a scoring matrix consists of a vector **S** of scores. There is general consensus that the scores can be derived from frequencies using a log-odds relationship (Berg and von Hippel, 1987; Stormo and Hartzell, 1989):

$$S_j = \log\left(p_j^*/q_j\right) \tag{11}$$

where q_j is the background frequency for amino acid j.

One exception to this frequency estimate approach is the *average-score* or *profile* method (Gribskov *et al.*, 1987). This method multiplies the observed frequencies by a substitution score matrix **T**, as follows:

$$S_j = \sum_{k=1}^{20} T_{jk}\hat{p}_k$$

The substitution score matrix **T** is typically derived from a substitution matrix **M** by a log-odds relationship.

Each of the frequency estimate methods requires pseudofrequencies, which may be either background or substitution pseudofrequencies. Therefore, we can speak of a constant background-pseudofrequency method and a constant substitution-pseudofrequency method. Moreover, when substitution pseudofrequencies are involved, the particular substitution matrix must be specified. Hence, we can speak of a square-root BLOSUM 30, a square-root BLOSUM 62, and a square-root BLOSUM 100 method. The average-score method, which does not use pseudofrequencies, nevertheless also requires specification of a substitution score matrix. Therefore, each of the methods can be implemented as several different variants.

2.4. *Weighting scatterplots*

Single-parameter pseudocount methods differ in their relative weighting of the observed data and prior information. We illustrate the differences among these methods graphically using *weighting scatterplots*. In general, as shown in Equation 3, a major determinant of β is the size N of the observed data, with β decreasing as N increases. A weighting scatterplot plots these two parameters against each other. We compute a scatterplot empirically by applying a pseudocount method to several alignment blocks with varying values of N.

Weighting scatterplots for the constant, square-root, and step-function methods are shown in the first row of Fig. 3. The constant and square-root methods select B, and hence β, solely as a function of N. Therefore, the points on these two scatterplots lie on a single curved line.

In contrast, the step-function method shows 20 discrete curves, each corresponding to one of the 20 choices for B. Therefore, the step function yields up to 20 different values of β for a given value of N. This variability

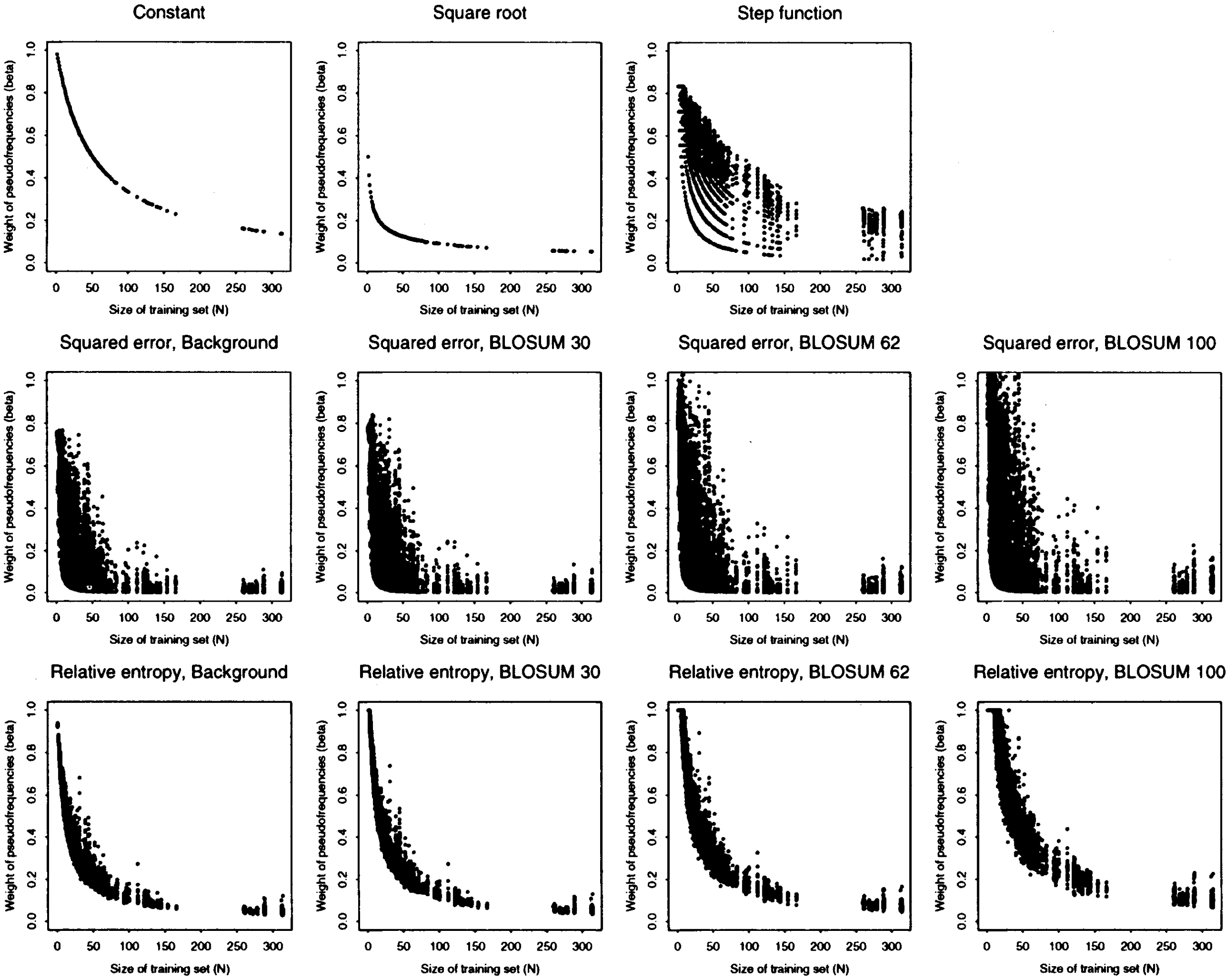
Constant
Square root
Step function
Squared error, Background
Squared error, BLOSUM 30
Squared error, BLOSUM 62
Squared error, BLOSUM 100
Relative entropy, Background
Relative entropy, BLOSUM 30
Relative entropy, BLOSUM 62
Relative entropy, BLOSUM 100

