ISPIP: Improved Prediction of Epitope Binding Sites

Thesis Submitted in Partial Fulfillment
of the Requirements
of the Jay and Jeanie Schottenstein Honors Program

Yeshiva College
Yeshiva University
February 2023

Abraham I. Bodzin
Mentor: Professor Rajalakshmi Viswanathan, Chemistry

Table of Contents

Introduction **1**

Methods **4**

ISPRED4 4

DockPred 7

SPPIDER 8

ISPIP 10

Dataset 10

Cross Validation 11

Training ISPIP 12

Linear Regression 12

Logistic Regression 13

Random Forest 14

Xgboost 14

Benchmarking 15

Performance Metrics 16

Results **19**

Conclusion **24**

Further Work **25**

References **27**

Introduction

Proteins are vital parts of any organism's functioning, involved in locomotion, cellular reproduction, hormone signaling, nutrient absorption, macromolecular synthesis, immune responses, and just about every other task performed. Proteins are made of linear chains of amino acids characteristic to the type of protein. The sequence of amino acids in a protein is known as its primary structure, while the hydrogen and other weak bonds between amino acids give rise to a protein's secondary structure, describing certain common and well-defined structures such as α-helices and β-sheets. A protein's tertiary structure describes the three dimensional shape of the entire protein. Finally, the quaternary structure of a protein describes the relationships between subunits of a protein – units whose shapes would be conserved even if it were removed from the rest of the protein. Though there are only 20 amino acids commonly used in proteins, their variations in characteristics – including size, polarity, hydrophobicity, acidity, aromaticity, and ability to form hydrogen bonds – allow for the extreme diversity seen in the proteome. These varying characteristics also allow for highly-specific interactions *(1)*.

Proteins are able to interact with other molecules via binding interfaces at their surfaces. These interactions can even be with other proteins, a protein-protein interaction (PPI). A protein's binding interfaces are highly specific to their ligand as a result of their chemical makeup and the resulting shape dictated by the weak bonds between amino acids. When a protein binds with its ligand its conformation changes to perform its intended action. This high specificity is seen, for example, in an adaptive immune response, during which the body makes many different antibodies in response to an antigen until one is shown to have a very high

affinity to the antigen. The body then increases production of that specific antibody to attack the antigen in the body. The resulting antibody is specific to that antigen, and is not useful against any other antigen the body may come in contact with *(1, 2)*.

The ability to predict an antigen's epitope(s), those area(s) another protein may bind to, based solely on its structure is of medical interest as it could provide information about the antigen's function and mechanism *(3)*. It would also allow the engineering and production of proteins designed to interact with a target antigen to promote or inhibit the target's activity. This would allow for the engineering of artificial antibodies as soon as an antigen could be identified and sequenced at low cost and minimum effort.

In the wet lab, finding epitopes has traditionally been done by crystallizing bound antibody-antigen structures and using methods such as X-ray crystallography and NMR to determine interacting residues, which has proven to be slow, inefficient, and costly. X-ray crystallography requires high-quality crystals which can be exceedingly difficult to obtain for certain proteins *(4)*. NMR is costly and difficult on molecules as large as antibodies *(5)*. In light of these, cryo-electron microscopy has become the dominant method for epitope discovery, but remains slow *(4)*. In order to complement these methods and decrease the barriers to epitope prediction, researchers have turned to computational methods of epitope prediction. These methods can broadly be divided into two categories – structure based methods, and template based methods. Structure based methods use data about the target's structure, including data about the characteristics of the target's amino acid composition to predict epitopes. Template based methods use the query protein's structure as input and search through a precompiled database of proteins for similar proteins whose complex structure is known. The query protein is then superimposed over the protein of known complex structure to identify the query protein's

interfacial residues. This relies on the earlier described principle that a protein's structure yields its shape and function; proteins with similar structures and shapes are likely to bind similarly *(3)*.

Although each of these strategies show promise, they each have limiting factors that have prevented them from being accurate enough for real-world use. Structure based methods are limited in that combining more features in predictions has demonstrated little effect on performance. Template based methods work well for proteins that have analogues whose complex structures are known, but are hampered by the relatively small number of proteins whose structures have been resolved in wet labs. In order to improve further in the realm of epitope prediction, work has been done to combine different methods into meta-methods that can take advantage of the strengths of each of their component methods *(6)*.

ISPIP is a meta-method designed to improve on previous classifiers by choosing components that use different strategies and using machine learning algorithms to train the model. It is based on Walder's Meta-DPI, but replacing PredUs 2.0 with SPPIDER, so that the three classifiers included are ISPRED4, SPPIDER, and DockPred. Another significant change is in our strategy for combining results of different technologies. Walder used a logistic regression, taking into account that any residue can only have one of two possible states – interface or not interface – which was an improvement over some previous meta-methods that used linear regression *(7)*. We tested several different algorithms including linear and logistic regression models, and machine learning models including random forest and xgboost, to find the best way to combine each of the classifier's predictions.

The development of ISPIP has been the work of a team led by Dr. Viswanathan and including Moshe Carrol, and Alexandra Roffe. My personal contribution was largely in comparing the results of ISPIP to another predictor, DiscoTope 2.0.

Methods

ISPRED4

ISPRED4 is a structure based PPI predictor that uses 46 different protein features in 11 groups, shown in Table 1, and a combination of machine learning techniques.

