BIOINFORMATICS ORIGINAL PAPER Vol. 24 no. 5 2008, pages 621–628
doi:10.1093/bioinformatics/btm633

Sequence analysis

Beyond the 'best' match: machine learning annotation of protein sequences by integration of different sources of information

Igor V. Tetko[1,*], Igor V. Rodchenkov[1], Mathias C. Walter[1], Thomas Rattei[2] and Hans-Werner Mewes[1,2]

[1]Helmholtz Zentrum München - German Research Center for Environmental Health (GmbH), Institute of Bioinformatics and Systems Biology, Ingolstädter Landstraße 1, 85764, Neuherberg and [2]Department of Genome-Oriented Bioinformatics, Wissenschaftszentrum Weihenstephan, Technische Universität München 85350 Freising, Germany

Received on October 7, 2007; revised on November 29, 2007; accepted on December 18, 2007

Advance Access publication January 3, 2008

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: Accurate automatic assignment of protein functions remains a challenge for genome annotation. We have developed and compared the automatic annotation of four bacterial genomes employing a 5-fold cross-validation procedure and several machine learning methods.

Results: The analyzed genomes were manually annotated with FunCat categories in MIPS providing a gold standard. Features describing a pair of sequences rather than each sequence alone were used. The descriptors were derived from sequence alignment scores, InterPro domains, synteny information, sequence length and calculated protein properties. Following training we scored all pairs from the validation sets, selected a pair with the highest predicted score and annotated the target protein with functional categories of the prototype protein. The data integration using machine-learning methods provided significantly higher annotation accuracy compared to the use of individual descriptors alone. The neural network approach showed the best performance. The descriptors derived from the InterPro domains and sequence similarity provided the highest contribution to the method performance. The predicted annotation scores allow differentiation of reliable versus non-reliable annotations. The developed approach was applied to annotate the protein sequences from 180 complete bacterial genomes.

Availability: The FUNcat Annotation Tool (FUNAT) is available on-line as Web Services at http://mips.gsf.de/proj/funat

Contact: i.tetko@gsf.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Numerous genome-sequencing projects have caused a rapid growth of the protein databases. In contrast to the pre-genomic era, when the annotation of sequences was highly biased towards known and characterized genes, the systematic exploration of genomes allows one to assign precise functional properties now on less studied genes. However, manual annotation of sequences is laborious and becomes more and more infeasible for growing amount of sequence data (Friedberg, 2006; Ruepp and Mewes, 2006; Valencia, 2005). The field of automatic functional annotation is rapidly evolving. Widely accepted approaches include the use of sequence alignment tools, such as BLAST, PSI-BLAST (Altschul *et al*., 1997), FASTA (Pearson, 1996) or more sensitive methods, such as hidden Markov models (Sonnhammer *et al*., 1997) or alignment-independent tools (Kocsor *et al*., 2006). These methods are applied to detect a set of candidate genes, which were previously manually annotated. When such set of genes is pre-selected, several strategies are used to provide the annotation. One of the most simplest and straightforward approaches is to annotate a target gene with functions of a prototype gene. Homology-based methods use the highest alignment score ('best bidirectional match') to do such annotation. They were implemented in the first workflow-based annotation systems for sequence analysis such as GeneQuiz (Andrade *et al*., 1999) and PEDANT (Mewes *et al*., 1999; Riley *et al*., 2007). In addition, these approaches performed supplementary analyses to add multiple evidences for the inferences made. Another strategy is to annotate proteins using clustering algorithms applied on many proteins. Orthologous families are detected and the consensus annotation of all genes in the cluster is assigned (Enright *et al*., 2002; Tetko *et al*., 2005a, b). More complicated methods that make use of a combination of annotations of more than one homolog have been also proposed (Abascal and Valencia, 2003; Biswas *et al*., 2002; Clare *et al*., 2006; Clare and King, 2003; Krebs and Bourne, 2004; Kretschmann *et al*., 2001; Levy *et al*., 2005; Meinel *et al*., 2005; von Mering *et al*., 2007). It has been shown that the latter strategies are suited to provide an improved accuracy of annotation over the best match approach.

Assignment of protein function by highest-scoring orthologous match, however, provides an obvious explanation of

*To whom correspondence should be addressed.

the performed annotation. Thus, in case of doubts of the annotation accuracy one can verify the prototype sequence, which was manually annotated, and re-evaluate evidences used for its annotation. By the same reason it allows an easy control of the error-propagation. However, the 'best' match annotation is usually based on just one source of information, which often is sequence similarity. The annotation may not be correct if the best match is not a true ortholog. Indeed, in some cases it is extremely difficult to differentiate the true ortholog amid several possible candidates with similar sequence similarity scores. The highest rank in the scoring list may be the result of considering a local but not a global similarity. While additional evidences could be used to enhance the annotation accuracy by pre- or post-processing (e.g. in case of similar similarity scores, preference for transfer of annotation should be given to genes from syntenic regions), there is a need for a comprehensive method integrating different types of evidences. The current work describes an algorithm that on the one hand keeps the simplicity and advantages of the best match approach but on the other hand provides data integration for automatic annotation of proteins. It significantly improves the annotation accuracy over the traditional 'best match' approach.

