Check for updates

METHOD ARTICLE

Large-scale protein function prediction using heterogeneous ensembles [version 1; peer review: 2 approved]

Linhua Wang [1], Jeffrey Law [2], Shiv D. Kale[3], T. M. Murali[4], Gaurav Pandey[1]

[1]Department of Genetics and Genomic Sciences and Icahn Institute for Genomics and Multiscale Biology, Icahn School of Medicine at Mount Sinai, New York, NY, 10029, USA
[2]Genetics, Bioinformatics, and Computational Biology Ph.D. Program, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061, USA
[3]Biocomplexity Institute, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061, USA
[4]Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24061, USA

v1 **First published:** 28 Sep 2018, **7**(ISCB Comm J):1577
https://doi.org/10.12688/f1000research.16415.1

Latest published: 28 Sep 2018, **7**(ISCB Comm J):1577
https://doi.org/10.12688/f1000research.16415.1

Abstract

Heterogeneous ensembles are an effective approach in scenarios where the ideal data type and/or individual predictor are unclear for a given problem. These ensembles have shown promise for protein function prediction (PFP), but their ability to improve PFP at a large scale is unclear. The overall goal of this study is to critically assess this ability of a variety of heterogeneous ensemble methods across a multitude of functional terms, proteins and organisms. Our results show that these methods, especially Stacking using Logistic Regression, indeed produce more accurate predictions for a variety of Gene Ontology terms differing in size and specificity. To enable the application of these methods to other related problems, we have publicly shared the HPC-enabled code underlying this work as LargeGOPred (https://github.com/GauravPandeyLab/LargeGOPred).

Keywords

protein function predictionheterogeneous ensemblesmachine learning high-performance computing performance evaluation

This article is included in the International Society for Computational Biology Community Journal gateway.

This article is included in the Python collection.

Open Peer Review

Reviewer Status ✓ ✓

	Invited Reviewers	
	1	**2**
version 1 28 Sep 2018	✓ report	✓ report

1 **Guoxian Yu** , Southwest University, Chongqing, China

2 **Predrag Radivojac**, Northeastern University, Boston, USA

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Gaurav Pandey (gaurav.pandey@mssm.edu)

Author roles: Wang L: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Law J**: Data Curation, Investigation, Resources, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Kale SD**: Data Curation, Funding Acquisition, Investigation, Resources, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Murali TM**: Data Curation, Funding Acquisition, Investigation, Methodology, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Pandey G**: Conceptualization, Formal Analysis, Funding Acquisition, Investigation, Methodology, Project Administration, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was supported in part by National Institutes of Health [R01GM114434] and by an IBM faculty award to GP. It was also partially supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the Army Research Office (ARO) under Cooperative Agreement Number [W911NF-17-2-0105]. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NIH, ODNI, IARPA, ARO, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

How to cite this article: Wang L, Law J, Kale SD *et al.* **Large-scale protein function prediction using heterogeneous ensembles [version 1; peer review: 2 approved]** F1000Research 2018, **7**(ISCB Comm J):1577 https://doi.org/10.12688/f1000research.16415.1

First published: 28 Sep 2018, **7**(ISCB Comm J):1577 https://doi.org/10.12688/f1000research.16415.1

Introduction

Given the large and rapidly growing gap between sequenced genomes and experimentally determined functional annotations of the constituent proteins, the automation of protein function prediction (PFP) using computational tools is critical[1,2]. However, diverse data sources, data quality issues, like noise and incompleteness, and a lack of consensus on the best predictor(s) for various types of data and functions pose serious challenges for PFP. Specifically, data types used by existing PFP methods have included amino acid sequences, protein structure information, gene expression profiles and protein-protein interaction networks. Similarly, prediction methodologies have ranged from homology-based sequence alignment to machine learning algorithms, network-based methods, and others. Several community-based critical assessments, especially CAFA[3,4], have been organized to objectively measure the performance of these diverse PFP methods. A central finding from these assessments was the variable performance of the tested methods/predictors for different functional terms from the Gene Ontology (GO)[5,6] and target proteins, demonstrating that there is no ideal predictor of all types of protein function.

A potential approach for improving prediction performance in such a scenario of diverse data types and individual/base predictors is to build *heterogeneous ensembles*[7]. These ensembles harness the consensus and diversity among the base predictors, and can help reduce potential overfitting and inaccuracies incurred by them. Unsupervised methods like majority vote and mean aggregation, and supervised approaches like stacking and ensemble selection are the most commonly used methods for building heterogeneous ensembles. Stacking builds such an ensemble by learning a function, also known as a meta-predictor, that optimally aggregates the outputs of the base predictors[8]. Ensemble selection methods iteratively add one or more base predictors to the current ensemble either greedily or to improve the overall diversity and performance of the ensemble[9–11]. These approaches have been successfully applied to a variety of prediction problems[12–15].

In previous work[7], we tested the efficacy of heterogeneous ensembles for annotating approximately 4,000 *Saccharomyces cerevisiae* proteins with GO terms. For this, we evaluated stacking using logistic regression as the meta-predictor and Caruana *et al.*'s ensemble selection (CES) algorithm[9,10], both implemented in our open-source package DataSink. The implementation uses a nested cross-validation setup[7] to train the base predictors and the ensembles independently with the aim of reducing overfitting[16] and improving prediction performance. These experiments yielded that both CES and stacking performed significantly better than stochastic gradient boosting[17], the best-performing base predictor for all the GO terms considered. This improvement was observed both in terms of the AUC score, as well as the $F_{\max}$ measure, which has been established to be more relevant for PFP evaluation[3,4].

A major limitation of this previous study was the relatively high computational cost of constructing heterogeneous ensembles, despite their high-performance computing (HPC)-enabled implementations in DataSink. Due to this cost, we were able to test the ensembles' performance on only three GO terms for proteins of only one organism (*S. cerevisiae*). Owing to the same limitation, only logistic regression was tested as the meta-predictor for stacking. Thus, despite the initial encouraging results, it remains unclear if heterogeneous ensembles provide the same improvement over individual base predictors for a substantial part of GO as well as for a large number of proteins from multiple organisms.

The overall goal of this study is to critically assess this ability of heterogeneous ensembles to improve PFP at a large scale across a multitude of functional terms, proteins and organisms. For this, we adopt an HPC-enabled strategy to evaluate heterogeneous ensembles, built using CES and stacking with eight meta-prediction algorithms, for large-scale PFP. This evaluation is conducted over 277 GO terms, and more than 60,000 proteins, from 19 pathogenic bacterial species. Specifically, we analyze the following aspects of of heterogeneous ensembles:

1. Prediction performance compared to that of the best-performing individual predictor for each GO term.
2. How this performance varies for different GO terms categorized by:
   (a) Number of genes annotated to each term (size).
   (b) Different depths in the GO hierarchy (levels of specificity).

We expect the results of this study to shed light on the efficacy of heterogeneous ensembles for large-scale protein function prediction. To enable the application of these ensembles to other related problems, we have publicly shared the HPC-enabled code underlying this work as LargeGOPred.