FIG. 3. Weighting scatterplots for different pseudocount methods. Each curve plots the value of β against the size of the sample. The top row has scatterplots for the constant, square root, and step function methods. The middle row has scatterplots for the squared-error method, for various choices of pseudofrequencies. From left to right, these are background pseudofrequencies, followed by substitution pseudofrequencies using the BLOSUM 30, 62, and 100 matrices. The bottom row has scatterplots for the relative-entropy method, for the same choices of pseudofrequencies.

allows the step function method to consider not only the size, but also the approximate composition, of the observed counts.

The remainder of Fig. 3 shows weighting scatterplots for the minimal-risk method over two loss metrics and a variety of pseudofrequencies. The scatterplots show that the minimal-risk method selects widely varying values of β for a given value of N. The fine variability within each graph, at each value of N, shows how the minimal-risk method accounts for the exact composition of the observed counts.

Furthermore, the minimal-risk method shows variability *across* graphs, showing how it models the different sources of pseudofrequencies. As the substitution matrix progresses from greater evolutionary distance (BLOSUM 30) to smaller evolutionary distance (BLOSUM 100), the weight given to the pseudofrequencies generally increases.

In some cases, especially for the BLOSUM 100 matrix, the value for β even exceeds 1. The reason for this can be seen in Fig. 1B. Note that the distance β is a *relative* distance, defined by the endpoints $\hat{\mathbf{p}}$ and $\boldsymbol{\lambda}$. If the two endpoints are close together, then β has a different scale than if the two endpoints are far apart. The distance between the endpoints is determined by the substitution matrix $\mathbf{M}$. For smaller evolutionary distances, the pseudofrequencies $\boldsymbol{\lambda}$ are close to the observed frequencies $\hat{\mathbf{p}}$, and β^* must increase to minimize risk. In extreme cases, the minimal-risk method chooses β^* to be greater than 1, indicating that it extrapolates beyond the given pseudofrequencies to minimize expected loss. Incidentally, the ability to choose such values for β is largely the motivation for our use of the weight formulation in Equation 2 rather than the count formulation in Equation 1.

3. RESULTS

3.1. Cross-validation test

To evaluate our method and compare it with existing methods for constructing scoring matrices, we developed a large-scale cross-validation test that measures how accurately a method predicts other members of a given family. The large scale is important to achieve comparisons that are statistically significant. We used the BLOCKS 9.3 (March 1997) database (Henikoff and Henikoff, 1991) as a source of protein families. This database contains 3417 alignment blocks that have been used widely in bioinformatics research.

We developed a two-fold cross-validation test. Each block was divided randomly into two parts. For half of the test, the first part served as a training set and the second part served as a test set. For the other half of the test, the two parts swapped roles as training and test sets. We tested each method by using the training set as the basis for a scoring matrix, and then using the matrix to identify other members of the given family. The source of possible identifications was the SWISSPROT 32 sequence database (Bairoch and Apweiler, 1996). Because scoring matrices are numeric, each method essentially ranked sequences in the database, according to their estimated closeness to the given family.

We used each test set as the benchmark for evaluating accuracy. In order to avoid penalizing sequences not in the test set but nevertheless in the given family, we used the PROSITE 13.0 database (Bairoch *et al.*, 1997). If a predicted sequence was not in the test set but included in the PROSITE database as a member of the family, we considered it neither a true positive nor a false positive. We converted each ranked list of true and false positives into an equivalence number (Pearson, 1995). The equivalence number is the rank where the number of true positives with lower scores is equal to the number of false positives with higher scores. This number measures how well a method discriminates between true positives and false positives, and it is robust to outliers in the test set. Therefore, the cross-validation test produced a series of equivalence numbers for each method.

Although the BLOCKS 9.3 database contains over 3000 alignment blocks, it was too computationally expensive to test all blocks for all methods. We excluded from our test those blocks that were the easiest to characterize, and therefore least discriminating, as indicated by a zero equivalence score on a screening run using the step-function BLOSUM 62 method. Of the 3417 alignment blocks in the database, 2408 (70%) had an equivalence number of zero. We excluded these blocks from our cross-validation test, leaving 1009 blocks with a screening equivalence number of 1 or more. To create our cross-validation test, we randomly divided each of these blocks into two parts. This random division was different from that used for the screening step, so the cross-validation test used a different set of training and test sets than the screening step did. Random division led to 33 blocks with an empty part, leaving 976 blocks. Our two-fold cross-validation test therefore consisted of 1952 training and test sets.

