Microenvironment-based Protein Function Analysis by Random Forest

Kazunori Okada*†, Lorenzo Flores*, Mike Wong†, Dragutin Petkovic*†
*Computer Science Department, San Francisco State University San Francisco, CA
†Center for Computing for Life Science, San Francisco State University, CA
Email: {kazokada,ltflores,mikewong,petkovic}@sfsu.edu

***Abstract*—Machine learning-based prediction of protein functions plays a key role in bioinformatics and pharmaceutical research, facilitating swift discovery of new drugs in high-throughput settings. This paper presents an adaptation of Random Forest to the structure-based protein function prediction. Our system represents protein's 3D physicochemical structural information in microenvironment descriptors whose spatial resolution is much finer than other sequence-based protein descriptors. We prepare our datasets for seven active sites from five protein function classes by using multiple public data banks and train Random Forest classifiers to identify these seven function models in proteins. This paper presents two experiment studies: 1) a 5-fold stratified cross-validation for comparing Random Forest with Naive Bayes and Support Vector Machine and 2) systematic comparison of Random Forest's two variable importance measures. Promising results of these studies demonstrate a potential for Random Forest to improve the accuracy of the current protein function assays.**

I. Introduction

Recent advances in bioinformatics and pharmaceutical research have pushed the growth rate of protein structure databases far beyond our ability to manually curate and annotate. To maximize utility of scarce lab resources and facilitate swift discovery of new drugs, automated tools are critical for prioritizing assays of protein structure with unconfirmed function. Our work builds on top of FEATURE [1], [2]: one of the major public online tools for automatic prediction and annotation of protein function as a high-precision filter to identify lowest risk assays. FEATURE analyzes protein's physicochemical 3D structure in microenvironments [3] while many current popular methods rely on analyzing 1D protein sequences [4]. The sequence-based methods are spatially imprecise, annotating sequences that can be hundreds of Angstroms long. On the other hands, the structure-based methods, such as FEATURE, can identify functional class and position within a few Angstroms in 3D space. Therefore predictions by FEATURE can be used in various position-sensitive applications, such as pocket recognition [5] and time series 3D analysis for studying molecular dynamics trajectories produced *in silico* [6].

The goal of the present research is to improve FEATURE's predictive performance in recall at a high precision setting, which has a tremendous payoff across multiple research and pharmaceutical applications. At the heart of the FEATURE is a predictive model of protein functions, built as a supervised machine learning classifier mapping the protein structure to its specific function via identifying a group of *active sites*: reactive atoms in key residues. In its original release, FEATURE employed Naive Bayes (NB) classifier for this purpose [2]. More recently, Support Vector Machines (SVM) has been adapted to FEATURE for improving its prediction capabilities [7].

The main contribution of this paper is to introduce Random Forest as a new predictive model for 3D protein function analysis using FEATURE. Random Forest (RF) [8] is a popular ensemble supervised classification method, which consists of a set of binary CART decision trees, combining the theory of Bagging [9] and Random Subspace [10]. Due to its superior accuracy and robustness, RF has effectively been applied to various machine learning applications, including bioinformatics [11] and medical imaging [12], however RF's application to the structure-based protein function analysis has not yet been fully explored in literature [13].

This paper presents a systematic experimental validation of RF, SVM, and NB in the FEATURE framework tested on seven protein function models. In order to account for the class-imbalance in our dataset, our experiments employ a stratified K-fold cross validation. The results demonstrate significantly improved recalls of our RF-based solution over the previous methods. Another unique advantage of RF over other competing machine learning methods is a set of powerful data mining tools for measuring importance of each input feature [14], [15]. Two such variable importance measures, mean-decrease-Gini (MDG) and mean-decrease-accuracy (MDA), are applied to our data and analyzed for identifying key features that strongly contribute to predicting protein functionality. We provide qualitative analyses of the selected features by the two measures.

To the best of our knowledge, the presented work is the first to apply RF for generalized protein function prediction with fully structure-based protein descriptors. Qi et al [16] used features from multiple sources, including high-throughput gene expression and sequenced-based information. Chen et al [17] and Sikic et al. [18] used RF to predict protein-protein interactions. The former was based on Pfam HMM profiles, which is also sequence-based, and the latter used secondary structural motifs and physicochemical properties inferred from 1D amino acid sequences, finding that pure sequence-derived features obtained the largest MDA values. Microenvironments descriptor used in this study offers structural information much finer than that explored in these previous studies. Moreover the systematic comparison of MDA and MDG in our application context is another major and unique contribution of this paper.

