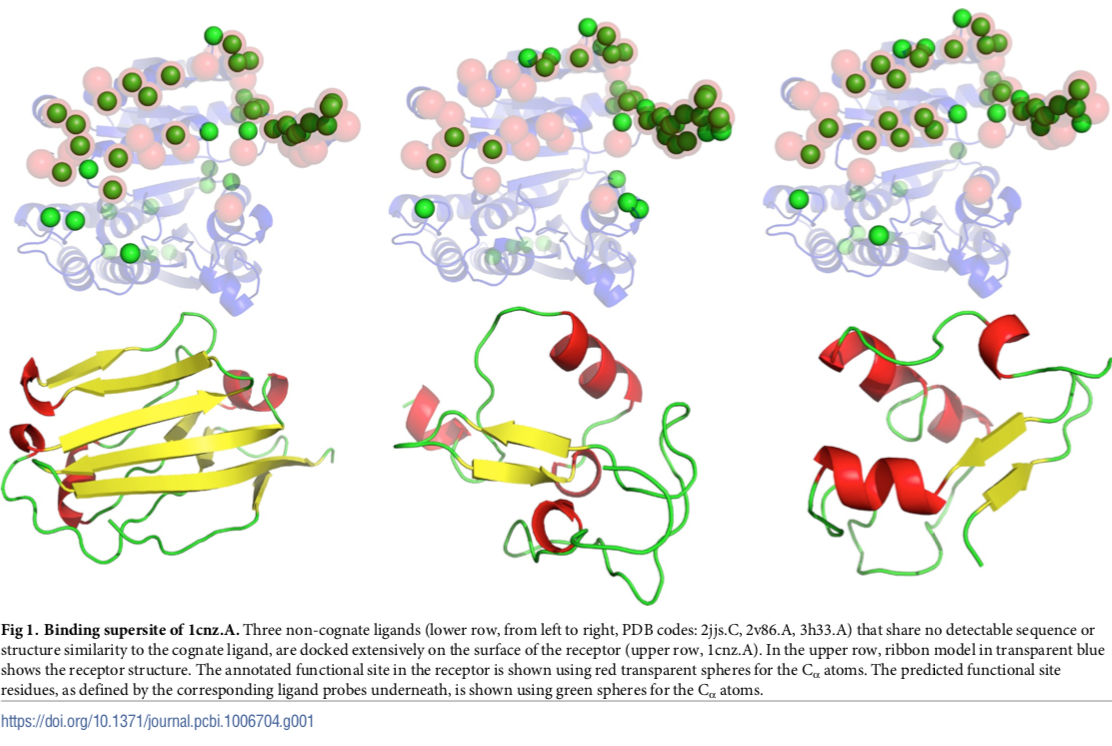
**II. Methods**

1. **DockPred**

It has previously been shown [2] that substrate and non-substrate small organic molecules have a tendency to bind to similar, energetically favorable sites on a target protein (“sticky” sites) regardless of their relevance to it. DockPred was created to test the hypothesis that proteins have a generic interaction site at which cognate and non-cognate ligands bind, just as it had been observed for small molecule “sticky” sites. The success of DockPred demonstrated that non-cognate ligands preferentially bind to the cognate binding site of a target protein (Figure 1). [3]



**Fig 1. [2] Binding supersite of 1cnz.A.** Three non-cognate ligands (lower row, from left to right, PDB codes: 2jjs.C, 2v86.A, 3h33.A) that share no detectable sequence or structure similarity to the cognate ligand, are docked extensively on the surface of the receptor (upper row, 1cnz.A). In the upper row, ribbon model in transparent blue shows the receptor structure. The annotated functional site in the receptor is shown using red transparent spheres for the Cα atoms. The predicted functional site residues, as defined by the corresponding ligand probes underneath, is shown using green spheres for the Cα atoms.

There were two datasets of query proteins used in the development of DockPred. One dataset contained 108 protein chains from the Docking Benchmark database, and the other contained 133 protein chains from the NOX database. [4, 5] Each query protein had a known complex and uncomplexed structure. The Contacts of Structural Units (CSU) program was used to define interface residues from the complex structures of the query proteins. [6] If any atom in a query residue was within 3.5Å of an atom in the query’s partner protein in the complexed structure, the residue was considered to be at the binding interface. This yielded the experimentally annotated interface residues for each query protein in the dataset.

13 non-cognate ligand probes were chosen to be used for DockPred based on their lack of sequence similarity to known ligands of the query proteins. The two docking programs ZDOCK and GRAMM were used to generate 2000 docked complexes for each uncomplexed query protein with each of the 13 non-cognate ligand probes. The CSU program was then used to analyze each of the docked complexes to identify the residues at the interface of each complex. A residue at the *i*th position of the query protein in the *k*th docked complex was denoted as Rik. I(Rik) = 0 for residues not at the interface of a docked complex, and I(Rik) = 1 for residues at the interface of a docked complex. For each residue at position *i* in the query protein, a Residue Interface Frequency (RIF) was calculated by summing over all docked complexes according to the formula

.

The top ranking residues with the largest Ni values of a query protein were considered to be the residues predicted to be at the binding interface. [3]

1. **PredUs 2.0**

The first version of PredUs, which was developed in 2011, made interface predictions for a query protein based on the known binding interfaces of the query’s structural neighbors. Two proteins are considered to be structural neighbors if their three-dimensional structures are similar. The secondary structure elements of two proteins can be similar without being composed of the same amino acids, so a prediction based only on structural similarity does not consider the query amino acids’ tendency to participate in binding. PredUs 2.0 was created in 2015 to address the flaw in making a structural template-based prediction alone. Using a Bayesian approach, PredUs 2.0 combines an amino acid interface propensity score with the template-based score of PredUs. [7]

The original PredUs program used the structural alignment program Ska to identify a query protein’s structural neighbors. Protein structural distance (PSD) is a measurement that quantifies the structural similarity between two proteins. It is calculated by superposing the two proteins’ structures in a manner that minimizes the root-mean-square deviation of the amino acids’ alpha carbons. [8] PredUs employed a PSD cutoff of 0.6 so that close and remote structural neighbors could be found. Neighbors that have a known complex structure were retained and ranked according to their structural alignment score. Using the program cd-hit [9], neighbors with a sequence similarity larger than 40% were grouped together, and only the protein with the higher PSD was kept. PredUs used the transformation relating the neighbor to the query protein to place the neighbor’s binding partner in the query’s coordinate system. If the neighbor’s binding partner was within 5Å of a query residue’s heavy atom (C, N, O), PredUs incremented the residue’s contact frequency score weighted by the PSD score between the neighbor and query. [10] After the transformation process was completed for every structural neighbor that was retained, each query residue has a total contact frequency score that sums the weighted scores from all the transformations.