Table 1: Types of Residue Features Used in ISPRED4

Feature Group	Number of Features
Sequence Profile	20
Conservation Score	1
Interface Propensity	1
Residue Properties	10
Mutual Information	2
PSICOV	2
Depth Indexes	3
Protrusion Indexes	4
Secondary Structures	3
Average B-factor	1
RSA Difference	1

ISPRED4's dataset were proteins from among the Docking Benchmark v5 (DBv5) dataset whose bound and unbound structures had both been obtained by x-ray crystallography and whose interface residues could unambiguously be mapped from the bound to unbound forms. This

resulted in a dataset, named DBv5Sel, of 151 protein complexes from DBv5's original 230. A support vector machine (SVM) and a grammar-restrained hidden conditional random field (GRHCRF) were trained on the dataset to produce ISPRED4 *(8)*.

SVMs are a classification method that, in their simplest form, use a linear equation to classify items into one of two categories based on two characteristics. First, members of a set whose category are known are plotted on a cartesian plane. A linear equation that best splits the data into two separate groups is then calculated, and any future points of unknown category are predicted based on which side of the line they fall *(8)*. An example is shown in Figure 1 *(9)*.

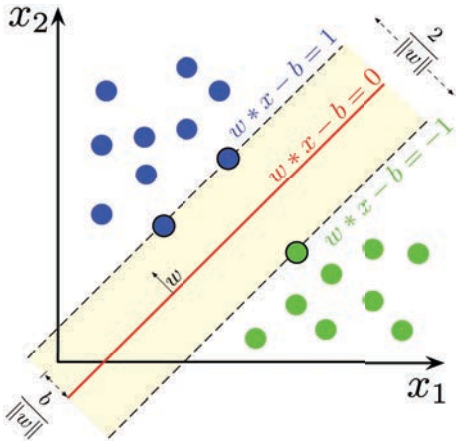

Figure 1: SVM Classification

(Larhmam, CC BY-SA 4.0, via Wikimedia Commons)

In more complex forms, SVMs can use non-linear equations and take into account many characteristics. In ISPRED4, each surface residue in DBv5Sel was represented as a 46-dimensional vector, and was labeled as interface if it's solvent-accessible surface area (SASA or ASA), as calculated by the DSSP program, decreased by 1 $Å^2$ or more between the unbound monomer and bound complex. Surface residues were defined as those with a relative solvent accessibility (RSA) – the ratio between a residue's ASA and its theoretical maximum ASA in a

tripeptide of Gly-X-Gly – greater than or equal to 20% *(10, 11)*. Although the resulting SVM could make a prediction about any protein residue (whose features are known), a GRHCRF was used to further improve results.

SVMs are a very powerful tool, but inherently treat each query item as independent from one another. In proteins, residues' physical attachment to each other contradicts this assumption. For example, an epitope site is unlikely to have interface residues in a ring with a hole in the middle. Because SVMs cannot understand context though, they might predict such an epitope. In contrast, a conditional random field (CRF) is a classification technique that does take into account the classification of nearby elements. This is done by summing weighted feature functions that each label an element based on properties such as the position of the element in the set and the label of the previous element. Grammatically-restrained hidden conditional random fields (GRHCRF) build on CRFs by allowing a pre-defined grammar to be put in place, restricting possible predictions to those that follow the grammar regardless of the sum of the feature functions. GRHCRFs allow known patterns and rules to be forced onto a CRF that might otherwise find solutions outside of those patterns. Combining a GRHCRF with its SVM allows ISPRED4 to avoid some of the weaknesses of an SVM only approach *(12)*.

ISPRED4 was tested on 22 bound structures that each had less than 30% sequence identity to any structures in the training set or other structures in the test set, sourced from CAPRI experiments *(10, 13)*.

DockPred

DockPred is a PPI predictor based on the earlier discovery that protein superfolds – families of proteins with the same overall structure despite having differing functions – have

similar binding "supersites", similar binding sites across the superfold, despite significant differences in primary structure. This led to the observation that protein binding sites are generic, and can be predicted by the frequency of receptor residues interacting with other ligands, regardless of their similarity to the target ligand *(14)*.

Between the NOX and Docking Benchmark databases, 241 proteins were chosen as the test set. 13 ligand probes, sharing no sequence similarity to known ligands of the query proteins, were then computationally docked to the query proteins. ZDOCK and GRAMM generated 2000 docked complexes of each query protein with each of the 13 probes *(15, 16)*. Interface residues, determined by CSU, were defined as any residue with one atom within 3.5 Å of any atom of the probe, and which establishes legitimate atomic contact as defined by CSU *(17)*. Each residue, R_{ik}, where i is the position number of the residue and k is the k^{th} docked complex structure, had an interface value, $I(R_{ik})$, of 1 if it was determined interface; 0 otherwise. By summing over all 2000 docked structures of each query protein, a residue interface score (RIF), N_i, was determined for every residue. The top 15 residues with the highest N_i in each protein were considered interface *(14)*.

SPPIDER

SPPIDER works by taking advantage of a discrepancy between real and predicted values of surface exposure. RSA predictions of amino acids were found to correlate with surface exposure in protein complexes, but not with surface exposure in the unbound structures of individual protein chains. Therefore, large differences in the observed and predicted surface exposure in the unbound chain (called dSA) signal the location of an interface residue. 435 non-redundant proteins that each had at least two chains, none less than 30 amino acids long;

contained no DNA or RNA portions; and had at least one chain with less than 50% sequence identity to any other chain, were used to train a neural network with 19 feature inputs derived from a sliding window approach. Eleven of the inputs are dSA, calculated for each residue in an 11 residue window centered on the residue of interest. Eight additional features that were averages computed across the window, seen in Table 2, were also inputs *(18)*. These 19 features were used to train a neural network to predict PPI sites.