II. Data: Microenvironments, Public Data Banks, Function Models

In FEATURE, 3D structural information is encoded into a feature vector by analyzing properties of atoms in small local regions referred to as *microenvironments* [3], [2]. A microenvironment consists of 6 concentric spherical shells, each 1.25 Angstrom thick, centered at a reactive atom of a biomolecular structure such as a protein. Each shell may contain atoms, which are treated as 3D points. For each atom, we calculate 80 physicochemical properties using information obtained from public protein structure databases and software: 1) RCSB Protein Data Bank (PDB) [19] for atom coordinate lists as the source of the most protein structure properties, 2) Dictionary of Protein Secondary Structure (DSSP) [1], [20] for properties related to secondary structure, and 3) a molecular simulation software, Assisted Model Building with Energy Refinement (AMBER) [21], whose parameter files are used to provision the properties related to bond geometries. The structural properties we consider include partial charge, Van der Waals radius, element (C, H, O, N, or S), residue name, secondary structure, etc. In each shell, each of the 80 properties are summed over all atoms present. Thus a microenvironment is characterized by a fixed-length feature vector of 480 coefficients for 80 properties in 6 shells.

Seven sets of validation data that we refer to as *function models* are prepared for this study. Protein active sites are evolutionarily conserved functional regions in proteins structures. An active site's function is facilitated by *reactive atoms* in key residues at specific positions in the active site. Each function model is prepared to identify a specific reactive atom of a target function, as identified by PROSITE [22]: an internationally-supported knowledge base for protein functions. Given a function model, prediction of a reactive atom in a protein structure implies the structure may have an active site. Therefore, supervised learning of our protein function predictor requires both positive and negative training samples with ground-truth labels of specific active site. For a chosen active site, we collect microenvironments computed from all structures in the PDB that match with the PROSITE pattern of the target function and have the same residue at the given pattern position. We label these microenvironments as positive samples. For negative samples, we first consider all structures from the entire PDB that are similar to but different from the target pattern and then randomly sample 50,000 microenvironments without replacement. Samples with deprecated, withdrawn, or otherwise missing protein structures are then discarded, resulting with less than 50,000 negative samples for some function models. These microenvironments that include atoms similar to the positive one is then labeled as negative samples.

Table I summarizes our seven function models. We specify each function model by a naming convention with four fields delimitated by a period. Each of the four fields from left to right denotes 1) PROSITE identifier of the target protein pattern (e.g. ASP_PROTEASE), 2) the position of amino acid sequence of the PROSITE pattern (e.g., 4(-th)), 3) a key residue of a protein function (e.g., ASP), and 4) a reactive atom in the residue (e.g., OD1). The fields 2, 3, and 4 identify the functional microenvironment in the active site. The PROSITE patterns for ASP_PROTEASE, EF_HAND_1, IG_MHC, PROTEIN_KINASE_ST, and TRYPSIN_HIS match proteins with the following functions: 1) aspartyl proteases, important to HIV replication, 2) EF-hand calcium binding domain, 3) immune system antibody domains, 4) serine/threonine protein kinases, important to cancer research, and 5) trypsin-like serine proteases, respectively. ASP, GLN, CYS and HIS specifies residues of aspartates, glutamines, cysteines, and histidines, respectively. OD1, OD2, NE2, SG, and ND1 indicates reactive atoms of the first delta oxygen, the second delta oxygen, the second epsilon nitrogen, the gamma sulfied, and the first delta nitrogen, respectively. These seven models are chosen due to relative familiarity in literature and availability of test results by previous predictive models of NB and SVM. The total number of samples N for each function model ranges from $47,212$ to $51,081$. And the data skew defined by the ratio of the number of negatives (# Neg) to positives (# Pos) ranges from 24.3 (IG_MHC.3.CYS.SG) to 3146.5 (EF_HAND_1.9.GLN.NE2), indicating high class-imbalance in our data.

TABLE I. Function models and their sample statistics.

Function Model	#Pos	#Neg	N	Skew(#N/#P)
ASP_PROTEASE.4.ASP.OD1	1585	47855	49440	30.2
EF_HAND_1.1.ASP.OD1	1811	47855	49666	27.4
EF_HAND_1.1.ASP.OD2	1811	47855	49666	27.4
EF_HAND_1.9.GLN.NE2	15	47197	47212	3146.5
IG_MHC.3.CYS.SG	2017	49064	51081	24.3
PROTEIN_KINASE_ST.5.ASP.OD1	1096	48924	50020	44.6
TRYPSIN_HIS.5.HIS.ND1	446	50000	50446	112.1

III. Methods

A. Random Forest Classifier

Random Forest (RF) classifier [8] is an ensemble of decision tree (DT) classifiers. Given a function model $\Theta = \{(\Theta_n, l_n)\}_{n=1}^{N}$ as training data, RF is trained with two free parameters *mTry* (the number of features sampled randomly when growing each tree node) and *nTree* (the number of trees in RF), where each feature vector Θ_n contains $M = 480$ physicochemical properties and a binary classification $l_n \in \{-,+\}$ is considered.

