



## Modeling the percolation of annotation errors in a database of protein sequences

Walter R. Gilks<sup>1,\*</sup>, Benjamin Audit<sup>2,†</sup>, Daniela De Angelis<sup>1,3</sup>,  
Sophia Tsoka<sup>2</sup> and Christos A. Ouzounis<sup>2</sup>

<sup>1</sup>Medical Research Council Biostatistics Unit, Cambridge,, <sup>2</sup>Computational Genomics Group, The European Bioinformatics Institute, EMBL Cambridge Outstation, Cambridge, CB10 1SD, UK and <sup>3</sup>Statistics Unit, Public Health Laboratory Service, London, UK

Received on April 5, 2002; revised on May 30, 2002; accepted on June 6, 2002

### ABSTRACT

Public sequence databases contain information on the sequence, structure and function of proteins. Genome sequencing projects have led to a rapid increase in protein sequence information, but reliable, experimentally verified, information on protein function lags a long way behind. To address this deficit, functional annotation in protein databases is often inferred by sequence similarity to homologous, annotated proteins, with the attendant possibility of error. Now, the functional annotation in these homologous proteins may itself have been acquired through sequence similarity to yet other proteins, and it is generally not possible to determine how the functional annotation of any given protein has been acquired. Thus the possibility of chains of misannotation arises, a process we term 'error percolation'. With some simple assumptions, we develop a dynamical probabilistic model for these misannotation chains. By exploring the consequences of the model for annotation quality it is evident that this iterative approach leads to a systematic deterioration of database quality.

**Contact:** WRG: wally.gilks@mrc-bsu.cam.ac.uk;  
BA and CAO: audit@ebi.ac.uk; ouzounis@ebi.ac.uk

### INTRODUCTION

The computational analysis of genome sequences involves the identification of gene structure, the translation of DNA into protein sequences and finally the annotation of genes and proteins (Tsoka and Ouzounis, 2000). Annotation can be defined as the step during which functional assignment is performed for genes or proteins, usually based on their similarity to previously characterized sequences in public databases (Bork *et al.*, 1998).

All the above steps involve the use of computer systems,

such as various algorithms and databases, with different capabilities—as well as manual operations from a variety of users with different degrees of expertise. This complex, ambiguous process of computational analysis of biological information is unavoidably error-prone (Bork and Koonin, 1998).

In particular, the process of similarity-based sequence annotation has been systematically studied to some extent (des Jardins *et al.*, 1997; Shah and Hunter, 1997) and certain sources of error have been identified (Bork and Koonin, 1998): these range from simple typographical mistakes (Kyrpides and Ouzounis, 1998) or mere omissions (Kyrpides *et al.*, 2000) to the delineation of complex multi-domain protein architectures (Smith and Zhang, 1997) or the annotation of less well characterized protein families (Iyer *et al.*, 2001). On a genome-wide scale the amount of errors of this type may actually proliferate (Devos and Valencia, 2001). Therefore, reliable similarity-based sequence annotation is crucial during the analysis of multiple genomes, which involves the automatic database-driven annotation of hundreds of thousands of genes (Iliopoulos *et al.*, 2000).

Here, we study for the first time how errors in sequence annotation propagate in public databases. In many cases the results of erroneous analyzes are included in sequence databases, which at the same time form the primary source of information. This iterative process might lead to the propagation of these errors in function assignment for an unknown number of gene or protein sequences (Karp, 1998).

In principle, these erroneous assignments might potentially corrupt the full information content of the database (Karp, 1998): we thus define this peculiar operation in bioinformatics as 'error percolation', an analogy to a fluid substance flowing through a porous medium. In the context of genome analysis, erroneous assignments have been described as 'false positive' descriptions with

\*To whom correspondence should be addressed.

† Both these authors contributed equally to this work.

respect to biological function (Kyrpides and Ouzounis, 1999). There is only a rather limited literature on this subject, representing either case studies with cautionary statements (Pallen *et al.*, 1999) or more systematic studies that address the problem of function assignment by sequence similarity (Brenner, 1999; Wilson *et al.*, 2000; Devos and Valencia, 2001; Hegyi and Gerstein, 2001).

We approach this issue by developing a probabilistic model that captures the principal properties of the error percolation process in protein sequence databases and attempts to identify the key factors that affect database quality.

## A MODEL OF ERROR PERCOLATION

### Functional classes of protein sequences

Suppose there exist  $n$  non-overlapping functional classes of protein in nature. In principal, these functional classes might represent any classification scheme. Let  $\Omega_i$  denote the set of all protein sequences in nature belonging to class  $i$ , for  $i = 1, \dots, n$ . Let  $\pi_i$  denote the probability that a sequence, randomly drawn from nature, belongs to  $\Omega_i$ . Thus  $\pi_i$  represents the relative size of class  $i$ .

### Matching sequences

For two sequences  $c$  and  $d$ , let  $c \sim d$  denote the event that sequence  $c$  matches sequence  $d$ , according to some standard sequence comparison algorithm, such as BLAST (Altschul *et al.*, 1990). We assume that  $c \sim d$  implies that  $d \sim c$ , i.e. the results of a sequence comparison are symmetrical. Let  $c \not\sim d$  denote the event that  $c$  does not match  $d$ . For a given sequence  $c \in \Omega_i$ , let  $\mu_{ij}$  denote the probability that  $d \sim c$ , where  $d$  is a sequence drawn at random from  $\Omega_j$ . That is,  $\mu_{ij}$  is the conditional *match* probability

$$\mu_{ij} = \text{prob}(c \sim d \mid c, c \in \Omega_i, d \in \Omega_j). \quad (1)$$

We assume that  $\mu_{ij}$  is constant over all  $c \in \Omega_i$ , so that  $\mu_{ij} = \mu_{ji}$ .