2 DATA

The MIPS BFAB data set (Tetko *et al.*, 2005a) composed of four bacterial genomes, *Bacillus subtilis*, *Helicobacter pylori*, *Listeria innocua* and *Listeria monocytogenes* (in total 7335 annotated sequences) was used. These genomes were manually annotated according to the Functional Catalogue FunCat (Ruepp *et al.*, 2004) by the MIPS curators over several years. For each analyzed protein we considered its sequence alignment to all other proteins from the BFAB set. The pre-calculated Smith-Waterman (SW) alignment scores (optscores $\geq$80) from the SIMAP database (Rattei *et al.*, 2008) were used. The choice of the optscore threshold has been described in the primary SIMAP publication (Arnold *et al.*, 2005) as a compromise of sensitivity and size of the matrix, which is currently 1.3 TB. Although this threshold value looks somewhat liberal, our assumption was that the other features, such as common InterPro domains, should play an important role and provide complementary information for the annotation of proteins. We did not consider hits where both proteins belong to the same bacteria specie, and also skipped pairs where both genes come from different Listeria genomes since they are highly conserved. A total number of 104 092 pairs was calculated.

We also automatically annotated complete bacterial genomes with calculated InterPro domains, which are available in the PEDANT (Riley *et al.*, 2007) database. A total number of 180 genomes (378 974 sequences) contributed 12 975 190 protein pairs. Each pair included a sequence from one of the 180 genomes and the other sequence from the annotated genes (BFAB set).

2.1 Descriptors to represent a protein pair

The basic object for our analysis was a pair of proteins. In order to distinguish between the sequences in the pair, we will refer to the sequence to be annotated as the 'target' and the sequence, which provides the annotation as the 'prototype'. Features describing a pair rather than each sequence alone were selected for the analysis. The descriptors were selected following discussions with members of MIPS annotation group and were subdivided into several categories:

(1) *Sequence similarity*. The pairwise sequence similarity optscore, calculated with the Smith-Waterman (SW) algorithm (Smith and Waterman, 1981) as well as length overlap, identity and global similarity, G_SIM, were retrieved from the MIPS SIMAP database (Rattei *et al.*, 2008). While the optscore (or its normalized analog, e-value) is traditionally used to estimate the quality of a sequence alignment, the last three descriptors provide a quantitative estimate of the degree of sequence similarity of both proteins.

(2) *Sequence length attributes*. The difference between lengths of proteins was mentioned by the MIPS curators as an important descriptor for manual annotation. Thus the absolute difference of sequence lengths L_DIF = abs(length(a) − length(b)), the difference weighted by sum of sequence lengths L_PERC = L_DIF/(length(a) + length(b)) were used. The third descriptor was based on the density distribution function $pdf(x)$ of lengths of all protein sequences in 180 genomes (x is the protein length). For a given pair the distance in the space of sequences, L_PDF, was calculated as an integral $\int pdf(x)dx$ from the shorter, L_{short}, to the longer protein, L_{long}, in the pair. It corresponded to a fraction of proteins having length in [L_{short}, L_{long}] interval in the database.

(3) *Synteny information*. Gene neighborhoods have been shown to be an important feature in detecting protein clusters with conserved functions (Kolesov *et al.*, 2001; Marcotte *et al.*, 1999). The synteny score represents the probability of a set of proteins, which are within a specified window (±10 genes or less) on a target genome, to be detected within the same window on the prototype genome. The $\log_{10}$ value of this score was used as a descriptor (SYN).

(4) *Alignment free sequence similarity*. Previous studies have indicated the importance of compression algorithms in comparing the similarity of sequence data. In this study we used bzip2 to calculate the alignment free sequence similarity of sequences using compression-based metrics (CBM) (Cilibrasi and Vitanyi, 2005; Kocsor *et al.*, 2006):

$$\text{CBM}(a,b) = \frac{(C(a+b) - \min[C(a), C(b)])}{\max[C(a), C(b)]} \qquad (1)$$

where $C(x)$ denotes the length of sequence (text string) x, that is compressed by the bzip2 algorithm.

(5) *InterPro Domain* (Mulder *et al.*, 2007) *composition*. The InterPro data (v 16.0) were downloaded from http://www.ebi.ac.uk/interpro/ and used as a primary source of domain composition of protein sequences. We counted the number of distinct domains in both proteins (I_DIST), the number of common domains (I_COM), the ratio of common to total distinct domains

(I_RATIO) and the log probability score of domains, k, detected in both proteins in the pair as

$$\text{I\_PROB} = \frac{\sum_i \log(c_i \times (c_i - 1))}{(N \times (N-1))} \quad (2)$$

where c_i is the total number of proteins with domain i, and N is the total number of proteins in the database. Summation was performed over all domains that were found in both analyzed proteins in the pair. Some InterPro domains have hierarchical structures, in which 'parent' domains are more general compared to the 'children' domains. Use of the previous measures, e.g. Equation (2), does not reflect this kind of relationship. To account for this problem, we introduced additional scores. The I_SCORE for the InterPro domains was calculated in similar way as the score used to estimate the accuracy of FunCat annotation (see section 2.2). Other descriptors were based on informatics-theoretic, I_ITS, (Azuaje *et al.*, 2006; Lin, 1998; Resnik, 1995) and the total ancestry method (Yu *et al.*, 2007).