PredUs used a support vector machine (SVM) algorithm to generate its template-based prediction score. Each residue r*i* on the query’s surface was combined with the 14 closest surface residues to form r*i*’s surface patch. A 31-element profile was assigned to thepatch, and it was composed of each residue’s contact frequency score and solvent accessible surface area (ASA), as well as the highest contact frequency score in the protein. The SVM mapped each patch profile to vectors in high-dimensional space, and it created a hyperplane that separated the vectors corresponding to interface residues from the vectors corresponding to non-interface residues. PredUs calculated an interfacial score for each residue based on its profile vector’s distance above or below the SVM hyperplane. [10]

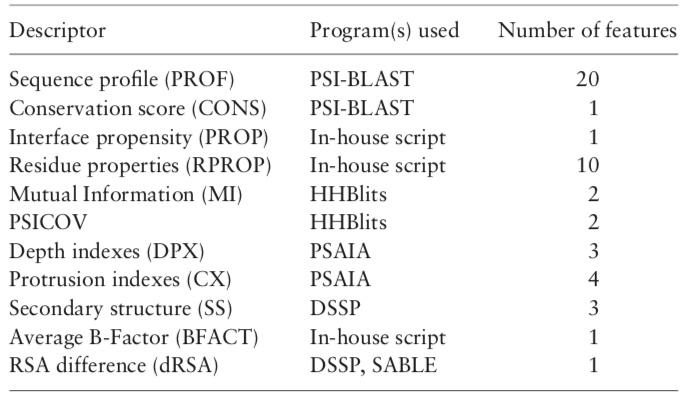
PredUs 2.0 calculated the interface propensity of each residue type *r* using a set of 2,766 heterodimeric complexes that had less than 40% sequence redundancy. Interface propensity was expressed as the relative ASA (RASA) contribution from residues of type *r* in protein-protein interfaces relative to their RASA contribution to the whole protein surface:

RASA­­*r*X represents the sum of RASAs of all type *r* residues with characteristic X (interface or surface) in all proteins from the set of heterodimeric complexes. RASA­*all*X represents the sum of RASAs of all residue types with characteristic X in all proteins from the dataset. RASA of protein residue is defined as the residue’s ASA as part of the protein, normalized by the residue’s area as part of an ALA-r-ALA tripeptide. Residues were considered to be at a protein’s surface if they had a RASA value of at least 0.05.

Since a residue’s interface propensity can vary with its RASA value, PredUs 2.0 calculated a weighted interface propensity score based on RASA values. The weighted propensity (WP) of a query protein residue *ri* of type *r* was calculated as follows:

The term represents the probability that a type *r* residue is at the interface, given its RASA value. Two scores were calculated for the surface patch associated with residue *ri*; one (WPA) was the average of the WPs of the patch residues, and the other (JP) was the joint probability for patch residues to be at the interface given their RASA values. The number of times a residue *ri* appeared in the top 15 patches ranked by WPA was denoted *n*, and the number of times a residue *ri* appeared in the top 15 patches ranked by JP was denoted *m*. The single patch score assigned to residue *ri* was (*n* + *m*). Using a naïve Bayes approach, PredUs 2.0 generated a likelihood ratio (LR) from the original PredUs and a LR from the propensity patch score. The final interface score that PredUs 2.0 assigns to each residue is the product of LRPredUs and LRpatch. [7]

1. **ISPRED4**

 ISPRED4 is one of the best performing structure-based protein binding interface predictors. It trained an SVM model on a dataset (DBv5Sel) of 314 different monomer chains with complex structures that had been resolved by X-ray crystallography. Interface residues were defined as those that lost at least 1Å2 of ASA (computed with the DSSP program [11]) when transitioning from a protein’s unbound to complex form. In the SVM model, each of the training proteins’ surface residues were represented by a 46-dimensional feature vector consisting of 10 different groups of descriptors (Table 1). ISPRED4 combined its SVM model with a Grammatical-Restrained Hidden Conditional Random Field (GRHCRF) to account for possible correlations between neighboring surface residues. For a given query protein, ISPRED4 calculated interface prediction scores by plugging the query residues’ feature vectors into its trained SVM/GRHCRF model. [12]

PSCICOV and MI are both used to calculate co-evolutionary scores, so they are considered together as one descriptor.

**Table 1.** [11] ISPRED4 Groups of Feature Descriptors

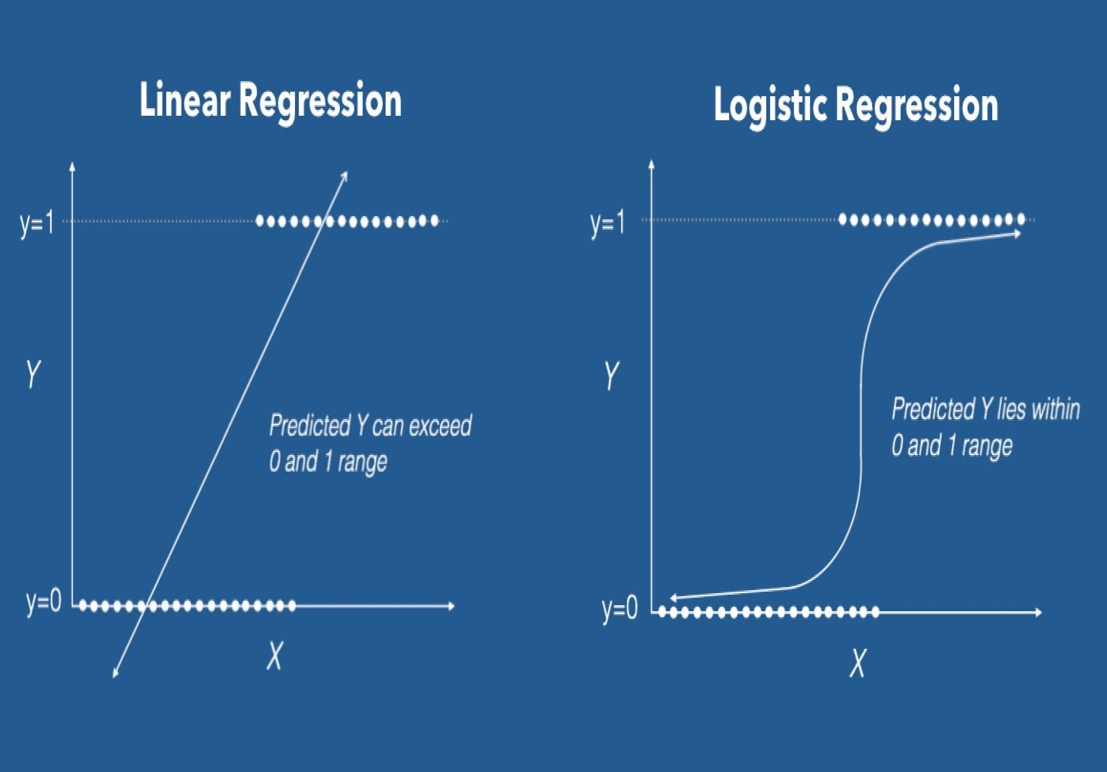
34 sequence-based features formed 5 groups of descriptors that encoded 34 dimensions of the feature vector. The sequence profile descriptor represented evolutionary information for each primary sequence position of a query protein. PSI-BLAST [13] was used to search the Uniprot Reference Cluster 90 database [14] for sequences similar to the query protein sequence, and the output served as the 20-dimensional sequence profile vector. Based on the sequence profile, a conservation score descriptor was calculated using the normalized Shannon’s entropy equation. [15] The interface propensity descriptor was calculated for each residue type *r* using the log-ratio of its interface frequency to its surface frequency. The 10 orthogonal properties introduced by Kidera *et al.* were incorporated into a group of 10 residue properties descriptors, which reflected the physico-chemical nature of each residue type *r*. [16] A multiple sequence alignment (MSA) for each query protein was generated using HHblits aligner against the UniprotKB database. [17] Based on the MSA, the PSICOV [18] and MI methods were each used to calculate two co-evolutionary scores, and each method’s scores formed a group of two descriptors.

12 structure-based features comprised 5 groups of descriptors that encoded 12 dimensions of the feature vector. ISPRED4 used the PSAIA toolkit [19] to compute protrusion and depth indexes for surface residues. Protrusion indexes consisted of a group of four descriptors, and depth indexes contained a group of three descriptors. Using the DSSP program, residues were assigned to one of three secondary structure classes: helix (H, G, I), strand (E, B), or coil (T, S). A group of three descriptors was computed for each residue *ri*, representing the frequency of helical, strand, and coil residues in its surface patch. An average B-factor descriptor was computed for a surface residue by averaging the B-factors for its individual atoms. The dRSA descriptor was calculated by subtracting a residue’s observed RSA value from its predicted RSA value (SABLE predictor [20]).