We consider also a related probability. For a given sequence  $c \in \Omega_i$ , let  $q_{ij}$  denote the probability that  $d \in \Omega_j$ , where  $d$  is a sequence drawn at random from all those sequences in nature that match  $c$ . That is,  $q_{ij}$  is the conditional *class* probability

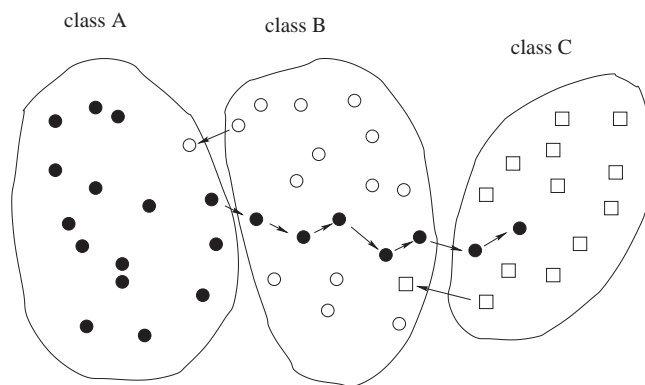
$$q_{ij} = \text{prob}(d \in \Omega_j \mid c, c \in \Omega_i, c \sim d). \quad (2)$$

Then by Bayes' theorem,

$$q_{ij} = \frac{\pi_j \mu_{ij}}{\sum_{k=1}^n \pi_k \mu_{ik}}, \quad (3)$$

so  $\sum_{j=1}^n q_{ij} = 1$ .

For a given sequence  $c \in \Omega_i$ , let  $\eta_i$  denote the probability that  $d \not\sim c$ , where sequence  $d$  is drawn at



**Fig. 1.** Illustrating error percolation. Each bag represents a protein class. Each point within a bag represents a protein of that class. Points close together are similar in terms of sequence. The shape and shading of each point represents its *annotated* class:  $\bullet \rightsquigarrow$  (a);  $\circ \rightsquigarrow$  (b);  $\square \rightsquigarrow$  (c). Arrows indicate the copying of annotation from one protein to another, resulting in a misannotation. Thus chains of misannotation can arise, extending across dissimilar classes, as illustrated for the set of annotations marked by  $\bullet$ .

random from nature. That is,  $\eta_i$  is the *no-match* probability

$$\eta_i = \text{prob}(d \not\sim c \mid c, c \in \Omega_i). \quad (4)$$

Then

$$\eta_i = 1 - \sum_{j=1}^n \pi_j \mu_{ij}. \quad (5)$$

### Annotated database of known sequences

Consider a database of all protein sequences known at time  $t$ . Let  $\mathcal{A}_t$  denote the subset of this database comprising all sequences which have been annotated. We define annotation as the assignment of a sequence to a functional class. Let  $N_t$  denote the size of  $\mathcal{A}_t$ . For now, we assume that one new sequence, drawn at random from nature, is annotated and then added to  $\mathcal{A}_t$  per unit time. That is, we assume  $N_t = t$ , i.e. we use database size as a time coordinate. We also assume that annotations are not updated, having been entered into  $\mathcal{A}_t$ . We relax these assumptions in the sections **Reannotation** and **Model exploration and consequences**.

Thus, at time  $t$ ,  $\mathcal{A}_t$  will contain  $t$  annotated protein sequences. Let  $c_{t+1}$  denote a sequence, drawn randomly from nature, to be added to  $\mathcal{A}_t$  at time  $t + 1$ . Write  $c_{t+1} \rightsquigarrow \Omega_k$  to denote the annotation that  $c_{t+1}$  is attributed to  $\Omega_k$ . We now describe a mechanism by which this attribution is made. This mechanism is not intended to describe precisely how annotations are acquired in the real world. Rather, through its random element, we loosely

represent the variety of ways in which annotations might be determined in practice.

Up to time  $t_e$ , we assume that all sequences have been annotated on the basis of biochemical experiment data. We will refer to such annotations as *experimental*. For all times  $t \geq t_e$ , we assume that  $c_{t+1}$  is annotated by *matching*, as follows. First, a subset  $\tilde{\mathcal{M}}_t(c_{t+1}) \subseteq \mathcal{A}_t$  is compiled, where, for any given sequence  $c$ ,  $\tilde{\mathcal{M}}_t(c)$  denotes all sequences  $d \in \mathcal{A}_t$  such that  $d \sim c$ . If  $\tilde{\mathcal{M}}_t(c_{t+1}) = \phi$ , where  $\phi$  denotes the empty set,  $c_{t+1}$  is not included in  $\mathcal{A}_t$  and a fresh sequence  $c_{t+1}$  is drawn from nature, the process being repeated as necessary until  $\tilde{\mathcal{M}}_t(c_{t+1}) \neq \phi$ . Second, a sequence  $d^*$  is uniformly selected from  $\tilde{\mathcal{M}}_t(c_{t+1})$ , and its annotation is copied to sequence  $c_{t+1}$ . That is, we set  $c_{t+1} \rightsquigarrow \Omega_k$  if  $d^* \rightsquigarrow \Omega_k$ . Third,  $c_{t+1}$  is added to  $\mathcal{A}_t$  to form  $\mathcal{A}_{t+1}$ . This process of annotation by *matching*, described here, simply amounts to the familiar operation of copying descriptions from a selected homologous annotated protein, if available.

### Probabilities of misannotation

Annotations may be inaccurate, i.e. we may have  $c_{t+1} \rightsquigarrow \Omega_k$  when in fact  $c_{t+1} \in \Omega_i$ , where  $i \neq k$ . We are particularly concerned with the potential for inaccurate annotations to percolate through the database, via the copying mechanism described above and illustrated in Figure 1.

Let  $p_{ik}^{(t)}$  denote the probability that  $c_{t+1} \rightsquigarrow \Omega_k$ , given that  $c_{t+1} \in \Omega_i$ . That is,  $p_{ik}^{(t)}$  is the *annotation probability* (correct if  $i = k$  and incorrect if  $i \neq k$ )

$$p_{ik}^{(t)} = \text{prob}(c_{t+1} \rightsquigarrow \Omega_k \mid c_{t+1}, c_{t+1} \in \Omega_i).$$