(6) *Calculated protein properties*. Distance measures calculated from PEDANT (Riley *et al.*, 2007) properties were used. The first two, S_EU and S_CLASS, were calculated using predictions of the secondary structure of proteins with PREDATOR (Frishman and Argos, 1997). The ratio of amino acids, $n(a)_i$, assigned to a given class of secondary structure, i = [helix, extended, coil] were used to calculate the first descriptor, $\text{S\_EU} = \text{sqrt}(\sum_i (n(a)_i - n(b)_i)^2)$. Binary descriptors (0,1) were used to describe similarity in the predicted class of secondary structure ('all alpha', 'all beta', 'alpha/beta'). We assigned a distance value of 0 if both proteins in the pair had the same prediction of the considered property, and 1 otherwise. The next five numerical descriptors were absolute differences in calculated isoelectric points (ip), percentage of low complexity regions (lc), number of transmembrane regions (TMHMM), disordered regions (do), and coiled-coil regions (co). PEDANT3 uses the SEG algorithm (Wootton and Federhen, 1993) to compute low complexity regions, TMHMM v. 2.0 (Krogh *et al.*, 2001) to predict transmembrane helices, GlobPlot v. 1.1 (Linding *et al.*, 2003) to predict regions with disordered secondary structure, and COILS v. 2.1 (Lupas, 1996) to predict coiled-coiled regions.

2.2 Annotation score

The MIPS FunCat categories (Ruepp *et al.*, 2004) were used for protein annotation. The FunCat is an annotation scheme for the functional description of proteins from prokaryotes, unicellular eukaryotes, plants and animals (Mewes *et al.*, 1997; Ruepp *et al.*, 2004). Taking into account the broad and highly diverse spectrum of known protein functions, FunCat consists of 28 main functional categories (or branches) that cover general fields like cellular transport, metabolism and signal transduction. The main branches exhibit a hierarchical, tree-like structure with up to six levels of increasing specificity. The FunCat 2 includes 1445 functional categories both for prokaryotic and eukaryotic genomes. The total number of 413 distinct categories was available for the BFAB set. The manual functional classifications were presented for 7335 proteins.

We used a matching score (Tetko *et al.*, 2005b) to estimate the accuracy of annotation. In order to illustrate it, let us consider a case when a protein with annotation

01.01.03.03, metabolism of proline;
01.05 C-compound and carbohydrate metabolism and
70.03, cytoplasm;

has a best match to a protein annotated with

01.01.03.03.01, biosynthesis of proline and
70.03, cytoplasm.

The annotation of both proteins is similar, but there are a number of categories that are different between them. The differences can be due to precise and more comprehensive biological knowledge of the second protein compared to the first one (01.01.03.03.01), or the result of incomplete annotation (missed category 01.05) assignment to the second protein. A set of non-redundant FunCat subcategories, i.e. 01, 01.01, 01.01.03, etc. could be determined for each annotation. In the above example the common annotation of both proteins consists of the categories 01.01.03.03 and 70.03, which contain 6 subcategories (01, 01.01, 01.01.03, 01.01.03.03, 70, 70.03). There are also two non-common subcategories, i.e. 01.05 and 01.01.03.03.01. The overall score is calculated as

$$\text{A\_score} = \frac{N_c}{N_{\text{tot}}} \quad (3)$$

where N_c is the number of common subcategories and N_{tot} is the number of all subcategories in both proteins. The A_score is $6/8 = 0.75$ in this example. This score measures the protein functional similarity based on a comprehensive functional classification scheme. It was used as a target value (accuracy of prediction) for the development of analyzed approaches.

3 METHODS

A 5-fold cross-validation procedure was used to benchmark algorithms. All descriptor values were normalized on [0,1] interval. The training set for each split included 4/5 of the cases while 1/5 of the samples were used as the test set and were not involved in training. The machine learning methods were trained to predict the functional annotation scores for each pair. After the training, the protein pairs from the test set were scored and the training procedure was repeated using another training and test set. At the end of the 5-fold cross-validation procedure, the scores for the test sets were used to annotate the proteins. First, a protein pair with the maximum calculated score was detected and then the annotation of the prototype protein in the pair was transferred to the target protein. Thus, the procedure of annotation used in our study is similar to annotation using the 'best match' according to sequence similarity, but it was different since the calculated scores were provided by machine learning methods.

Three methods were analyzed. *Multiple Linear Regression* (MLRA) is an *in-house* program developed using the IMSL Fortran library as described elsewhere (Tetko *et al.*, 2006). The *k-Nearest Neighbor Method* (kNN) method programmed *in-house* (Tetko *et al.*, 2006) was applied using Euclidian distance. The number of neighbors, k, was

optimized for eachvalidation split using the corresponding training set. The *Associative Neural Network (ASNN)* represents a combination of an ensemble of feed-forward neural networks and kNN. The ASNN version available on-line at http://www.vcclab.org was used (Tetko *et al*., 2005c). The algorithm calculates correlation between ensemble responses as a measure of distance amid the analyzed cases and performs the kNN (or Parzen window) correction as described elsewhere (Tetko, 2002a, b). Thus, ASNN does kNN in the space of ensemble residuals (i.e. space of models). Neural network ensembles of 100 networks with one hidden layer were used. The number of neurons selected after few preliminary runs was 3 in the hidden layer. Its variation in the range of 2–5 did not affect the performance of the method. We also developed a model using the entire training set (four organisms) to predict proteins from 180 genomes. The *InParanoid algorithm* proposed by Sonnhammer *et al*. (Remm *et al*., 2001) detects in-paralogs, i.e. orthologous groups of genes, which were duplicated after the specification event. In case of one-to-many or many-to-one types of orthology, the InParanoid algorithm assigns confidence bootstrap scores for the in-paralogs. A recent version of the algorithm was obtained from the authors.

To compare methods, we calculated the AC50 value, which corresponds to the annotation accuracy of 50% of the proteins with the highest scores.

