BIOINFORMATICS ORIGINAL PAPER Vol. 24 no. 6 2008, pages 798–806
doi:10.1093/bioinformatics/btn037

Sequence analysis

ConFunc—functional annotation in the twilight zone

Mark N. Wass and Michael J. E. Sternberg*

Structural Bioinformatics Group, Biochemistry Building, Division of Molecular Biosciences, Imperial College London, London SW7 2AZ, UK

Received on May 22, 2007; revised on October 12, 2007; accepted on January 24, 2008

Advance Access publication February 8, 2008

Associate Editor: Dmitrij Frishman

ABSTRACT

Motivation: The success of genome sequencing has resulted in many protein sequences without functional annotation. We present ConFunc, an automated Gene Ontology (GO)-based protein function prediction approach, which uses conserved residues to generate sequence profiles to infer function. ConFunc split sets of sequences identified by PSI-BLAST into sub-alignments according to their GO annotations. Conserved residues are identified for each GO term sub-alignment for which a position specific scoring matrix is generated. This combination of steps produces a set of feature (GO annotation) derived profiles from which protein function is predicted.

Results: We assess the ability of ConFunc, BLAST and PSI-BLAST to predict protein function in the twilight zone of sequence similarity. ConFunc significantly outperforms BLAST & PSI-BLAST obtaining levels of recall and precision that are not obtained by either method and maximum precision 24% greater than BLAST. Further for a large test set of sequences with homologues of low sequence identity, at high levels of presicision, ConFunc obtains recall six times greater than BLAST. These results demonstrate the potential for ConFunc to form part of an automated genomics annotation pipeline.

Availability: http://www.sbg.bio.ic.ac.uk/confunc

Contact: m.sternberg@imperial.ac.uk

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

1 INTRODUCTION

Protein functional annotation is an important task of the genomics era. The ability to obtain rapidly protein and genome sequences has resulted in many proteins whose function has not been experimentally characterized. Further this characterization process is slow compared to sequencing itself, resulting in a need for approaches to predict protein function in order to obtain accurate annotations. The number of sequences requiring annotation makes it important that such methods are automated, enabling them to annotate whole genomes without human intervention.

The simplest approach for predicting protein function is sequence searching to identify homologues with known annotation. The accuracy of directly transferring annotations from a sequence of known annotation to a homologue of unknown function was initially investigated by Hegyi and Gerstein (1999). They demonstrated that annotation transfer among enzymes, using the E.C. enzyme classification, is successful where sequence similarity is high and that sequences with low levels of identity are likely to have different functions making annotation transfer unreliable. Subsequent studies (Devos and Valencia, 2000; Todd *et al*., 2001; Wilson *et al*., 2000) of this relationship suggests that complete function (all four E.C. digits) is conserved at high-sequence identity, three E.C. digits are also likely to be conserved down to 40%. Below this level function is often different. More recent analyses (Rost, 2002; Tian and Skolnick, 2003) show that functional divergence can occur at higher levels of sequence identity, between 60% (Tian and Skolnick, 2003) and 70% (Rost, 2002), suggesting that functional transfer between homologues with levels of identity lower than these may be inaccurate. These studies demonstrate that while annotation transfer is useful, it is limited and other approaches are required for the effective prediction of protein function. However, the use of sequence searching programs such as BLAST (Altschul *et al*., 1990) and PSI-BLAST (Altschul *et al*., 1997) remains a common first step for inferring protein function.

An alternative approach is the comparison of sequences with motif- or domain-based resources such as PFAM (Finn *et al*., 2006) or Interpro (Mulder *et al*., 2005). PFAM contains multiple sequence alignments and hidden Markov models that represent domains or families of proteins. Matches to PFAM can infer protein family or domain. Function can also be inferred by mapping the functions present within the domain or family to a query sequence.

The development of Gene Ontology (GO) (Ashburner *et al*., 2000) has enabled the classification of both enzyme and non-enzyme functions, its directed acyclic graph structure effectively provides a functional hierarchy moving from general to specific terms. A number of sequence similarity-based methods utilize GO to predict functional annotations. The earliest methods use BLAST to identify homologues with known GO annotations and weight the GO terms according to the BLAST *e*-values of the sequences they are associated with (Groth *et al*., 2004; Khan *et al*., 2003; Zehetner, 2003). Martin *et al*. (2004) used a similar approach for Gotcha. GOtcha utilizes the GO structure to combine the *e*-values from all BLAST homologues to make predictions for individual GO terms each of which is associated with a confidence score. PFP (Hawkins *et al*., 2006) uses a similar approach to GOtcha by making predictions based

*To whom correspondence should be addressed.

upon the frequency of GO terms within a set of PSI-BLAST hits.

Phylogenomics approaches have also been used to predict protein function. SIFTER (Engelhardt *et al.*, 2005), for example takes a PFAM protein family and generates a reconciled phylogenetic tree and uses a statistical model of protein function evolution to infer annotations for the unannotated sequences in the family.

Early work by Hannenhalli and Russell (2000) used functional residues to aid function prediction. Their approach assigns enzymes of a known class to a sub-class by generating hidden Markov models to extract subfamily-specific functional sites which are then used to assign protein function. This approach requires knowledge of protein families in order to infer enzyme functional subtypes. George *et al.* (2005) have used the Catalytic Site Atlas (Porter *et al.*, 2004), which is a manually curated database of enzyme catalytic residues, to predict enzyme function using protein sequence.