Let  $\bar{p}_{ik}^{(t)}$  denote the average of such probabilities up to time  $t$ :

$$\bar{p}_{ik}^{(t)} = \frac{1}{t} \sum_{s=1}^t p_{ik}^{(s)}. \quad (6)$$

To calculate  $p_{ik}^{(t+1)}$ , we must consider the ways in which the annotation  $c_{t+1} \rightsquigarrow \Omega_k$  can arise. For each  $j = 1, \dots, n$ , this annotation may arise because it happens that the selected sequence,  $d^*$ , belongs to  $\Omega_j$  but has been attributed to class  $\Omega_k$ . Note that the model does not discriminate between different causes of misannotation, i.e.  $\mu_{ij}$  encompasses all possible errors. Assuming, for all  $s \leq t$ , that  $\mathcal{A}_s$  is approximately an unbiased random sample from nature, it can be easily shown that, for  $t > t_e$ ,

$$p_{ik}^{(t+1)} = \sum_{j=1}^n q_{ij} \bar{p}_{jk}^{(t)}. \quad (7)$$

The above assumption on the unbiasedness of  $\mathcal{A}_s$  is not exact, as a consequence of our requirement that  $\tilde{\mathcal{M}}_t(c_{t+1}) \neq \phi$ . It can be shown that

$$\text{prob}(\tilde{\mathcal{M}}_t(c_{t+1}) = \phi) = \eta_i^t, \quad (8)$$

again assuming unbiasedness of  $\mathcal{A}_s$ . A more exact theory avoiding this assumption would entail considerable additional complexity. However, the assumption may be adequate for our limited purposes, provided that the probability (8) is not large. Indeed, Equation (8) suggests that, for large  $t$ , this probability will be small.

It is convenient to rewrite Equation (7) in matrix form. Let  $P(t)$  denote the  $n \times n$  matrix whose  $(i, k)$ th element is  $p_{ik}^{(t)}$ . Similarly, let  $\bar{P}(t)$  and  $Q_n$  denote the  $n \times n$  matrices formed from  $\bar{p}_{ik}^{(t)}$  and  $q_{ik}$ . Let  $I_n$  denote the  $n \times n$  identity matrix. Then (7) becomes

$$P(t+1) = Q_n \bar{P}(t). \quad (9)$$

From definition (6), we have

$$\bar{P}(t+1) = \frac{t\bar{P}(t) + P(t+1)}{t+1}$$

from which, using (9), we obtain

$$\bar{P}(t+1) - \bar{P}(t) = -\frac{1}{t+1} (I_n - Q_n) \bar{P}(t). \quad (10)$$

It is evident that the rate of change in mean annotation probabilities is inversely proportional to database size.

### Continuous-time approximation

Equation (10) is set in discrete time. Its continuous-time approximation is

$$\frac{d\bar{P}(t)}{dt} = -\frac{1}{t} (I_n - Q_n) \bar{P}(t), \quad (11)$$

and the continuous-time version of (9) is

$$P(t) = Q_n \bar{P}(t), \quad (12)$$

where  $t \geq t_e$ . The general solution to differential Equation (11) can be written, for  $t \geq t_e$ ,

$$\bar{P}(t) = \left\{ \sum_{i=1}^n R(t)^{\lambda_i - 1} \vec{v}_i \vec{v}_i^T \right\} V_n^{-1} \bar{P}(t_e), \quad (13)$$

where  $R(t) = N_t/N_{t_e} = t/t_e$ , the relative size of the database at time  $t$ ;  $\lambda_i$  and  $\vec{v}_i$  are the  $i$ th eigenvalue and right normalized eigenvector of matrix  $Q_n$ ; and

$$V_n = \sum_{i=1}^n \vec{v}_i \vec{v}_i^T. \quad (14)$$

Note that  $\lambda_1 = 1$  and  $\vec{v}_1 = \mathbf{1}_n/\sqrt{n}$ , where  $\mathbf{1}_n$  is the vector of length  $n$  having all elements equal to 1. Thus the first term of the summations in (13) and (14) is  $\mathbf{1}_n \mathbf{1}_n^T/n$ , which does not depend on time,  $t$ .

Equation (13) shows that information on different aspects of classification (represented by different eigenvectors) will be lost at different rates. For example, it may be that the ability to correctly assign sequences from a particular set of classes will be lost more rapidly than for other classes. Overall, we see that information loss is characterized by up to  $(n - 1)$  distinct rates of decay,  $1 - \lambda_i$ , the actual number of distinct rates depending on the structure of  $Q_n$ . Below we consider two possible forms for  $Q_n$ .

As  $Q_n$  is a stochastic matrix, we have that  $|\lambda_i| \leq 1$  for all  $i$ . If the strict inequality  $\lambda_i < 1$  were to hold for each  $i > 1$ , Equation (13) would imply that

$$\bar{P}(t) \longrightarrow \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T V_n^{-1} \bar{P}(t_e) \quad (15)$$

as  $t \longrightarrow \infty$ . Each row of (15) is identical, demonstrating that, eventually, the database would contain no information whatsoever on correct protein classification. Alternatively, if  $\lambda_i = 1$  for some  $i > 1$ , then some aspect of classification would not degenerate with time. For example, if the classes  $\{\Omega_i\}$  were partitioned into sets, and if the sequence-matching tool never matched two sequences from different class sets, then the ability to classify correctly into sets would remain at its level in  $\bar{P}(t_e)$ .

## REANNOTATION

To this point, we have assumed that annotated sequences are not reannotated in the light of subsequent data. Here we consider a scheme for reannotation, based on matching against recent experimental annotations or general corrections.

Suppose that all new sequences,  $c$ , arriving during a given time interval  $(t_r, t_s]$ , have experimental annotations, and are not subject to the usual annotation by matching. Let  $P_e$  denote the matrix of experimental annotation probabilities. Thus  $P(t) = P_e$  for  $t \in (t_r, t_s]$ . Furthermore, suppose the annotation of each such sequence  $c$  is copied to all matching sequences already in the database, thereby reannotating them. Stating this more formally, at each time  $t \in (t_r, t_s]$ , all sequences  $d \in \tilde{\mathcal{M}}_t(c_{t+1})$  are reannotated as  $d \rightsquigarrow \Omega_k$ , where  $k$  is such that  $c_{t+1} \rightsquigarrow \Omega_k$ .