4 RESULTS

The accuracy of annotation using several descriptors as scores is shown in Figure 1. The ideal annotation, e.g. when each protein is correctly annotated with all its functional categories, corresponds to a value of 1 for all proteins. However, since we consider only pairs with the significant SW scores (i.e. optscore $\geq$80), there is a fraction of proteins that do not have any corresponding prototype proteins with exactly the same annotation amid all considered pairs. Therefore, the 'achievable' maximum annotation score for such proteins is below 1. The second upper curve indicates this 'theoretically achievable annotation' for our data set. One can notice that ~67% of proteins can be theoretically annotated without any errors with a score '1' in the analyzed data set. The lowest curve ('random annotation') indicates the annotation accuracy (0.27) that can be achieved by annotating a pair of proteins (having a significant SW score) by chance.

The accuracy of annotation increased and then stayed approximately constant at 0.90 even for very high values of the Smith-Waterman optscore. This descriptor calculated the highest accuracy according to the AC50 value. Other descriptors also show strong positive correlation with the functional annotation score. However, each descriptor alone achieved annotation scores above 0.90 only for a small fraction of proteins. The sequences with synteny scores reached the highest accuracy of 0.95 for ~10% of the proteins. This result confirms previous studies indicating the importance of gene neighborhood as an indication of the conservation of functions of protein sequences. However, synteny scores were available for <30% (2206) of the proteins and thus only a small number of sequences could be annotated using this descriptor. The quality of annotation using machine learning methods (Fig. 2) was considerably higher compared to the annotation using optscore. The highest accuracy was calculated using the ASNN method. This method annotated ~45% of the proteins with an average accuracy >0.96.

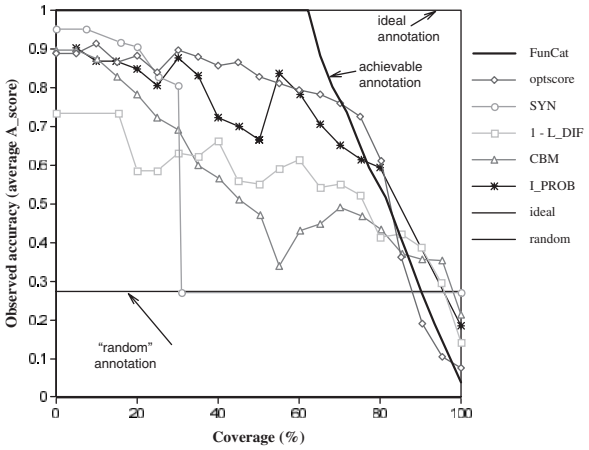

Fig. 1. Annotation scores for MIPS FunCat calculated using different descriptors. For each target protein in the data set we selected a prototype protein with the maximum descriptor value. The functional categories of the prototype protein were used to annotate target proteins and the annotation scores were calculated. These annotation scores were sorted in order of decreasing values of the descriptors. Average annotation scores as a function of the coverage (the averaging was done to cover at least 5% or more proteins with the identical score) are shown in the figure. For example, the average annotation score of 0.89 was calculated for annotation of 5% of proteins with maximum optscores (in the range of 6574–2212, the values are not shown). The higher average annotation score 0.91 was calculated for the next 5% of proteins. The average annotation accuracy 0.73 was calculated for 16% proteins for which a prototype protein with exactly the same length was found (notice a larger bin). Thus, the higher values of descriptors correspond to pairs with higher annotation scores. Notice, that optscore corresponds to the traditional annotation by sequence similarity.

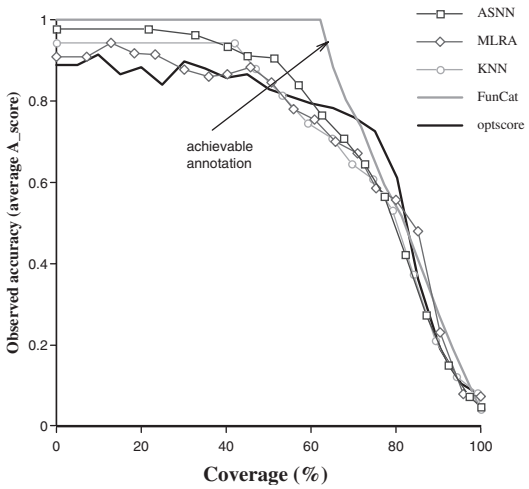

Fig. 2. Automatic functional annotation of BFAB proteins with different machine learning methods. The predictions were sorted in order of decreasing annotation scores (see Fig. 1 for details) and the plot shows the annotation accuracy versus the coverage. The upper green line corresponds to maximal theoretically achievable accuracy of annotation (see Results). The annotation results using optscore are also shown for comparison.

Neural network methods have a long tradition of applications for functional annotation of protein sequence as exemplified by, e.g. works of Brunak and collaborators (Bendtsen *et al*., 2004; Jensen *et al*., 2002, 2003; Nielsen *et al*., 1999). In this work we used an extension of this method, the associative neural networks, which may provide the higher accuracy of the traditional methods due to the bias correction of the ensemble using the kNN method, as described elsewhere (Tetko, 2002a, b).

The SVM using the Radial Basis Function kernel implemented in the libSVM package (Chang and Lin, 2001) was also used as suggested by the reviewers. We performed grid optimization of SVM parameters (width of the kernel, γ, and two parameters, C, ε, controlling the SVM regression) for each cross-validation fold as proposed in the manual of this package and described elsewhere (Tetko *et al*., 2006). The optimized parameters were used to predict the corresponding test sets. This analysis was completed in 34 days and the calculated score AC50 = 0.94 was lower compared to the score AC50 = 0.96 for the neural networks.