1. **Meta-DPI**

Meta-DPI was developed on the two datasets of query proteins used in the development of DockPred. The DB dataset contained 107 protein chains from the Docking Benchmark database, and the NOX dataset contained 116 protein chains from the NOX database (see Appendix A). [4, 5] Each residue of every query protein was assigned an interface score *i* of either 0 (non-interface) or 1 (at interface) based on the experimentally determined complex structures. DockPred, PredUs 2.0, and ISPRED4 were used to analyze the non-complexed query protein chains. The three methods generated prediction scores () for each residue of every query protein; the higher the score assigned to a residue, the more likely it was to be at the interface. Appendix B includes each method’s prediction file for an example query protein.

A logistic regression model was employed to determine how to combine the prediction scores of the three methods into a metamethod prediction score. Although linear regression has been used for some previous metamethods, such as meta-PPISP [21], logistic regression was chosen for meta-DPI because the interface score (dependent variable) can only have discrete values (0 or 1). Figure 2 illustrates how a logistic regression model fits discrete categorical data better than a linear regression model. The function used in a logistic model is:

where refers to the probability that given the values of . The target dependent variable is *i*, and the explanatory independent variables are . Based on previously collected data for *i* and *xj*, maximum likelihood is used to estimate the parameters *b0* and *bj* (“fitting” the logistic model). The likelihood function is a function of the unknown parameters, and it represents the joint probability of observing the obtained data. The parameters are estimated by setting each parameter’s partial derivative to zero (maximization), which results in a system of equations that can be solved iteratively with a computer program. [22]

**Fig 2. [23] Linear Vs. Logistic Regression.** The dependent variable Y only has discrete values of 0 and 1. The linear regression model does not fit the discrete data well, and it predicts Y values outside of the 0-1 range. The logistic regression model does fit the discrete data well, and it predicts Y values within the 0-1 range.

Each query protein residue served as a data point on which the logistic regression model was trained; the prediction scores from the individual methods (DockPred, PredUs 2.0, & ISPRED4) were the independent variables, and the interface score was the dependent variable. The logistic regression model was trained on each set of proteins separately in order to cross-validate the results. The coefficients generated from training the model on DB proteins were used to calculate metamethod prediction scores for NOX proteins, and the coefficients generated from training on NOX proteins were used to calculate metamethod prediction scores for DB proteins. The logistic function was used to calculate the metamethod prediction scores as follows:

where refers to the coefficient generated for PredUs 2.0, and refers to the prediction score assigned by PredUs 2.0 for a given residue, and the other coefficients similarly describe the other two methods.

In order to acquire a better insight into how each individual method contributed to meta-DPI, three additional metamethods were created by removing DockPred, PredUs 2.0, or ISPRED4. Meta-DI combined DockPred and ISPRED4, meta-DP combined DockPred and PredUs 2.0, and meta-PI combined PredUs 2.0 and ISPRED4. The cross-validation and calculation of metamethod prediction scores was performed in the same manner described above for meta-DPI. The python script written to execute the logistic regression analysis is included in Appendix C.

1. **Evaluation of Prediction Methods**

Each query protein is composed of *n+* experimentally-determined interfacial residues in the positive class and *n-* experimentally-determined non-interfacial residues in the negative class. Since protein residues belong to one of two categories, interface prediction methods are considered binary classifiers. After a classifier generates prediction scores for a query protein, the query residues are divided into *m+* predicted interfacial residues and *m-* predicted non-interfacial residues. One way to split a protein’s residues is to sort them according to their prediction scores (, and the top *k* number of residues are assigned to the *m+* class and the remaining residues to the *m-* class. Another approach is to assign residues for which to the *m+* class and residues for which to the *m-* class (T refers to the threshold value).

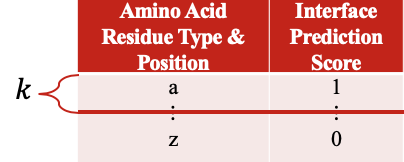
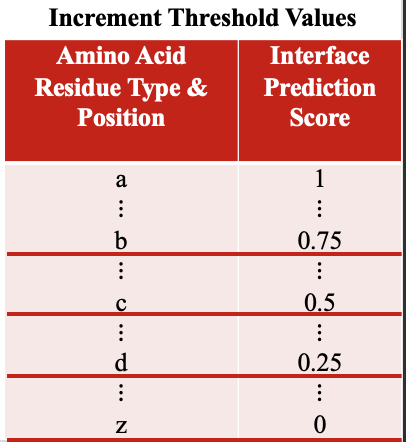
Once a classification method has been applied to the proteins in a dataset, each residue can fall into one of four categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). TP refers to predicted interfacial residues that were experimentally determined to be interfacial , while FP refers to predicted interfacial residues that were experimentally determined to be non-interfacial . TN refers to predicted non-interfacial residues that were experimentally determined to be non-interfacial , while FN refers to predicted non-interfacial residues that were experimentally determined to be interfacial . The four binary classification outcomes can be presented in a 2 x 2 confusion matrix . When creating a binary classification model, the goal is to maximize the number of TP and TN, while minimizing the number of FP and FN. A perfect classification generates the confusion matrix , and a perfect misclassification generates the confusion matrix . [24]

A variety of classifier evaluation metrics can be calculated from confusion matrix values (Table 2). Precision refers to the fraction of a classifier’s *m+* class that also belongs to the *n+* class, and recall refers to the fraction of the *n+* class included in a classifier’s *m+* class. The F1­ ­score is the harmonic mean between precision and recall, so it summarizes a classifier’s performance in generating its *m+* class. The MCC is the only metric that achieves a high score only if a classifier correctly predicted the majority of the *n+* class and the majority of the *n-* class. Since the F1­ ­score is independent of the number of TN and changes when the *n+* and *n-* classes are swapped, it is a less informative than the MCC metric.

**Table 2.** Binary Classifier Evaluation Metrics

|  |  |
| --- | --- |
| Evaluation Metric | Formula |
| Precision |  |
| Recall / True Positive Rate |  |
| False Positive Rate |  |
| F1 Score |  |
| Matthews Correlation Coefficient |  |

For a given query protein, the number of residues *k* to assign to the *m+* class was determined using the dynamic cutoff formula [7] proposed by PredUs 2.0: , where N refers to the number of query protein surface residues (Figure 3A). The values of for the query proteins in our dataset is included in Appendix A. F1 and MCC scores were calculated for each individual prediction method and metamethod for all query proteins in the DB and NOX datasets. Global F1 score and MCC, which reflect a prediction classifier’s performance on all of the proteins in a dataset as a whole, were subsequently calculated. The perl6 scripts written to ascertain the F1 score and MCC are included in Appendices D and E.



1. **Incrementing Threshold Values**

**Fig 3. (A)** The dynamic cutoff *k* indicates the number of residues assigned to the *m+* class for a given query protein. The resulting confusion matrix is used to calculate the F1 score and MCC. **(B)** The Precision, TPR, and TPR values calculated at each threshold value serve as the data points in the PR and ROC plots.

