Quantitative assessment of protein function prediction programs

B.N. Rodrigues[1], M.B.R. Steffens[1,2], R.T. Raittz[1], I.C.R. Santos-Weiss[1] and J.N. Marchaukoski[1]

[1]Programa de Pós-Graduação em Bioinformática, Universidade Federal do Paraná, Curitiba, PR, Brasil
[2]Departamento de Bioquímica e Biologia Molecular, Universidade Federal do Paraná, Curitiba, PR, Brasil

Corresponding author: J.N. Marchaukoski
E-mail: jeroniza@ufpr.br

Genet. Mol. Res. 14 (4): 17555-17566 (2015)
Received April 10, 2015
Accepted September 24, 2015
Published December 21, 2015
DOI http://dx.doi.org/10.4238/2015.December.21.28

ABSTRACT. Fast prediction of protein function is essential for high-throughput sequencing analysis. Bioinformatic resources provide cheaper and faster techniques for function prediction and have helped to accelerate the process of protein sequence characterization. In this study, we assessed protein function prediction programs that accept amino acid sequences as input. We analyzed the classification, equality, and similarity between programs, and, additionally, compared program performance. The following programs were selected for our assessment: Blast2GO, InterProScan, PANTHER, Pfam, and ScanProsite. This selection was based on the high number of citations (over 500), fully automatic analysis, and the possibility of returning a single best classification per sequence. We tested these programs using 12 gold standard datasets from four different sources. The gold standard classification of the databases was based on expert analysis, the Protein Data Bank, or the Structure-Function Linkage Database. We found that the miss rate among the programs is globally over 50%. Furthermore, we observed little overlap in the correct predictions from each program. Therefore, a combination of multiple types of sources

and methods, including experimental data, protein-protein interaction, and data mining, may be the best way to generate more reliable predictions and decrease the miss rate.

Key words: Protein function prediction; Comparison; Resources for protein function prediction; Sequence characterization

INTRODUCTION

Many protein sequences are already known and are available in public databases (Godzik et al., 2007; Chitale and Kihara, 2011), but the newly discovered sequences require fast and reliable functional annotation to increase data utility in subsequent searches (Godzik et al., 2007; Gerlt et al., 2012). Experimental characterization is still the most reliable way to define the function associated with a protein sequence, but it is a slow, expensive, and time-consuming process to perform for newly discovered sequences. On the other hand, bioinformatics resources are cheaper and faster techniques for function prediction and they have helped accelerate the process of protein sequence characterization (Godzik et al., 2007; Rentzsch and Orengo, 2009; Blaby-Haas and de Crécy-Lagard, 2011). About 100 computational resources are available for protein function prediction and most of them use sequences as input (Godzik et al., 2007). However, the success of these prediction programs needs to be assessed and some approaches are available, such as the Critical Assessment of Protein Structure Predictions (CASP) and the Critical Assessment of Protein Function Annotation (CAFA). These approaches measure only the hit level using metrics for Gene Ontology (GO) (The Gene Ontology Consortium, 2013) term similarity and do not use multivariate analysis; thus, they evaluate each prediction program individually and with a single dataset at a time (Soro and Tramontano, 2005; Conesa and Götz, 2008; Rentzsch and Orengo, 2009; Chitale and Kihara, 2011; Thomas, 2011; Radivojac et al., 2013).

In the present study, we show a new approach to assess protein function prediction programs. We carried out qualitative and quantitative analyses of these programs to determine which program is best suited to perform protein function prediction. We considered the evaluation of the amount of correctly predicted sequences, run time, differences in results, and characteristics of each program. Furthermore, we developed General Linear Models (GLMs) for total-of-hits analysis, which allowed the simultaneous comparison between different programs and different test datasets. Therefore, we assessed five distinct programs and 12 distinct test datasets.

The five programs assessed were Blast2GO (Conesa and Götz, 2008), InterProScan (Mulder et al., 2007), PANTHER (Thomas, 2011), Pfam (Genome Research Ltd., 2010), and ScanProsite (De Castro et al., 2006). These programs were selected based on a high citation rate (over 500 citations each) in the literature (Thomas et al., 2003b; Conesa et al., 2005; Hunter et al., 2009; Sigrist et al., 2010; Punta et al., 2012), recent last release date (no more than 3 years ago), fully automatic analysis, and the possibility of returning a single best classification per sequence.

