Predicting Protein-Fragment Binding

Emily Flynn and Michelle Wu

December 11, 2014

1 Background

Understanding the local structure surrounding the site of a protein-drug interaction can provide important insights into the mechanism of the drug and can help inform drug design and repurposing[1]. While structural information exists for a subset of protein-ligand interactions, it does not cover the space of all experimentally known interactions. As a result, the sites on proteins to which drugs or ligands bind are not always known. To address this problem, our eventual goal is to use existing structures of protein-ligand interactions to train machine-learning classifiers to predict locations of ligand binding. Because the chemical search space for ligands is large, we will approach this problem from a fragment perspective. Fragments are low molecular weight compounds (150-250 Da) that are parts of a given ligand. The same fragment may be present in a variety of ligands, so examining this problem from a fragment perspective is highly useful because it reduces the number of molecules to evaluate and allows data from multiple different ligands but the same fragment to be combined[2;3]. In order to predict the locations of fragment binding, we will focus on the sub-problem of:

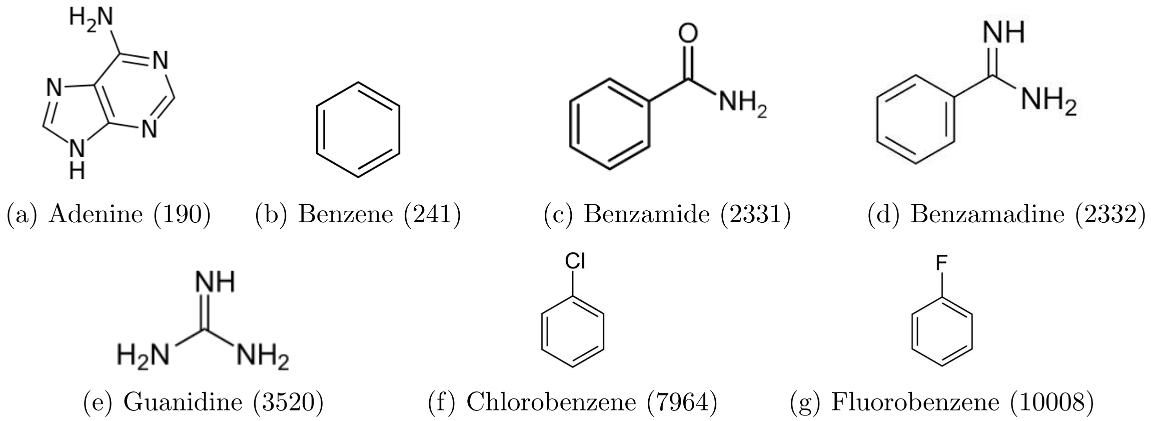
Given a location on given protein and a particular fragment, will the fragment bind to the protein at that site?

To tackle this problem, we use structures from the Protein Data Bank (PDB), a repository for structural information. Close to 100,000 experimentally-determined 3D protein structures are readily available in the PDB[4]. Many of these structures are ligand-bound, providing abundant examples of local protein environments (microenvironments) that are involved binding ligand fragments. In order to capture information about these microenvironments, we employ FEATURE[5], a computational tool developed by Russ Altman's group. This method integrates spatial and physicochemical information about a site on a 3D structure to create a 480-feature vector that can be easily used for training a model. Previous work has shown that use of this set of features is be effective for predicting fragments that bind areas of a given protein[6]. However, it has not been used to predict where a particular fragment binds on a protein, given that we know there is an interaction. A knowledge base has been previously constructed of FEATURE vectors at locations in all of protein-fragment interfaces in the PDB, each of which is annotated by the fragments that bind[6]. This database contains physicochemical information about 1.7 million local fragment-binding environments from 34,000 protein-ligand complexes. Using this as our dataset, we employ machine learning methods to train a model to classify whether a given a protein microenvironment binds a specific fragment.

2 Methods

2.1 Data

Data from the existing knowledge base from FragFEATURE[6] were used in this work. Because there is a high degree of redundancy in existing structural data, the knowledge base was filtered to contain structures with less than 30% sequence similarity. The sequence similarity cutoff of 30% was chosen because it is generally considered to be the cutoff between homologous and non-homologous structures. We chose to use such a stringent cutoff because we want to be able to predict whether a fragment will bind a given site on a protein even if the protein is not similar to known proteins. The best resolution chain from each BLASTclust 30% sequence similarity cluster was selected. The subset of FEATURE vectors in the knowledge base generated from this set of dissimilar protein chains was then used as our data set. These data were then divided into positive and negative examples for each of seven fragments; positive



(a) Adenine (190) (b) Benzene (241) (c) Benzamide (2331) (d) Benzamadine (2332)

(e) Guanidine (3520) (f) Chlorobenzene (7964) (g) Fluorobenzene (10008)

Figure 1: Fragments classified in this study.

examples are FEATURE vectors collected from sites bound the given fragment, while negative examples are not bound to the given fragment in the knowledge base.