We implemented the seven approaches discussed in Methods (counting squared error and relative entropy as separate methods), combining each approach with both background and substitution pseudofrequencies, as appropriate. Background pseudofrequencies were taken from SWISSPROT version 32. We used three substitution matrices: the BLOSUM 30, BLOSUM 62, and BLOSUM 100 matrices from BLOSUM version 5.0. We implemented and evaluated the nine-component Dirichlet-mixture method of Sjölander *et al.* (1996).

We also studied the effect of sequence weighting, which has been shown to improve the sensitivity of sequence analysis by increasing the weight of distant members of a given family. The literature (Henikoff, 1996) and our own experience suggest that existing weighting techniques generally give similar results. Hence, we implemented a simple position-specific weighting scheme proposed by Henikoff and Henikoff (1994). Thus, we tested each method on both unweighted and weighted training data. Altogether, we performed our cross-validation test on 48 different variants of the seven approaches. For simplicity, we will henceforth call each of these variants a separate "method."

3.2. *Ranking of methods*

We ranked methods by comparing them pairwise and aggregating the results, essentially equivalent to conducting a round-robin tournament. For each pairwise comparison of method A versus method B, we scored the number of blocks for which A did better (points for, *PF*) and the number for which B did better (points against, *PA*), based on their equivalence scores. We performed a binomial test for obtaining *PF* points out of $PF + PA$ total trials, assuming that the probability of obtaining each point was 0.5. Based on this test, we labeled each pairwise comparison as a win or loss (if the difference was statistically significant) or a tie (if it was not significant).

We aggregated these results by totaling wins, losses, and ties for each method against competing methods under a significance threshold of $p < 0.05$. If two or more methods had equivalent ranks under this threshold, we used the wins, losses, and ties under a looser threshold of $p < 0.20$ to differentiate their ranks. The resulting rankings are shown in Table 1.

These rankings show that the best-performing method was the minimal-risk method using the squared-error metric, BLOSUM 100 substitution pseudofrequencies, and no sequence weighting. This method outperformed all 47 other methods, of which 40 wins were statistically significant at $p < 0.05$. The second best method was the same method, except with a weighted training set. The third best method was the weighted step-function BLOSUM 100 method, followed by the BLOSUM 62 version of the same method.

The rankings are summarized in Table 2. This table shows the rankings for each method as a function of the source of prior information and sequence weighting. Overall, the squared-error method performed well over a wide range of conditions. The step-function method performed almost as well. Among the remaining pseudocount methods, the constant method was generally the next best method, followed by the square-root method and the relative-entropy method. The average-score method appeared to give the worst results. The Dirichlet-mixture method performed well when the sequences were weighted, but poorly when they were not.

With the exception of the Dirichlet-mixture method, the presence of sequence weighting had relatively little influence on the rankings. Sequence weighting usually improved results slightly, although it appears to have worsened the results for the square-root method in each case. In particular, the squared-error method was affected only minimally by the presence of sequence weighting.

The choice of pseudofrequencies had a much greater effect on the results. For almost all methods, substitution pseudofrequencies based on the BLOSUM 100 matrix improved performance on our test relative to the BLOSUM 62 and BLOSUM 30 methods. One exception is the constant method with a BLOSUM 62 matrix and sequence weighting. One possible explanation is that the choice of 50 pseudocounts was chosen empirically for that particular matrix and sequence weighting (Henikoff and Henikoff, 1996). Background pseudofrequencies generally did relatively poorly.

3.3. *Previous studies*

Previous studies of scoring matrices have been performed by other researchers. Karplus (1995) studied methods from a theoretical perspective, rather than empirical performance in sequence analysis. He computed an information-based encoding cost for various methods and obtained the best results for Dirichlet mixture models that he optimized for each particular value of N. However, he tested a specific set of pseudocount methods different from those in the literature. His pseudocount methods are roughly equivalent to a constant method where $B = 1$; a pure pseudocount method with $\alpha = 0$; and a "scaled" pseudocount method