1. **Dynamic Cutoff**

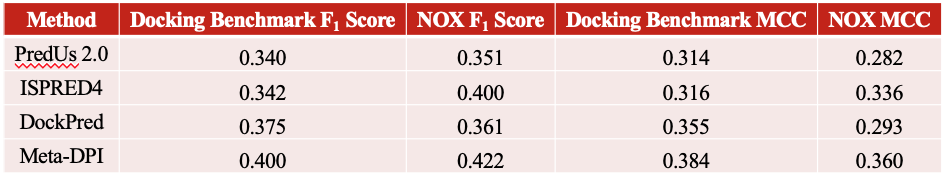
The Receiver Operator Characteristic (ROC) and Precision-Recall (PR) curves give an overview of a classifier’s performance over a range of thresholds. The ROC plot shows pairs of true positive rate (TPR) and false positive rate (FPR) values at all possible thresholds, and the PR plot displays pairs of precision and recall values at all possible thresholds. TPR is equivalent to recall, and FPR refers to the fraction of the *n-* class included in a classifier’s *m+* class. The area under the curve (AUC) is a metric that quantifies the results in ROC and PR curves. A random unskilled classifier will yield a ROC plot of (TPR = FPR), which has a ROC AUC of 0.5. A PR plot of will be generated by a random unskilled classifier, which has a PR AUC of . Generally, a small fraction of a protein’s residues appears at its interface, so and are imbalanced classes (). Figure 4 illustrates the difference between the confusion matrices of balanced and imbalanced classes. Since precision can differentiate between a classifier’s performance on balanced classes versus imbalanced classes, unlike TPR and FPR, PR curves are more informative than ROC curves for a problem like interface prediction that involves imbalanced classes. [1]

**Fig 4.** [1] Both confusion matrices have 20 total elements. The balanced CM has 10 elements in the class and 10 elements in the class. The imbalanced CM has 5 elements in the class and 15 elements in the class.

For each prediction method, threshold values (T) were incremented by 0.01, starting from 0 and ending at 1 (Figure 3B). Residues with prediction scores were assigned to the *m+* class, and confusion matrices were generated accordingly at each T value. TPR, FPR, precision, and recall were calculated at every T value. The TPR and FPR at a single T value served as a data point on the ROC plot, and the precision and recall at a single T value served as a data point on the PR plot. Using this approach, ROC and PR curves were generated to display each prediction method’s performance on the DB and NOX datasets. The AUC for the ROC and PR plots were approximated using the trapezoidal rule. Appendix F contains the perl6 scripts written to increment T values and calculate the evaluation metrics for the ROC and PR curves.

**III. Results & Discussion**

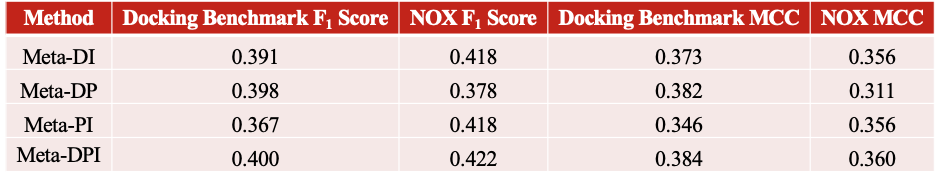
1. **Single Threshold Evaluation Metrics: F1­ Score & MCC**

**Table 3**. F1 Score & MCC of Individual Classifiers and meta-DPI for DB and NOX Databases

In Table 3, the Docking Benchmark F1 score and MCC reflect the performance of each classification method on the proteins in the DB dataset as a whole, while the NOX F1 score and MCC reflect each classification method’s performance on the proteins in the NOX dataset as a whole. The meta-DPI prediction values for DB query protein residues, which were calculated using the coefficients obtained from the logistic regression performed on the NOX dataset, were used to calculate the DB F1 score and MCC for meta-DPI. The NOX F1 score and MCC for meta-DPI were calculated in the opposite manner. As shown in Table 3, meta-DPI outperformed each of its constituent methods according to both evaluation metrics, regardless of the dataset on which it was trained.

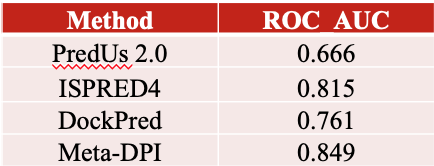
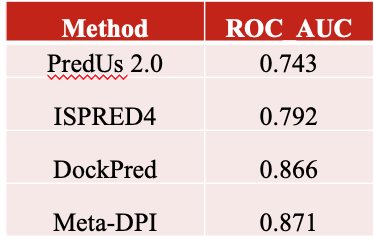
For most query proteins, the number of dynamic cutoff residues was slightly higher than the number of experimental annotated residues (see Appendix A). For the NOX query protein 2PCB.A, which had 25 more cutoff residues (31) than annotated residues (6), each classifier had a lower precision than its average precision and a higher recall than its average recall. On the other hand, for the NOX query protein 1EFV.A, which had 28 fewer cutoff residues (31) than annotated residues (59), each classifier had a higher precision than its average precision and a lower recall than its average recall. Thus, the number of cutoff residues relative to annotated residues directly influences the two components of the F1 score, precision and recall.

When comparing each classifier’s average MCC with their MCC for the aforementioned NOX query proteins, no consistent pattern of outliers emerges. This is because the MCC depends on all four categories of the confusion matrix equally. Since MCC is not influenced by the difference between the number of cutoff and annotated residues, it is a more informative metric than F­1 score, especially for query proteins with an unknown complex structure and unknown number of annotated residues.

**Table 4**. F1 Score & MCC of All Metamethods for DB and NOX Database

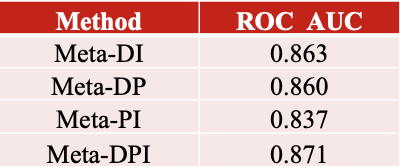
In Table 4, the performance of meta-DPI is compared to the performance of the metamethods that combined two of the three classifiers. Meta-DPI performed better than the other metamethods according to both F1 score and MCC, regardless of the dataset on which the metamethods were trained. Thus, combining all three methods into a metamethod yielded a classifier superior to the ones obtained when only combining two of the methods. By comparing the results in Tables 3 and 4, it can be seen that the performance of the individual classifiers in a metamethod correlates with the metamethod’s performance. The DB F1 score of DockPred (0.375) was 10% higher than the scores of ISPRED4 (0.342) and PredUs 2.0 (0.340). Therefore, the DB F1 score of meta-PI (0.367), which combined the lower scoring ISPRED4 and PredUs 2.0, was significantly lower than the scores of the metamethods that included DockPred, meta-DP (0.398) and meta-DI (0.391).

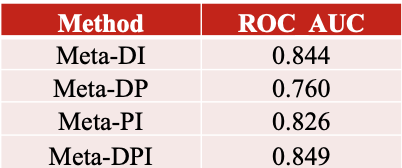
1. **Threshold-free Evaluation Metrics: ROC & PR Curves**



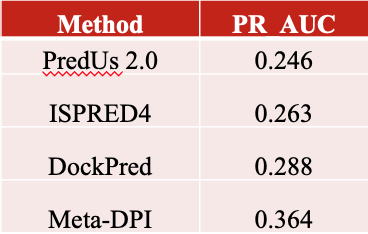
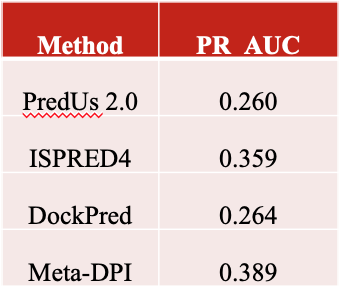
**Fig 5. ROC Graphs for DB & NOX Datasets.** Each classifier’s AUC value is included in the enclosed table. The dashed line represents the ROC plot of a random unskilled classifier (TPR = FPR), which has a ROC AUC of 0.5.