The results show that the predictions only from sequence data have a high miss raterate of error. Therefore, the development of new techniques for protein function prediction using only sequence data are required and predictions with more data, other than just sequence, may be more reliable.

MATERIAL AND METHODS

Test datasets

We used 12 test datasets from four different resources (Dobson and Doig, 2003; Pegg et al., 2006; Brown et al., 2006; Brown et al., 2007). Their authors described the sequences as well-characterized and well-known. They are considered as the gold standard according to expert analysis, Protein Data Bank (PDB) (Berman et al., 2000), and Structure-Function Linkage Database (SFLD) nomenclatures. We consider these classifications as standard for comparison and the dataset characteristics and sources are presented in Table 1. The sequences classified by experts were: aminergic G protein coupled receptor (Aminergic GPCR), nuclear hormone receptor (NHR), and secretin-like (Brown et al., 2007). The Enzyme and Non-enzyme datasets came from PDB identifiers (PDB IDs) of reference sequences (Dobson and Doig, 2003). The bifunctional dataset was built from the previous dataset by Dobson and Doig (2003) using the structure title field and chosing the ones with the "bifunctional" term while excluding the "putative", uncharacterized", or "unknown" terms". The enolase, crotonase, haloacid dehalogenase, vicinal oxygen chelate, and radical *S*-adenosyl methionine (Radical SAM) datasets are from superfamilies and came from the SFLD database (Pegg et al., 2006). We only selected sequences with known function and with at least one linked PDB ID. The SFLD was presented and recommended by Schnoes et al. (2009). Brown et al. (2006) described a dataset (we refer to it using the S set-), with many adequate sequences for a test of clustering and functional classification of sequences, providing the GenInfo Identifier (GI) of National Center for Biotechnology Information (NCBI) for each sequence (Brown et al., 2006).

Analysis workflow

We implemented a workflow (Figure 1) in Shell script for program execution and generation of results for analysis. The workflow submitted all datasets to each of the five focus programs, keeping track of the starting and ending times of each execution. A C++ script pre-processed the programs output, which standardized all outputs to a single format: sequence identifier and program sequence classification. An R script standardized the pre-processed output and reference classification, using a methodology adapted from Tsuruoka et al. (2008), and created the results for analysis.

Data analysis

We describe below the four parameters used to assess the protein function prediction programs, including input data information, accounts, and methods used.

Total of hits

We used a total match of function name character strings in this assessment. We determined if the classifications from each program were the same or not as the gold standard classification and the amount of correct predictions determined which program had the most significant number of hits. We developed and used GLMs (Turkman and Silva, 2000) from the "Stats" default package of R script (R Core Team, 2012) to analyze the count data without using non-parametric methods,

and tested many combinations of factors that may affect the predictions, e.g., constant factor, test dataset, prediction program, and factor interaction. The Akaike Information Criterion (AIC) (Christensen, 1997; Turkman and Silva, 2000), also included in R script, was used to select the best model (model with the lowest index value) among the five models built.

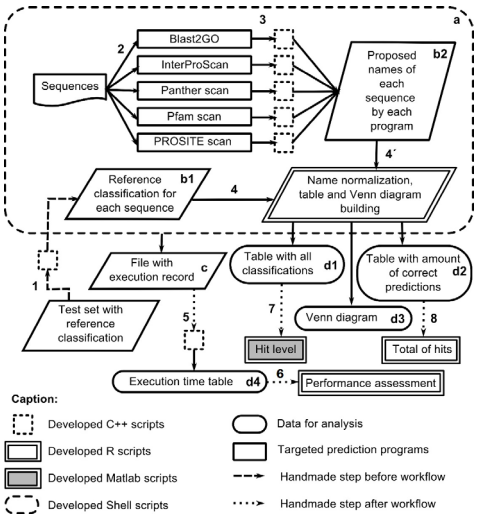

Figure 1. Analysis of workflow. The datasets with reference classification (gold standard) were submitted manually to C++ specific algorithms. These algorithms standardized the data before Shell script execution (1). The Shell script algorithm (a) submitted the sequences to each program (2), standardized the programs output using C++ specific algorithms (3), and submitted the standardized reference classification (made in step 1) (b1) and standardized programs output (b2) to an R script algorithm (4 and 4'). The Shell script execution outputted a file with the execution record of Shell script execution (c), a table with all classifications (d1), a table with the amount of correct predictions (d2), and a Venn diagram (d3). The file with the execution record (c) was submitted manually to a C++ algorithm (5) that generated a table with execution time (d4), which was submitted manually to an R script algorithm for the performance assessment (6). The table with all classifications was manually submitted to a MATLAB algorithm for hit level assessment (7) and the table with the amount of correct predictions to an R script algorithm for total-of-hits analysis (8).