TABLE 1. RANKING OF METHODS

Rank	Method	$p < 0.05$			$p < 0.20$		
		Wins	*Losses*	*Ties*	*Wins*	*Losses*	*Ties*
1	*Squared error*, BLOSUM 100, unweighted	40	0	7	44	0	3
2	*Squared error*, BLOSUM 100, weighted	38	0	9	42	0	5
3	Step function, BLOSUM 100, weighted	38	0	9	40	0	7
4	Step function, BLOSUM 62, weighted	37	0	10	41	0	6
5	Dirichlet mixture, weighted	37	0	10	37	4	6
5	Constant, BLOSUM 62, weighted	37	0	10	37	4	6
7	*Squared error*, BLOSUM 62, unweighted	37	1	9	37	1	9
7	Step function, BLOSUM 100, unweighted	37	1	9	37	1	9
9	*Squared error*, BLOSUM 62, weighted	37	1	9	37	2	8
10	Step function, BLOSUM 62, unweighted	36	1	10	37	3	7
11	Constant, BLOSUM 100, weighted	36	1	10	37	4	6
12	Constant, BLOSUM 100, unweighted	33	10	4	34	11	2
13	Step function, BLOSUM 30, weighted	31	10	6	34	11	2
14	*Squared error*, BLOSUM 30, weighted	30	11	6	32	11	4
15	*Squared error*, BLOSUM 30, unweighted	30	11	6	31	14	2
16	Constant, BLOSUM 62, unweighted	26	12	9	28	13	6
17	Square root, BLOSUM 100, unweighted	26	13	8	28	13	6
18	Step function, BLOSUM 30, unweighted	25	12	10	29	15	3
19	*Squared error*, Background, unweighted	23	15	9	23	16	8
20	*Squared error*, Background, weighted	22	15	10	24	17	6
20	Square root, BLOSUM 100, weighted	22	15	10	24	17	6
22	Square root, BLOSUM 62, unweighted	22	16	9	24	17	6
23	Square root, BLOSUM 62, weighted	20	17	10	23	18	6
24	Step function, Background, weighted	20	17	10	21	18	8
25	Constant, BLOSUM 30, weighted	19	19	9	19	22	6
26	*Relative entropy*, Background, weighted	19	21	7	20	23	4
27	Step function, Background, unweighted	17	19	11	19	22	6
28	Square root, BLOSUM 30, unweighted	15	22	10	18	24	5
29	Square root, BLOSUM 30, weighted	13	24	10	17	25	5
30	Dirichlet mixture, unweighted	11	26	10	11	27	9
31	Constant, Background, weighted	11	27	9	12	29	6
32	*Relative entropy*, Background, unweighted	11	28	8	12	28	7
33	*Relative entropy*, BLOSUM 100, weighted	10	26	11	10	29	8
34	Square root, Background, weighted	10	29	8	11	29	7
35	Square root, Background, unweighted	9	28	10	10	29	8
36	Constant, BLOSUM 30, unweighted	9	29	9	10	30	7
37	*Relative entropy*, BLOSUM 62, weighted	8	29	10	9	32	6
38	*Relative entropy*, BLOSUM 100, unweighted	8	34	5	9	35	3
39	*Relative entropy*, BLOSUM 62, unweighted	7	36	4	8	36	3
40	Average score, BLOSUM 100, weighted	6	33	8	7	35	5
41	Constant, Background, unweighted	5	36	6	6	39	2
42	*Relative entropy*, BLOSUM 30, weighted	5	39	3	5	40	2
43	Average score, BLOSUM 62, weighted	3	39	5	3	40	4
44	*Relative entropy*, BLOSUM 30, unweighted	3	42	2	4	42	1
45	Average score, BLOSUM 100, unweighted	3	42	2	3	43	1
46	Average score, BLOSUM 62, unweighted	2	45	0	2	45	0
47	Average score, BLOSUM 30, weighted	1	46	0	1	46	0
48	Average score, BLOSUM 30, unweighted	0	47	0	0	47	0

Each line indicates the ranking, followed by the method used, the source of pseudofrequencies if applicable, and the presence or absence of sequence weighting. Lines in italics represent methods introduced in this paper. The columns indicate the numbers of statistically significant wins, losses, and ties for each method when compared pairwise against other methods. Rankings are based primarily at a statistical significance of $p < 0.05$, with equal ranks broken at a statistical significance of $p < 0.20$. Remaining equal ranks are indicated by duplicate ranks.

TABLE 2. SUMMARY OF RANKINGS

	Const		Sq root		Step		Sq err		Rel ent		Avg sc		Dir	
Prior information	*U*	*W*	*U*	*W*	*U*	*W*	*U*	*W*	*U*	*W*	*U*	*W*	*U*	*W*
BLOSUM 100	12	11	17	20	7	3	1	2	38	33	45	40		
BLOSUM 62	16	5	22	23	10	4	7	9	39	37	46	43		
BLOSUM 30	36	25	28	29	18	13	15	14	44	42	48	47		
Background	41	31	35	34	27	24	19	20	32	26				
9 components													30	5

Each entry indicates the ranking for a particular method, as a function of the prior information used and the presence of sequence weighting. The letter U indicates unweighted training sets; W indicates a weighted training sets. The methods are abbreviated as follows: Const, constant; Sq root, square root; Step, step function; Sq err, squared error; Rel ent, relative entropy; Avg sc, average score; Dir, Dirichlet mixture.

with $B = 1/N$. He found that the scaled pseudocount method gave results almost as good as his Dirichlet models.

Other studies have looked at the behavior of scoring matrices in more realistic biological tasks. Tatusov *et al.* (1994) studied nine alignment blocks for which the true family members were chosen manually. They studied the Dirichlet-mixture and average-score methods, as well as the square-root method with both background and BLOSUM 62 pseudofrequencies. Training sets in their studied were not weighted. They found that the Dirichlet-mixture method performed best, followed by similar performances between the square-root BLOSUM 62 and average-score methods; the background square-root method performed worst. In our study, we found that these methods generally did poorly, with ranks of 30, 22, 46, and 35, respectively.