In training, Θ is first randomly sampled with replacement, yielding *nTree* bootstrapped datasets of N samples [9]. With each bootstrapped set, a DT is built by following the CART algorithm [23] except that the best feature at each tree node is selected only from a randomly sampled subset containing *mTry* $< M$ proprieties [10]. We denote a trained RF as $h(\mathbf{x}, \Theta, nTree, mTry)$, where $\mathbf{x}$ represents a test observation.

A novel test observation $\mathbf{x}$ is then classified by the plurality of binary decisions by the *nTree* DTs given $\mathbf{x}$ as the input. The bootstrap sampling in the above training procedure leaves roughly one-third of total data unused for training in each bootstrapped set. These unused samples are called *out-of-bag* (OOB) samples and are used to estimate generalization errors of $h(\mathbf{x}, \Theta, nTree, mTry)$ by averaging the errors for each OOB cases.

B. Variable Importance by Random Forest

RF provides variable importance measures: ways to rank each feature in terms of the amount of its contribution for making correct classifications so that important features can be selected. We employ two such measures that come with the standard RF algorithm: mean decrease Gini (MDG) and mean decrease accuracy (MDA) [8], [14].

TABLE II. PARAMETER SPACES FOR RF, SVN, AND NB.

Algorithm	Parameter Space	#Param
RF	*nTree* = $\{100, 500, 1000, 1500\}$, *mTry* = $\{5, 10, 20, 30, 40, 50\}$	24
SVM-1	*C* = $\{-5, -4, ..., -2, -1, 0, 1, 2, ..., 4, 5\}$	11
SVM-2	*C* = $\{-10, -9, ..., -2, -1, 0, 1, 2, ..., 9, 10\}$	21
NB	*P* = $\{-6.0, -4.0, -2.0, -1.0, -0.3, -0.1\}$	6

TABLE III. MINIMUM AVERAGE OOB ERRORS.

Function Model	nTree	mTry	aveOOBerr (%)
ASP_PROTEASE.4.ASP.OD1	500	20	2.9 ±0.6
EF_HAND_1.1.ASP.OD1	500	50	25.5 ±0.5
EF_HAND_1.1.ASP.OD2	750	50	24.0 ±0.5
EF_HAND_1.9.GLN.NE2	100	5	3.0 ±0.0
IG_MHC.3.CYS.SG	1000	40	12.3 ±0.5
PROTEIN_KINASE_ST.5.ASP.OD1	1000	50	10.0 ±0.4
TRYPSIN_HIS.5.HIS.ND1	1000	50	4.6 ±0.5

MDG is a filter-type measure that is computed when a RF is trained. RF training builds each DT by iteratively partitioning the dataset along the best feature which increases the label-homogeneity in the resulting partitions most or equivalently decreasing Gini impurity most. MDG records these Gini decreases at every node of RF then aggregates them along their corresponding features.

MDA is a wrapper-type measure that is computed with the OOB test cases. MDA measures the average increase of the error rate (i.e., decrease of accuracy) against random permutation of values across OOB cases. With a trained RF, the values of OOB cases for a tree are first permuted along the m-th feature. Then error rate with and without this permutation are recorded and their difference computed. This is repeated for all DTs and the average of these differences gives the m-th feature's MDA.

MDA often results in more robust measure than MDG. However MDA computation is more time consuming than MDG. It is also stochastic, so multiple runs on the same RF, using the same data, can yield different results.

IV. EXPERIMENTS

The following describes our experimental study for assessing the efficacy of RF for protein function prediction and for discovering key physicochemical properties in specific protein functions. Throughout this study we employ R's RF software package *randomForest 4.6-7* [14]. To validate its merit, RF is compared against two other popular classifiers: SVM and NB. For NB, we employ FEATURE's Naive Bayes implementation described in [2]. One of the standard SVM packages, libsvm [24], is also used. Our cross-validation experiments are scripted in Perl and executed on computer clusters: Amazon AWS and our in-house Sun Grid Engine cluster system with 160 compute cores.