Hit level

We used an equation to compare two classification character strings, which results in a score for comparison between the program classification and reference classification (gold standard). This was done by modifying the Smith-Waterman algorithm score, which is an algorithm developed for sequence comparison, implemented in MATLAB ("swalign" function) (The MathWorks, 2013) to work on the (non-sequence) classification character strings. This modification required alterations in the algorithm input data: conversion of non-amino acid characters to amino acid characters (see section 1 of the **Supplementary material**), use of the identity matrix as the substitution matrix, and Fasta file construction within classification strings in the sequence field using the specific C++ algorithm. The equation includes the possible average values for two-sequence comparison and divides this average by the summed average of self-comparison sequence results. The equation result is a number between 0 and 1, with numbers close to 1 indicating that the sequences are the same (Equation 1). This score enables a comparison analysis for all classifications.

$$y = \frac{\left[\frac{\left(\frac{x_{12}}{x_{11}} + \frac{x_{12}}{x_{22}}\right)}{2} + \frac{\left(\frac{x_{21}}{x_{11}} + \frac{x_{21}}{x_{22}}\right)}{2}\right]}{2} \quad \text{(Equation 1)}$$

where y is a similarity score between one program classification and one reference classification and x is the Smith-Waterman algorithm similarity, wherein x_{11} is a reference classification self-score, x_{22} is a program classification self-score, x_{12} is a reference classification versus program classification score, and x_{21} is a program classification versus reference classification score.

Performance assessment

We calculated analysis of variance (ANOVA) with the Tukey method (Callegari-Jacques, 2003) to determine the best performance program from the execution time data. We used the total execution time per sequence and per amino acid residue. The data processing time aims to highlight the fastest program and is calculated from execution times (Fonseca et al., 2012).

Program characteristics

We listed the characteristics that would lead a user to choose a program, such as file size, program execution pre-requirements, installation, and user-friendliness (Fonseca et al., 2012).

Other informative analysis are present in other prediction program assessments (Pandey et al., 2006; Rentzsch and Orengo, 2009; Henry et al., 2011), such as the accuracy that arises from program hit data. This study includes the level hit data, besides the hit or miss data.

RESULTS

We analyzed 12 test datasets and the assessed prediction programs did not correctly predict any sequence from four test datasets: AminergicGPCR, NHR, Secretin-like, and Crotonase (Table 1). Therefore, we excluded those datasets from the total-of-hits analysis. The programs exhibited significant divergences of prediction for the other eight datasets. Blast2GO showed the largest number of hits and this represents less than 30% of the sequences dataset (see section 2 of **Supplementary material**). According to the best fitted GLM of total-of-hits analysis ($P < 0.05$) (Equation 2, Figure 2) and the level hit analysis ($P < 0.5$; Figure 3), Blast2GO exhibited higher hit probability than any other assessed program, and InterProScan was the second program with the highest hit probability. Pfam and ScanProsite showed the lowest hit probabilities (Figure 2).

$$\begin{aligned} y_i &= \beta_{program1} Program1_i + \ldots + \beta_{program5} Program5_i \\ &+ \beta_{Set1} Set1_i + \ldots + \beta_{Set12} Set12_i \\ &+ \beta_{Interaction1} Interaction1_i + \ldots + \beta_{Interaction60} Interaction60_i \end{aligned} \quad \text{(Equation 2)}$$

where y is the answer variable. *Program*, *Set*, and *Interaction* are predictive variable covariates; β is the estimator; and i is a positive integer representing the number of the experimental unit.

The highest similarity level possible is 1.0 and this value shows that the predicted nomenclature string completely matched the gold standard nomenclature string. The values between the highest and lowest level hits represent a ratio of the predicted nomenclature string that is equal to the gold standard nomenclature string (x-axis in Figure 3).

Table 1. Dataset description.