Henikoff and Henikoff (1996) studied nine different methods for constructing scoring matrices, using 1673 alignment blocks from BLOCKS 5.0. All training sets were weighted according to same weighting method that we used. They used BLOSUM 62 as needed for pseudofrequencies, and PROSITE 12 as their criterion for family membership. They compared each method against a control method, based on odds ratios. Henikoff and Henikoff found that the step-function BLOSUM 62 method performed best, followed by similar performances by the constant substitution-pseudofrequency method and Dirichlet-mixture method. The methods they studied, in their order of performance, correspond to those ranked 4, 5 (tie), 5 (tie), 34, 23, and 43 in our study. Thus, our relative ranking and theirs were roughly similar.

4. DISCUSSION

In this paper, we have presented a theory for weighting the relative importance of observed data and prior information in characterizing protein families. Whereas previous methods have relied on intuitive or empirical grounds, our method is based on an objective criterion and model of the source of pseudofrequencies. Our objective criterion is to minimize risk, using either a squared-error or a relative-entropy metric. Our formulas for minimizing risk are based on models of the source of the pseudofrequencies: whether they are background or substitution pseudofrequencies, and if they are substitution pseudofrequencies, the substitution matrix used.

We have conducted a large-scale study to test our method and compare it with existing ones. Our study examines more methods than previous studies. In particular, we have studied several possible variants of each method. We have used a statistical test to identify significant pairwise differences between methods, and we therefore believe that our rankings are reliable. Furthermore, the agreement between the relative ranking of methods in our study and those in a previous large-scale study suggests that the rankings should withstand differences in methodology.

Our method, under a squared-error metric, performed well in our experiment, with rankings in the upper half of all methods. On the other hand, the relative-entropy metric performed relatively poorly, with rankings in the lower half of all methods. One possible explanation is that the relative-entropy metric uses relative accuracies of frequency estimates, so it emphasizes smaller frequencies at the expense of larger ones. However, the larger frequencies are the ones most likely to occur in a given family. A squared-error metric, which measures absolute accuracies, appears to be more appropriate for finding homologs.

Our study also looked at the effect of sequence weighting and of different substitution matrices. In general, sequence weighting had a relatively minor effect on performance, compared with the choice of method. One exception was the Dirichlet-mixture method, where sequence weighting was critical to the success of the method. The BLOSUM 100 matrix performed better than the BLOSUM 62 matrix, which in turn performed better than the BLOSUM 30 matrix. However, the ultimate choice of a substitution matrix may depend on the evolutionary distance desired. The BLOSUM 100 matrix may be more suitable for finding homologs that are close in evolutionary distance, whereas the BLOSUM 30 matrix may be better for finding more distant homologs. Because our method is not fine-tuned for a particular substitution matrix, it should give consistent results over a range of evolutionary distances.

Pseudocount methods, including ours, depend on the value of N, the number of sequences in the block. However, this estimate of N assumes that the sequences are independent observations from an underlying distribution. Often, sequences are not independent, which in fact is one of the motivations for sequence weighting. Altschul *et al.* (1997) propose that an "effective" size N_C should be used instead, and they describe a method for estimating this value. An effective size could be incorporated into our method, as well as other pseudocount methods, and this modification may have changed our empirical results. We believe that this issue is important and worthy of further investigation.

Estimation of amino acid frequencies is challenging because it requires search in a 19-dimensional space. (One degree of freedom is constrained by requiring that frequencies sum to 1.) Single-parameter pseudocount methods reduce the problem to a search along one dimension, which is oriented in a biologically meaningful direction by an appropriate choice of pseudofrequencies. Approaches that use higher-dimensional search spaces, such as the nine-component Dirichlet-mixture method, are possible, and we could extend the minimal-risk theory to handle such an approach. However, too much freedom in the search process can worsen performance through overfitting, whereby extra parameters model noise in the observed data, resulting in higher variance and lower quality of the estimates.

Accurate frequency estimates are central to many problems in computational biology. Our theory not only produces minimal-risk scoring matrices for finding homologs and analyzing sequences, but also generates frequency estimates that may be used for aligning sequences or discovering conserved regions in protein families. Our frequency estimates may be used as the basis for hidden Markov models (Krogh *et al.*, 1994) or Gibbs sampling techniques (Lawrence *et al.*, 1993). By characterizing conservation and variability in protein families, minimal-risk estimation may lead not only to more accurate tools for protein sequence analysis, but also to a better understanding of the evolutionary process.

A. APPENDIX

In this Appendix, we derive closed-form solutions for the squared-error metric (Equation 8) and provide details for the numerical solution for the relative-entropy metric.

A.1. Squared-error metric: background pseudofrequencies

We begin with the equation for risk using a squared-error metric and background pseudofrequencies $\mathbf{q}$:

$$R = \sum_{j=1}^{20} E\left(p_j^* - p_j\right)^2 \tag{12}$$

$$= \sum_{j=1}^{20} E((\alpha C_j/N + \beta q_j) - p_j)^2 \tag{13}$$

$$= \sum_{j=1}^{20} E\left[\alpha^2 C_j^2/N^2 + \beta^2 q_j^2 + p_j^2 + 2\alpha\beta C_j q_j/N - 2\alpha C_j p_j/N - 2\beta q_j p_j\right] \tag{14}$$

Note that C_j is a set of random variables, generated according to a multinomial distribution with parameters $\mathbf{p}$. In order to compute the expected values involving C_j, we require the following lemma:

Lemma 1. *Let $C \sim Mult(N, \mathbf{p})$ be a set of counts generated by a multinomial distribution with parameters $\mathbf{p} = \langle p_1, \ldots, p_t \rangle$. Then the following expectations hold:*

$$E(C_j) = Np_j \tag{15}$$

$$E(C_j^2) = Np_j + N(N-1)p_j^2 \tag{16}$$

$$E(C_jC_k) = N(N-1)p_jp_k \quad \textit{for } j \neq k \tag{17}$$

Proof. These quantities follow from the fact that C_j equals the sum of N independent Bernoulli trials, each with probability p_j of success. Therefore, $C_j = \sum_{n=1}^{N} Y_{jn}$, where the result of the nth trial, Y_{jn}, equals 1 with probability p_j and 0 otherwise. The lemma follows from using the Bernoulli sum, and observing that Y_{jm} and Y_{jn} are independent for $m \neq n$, and that Y_{jn} and Y_{kn} cannot both equal 1 for $j \neq k$. ■

We substitute the results of Lemma 1 into Equation 14 to obtain

$$R = \sum_{j=1}^{20} \beta^2 \frac{p_j(1-p_j) + N(p_j - q_j)^2}{N} - 2\beta p_j(1-p_j) + p_j(1-p_j) \tag{18}$$

We differentiate with respect to β and set the result equal to zero to obtain the value for β that minimizes R.

A.2. Squared-error metric: substitution pseudofrequencies

When the pseudofrequencies are based on a substitution matrix, we have the following risk function:

$$\begin{aligned} R = \sum_{j=1}^{20} E\Bigg[& \alpha^2 C_j^2/N^2 + \beta^2 \left(\sum_{k=1}^{20} M_{jk}C_k/N \right)^2 + p_j^2 \\ & + 2\alpha\beta C_j \sum_{k=1}^{20} M_{jk}C_k/N - 2\alpha C_j p_j/N - 2\beta p_j \sum_{k=1}^{20} M_{jk}C_k/N \Bigg] \end{aligned} \tag{19}$$

In order to compute the expected values in this formula, we require Lemma 1 as well as the following lemma:

Lemma 2. *Let M be a $t \times t$ matrix with entries M_{jk}. Define the following quantities:*

$$r_j = \sum_{k=1}^{t} M_{jk}\hat{p}_k \tag{20}$$

$$s_j = \sum_{k=1}^{t} M_{jk}p_k \tag{21}$$

Then the following expectations hold:

$$E\left(\sum_{k=1}^{t} M_{jk}C_k \right) = Ns_j \tag{22}$$

$$E\left(\sum_{k=1}^{t} M_{jk}C_k \right)^2 = N\left[\sum_{k=1}^{t} M_{jk}^2 p_k + (N-1)s_j^2 \right] \tag{23}$$

$$E\left(C_j \sum_{k=1}^{t} M_{jk}C_k \right) = N\left[M_{jj}p_j + (N-1)p_js_j \right] \tag{24}$$

Proof. Equation 22 follows from applying Equation 15 to the left-hand side. Equation 23 follows by expanding its left-hand side as follows

$$E\left(\sum_{k=1}^{t} M_{jk}^2 C_k^2 + \sum_{k=1}^{t}\sum_{l\neq k} M_{jk}M_{jl}C_kC_l\right)$$

and substituting Equations 16 and 17. Equation 24 follows by expanding its left-hand side as follows

$$E\left(M_{jj}C_j^2 + \sum_{k\neq j} M_{jk}C_jC_k\right)$$

and substituting Equations 16 and 17. ■

We apply Lemmas 1 and 2 to Equation 19 to obtain

$$R = (D\beta^2 - 2E\beta + F)/N \tag{25}$$

where

$$D = 1 + \sum_{j=1}^{20}\left[(N-1)(p_j - s_j)^2 - 2M_{jj}p_j + \sum_{k=1}^{20} M_{jk}^2 p_k\right]$$

$$E = 1 - \sum_{j=1}^{20}[p_j(p_j - s_j) + M_{jj}p_j]$$

$$F = 1 - \sum_{j=1}^{20} p_j^2$$

We differentiate with respect to β and set the result equal to zero to obtain the value for β that minimizes risk.

A.3. Relative-entropy metric: background pseudofrequencies

The equation for risk using a relative-entropy metric and background frequencies $\mathbf{q}$ is

$$R = \sum_{j=1}^{20} E\left((\alpha C_j/N + \beta q_j)\log\left(\frac{\alpha C_j/N + \beta q_j}{p_j}\right)\right) \tag{26}$$

Again, the vector $\mathbf{C}$ obeys a multinomial distribution with parameters N and $\mathbf{p}$. However, we can also view each quantity C_j as a random variable that obeys a binomial distribution with parameters N and p_j. Let $x = C_j/N$. Then, we wish to find the expectation $E[f(x)]$ of a function of a binomial variable, where the function is

$$f(x) = (\alpha x + \beta q_j)\log\frac{\alpha x + \beta q_j}{p_j} \tag{27}$$

This type of problem was solved recently by Abe (1996). He used a Taylor expansion to show that

$$E\left[f\left(\frac{C_j}{N}\right)\right] \approx \sum_{k=0}^{K}\frac{f^{(k)}(p_j)E[(C_j - Np_j)^k]}{k!N^k} \tag{28}$$

where K is the order of the approximation.