Methods

Data used in the study

We extracted the amino acid sequences of 63,449 proteins from 19 clinically relevant bacterial pathogens, which include a subset of organisms from the Health and Human Services (HHS) list of select agents and those with current high clinical relevance[18,19]. The annotations of these proteins to GO terms used in this study were either inferred by a curator (evidence codes: ISS, ISO, ISA, ISM, IGC, IBA, IBD, IKR, IRD, RCA, TAS, NAS and IC) or from experiments (evidence codes: EXP, IDA, IPI, IMP, IGI and IEP), but not from electronic annotations (IEA) in the UniProt database[20]. We selected 277 molecular function (MF) and biological process (BP) GO terms with more than 200 annotated proteins across all the 19 bacteria. The constantly changing contents of the GO ontology and annotations, as well as our incomplete knowledge of the latter make it possible for sequences not annotated to a GO term to be annotated in the future. Thus, to prepare more well-defined datasets, for each GO term, we defined proteins annotated to it as positive samples and any proteins that are neither annotated to the GO term nor its ancestors or descendants as negative samples[21]. The resultant distributions of GO terms with regard to the number of proteins positively annotated to them for each organism and across all organisms are shown in Table 1.

Table 1. Overview of the data used in this study. The ‘#Proteins’ column shows the number of proteins in the corresponding bacterial pathogen listed in the ‘Organism’ column. The disease(s) each of these pathogens has been implicated in are listed in the ‘Disease(s)’ column. The ‘Distribution of GO terms’ column with 3 sub-columns shows the number of proteins annotated with GO terms with that range of #annotations, with the corresponding number of GO terms shown in parenthesis. The final row of the table shows the total number of proteins and GO terms considered in this study. Ranges of distributions of GO terms for all species are shown in the parenthesis of the three ‘#annotations’ sub-columns. Since each GO term is considered independently, each protein may be counted as annotated to multiple GO terms.

Organism	Disease(s)	#Proteins	Distribution of GO terms (#annotations)		
			0-10 (200-500)	10-100 (500-1000)	>100 (>1000)
Yersinia pestis	plague, black death	7375	164 (26)	7397 (218)	6773 (33)
Mycobacterium tuberculosis	tuberculosis (TB)	6112	53 (12)	8850 (186)	19095 (79)
Burkholderia vietnamiensis	severe respiratory disease	4889	49 (277)	0	0
Pseudomonas aeruginosa	nosocomial infection	4488	44 (6)	8515 (171)	23891 (100)
Klebsiella pneumoniae	nosocomial infection, pneumonia	4140	66 (277)	0	0
Escherichia coli	severe abdominal cramps	4067	1 (1)	6811 (104)	53731 (172)
Vibrio cholerae	cholera	3756	100 (13)	8218 (164)	27961 (100)
Salmonella typhimurium	gastroenteritis	3713	64 (11)	8861 (224)	9532 (42)
Shigella dysenteriae	shigellosis	3039	69 (277)	0	0
Peptoclostridium difficile	pseudomembranous colitis	2925	168 (277)	0	0
Bordetella pertussis	pertussis or whooping cough	2688	123 (277)	0	0
Clostridium botulinum	botulism poisoning	2678	277 (64)	5609 (191)	4076 (22)
Enterococcus faecium	neonatal meningitis or endocarditis	2343	0 (277)	0	0
Staphylococcus aureus	severe skin infections	2142	415 (72)	5628 (184)	3863 (21)
Acinetobacter baumannii	nosocomial infection	1946	0 (277)	0	0
Haemophilus influenzae	bacteremia, pneumonia	1500	526 (79)	5233 (178)	3947 (20)
Neisseria gonorrhoeae	sexually transmitted disease	1464	141 (270)	175 (7)	0
Streptococcus pyogenes	pharyngitis, impetigo	1332	154 (277)	0	0
Helicobacter pylori	peptic ulcers, gastritis, stomach cancer	1145	374 (272)	217 (5)	0
Total		**63449**	**47226 (152)**	**51720 (71)**	**122225 (54)**

We chose normalized k-mer frequencies, extracted using the khmer package (2.1.1)[22], as our feature set to represent the information contained in the amino acid sequences and construct a feature matrix that can serve as input for LargeGOPred. K-mers have been used for similar purposes in several PFP studies[1], as well as related problems like the prediction of protein secondary structure[23] and RNA-protein interactions[24]. Since the size of the feature set (all possible k-mers) grows rapidly with increasing value of k, setting k to a high value may be impractical for large-scale PFP tasks like ours. Additionally, 1- and 2-mers may not provide enough context information about the sequence. Thus, we set k = 3 since this value strikes a balance between the information captured by the k-mers and computational scalability. For each amino acid sequence, we extracted frequencies for all possible 8,000 3-mers at each position of the sequence. We then normalized these frequencies by the length of the sequence to reduce the potential bias due to the variation of sequence lengths among the proteins.

All the processed data are available from https://zenodo.org/record/1434450#.W6lU2hNKhBx (doi: 10.5281/zenodo.1434450)[25].

Overview of the prediction approach

The overall approach adopted for this study is visualized and described in detail in Figure 1. Two key components of the approach, specifically the heterogeneous ensemble methods used and nested cross-validation, are described in the following subsections, as well in our previous work[7]. The prediction performance of all the predictors tested in this study, specifically the base classifiers and ensembles, was evaluated in terms of the F_{max} measure, which is the maximum value of F-measure[26] across all binarization thresholds, and has been recommended as a