Test dataset	Number of sequences	Number of classifications	Number of bases
AminergicGPCR [1]	358	31	104335
NHR [1]	412	27	72227
Secretin-like [1]	153	15	38025
Enzymes [2]	690	630	226335
Non-enzymes [2]	487	449	88397
Bifunctional (adapted from [2])	60	51	22074
Enolase [3]	927	25	357425
Crotonase [3]	262	18	86773
Haloacid dehalogenase [3]	389	22	263851
Vicinal oxygen chelate [3]	145	12	39592
SAM [3]	145	19	52027
S [4]	863	90	353221

NHR: Nuclear Hormone Receptor; SAM: Radical S-adenosyl methionine; S: Brown SD's dataset (Brown et al., 2006). Data source: [1] (Brown et al., 2007) , [2] (Dobson and Doig, 2003), [3] (Pegg et al., 2006), [4] (Brown et al., 2006).

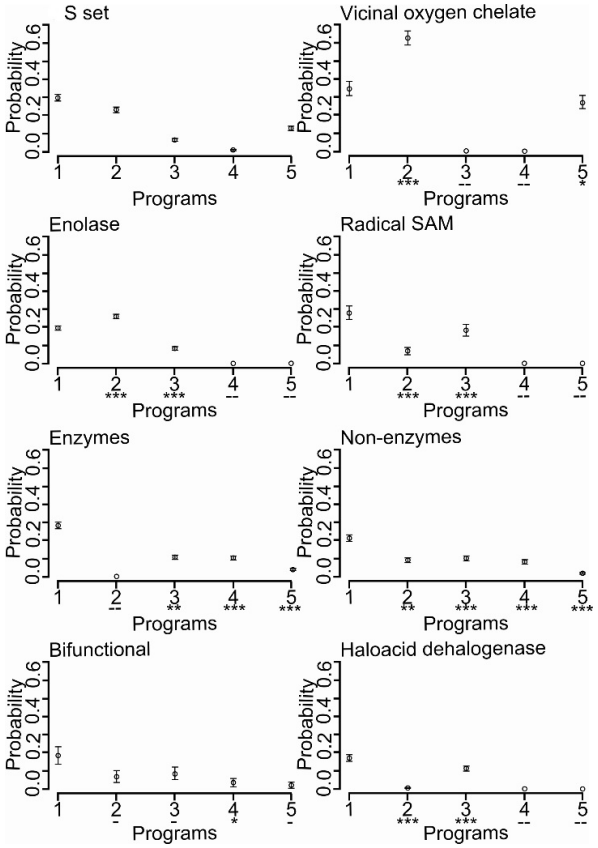

Figure 2. Total-of-hits analysis. Hit probability (y-axis) for each program (x-axis) (1) Blast2GO, (2) InterProScan, (3) PANTHER, (4) Pfam, and (5) ScanProsite, according to the best-fitted GLM for the test datasets: enzymes, non-enzymes, bifunctional, enolase, haloacid dehalogenase, vicinal oxygen chelate, radical SAM, and S sets. P values lower than 0.05 are designated as (*), lower than 0.01 as (**), and lower than 0.001 as (***). High-standard deviations (over 300) related to low-hit counts are signed with (-) and related to null-hit counts are designated as (--).