We have previously demonstrated the ability to use Position Specific Scoring Matrices (PSSMs) to predict protein molecular function using GO. Phunctioner (Pazos and Sternberg, 2004) uses protein structural alignments from which PSSMs are generated for each potential GO term present among the initial protein structures used. A query protein is then scored against each PSSM to predict its function. As Phunctioner relies upon structural alignments it is limited by structural space. Here we demonstrate a general approach, ConFunc, similar to Phunctioner that is applicable to the more extensive sequence space, which could prove an effective tool for genome annotation. ConFunc is available for academic use as a web server at http://www.sbg.bio.ic.ac.uk/confunc

ConFunc uses GO to direct the function prediction process, by splitting sets of sequences identified by PSI-BLAST into sub-alignments according to their GO annotations. Each GO term sub-alignment is then used to identify conserved residues within that group, for which a PSSM profile is generated. This combination of steps produces a set of feature- derived (i.e. GO annotation) profiles from which protein function is predicted. Many methods that predict functional residues use phylogenetics approaches (Aloy *et al.*, 2001; Berezin *et al.*, 2004; Lichtarge *et al.*, 1996) to group homologous sequences. The power of ConFunc is that the grouping of sequences by GO annotation not only enables the identification of conserved residues potentially associated with a particular function but further enables them to then predict protein function. The use of GO makes it possible to predict a full range of protein functions and is not limited to enzyme function as other methods utilizing functional residues are (George *et al.*, 2005; Hannenhalli and Russell, 2000).

As direct transfer of function from a homologue is ineffective when sequences have low levels of identity, it is important that alternative methods can perform well in such cases. The performance of ConFunc has therefore been assessed for a set of protein sequences where homologues above 30% sequence identity have been removed to simulate this scenario. Initially we impose a further constraint to assess ConFunc performance in the twilight zone (Rost, 1999) by only using sequences in the test set for which the top hit has a BLAST *e*-value greater than 1×10^{-20}. ConFunc performance is also assessed for all sequences in the test set.

ConFunc performance is compared to the predictions of annotation transfer from the top BLAST and PSI-BLAST hits. Annotation transfer using BLAST is a common first step for inferring protein function, so this comparison importantly provides an assessment of the ability of this approach to predict protein function at low levels of sequence identity. Comparing the performance of the methods at this level of sequence identity also removes any bias that BLAST and PSI-BLAST might have due to the potential use of sequence similarity by curators when assigning annotations (for example, annotations with the Inferred by sequence similarity (ISS) evidence code). A limited comparison with PFAM is also performed because it was not possible to simulate a scenario where sequences with greater than 30% sequence identity with each query are removed from PFAM and therefore PFAM predictions have the advantage of using sequences with greater than 30% identity.

2 METHODS

The ConFunc method is outlined in Figure 1. Homologues of a query sequence are identified by running up to three iterations of PSI-BLAST against Swiss-Prot (Wu *et al.*, 2006). Sequences identified by PSI-BLAST that also have EBI GO annotations [not of evidence type IEA (inferred by electronic annotation) or NR (no record)—see GO annotations below for details] are extracted and their full length Swiss-Prot sequences are aligned using MUSCLE (Edgar, 2004). All sequences with greater than 30% identity with the query sequence are removed from the analysis to assess performance at low levels of sequence identity. The aligned sequences are then grouped according to their GO annotations, resulting in sub-alignments representing each of the GO terms present in the set of homologous sequences, which are then used to determine the predicted function of the query sequence. Only terms with three or more homologous sequences are used for prediction purposes to ensure that a good signal is obtained from the profiles.

2.1 Identification of conserved residues

For each GO sub-alignment, residue conservation scores are calculated using a Vingron-type sequence weighting method (Valdar, 2002; Vingron and Argos, 1989). This ensures that similar sequences are not over-represented in the calculation of residue conservation. Each sequence is weighted according to the average distance between it and the other sequences in the sub-alignment. The weighting of sequence *i* in a group of *n* sequences is

$$w_i = \frac{1}{n-1}\sum_{j \neq i}^{n} 1 - d(i,j)$$

where $d(i,j)$ is the distance between sequences *i* and *j*, which in this case is their sequence identity. Identical sequences will have a distance of one and receive a smaller weighting in the calculation, while two sequences with no identity will have a distance of zero and a higher weighting. The conservation of a position *x* in a sub-alignment of *n* sequences is given by

$$C_x = \frac{1}{\sum_i^n \sum_{j>i}^n w_i w_{\mathrm{j}}} \sum_i^n \sum_{j>i}^n w_i w_j \mathrm{sub}(i_x, j_x)$$

where sub (i_x, j_x) is the value from the BLOSUM62 substitution matrix (Henikoff and Henikoff, 1992) for the residues at position *x* in