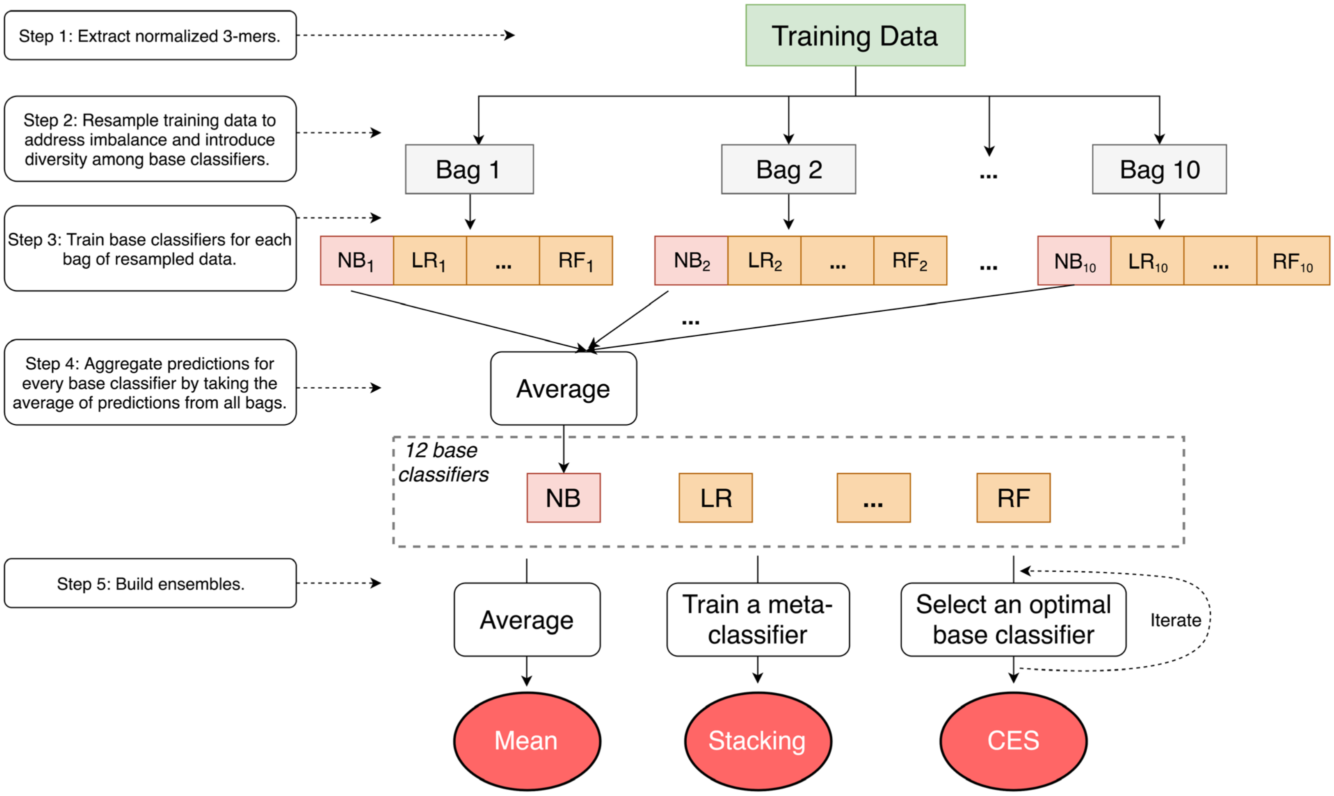
Step 1: Extract normalized 3-mers.
Training Data
Step 2: Resample training data to address imbalance and introduce diversity among base classifiers.
Bag 1
Bag 2
...
Bag 10
Step 3: Train base classifiers for each bag of resampled data.
NB1
LR1
...
RF1
NB2
LR2
...
RF2
...
NB10
LR10
...
RF10
...
Step 4: Aggregate predictions for every base classifier by taking the average of predictions from all bags.
Average
12 base classifiers
NB
LR
...
RF
Step 5: Build ensembles.
Average
Train a meta-classifier
Select an optimal base classifier
Iterate
Mean
Stacking
CES

Figure 1. Overview of the prediction approach. We first extracted normalized 3-mer frequencies from the amino acid sequences as features. Training data for 12 types of base classifiers (upper half of Table 2) were randomly under-sampled into 10 bags containing equal numbers of positive and negative samples to address class imbalance and to introduce diversity among base classifiers, even among those of the same type. The predictions from these bags were averaged for each base classifier and collected to train the heterogeneous ensembles using three types of methods, namely mean aggregation, 8 stacking meta-classifiers (bottom half of Table 2), and Caruana *et al.*'s ensemble selection (CES). Separate test data were used to evaluate the heterogeneous ensembles. The entire process was conducted within a nested cross-validation setup (described below) executed for each target GO term separately.

PFP evaluation measure by CAFA[3,4]. We also evaluated the statistical significance of the difference between the performance of the various predictors (described below)[27]. Finally, since we approach GO term prediction as a binary classification problem, the terms "predictor" and "classifier", and their variants will be used interchangeably as appropriate in the rest of the paper.

Heterogeneous ensemble methods

We used 12 diverse base predictors from the Weka machine learning suite (3.7.10)[28] (upper half of Table 2) and built 3 types of unsupervised and supervised heterogeneous ensembles on top of them. The unsupervised mean method simply takes the average of the predictions from base classifiers as the final prediction. For supervised heterogeneous ensembles, we tested various stacking methods and one of the most widely used ensemble selection methods, namely CES.

Stacking. Stacking builds a heterogeneous ensemble by learning a meta-classifier that optimally aggregates the outputs of the base predictors. Unlike our previous study, where only stacking using logistic regression as the meta-classifier was tested, we used 8 different meta-classifiers in this study (bottom half of Table 2), and statistically compared their performance over all the target prediction problems.

Ensemble selection and CES. Ensemble selection is a process to selecting a subset of all the base classifiers that are mutually complementary such that the resultant ensemble is as predictive as possible.

In this study, we tested Caruana *et al*'s ensemble selection (CES) algorithm for large-scale PFP[9,10]. CES is an iterative algorithm that starts with an empty ensemble, and in each iteration, adds the base predictor that best improves the resultant ensemble's performance, partly due to the added predictor's complementarity to the current ensemble. The process continues until the ensemble's performance doesn't improve anymore, or even starts decreasing. In this work, we tested the version of CES in which the base predictor to be added to the ensemble was sampled with replacement in each iteration[9].

Nested cross-validation

Cross validation (CV) is a frequently used methodology for training and testing classifiers and other predictors[29].

Table 2. Base classifiers used to construct all the heterogeneous ensemble methods tested in this study (upper half), and meta-classifiers used to construct stacking-based ensembles (lower half). The base and meta-classifiers were adopted from Weka[28] and scikit-learn[30] respectively.

Base classifiers	
Classifier name	**Weka class name**
Naive Bayes (NB)	*weka.predictors.bayes.NaiveBayes*
Logistic Regression (LR)	*weka.predictors.functions.Logistic*
Stochastic Gradient Descent (SGD)	*weka.predictors.functions.SGD*
Voted Perceptron (VP)	*weka.predictors.functions.VotedPerceptron*
AdaBoost (AB)	*weka.predictors.meta.AdaBoostM1*
Decision Tree (DT)	*weka.predictors.trees.J48*
Logit Boost (LB)	*weka.predictors.meta.LogitBoost*
Random Tree (RT)	*weka.predictors.trees.RandomTree*
Random Forest (RF)	*weka.predictors.trees.RandomForest*
RIPPER	*weka.predictors.rules.JRip*
PART	*weka.predictors.rules.PART*
K-nearest Neighbors (KNN)	*weka.predictors.lazy.IBk*
Meta-classifiers	
Meta-classifier	**Scikit-learn class name**
Naive Bayes (NB)	*sklearn.naive_bayes.GaussianNB*
AdaBoost (AB)	*sklearn.ensemble.AdaBoostpredictor*
Decision Tree (DT)	*sklearn.tree.DecisionTreepredictor*
LogitBoost (LB)	*sklearn.ensemble.GradientBoostingpredictor*
K-nearest Neighbors (KNN)	*sklearn.neighbors.KNeighborspredictor*
Logistic Regression (LR)	sklearn.linear_model.LogisticRegression
Stochastic Gradient Descent (SGD)	sklearn.linear_model.SGDpredictor
Random Forest (RF)	*sklearn.ensemble.RandomForestpredictor*

However, in the case of learning supervised ensembles like ours that involve two rounds of training (first the base classifiers and then the ensembles), using standard cross-validation may lead to overfitting of the ensemble. Thus, as explained in our previous work[7], we devised a nested cross-validation procedure to be used for training and testing supervised ensembles. In this procedure, the entire dataset was split into *outer* training and test CV splits and each outer training split was further divided into *inner* CV folds. Base classifiers were trained on the inner training split and used to predict on the corresponding inner test split. Predictions made by the base classifiers were collected across all inner testing folds and used as the base data to train the heterogeneous ensembles. The outer test splits were then used to evaluate the performance of the trained ensembles. The nested cross-validation strategy ensures that the base classifiers and ensembles are trained on separate subsets of the data set, thus reducing the chances of bias and overfitting.