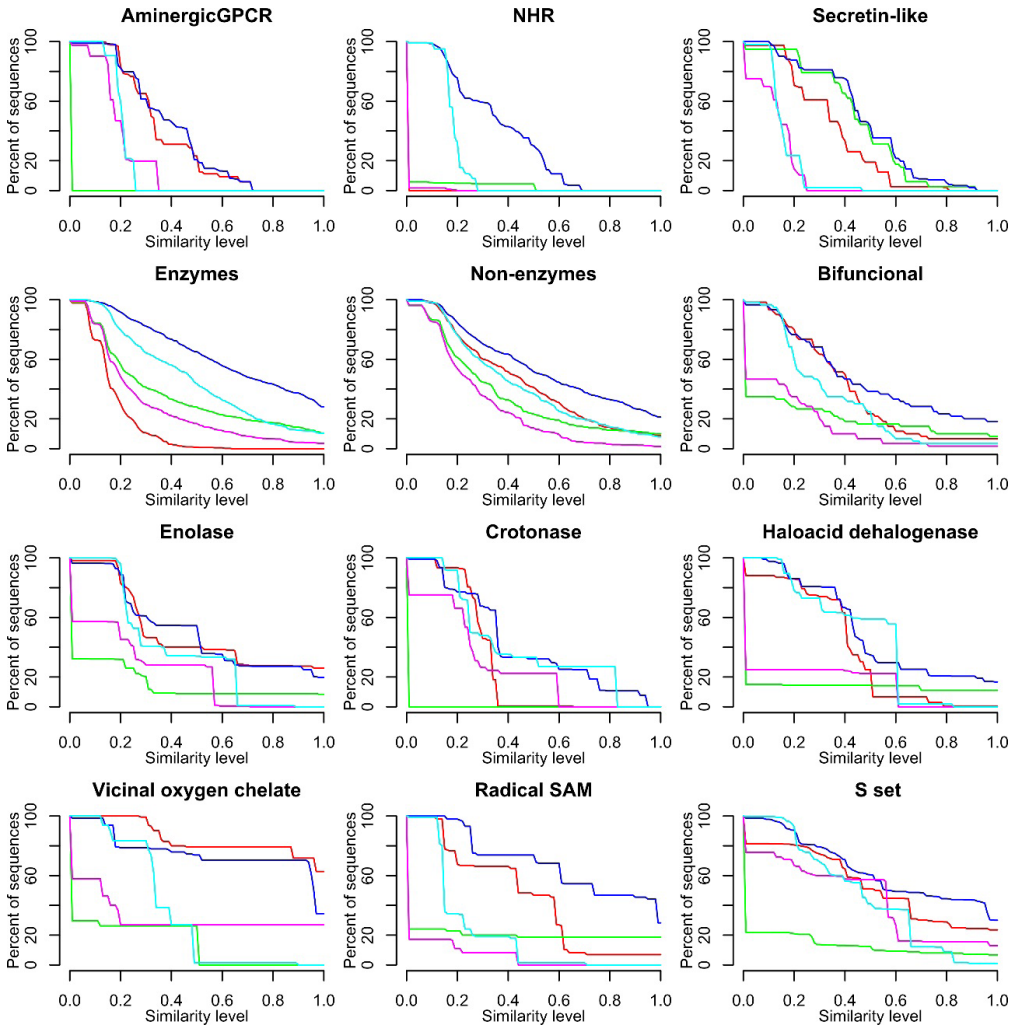

Figure 3. Hit level. Sequence percent (y-axis) for a specific hit level (x-axis) between Blast2GO (-), InterProScan (-), PANTHER (-), Pfam (-), and ScanProsite (-) predictions and the reference nomenclature. Note that for the AminergicGPCR, NHR, Secretin-like, and Crotonase sets, none of the programs reached the highest similarity level.

No sequence was predicted by all the programs simultaneously (central value in Figure 4B); 1433 sequences were correctly predicted from a total of 4891 sequences (or 3706 sequences, excluding the 4 sets without any hit). Blast2GO and InterProScan had the highest number of hits, either by themselves or combined (Figure 4B). Blast2GO and InterProScan correctly predicted 905 and 593 sequences, respectively, representing 19 and 12% (24 and 16% without those 4 sets) from the total number of sequences. Both programs correctly predicted 1233 sequences, representing 25% (33% without those 4 sets) from the total number of sequences. The correctly predicted sequences of Blast2GO contained around 80% of the correctly predicted sequences of PANTHER and more than 45% of the correctly predicted sequences of Pfam (Figure 4B). The

programs that shared the lowest number of correctly predicted sequences with other programs were ScanProsite, followed by InterProScan (Figure 4B). There were 10 sequences that none of the assessed programs predicted (central value in Figure 4A), with one coming from the NHR dataset, three from the Enzymes dataset, and six from the Non-enzymes dataset (see section 3 of **Supplementary material**). PANTHER and ScanProsite had the highest number of sequences without prediction: 3615 for PANTHER and 2459 for ScanProsite (Figure 4A). Both programs also simultaneously shared the highest number of sequences without prediction, that is, 1758 (Figure 4A). Pfam had the lowest number of sequences without prediction, with all its non-classified sequences being shared with all other programs (Figure 4A).

The execution times of the programs were significantly divergent ($P < 0.01$). PANTHER, Pfam, and ScanProsite had execution times lower than Blast2GO and InterProScan (Figure 5).

InterProScan, PANTHER, Pfam, and ScanProsite exhibited the easiest and fastest installations, and were more intuitive than Blast2GO in terms of command line use. The local Blast2GO installation requires database knowledge for the GO database installation. Blast2GO also spend the largest hard disc space due to the installation of the GO database. However, Blast2GO was the only program with both local user-friendly interface and local command line interfaces.