InParanoid detected in-paralogs for 5559 genes, including 4307 genes having pairs with the highest confidence score of 100. The calculated AC50 = 0.87 for this method coincided with the score calculated using the optscore only. This result is not surprising considering that the detection of ortholog relations in InParanoid algorithm is done according the sequence similarity, namely the BLAST scores only. Notice, that in Figures 1 and 2 several curves, e.g. the curve for optscore, have accuracies higher than the achievable annotation, i.e. FunCat line, for some ranges of coverage values. Since, there are only 67% with achievable accuracy of 1, no protein with A_score = 1 can be detected in, e.g. coverage range [95%, 100%]. Indeed, only proteins with the lowest A_scores will be in this region. The ordering of proteins according to optscore will ignore information on the accuracy of annotation. Instead, protein pairs with optscores similar to the threshold value of 80 will be collected in this region. It is possible that some of the protein pairs with optscores of 80 will have A_score = 1 for this coverage range. Thus the accuracy of annotation for this coverage range will be higher for protein ordered according to the optscore compared to the proteins ordered according to their achievable accuracy of annotation.

Figure S1 shows that the predicted scores are strongly correlated with the observed accuracy of annotations. In the ideal case both scores should be close to identity. This was the case when predicted and observed scores were considered for all 104 092 pairs. However, since we performed selection of 'best' pairs with highest scores (we selected only 7335 pairs—one per protein from the benchmarking set), we introduced a selection bias. Thus values detected by such selection procedure are 'over optimistic' and may not correspond to the observed annotation values (notice a similar problem for multiple tests in statistics, e.g. the Bonferonni correction or the Gumble distribution in sequence similarity searches). The predicted scores, nevertheless, are well suited to distinguish reliable versus non-reliable predictions. Using the calibration curve shown in Figure S1 one can estimate the expected annotation accuracy for each predicted value. For example, if the maximal calculated score is only 0.5, one can expect annotation accuracy of ca 0.3, i.e. about the same as annotating a protein simply by chance.

Table 1. Annotation accuracy for different groups of descriptors

Descriptors (number)	AC50 ± standard error of the mean	
	only descriptors from col. 1	excluding descriptors from col. 1
All (24)	0.96 ± 0.02	–
Optscore (1)	0.87 ± 0.03[a]	0.96 ± 0.03
Sequence similarity (4)	0.88 ± 0.04	0.95 ± 0.03*
Sequence length attributes (3)	0.69 ± 0.06	0.96 ± 0.03
Synteny scores (1)	0.64 ± 0.07[a]	0.96 ± 0.03
Alignment free sequence similarity (1)	0.70 ± 0.06[a]	0.96 ± 0.03
InterPro domain (8)	0.88 ± 0.04	0.92 ± 0.03*
PEDANT3 properties (7)	0.49 ± 0.07	0.99 ± 0.03
Sequence similarity + InterPro domain (12)	0.95 ± 0.03	0.85 ± 0.05
PEDANT3 properties + InterPro domain (15)	0.93 ± 0.07	0.91 ± 0.05*

[a]No machine learning was used; *significant difference at $P < 0.01$ according to the bootstrap test with 10 000 replicas.

Which groups of descriptors did contribute most to the accuracy of annotation? Table 1 summarizes the prediction accuracy of the ASNN method when excluding different groups of descriptors. The exclusion of InterPro descriptors provided the largest decrease in the performance of the method. Thus the information on the domain composition of proteins represented the strongest signal for the annotation. The sequence similarity descriptors provided the second major contribution. The exclusion of each of the other group only slightly decreased the prediction accuracy of the method.

An alternative method to estimate the importance of groups of descriptors consisted of the development of neural networks using just one or few types of them. The predictors developed using sequence similarity descriptors or InterPro domains had the same AC50 value 0.88. However, the accuracies of either of these predictors were below 0.93 even for the most reliable predictions (data not shown). The combination of sequence similarity scores and InterPro domains dramatically increased the accuracy of the method (which was nevertheless significantly lower, $P < 0.01$, compared to the performance of the method developed using all descriptors). Thus, both InterPro and sequence similarity scores contributed complementary information which was nicely integrated by the ASNN method.

A number of descriptors used for model development were highly interrelated. Which descriptors contribute most to the accuracy of annotation? We addressed this question using so-called neural network pruning methods, which score each descriptor from the training set according to the neuron weights (LeCun *et al*., 1990; Wikel and Dow, 1993) using the ensemble of neural networks (Tetko *et al*., 1996). The descriptor with minimal score is deleted ('pruned') and training process, scoring and pruning is repeated again until all descriptors

are eliminated. Each combination of descriptors was analyzed using 5-fold cross-validation. The pruning procedure and evaluation of each subset of descriptors required about a week using Athlon 3 CPU for one pruning method. The pruning of up to 14 descriptors did not significantly influence the prediction ability of the models, which fluctuated and stayed within $P > 0.05$ of the $AC50_{all}$ calculated with all descriptors.

However, there was a significant drop in the performance (compared to the performance calculated with all descriptors) of models developed with <8–10 descriptors. The minimal sets of descriptors, which provided AC50 values not significantly different compared to the $AC50_{all}$ are summarized in Table S2. The descriptors derived from the InterPro domains dominated amid descriptors that were found as significant with pruning methods. The use of (Wikel and Dow, 1993) method calculated a minimal set of descriptors. It is interesting that there were no PEDANT3 descriptors in this set. The sets of descriptors selected with other two pruning methods included S_EU and DO PEDANT3 properties (Table S2). Thus, in some combinations the PEDANT3 descriptors also provided important contribution to the prediction performance of the method. The difference in the sequence lengths was selected as a significant descriptor for (Tetko *et al.*, 1996) method. Since 24 descriptors is a small number compared to >100 000 pairs, we decided to keep all descriptors in the final model.