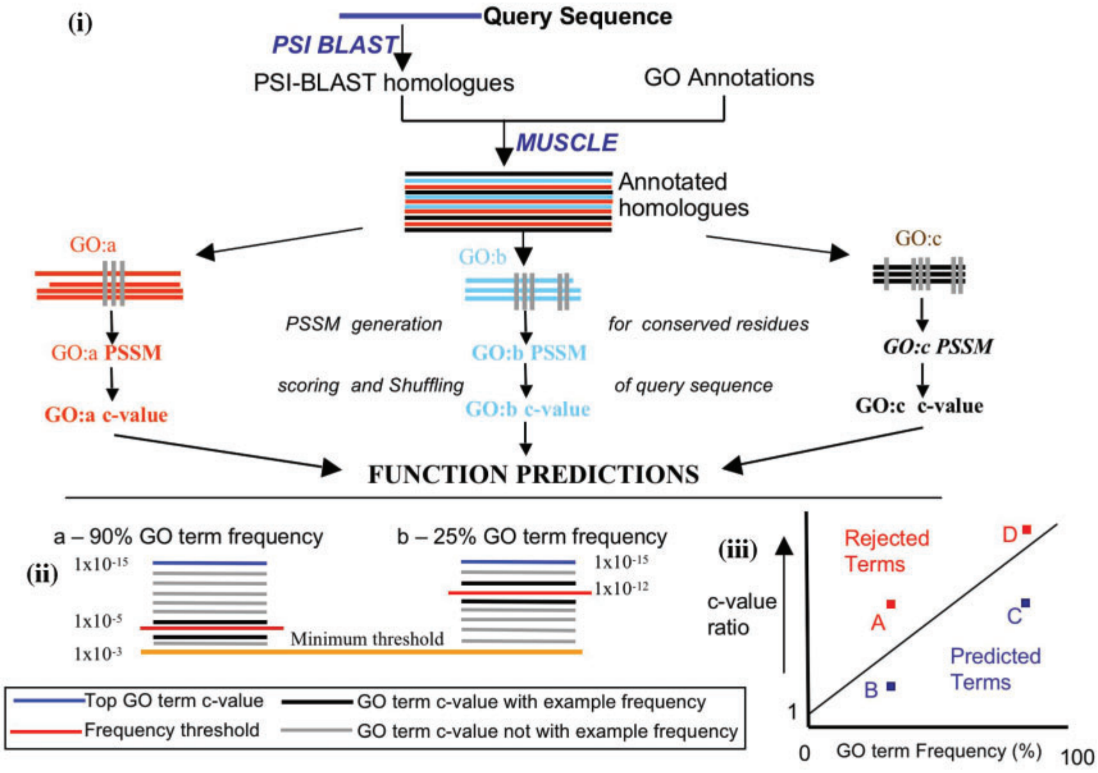

Fig. 1. Outline of the ConFunc method. (**i**) Schematic representation of the ConFunc method. (**ii**) Example of ConFunc threshold for GO term *c*-value. The diagram shows 2 cases of a set of GO term *c*-values for a query sequence. The top *c*-value (lowest *c*-value) is shown in blue. The threshold for acceptance of each *c*-value (expectation value) is determined by its frequency within the set of homologues identified by PSI-BLAST and the top GO term *c*-value. Cases a and b show the threshold for GO terms present in 90 and 25% of the annotated sequences respectively. In each case the GO term above the threshold line is accepted as a predicted function of the sequence and the term below the line is rejected. (**iii**) schematic demonstrating how the *c*-value ratio and GO term frequency are combined.

sequences *i* and *j*. To identify the conservation of each position compared to all other positions within the sub-alignment a Z score is calculated for the conservation at each position as

$$Z_x = \frac{C_x - \overline{C}}{\sigma}$$

where C_x is the conservation score at position x, $\bar{C}$ is the average conservation value of all the positions in the sub-alignment and σ is the standard deviation. All residue positions with a Z score greater than a given threshold are considered to be functionally important residues and used for the scoring of the GO term PSSMs (see below) against the query sequence.

2.2 Generating GO term PSSMs

PSSMs are generated for each sub-alignment using the same method as PSI-BLAST (Altschul *et al.*, 1997). The query sequence is scored against each sub-alignment at only the positions that have been identified as conserved and have a conservation Z score greater than the threshold. The score S for a sequence against a PSSM is

$$S = \sum_{i=1}^{n} P_{ik}$$

where P_{ik} is the value in the PSSM for residue k at position i. To test the statistical significance of the score, an expectation value is calculated for each PSSM score. Expectation values are calculated by fitting the scores from the shuffled sequences to an extreme-value distribution using maximum likelihood fitting as described by Eddy (maximum likelihood fitting of extreme value distributions are available from http://selab.janelia.org/publications.html). Using the Kolmogorov–Smirnov test for a subset of the sequences the data was found to have a good fit to the extreme value distribution at the $P = 0.001$ level.

2.3 Using feature derived scores to predict function

We have described the process that generates GO term-specific expectation values. To avoid confusion with BLAST *e*-values, GO term specific expectation values will be referred to as *c*-values. The GO term specific *c*-values are used to determine which functions are predicted and they therefore discriminate between correct and incorrect terms. A simple threshold is used to initially remove any terms that have poor *c*-values (greater than 1×10^{-3}). However, using this single threshold is not sufficient for accurate prediction because of differences in GO term *c*-values between sequences and the high ratio of non-annotated terms to annotated terms present in the pool of potential GO terms that can be predicted. To discriminate better between correct and incorrect terms, an additional threshold is used that relates each GO term *c*-value to the *c*-value of the top GO term and the frequency of each GO term within the set of homologues present in the complete alignment. This threshold is different for each GO term and each query sequence, but it is never allowed to be greater than the initial minimum threshold. A schematic of this threshold is shown in Figure 1 part ii. All potential GO terms for a query are sorted by *c*-value. The top term is predicted as a function of the query and the prediction of all other

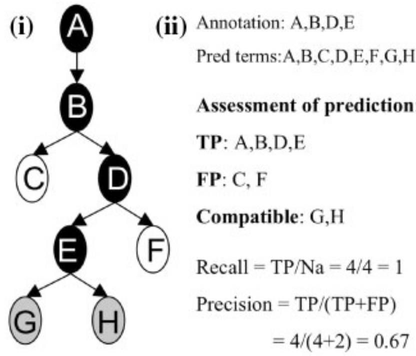

Fig. 2. Assessing GO Function Prediction using recall and precision. (**i**) Schematic GO annotation. Each circle represents a GO term, black circles are terms for which the query sequence is annotated and white circles represent terms that are not annotated for the query. Light grey circles represent GO terms that are compatible to the annotation, i.e. they are more specific than the most specific annotation which in this case is term E. More specific terms of intermediate annotations are not compatible as shown in this example by terms C and F. (**ii**) The GO terms predicted by a method are displayed and the number of true positives (TP), false positives (FP) and compatible terms are shown and used to calculate recall (Na is number of annotations) and precision for the prediction.

terms is determined by the ratio of their *c*-value and that of the top term and the frequency of that GO term. Figure 1 part iii shows a schematic graph of how these values are combined. GO term *c*-value ratio and frequency are plotted on the graph and a division (line) separates those terms which are predicted as functions of the query and those which are rejected. The terms shown in Figure 2 part iii represent those in part ii of the figure, showing that a term present in 25% of sequences (A,B) must have a smaller *c*-value ratio to be inferred as a function of the query than a term present in 90% of sequences (C,D). Simply the fewer sequences present for a GO term the closer its *c*-value must be to the top value to be inferred as a function of the query sequence. Conversely terms represented by a greater number of sequences can have *c*-values that are further from the top term (Fig. 1 part ii). The settings of this threshold are varied in the analysis of ConFunc to obtain results for a full range of performance.