We addressed the potentially high computational costs by parallelizing all the independent units of the nested CV process, namely the training and testing of base and ensemble predictors over all the inner and outer CV splits. These units were then executed on separate processors in a large HPC cluster, with the outputs of inner CV folds flowing into the outer ones as described in our earlier work[7]. We have made this HPC-enabled implementation of the heterogeneous ensemble PFP process publicly available as LargeGOPred.

Statistical comparison of PFP performance

In this study, we compared multiple heterogeneous ensembles and base classifiers on their ability to predict annotations to a large number of GO terms. In such situations, it is critical to assess the statistical significance of these numerous comparisons to derive reliable conclusions. For this, we used Friedman's and Nemenyi's tests and visualized their results in easily interpretable critical difference (CD) diagrams[27]. Friedman's test ranks all the tested classifiers over all datasets (here, GO terms) and tests if the mean ranks of all classifiers are statistically equivalent, while Nemenyi's test performs the equivalent of multiple hypothesis correction for these comparisons. We used the scmamp (0.3.2)[31] R package to perform these tests and visualize their results as CD diagrams.

Results

Overall PFP performance

We first evaluated if and to what extent heterogeneous ensembles enable the prediction of protein function as compared to individual predictors. Figure 2 shows the results of this evaluation in terms of the difference of the performance of a variety of ensembles from that of the best base classifier for each GO term, with the terms themselves categorized by their sizes. Although there is substantial variability in the values of ΔF_{max} across ensemble methods and GO term categories, some trends can still be observed. First, the values of ΔF_{max} across ensembles increase as the sizes of the GO terms considered also

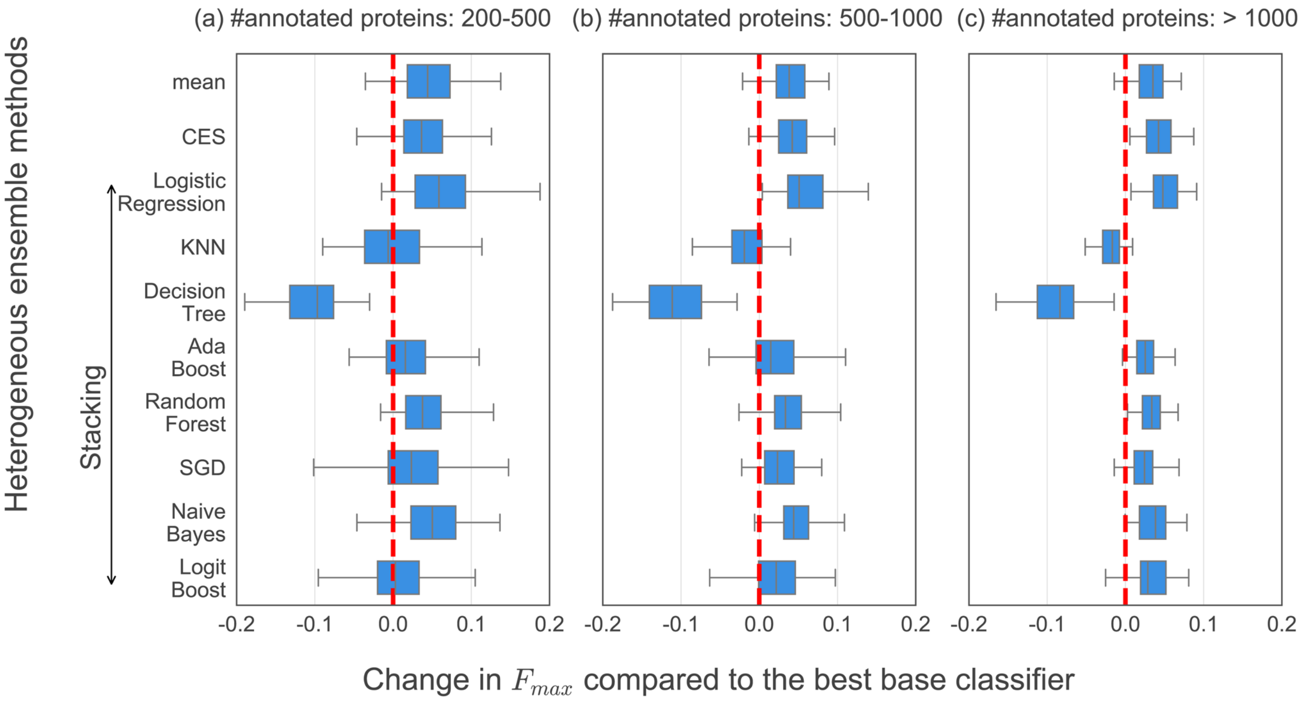

Figure 2. Boxplots denoting the distributions of the heterogeneous ensembles' PFP performance compared to that of the best base classifier for each GO term. The Y-axis shows all heterogeneous ensembles tested, specifically mean (aggregation), Caruana *et al.*'s ensemble selection (CES) and 8 stacking methods using different meta-classifiers named here. The X-axis denotes the difference between the F_{max} of each heterogeneous ensemble and the best base classifier for each GO term (ΔF_{max}), which are categorized into (**a**) 152 small, (**b**) 71 medium and (**c**) 54 large GO terms with 200-500, 500-1000 and over 1000 annotated sequences in our dataset (Table 1). The broken vertical red line in each subplot represents ΔF_{max}=0.

increase. This is illustrated by the fact that zero, one (Stacking with Logistic Regression) and four (CES and Stacking with Logistic Regression, Random Forest and Naive Bayes) ensembles produce ΔF_{max}>0 for every GO term tested in the small, medium and large categories (from left (a) to right (c) in Figure 2). This trend is expected, since the availability of more positively annotated genes in the larger GO terms enhances the ability of the ensembles, especially the supervised ones, to improve PFP performance. Due to the same reason of more training data, the variability of PFP performance for the large terms, represented by the widths of the boxes and whiskers, is smaller, illustrating increased robustness of the ensembles.

To analyze these results in further detail and derive reliable conclusions from them, we used Friedman's and Nemenyi's tests to statistically assess the ΔF_{max} values shown in Figure 2. Figure 3 shows the results of these tests visualized as Critical Difference (CD) diagrams for the three categories of GO terms shown in Figure 2A–C, as well as all of them taken together (Figure 2D). These results show that several heterogeneous ensemble methods, such as LR.S, NB.S, Mean, RF.S, CES and SGD.S, performed better than the respective best base classifier in terms of their average rank[27]. In contrast, KNN.S and DT.S performed worse than the best base classifier for each category of GO terms considered.

A consistent observation from Figure 3 is that Stacking using Logistic Regression (LR.S) performed the best among all the tested predictors (leftmost entry in the CD diagrams) regardless of the GO term category considered. It performed statistically equivalently with NB.S and CES for the small (Figure 3A) and large (Figure 3C) GO terms respectively, statistically confirming the observations made from Figure 2. In particular, LR.S exclusively performed the best among all the predictors over all the GO terms examined, consistent with its good performance over a limited number of GO terms in our previous work[7]. Thus, we further analyzed the performance of this predictor across the hierarchical structure of the Gene Ontology.