To simplify the analysis, we assume that  $\pi_i = 1/n$  and  $\eta_i = \eta$  for each  $i$ . Then it can be shown that, provided  $(s - r)/s$  is small, approximately,

$$\bar{P}(t + 1) = \eta \bar{P}(t) + \frac{1}{n} M_n P_e, \quad (16)$$

where  $M_n$  is the  $n \times n$  matrix whose elements are  $\mu_{ij}$ . A

continuous-time approximation derived from (16) is

$$\frac{d\bar{P}(t)}{dt} = -(1 - \eta) \bar{P}(t) + \frac{1}{n} M_n P_e,$$

which can be solved to give, for  $t \in (t_r, t_s]$ ,

$$\bar{P}(t) = Q_n P_e + (\bar{P}(t_r) - Q_n P_e) e^{-(1-\eta)(t-t_r)}, \quad (17)$$

using (3). The formula suggests rapid (exponential) convergence towards  $Q_n P_e$ . However, the element of approximation in (17) renders it inapplicable for large  $t$ . Nevertheless, it is useful for assessing the impact of occasional bursts of reannotation activity, which is the most that is likely to be achieved during the current phase of database expansion.

It is possible to continue the simulation for the post-reannotation period,  $t > t_s$ , simply by using Equation (13) again, but with  $t_e$  replaced by  $t_s$ .

## A SIMPLE MODEL OF MATCH PROBABILITIES

In the section **Matching sequences** we assumed homogeneity within each functional class (the match probability  $\mu_{ij}$  is the same for all pairs of proteins from  $(\Omega_i, \Omega_j)$ ). Here we also assume symmetry between protein classes, so that  $\pi_i = 1/n$  for each class  $i$ , and

$$\mu_{ij} = \begin{cases} \delta, & i = j \\ \epsilon, & i \neq j \end{cases}, \quad (18)$$

where  $\delta$  and  $\epsilon$  are parameters to be determined.  $\delta$  represents the match probability between two proteins belonging to same functional class whereas  $\epsilon$  is the match probability between two proteins originating from different classes. These assumptions impose the following structures on the matrix of conditional match probabilities  $Q_n$  and the no-match probabilities  $\eta_i$ :

$$Q_n = \frac{1}{nd_n} \left\{ (\delta - \epsilon) I_n + \epsilon \mathbf{1}_n \mathbf{1}_n^T \right\} \quad (19)$$

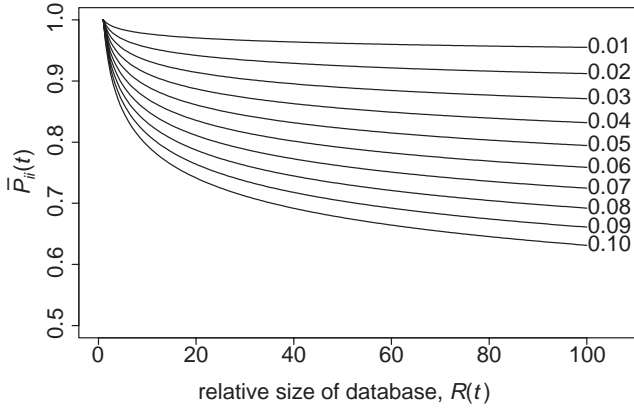
$$\eta_i = \eta = 1 - d_n, \quad (20)$$

where  $d_n = \epsilon + (\delta - \epsilon)/n$ . With the above form for  $Q_n$ , it can be shown that eigenvalues  $\lambda_2 = \lambda_3 = \dots = \lambda_n = 1 - \epsilon/d_n$ , and eigenvectors  $\vec{v}_2, \vec{v}_3, \dots, \vec{v}_n$  are such that

$$\sum_{i=2}^n \vec{v}_i \vec{v}_i^T = I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T.$$

Therefore  $V_n = I_n$ , and Equations (12) and (13) become

$$P(t) = \left\{ \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T + R(t)^{-\alpha} \left( Q_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \right) \right\} \bar{P}(t_e) \quad (21)$$



**Fig. 2.** The decay in the proportion of correctly classified proteins in the database,  $\bar{P}_{ii}(t)$ , with increasing database size,  $R(t)$ , according to (22). Each curve represents the decay for a particular value of  $\alpha$ , assuming  $n = 1000$  classes with experimental misclassification rate  $\rho = 0$ .

$$\bar{P}(t) = \left\{ \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T + R(t)^{-\alpha} \left( I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \right) \right\} \bar{P}(t_e), \quad (22)$$

for  $t \geq t_e$  and where  $\alpha = \epsilon/d_n$  is the exponent of error percolation.

Now  $\bar{P}(t_e) = P_e$ , where  $P_e$  is the matrix of experimental annotation probabilities (see **Reannotation**). If we assume that  $P_e$  has a form similar to that of  $Q_n$ , i.e. the rate of experimental error between two functional classes is a constant, then

$$P_e = (1 - \rho)I_n + \frac{\rho}{n} \mathbf{1}_n \mathbf{1}_n^T, \quad (23)$$

where the probability of experimental error  $\rho$  is a parameter to be determined. Finally, Equations (21) and (22) become

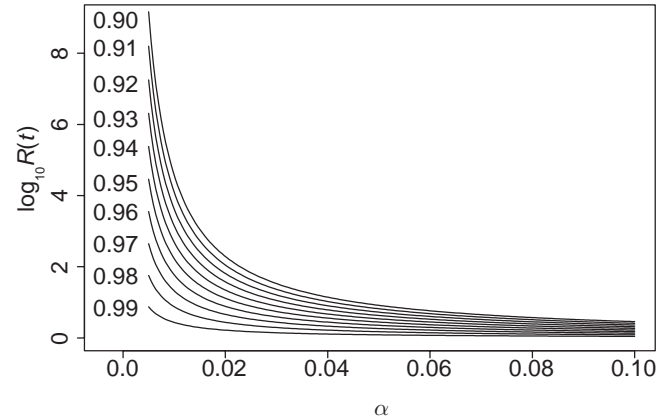
$$P(t) = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T + (1 - \rho)R(t)^{-\alpha} \left( Q_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \right) \quad (24)$$

$$\bar{P}(t) = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T + (1 - \rho)R(t)^{-\alpha} \left( I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \right). \quad (25)$$

*Reannotation* The above results can be integrated in Equation (17) to give annotation probabilities during a reannotation period  $(t_r, t_s]$ .

## ESTIMATION OF ANNOTATION PARAMETERS

In order to estimate annotation parameters we analyze a database of protein families, which contains complete genome sequences and SWISS-PROT, as well as computed pairwise similarity relationships.