We used our algorithm to automatically annotate 180 complete bacterial genomes available in the MIPS database. The proteins from the benchmarking set analyzed in the article were used as prototypes. Thus we propagated the annotation from four genomes to 180 new genomes. Table S1 summarizes the annotation results for new proteins. The developed method annotated 33% of the proteins with an expected accuracy of 0.96. These protein pairs had the largest number of common InterPro domains, high SW score and 17% of them were from syntenic regions (Table S1). A considerable number of proteins, 34%, had predicted annotation score of 0.5 or lower. These proteins had low SW scores (average 100) compared to other proteins in the BFAM set, which were near to the minimum value (80). Less than 10% of these proteins had InterPro domains. It appears that this set enclosed sequences that had functions very different to those observed for the proteins in the BFAB set. The annotation of such proteins without additional experiments could be hardly possible. The other scores corresponded to annotation with an intermediate accuracy. The use of general upper-level FunCat categories increased the prediction ability of the method. For example, for proteins with predicted scores in the range of 0.8–0.96 the expected accuracy increased from 0.82 to 0.86 and 0.84 when FunCat categories were restricted to one (e.g. 01—metabolism, 02—energy) and to two (e.g. 01.01—amino acid metabolism, 01.02—nitrogen and sulfur metabolism) upper-level categories, respectively.

5 DISCUSSION

One of the most important features of the proposed method is its ability to predict the accuracy of annotation. This makes it possible to distinguish between reliable versus non-reliable predictions and to decide whether the current annotation has an acceptable quality or not. To our knowledge, only few other methods, e.g. ProtoMap (Yona *et al.*, 2000), ProtoNet (Kaplan *et al.*, 2004), or annotation of protein function based on family identification (Abascal and Valencia, 2003) also provide a confidence of annotation. These methods, however, mainly explore sequence similarity, while our approach derives annotation scores by integrating different sources of information. A comparison with InParanoid indicated that integration of multiple evidences in new method provided better identification of orthologs compared to the former method, as evidenced by the higher accuracy of annotation of the FUNAT. In principle, the developed approach can be simplistically considered as an ortholog detection tool that is based on machine learning approaches and data integration. For such use the developed system can be applied to any pair of genomes independently whether the analyzed genomes are annotated or not.

The developed system allowed an accurate annotation of ~33% of genes from 180 genomes. With growth of the database containing experimentally verified functional assignment, the number of annotations with high prediction accuracy will increase. The developed system was implemented as a first public server, which provides real-time on-line annotation of new protein sequences with the MIPS FunCat categories. Notice that, e.g. PEDANT provides on-line annotation of sequences from full genomes available in its database only. We have also shown that data integration can provide a significant increase in the performance of methods compared to the use of the only one source of information, thus confirming previous conclusions of (Lanckriet *et al.*, 2004; Troyanskaya *et al.*, 2003).

Manual annotation is difficult and time-consuming work. Annotation using the 'best' match approach is very popular due to the simplicity of interpreting the results: one can always identify the prototype protein used to assign annotation to the target protein. Such annotation method can easily prevent propagation of annotation errors, i.e. in case of annotation miss-assignments one can always trace back their origin. Moreover, as soon as the annotation of the prototype protein has been changed, e.g. due to new experimental facts, the annotation of all target proteins can be also reassigned. An analysis of annotations of proteins, which have high-calculated scores, but different experimental annotations between target and prototype proteins can help to easily detect mistakes in the annotation.

If required, one can easily expand the method by averaging the annotation of k prototype proteins. Such extension can predict new combinations of FunCats, which are not currently present in the database. This analysis, however, is beyond the scope of this study.

The object of our analysis was a pair rather than a single protein sequence and we used the annotation score as the target value for the development of machine learning methods. This allowed us to dramatically decrease the complexity of analysis by using a unified framework for the comparison of sequences. It also allowed the generation of a large training data set for our analysis. To some extent a similar idea to use machine learning to represent protein similarity was used by Paccanaro *et al.* (2006) to cluster protein sequences. Our approach has a number of apparent advantages over the previous studies, where the authors attempted to develop a predictor for each

functional category (multi-class approaches) (Biswas *et al.*, 2002; Clare *et al.*, 2006; Clare and King, 2003; Kretschmann *et al.*, 2001). Firstly, for the development of multi-class approaches one could easily face a problem of inadequate data for some rare categories, which have just few data cases (i.e. just 1–2 proteins per category). Therefore, in some studies only predictors for the FunCat categories of sufficient size were considered (Clare and King, 2003). Secondly, when predicting functions with multi-class approaches, one should be aware of the problem of multiple testing: indeed, theoretically each target protein can be assigned to each of the MIPS FunCat categories and up to 413 classifiers can be tested to assign function. While the number of classifiers can be decreased by incorporating knowledge on, e.g. hierarchical structure of the catalog (Barutcuoglu *et al.*, 2006), the use of multiple classifiers can result in a higher rate of false positive predictions (false positive predictions could be expected for each of the predictions) compared to the use of single predictor. Thirdly, the results of multi-class approaches are more difficult to interpret (i.e. different classifiers were used to annotate different functional categories) and thus can easily contribute to annotation error propagation.

The methodology proposed in this article is free of these limitations and, moreover, it provides an easy generalization across multiple subsets of data, which would be treated separately for the development of multi-class approaches. The approach can be straightforward extended to include new descriptors derived from, subcellular localization (Nakai and Horton, 1999), post-translational modifications (Jensen *et al.*, 2002), microarray data (Mateos *et al.*, 2002) protein fusion (von Mering *et al.*, 2007) or protein–protein interactions (Vazquez *et al.*, 2003) and etc. These descriptors could increase accuracy of inter-genome annotations.