Table 2: Features Averaged Across a Sliding Window

Feature	Average Used	Definitions
dSA	$P_0 + \sum_{i=1}^{N} \frac{P_i}{d_i}$	P is the value of the property, d is the distance to the i-th residue, and N is the total number of neighbors in a 15 Å residue sphere.
Predicted RSA		
Conservation of Charge		
Conservation of Hydrophobicity		
Conservation of Size		
Conservation of Amino Acid Type		
Contact Number	$\sum_{i=0}^{N} P_i RSA_i$	
Hydrophobicity		

A neural network consists of a number of "layers" of "nodes". Each node in a layer receives inputs – either 0 or 1 – from all of the nodes in the previous layer, and sends an output value – again, 0 or 1 – to all of the nodes in the next layer. Every connection between two nodes has a weight and each node has a threshold value. If the sum of the weighted inputs to a node is greater than or equal to the node's threshold, that node's output is 1. Otherwise the node's output is 0. The weights and thresholds are assigned randomly to begin with, and are slowly changed

during the training process to decrease error. Given enough layers and fine-tuning of weights, neural networks have shown to be incredibly capable at classification problems *(19)*.

ISPIP

ISPIP is the meta-method that we have designed, combining SPPIDER, DockPred, and ISPRED4. Several different methods were tested to combine the results of these methods into those of ISPIP.

Dataset

Jesperson et al. identified a set of 335 bound antibody-antigen complexes with annotated interface residues (known interface and non-interface residues). Of these, those antigens with epitopes on more than one chain were removed; 275 antigen-antibody pairs remained. From the 275, 195 structural analogues with >95% sequence similarity were found from the PDB, and the interfacial residues from the bound structures were mapped to the unbound antigen analogues *(20)*. We further limited the dataset to those with <30% sequence identity with any other protein in the set. This resulted in a boud dataset of 107 bound proteins and an unbound dataset of 76 unbound proteins for the training and testing of ISPIP.

Training ISPIP

Linear Regression

Linear regression is a model of the relationship between one or more independent variables and one dependent variable, modeled with a number of linear equations. Several methods exist to fit the regression line to the data, each of whose goal is to minimize the distance

between the regression line and the data points. Some set of labeled data is required to fit the line initially, after which the line can be used to predict the value of the independent variable given known dependent variable values. Figure 2 shows a simple linear regression, with only one independent variable, on the horizontal axis, and the dependent variable on the vertical axis *(22)*.

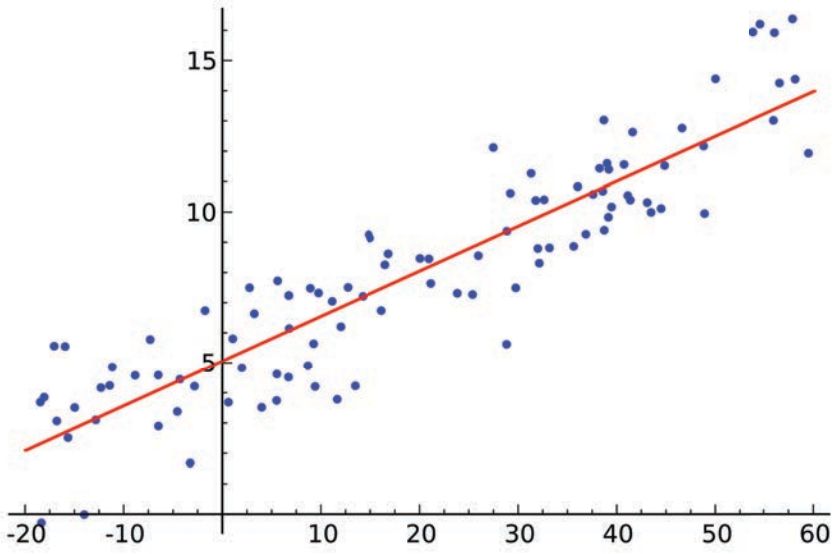

Figure 2: Example of Simple Linear Regression
Sewaqu, Public domain, via Wikimedia Commons

The general formula for multiple linear regression, with more than one dependent variable, is

$$Y_i = \beta_0 + \sum_{k=1}^{p} \beta_k X_{ik} + \epsilon_i$$

where Y_i is the *i*-th observation of the dependent variable, X_{ik} is the *i*-th observation of the *k*-th independent variable, of which there are a total of *p*, and ε_i represents the error in the measurement of X_i *(23)*.

Logistic Regression

In logistic regression, the probability of an event is modeled on a sigmoid function which stays close to 0 until rapidly increasing to close to 1. This makes it very useful for binary

classification purposes such as whether a residue is binding or not. In order to create a model, a likelihood function is maximized on a set of labeled data. In ISPIP, each residue was used as a data point and the independent variables were the predictions of each of ISPIP's constituent methods *(24)*.

Random Forest

Random forest is a machine learning technique that combines many decision trees to make a final decision. A decision tree consists of a number of decision nodes that can lead to either another decision node or to a leaf node which provides a classification. At each decision node, a data point's feature values are compared to the condition in the decision node and the comparison is used to determine which branch the data will continue down. In random forest, a large number of subsets of the labeled data are created, such that any datum may be repeated in a single subset (i.e. subsets are created with replacement). This allows the subsets to be more different from each other than sampling without replacement would allow, which makes the final predictor more robust. All of the subsets are then assigned the same number of randomly chosen features of the data, and decision trees are made for each of the subsets, with each using only its assigned features to classify. Choosing random features ensures that no individual feature with strong predicting power dominates all of the trees. New data being predicted is then run through all of the trees in the forest and is classified as whichever classification the majority of the trees predict *(25)*. In ISPIP each residue is a datum whose features are each of the constituent predictor's scores.