**Fig. 3.** Database size,  $R(t)$ , as a function of  $\alpha$ , for each of several values of  $\bar{P}_{ii}(t)$ , under the conditions described in Figure 2.

- (1) The *Tribes* protein family database, accessible through the Web<sup>†</sup> [Enright *et al.*, in preparation]. *Tribes* classifies the 308 879 proteins into 48 045 families. The largest family contains 2775 proteins, 100 families contain more than 100 proteins and 30 139 families consist of only one member.
- (2) The full all-against-all BLAST analysis results, stored in a database called *SIM*. These represent 21 934 887 BLAST hits with an *E*-value smaller than  $10^{-5}$ , filtered for compositionally biased region using CAST.

We consider the *Tribes* families  $\mathcal{F}_i$  to be the functional classes  $\Omega_i$ . We assume that BLAST is our sequence matching tool. For two proteins  $p_1$  and  $p_2$  and given an *E*-value cutoff ( $< 10^{-5}$ ), it can be established, using *SIM*, whether  $p_1$  matches  $p_2$  ( $p_1 \sim p_2$ ) or not ( $p_1 \not\sim p_2$ ). Note this relationship is not necessarily symmetrical ( $p_1 \sim p_2 \not\Rightarrow p_2 \sim p_1$ ) and contains self-comparisons as well as multiple entries for a given pair of proteins.

The *match* probabilities and the derivative quantities are measured by simple counting procedures. We exclude self-matches and consider at most one match per protein pair but we do not symmetrify the relationship.

## Estimations using one family only

We denote the protein database size with  $N$  and the number of protein families with  $n$ . For a given family  $\mathcal{F}_i$ , we define:

- (i)  $N_i$  the family size;
- (ii)  $TN_i$  the number of possible true matches in *SIM* (i.e. maximum number of true positives):

$$TN_i = N_i(N_i - 1); \quad (26)$$

<sup>†</sup> <http://maine.ebi.ac.uk:8000/services/tribes/>

- (iii)  $FN_i$  the number of possible false matches (i.e. maximum number of false positives):

$$FN_i = 2N_i(N - N_i); \quad (27)$$

- (iv)  $TM_i$  the actual true match count in *SIM*;
- (v)  $FM_i$  the actual false match count.

Then straightforward calculations lead to the parameter estimations for family  $\mathcal{F}_i$ :

- The probability  $\delta_i$  that two proteins, belonging to  $\mathcal{F}_i$ , match each other is estimated by:

$$\widehat{\delta}_i = \frac{TM_i}{TN_i} = \frac{TM_i}{N_i(N_i - 1)}. \quad (28)$$

- The probability  $\epsilon_i$  that two proteins, one belonging to  $\mathcal{F}_i$  and one not belonging to  $\mathcal{F}_i$ , match each other is estimated by:

$$\widehat{\epsilon}_i = \frac{FM_i}{FN_i} = \frac{FM_i}{2N_i(N - N_i)}, \quad (29)$$

assuming (*simple model*) that families  $\mathcal{F}_{j \neq i}$  are all equivalent.

- The probability  $\eta_i$  that a protein from  $\mathcal{F}_i$  does not match another protein (selected at random) is estimated by:

$$\widehat{\eta}_i = 1 - \frac{TM_i + \frac{1}{2}FM_i}{N_i(N - 1)}, \quad (30)$$

where  $N_i(N - 1)$  is the number of possible matches for a protein belonging to  $\mathcal{F}_i$  and  $\frac{1}{2}$  scales the number of false matches (overall false matches are counted twice). The equation can also be recovered noticing:  $1 - \widehat{\eta}_i = \frac{1}{N-1} [(N_i - 1)\widehat{\delta}_i + (N - N_i)\widehat{\epsilon}_i]$ .

- Using Equations (29) and (30), the resulting family exponent of error percolation  $\alpha_i$  is:

$$\widehat{\alpha}_i = \frac{\widehat{\epsilon}_i}{1 - \widehat{\eta}_i} = \frac{N - 1}{2(N - N_i)} \frac{FM_i}{TM_i + \frac{1}{2}FM_i}, \quad (31)$$

i.e. as the ratio between the probability of false match and the probability of match. (We consider this approach rather than the alternative formula:  $\alpha_i = \epsilon_i/d_{ni}$  where  $d_{ni} = \epsilon_i + (\delta_i - \epsilon_i)/n$ , because the latter is sensitive to the assumption that all families have the same size.)

### Estimations using more than one family

The previous calculations can easily be generalized in order to use the information in a subset of families  $\{\mathcal{F}_i\}_{i \in \mathcal{J}}$  or in the complete database ( $\mathcal{J} = [1 \dots n]$ ). We only have to sum counts over the set of family indexes  $\mathcal{J}$ , in analogy to Equations (28) to (31):

$$\widehat{\delta}_{\mathcal{J}} = \frac{\sum_{\mathcal{J}} TM_i}{\sum_{\mathcal{J}} (N_i(N_i - 1))}, \quad (32)$$

$$\widehat{\epsilon}_{\mathcal{J}} = \frac{\sum_{\mathcal{J}} FM_i}{2 \sum_{\mathcal{J}} (N_i(N - N_i))}, \quad (33)$$

$$\widehat{\eta}_{\mathcal{J}} = 1 - \frac{\sum_{\mathcal{J}} (TM_i + \frac{1}{2}FM_i)}{(N - 1) \sum_{\mathcal{J}} N_i}, \quad (34)$$

$$\widehat{\alpha}_{\mathcal{J}} = \frac{(N - 1) \sum_{\mathcal{J}} N_i}{2 \sum_{\mathcal{J}} (N_i(N - N_i))} \frac{\sum_{\mathcal{J}} FM_i}{\sum_{\mathcal{J}} (TM_i + \frac{1}{2}FM_i)}. \quad (35)$$