In order to assure a fair comparison of the three classification models, we first explore the best parameter settings by exhaustive grid search. Table II lists the range and the interval of the parameter spaces for the three classification methods explored by our grid search. SVM's parameter *C* indicates an exponent of the penalty for mis-classification in the linear soft-margin SVM cost used in this study: $\text{argmin}_{\mathbf{w},\xi_i,b}\{\frac{1}{2}||\mathbf{w}||^2 + 2^C \sum_{i=1}^{n} \xi_i\}$ subject for all $i = 1, .., n$ to $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i$, $\xi_i \geq 0$, where $\mathbf{w} \cdot \mathbf{x} - b = 0$ defines the classification hyperplane and ξ_i is a slack variable measuring the degree of error for $\mathbf{x}_i$. NB's parameter *P* is an exponent of the prior probability for positive classification: P(positive) = 10^P. The NB's parameter range with the 6 values follows the original work by Wei and Altman in [25]. As our performance statistics, we elect to use averaged recall value at 99% precision computed by 5-fold cross validation (CV). For each function model, we select the parameter setting that maximizes this statistic. When multiple parameter settings yield the same performance, we opt for small values. Note that the training procedure of RF is stochastic, while the other two methods are deterministic, in that we would have different RF classifiers when training a RF multiple times on exactly the same training data. To marginalize over this random factor, we repeat the RF's training ten times without fixing the random seed then average the recall values over these 10 repeats. The resulting parameters of RF's (*nTree*,*mTry*) are (1500,30), (1500,20), (500,20), (500,10), (500,40), (1000,40), and (1500,50) for ASP_PROTEASE*, EF_HAND*OD1, EF_HAND*OD2, EF_HAND*NE2, IG_MHC*, PROTEIN_KINASE*, and TRYPSIN_HIS*, respectively

The OOB error described in Sec III-A also offers a common performance statistic for RF. For reference, we compute the minimum average OOB errors among the same parameter space for the seven models, summarized in Table III. We found that the favorably low OOB errors in this result are misleading when dealing with data with high class-imbalance. In such a case, the OOB error estimates tend to report falsely optimistic performance by miss-classifying most positive samples. For example, the reported OOB error for the model EF_HAND_1.9.GLN.NE2 was 3% while missing 80% of the positive samples. For this reason, we elect not to use OOB error as our performance statistic. The usage of the recall measure with 5-fold CV also allows a fair comparison of the three classification methods since the OOB error is specific to RF thus not available to SVM and NB.

A. *Comparison of RF, SVM and NB by Stratified K-fold Cross-Validation*

The high class-imbalance in our datasets undermines the standard K-fold cross validation procedure. This is because the random sampling used in partitioning data into K folds may arbitrary alter the class distribution in each fold, which could arbitrarily skew the performance measure. One way to address this issue is to stratify the sampling scheme in our CV procedure. In this stratified K-fold CV, we independently partition samples to 5 folds for positive and negative sample pools first, then merge positive and negative folds to form the K folds that preserves the class distribution of the original dataset. The rest of the CV procedure is repeated after this. For testing with each fold, recall values are computed for precision values ranging from 0 to 1 in 0.005 step increments.

Table IV show the results of our comparison for RF, SVM and NB. The table displays the recall values in percent at 99% precision for the 7 models evaluated with the best parameter settings found by our grid search. The recall values for RF are averaged over the 10 repeats to marginalize over its randomness. We evaluate SVM with two parameter settings of narrow (SVM-1) and wide (SVM-2) ranges as defined in Table II. The best performance is indicated by bold-type for each model. Results show that for all 7 models RF outperformed NB and

TABLE IV. COMPARISON OF RF, SVM, NB BY RECALL MEASURE AT 99% PRECISION FOR THE 7 FUNCTION MODELS.

Function Model	**NB**	**SVM-1**	**SVM-2**	**RF**
ASP_PROTEASE.4.ASP.OD1	96.9	99.9	**100**	**100 ±0.0**
EF_HAND_1.1.ASP.OD1	69.1	87.5	93.9	**97.2 ±2.5**
EF_HAND_1.1.ASP.OD2	68.6	87.5	93.2	**96.2 ±1.9**
EF_HAND_1.9.GLN.NE2	13.3	20.0	27.1	**31.4 ±19.9**
IG_MHC.3.CYS.SG	74.1	90.2	**99.8**	95.8 ±1.4
PROTEIN_KINASE_ST.ASP.OD1	74.1	90.2	**98.6**	97.4 ±0.2
TRYPSIN_HIS.5.HIS.ND1	91.3	94.8	**96.8**	**96.9 ±0.3**
Average Recalls	69.6±27.2	81.4±27.5	87.1±26.6	**87.8 ±24.9**
p-values (vs NB, Wilcoxon)	-	0.2486	0.05502	**0.04716** *