Performance of Stacking using Logistic Regression (LR.S) across the GO hierarchy

GO terms are not a flat set of labels, but are rather organized in hierarchical ontologies structured as directed acyclic graphs (DAGs)[5,6]. Terms vary in their depth, or level, with deeper terms representing more specific functions as compared to those at shallower levels. Using the definition of the level of a GO term as the length of the shortest path to it from the root of the hierarchy, implemented in the GOATOOLS python package (0.8.4)[32], we observed that the levels of the terms in our dataset varied between 1 and 8 (Figure 4(A)). In terms of the number of genes annotated, as expected, most of the annotations are to the

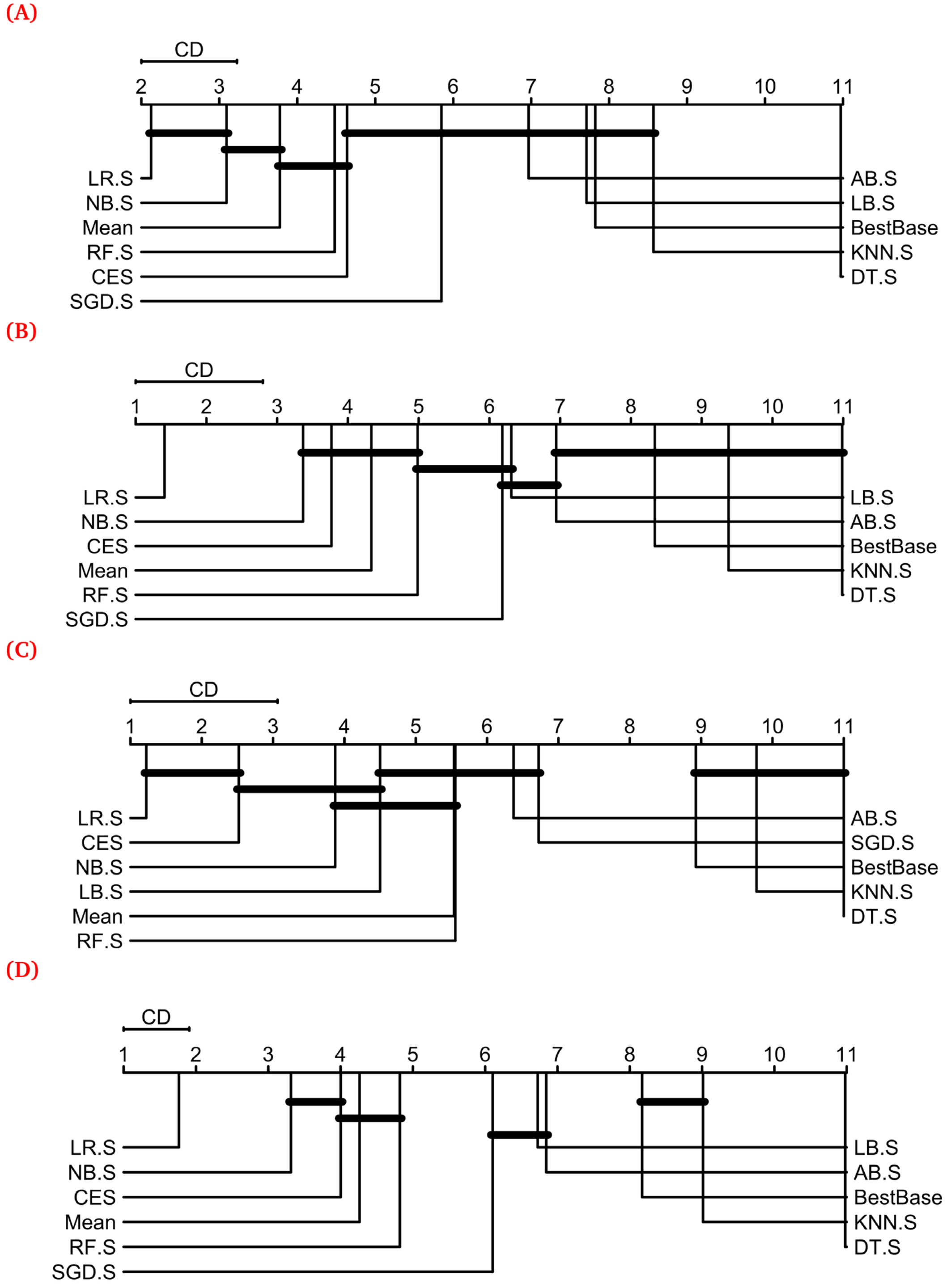

Figure 3. Critical Difference (CD) diagrams showing the results of a statistical comparison of the performance of all the heterogeneous ensemble methods shown in Figure 2 and the best base classifier for each GO term, conducted using Friedman and Nemenyi's tests[27]. In these diagrams, PFP methods, represented by vertical+horizontal lines, are displayed from left to right in terms of the average rank obtained by their resultant models for each GO term included. The groups of methods producing statistically equivalent performance are connected by horizontal lines. (**A**)–(**C**) show the CD diagrams for the three categories of GO terms shown in Figure 2, while (**D**) shows the one for all the 277 GO terms considered in this study. The *scmamp* R package[31] was used to perform the Friedman and Nemenyi's tests and plot the CD diagrams. Meta-classifiers used within stacking are denoted by their commonly used acronyms, e.g. LR for Logistic Regression, appended with ".S".