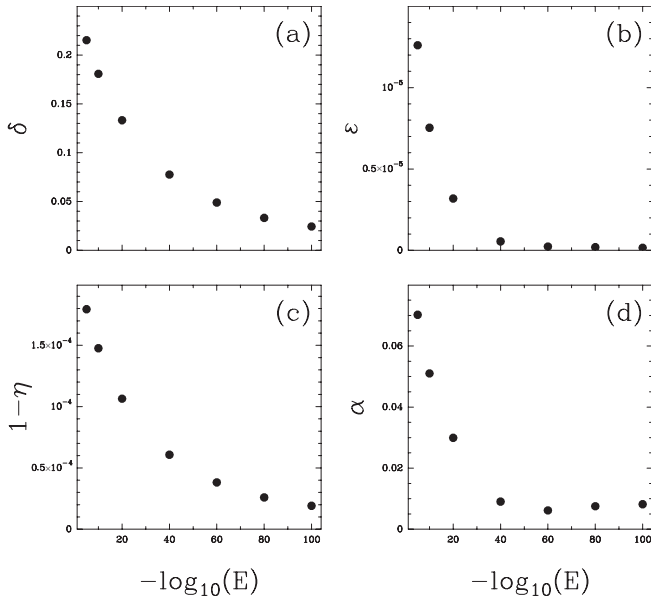
As for Equation (30), Equation (34) can be obtained noticing:  $\widehat{\eta}_{\mathcal{J}} = \sum_{\mathcal{J}} \frac{N_i}{\sum_{\mathcal{J}} N_i} \widehat{\eta}_i$ .

### MODEL EXPLORATION AND CONSEQUENCES

In the present work, we have analyzed for the first time the effect of the iterative strategy for computational protein sequence annotation. We have developed a probabilistic framework to address this issue and, using a continuous-time approximation, have computed an analytical solution (13) expressed in terms of eigenvalues and eigenvectors of the class probability matrix,  $Q$ . Here, we define *database quality* as the proportion of sequences correctly assigned to their true functional class, corresponding to the diagonal terms of matrix  $\bar{P}$ . Also, we define *annotation quality* as the probability of correctly assigning a sequence to its true functional class, corresponding to the diagonal terms of matrix  $P$ . Two important results arise from these calculations:

- In general, the quality of the database decays as a power law of the (relative) database size. This is natural, given that the quality of annotation of an incoming protein is less than the quality of the database (Equation (6), assuming  $\bar{p}_{jj}^{(t)} \geq \bar{p}_{jk}^{(t)}$ , hence the decrease in database quality ( $\bar{p}_{jj}^{(t+1)} \leq \bar{p}_{jj}^{(t)}$ , Equation (7)).
- In particular, there could be a component (corresponding to eigenvalues equal to 1, see Equation (13)) of the classification that will remain unaffected by error percolation.

An important conclusion is that we cannot usually expect to gain information by increasing the database size with sequences annotated by homology. On the contrary,



**Fig. 4.** Parameter estimation for the simple model of match probabilities as a function of the BLAST  $E$ -value cutoff  $E$  used to establish the matching relationship with SIM (see **Estimation of annotation parameters**). The complete Tribes family database was used i.e. estimations were obtained with Equations (32–35) and  $J = [1 \dots n_T]$  where  $n_T = 48045$  is the total number of Tribe families.

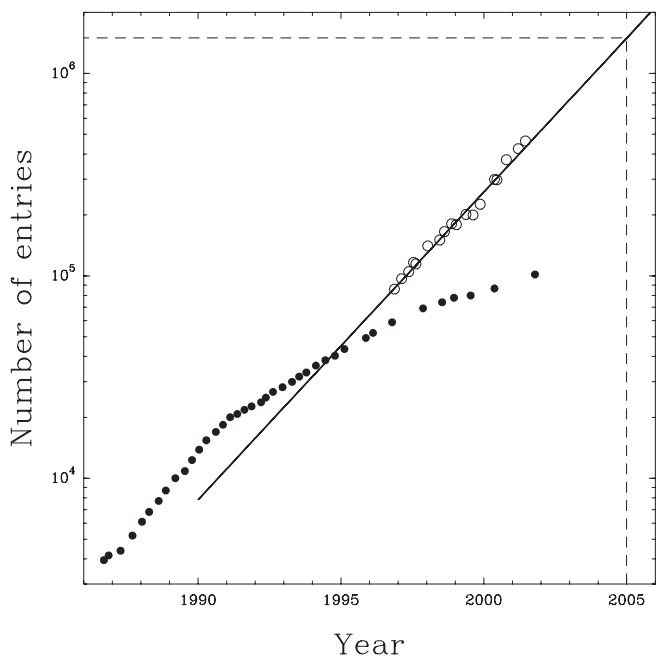
the model suggests that ultimately complete loss of information will take place (Equation (15)). What this work implies is that homology-based annotations should be performed using the experimental annotation only. This has not been common practice in the field, even though preference for curated databases over sequence archives has been given. For example GeneQuiz embodies such rules (Andrade *et al.*, 1999). To achieve this in practice, keeping track of the origin of each annotation would be sufficient (Karp, 1998).

We have specified the properties of the matching tools, describing them by just two probabilities: a true match probability  $\delta$  and a false match probability  $\epsilon$  (Equation (18)). Under this simple model, the rate of loss of valid classification information is controlled by the exponent of error percolation  $\alpha = \epsilon/d_n$  (Equation (22)). Using Equation (20), we can rewrite:  $\alpha = \epsilon/(1 - \eta)$ . In a very logical way,  $\alpha$  is the ratio between the false match probability  $\epsilon$  and the overall match probability  $1 - \eta$  for any given pair of proteins, representing a measure of the discriminating power of the matching tool. Figure 2 shows how the proportion of correctly classified proteins decays over time. We see that the decay is initially very rapid, but appears to slow down as the database size increases. An alternative view of these data is presented

in Figure 3, which might be used to determine how large a database can become before database quality deteriorates below a minimum acceptable level. For each of several settings for the minimum allowable proportion of correctly classified proteins,  $\bar{P}_{ii}(t)$ , Figure 3 plots a curve relating the maximum database size,  $R(t)$ , to the decay exponent  $\alpha$ .