Xgboost

Xgboost is a machine learning algorithm that also uses many decision trees to make its predictions, but differs from random forest in that its trees are not made from random samples. In xgboost a number of shallow decision trees (decision trees with few nodes) are combined, where each tree is designed based on the error residuals of the previous tree. This has the effect of slowly decreasing the residuals with each additional tree. Xgboost includes several measures to avoid overfitting a model to its training set, including scaling the results of each tree down with a learning factor, and including a regularization parameter which makes a leaf more susceptible to pruning, pruning itself being a way to decrease overfitting by cutting a decision tree short *(26)*.

Cross Validation

In order to be able to score a model's performance it must be tested on data that is annotated, but that annotated data should not be part of the training set, so as to avoid issues of overfitting. To satisfy both of these requirements, cross validation (CV) is used, wherein a dataset is split into n groups of equal size, consisting of k=n-1 training sets and 1 test set. The model is then trained on each combination of k-1 training sets and tested on the one remaining, resulting in k different trainings. Performance metrics are then based on the average of the parameters found this way used on the test set. ISPIP had similar results using both 5 and 3-fold CV, so to increase the size of the test set 3-fold CV was kept.

Given enough data, randomly assigning groups can work, but with small datasets randomly assigned groups can result in one group having a largely disproportionate number of similar data, which can lead to poor performance when that set is tested on, and falsely deflated performance figures. In order to avoid this, we curated both bound and unbound datasets to have

similar numbers of proteins of similar sizes, and similar numbers of proteins from certain high frequency CATH classifications. CATH is a hierarchical protein database that groups proteins based on both structural features (class, architecture, and topology) and evolutionary relationships (homologous structures) *(21)*. Any topology represented by more than 3 proteins in the bound set were manually distributed through the 4 sets as equally as possible; there were three such CATH classifications, seen in Table 3. All other proteins were randomly distributed.

Table 3: Highly Represented CATH Classifications

CATH Classification	Classification Name	Proteins in Bound Set
1.20.1250	Growth Hormone; Chain: A;	7
2.60.40	Immunoglobulin-like	21
3.40.50	Rossmann fold	7

Benchmarking

In order to benchmark ISPIP's results, in addition to comparing it to its constituent classifiers it was also compared to the unrelated classifier DiscoTope 2.0. DiscoTope 2.0 is a classifier designed specifically for the prediction of antigen epitopes. In short, it works by combining a propensity score that describes the likelihood of any residue being interface given

its local environment and a surface score that makes residues on the surface of an antigen more likely to be predicted as interfacial. A DiscoTope score (DS) is calculated as follows:

$$DS(r, \alpha) = -\ \alpha \cdot SS(r) + (1 - \alpha) \cdot PS(r)$$

where r is the query residue, α is a constant between 0 and 1 found by grid search, SS is the upper half-sphere (UHS) surface score, and PS is the log-odds ratio propensity score. The UHS is calculated by creating a sphere around the query residue's C_α of radius k_{sur} found through grid search, drawing a plane through that C_α perpendicular to the C_β, and counting the number of C_αs of other residues in the half of the sphere containing C_β *(27)*. The PS is defined as follows:

$$PS(r, w, k_{ps}) = \sum_i ((0.8 \cdot (1 - \frac{d_i}{k_{ps}}) + 0.2) \cdot ls(r_i, w))$$

where r is the query residue, r_i is any residue within a distance k_{ps} of r, d_i is the distance between r and r_i, and $ls(r_i,w)$ is the log-odds ratio of r_i sequentially averaged over w residues. Both w and k_{ps} were found by grid searches. The log-odds ratio of any given amino acid was calculated by sliding a 9-residue window over the primary sequence of each protein, assigning each frame to an epitope or non-epitope group based on the annotation of the center residue, and calculating a positional weight matrix for both groups. Finally, a log-odds ratio for each type of amino acid was computed by taking the log of the ratio between the epitope weight matrix value at position 5 for that amino acid to the non-epitope weight matrix value for the same amino acid at position 5. Summing over several nearby residues and giving more weight to residues closer to r, as well as including the surface score, allows DiscoTope to take into account the local environment of a residue rather than using only information about the residue itself *(28)*.

Performance Metrics

In order to evaluate classification techniques, a confusion matrix like that seen in Table 4 is built that shows the number of items predicted positive/negative as they intersect with those that are truly positive/negative, producing true/false positives and negatives. The aim of any classifier is to maximize the number of true positives and true negatives while minimizing the number of false positives or negatives.

Table 4: 2x2 Confusion Matrix

	Predicted Positive (PP)	Predicted Negative (PN)
Positive (P)	True Positive (TP)	False Negative (FN)
Negative (N)	False Positive (FP)	True Negative (TN)

In the case of antibodies and antigens, experimentally determined interface residues are marked positive, while experimentally determined non-interface residues are marked negative. Because ISPIP and each of its constituent classifiers return a value between 0 and 1 for each residue rather than predicted positive or predicted negative, some cutoff for predicted interface must be decided. In order to divide scores into these two classes a method's output is ranked in descending order by score. A static method could then be applied, taking the top k residues and labeling them predicted positive. While some most accurate k may exist, for ISPIP a dynamic cutoff was used as introduced by PredUS 2.0, $k = 6.1N^{0.3}$, where N is the number of amino acids at the surface of the protein, which are in turn defined as any amino acid with RSA>0.4 (as calculated by ISPRED4) *(29)*. This is a more flexible cutoff that takes into account the size and

shape of a protein. The *k* residues with the highest scores are labeled as predicted positive, while all others are labeled as predicted negative.