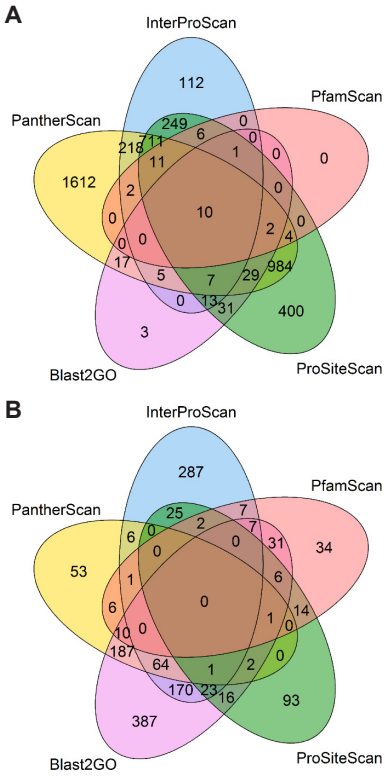

Figure 4. Venn diagram. **A.** Diagram shows the number of non-classified sequences. Note that PANTHER had the lowest number of sequences without prediction and that 10 sequences were not classified by any of the assessed programs. **B.** Diagram shows the number of sequences correctly predicted. Note that the programs that shared most of the sequences correctly predicted were Blast2GO and InterProScan, and that those shared sequences are less than half of the total of hits for each program.

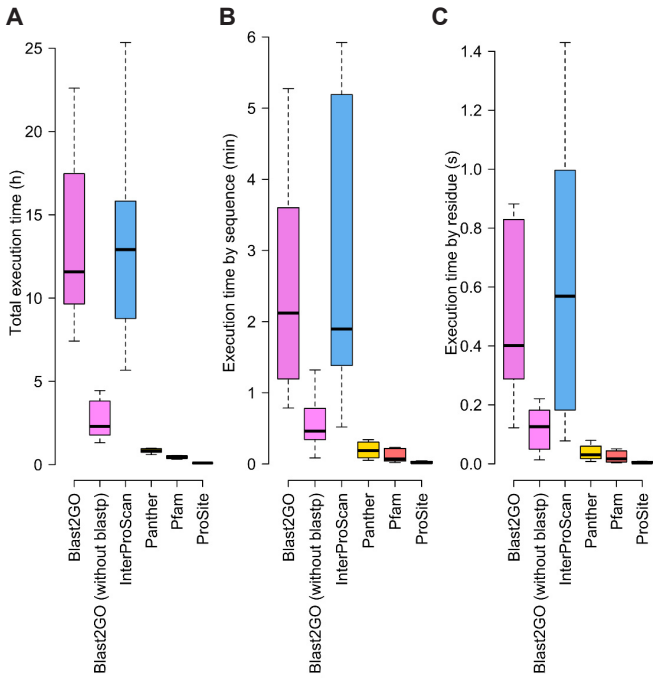

Figure 5. Execution time by program. Boxplot representations show the total execution time (**A**), the execution time by sequence (**B**), and the execution time by amino acid residue (**C**), for each program.

DISCUSSION

In this study, we performed a wider assessment of protein function prediction programs than that performed in previous studies. We analyzed the total hits, hit level, performance assessment, and software characteristics. The total hits and hit level were assessed simultaneously in all programs with all test datasets.

Other studies compared the protein function prediction programs by measuring GO term distance or contingency table data (Pandey et al., 2006; Rentzsch and Orego, 2009; Blaby-Haas and de Crécy-Lagard, 2011; Henry et al., 2011). The proposal of GO is to characterize the gene product by ontologies rather than by function nomenclature (molecular name) as it is done in this work (Chitale and Kihara, 2011; The Gene Ontology Consortium, 2013). The function nomenclature is the role of the protein in the organism in which it is expressed (Tripton and Boyce, 2000) and it is present in the “definition” field of the GenBank format file (file format for gene sequences) or GenPept (format file for protein sequences) in NCBI (U.S. National Library of Medicine, 2014). The contingency table methodology used in other assessments needs to reduce the classifications into two classification categories and only allows the assessment of one program and one dataset at each time (Thomas et al., 2003a; Prati et al., 2008).