The variation of *c*-values described above can be demonstrated by the difference between *c*-values obtained for related GO terms. Functionally specific GO terms (distant from the root) are generally present in fewer sequences in the set of homologues for a given query, while more general functional terms have a much higher frequency. Specific functions often have much smaller *c*-values than those of related (i.e. on the path from the specific term to the root node) more general terms. Incorporating the frequency of each GO term within the set of homologous sequences resolves this issue.

For each of the analyses discussed, ConFunc has been run using a PSI-BLAST *e*-value threshold of 1×10^{-8}. ConFunc has been run with a conservation *z*-score threshold of 1.5 and a maximum GO term *c*-value threshold of 1×10^{-3}. On average, 8% of residues are above this conservation threshold, and in 79% of sequences less than 10% of residues are above the threshold.

2.4 Gene ontology annotations

The EBI (European Bioinformatics Institute) gene ontology annotations (GOA) released in April 2005 were used as a source of GO annotations for sequences in Swiss-Prot. Each GO annotation is associated with an evidence code describing the source used for inferring the annotation and therefore gives an indication of the confidence of annotation. For example, annotations with evidence codes determined from traceable author statement (TAS), experiment (Inferred from direct assay IDA) or organism mutant phenotypes (Inferred from mutant phenotype IMP) have greater reliability than those for which there is no record (NR) of how the annotation was generated or those that are electronically inferred (IEA). To ensure that confident predictions are made, all IEA and NR evidence code annotations are excluded from the prediction process. The EBI IEA GO annotations are generated by mapping annotations from other data sources, including Swiss-Prot keywords and E.C. To assess further the performance of ConFunc and BLAST, their predictions are also compared with the full set of annotations, which only excludes annotations with an NR evidence code.

2.5 Protein sequence test set

Swiss-Prot (release 47) was used to generate a protein sequence test set. All sequences with only IEA or NR GO annotations were removed. Further any sequences labelled as fragments in Swiss-Prot were also removed as were any containing 'X' in place of a residue. A non-redundant test set of GO annotated sequences was generated from the remaining sequences, using CD-HIT (Li *et al.*, 2002) at 40% identity. Finally, any remaining sequences for which no GO annotated homologues were identified by three iterations of PSI-BLAST were also removed, resulting in a test set of 7150 sequences.

To assess performance in the twilight zone of sequence similarity, initially only sequences in the test set that have a top BLAST hit greater than 1×10^{-20} and for which all three methods make function predictions are considered. This set considers 1675 sequences from the full test set.

The full test set (7150 sequences) includes proteins with annotations from all the main functional categories in the GO molecular function component. Catalytic activity and binding functions are the largest categories and account for 27 and 34% of the annotations, respectively. Signal transduction, transcription regulation and transporter functions represent 12, 8 and 7% of the annotations in the test set, with the final 12% of annotations split between the remaining molecular function categories.

2.6 Comparison with BLAST and PSI-BLAST

The performance of ConFunc has been compared to the annotations predicted by the top BLAST and top PSI-BLAST hit for each query sequence against Swiss-Prot. The non-electronic set of GO annotations does not provide annotations for all sequences in Swiss-Prot, so where the top hit is not annotated, the first annotated hit is accepted. All sequences with greater than 30% sequence identity to the query are removed. To assess the range of performance obtainable by BLAST and PSI-BLAST the *e*-value cut off for inclusion of each top hit is varied between 0.1 and 1×10^{-100}. For example, with an *e*-value cut off of 1×10^{-10} predictions are only made for sequences that have a GO annotated homologue identified by BLAST (or PSI-BLAST) with an *e*-value of less than 1×10^{-10}.

2.7 Comparison with PFAM

PFAM (release 17) was obtained. For each query sequence the PFAM mysql database was queried to identify all significant hits. Hits were converted to GO annotations using the pfamToGo mapping file downloaded from the Gene Ontology website in April 2005. The PFAM analysis is not directly comparable to the ConFunc and BLAST analyses because the PFAM alignments and HMMs were generated without using a 30% sequence identity threshold as has been done for ConFunc and BLAST (see Section 2.6). Like the BLAST analysis the

PFAM *e*-value threshold for inclusion is varied to obtain a range of results.

2.8 Assessment of results

Numerous approaches have been proposed for assessing GO protein function prediction methods. Lord *et al.* (2003) proposed the use of semantic similarity to compare predictions with annotations. Schlicker *et al.* (2006) have also devised scores related to semantic similarity. Others have used recall and precision (Jones *et al.*, 2005) and variations of recall and precision that address the hierarchical nature of GO (Verspoor *et al.*, 2006). A protein annotated with a GO term is also annotated with all parents of that term, here we use this relationship to calculate recall and precision over all levels of Gene Ontology (see Fig. 2 and Supplementary Material). Throughout we refer to existing annotations as annotations and the functions inferred by each method as predictions. Our calculation of recall and precision considers the parent terms of each annotation and compares these with the predictions made at each level as shown in Figure 2 part ii, with recall and precision defined for the test set as

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad \text{Recall} = \frac{TP}{N_A}$$

where TP and FP are the total number of true positive and false positive predictions respectively and N_A is the total number of annotations in the test set.

Assessing function prediction is complicated by the complex nature of protein function, unlike the comparison of protein structure predictions with the coordinates of a protein structure. While the known coordinates of a protein structure fully describe its structure, it is possible that functional annotations do not fully describe protein function. Annotations can be too general, with a general function assigned to a protein when a more specific related functional term better describes the function. Proteins (especially for multi domain sequences) may also have more functions than those they are annotated with. It is therefore possible that predictions that are more specific than existing annotations and even those which are completely different from existing annotations are correct. To account for these potential issues, functions that are compatible to the correct annotation are not considered in the calculation of recall and precision. We class GO terms that are descendents of the most specific annotation of each protein as compatible (Fig. 2 and Supplementary Information). Accepting terms that are descendents of intermediate nodes in the annotation would allow too many different GO terms to be accepted as correct (see Supplementary Information).