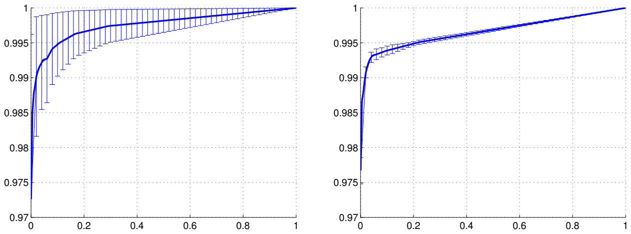

Fig. 1. Two example of ROC plots (sensitivity vs 1-specificity). Left: EF_HAND_1.1.ASP.OD1 where RF performed the best, Right: PROTEIN_KINASE_ST.5.ASP.OD1, where SVM performed the best.

SVM-1 with the narrow parameter range. When comparing RF with SVM-2 with the wider parameter range, RF's performance was similar to SVM. RF's performance was better than SVM-2 for the three EF_HAND_1 models, roughly equivalent to SVM-2 for ASP_PROTEASE and TRYPSIN_HIS, and inferior to SVM-2 for IG_MHC and PROTEIN_KINASE. Note however that for the models all classifiers struggled with (e.g., EF_HAND_1), RF tends to perform better than SVM. On average over the seven models, RF scored the best at 87.8±24.9%. Wilcoxon rank sum test revealed that RF performed significantly better than NB ($p = 0.047$), while we observed no statistically significant differences between other classifier pairs ($p > 0.05$). Fig. 1 shows two illustrative examples for mean ROC plots by RF: 1) a case that RF performed the best (EF_HAND_1*OD1) and 2) a case that SVM-2 performed the best (PROTEIN_KINASE*). Ten ROC curves are averaged by linearly interpolating sensitivity values at a set of fixed specificity values. We observed a tendency of a wider variance for the RF's ROC curves computed over repeated CV tests when training for challenging function models.

B. Variable Importance Analysis: Identification of Key Features

MDG and MDA measures of RF are used to explore important features for the 7 function models. To address the stochastic nature of RF, we repeated (10 times) the sequence of RF training with entire model dataset followed by computing the MDG and MDA measures for all 480 features. The results are averaged over the 10 repeats for each feature then we sort the features according to the mean measures. Figs. 2-8 compare the top 10 features by MDG and MDA for the 7 models, respectively. A detail caption for these figures are provided in Fig. 2.

For many of the 7 models, features selected by MDG and MDA agreed. Models, ASP_PROTEASE*, EF_HAND*OD1, EF_HAND*OD2, EF_HAND*NE2, IG_MHC*, PROTEIN_KINASE*, and TRYPSIN_HIS* had respective 8, 3, 2, 4, 6, 7, and 8 features shared among the top 10 list of MDG and MDA. For ASP_PROTEASE* and IG_MHC*, the top three features (RESIDUE_NAME_IS_GLY_shell2, RESIDUE_CLASS1_IS_UNKNOWN_shell2, RESIDUE_NAME_IS_THR_shell4) and the top two features (SECONDARY_STRUCTURE1_IS_STRAND_shell5, SECONDARY_STRUCTURE1_IS_STRAND_shell4) were ranked exactly same by MDG and MDA, respectively. PROTEIN_KINASE* shared the 2nd feature (SECONDARY_STRUCTURE1_IS_COIL_shell5), 4th feature (SECONDARY_STRUCTURE1_IS_3HELIX_shell5), and 5th feature (SECONDARY_STRUCTURE1_IS_3HELIX_shell4) and EF_HAND_1*OD1 shared the 7th feature (SECONDARY_STRUCTURE1_IS_BEND_shell3) between MDG and MDA. Interestingly, these degrees of agreement between top features selected by MDG and MDA correlate with RF's accuracy estimated by our CV experiment in Table IV, suggesting a potential importance for considering both MDG and MDA together. Qualitatively, MDG tends to come with greater change of importance values and more break-points (i.e., a large change of value between successively ranked features) than MDA. We also observe that some selected features are biochemically plausible (e.g., glycine (required at the 2nd residue position away from the reactive aspartate) selected 1st for ASP_PROTEASE; and solvent accessibility (needed for EF_HAND_1 loop) and 4helix at shell5 (in its helix-loop-helix motif) selected 1st and 2nd by MDA for both EF_HAND_1*OD1 and EF_HAND_1*OD2), demonstrating their potential merit.