For our particular problem, we use the following quantities:

$$f^{(1)}(p_j) = \alpha\left(1 + \log\frac{\alpha p_j + \beta q_j}{p_j}\right) \tag{29}$$

$$f^{(2)}(p_j) = \frac{\alpha^2}{\alpha p_j + \beta q_j} \tag{30}$$

$$f^{(3)}(p_j) = -\frac{\alpha^3}{(\alpha p_j + \beta q_j)^2} \tag{31}$$

$$f^{(4)}(p_j) = \frac{2\alpha^4}{(\alpha p_j + \beta q_j)^3} \tag{32}$$

$$E\big[(C_j - Np_j)^1\big] = 0 \tag{33}$$

$$E\big[(C_j - Np_j)^2\big] = Np_j(1 - p_j) \tag{34}$$

$$E\big[(C_j - Np_j)^3\big] = Np_j(1 - p_j)(1 - 2p_j) \tag{35}$$

$$E\big[(C_j - Np_j)^4\big] = Np_j(1 - p_j)[1 + 3(N - 2)p_j(1 - p_j)] \tag{36}$$

If we substitute these quantities into Equation 28 and then into Equation 26, we obtain a fourth-order approximation for relative-entropy risk:

$$R \approx \sum_{j=1}^{20} (\alpha p_j + \beta q_j) \log\left(\frac{\alpha p_j + \beta q_j}{p_j}\right) + \frac{\alpha^2 p_j(1 - p_j)}{2N(\alpha p_j + \beta q_j)} - \frac{\alpha^3 p_j(1 - p_j)(1 - 2p_j)}{6N^2(\alpha p_j + \beta q_j)^2}$$
$$+ \frac{\alpha^4 p_j(1 - p_j)[1 + 3(N - 2)p_j(1 - p_j)]}{12N^3(\alpha p_j + \beta q_j)^3} \tag{37}$$

For a particular problem, we may use numerical techniques to find values for α and β that minimize risk, subject to the constraint that $\alpha + \beta = 1$.

A.4. Relative-entropy metric: substitution pseudofrequencies

For substitution pseudofrequencies, our risk function becomes

$$R = \sum_{j=1}^{20} E\left[\left(\alpha \frac{C_j}{N} + \beta \sum_{k=1}^{20} M_{jk} \frac{C_k}{N}\right) \log \frac{\alpha \frac{C_j}{N} + \beta \sum_{k=1}^{20} M_{jk} \frac{C_k}{N}}{p_j}\right] \tag{38}$$

In contrast with Equation 26, this equation for risk is nonlinear in C_k and therefore cannot be expressed strictly as a sum of expectations of single binomial variables. However, we can make an approximation that has the correct format by ignoring the dependency between C_j and C_k within each term. Therefore, we treat C_k as constant within the jth term, for $j \neq k$. We may then apply Equation 28 as before, by taking derivatives with respect to C_j/N:

$$f^{(1)}(p_j) = (\alpha + \beta M_{jj})\left(1 + \log \frac{(\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk} C_k/N}{p_j}\right) \tag{39}$$

$$f^{(2)}(p_j) = \frac{(\alpha + \beta M_{jj})^2}{(\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk} C_k/N} \tag{40}$$

$$f^{(3)}(p_j) = -\frac{(\alpha + \beta M_{jj})^3}{\left((\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk} C_k/N\right)^2} \tag{41}$$

$$f^{(4)}(p_j) = \frac{2(\alpha + \beta M_{jj})^4}{\left((\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk} C_k/N\right)^3} \tag{42}$$

This approximation yields the following formula for risk:

$$
\begin{aligned}
R \approx \sum_{j=1}^{20} & \left((\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk}C_k/N\right) \log\left(\frac{(\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk}C_k/N}{p_j}\right) \\
& + \frac{(\alpha + \beta M_{jj})^2 p_j(1 - p_j)}{2N\big((\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk}C_k/N\big)} \\
& - \frac{(\alpha + \beta M_{jj})^3 p_j(1 - p_j)(1 - 2p_j)}{6N^2\big((\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk}C_k/N\big)^2} \\
& + \frac{(\alpha + \beta M_{jj})^4 p_j(1 - p_j)[1 + 3(N - 2)p_j(1 - p_j)]}{12N^3\big((\alpha + \beta M_{jj})p_j + \beta \sum_{k \neq j} M_{jk}C_k/N\big)^3} \qquad (43)
\end{aligned}
$$

We may use numerical techniques to obtain values of α and β that minimize risk, subject to the constraint $\alpha + \beta = 1$.

ACKNOWLEDGMENTS

We thank Art Owen for helpful discussions about estimation and for pointers to the statistical literature. T.D.W. is a Howard Hughes Medical Institute Physician Postdoctoral Fellow. C.G.N. and D.L.B. are supported by a grant from Smith-Kline Beecham and by NLM grant LM-05716.

REFERENCES

Abe, S. 1996. Expected relative entropy between a finite distribution and its empirical distribution. *SUT J. Math.* 32, 149–156.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410.

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.* 25:3389–3402.