3 RESULTS

3.1 Functional annotation in the twilight zone

Our aim in developing ConFunc is to augment the ability to predict function using other methods, including approaches as simple as annotation transfer. To assess the ability of ConFunc to predict function where annotation transfer may be more limited, we initially consider a subset of the test set, only taking into account query sequences for which all three methods make predictions, where the top annotated hit has a BLAST *e*-value greater than 1×10^{-20} in addition to the removal of all sequences with greater than 30% sequence identity. This results in a set of 1675 sequences from the original test set of 7150 sequences. These settings assess the ability of ConFunc to predict function in the twilight zone of sequence similarity.

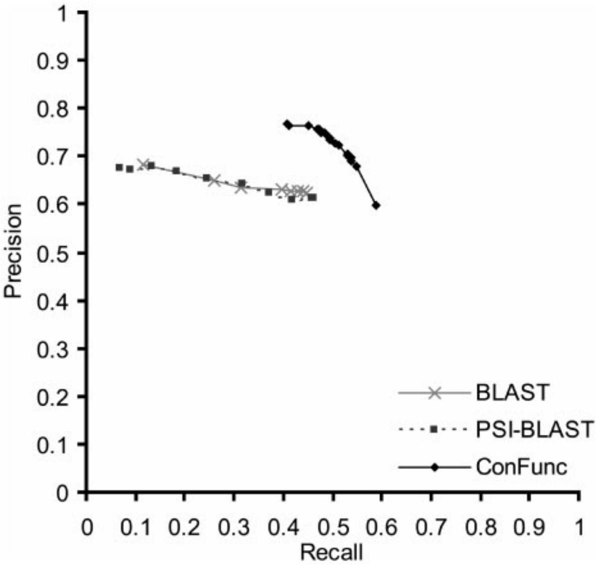

Fig. 3. Assessing function prediction in the twilight zone. Recall and precision analysis using non-electronic annotations. The recall and precision obtained by ConFunc, BLAST and PSI-BLAST for sequences with top BLAST hit e-value greater than 1×10^{-20}. Predictions are compared to non-electronic annotations.

The results of this analysis are displayed as a Precision-Recall graph (Fig. 3). Precision-Recall graphs provide a good assessment of the performance of methods where the class distribution is skewed (Davis and Goadrich, 2006). In this case, the number of annotations is much smaller than the number of potential functions that can be assigned. A perfect predictor would be represented by a point at 1,1 on a Precision–Recall graph, i.e. predicting all annotations without making any false predictions. Therefore the better a predictor, the closer it will be to the top right corner of the graph.

The Precision–Recall graph shows that ConFunc outperforms annotation transfer by both BLAST and PSI-BLAST. ConFunc obtains precision as high as 0.77 compared to a maximum of 0.68 for BLAST. Importantly as shown in Figure 3, ConFunc recall is 0.41 compared to 0.12 for BLAST at these levels of precision. BLAST is able to obtain a comparable recall of 0.41 but only with precision of 0.62. This analysis highlights that ConFunc functional predictions are of greatest value in the twilight zone of sequence similarity.

3.2 Predictions assessed against non-IEA annotations

The previous section assessed ConFunc performance in the twilight zone of sequence similarity. This further analysis considers the full test set of sequences and continues to use the 30% sequence identity threshold. The results of this analysis are shown in Figure 4. ConFunc makes predictions for 4844 sequences in the test set and performs better at higher levels of precision than both BLAST and PSI-BLAST (Fig. 4). The statistical significance of this difference in performance has been tested using the McNemar test (McNemar, 1947), which considers the number of misclassifications in each method that are classified correctly in the other. Individual results are

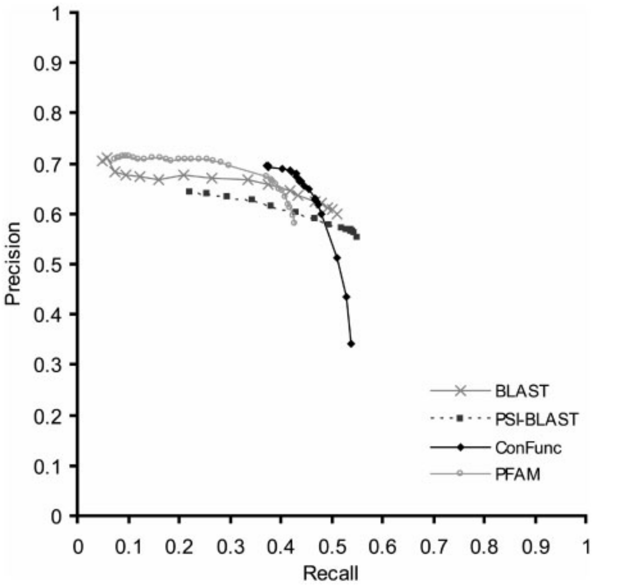

Fig. 4. Recall and precision analysis using non-electronic annotations. The recall and precision obtained by ConFunc, BLAST and PSI-BLAST when predictions are compared to non-electronic annotations.

compared separately with the most precise ConFunc result compared with the BLAST results that have the closest recall and precision to this ConFunc result. The result is significantly different from BLAST using a 1×10^{-10} *e*-value threshold (closest recall) at a $P = 0.001$ level and also significantly different from BLAST performance using a 1×10^{-100} *e*-value threshold (closest precision) at the same $P = 0.001$ significance level. At this level of precision, ConFunc recall is more than six times greater than BLAST and BLAST coverage (percentage of test set that predictions are aid for) is reduced to 6% compared to 68% for ConFunc.