Several measures have been defined to help summarize the results of a confusion matrix. The true positive rate (TPR), or recall, describes the fraction of the positive class that was predicted correctly. Meanwhile, the precision measures the fraction of the predicted positive class that are correctly labeled. The false positive rate (FPR) measures the fraction of true negatives misclassified.

$$TPR = \frac{TP}{TP+FN} = \frac{TP}{P}$$

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{PP}$$

$$FPR = \frac{FP}{FP+TN} = \frac{FP}{N}$$

None of these measures are enough to measure the accuracy of a classifier on their own though. Labeling all queries as positive leads to a TPR of 100%. Similarly, labeling all as negative results in a 0% FPR. And precision ignores any false negatives, meaning that a high score can be achieved with an unhelpfully conservative labeling algorithm. To address these issues, the F_1 score and Matthews Correlation Coefficient (MCC) are used. The F_1 score is the harmonic mean between precision and recall.

$$F_1 = \frac{2TP}{2TP+FP+FN}$$

Although the F_1 score is widely used, ignoring the TN class can lead to deceptively high scores. The F_1 score does a good job marking a classification technique's ability to mark the positive class but ignores accuracy in marking the negative class, which is equally important. The MCC takes into account all four quadrants of the confusion matrix, resulting in a high score only when a classifier successfully classifies both positive and negative classes.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

In addition to static scores with a set cutoff for the predicted positive class, dynamic scoring methods are in use that graphically represent the performance of a classifier given different cutoffs. In precision-recall curves (PR curves) the threshold (T) is slowly increased and at each value of T the precision is plotted against the recall. In receiver operator characteristic (ROC) curves TPR is plotted against FPR in a similar fashion. To quantify these curves the area under the curve (AUC) is measured. A random classifier produces a PR AUC of $\frac{P}{P+N}$. This follows from the fact that for any random subset of the data the portion of that subset in the P class is equivalent to the portion of the full data set in the P class. In an ROC generated from a random classifier the P and N classes will be split equally between the PP and PN classes regardless of the size of those classes. This results in a case where $FPR = TPR$ regardless of T, and an ROC AUC of 0.5.

For benchmarking purposes DiscoTope and ISPIP's constituent methods were tested on the same test set as ISPIP. The small test sets used in PPI prediction (e.g. DiscoTope: 15 antigens; ISPIP bound set: 12; ISPIP unbound set: 9) mean that small differences in datasets can significantly alter results. Additionally, because F_1 scores are related to the ratio between positive and negative test cases, comparisons between F scores calculated on data sets of differing ratios are incorrect. Comparing all methods on the same data set avoided these and any other obstacles in comparison.

Results

Meta-DPI, ISPIP's predecessor, was shown to successfully outperform the methods of which it was composed, but with the replacement of PredUs 2.0 by SPPIDER, that finding is not applicable to ISPIP *(7)*. Therefore ISPIP (with each training algorithm) was tested against its components in a 3-fold CV as described above. Table 5 shows the f-scores and MCC's averaged across the test proteins when ISPIP was trained and all methods were tested on the bound dataset. Although ISPIP with xgboost did perform better than any other method, it was not statistically significant by the KS test.

Table 5: 3-Fold CV Results – Bound Training and Test Sets

	SPPIDER	ISPRED 4	DockPred	Linear Regression	Logistic Regression	Random Forest	xgboost
Average f-score	0.165	0.272	0.207	0.245	0.247	0.275	0.320
Average MCC	0.055	0.191	0.106	0.156	0.158	0.193	0.246

ROC and PR curves for these tests are seen in Figure 3 *(30)* below. SPPIDER appears to do very poorly, performing even worse than a random classifier would (shown by the gray dotted line) when the cutoff is high.