We observe a substantial impact of the exponent of percolation  $\alpha$  on database quality and achievable size, shown in Figures 2 and 3 respectively. To gain some idea of a value for  $\alpha$  that might be applicable in practice, we perform an estimation of the parameters for the simple model using the complete Tribes protein family database (see **Estimations using more than one family**). The results of the estimations, as a function of the BLAST  $E$ -value cutoff used to establish the matching relationships from SIM, are presented in Figure 4a–d. The optimal value for the exponent  $\alpha$  is  $\alpha_{\text{opt}} = 6.2 \cdot 10^{-3}$ , corresponding to an  $E$ -value cutoff of  $E = 10^{-60}$  and a true match probability of  $\delta \simeq 5\%$ . Such a small value of  $\delta$  would penalize small families of proteins, as it would lead to underpredictions, i.e. leaving uncharacterized many proteins that have annotated homologues in the database. In practice, there is a tradeoff between the quality of the annotation (no overpredictions) and the number of annotated, characterized proteins (no underpredictions). For the smaller  $E$ -value cutoffs in Figure 4a–d, the false match probability  $\epsilon$  settles around  $1.5 \cdot 10^{-7}$  and the overall match probability  $1 - \eta$  is of the order of  $10^{-4}$ . Injecting this latter value into Equation (8) along with a database size  $t = 50\,000$  which corresponds to the database sizes in 1995 (Figure 5), we get an upper estimate for the probability that an incoming protein would not have any match in the database after 1995,  $\text{prob}(\tilde{\mathcal{M}}_t = \phi) \lesssim 0.7\%$ . This justifies *a posteriori* the assumption of unbiasedness of  $\mathcal{A}_s$  made in **Probabilities of misannotation**.

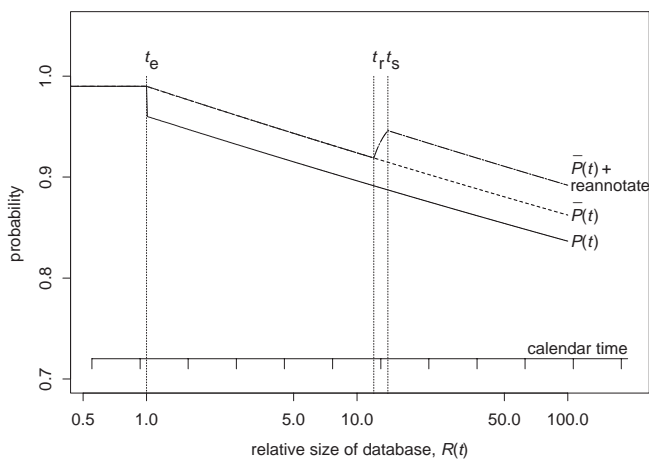
To estimate the maximum acceptable database size at a given quality level, we choose the  $E$ -value cutoff  $E = 10^{-20}$  as a realistic value used in practice. The corresponding parameter estimates are  $\hat{\delta} = 13\%$ ,  $\hat{\epsilon} = 3.2 \cdot 10^{-6}$ , and  $\hat{\alpha} = 0.03$  (Figure 4). These estimates are consistent with an ‘effective’ number of classes,  $\hat{n} = 1256$  (Equation (20)). We do not use the actual  $n$  from Tribes, as it contains a high proportion of singleton classes, themselves defined by lack of BLAST matches. For the two values of  $\alpha$  mentioned above ( $\alpha_{\text{opt}}$  and  $\hat{\alpha}$ ), if we set the minimum acceptable level of database quality to 0.95 i.e. 5% of annotation error, we get a maximum relative database size of 5000 for  $\alpha_{\text{opt}}$  and 5 for  $\hat{\alpha}$ ; yet, if we set this minimum acceptable level to 0.99 these numbers dramatically decrease to 5 and 1.4 respectively. Following Figure 5, relative size can be converted to absolute time: 5000-fold corresponds to 25 years, 5-fold to 5 years



**Fig. 5.** Growth of the number of entries in two data banks of proteins: ● SWISS-PROT and ○ TrEMBL. (—) corresponds to the fit to the TrEMBL data with an exponential model:  $N = N(1995)2^{(y-1995)/\tau}$  with  $N(1995) = 45\,252$  and a doubling time  $\tau = 1.98$  year.

and 1.4-fold to 1 year. In plain terms, given the current database growth and imposing these levels of desirable quality, these estimates suggest that the iterative homology based annotation strategy may not be sustainable for more than a few years.

For the parameter values corresponding to  $E$ -value cutoff  $E = 10^{-20}$ , the system dynamics are shown in Figure 6. Note the use of the logarithmic horizontal scale in Figure 6. Currently, protein sequence databases are growing at an almost exponential rate (Figure 5) therefore the logarithmic scale in  $R(t)$  might be taken to represent a linear scale in calendar time, as indicated on the figure. In this process the doubling time  $\tau$  of the protein database size is transformed to a half-life  $\tau/\alpha$  with respect to database quality. For  $\tau = 1.98$  year (Figure 5) and  $\alpha = \hat{\alpha}$  (resp.  $\alpha = \hat{\alpha}_{opt}$ ), we get  $\tau/\alpha \simeq 66$  year (resp.  $\tau/\alpha \simeq 320$  year). Simply put, it would take decades to centuries before half of the correct annotations will be corrupted by error percolation. These values are large compared to the span of calendar time in Figure 6 and consequently the exponential decrease appears as a straight line. Thus, we see an almost linear decline in database quality over calendar time whose slope can be estimated by  $\ln(2)\alpha/\tau$ . The optimal values of the exponent of error percolation  $\alpha_{opt}$  lead to a decrease of database quality by 0.22% per



**Fig. 6.** The probability  $P_{ii}(t)$  that the next sequence to arrive at time  $t$  will be correctly classified (lower curve); the proportion  $\bar{P}_{ii}(t)$  of correctly classified proteins in the database at time  $t$  (middle curve); and  $\bar{P}_{ii}(t)$  when there is reannotation during time interval  $(t_r, t_s]$  (upper curve), plotted against relative database size,  $R(t)$ , on a logarithmic scale. The linear ‘calendar time’ axis would be relevant to a scenario where  $R(t)$  increases exponentially with calendar time (see text). All curves correspond to the simple model in section **A simple model of match probabilities**, setting  $\delta = 13\%$ ,  $\epsilon = 3.2 \cdot 10^{-6}$ ,  $n = 1256$ , and  $\rho = 0.01$ . These values imply  $\alpha = 0.030$ .

year whereas the realistic exponent  $\hat{\alpha}$  leads to a decrease by 1% per year.