Figure 4 shows that BLAST generally outperforms PSI-BLAST, this occurs because a GO annotated homologue is not identified by BLAST for some query sequences whereas one or more are retrieved by PSI-BLAST. In such cases, PSI-BLAST often identifies a remote homologue of the query sequence and transfers its annotation, which may account for the difference in performance. At lower levels of precision, BLAST and PSI-BLAST obtain greater recall than ConFunc at equivalent precision. We consider high precision to be more important than high recall; it is preferable to have a smaller set of mainly correct annotations than a large set of annotations with a high proportion of errors. The acceptance of many false positive predictions can result in the accumulation of annotation errors in databases, which often propagate (Brenner, 1999; Devos and Valencia, 2001).

While the recall values obtained are low, all homologues with greater than 30% identity with each query sequence have been removed and as such overall performance is poorer compared to a standard case where no homologues have been removed. Further recall is calculated over the complete test set of 7150 sequences and using an *e*-value threshold of 0.1, BLAST makes predictions for 6157 sequences (PSI-BLAST makes predictions for the complete set) whereas ConFunc only makes predictions for 4844 sequences (68%). If only the sequences in the test set for which all three methods make predictions are considered (see Supplementary Material), ConFunc obtains greater recall ranging between 0.52 and 0.75 compared with 0.37–0.54 over the full test set. This demonstrates that ConFunc is able to obtain high levels of both recall and precision.

In this analysis, ConFunc performance is also compared with PFAM-based function predictions. PFAM is a hand curated set of sequence alignments and Hidden Markov Models and as such has not been generated with the 30% sequence identity requirements that have been used in both the ConFunc and BLAST benchmarking. This complicates the comparison between the methods. While PFAM function predictions outperform BLAST and PSI-BLAST at most settings, this difference is small and the improved performance is likely to be due to the presence of close homologues in the PFAM alignments that have been used to make the predictions. The comparison of ConFunc and PFAM is more important and it demonstrates that despite the inherent advantage of PFAM, it is not able to perform better, with ConFunc obtaining greater recall at equivalent levels of precision (Fig. 4). At the highest level of precision, PFAM obtains 0.72 precision with recall of 0.10 compared to 0.70 precision and 0.37 recall for ConFunc. Like BLAST, at this level of precision the coverage of PFAM predictions is low at 15% comapred to 68% for ConFunc.

As the PFAM analysis does not simulate a low sequence identity scenario, it might be surprising that the recall ranges obtained are similar to BLAST and ConFunc (Fig. 4). This is likely to be caused by the variation in function within PFAM families as family members often share similar general functions with various different specific functions (e.g. enzyme substrate specificity) (Abhiman and Sonnhammer, 2005), and in many cases the mapping of PFAM to GO is only able to assign general functions.

3.3 Assessment of predictions with IEA annotations

In the previous analysis, non-electronic GO annotations were used for the ConFunc prediction process to generate GO term specific sub-alignments and subsequent *c*-values. They were also used for the annotation transfer predictions made by BLAST and PSI-BLAST. The predictions made were compared to these non-electronic annotations. In this analysis we continue to use the non-electronic annotations for the prediction process, as there is greater confidence in their reliability. However, while non-electronic annotations are used for the predictive process, it is possible to use electronic annotations for the testing of these predictions. In this case, the non-electronic annotations can be considered as a learning set and the electronic annotations the test set. This approach of using a set of highly confident annotations for the predictive process and comparing the results with a large set of annotations from more varied sources with different reliability has already been used in the assessment of SIFTER (Engelhardt *et al*., 2005), where experimental annotations from the Gene Ontology Annotation (GOA) database (Camon *et al*., 2004) (annotations with evidence codes IMP and IDA) are used to make predictions, which are then compared to the full set of annotations present in the GOA database.

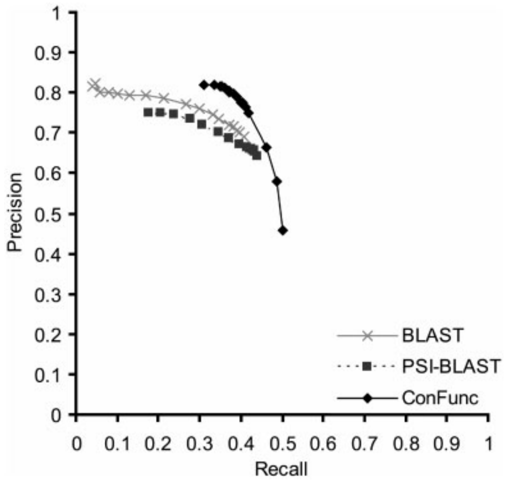

Fig. 5. Recall and precision analysis using electronic annotations. The recall and precision obtained by ConFunc, BLAST and PSI-BLAST when predictions are compared to electronic annotations.

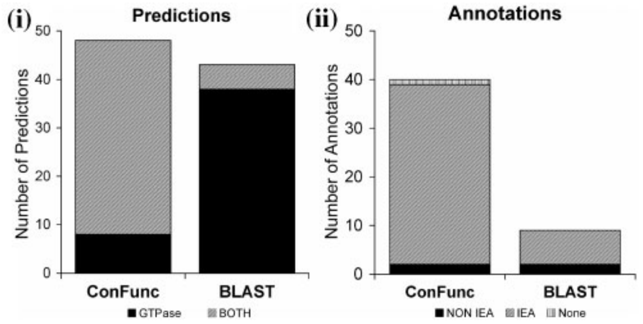

Fig. 6. Predictions of GTPase functions. (**i**) GTPase and GTP binding function predictions made by ConFunc and BLAST. GTPase predictions without GTP binding predictions are shown in black and sequences where both GTPase and GTP binding are predicted are shown in grey. (**ii**) The annotations of the GTPase sequences predicted by ConFunc and BLAST. In black are the sequences that have both a GTPase and GTP binding annotation in the non-IEA set, while those with both these annotations in the electronic annotations are shown in grey.