The current study was to specifically develop a prediction system for the annotation of protein sequences using FunCat categories. However, the proposed annotation scheme is not limited to the FunCat only. It can be also applied to provide functional annotation using other functional schemas, e.g. GO categories (Ashburner *et al.*, 2000) or SwissProt (Bairoch *et al.*, 2005) keywords. However, when applying the algorithm to such data a correct selection of the scoring function may dominate in the performance of the method. The FunCat manual annotations used in our study were done by one team of highly skilled scientists and during a relatively short period of time (<6 months). All annotations are consistent and have similar accuracy. A simple scoring function allowed us to achieve high annotation accuracy. The use of inconsistently annotated data, which happens when annotation is contributed by different groups [see e.g. (Frishman, 2007)], may require a search of a different scoring function, the choice of which can significantly contribute to the performance of the method. While such study is definitely challenging, it is far beyond the scope of this article, which has proposed an original and highly accurate approach to predict MIPS FunCat categories for bacterial genomes. Moreover, the functional similarity calculated with our method has built-in measure of accuracy of prediction and allows differentiation of reliable versus non-reliable automatic annotations. The data (Tetko *et al.*, 2005a) as well as results of this study are freely available and can be used for benchmarking of annotation methods by other groups.

The developed approach can be used to perform large-scale annotations of *in-house* data. The users can submit new sequences using the Funat WebServices (see for more details and example of use http://mips.gsf.de/proj/funat/funatclient.html) and retrieve the results in asynchronous mode as soon as annotation will be completed. In case if the analyzed sequence was already pre-calculated, the annotation results will be available immediately. It is also possible to download the ASNN program from http://www.vcclab.org and to develop new models.

ACKNOWLEDGEMENTS

This work was partially supported with German Science Foundation DFG (TE 389/1-1) and 031U212C BFAM (BMFB) grants. The authors thank Volker Stümpflen and Roland Arnold for their assistance to solve various technical problems, Philip Wong for English proofreading of the manuscript and members of MIPS annotation group for their helpful advices.

Conflict of Interest: none declared.

REFERENCES

Abascal,F. and Valencia,A. (2003) Automatic annotation of protein function based on family identification. *Proteins*, **53**, 683–692.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Andrade,M.A. *et al.* (1999) Automated genome sequence analysis and annotation. *Bioinformatics*, **15**, 391–412.

Arnold,R. *et al.* (2005) SIMAP—The similarity matrix of proteins. *Bioinformatics*, **21** (Suppl. 2), ii42–ii46.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.

Azuaje,F. *et al.* (2006) Predictive integration of Gene Ontology-driven similarity and functional interactions. In *Proceedings of IEEE-ICDM 2006 Workshop on Data Mining in Bioinformatics*.

Bairoch,A. *et al.* (2005) The universal protein resource (UniProt). *Nucleic Acids Res*, **33**, D154–D159.

Barutcuoglu,Z. *et al.* (2006) Hierarchical multi-label prediction of gene function. *Bioinformatics*, **22**, 830–836.

Bendtsen,J.D. *et al.* (2004) Improved prediction of signal peptides: SignalP 3.0. *J. Mol. Biol.*, **340**, 783–795.

Biswas,M. *et al.* (2002) Applications of interPro in protein annotation and genome analysis. *Brief Bioinform.*, **3**, 285–295.

Chang,C.C. and Lin,C.J. (2001) LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Cilibrasi,R. and Vitanyi,P.M.B. (2005) Clustering by compression. *IEEE Trans. Inf. Theory*, **51**, 1523–1545.

Clare,A. *et al.* (2006) Functional bioinformatics for Arabidopsis thaliana. *Bioinformatics*, **22**, 1130–1136.

Clare,A. and King,R.D. (2003) Predicting gene function in Saccharomyces cerevisiae. *Bioinformatics*, **19**, II42–II49.

Enright,A.J. *et al.* (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.*, **30**, 1575–1584.

Friedberg,I. (2006) Automated protein function prediction–the genomic challenge. *Brief Bioinform.*, **7**, 225–242.

Frishman,D. (2007) Protein annotation at genomic scale: the current status. *Chem. Rev.*, **107**, 3448–3466.

Frishman,D. and Argos,P. (1997) Seventy-five percent accuracy in protein secondary structure prediction. *Proteins*, **27**, 329–335.