In this work, we applied a methodology that used GLMs for total-of-hits analysis and Equation 1 for level hit analysis, which enable a simultaneous assessment of all programs, test datasets, and functional classifications. This analysis suggested that the hit capacity depends on the test dataset, since the best-fitted GLM (including the program/dataset interaction effect;

Equation 2 and Figure 2) and level hit plots (Figure 3) showed that some datasets had more similarities between the programs and reference classifications. The dependence relationship between program and dataset can explain the observed differences in programs accuracy in other assessments. Blast2GO had 70% accuracy when annotating proteins from *Arabidopsis* sp (Conesa and Götz, 2008) and 47.7% F-score (Radivojac et al., 2013), using GO term distance comparison in both cases, but with different sequence test datasets.

The literature describes an accuracy of no more than 80% for protein function prediction programs that use only sequence data (Conesa and Götz, 2008; Radivojac et al., 2013). We showed that only in one dataset, the vicinal oxygen chelate dataset, the hit rate was over 50% (~62%), and only for the InterProScan prediction. In any other prediction situations, all programs exhibited a hit rate or probability under 35%, which included Blast2GO, the program with the highest hit probability.

Although programs can predict protein function faster, cheaper, and easier than experimental data (such as RNA-seq or microarrays) (Friedberg, 2006; Chitale and Kihara, 2011; Clark and Radivojac, 2011), a prediction based only on sequence data can generate wrong deposits (misannotated sequences), thereby affecting the quality of sequence annotated data in public databases. The Non-redundant database of NCBI (NR) showed 40% of misannotated sequences in 2005, with 85% of them being annotation mistakes without specific evidence to support them (Brown et al., 2007). The use of sequence similarity data with other methodologies and data could improve the accuracy of these annotations (Blaby-Haas and de Crécy-Lagard, 2011).

InterProScan presented different characteristics when compared to other assessed programs. It has an easy and fast installation, it does not use the NR database as reference, and it was the second program with more hits in protein function predictions. Additionally, InterProScan combines information from other databases (Pfam, Prosite and PANTHER) with manually curated data (Hunter et al., 2009), which results in low overlap with the output from those programs (Figure 4).

The methodologies for function prediction from experimental data are still not possible on the high-throughput scale, due to their high cost and slow speed (Godzik et al., 2007; Blaby-Haas and de Crécy-Lagard, 2011; Gerlt et al., 2012). Consequently, there is a gap for high-throughput protein characterization. Thus, there is a need to develop new methodologies to close this gap, such as the development of sequence-based function prediction programs that give a reliable function prediction.

Conflicts of interest

The authors declare no conflict of interest.

ACKNOWLEDGMENTS

Research supported by CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior; CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico, and INCT - Institutos Nacionais de Ciência e Tecnologia da Fixação Biológica de Nitrogênio.

Supplementary material

REFERENCES

Berman HM, Westbrook J, Feng Z, Gilliland G, et al. (2000). The Protein Data Bank. *Nucleic Acids Res.* 28: 235-242.

Blaby-Haas CE and de Crécy-Lagard V (2011). Mining high-throughput experimental data to link gene and function. *Trends Biotechnol*. 29: 174-182.

Brown DP, Krishnamurthy N and Sjölander K (2007). Automated protein subfamily identification and classification. *PLoS Comput. Biol.* 3: e160.

Brown SD, Gerlt JA, Seffernick JL and Babbitt PC (2006). A gold standard set of mechanistically diverse enzyme superfamilies. *Genome Biol.* 7: R8.

Callegari-Jacques SM (2003). Bioestatística: princípios e aplicações. Artmed, Porto Alegre.

Chitale M and Kihara D (2011). Computational protein function prediction: framework and challenges. In: Protein function prediction for omics era (Kihara D, ed.). Springer Netherlands, Dordrecht, 1-17.

Christensen R (1997). Log-linear models and logistic regression. 2nd edn. Springer New York, New York.

Clark WT and Radivojac P (2011). Analysis of protein function and its prediction from amino acid sequence. *Proteins* 79: 2086-2096.

Conesa A and Götz S (2008). Blast2GO: A comprehensive suite for functional analysis in plant genomics. *Int. J. Plant Genomics* 2008: 619832.

Conesa A, Götz S, García-Gómez JM, Terol J, et al. (2005). Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* 21: 3674-3676.

De Castro E, Sigrist CJA, Gattiker A, Bulliard V, et al. (2006). ScanProsite: detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins. *Nucleic Acids Res.* 34: W362-W365.

Dobson PD and Doig AJ (2003). Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.* 330: 771-783.

Fonseca NA, Rung J, Brazma A and Marioni JC (2012). Tools for mapping high-throughput sequencing data. *Bioinformatics* 28: 3169-3177.