Figure 5 shows the range of recall and precision obtained by ConFunc, BLAST and PSI-BLAST when compared with this extended set of annotations. There is an increase in precision in all cases (compared with performance against non-electronic annotations Fig. 3), indicating agreement between the electronic annotations generated from mappings of other functional annotation types (e.g. Swiss-Prot keywords and E.C.) and the predictions made by ConFunc and annotation transfer. A reduction in recall is also observed in comparison to the non-electronic annotation results because the increase of the total number of annotations present in the test set is greater than the increase of the true positive predictions. A clear difference between the predictive performance of ConFunc and BLAST is observed with ConFunc obtaining greater recall than BLAST at all levels of precision. The McNemar test was used to test the significance between the most precise ConFunc result and the BLAST results with equivalent recall and precision, ConFunc is significantly better in both cases at the $P = 0.001$ level.

3.4 Differences between non-electronic and electronic GO annotations

The electronic annotations provide a much larger set of annotations to compare predictions against. They result in performance differences for ConFunc, BLAST and PSI-BLAST when used for testing compared to using non-electronic annotations. Are the differences in performance observed due to the agreement of incorrect predictions with incorrect electronic annotations, or are correct predictions being made that are simply missing from the non-electronic set? We use the example of GTPase enzymes to demonstrate that a large source of these differences is due to the latter case; correct predictions are being made that are not present in the non-electronic annotations. GTPases (GO:0003924) hydrolyse GTP to GDP. They should therefore be annotated with this catalytic function and also with the GTP binding (GO:0005525) function. However, very few sequences annotated as GTPases are also annotated with the related binding annotation in the non-electronic set of annotations (Fig. 6 and Supplementary Material), demonstrating incompleteness in the non-electronic set of annotations. Most of these sequences are annotated with the binding function when electronic annotations are included (Fig. 6 and Supplementary Material).

While it might be clear to someone using these annotations that a GTPase is likely to bind GTP, this difference in annotation will have a greater effect upon the perceived performance of a function prediction algorithm. Predicting a GTP binding function for a GTPase that is not annotated with this binding function will be classed as a false positive prediction, therefore reducing its performance. For the majority of GTPase predictions ConFunc also predicts GTP binding (40 out of 48 see Fig. 6i). Only two of these sequences have non-electronic GTP binding annotations (Fig. 6ii) and as a result 38 of the 40 are classed as false positive predictions. The electronic annotations include GTP binding for a further 37 of these sequences, so only one of the GTP binding annotations is classed as a false positive when compared to this annotation set (Fig. 6ii). A similar pattern is observed for BLAST and PSI-BLAST predictions; eight GTP binding predictions are made for GTPase sequences, only one of these is present in the non-electronic set while the remaining seven are all present in the electronic annotations (Fig. 6). This pattern has been observed for other types of enzyme (data not shown) and shows that the increased precision obtained when electronic annotations are included in the assessment of predictions is due to the prediction of correct functional terms that should be present in the non-electronic set.

The analysis of predictions for GTPases also illustrates a difference between ConFunc and BLAST predictions. Both methods predict the GTPase function for a similar number of sequences (Fig. 6i), and ConFunc predicts the GTP binding function for the majority of GTPase sequences, while BLAST predicts GTP binding in very few cases. This occurs because BLAST transfers the annotation of the top hit which appears to be often a sequence annotated as a GTPase but not as a GTP binding protein, whereas ConFunc assesses all of the GO terms present in the set of PSI-BLAST homologues, giving it the potential to predict more and different terms than those present in the top BLAST hit. It also results in more false positive predictions for ConFunc (when compared to non-electronic annotations), as BLAST only predicts eight GTP binding functions compared to 37 by ConFunc.

4 CONCLUDING REMARKS

We have developed ConFunc, which uses Gene Ontology to direct the function prediction process. Our analysis has assessed the ability of ConFunc, BLAST and PSI-BLAST to predict protein function at low levels of sequence identity. ConFunc provides the greatest improvement over BLAST in the twilight zone of sequence similarity, obtaining levels of precision not obtained by either BLAST or PSI-BLAST. ConFunc also obtains greater recall than BLAST under such conditions, demonstrating the advantage of ConFunc when close homologues do not exist. ConFunc performs well under such conditions because it combines the annotations and sequence information present in all the distant homologues identified by PSI-BLAST to make functional predictions, whereas BLAST simply transfers the annotation of the top hit. As the number of sequences without close homologues of known function increases, it is important that this analysis has been performed at low levels of sequence identity showing that ConFunc can annotate proteins under such conditions.

Further, benchmarking for a larger set of sequences demonstrates that ConFunc outperforms annotation transfer by both BLAST and PSI-BLAST and, at high levels of precision, ConFunc is able to obtain over six times greater recall than BLAST. Our analysis also demonstrates that ConFunc outperforms PFAM-based function predictions, even though PFAM has the advantage of not removing sequences with greater than 30% sequence identity to the query sequences in our test set. As previously discussed, PFAM obtains low levels of recall because it is often only able to make general GO function predictions.

Annotation transfer by BLAST and PSI-BLAST are limited by their ability to identify the closest homologue of a query protein. While ConFunc utilizes PSI-BLAST results it does not rely on the ability of the search method to identify the closest homologue, but on its ability to identify a group of homologous proteins which represent a pool of potential GO terms that can be assigned to a query sequence. Further each of these GO terms has an associated *c*-value and frequency, which can be used to give an indication of the confidence in prediction of each individual term. It would be desirable to compare ConFunc with other recently developed function prediction methods such as (Engelhardt *et al*., 2005; Martin *et al*., 2004). However, the setup of these systems often makes it difficult to ensure that query sequences are not used within the predictive process, a problem encountered by Engelhardt *et al*., (2005). For this reason annotation transfer by BLAST and PSI-BLAST has been used for the comparison. Our analysis simulated a scenario where only low identity homologues are present, this would be even more difficult to ensure for external methods.

Our analysis also considers the effect of using GO annotations with different levels of confidence (i.e. evidence codes) to assess function prediction methods. We have demonstrated that using more extensive electronic annotations results in improved precision compared to a set of only non-electronic annotations. Using GTPases we have shown that the lack of coverage in non-electronic annotations is a source of the differences observed.