Jensen,L.J. *et al.* (2002) Prediction of human protein function from post-translational modifications and localization features. *J. Mol. Biol.*, **319**, 1257–1265.
Jensen,L.J. *et al.* (2003) Prediction of human protein function according to Gene Ontology categories. *Bioinformatics*, **19**, 635–642.
Kaplan,N. *et al.* (2004) A functional hierarchical organization of the protein sequence space. *BMC Bioinformatics*, **5**, 196.
Kocsor,A. *et al.* (2006) Application of compression-based distance measures to protein sequence classification: a methodological study. *Bioinformatics*, **22**, 407–412.
Kolesov,G. *et al.* (2001) SNAPping up functionally related genes based on context information: a colinearity-free approach. *J. Mol. Biol.*, **311**, 639–656.
Krebs,W.G. and Bourne,P.E. (2004) Statistically rigorous automated protein annotation. *Bioinformatics*, **20**, 1066–1073.
Kretschmann,E. *et al.* (2001) Automatic rule generation for protein annotation with the C4.5 data mining algorithm applied on SWISS-PROT. *Bioinformatics*, **17**, 920–926.
Krogh,A. *et al.* (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J. Mol. Biol.*, **305**, 567–580.
Lanckriet,G.R.G. *et al.* (2004) A statistical framework for genomic data fusion. *Bioinformatics*, **20**, 2626–2635.
LeCun,Y. *et al.* (1990) Optimal Brain Damage. In Touretzky,D. (ed.) *Advances in Neural Processing Systems II (NIPS*2)*. Morgan-Kaufmann, San Mateo, CA, pp. 598–605.
Levy,E.D. *et al.* (2005) Probabilistic annotation of protein sequences based on functional classifications. *BMC Bioinformatics*, **6**, 302.
Lin,D. (1998) An information-theoretic definition of similarity. In *Proceedings of 15th International Conference on Machine Learning*. Morgan Kaufmann: San Francisco, pp. 296–304.
Linding,R. *et al.* (2003) GlobPlot: Exploring protein sequences for globularity and disorder. *Nucleic Acids Res.*, **31**, 3701–3708.
Lupas,A. (1996) Prediction and analysis of coiled-coil structures. *Methods Enzymol.*, **266**, 513–525.
Marcotte,E.M. *et al.* (1999) A combined algorithm for genome-wide prediction of protein function. *Nature*, **402**, 83–86.
Mateos,A. *et al.* (2002) Systematic learning of gene functional classes from DNA array expression data by using multilayer perceptrons. *Genome Res.*, **12**, 1703–1715.
Meinel,T. *et al.* (2005) The SYSTERS protein family database in 2005. *Nucleic Acids Res.*, **33**, D226–D229.
Mewes,H.W. *et al.* (1997) Overview of the yeast genome. *Nature*, **387**, 7–65.
Mewes,H.W. *et al.* (1999) MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.*, **27**, 44–48.
Mulder,N.J. *et al.* (2007) New developments in the interPro database. *Nucleic Acids Res.*, **35**, D224–D228.
Nakai,K. and Horton,P. (1999) PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends Biochem. Sci.*, **24**, 34–36.
Nielsen,H. *et al.* (1999) Machine learning approaches for the prediction of signal peptides and other protein sorting signals. *Protein Eng.*, **12**, 3–9.
Paccanaro,A. *et al.* (2006) Spectral clustering of protein sequences. *Nucleic Acids Res.*, **34**, 1571–1580.
Pearson,W.R. (1996) Effective protein sequence comparison. *Methods Enzymol.*, **266**, 227–258.
Rattei,T. *et al.* (2008) SIMAP structuring the network of protein similarities. *Nucleic Acids Res.*, **36**, D289–D292.
Remm,M. *et al.* (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, **314**, 1041–1052.
Resnik,R. (1995) Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. Montreal, pp. 448–453.
Riley,M.L. *et al.* (2007) PEDANT genome database: 10 years online. *Nucleic Acids Res.*, **35**, D354–D357.
Ruepp,A. and Mewes,H.W. (2006) Prediction and Classification of Protein Functions. *Drug Discov. Today: Tech.*, **3**, 145–151.
Ruepp,A. *et al.* (2004) The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res.*, **32**, 5539–5545.
Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.
Sonnhammer,E.L. *et al.* (1997) Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins*, **28**, 405–420.
Tetko,I.V. (2002a) Associative neural network. *Neural Process. Lett.*, **16**, 187–199.
Tetko,I.V. (2002b) Neural network studies. 4. Introduction to associative neural networks. *J. Chem. Inf. Comput. Sci.*, **42**, 717–728.
Tetko,I.V. *et al.* (2005a) MIPS bacterial genomes functional annotation benchmark dataset. *Bioinformatics*, **21**, 2520–2521.
Tetko,I.V. *et al.* (2005b) Super paramagnetic clustering of protein sequences. *BMC Bioinformatics*, **6**, 82.
Tetko,I.V. *et al.* (2005c) Virtual computational chemistry laboratory - design and description. *J. Comput.-Aided Mol. Des.*, **19**, 453–463.
Tetko,I.V. *et al.* (2006) Benchmarking of linear and nonlinear approaches for quantitative structure-property relationship studies of metal complexation with ionophores. *J. Chem. Inf. Model.*, **46**, 808–819.
Tetko,I.V. *et al.* (1996) Neural network studies. 2. Variable selection. *J. Chem. Inf. Comput. Sci.*, **36**, 794–803.
Troyanskaya,O.G. *et al.* (2003) A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *Proc. Natl Acad. Sci. USA*, **100**, 8348–8353.
Valencia,A. (2005) Automatic annotation of protein function. *Curr. Opin. Struct. Biol.*, **15**, 267–274.
Vazquez,A. *et al.* (2003) Global protein function prediction from protein-protein interaction networks. *Nat. Biotechnol.*, **21**, 697–700.
von Mering,C. *et al.* (2007) STRING 7—recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res.*, **35**, D358–D362.
Wikel,J.H. and Dow,E.R. (1993) The Use of Neural Networks for Variable Selection in QSAR. *Bioorg. Med. Chem. Lett.*, **3**, 645–651.
Wootton,J.C. and Federhen,S. (1993) Statistics of local complexity in amino acid sequences and sequence databases. *Comput. Chem.*, **17**, 149–163.
Yona,G. *et al.* (2000) ProtoMap: automatic classification of protein sequences and hierarchy of protein families. *Nucleic Acids Res.*, **28**, 49–55.
Yu,H. *et al.* (2007) Total ancestry measure: quantifying the similarity in tree-like classification, with genomic applications. *Bioinformatics*, **23**, 2163–2173.