For k-nearest neighbors and Bernoulli naive Bayes classifiers as well as SVM models, data was binarized based on medians previously drawn from a larger set of protein microenvironments.

2.2 Supervised Learning Algorithms

All learning algorithms were implemented in Python using the `scikit-learn` library. Models were evaluated using stratified 10-fold cross validation. The ensemble method aggregated the three top performing models - SVR.linear, SVR.Gaussian, and RFR. Random forests were built with ten trees. Bernoulli naive Bayes was implemented with Laplace smoothing. Learning curves were generated by computing errors on 70% training and 30% test sets.

Reduced dimensionality FEATURE vectors were generated using two methods. First, PCA was used to generate principal components capturing the highest variance directions. Based on previous analysis, the top 100 features were used for training of the model. Second, features were clustered and representative features were selected from each cluster to produce a 100-feature vector that contains maximal information content. These alternative feature sets were used to train models as before.

3 Results and Discussion

Seven fragments, whose structures and ID numbers are shown in Figure 1, were chosen as representative fragments for the creation of algorithms to predict binding because of their high frequency in the database and varying structural and functional groups. A number of classification and regression algorithms were applied to the dataset to create models for each fragment. The output of regression models were interpreted as a binding score, representing the affinity with which the protein binds the ligand.

Initially, we used the entire data set of FEATURE vectors to predict whether a given site will bind that fragment, however this initial performance was poor. Because environments vary greatly across residues, we decided to separate the data by the amino acid residue the FEATURE vector was centered around. We then trained a model for each fragment using data from the amino acid the fragment was most frequently found next to. This separation of the data greatly improved the results of our classifier, which are shown in Figure 2.

We tried a variety of models on our data set; the training and testing errors for each are listed in 1. In general, the use of SVM as a model for regression was the most effective. Both the linear

	KNN (5)	NB	RFR	SVR linear	SVR rbf	SVC linear	SVC rbf	ensemble
train	0.1183	0.3019	0.1208	0.1991	0.000	0.1484	0.0249	0.0150
test	0.1844	0.3019	0.1811	0.1998	0.1504	0.1668	0.1658	0.1822

Table 1: Training and testing error for adenine fragment residue valine models. Note that a naive classifier with only negative labels results in a 0.25 error rate.

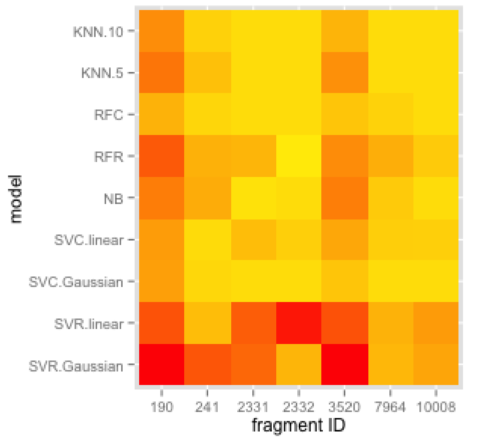


Figure 2: This heatmap shows the area under the ROC curve for various models over all 7 fragments.

and Gaussian kernel performed relatively well, although the optimal kernel varied depending on the fragment being considered. Interestingly, applying SVM when viewing the problem as a classification resulted in far lower performance. This suggests that it is most informative to view ligand binding on a spectrum rather than as a binary event. This observation is consistent with the biological framework, as proteins may bind their ligands with varying affinities. Further, microenvironments within a binding pocket may contribute differently to the binding of ligand fragments.

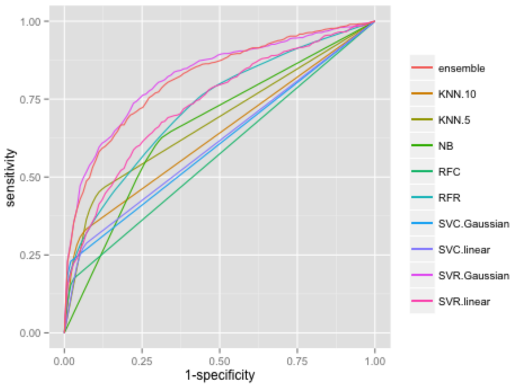


Figure 3: ROC curves show the performance of various models for adenine binding.

We were most successful at classifying adenine (fragment ID 190). ROC curves for all models of adenine binding are shown in Figure 3. The relative performance of models for adenine reflects a trend similar to that of other fragments, with both SVM and RF regressors showing the highest performance. KNN, SVM and RF classifiers show low sensitivity, meaning that they are producing many false negatives. The ensemble method does not improve performance over the the top-performing model, suggesting that the various models are likely making the same mistakes.