ConFunc performance is limited by the current level of annotated sequences available, as it can only make predictions for GO terms present in three or more of a query sequence's homologues. However, even with limited coverage, ConFunc is able to outperform BLAST annotation transfer, particularly in the twilight zone (Fig. 3). ConFunc is fully automated giving it the potential for use in a genomics annotation pipeline, automatically identifying conserved residues and inferring annotations for newly identified genome sequences.

ACKNOWLEDGEMENTS

The authors would like to thank members of the Sternberg lab for interesting discussion. MNW is funded by a BBSRC grant.

Conflict of Interest: none declared.

REFERENCES

Abhiman,S. and Sonnhammer,E.L. (2005) FunShift: a database of function shift analysis on protein subfamilies. *Nucl. Acids Res*., **33**, D197–D200.

Aloy,P. *et al*. (2001) Automated structure-based prediction of functional sites in proteins: applications to assessing the validity of inheriting protein function from homology in genome annotation and to protein docking. *J. Mol. Biol*., **311**, 395–408.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

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

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

Berezin,C. *et al*. (2004) ConSeq: the identification of functionally and structurally important residues in protein sequences. *Bioinformatics*, **20**, 1322–1324.

Brenner,S.E. (1999) Errors in genome annotation, *Trends Genet*., **15**, 132–133.

Camon,E. *et al*. (2004) The gene ontology annotation (GOA) database—an integrated resource of GO annotations to the UniProt Knowledgebase. *Int. Silico Biol*., **4**, 5–6.

Davis,J. and Goadrich,M. (2006) The relationship between Precision–Recall and ROC Curves. In *23rd International Conference on Machine Learning (ICML)*, Pittsburgh, PA, USA, June 26–28, 2006.

Devos,D. and Valencia,A. (2000) Practical limits of function prediction. *Proteins*, **41**, 98–107.

Devos,D. and Valencia,A. (2001) Intrinsic errors in genome annotation. *Trends Genet*., **17**, 429–431.

Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res*., **32**, 1792–1797.

Engelhardt,B.E. *et al*. (2005) Protein molecular function prediction by bayesian phylogenomics. *PLoS Comput. Biol*., **1**, e45.

Finn,R.D. *et al*. (2006) Pfam: clans, web tools and services. *Nucl. Acids Res*., **34**, D247–D251.

George,R.A. *et al*. (2005) Effective function annotation through catalytic residue conservation. *Proc Natl Acad. Sci. USA*, **102**, 12299–12304.

Groth,D. *et al*. (2004) GOblet: a platform for gene ontology annotation of anonymous sequence data. *Nucl. Acids Res*., **32**, W313–W317.

Hannenhalli,S.S. and Russell,R.B. (2000) Analysis and prediction of functional sub-types from protein sequence alignments. *J. Mol. Biol.*, **303**, 61–76.

Hawkins,T. *et al.* (2006) Enhanced automated function prediction using distantly related sequences and contextual association by PFP. *Protein Sci.*, **15**, 1550–1556.

Hegyi,H. and Gerstein,M. (1999) The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. *J. Mol. Biol.*, **288**, 147–164.

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

Jones,C.E. *et al.* (2005) Automated methods of predicting the function of biological sequences using GO and BLAST. *BMC Bioinformatics*, **6**, 272.

Khan,S. *et al.* (2003) GoFigure: automated Gene Ontology annotation. *Bioinformatics*, **19**, 2484–2485.

Li,W. *et al.* (2002) Tolerating some redundancy significantly speeds up clustering of large protein databases. *Bioinformatics*, **18**, 77–82.

Lichtarge,O. *et al.* (1996) An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol.*, **257**, 342–358.

Lord,P.W. *et al.* (2003) Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, **19**, 1275–1283.

Martin,D.M. *et al.* (2004) GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics*, **5**, 178.

McNemar,Q. (1947) Note on the sampling error of the difference between correlated proportions of percentages. *Psychometrica.*, **12**, 153–157.

Mulder,N.J. *et al.* (2005) InterPro, progress and status in 2005. *Nucl. Acids Res.*, **33**, D201–D205.

Pazos,F. and Sternberg,M.J. (2004) Automated prediction of protein function and detection of functional sites from structure. *Proc. Natl Acad. Sci. USA*, **101**, 14754–14759.

Porter,C.T. *et al.* (2004) The catalytic site atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucl. Acids Res.*, **32**, D129–D133.

Rost,B. (1999) Twilight zone of protein sequence alignments. *Protein Eng.*, **12**, 85–94.

Rost,B. (2002) Enzyme function less conserved than anticipated. *J. Mol. Biol.*, **318**, 595–608.

Schlicker,A. *et al.* (2006) A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, **7**, 302.

Tian,W. and Skolnick,J. (2003) How well is enzyme function conserved as a function of pairwise sequence identity? *J. Mol. Biol.*, **333**, 863–882.

Todd,A.E. *et al.* (2001) Evolution of function in protein superfamilies, from a structural perspective. *J. Mol. Biol.*, **307**, 1113–1143.

Valdar,W.S. (2002) Scoring residue conservation. *Proteins*, **48**, 227–241.

Verspoor,K. *et al.* (2006) A categorization approach to automated ontological function annotation. *Protein Sci.*, **15**, 1544–1549.

Vingron,M. and Argos,P. (1989) A fast and sensitive multiple sequence alignment algorithm. *Comput. Appl. Biosci.*, **5**, 115–121.

Wilson,C.A. *et al.* (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.

Wu,C.H. *et al.* (2006) The universal protein resource (UniProt): an expanding universe of protein information. *Nucl. Acids Res.*, **34**, D187–D191.

Zehetner,G. (2003) OntoBlast function: from sequence similarities directly to potential functional annotations by ontology terms. *Nucl. Acids Res.*, **31**, 3799–3803.