For both the DB and NOX datasets, the ROC AUC of meta-DPI was higher than the ROC AUC of the individual classifiers (Figure 5). While the F1 score and MCC demonstrated meta-DPI’s enhanced ability to classify query protein residues above the dynamic cutoff, the ROC plots confirmed meta-DPI’s superior classification of all query protein residues.

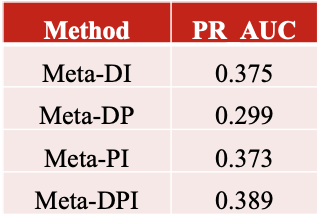
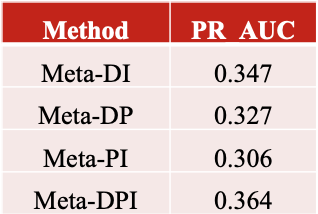




**Fig 6. ROC Graphs for Metamethods for DB & NOX Datasets.** Each metamethod’s AUC value is included in the enclosed table. The dashed line represents the ROC plot of a random unskilled classifier (TPR = FPR), which has a ROC AUC of 0.5.

For both the DB and NOX datasets, the ROC AUC of meta-DPI was higher than the ROC AUC of the metamethods that only combined two classifiers (Figure 6). By comparing the results in Figures 5 and 6, it can be seen that the performance of the individual classifiers in a metamethod correlates with the metamethod’s performance. ISPRED4’s ROC AUC (0.815) for the NOX database was significantly higher than the ROC AUC of DockPred (0.761) and PredUs 2.0 (0.666). Consequently, the ROC AUC of meta-DP (0.760), which combined the poorer performing DockPred and PredUs 2.0, was lower than the ROC AUC of the metamethods that included ISPRED4, meta-DI (0.826) and meta-PI (0.844).

**Fig 7. Precision-Recall Graphs for DB & NOX Datasets.** Each classifier’s AUC value is included in the enclosed table. The dashed line represents the PR plot of a random unskilled classifier (), which is for the DB dataset and for the NOX dataset.

For the DB dataset, the PR AUC of meta-DPI (0.364) was 26% greater than the PR AUC of DockPred (0.288), which was the highest among individual classifiers. As can be seen in the top graph in Figure 7, meta-DPI could correctly classify 0-40% of the residues in the DB dataset, while maintaining a significantly higher degree of precision than the individual classifiers. The results displayed in Figures 7 and 8 further validate meta-DPI’s superior classification ability even by a metric like precision, which is sensitive to class imbalance.

**Fig 8. Precision-Recall Graphs for Metamethods for DB & NOX Datasets.** Each metamethod’s AUC value is included in the enclosed table. The dashed line represents the PR plot of a random unskilled classifier (), which is for the DB dataset and for the NOX dataset.

Among the individual classifiers, PredUs 2.0 consistently performed the worst out of the individual classifiers in terms of global evaluation metrics. For query proteins that had a close structural neighbor with a known complex structure, PredUs 2.0 performed well. However, for query proteins that did not have a close structural neighbor with a known complex structure, PredUs 2.0 performed poorly.

V. References

1. Saito, T. and M. Rehmsmeier, *The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets.* PLoS One, 2015. **10**(3): p. e0118432.

2. Hajduk, P.J., J.R. Huth, and S.W. Fesik, *Druggability indices for protein targets derived from NMR-based screening data.* J Med Chem, 2005. **48**(7): p. 2518-25.

3. Viswanathan, R., et al., *Protein-protein binding supersites.* PLoS Comput Biol, 2019. **15**(1): p. e1006704.

4. Zhu, H., et al., *NOXclass: prediction of protein-protein interaction types.* BMC Bioinformatics, 2006. **7**: p. 27.

5. Vreven, T., et al., *Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2.* J Mol Biol, 2015. **427**(19): p. 3031-41.

6. Sobolev, V., et al., *Automated analysis of interatomic contacts in proteins.* Bioinformatics, 1999. **15**(4): p. 327-32.

7. Hwang, H., D. Petrey, and B. Honig, *A hybrid method for protein-protein interface prediction.* Protein Sci, 2016. **25**(1): p. 159-65.

8. Yang, A.S. and B. Honig, *An integrated approach to the analysis and modeling of protein sequences and structures. I. Protein structural alignment and a quantitative measure for protein structural distance.* J Mol Biol, 2000. **301**(3): p. 665-78.

9. Li, W. and A. Godzik, *Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences.* Bioinformatics, 2006. **22**(13): p. 1658-9.

10. Zhang, Q.C., et al., *PredUs: a web server for predicting protein interfaces using structural neighbors.* Nucleic Acids Res, 2011. **39**(Web Server issue): p. W283-7.

11. Kabsch, W. and C. Sander, *Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features.* Biopolymers, 1983. **22**(12): p. 2577-637.

12. Savojardo, C., et al., *ISPRED4: interaction sites PREDiction in protein structures with a refining grammar model.* Bioinformatics, 2017. **33**(11): p. 1656-1663.

13. Altschul, S.F., et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.* Nucleic Acids Res, 1997. **25**(17): p. 3389-402.

14. Suzek, B.E., et al., *UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches.* Bioinformatics, 2015. **31**(6): p. 926-32.

15. Sander, C. and R. Schneider, *Database of homology-derived protein structures and the structural meaning of sequence alignment.* Proteins, 1991. **9**(1): p. 56-68.

16. Kidera, A., *Statistical analysis of the physical properties of the 20 naturally occurring amino acids* J Protein Chem. , 1985. **4**: p. 23-55.

17. Remmert, M., et al., *HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment.* Nat Methods, 2011. **9**(2): p. 173-5.

18. Jones, D.T., et al., *PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments.* Bioinformatics, 2012. **28**(2): p. 184-90.

19. Mihel, J., et al., *PSAIA - protein structure and interaction analyzer.* BMC Struct Biol, 2008. **8**: p. 21.

20. Adamczak, R., A. Porollo, and J. Meller, *Accurate prediction of solvent accessibility using neural networks-based regression.* Proteins, 2004. **56**(4): p. 753-67.

21. Qin, S. and H.X. Zhou, *meta-PPISP: a meta web server for protein-protein interaction site prediction.* Bioinformatics, 2007. **23**(24): p. 3386-7.

22. Klein, D.G.K.M., *Logistic Regression: A Self-Learning Text*. 2nd ed. Statistics for Biology and Health, ed. K. Dietz. 2002, New York: Springer-Verlag.

23. Brid, R.S. *Logistic Regression* Data Science and Machine Learning 2018 [cited 2020 4/2/2020]; Available from: <https://medium.com/greyatom/logistic-regression-89e496433063>.