Friedberg I (2006). Automated protein function prediction - the genomic challenge. *Briefings Bioinf.* 7: 225-242.

Radivojac P, Clark WT, Oron TR, Schnoes AM, et al. (2013). A large-scale evaluation of computational protein function prediction. *Nat. Methods* 10: 221-227.

Genome Research Ltd. (2010). Pfam version 1.3. Available at [ftp://ftp.sanger.ac.uk/pub/databases/Pfam/Tools]. Accessed October 18, 2011.

Gerlt JA, Babbitt PC, Jacobson MP and Almo SC (2012). Divergent evolution in enolase superfamily: strategies for assigning functions. *J. Biol. Chem.* 287: 29-34.

Godzik A, Jambon M and Friedberg I (2007). Computational protein function prediction: are we making progress? *Cell. Mol. Life Sci.* 64: 2505-2511.

Henry CS, Overbeek R, Xia F, Best AA, et al. (2011). Connecting genotype to phenotype in the era of high-throughput sequencing. *Biochim. Biophys. Acta* 1810: 967-977.

Hunter S, Apweiler R, Attwood TK, Bairoch A, et al. (2009). InterPro: the integrative protein signature database. *Nucleic Acids Res.* 37: D211-D215.

Mulder NJ, Apweiler R, Attwood TK, Bairoch A, et al. (2007). New developments in the InterPro database. *Nucleic Acids Res.* 35: D224-D228.

Pandey G, Kumar V and Steinbach M (2006). Computational approaches for protein function prediction: a survey. Technical Report 06-028, Department of Computer Science and Engineering, University of Minnesota, Twin Cities.

Pegg SC, Brown SD, Ojha S, Seffernick J, et al. (2006). Leveraging enzyme structure-function relationships for functional inference and experimental design: the structure-function linkage database. *Biochemistry* 45: 2545-2555

Prati RC, Batista GEAPA and Monard MC (2008). Curvas ROC para avaliação de classificadores. *Rev. IEEE América Latina* 6: 215-222.

Punta M, Coggill PC, Eberhardt RY, Mistry J, et al. (2012). The Pfam protein families database. *Nucleic Acids Res.* 40: D290-D301.

R Core Team (2012). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. Available at [http://www.R-project.org].

Rentzsch R and Orengo CA (2009). Protein function prediction-the power of multiplicity. *Trends Biotechnol*. 27: 210-219.

Schnoes AM, Brown SD, Dodevski I and Babbitt PC (2009). Annotation error in public databases: misannotation of molecular function in enzyme superfamilies. *PLoS Comput. Biol*. 5: e1000605.

Sigrist CJ, Cerutti L, de Castro E, Langendijk-Genevaux PS, et al. (2010). PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Res.* 38: D161-D166.

Soro S, Tramontano A (2005). The prediction of protein function at CASP6. *Proteins* 61 (Suppl 7): 201-213.

The Gene Ontology Consortium (2013). An Introduction to the Gene Ontology. The Scope of GO. Available at [http://www.

geneontology.org/GO]. Accessed February 26, 2014.

The MathWorks (2013). MATLAB, Version 8.1. The MathWorks Inc., Natick, Massachusetts.

Thomas PD (2011). PANTHER HMM scoring tools, Version 1.03. Available at [http://pantherdb.org/downloads]. Accessed October 4, 2011.

Thomas PD, Kejariwal A, Campbell MJ, Mi H, et al. (2003a). PANTHER: a browsable database of gene products organized by biological function, using curated protein family and subfamily classification. *Nucleic Acids Res.* 31: 334-341.

Thomas PD, Campbell MJ, Kejariwal A, Mi H, et al. (2003b). PANTHER: a library of protein families and subfamilies indexed by function. *Genome Res*. 13: 2129-2141.

Tripton K and Boyce S (2000). History of the enzyme nomenclature system. *Bioinformatics* 16: 34-40.

Tsuruoka Y, McNaught J and Ananiadou S (2008). Normalizing biomedical terms by minimizing ambiguity and variability. *BMC Bioinf.* 9 (Suppl 3): S2.

Turkman MAA and Silva GL (2000). Modelos Lineares Generalizados - da teoria à prática. Sociedade Portuguesa de Estatística, Lisboa.

U.S. National Library of Medicine (2014). National Center for Biotechnology Information - NCBI. Available at [http://www.ncbi.nlm.nih.gov/]. Accessed March 4, 2014.