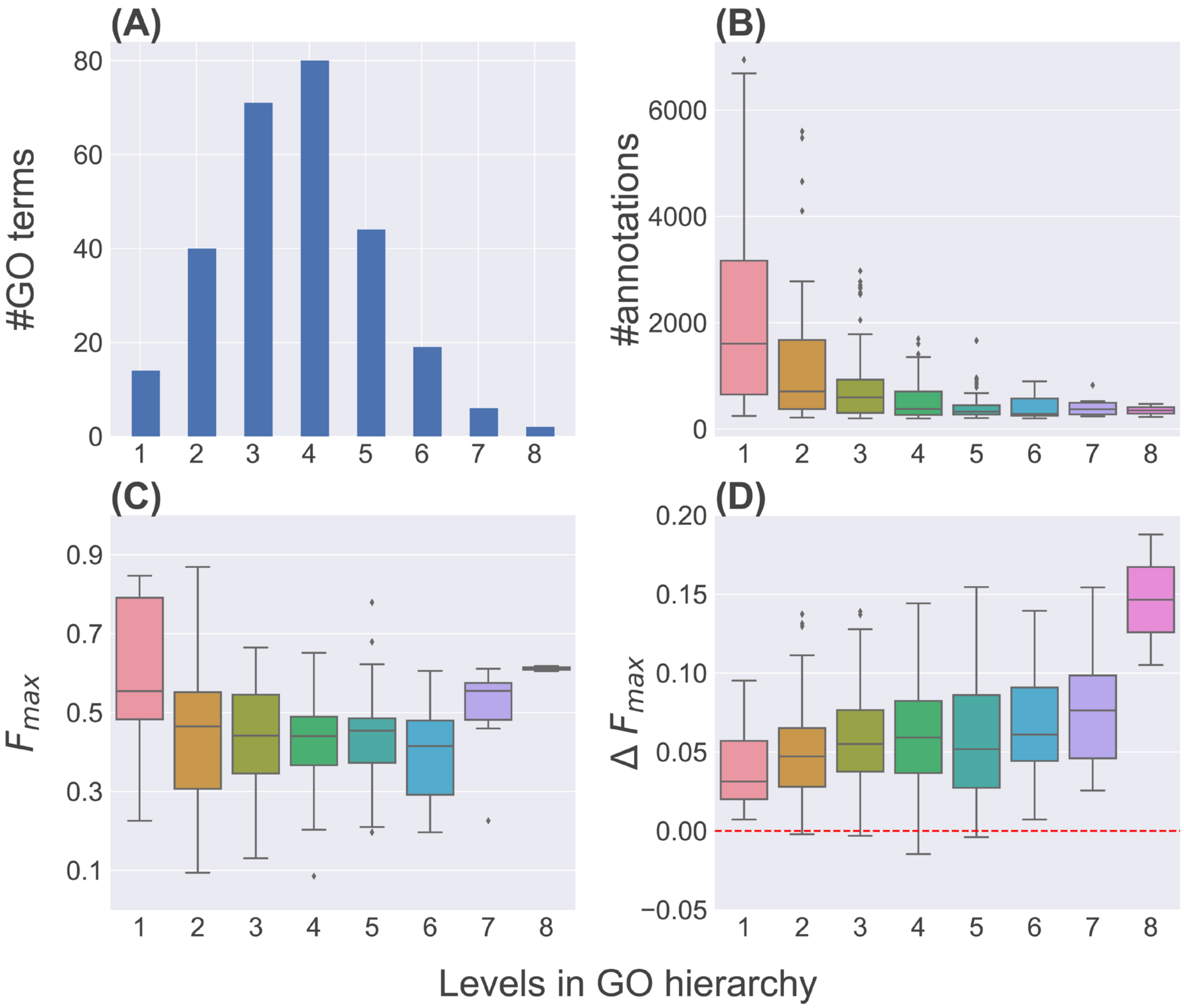

Figure 4. Performance of Logistic Regression (LR.S) for terms at different levels of the GO hierarchy. (**A**) and (**B**) show the distributions of the number of GO terms and the number of genes annotated to these terms at different levels respectively. (**C**) and (**D**) show the distributions of LR.S's F_{max} scores and their differences from the corresponding scores of the best classifier (ΔF_{max}) for these GO terms at the various levels.

shallower GO terms and only a small number to the deeper ones (Figure 4(B)).

We analyzed the ability of LR.S to predict annotations to these terms, measured in terms of F_{max}, at different levels (Figure 4(C)). The performance is reasonably high at level 1, but decreases gradually until level 6 due to fewer annotations available for training the base classifiers and ensembles (Figure 4(B)). The performance improves slightly at levels 7 and 8, likely due to the increased specificity of the corresponding terms and thus better signal in the corresponding training data.

Finally, we analyzed how LR.S's performance compared with that of the best classifier for the tested GO terms at different levels of the hierarchy. For this, we calculated and plotted in Figure 4(D) the same ΔF_{max} measure shown in Figure 2, this time categorized by levels. The results in Figure 4(D) show that ΔF_{max} increases overall for GO terms at increasingly deeper levels in the hierarchy. The increases are statistically significant (Wilcoxon rank-sign test p-value<0.05) at levels 1–7, although not significant (p-value=0.17) for only two terms at level 8 (Figure 4(A)). These results indicate the benefit heterogeneous ensembles, specifically LR.S, can provide for deeper GO terms with fewer annotations where individual predictors may not be effective.

Discussion

Owing to the diversity of available data types and computational methodologies, a variety of methods have been proposed for protein function prediction (PFP)[1,2]. CAFA[3,4] and other large-scale assessment efforts demonstrated that there is no ideal method for predicting different types of functions. In this paper, we have demonstrated a potential approach to address this problem, namely assimilating individual methods/predictors into heterogeneous ensembles that may be more robust, generalizable and predictive across functions. Although we had provided preliminary results supporting this approach in our previous work[7], those results were limited to predicting annotations to only three GO terms. In this paper, we report the first comprehensive and large-scale assessment of protein function prediction using heterogeneous ensembles. Specifically, using a data set of over 60,000 bacterial proteins annotated to almost 300 GO terms, we assessed how the mean aggregation, CES and stacking using multiple meta-classifiers performed for PFP.

Several of the tested heterogeneous ensembles performed better than the best base/individual predictor for many of the GO terms examined. In particular, the performance improvements obtained by heterogeneous ensembles generally increased with more annotations available for a given GO term, i.e. its size, which can be expected due to the larger amount of more positive data available for training the base predictors and ensembles.

A rigorous statistical comparison of all the heterogeneous ensembles and best base predictors tested over different categories of GO terms based on their sizes reaffirmed the effective performance of ensembles for PFP. In particular, Stacking using Logistic Regression (LR.S) was consistently the best-performing ensemble method across all the GO term categories, a finding consistent with our earlier work[7]. The effectiveness of LR.S can be attributed to the simplicity of the logistic regression function, which can help control overfitting at the meta-learning level during stacking. This effectiveness was also reflected in our observation that LR.S's is increasingly more accurately predictive for GO terms deeper in the hierarchy, for which the small number of annotations available may adversely affect individual predictors. Overall, our study and results demonstrate the potential of heterogeneous ensembles to advance protein function prediction on top of the progress in individual predictors already being reported in CAFA[3,4] and other exercises.

A key feature of our work was the effective utilization of high-performance computing (HPC) to enable efficient large-scale PFP. Specifically, using a large number processors in a sizeable HPC cluster, we successfully built and evaluated heterogeneous ensembles for over 60,000 bacterial proteins annotated to almost 300 GO terms in under 48 hours. While this increase in efficiency is already appreciable, it can be improved further by utilizing more parallelized formulations of the process, such as using parallel implementations of base classification methods[33] instead of the serial versions used in this work.

Although the results of our study are encouraging, they were derived using data from only 19 pathogenic species due to our group's general interest in PFP to better understand and predict annotated and unannotated pathogenicity in the context of clinically relevant bacteria. The inclusion of a larger number of and more diverse species, both prokaryotic and eukaryotic, in this evaluation can help assess how well our methods generalize to other species. The same can be said for including other types of data as well, such as the gene expression profiles used in our previous work[7].

We also only used normalized k-mer frequencies derived from amino acid sequences to represent proteins. This could be extended to test other representations such as short linear motifs (SLiMs)[34], hidden Markov models (HMMs)[35] and learned protein embeddings[36]. Moreover, regardless of the representation, another potential issue is that highly conserved and thus similar sequences across the 19 species tested in this study might be separated into both the training and test sets, which may result in an overestimation of prediction performance. Though UniProt controls for within species redundancy, it does not remove redundancy between species, an issue also true for our dataset. To address this issue, non-redundant versions of UniProt, such as UniRef100 or UniRef90[20], could be used to design more representative training and test sets. However, since the same prediction and evaluation process is used throughout our study, this issue should not adversely affect the fairness of the comparison between the performance of base predictors and heterogeneous ensembles.