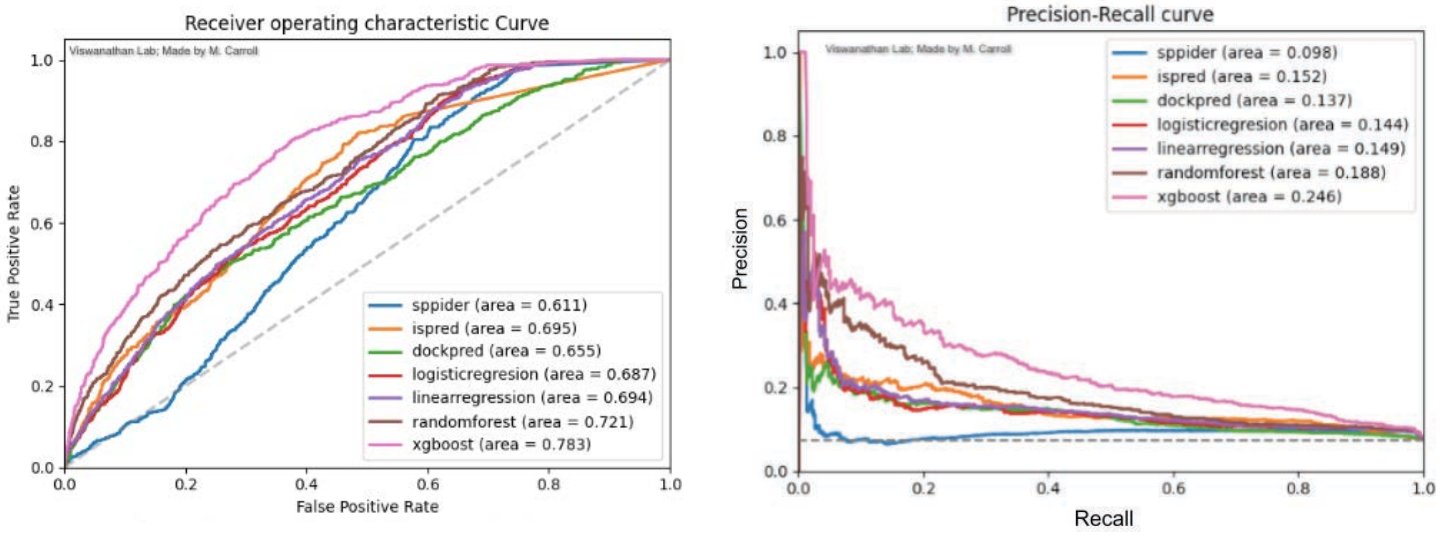

Figure 3 : Dynamic Scoring Methods: Training and Testing with Bound Dataset

In practice though, ISPIP is not designed to be used to predict the epitopes of bound proteins. Ideally, ISPIP should be able to predict the epitopes of unbound antigens, so that medical intervention could begin with only the isolation of the antigen. And it should ideally be able to do so with only bound training data, as in a real world use-case bound antigens are easier to come by. When all methods are tested on unbound antigens, ISPIP trained on the bound dataset does significantly better. These results are seen in Table 6 and Figure 4 *(30)*.

Table 6: 3-Fold CV Results – Bound Training and Unbound Test Sets

	SPPIDER	ISPRED4	DockPred	Linear Regression	Logistic Regression	Random Forest	xgboost
Average f-score	0.145	0.189	0.142	0.203	0.194	0.222	0.350
Average MCC	0.033	0.083	0.031	0.105	0.094	0.129	0.283

All methods except for ISPIP with xgboost have lower scores in all scoring methods. This is typical of unbound antigen epitope prediction which proves to be more difficult, likely due to the decreased amount of training data available. The increase in scores across scoring methods for ISPIP using xgboost is surprising and speaks towards its success as an epitope predictor. Because of its success ISPIP using xgboost and bound training data was used for future tests, and moving forward ISPIP will refer to this version.

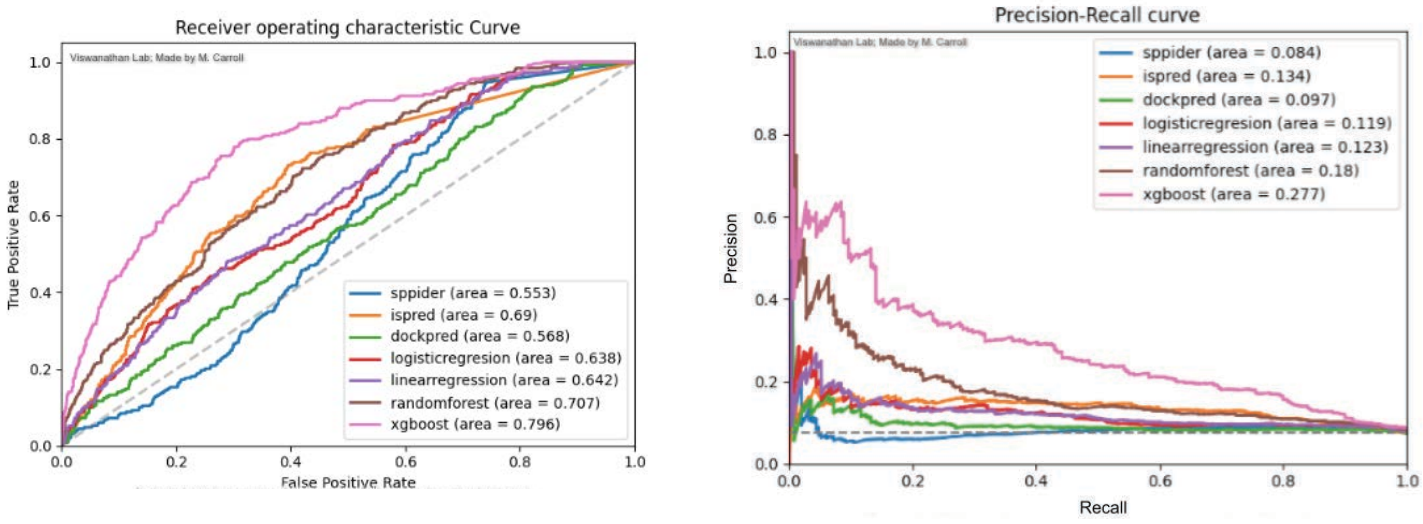

Figure 4: Dynamic Scoring Methods: Training on Bound Set; Testing with Unbound Set

Once ISPIP was shown to be better than its parts, it was benchmarked against DiscoTope 2.0. While I was most heavily involved with the data preparation for this comparison, the scores were finally calculated by other members of the team. The MCC and f-scores are seen in Table 7, and the ROC and PR curves are seen in Figure 5. The PR and ROC curves show that ISPIP vastly outperforms DiscoTope 2.0 in the task, especially for unbound antigens.