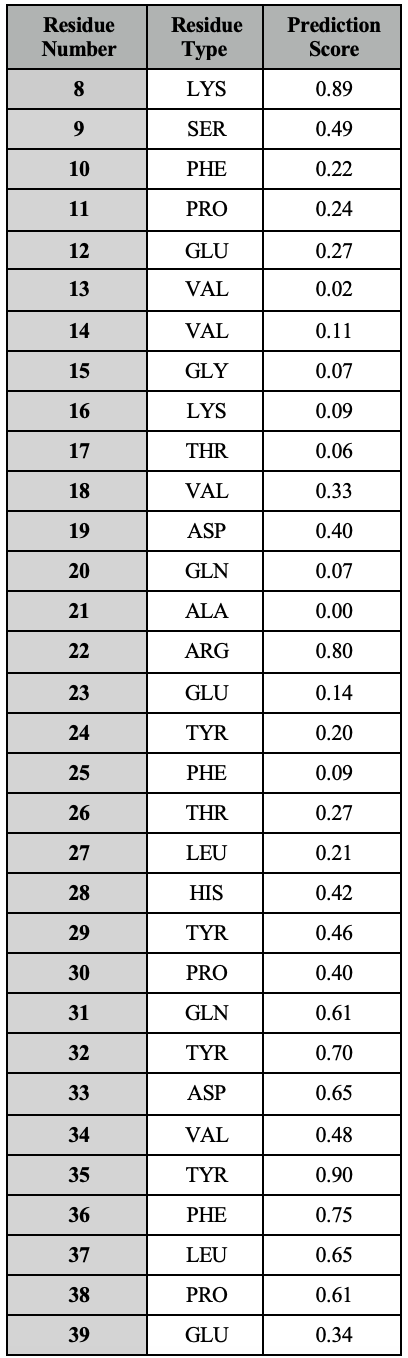
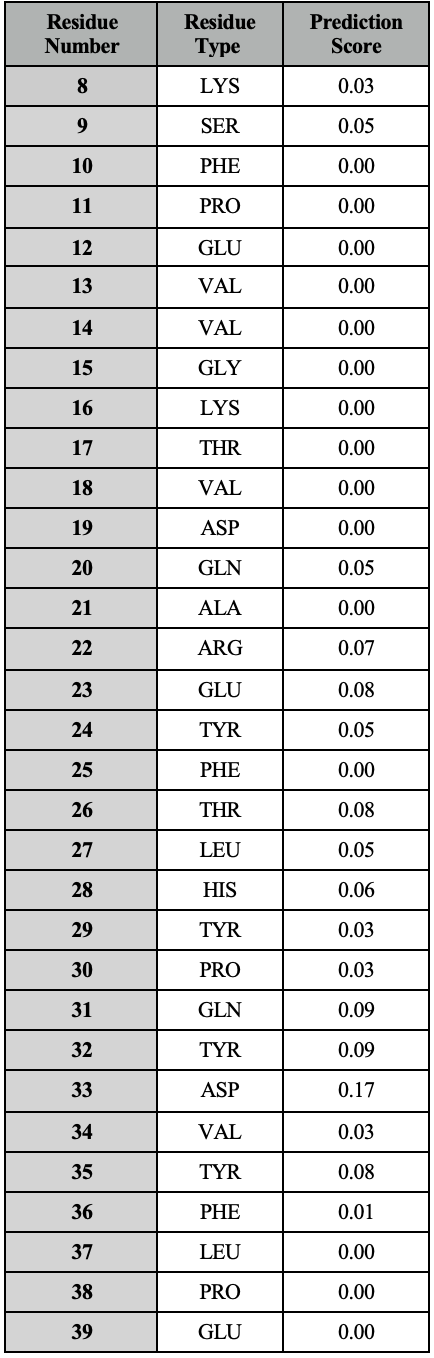
24. Chicco, D. and G. Jurman, *The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation.* BMC Genomics, 2020. **21**(1): p. 6.