Attwood, T.K., and Beck, M.E. 1994. PRINTS—a protein motif fingerprint database. *Protein Eng.* 7, 841–848.

Bairoch, A. 1992. PROSITE: a dictionary of sites and patterns in proteins. *Nucleic Acids Res.* 20, 2013–2018.

Bairoch, A., and Apweiler, R. 1996. The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucleic Acids Res.* 24, 21–25.

Bairoch, A., Bucher, P., and Hofmann, K. 1997. The PROSITE database: its status in 1997. *Nucleic Acids Res.* 25, 217–221.

Berg, O.G., and von Hippel, P.H. 1987. Selection of DNA binding sites by regulatory proteins: statistical-mechanical theory and application to operators and promoters. *J. Mol. Biol.* 193, 723–750.

Bishop, Y.M.M., Fienberg, S.E., and Holland, P.W. 1975. *Discrete Multivariate Analysis.* MIT Press, Cambridge, MA.

Brown, M., Hughey, R., Krogh, A., Mian, I.S., Sjölander, K., and Haussler, D. 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families, 47–55. *In* Hunter, L., Searls, D., and Shavlik, J., eds., *Proceedings, First International Conference on Intelligent Systems in Molecular Biology.* AAAI Press, Menlo Park, CA.

Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. 1978. A model of evolutionary change in proteins, 345–358. *In* Dayhoff, M., ed., *Atlas of Protein Sequence and Structure.* Volume 5. National Biomedical Research Foundation, Washington, DC.

Dempster, A.P., Laird, N.M., and Rubin, D.B. 1977. Maximimum likelihood estimation from incomplete data via the EM algorithm. *J. R. Stat. Soc. B* 39, 1–38.

Eddy, S.R. 1996. Hidden Markov models. *Curr. Opin. Struct. Biol.* 6, 361–365.

Gonnet, G.H., Cohen, M.A., and Benner, S.A. 1992. Exhaustive matching of the entire protein sequence database. *Science* 256, 1443–1445.

Gribskov, M., McLachlan, A.D., and Eisenberg, D. 1987. Profile analysis: detection of distantly related proteins. *Proc. Natl. Acad. Sci. U.S.A.* 84, 4355–4358.

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

Henikoff, S. 1996. Scores for sequence searches and alignments. *Curr. Opin. Struct. Biol.* 6, 353–360.

Henikoff, S., and Henikoff, J.G. 1991. Automated assembly of protein blocks for database searching. *Nucleic Acids Res.* 19, 6565–6572.

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

Henikoff, S., and Henikoff, J.G. 1994. Position-based sequence weights. *J. Mol. Biol.* 243, 574–578.

Henikoff, S., Henikoff, J.G., Alford, W.J., and Pietrokovski, S. 1995. Automated construction and graphical presentation of protein blocks from unaligned sequences. *Gene* 163, GC17–GC26.

Jones, D.T., Taylor, W.R., and Thornton, J.M. 1992. The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.* 8, 275–282.

Karplus, K. 1995. Evaluating regularizers for estimating distributions of amino acids, 188–196. *In:* Rawlings, C., Clark, D., Altman, R., Hunter, L., Lengauer, T., and Wodak, S., eds., *Proceedings, Third International Conference on Intelligent Systems in Molecular Biology*, AAAI Press, Menlo Park, CA.

Krogh, A., Brown, M., Mian, I.S., Sjölander, K., and Haussler, D. 1994. Hidden Markov models in computational biology. Applications to protein modeling. *J. Mol. Biol.* 235, 1501–1531.

Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., and Wooton, J.C. 1993. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* 262, 208–214.

Nevill-Manning, C.G., Wu, T.D., and Brutlag, D.L. 1990. Highly specific protein sequence motifs for genome analysis. *Proc. Natl. Acad. Sci. U.S.A.* 95, 5865–5871.

Pearson, W.R. 1995. Comparison of methods for searching protein sequence databases. *Protein Sci.* 4, 1145–1160.

Pearson, W.R., and Lipman, D.J. 1988. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. U.S.A.* 85, 2444–2448.

Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., and Haussler, D. 1996. Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Appl. Biosci.* 12, 327–345.

Staden, R. 1990. Searching for patterns in protein and nucleic acid sequences. *Methods Enzymol.* 183, 193–211.

Stormo, G.D., and Hartzell III, G.W. 1989. Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci. U.S.A.* 86, 1183–1187.

Tarpey, T., and Flury, B. 1996. Self-consistency: a fundamental concept in statistics. *Stat. Sci.* 11, 229–243.

Tatusov, R.L., Altschul, S.F., and Koonin, E.V. 1994. Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks. *Proc. Natl. Acad. Sci. U.S.A.* 91, 12091–12095.

Trybula, S. 1958. Some problems of simultaneous minimax estimation. *Ann. Math. Stat.* 29, 234–253.

Worley, K.C., Wiese, B.A., and Smith, R.F. 1995. BEAUTY: an enhanced BLAST-based search tool that integrates multiple biological information resources into sequence similarity search results. *Genome Res.* 5, 173–184.

Address reprint requests to:
Thomas D. Wu
Beckman Center B400
Department of Biochemistry
Stanford University School of Medicine
Stanford, CA 94305-5307

E-mail: thomas.wu@stanford.edu