Table 7: Comparison of ISPIP and DiscoTope 2.0		
	ISPIP	DiscoTope 2.0
Average f-score on bound test set	0.320	0.238
Average MCC on bound test set	0.246	0.135
Average f-score on unbound test set	0.350	0.171
Average MCC on unbound test set	0.283	0.050

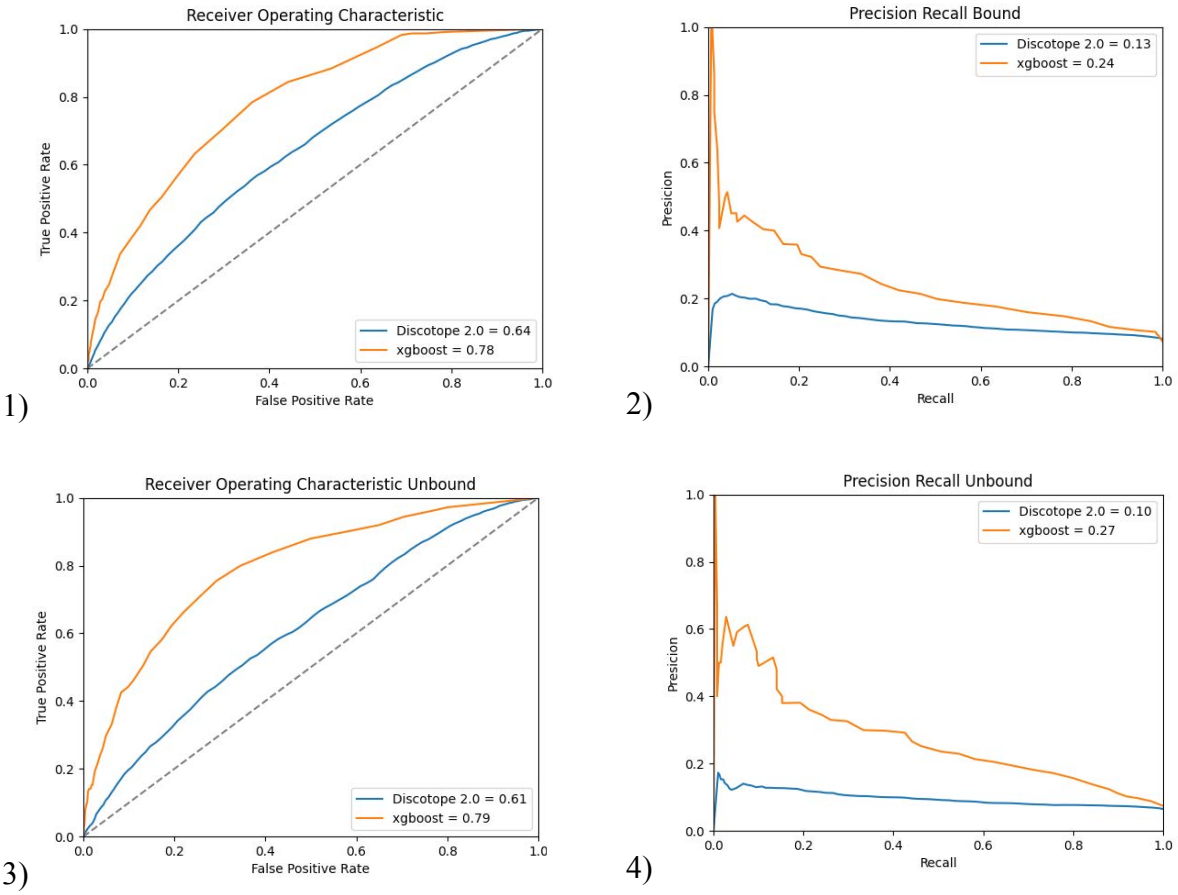

Figure 5: Discotope and ISPIP Performance: 1) ROC curve and 2) PR curve of bound test set 3) ROC curve and 4) PR curve of unbound test set

Conclusion

When trained with xgboost, ISPIP was shown to outperform each of its components and an unrelated classifier designed specifically for epitope prediction. These results suggest that ISPIP has very strong predictive power and brings reliably accurate epitope prediction one step closer. They also highlight the need for more experimentally annotated proteins and antigens so that computational methods that can theoretically be extremely powerful, have the training the data they need to reach that point.

Further Work

The individual methods that comprise ISPIP were chosen for their features without knowing how they would perform in epitope prediction. SPPIDER performed particularly badly, at times worse than a random classifier, and raises the question whether it should be included at all. Further research should be done to quantify the effects of each component method on ISPIP as a whole; it is entirely possible that ISPIP would provide better predictions if it were composed solely of ISPRED4 and DockPred. Furthermore, while ISPIP does combine methods with different strategies, none of ISPIP's constituent methods are so purely template based so as to satisfactorily replace PredUs 2.0. A true template based approach is being sought out that may replace one of the other methods or be added to them in a future version of ISPIP. Combining more orthologous prediction methods should produce the strongest results.

In addition to examining the data classifiers, ISPIP's dataset should be increased. Although small datasets are common in the field of epitope prediction due to the lack of

available data, more sources of antigens are actively being searched for in order to increase the size of ISPIP's test set. Results from a larger test set would be even more reliable and provide greater confidence that ISPIP can be generalized.

While these would help improve facets of ISPIP that already exist, another technique could add a new dimension of predictive power. Many antigens have more than one epitope, each binding to a different antibody, but not all of an antigen's epitopes have necessarily been experimentally observed binding to antibodies. By separating an antigen's interface residues into "patches" calculated based on the distance between groups of residues, individual patches may have better performance metrics. Additional patches can be interpreted as areas of the antigen that could bind to an undiscovered antibody, offering otherwise unknown avenues to target the antigen.