**Appendix A: List of Proteins in Docking Benchmark and NOX Datasets**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PDB ID** | **Dataset** | **Surface Residues** | **Dynamic Cutoff Residues** | **Experimental Annotated Residues** |
| **2SIC.E** | Docking Benchmark | 173 | 29 | 17 |
| **2A5T.A** | Docking Benchmark | 192 | 30 | 10 |
| **1DFJ.I** | Docking Benchmark | 325 | 35 | 18 |
| **2BTF.A** | Docking Benchmark | 267 | 33 | 18 |
| **1KLU.D** | Docking Benchmark | 165 | 28 | 11 |
| **4JCV.E** | Docking Benchmark | 177 | 29 | 11 |
| **4G6M.H** | Docking Benchmark | 320 | 34 | 14 |
| **3R9A.B** | Docking Benchmark | 217 | 31 | 17 |
| **2B42.B** | Docking Benchmark | 141 | 27 | 21 |
| **1KAC.A** | Docking Benchmark | 143 | 27 | 9 |
| **1GXD.C** | Docking Benchmark | 167 | 28 | 18 |
| **2FD6.H** | Docking Benchmark | 327 | 35 | 14 |
| **1DE4.E** | Docking Benchmark | 87 | 23 | 21 |
| **3G6D.L** | Docking Benchmark | 329 | 35 | 15 |
| **3SZK.F** | Docking Benchmark | 116 | 25 | 11 |
| **1JIW.I** | Docking Benchmark | 84 | 23 | 15 |
| **1JIW.P** | Docking Benchmark | 332 | 35 | 20 |
| **1ACB.I** | Docking Benchmark | 55 | 20 | 11 |
| **1GHQ.A** | Docking Benchmark | 197 | 30 | 5 |
| **1VFB.A** | Docking Benchmark | 166 | 28 | 15 |
| **3SZK.E** | Docking Benchmark | 107 | 25 | 12 |
| **2B42.A** | Docking Benchmark | 269 | 33 | 19 |
| **1KAC.B** | Docking Benchmark | 101 | 24 | 12 |
| **1ZHH.A** | Docking Benchmark | 240 | 32 | 13 |
| **3EO1.A** | Docking Benchmark | 330 | 35 | 12 |
| **2W9E.H** | Docking Benchmark | 328 | 35 | 17 |
| **1DQJ.A** | Docking Benchmark | 327 | 35 | 18 |
| **3EOA.L** | Docking Benchmark | 322 | 34 | 12 |
| **1FC2.D** | Docking Benchmark | 174 | 29 | 10 |
| **2A5T.B** | Docking Benchmark | 214 | 31 | 14 |
| **2C0L.A** | Docking Benchmark | 214 | 31 | 19 |
| **1TMQ.A** | Docking Benchmark | 306 | 34 | 18 |
| **1US7.A** | Docking Benchmark | 151 | 27 | 7 |
| **4M76.B** | Docking Benchmark | 130 | 26 | 12 |
| **PDB ID** | **Dataset** | **Surface Residues** | **Dynamic Cutoff Residues** | **Experimental Annotated Residues** |
| **2HLE.B** | Docking Benchmark | 107 | 25 | 13 |
| **2B4J.C** | Docking Benchmark | 69 | 22 | 8 |
| **1AVX.A** | Docking Benchmark | 158 | 28 | 19 |
| **2CFH.C** | Docking Benchmark | 118 | 26 | 10 |
| **1XQS.A** | Docking Benchmark | 186 | 29 | 23 |
| **3L5W.L** | Docking Benchmark | 326 | 35 | 10 |
| **1T6B.Y** | Docking Benchmark | 122 | 26 | 14 |
| **1ML0.A** | Docking Benchmark | 278 | 33 | 9 |
| **1OFU.A** | Docking Benchmark | 214 | 31 | 12 |
| **1ZHI.B** | Docking Benchmark | 108 | 25 | 8 |
| **1F34.A** | Docking Benchmark | 235 | 31 | 22 |
| **2AJF.E** | Docking Benchmark | 147 | 27 | 11 |
| **1ZHI.A** | Docking Benchmark | 162 | 28 | 9 |
| **3S9D.B** | Docking Benchmark | 152 | 28 | 12 |
| **1US7.B** | Docking Benchmark | 156 | 28 | 8 |
| **1E6J.H** | Docking Benchmark | 330 | 35 | 7 |
| **1TMQ.B** | Docking Benchmark | 93 | 24 | 15 |
| **2BTF.P** | Docking Benchmark | 106 | 25 | 18 |
| **3BX7.C** | Docking Benchmark | 101 | 24 | 16 |
| **1E4K.C** | Docking Benchmark | 141 | 27 | 12 |
| **2VDB.A** | Docking Benchmark | 499 | 39 | 11 |
| **1KXP.A** | Docking Benchmark | 250 | 32 | 20 |
| **1FLE.E** | Docking Benchmark | 167 | 28 | 11 |
| **1AK4.A** | Docking Benchmark | 119 | 26 | 9 |
| **3MXW.L** | Docking Benchmark | 327 | 35 | 14 |
| **2VXT.H** | Docking Benchmark | 324 | 35 | 13 |
| **4DN4.L** | Docking Benchmark | 310 | 34 | 14 |
| **2VIS.A** | Docking Benchmark | 332 | 35 | 12 |
| **1QA9.A** | Docking Benchmark | 83 | 23 | 13 |
| **1MQ8.B** | Docking Benchmark | 129 | 26 | 12 |
| **3DAW.B** | Docking Benchmark | 108 | 25 | 16 |
| **3HMX.L** | Docking Benchmark | 329 | 35 | 16 |
| **3F1P.B** | Docking Benchmark | 90 | 24 | 18 |
| **3AAA.C** | Docking Benchmark | 93 | 24 | 11 |
| **3F1P.A** | Docking Benchmark | 94 | 24 | 13 |
| **PDB ID** | **Dataset** | **Surface Residues** | **Dynamic Cutoff Residues** | **Experimental Annotated Residues** |
| **3BX7.A** | Docking Benchmark | 138 | 27 | 16 |
| **3DAW.A** | Docking Benchmark | 278 | 33 | 20 |
| **1QA9.B** | Docking Benchmark | 82 | 23 | 15 |
| **1MLC.A** | Docking Benchmark | 327 | 35 | 13 |
| **4G6J.H** | Docking Benchmark | 320 | 34 | 16 |
| **1T6B.X** | Docking Benchmark | 491 | 39 | 12 |
| **1JPS.H** | Docking Benchmark | 324 | 35 | 16 |
| **2J0T.A** | Docking Benchmark | 128 | 26 | 18 |
| **4FQI.H** | Docking Benchmark | 332 | 35 | 11 |
| **1AHW.A** | Docking Benchmark | 325 | 35 | 20 |
| **2HQS.A** | Docking Benchmark | 308 | 34 | 17 |
| **1FFW.A** | Docking Benchmark | 98 | 24 | 8 |
| **3HI6.X** | Docking Benchmark | 324 | 35 | 15 |
| **7CEI.B** | Docking Benchmark | 107 | 25 | 10 |
| **1GXD.A** | Docking Benchmark | 504 | 39 | 21 |
| **3V6Z.A** | Docking Benchmark | 329 | 35 | 21 |
| **1ACB.E** | Docking Benchmark | 172 | 29 | 13 |
| **1AY7.A** | Docking Benchmark | 85 | 23 | 9 |
| **1BGX.H** | Docking Benchmark | 332 | 35 | 31 |
| **1KXP.D** | Docking Benchmark | 361 | 36 | 26 |
| **3VLB.A** | Docking Benchmark | 286 | 33 | 15 |
| **1KTZ.A** | Docking Benchmark | 80 | 23 | 7 |
| **3BIW.A** | Docking Benchmark | 352 | 35 | 9 |
| **1EXB.E** | Docking Benchmark | 76 | 22 | 3 |
| **2O3B.A** | Docking Benchmark | 170 | 28 | 15 |
| **2SIC.I** | Docking Benchmark | 94 | 24 | 11 |
| **3VLB.B** | Docking Benchmark | 157 | 28 | 17 |
| **1DFJ.E** | Docking Benchmark | 103 | 25 | 18 |
| **1IRA.X** | Docking Benchmark | 113 | 25 | 23 |
| **1AY7.B** | Docking Benchmark | 67 | 22 | 9 |
| **2HRK.B** | Docking Benchmark | 102 | 24 | 8 |
| **2J0T.D** | Docking Benchmark | 105 | 25 | 11 |
| **7CEI.A** | Docking Benchmark | 73 | 22 | 13 |
| **1FFW.B** | Docking Benchmark | 59 | 21 | 10 |
| **1PVH.B** | Docking Benchmark | 139 | 27 | 8 |
| **PDB ID** | **Dataset** | **Surface Residues** | **Dynamic Cutoff Residues** | **Experimental Annotated Residues** |
| **3RVW.C** | Docking Benchmark | 321 | 34 | 13 |
| **1WEJ.H** | Docking Benchmark | 327 | 35 | 11 |
| **1S1Q.A** | Docking Benchmark | 115 | 25 | 11 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PDB ID** | **Dataset** | **Surface Residues** | **Dynamic Cutoff Residues** | **Experimental Annotated Residues** |
| **1DOR.A** | NOX | 213 | 30 | 25 |
| **1YVE.I** | NOX | 331 | 35 | 25 |
| **1DOW.A** | NOX | 179 | 29 | 14 |
| **1NSE.A** | NOX | 313 | 34 | 39 |
| **1EMV.A** | NOX | 73 | 22 | 13 |
| **1CP2.A** | NOX | 179 | 29 | 13 |
| **1RRP.A** | NOX | 162 | 28 | 32 |
| **1XIK.A** | NOX | 239 | 32 | 37 |
| **1BKD.R** | NOX | 127 | 26 | 25 |
| **1TCO.A** | NOX | 232 | 31 | 21 |
| **1JKM.A** | NOX | 244 | 32 | 16 |
| **1I8L.A** | NOX | 164 | 28 | 8 |
| **1QBI.A** | NOX | 298 | 34 | 17 |
| **1QFH.A** | NOX | 180 | 29 | 37 |
| **1D09.A** | NOX | 206 | 30 | 15 |
| **1B6C.A** | NOX | 88 | 23 | 15 |
| **1CMX.A** | NOX | 164 | 28 | 22 |
| **1ETH.A** | NOX | 302 | 34 | 14 |
| **3C98.A** | NOX | 400 | 37 | 27 |
| **1BVN.T** | NOX | 61 | 21 | 15 |
| **1REQ.A** | NOX | 495 | 39 | 64 |
| **1STF.E** | NOX | 148 | 27 | 12 |
| **4SGB.I** | NOX | 48 | 19 | 6 |
| **1ONE.A** | NOX | 266 | 33 | 34 |
| **2NAC.A** | NOX | 275 | 33 | 56 |
| **1BJN.A** | NOX | 253 | 32 | 32 |
| **1CNZ.A** | NOX | 253 | 32 | 43 |
| **1SMP.I** | NOX | 81 | 23 | 12 |
| **PDB ID** | **Dataset** | **Surface Residues** | **Dynamic Cutoff Residues** | **Experimental Annotated Residues** |
| **1B3A.A** | NOX | 58 | 21 | 12 |
| **1B34.A** | NOX | 70 | 22 | 17 |
| **1AT3.A** | NOX | 159 | 28 | 15 |
| **1ISA.A** | NOX | 144 | 27 | 9 |
| **1MSP.A** | NOX | 107 | 25 | 12 |
| **1CMB.A** | NOX | 94 | 24 | 15 |
| **1I2M.A** | NOX | 126 | 26 | 24 |
| **1TRK.A** | NOX | 429 | 38 | 67 |
| **3HHR.A** | NOX | 139 | 27 | 21 |
| **1SPU.A** | NOX | 514 | 40 | 112 |
| **1LFD.A** | NOX | 74 | 22 | 11 |
| **1VOK.A** | NOX | 149 | 27 | 19 |
| **1AVW.A** | NOX | 159 | 28 | 19 |
| **1BRM.A** | NOX | 261 | 32 | 54 |
| **1UEA.A** | NOX | 133 | 26 | 22 |
| **1CSE.I** | NOX | 55 | 20 | 10 |
| **1ITB.A** | NOX | 119 | 26 | 25 |
| **1SMT.A** | NOX | 90 | 24 | 22 |
| **1EG9.A** | NOX | 293 | 34 | 31 |
| **1CVS.A** | NOX | 100 | 24 | 17 |
| **1FRV.A** | NOX | 198 | 30 | 59 |
| **1BI7.A** | NOX | 202 | 30 | 25 |
| **1TX4.A** | NOX | 145 | 27 | 17 |
| **1B5E.A** | NOX | 187 | 29 | 36 |
| **1F6Y.A** | NOX | 183 | 29 | 19 |
| **4MDH.A** | NOX | 239 | 32 | 26 |
| **1HGX.A** | NOX | 125 | 26 | 18 |
| **2AE2.A** | NOX | 196 | 30 | 18 |
| **1C0F.S** | NOX | 101 | 24 | 22 |
| **1HLU.A** | NOX | 277 | 33 | 11 |
| **1DHK.A** | NOX | 321 | 34 | 26 |
| **1WGJ.A** | NOX | 207 | 30 | 11 |
| **4XXH.A** | NOX | 187 | 29 | 19 |
| **2AAI.A** | NOX | 199 | 30 | 25 |
| **2PFL.A** | NOX | 444 | 38 | 23 |
| **PDB ID** | **Dataset** | **Surface Residues** | **Dynamic Cutoff Residues** | **Experimental Annotated Residues** |
| **1QAE.A** | NOX | 164 | 28 | 13 |
| **1WQ1.R** | NOX | 120 | 26 | 14 |
| **1BML.A** | NOX | 171 | 29 | 26 |
| **1GUX.A** | NOX | 137 | 27 | 13 |
| **1EUV.A** | NOX | 158 | 28 | 27 |
| **1GPE.A** | NOX | 351 | 35 | 28 |
| **1CC0.A** | NOX | 130 | 26 | 9 |
| **1PDK.A** | NOX | 170 | 28 | 28 |
| **1DCE.A** | NOX | 433 | 38 | 39 |
| **2PTC.I** | NOX | 53 | 20 | 8 |
| **1TGS.I** | NOX | 49 | 20 | 13 |
| **1QFE.A** | NOX | 170 | 28 | 7 |
| **1JTD.A** | NOX | 177 | 29 | 12 |
| **1B9M.A** | NOX | 217 | 31 | 41 |
| **2HHM.A** | NOX | 193 | 30 | 21 |
| **1VLT.A** | NOX | 120 | 26 | 14 |
| **1CLI.A** | NOX | 246 | 32 | 42 |
| **1YCS.A** | NOX | 153 | 28 | 11 |
| **1SOX.A** | NOX | 333 | 35 | 24 |
| **1GLA.F** | NOX | 121 | 26 | 7 |
| **1EFV.A** | NOX | 228 | 31 | 59 |
| **1FIN.A** | NOX | 223 | 31 | 29 |
| **1PP2.L** | NOX | 105 | 25 | 17 |
| **1BYF.A** | NOX | 94 | 24 | 16 |
| **1B8J.A** | NOX | 289 | 33 | 60 |
| **1BUH.A** | NOX | 221 | 31 | 12 |
| **1COZ.A** | NOX | 102 | 24 | 10 |
| **1HJR.A** | NOX | 126 | 26 | 15 |
| **1PNK.A** | NOX | 172 | 29 | 80 |
| **1QAX.A** | NOX | 327 | 35 | 78 |
| **1QOR.A** | NOX | 235 | 31 | 21 |
| **1FSS.A** | NOX | 341 | 35 | 15 |
| **1AVZ.B** | NOX | 89 | 23 | 11 |
| **1LUC.A** | NOX | 239 | 32 | 30 |
| **1AVA.A** | NOX | 259 | 32 | 17 |
| **PDB ID** | **Dataset** | **Surface Residues** | **Dynamic Cutoff Residues** | **Experimental Annotated Residues** |
| **1H2A.L** | NOX | 349 | 35 | 55 |
| **1CQI.A** | NOX | 201 | 30 | 18 |
| **2HDH.A** | NOX | 226 | 31 | 16 |
| **1YPI.A** | NOX | 179 | 29 | 23 |
| **2UTG.A** | NOX | 66 | 21 | 14 |
| **3TMK.A** | NOX | 167 | 28 | 11 |
| **1HSS.A** | NOX | 92 | 24 | 15 |
| **1ZBD.A** | NOX | 131 | 26 | 19 |
| **1BO1.A** | NOX | 252 | 32 | 18 |
| **1XSO.A** | NOX | 107 | 25 | 7 |
| **1B8A.A** | NOX | 333 | 35 | 56 |
| **1B7B.A** | NOX | 230 | 31 | 28 |
| **1KPE.A** | NOX | 93 | 24 | 33 |
| **1QAV.A** | NOX | 76 | 22 | 13 |
| **1EAI.C** | NOX | 60 | 21 | 10 |
| **1F60.A** | NOX | 319 | 34 | 28 |
| **2PCB.A** | NOX | 224 | 31 | 6 |
| **1ATN.A** | NOX | 270 | 33 | 16 |