V. CONCLUSIONS

This paper introduces an adaptation of RF to improve the accuracy of structure-based protein function prediction with microenvironment descriptors used in the FEATURE framework. Using a well-balanced set of seven function models prepared from multiple public data banks, we conducted systematic experimental evaluations of our RF-based system in its accuracy with respect to SVM and NB, and in its efficacy of the two VI measures. Our 5-fold stratified CV experiments show that RF's high accuracy is significantly better than NB and matches SVM and after careful parameter tuning. Our experiments on MDG and MDA shows that top features selected by the two measures often correlates with each other and also with the classification accuracy. These results suggest the potential merit of RF in generalized protein function assays.

One of the limitations of our system is due to the class-imbalance in dataset. In our dataset, EF_HAND*ND2 model was with an extreme data skew of a factor more than 3000. Although our RF-based system performed best among those tested, the absolute performance for this model was significantly lower than other models. This caused high standard

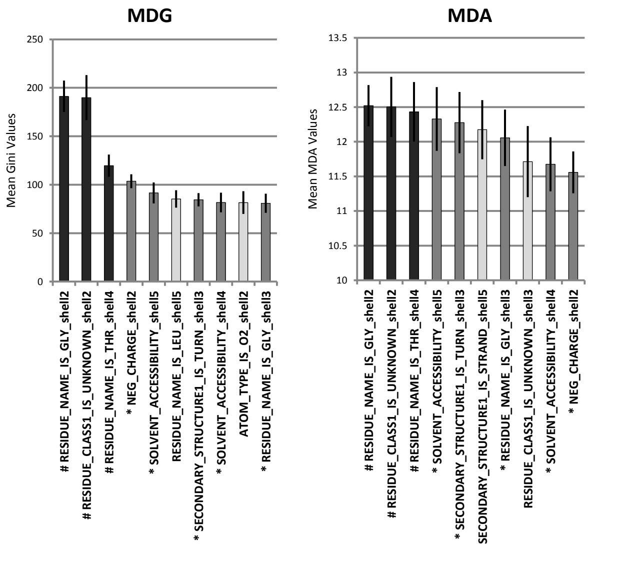

Fig. 2. Top 10 features selected by MDG and MDA for ASP_PROTEASE.4.ASP.OD1 model. In each plot, features are listed by the decreasing order of importance. Features noted by ”#” and by the black-colored bars indicate those ranked the same by MDG and MDA. Features noted by ”*” and by the gray-colored bars indicate those appeared in the top 10 list by both MDG and MDA.

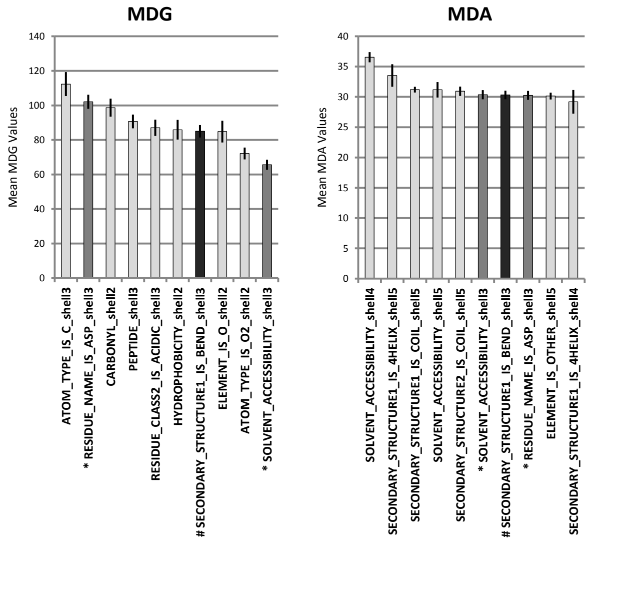

Fig. 3. For EF_HAND_1.1.ASP.OD1 model. See Fig. 2 for details.

deviations of the recalls in Table IV. Improving our system to better handle such high class-imbalance and evaluating its performance with more data are important future work for us. Furthermore, we plan to expand our qualitative analysis of the features selected by the VI measures with respect to the existing biochemical knowledge for more function models.

Acknowledgment

We thank Russ Altman, Byron Dom, Ljubomir Buturovic, and Grace Tang for their valuable advice and support. This work was partially supported by NIH grant 5R01LM005652, Amazon Web Services, and SFSU Center for Computing for Life Sciences.