References

1. Branden, C. I.; Tooze, J. *Introduction to Protein Structure;* Garland Science: Independence, 1999; .
2. Parkin, J.; Cohen, B. An overview of the immune system. *The Lancet* **2001***, 357*, 1777-1789.
3. Yang, X.; Yu, X. An introduction to epitope prediction methods and software. *Rev. Med. Virol.* **2009***, 19*, 77-96.
4. Cheng, Y. Single-particle cryo-EM—How did it get here and where will it go. *Science* **2018***, 361*, 876-880.
5. Bardelli, M.; Livoti, E.; Simonelli, L.; Pedotti, M.; Moraes, A.; Valente, A. P.; Varani, L. Epitope mapping by solution NMR spectroscopy. *Journal of Molecular Recognition* **2015***, 28*, 393-400.
6. Esmaielbeiki, R.; Krawczyk, K.; Knapp, B.; Nebel, J.; Deane, C. M. Progress and challenges in predicting protein interfaces. *Briefings in bioinformatics* **2016***, 17*, 117-131.
7. Walder, M. A. Meta-DPI: A Computational Metamethod for Predicting Protein-Protein Interfaces, Yeshiva University, 2020.
8. Noble, W. S. What is a support vector machine? *Nat. Biotechnol.* **2006***, 24*, 1565-1567.
9. Anonymous. Support Vector Machine. *https://en.wikipedia.org/wiki/Support_vector_machine* (accessed 1/17/, 2023).
10. Savojardo, C.; Fariselli, P.; Martelli, P. L.; Casadio, R. ISPRED4: interaction sites PREDiction in protein structures with a refining grammar model. *Bioinformatics* **2017***, 33*, 1656.
11. Kabsch, W.; Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen‐bonded and geometrical features. *Biopolymers: Original Research on Biomolecules* **1983***, 22*, 2577-2637.
12. Fariselli, P.; Savojardo, C.; Martelli, P. L.; Casadio, R. Grammatical-Restrained Hidden Conditional Random Fields for Bioinformatics applications. *Algorithms for Molecular Biology* **2009***, 4*, 13.
13. Institute, E. B. EMBL-EBI homepage. *https://www.ebi.ac.uk/* (accessed Feb 15, 2023).
14. Viswanathan, R.; Fajardo, E.; Steinberg, G.; Haller, M.; Fiser, A. Protein—protein binding supersites. *PLoS Computational Biology* **2019***, 15*, e1006704.
15. Pierce, B. G.; Wiehe, K.; Hwang, H.; Kim, B.; Vreven, T.; Weng, Z. ZDOCK server: interactive docking prediction of protein–protein complexes and symmetric multimers. *Bioinformatics* **2014***, 30*, 1771-1773.
16. Vakser, I. A. Main-chain complementarity in protein-protein recognition. *Protein Engineering, Design and Selection* **1996***, 9*, 741-744.
17. Sobolev, V.; Sorokine, A.; Prilusky, J.; Abola, E. E.; Edelman, M. Automated analysis of interatomic contacts in proteins. *Bioinformatics (Oxford, England)* **1999***, 15*, 327-332.
18. Porollo, A.; Meller, J. Prediction-based fingerprints of protein-protein interactions. *Proteins, structure, function, and bioinformatics* **2007***, 66*, 630-645.
19. Krogh, A. What are artificial neural networks? *Nat. Biotechnol.* **2008***, 26*, 195-197.

20. Jespersen, M. C.; Mahajan, S.; Peters, B.; Nielsen, M.; Marcatili, P. Antibody Specific B-Cell Epitope Predictions: Leveraging Information From Antibody-Antigen Protein Complexes. *Front. Immunol.* **2019***, 10*.
21. Sillitoe, I.; Bordin, N.; Dawson, N.; Waman, V. P.; Ashford, P.; Scholes, H. M.; Pang, C. S.; Woodridge, L.; Rauer, C.; Sen, N. CATH: increased structural coverage of functional space. *Nucleic Acids Res.* **2021***, 49*, D266-D273.
22. Anonymous. Linear Regression. *https://en.wikipedia.org/wiki/Linear_regression* (accessed 1/25/, 2023).
23. Baždarić, K.; Šverko, D.; Salarić, I.; Martinović, A.; Lucijanić, M. The ABC of linear regression analysis: What every author and editor should know. *European science editing* **2021***, 47*.
24. Cleary, P. D.; Angel, R. The analysis of relationships involving dichotomous dependent variables. *J. Health Soc. Behav.* **1984**, 334-348.
25. Biau, G.; Scornet, E. A random forest guided tour. *Test* **2016***, 25*, 197-227.
26. Zopluoglu, C. Detecting Examinees With Item Preknowledge in Large-Scale Testing Using Extreme Gradient Boosting (XGBoost). *Educational and psychological measurement* **2019***, 79*, 931-961.
27. Hamelryck, T. An amino acid has two sides: A new 2D measure provides a different view of solvent exposure. *Proteins, structure, function, and bioinformatics* **2005***, 59*, 38-48.
28. Kringelum, J. V.; Lundegaard, C.; Lund, O.; Nielsen, M. Reliable B Cell Epitope Predictions: Impacts of Method Development and Improved Benchmarking. *PLOS COMPUTATIONAL BIOLOGY* **2012***, 8*, e1002829.
29. Hwang, H.; Petrey, D.; Honig, B. A hybrid method for protein–protein interface prediction. *Protein Science* **2016***, 25*, 159-165.
30. Carroll, M.; Roffe, A.; Viswanathan, R.; Bodzin, A. Prediction of Antibody-Antigen Epitopes Using Computational Methods. **2022**.