Finally, in this study, we considered GO terms as independent units of protein function, but they are actually related because of their organization in the hierarchical structure of GO. Information from ancestors and closely related siblings in the hierarchy may provide useful information for protein function prediction, including through heterogeneous ensembles. Previous work has utilized this information for advancing individual and ensemble PFP algorithms[37–39], and similar ideas can be used to improve heterogeneous ensembles as well.

Data availability

The data underlying this study is available from Zenodo. Dataset 1: Data for LargeGOPred. http://doi.org/10.5281/zenodo.1434450[25]

This dataset is available under a Creative Commons Attribution 4.0

Software availability

Source code underlying this work is available from GitHub: https://github.com/GauravPandeyLab/LargeGOPred

Archived source code at time of publication http://doi.org/10.5281/zenodo.1434321[40]

License: GNU General Public License, version 2 (GPL-2.0)).

Author contributions

LW and GP conceived the study. LW carried out all the computational analyses and wrote the first draft of the manuscript. JL, SDK and TMM prepared the initial data used in the study and assisted with the evaluation of the results. GP supervised the work. All authors read, edited and approved the manuscript.

Grant information

This work was supported in part by National Institutes of Health [R01GM114434] and by an IBM faculty award to GP. It was also partially supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the Army Research Office (ARO) under Cooperative Agreement Number [W911NF-17-2-0105]. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NIH, ODNI, IARPA, ARO, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgements

This work was enabled by the computational resources and staff expertise provided by Scientific Computing at the Icahn School of Medicine at Mount Sinai. We would also like to thank members of the FunGCAT/IGACAT team, as well as colleagues at Mount Sinai, for discussions, suggestions and criticisms of this study.

References

1. Pandey G, Kumar V, Steinbach M: **Computational Approaches for Protein Function Prediction: A Survey.** Technical Report 06-028, University of Minnesota, 2006.
Reference Source
2. Sharan R, Ulitsky I, Shamir R: **Network-based prediction of protein function.** *Mol Syst Biol.* 2007; **3**(1): 88.
PubMed Abstract | Publisher Full Text | Free Full Text
3. Radivojac P, Clark WT, Oron TR, *et al.*: **A large-scale evaluation of computational protein function prediction.** *Nat Methods.* 2013; **10**(3): 221–7.
PubMed Abstract | Publisher Full Text | Free Full Text
4. Jiang Y, Oron TR, Clark WT, *et al.*: **An expanded evaluation of protein function prediction methods shows an improvement in accuracy.** *Genome Biol.* 2016; **17**(1): 184.
PubMed Abstract | Publisher Full Text | Free Full Text
5. Ashburner M, Ball CA, Blake JA, *et al.*: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nat Genet.* 2000; **25**(1): 25–9.
PubMed Abstract | Publisher Full Text | Free Full Text
6. The Gene Ontology Consortium: **Expansion of the Gene Ontology knowledgebase and resources.** *Nucleic Acids Res.* 2017; **45**(D1): D331–D338.
PubMed Abstract | Publisher Full Text | Free Full Text
7. Whalen S, Pandey OP, Pandey G: **Predicting protein function and other biomedical characteristics with heterogeneous ensembles.** *Methods.* 2016; **93**: 92–102.
PubMed Abstract | Publisher Full Text | Free Full Text
8. Wolpert DH: **Stacked Generalization.** *Neural Netw.* 1992; **5**(2): 241–259.
Publisher Full Text
9. Caruana R, Niculescu-Mizil A, Crew G, *et al.*: **Ensemble selection from libraries of models.** In *Proceedings of the Twenty-first International Conference on Machine Learning.* 2004; 18.
Publisher Full Text
10. Caruana R, Munson A, Niculescu-Mizil A: **Getting the Most Out of Ensemble Selection**. In *Proceedings of the Sixth International Conference on Data Mining.* 2006; 828–833.
Publisher Full Text
11. Stanescu A, Pandey G: **Learning Parsimonious Ensembles For Unbalanced Computational Genomics Problems**. In *Pac Symp Biocomput.* 2017; **22**: 288–299.
PubMed Abstract | Publisher Full Text
12. Altmann A, Rosen-Zvi M, Prosperi M, *et al.*: **Comparison of classifier fusion methods for predicting response to anti HIV-1 therapy.** *PLoS One.* 2008; **3**(10): e3470.
PubMed Abstract | Publisher Full Text | Free Full Text
13. Tuarob S, Tucker CS, Salathe M, *et al.*: **An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages.** *J Biomed Inform.* 2014; **49**: 255–268.
PubMed Abstract | Publisher Full Text
14. Wang H, Zhao T: **Identifying named entities in biomedical text based on stacked generalization**. In *Proceedings of the 7th World Congress on Intelligent Control and Automation.* 2008; 160–164.
Publisher Full Text
15. Niculescu-Mizil A, Perlich C, Swirszcz G, *et al.*: **Winning the KDD Cup Orange Challenge with Ensemble Selection.** *J Mach Learn Res.* 2009; **7**: 23–34.
Reference Source
16. Varma S, Simon R: **Bias in error estimation when using cross-validation for model selection.** *BMC Bioinformatics.* 2006; **7**(1): 91.
PubMed Abstract | Publisher Full Text | Free Full Text
17. Friedman JH: **Stochastic gradient boosting.** *Comput Stat Data Anal.* 2002; **38**(4): 367–378.
Publisher Full Text
18. Centers for Disease Control and Prevention (CDC), Department of Health and Human Services (HHS): **Possession, Use, and Transfer of Select Agents and Toxins; Biennial Review of the List of Select Agents and Toxins and Enhanced Biosafety Requirements. Final rule.** *Fed Regist.* 2017; **82**(12): 6278–94.
PubMed Abstract
19. Santajit S, Indrawattana N: **Mechanisms of Antimicrobial Resistance in ESKAPE Pathogens.** *BioMed Res Int.* 2016; **2016**: 2475067.
PubMed Abstract | Publisher Full Text | Free Full Text
20. UniProt Consortium T: **UniProt: the universal protein knowledgebase.** *Nucleic Acids Res.* 2018; **46**(5): 2699.
PubMed Abstract | Publisher Full Text | Free Full Text
21. Mostafavi S, Ray D, Warde-Farley D, *et al.*: **GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function.** *Genome Biol.* 2008; **9 Suppl 1**: S4.
PubMed Abstract | Publisher Full Text | Free Full Text
22. Crusoe MR, Alameldin HF, Awad S, *et al.*: **The khmer software package: enabling efficient nucleotide sequence analysis [version 1; referees: 2 approved, 1 approved with reservations].** *F1000Res.* 2015; **4**: 900.
PubMed Abstract | Publisher Full Text | Free Full Text
23. Madera M, Calmus R, Thiltgen G, *et al.*: **Improving protein secondary structure prediction using a simple *k*-mer model.** *Bioinformatics.* 2010; **26**(5): 596–602.
PubMed Abstract | Publisher Full Text | Free Full Text
24. Muppirala UK, Honavar VG, Dobbs D: **Predicting RNA-protein interactions using only sequence information.** *BMC Bioinformatics.* 2011; **12**(1): 489.
PubMed Abstract | Publisher Full Text | Free Full Text
25. Linhua W: **Data for LargeGOPred [Data set].** *Zenodo.* 2018.
http://www.doi.org/10.5281/zenodo.1434450
26. Lever J, Krzywinski M, Altman N: **Points of significance: classification evaluation.** *Nat Methods.* 2016; **13**: 603–604.
Publisher Full Text
27. Demsar J: **Statistical Comparisons of Classifiers over Multiple Data Sets.** *J Mach Learn Res.* 2006; **7**: 1–30.
Reference Source
28. Hall M, Frank E, Holmes G, *et al.*: **The WEKA Data Mining Software: An Update.** *SIGKDD Explorations Newsletter.* 2009; **11**(1): 10–18.
Publisher Full Text
29. Arlot S, Celisse A: **A survey of cross-validation procedures for model selection.** *Stat Surv.* 2010; **4**: 40–79.
Publisher Full Text
30. Pedregosa F, Varoquaux G, Gramfort A, *et al.*: **Scikit-learn: Machine learning in Python.** *J Mach Learn Res.* 2011; **12**: 2825–2830.
Reference Source
31. Calvo B, Santafé G: **scmamp: Statistical comparison of multiple algorithms in multiple problems.** *R J.* 2016; **8/1**.
Reference Source
32. Klopfenstein DV, Zhang L, Pedersen BS, *et al.*: **GOATOOLS: A Python library for Gene Ontology analyses.** *Sci Rep.* 2018; **8**(1): 10872.
PubMed Abstract | Publisher Full Text | Free Full Text
33. Bekkerman R, Bilenko M, Langford J: **Scaling up machine learning: Parallel and distributed approaches**. Cambridge University Press, 2011.
Publisher Full Text
34. Haslam NJ, Shields DC: **Profile-based short linear protein motif discovery.** *BMC*