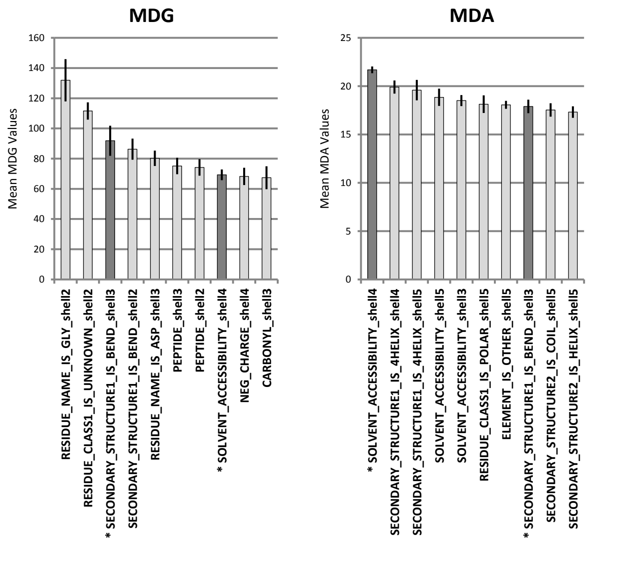

Fig. 4. For EF_HAND_1.1.ASP.OD2 model. See Fig. 2 for details.

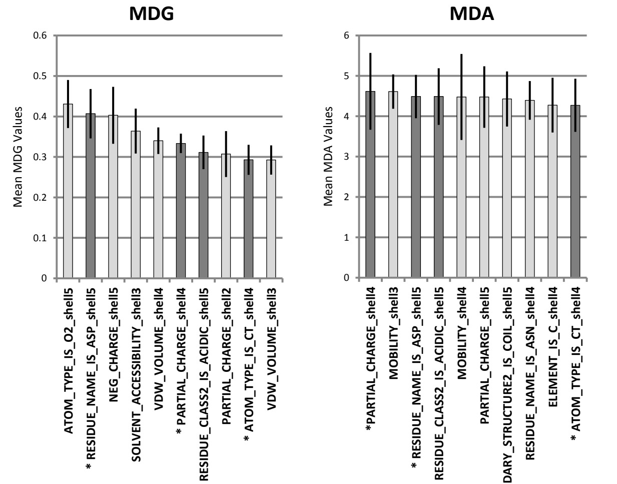

Fig. 5. For EF_HAND_1.9.GLN.NE2 model. See Fig. 2 for details.

References

[1] R. P. Joosten, T. A. H. te Beek, E. Krieger, M. L. Hekkelman, R. W. W. Hooft, R. Schneider, C. Sander, and G. Vriend, “A series of PDB related databases for everyday needs,” *Nucleic Acids Research*, vol. 39, pp. D411–9, 2011.

[2] L. Wei and R. B. Altman, “Recognizing complex, asymmetric functional sites in protein structures using a Bayesian scoring function,” *J Bioinform Comput Biol.*, vol. 1, no. 1, pp. 119–38, 2003.

[3] S. C. Bagley and R. B. Altman, “Characterizing the microenvironments surrounding protein sites,” *Protein Sci*, vol. 4, no. 4, pp. 622–35, 1995.

[4] I. Friedberg, “Automated protein function prediction - the genomic challenge,” *Briefings in Bioinformatics*, vol. 7, no. 3, pp. 225–242, 2006.

[5] T. Liu and R. B. Altman, “Using multiple microenvironments to find similar ligand-binding sites: application to kinase inhibitor binding,” *PLoS Comput. Biol.*, vol. 7, no. 12, p. e1002326, 2011.

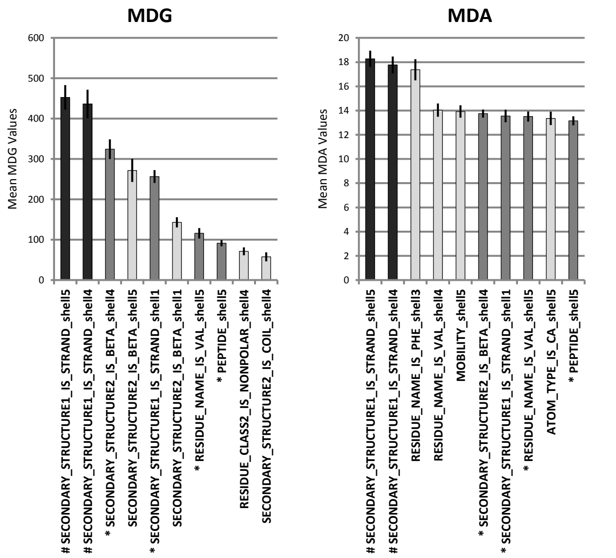

Fig. 6. For IG_MHC.3.CYS.SG model. See Fig. 2 for details.