Figure 6 also shows the relationship between the quality of existing annotations,  $\bar{P}(t)$ , and the quality of new annotations,  $P(t)$ . Up to time  $t_e$ , the two curves are identical. Once annotation by matching has commenced at time  $t_e$ ,  $P_{ii}(t)$  immediately drops, and thereafter remains at a constant distance below  $\bar{P}_{ii}(t)$ , as required by Equations (24) and (25). Also shown on Figure 6 is the effect of a sustained period of reannotation, between times  $t_r$  and  $t_s$ , according to the protocol described in **Reannotation**. We see a rapid improvement in this interval, during which  $\bar{P}_{ii}(t)$  increases by 0.03 and  $R(t)$  increases from 12 to 14, but thereafter a decline parallel to the original track of descent. When  $R(t) = 37$  that is roughly 3 years after the end of the reannotation period (Figure 5),  $\bar{P}_{ii}(t)$  returns to the level it was at just prior to reannotation. In short, by incorporating twice as many experimental annotations as was originally available at time  $t_e$ , we get a long-lasting improvement of the database quality of 3%. However, the experimental effort involved in such a burst of reannotation would probably be prohibitive. These results underline our views stated above that experimentally derived annotations should be kept separately.

The iterative process of homology based annotation and incorporation of newly annotated sequences in the original database can have dramatic effects on the overall information content of these repositories because of the percolation of annotation errors. Even though percolation of errors can be maintained within reasonable bounds by the use of very strict criteria of matching, the iterative process will always lead to a deterioration of the database quality.

## FUTURE DIRECTIONS

We are currently investigating extensions to the simple model of error percolation analyzed in the present work. Preliminary calculations for a model with two types of functional classes differing in size show that the qualitative results remain the same but the fundamental role of the relative size of the family in the database becomes prominent. A similar approach is developed to test the effect of the assumption that matching parameters are homogeneous across functional classes.

In the context of homology based annotation transfer, it is crucial to assess the level of description that is acceptable in order to copy information (Devos and Valencia, 2001). We are addressing this issue by developing a model that uses a hierarchical classification of functional classes. Besides the specificity issue, the problem of multi-domain/multi-functional proteins can also be tackled by allowing proteins to belong to more than one functional class in the model. Finally, the probabilistic framework developed here will be employed to building scoring mechanisms to quantitatively evaluate homology-based annotation.

## ACKNOWLEDGEMENTS

We thank Peter Karp (SRI International) for discussions, Anton Enright for the *Tribes* database and members of the Computational Genomics Group for comments. BA is currently supported by a Marie Curie Fellowship of the European Community programme 'Improving Human Research Potential and the Socio-economics Knowledge Base' under contract number HPMF-CT-2001-01321 and ST by the Medical Research Council (UK). Additional support was provided by the European Molecular Biology Laboratory.

## REFERENCES

- 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.
- Andrade,M.A., Brown,N.P., Leroy,C., Hoersch,S., de Daruvar,A., Reich,C., Franchini,A., Tamames,J., Valencia,A., Ouzounis,C. and Sander,C. (1999) Automated genome sequence analysis and annotation. *Bioinformatics*, **15**, 391–412.
- Bork,P. and Koonin,E.V. (1998) Predicting functions from protein sequences—where are the bottlenecks? *Nature Genet.*, **18**, 313–318.
- Bork,P., Dandekar,T., Diaz-Lazcoz,Y., Eisenhaber,F., Huynen,M. and Yuan,Y. (1998) Predicting function: from genes to genomes and back. *J. Mol. Biol.*, **283**, 707–725.
- Brenner,S.E. (1999) Errors in genome annotation. *Trends Genet.*, **15**, 132–133.
- Devos,D. and Valencia,A. (2001) Intrinsic errors in genome annotation. *Trends Genet.*, **17**, 429–431.
- Hegy,H. and Gerstein,M. (2001) Annotation transfer for genomics: measuring functional divergence in multi-domain proteins. *Genome Res.*, **11**, 1632–1640.
- Iliopoulos,I., Tsoka,S., Andrade,M.A., Janssen,P., Audit,B., Tramontano,A., Valencia,A., Leroy,C., Sander,C. and Ouzounis,C.A. (2000) Genome sequences and great expectations. *Genome Biol.*, **2**, interactions0001.1–0001.3.
- Iyer,L.M., Aravind,L., Bork,P., Hofmann,K., Mushegian,A.R., Zhulin,I.B. and Koonin,E.V. (2001) *Quod erat demonstrandum?* The mystery of experimental validation of apparently erroneous analyses of protein sequences. *Genome Biol.*, **2**, research0051.1–0051.11.
- des Jardins,M., Karp,P.D., Krummenacker,M., Lee,T.J. and Ouzounis,C.A. (1997) Prediction of enzyme classification from protein sequence without the use of sequence similarity. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **5**, 92–99.
- Karp,P.D. (1998) What we do not know about sequence analysis and sequence databases. *Bioinformatics*, **14**, 753–754.
- Kyrpides,N.C. and Ouzounis,C.A. (1998) Errors in genome reviews. *Science*, **281**, 1457–1457.
- Kyrpides,N.C. and Ouzounis,C.A. (1999) Whole-genome sequence annotation: 'Going wrong with confidence'. *Mol. Microbiol.*, **32**, 886–887.
- Kyrpides,N.C., Ouzounis,C.A., Iliopoulos,I., Vonstein,V. and Overbeek,R. (2000) Analysis of the *Thermotoga maritima* genome combining a variety of sequence similarity and genome context tools. *Nucleic Acids Res.*, **28**, 4573–4576.
- Pallen,M., Wren,B. and Parkhill,J. (1999) 'Going wrong with confidence': misleading sequence analyses of CiaB and ClpX. *Mol. Microbiol.*, **34**, 195–195.
- Shah,I. and Hunter,L. (1997) Predicting enzyme function from sequence: a systematic appraisal. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **5**, 276–283.
- Smith,T.F. and Zhang,X. (1997) The challenges of genome sequence annotation or 'the devil is in the details'. *Nat. Biotechnol.*, **15**, 1222–1223.
- Tsoka,S. and Ouzounis,C.A. (2000) Recent developments and future directions in computational genomics. *FEBS Lett.*, **480**, 42–48.
- Wilson,C.A., Kreychman,J. and Gerstein,M. (2000) Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. *J. Mol. Biol.*, **297**, 233–249.