Bioinformatics. 2012; **13**(1): 104.
PubMed Abstract | Publisher Full Text | Free Full Text

35. Yoon BJ: **Hidden Markov Models and their Applications in Biological Sequence Analysis.** *Curr Genomics.* 2009; **10**(6): 402–415.
PubMed Abstract | Publisher Full Text | Free Full Text

36. Yang KK, Wu Z, Bedbrook CN, *et al.*: **Learned protein embeddings for machine learning.** *Bioinformatics.* 2018; **34**(15): 2642–2648.
PubMed Abstract | Publisher Full Text | Free Full Text

37. Pandey G, Myers CL, Kumar V: **Incorporating functional inter-relationships into protein function prediction algorithms.** *BMC Bioinformatics.* 2009; **10**(1): 142.
PubMed Abstract | Publisher Full Text | Free Full Text

38. Yu G, Luo W, Fu G, *et al.*: **Interspecies gene function prediction using semantic similarity.** *BMC Syst Biol.* 2016; **10**(Suppl 4): 121.
PubMed Abstract | Publisher Full Text | Free Full Text

39. Zhang L, Shah SK, Kakadiaris IA: **Hierarchical Multi-label Classification using Fully Associative Ensemble Learning.** *Pattern Recognit.* 2017; **70**: 89–103.
Publisher Full Text

40. linhuawang: **linhuawang/LargeGOPred: first release (Version 0.0.0).** *Zenodo.* 2018.
http://www.doi.org/10.5281/zenodo.1434321

Open Peer Review

Current Peer Review Status: ✓ ✓

Version 1

Reviewer Report 05 November 2018

https://doi.org/10.5256/f1000research.17934.r38881

✓ **Predrag Radivojac**
College of Computer and Information Science, Northeastern University, Boston, MA, 02115, USA

This study evaluates protein function prediction using heterogeneous ensembles. The authors collected a set of 19 organisms with functional annotations and used a complex cross-validation setup to explore the value of obtaining improved classification performance using model averaging, stacking, and previously proposed techniques by Caruana et al. They considered 277 binary classification problems, each with its own data set of positive and putatively negative genes. The base classifiers were built upon a simple 3-mer feature representation.

Overall, this work is well presented and is clear in its exposition and contributions: there is value in developing heterogeneous ensembles though the computational cost is significant (here, an HPC solution was necessary to complete the study). Simple stacking models with logistic regression seem to be performing the best. This comes as a small surprise because one would expect nonlinear models to have an edge. On the other hand the base models were already nonlinear which might contribute to this effect.

Software for this work is available which is a plus.

Specific comments:

1. (the basis for answering one of the questions with "partly") Page 3, "Data used in the study"

The authors say that no electronic annotations have been used, but the majority of the evidence codes provided is in fact electronic annotation. See

http://www.geneontology.org/page/guide-go-evidence-codes

Some of the results of this work might be less realistic if the models were trained on predicted annotations. On the other hand, given the state of annotation of bacterial genomes, it is not clear whether there was an alternative. Nonetheless, this requires clarification, discussion and changes in this paragraph or perhaps elsewhere too.

2. The authors refer to their previous work on the inner and outer cross-validation folds. Although I believe I understood the process, it would be useful to mention whether at any point a base classifier was

trained on a particular protein and then the stacked model included that same protein in its training.

3. Figure 1, lower part, ended up not being useful for me. Once we train an ensemble of base classifiers in step 3, I was confused by step 4. This seems to be some intermediate averaging that comes before stacking. This point would be good to explicitly point to the reader as it confused me at one point.

4. Not a mandatory request, but it would be useful to perform a leave-one-species-out type of accuracy estimation. This might combat the problems related to sequence similarity that are discussed near the end of the paper. It would also provide evidence on what to expect from computational models when a new species is sequenced.

5. The manuscript would greatly benefit from proofreading and learning up some sentence structure and language issues.

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Partly

Are sufficient details provided to allow replication of the method development and its use by others?
Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 24 October 2018

https://doi.org/10.5256/f1000research.17934.r38879

✓ **Guoxian Yu**
College of Computer and Information Sciences, Southwest University, Chongqing, China

This paper investigates the potential of heterogeneous ensembles for protein function prediction by quantitatively comparing several classical base classifiers and ensembles on them. This investigative study is interesting, innovative and informative for future study on protein function prediction. This manuscript is clearly presented, well designed and organized. This investigation can be further improved in the following aspects:

1. The used data are only Amino Acid sequences, will the results and conclusions be changed when other types of data are used and integrated? The heterogeneous ensembles are intended for heterogeneous data types.
2. The considered GO terms (annotated to 200-300 proteins) are quite small, compared with the large GO terms space, more specific GO terms (annotated to <200 and >=10 proteins) should be tested. PFP is an imbalanced function prediction problem.
3. Smin is another more stringent evaluation metric in CAFA, and it refers to GO hierarchy when measuring the performance. This metric should be additionally used to quantify the performance of PFP.
4. There are some classifier ensemble based PFP solutions omitted. They should be cited and acknowledged.

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Yes

Are sufficient details provided to allow replication of the method development and its use by others?
Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Gene function prediction, Bioinformatics, Data mining

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.