**Appendix B**



|  |  |  |
| --- | --- | --- |
| **Residue Number** | **Residue Type** | **Prediction Score** |
| **8** | LYS | 0.31 |
| **9** | SER | 0.06 |
| **10** | PHE | 0.00 |
| **11** | PRO | 0.02 |
| **12** | GLU | 0.02 |
| **13** | VAL | 0.00 |
| **14** | VAL | 0.06 |
| **15** | GLY | 0.01 |
| **16** | LYS | 0.01 |
| **17** | THR | 0.01 |
| **18** | VAL | 0.00 |
| **19** | ASP | 0.02 |
| **20** | GLN | 0.01 |
| **21** | ALA | 0.00 |
| **22** | ARG | 0.01 |
| **23** | GLU | 0.01 |
| **24** | TYR | 0.01 |
| **25** | PHE | 0.00 |
| **26** | THR | 0.02 |
| **27** | LEU | 0.31 |
| **28** | HIS | 0.17 |
| **29** | TYR | 0.03 |
| **30** | PRO | 0.02 |
| **31** | GLN | 0.02 |
| **32** | TYR | 0.00 |
| **33** | ASP | 0.05 |
| **34** | VAL | 0.00 |
| **35** | TYR | 0.02 |
| **36** | PHE | 0.00 |
| **37** | LEU | 0.04 |
| **38** | PRO | 0.03 |
| **39** | GLU | 0.02 |
| **40** | GLY | 0.05 |
| **41** | SER | 0.35 |
| **42** | PRO | 0.96 |
| **43** | VAL | 0.25 |
| **44** | THR | 0.96 |
| **45** | LEU | 0.67 |
| **46** | ASP | 0.98 |
| **47** | LEU | 0.97 |
| **48** | ARG | 0.17 |
| **49** | TYR | 0.24 |
| **50** | ASN | 0.12 |
| **51** | ARG | 0.00 |
| **52** | VAL | 0.00 |
| **53** | ARG | 0.1 |
| **54** | VAL | 0.00 |
| **55** | PHE | 0.26 |
| **56** | TYR | 0.05 |
| **57** | ASN | 0.04 |
| **58** | PRO | 0.01 |
| **59** | GLY | 0.01 |
| **60** | THR | 0.01 |
| **61** | ASN | 0.01 |
| **62** | VAL | 0.02 |
| **63** | VAL | 0.00 |
| **64** | ASN | 0.01 |
| **65** | HIS | 0.01 |
| **66** | VAL | 0.02 |
| **67** | PRO | 0.00 |
| **68** | HIS | 0.01 |
| **69** | VAL | 0.00 |
| **70** | GLY | 0.01 |