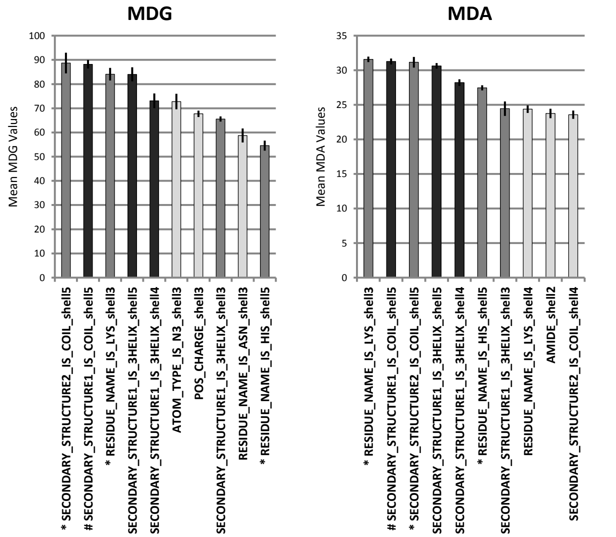

Fig. 7. For PROTEIN_KINASE_ST.5.ASP.OD1 model. See Fig. 2 for details.

[6] D. S. Glazer, R. J. Radmer, and R. B. Altman, "Improving structure-based function prediction using molecular dynamics," *Structure*, vol. 17, no. 7, pp. 919–929, 2009.

[7] H. Min, S. Yu, T. Lee, and S. Yoon, "Support vector machine based classification of 3-dimensional protein physicochemical environments for automated function annotation," *Archives of Pharmacal Research*, vol. 33, pp. 1451–1459, 2010.

[8] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] ——, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[10] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[11] R. Diaz-Uriarte and S. Alvarez de Andres, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, no. 1, p. 3, 2006.

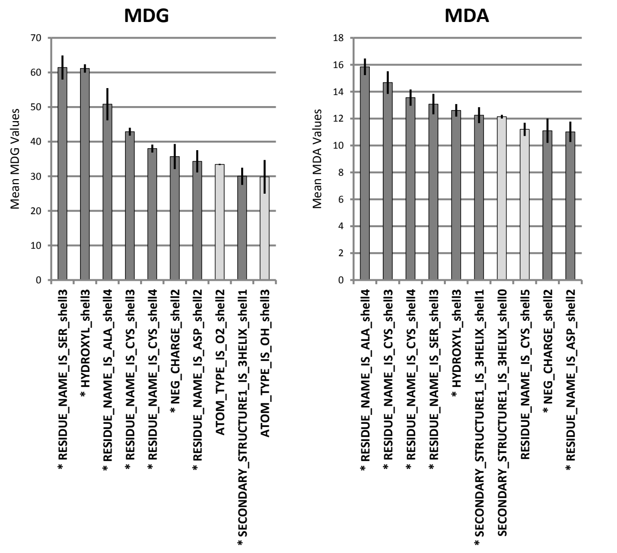

Fig. 8. For TRYPSIN_HIS.5.HIS.ND1 model. See Fig. 2 for details.

[12] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Found. Trends. Comput. Graph. Vis.*, vol. 7, no. 2-3, pp. 81–227, 2012.

[13] I. Ezkurdia, L. Bartoli, P. Fariselli, R. Casadio, A. Valencia, and M. L. Tress, "Progress and challenges in predicting proteinprotein interaction sites," *Briefings in Bioinformatics*, vol. 10, no. 3, pp. 233–246, 2009.

[14] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: http://CRAN.R-project.org/doc/Rnews/

[15] C. Strobl, A. L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, no. 1, p. 307, 2008.

[16] Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph, "Random forest similarity for protein-protein interaction prediction from multiple sources," in *Pac Symp Biocomput*, 2005, pp. 531–542.

[17] X. W. Chen and M. Liu, "Prediction of protein-protein interactions using random decision forest framework," *Bioinformatics*, vol. 21, pp. 4394–4400, 2005.

[18] M. Sikic, S. Tomic, and K. Vlahovicek, "Prediction of protein-protein interaction sites in sequences and 3D structures by random forests," *PLoS Comput Biol*, vol. 5, no. 1, p. e1000278, 2009.

[19] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[20] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983.

[21] D. A. Case et al., "Amber 12," 2012. [Online]. Available: http://ambermd.org/

[22] C. J. A. Sigrist, E. D. Castro, L. Cerutti, B. A. Cuche, N. Hulo, A. Bridge, L. Bougueleret, and I. Xenarios, "New and continuing developments at prosite." *Nucleic Acids Research*, vol. 41, pp. 344–347, 2013.

[23] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Chapman & Hall, New York, NY, 1984.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[25] L. Wei and R. B. Altman, "Recognizing protein binding sites using statistical descriptions of their 3D environments," in *Pac Symp Biocomput*, 1998, pp. 497–508.