In order to diagnose and assess the performance of our classifier, we examined the training and test error on increasing portions of our data set. We focused on the errors associated with our best performing classifier (support vector regression with a Gaussian kernel) on adenine. Figure 4 shows the resulting learning curve. Training error is low across all sample sizes, and only slightly increases for with the largest portion of the data set, while test error slowly decreases with increasing sample size. This indicates that we have problems with high variance in our model, which suggests that increasing our training set size and trying a smaller number of features would be helpful.

Given this diagnosis that our model is overfitting the training data, we varied the regularization parameter in order to put a stronger constraint on the fitting parameters. However, this was not effective in reducing error on the test

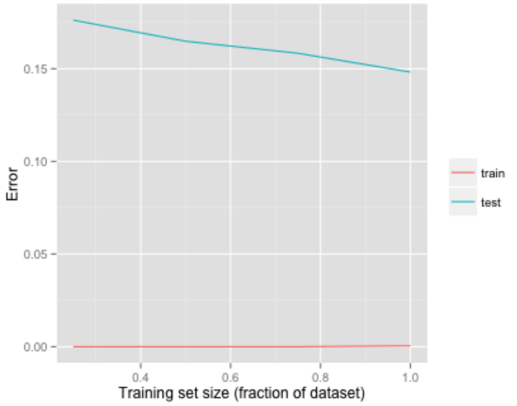


Figure 4: Error curve for adenine using support vector regression

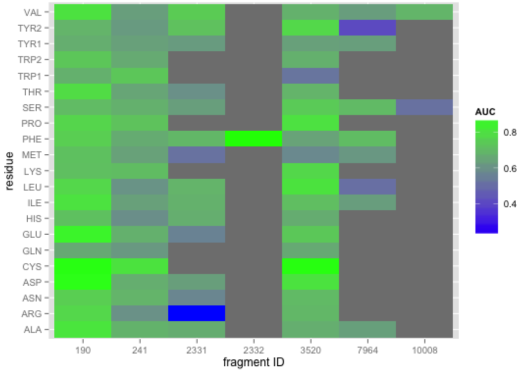


Figure 5: This heatmap shows the AUROC for each residue type for all fragments classified. Gray boxes indicate insufficient samples in that category.

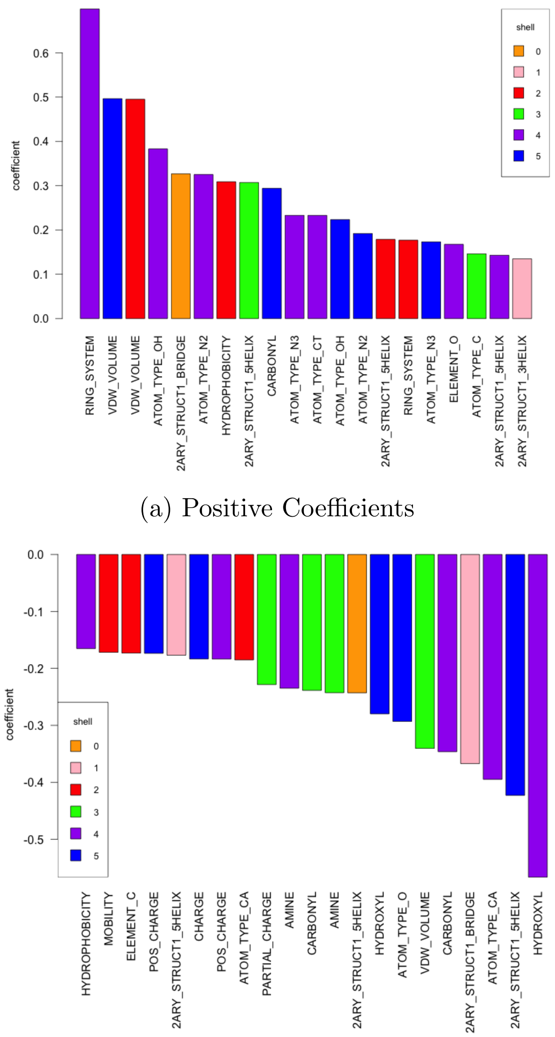
set. Decreasing the parameter, which loosens the contraint on the fitting parameters, made performance worse, as expected, but increasing the parameter did not have an effect on error rates. This indicates that the original conditions, with a regularization parameter of 1, already produced a relatively sparse model. Additionally, we tried reducing the dimensionality of our feature space using two different methods previously shown to be effective in improving FEATURE models. Neither the use of principal components nor reduced feature sets decreased the large gap between training and test error. This suggests that the original model may already be implicitly decreasing the dimensionality by putting small weights on certain features, further supporting the theory that the original model is sparse.

Further evaluation of our best model, SVR with a Gaussian kernel, showed that performance varied widely over different residue types, as shown in Figure 5. This indicates that specific amino acids are important in the binding of each ligand. For example, using microenvironments centered around cysteine, glutamate, and aspartate residues were best for the prediction of adenine-binding. This could be a result of hydrogen bonding or other electrostatic interactions that serve as the center of the fragment-protein interaction; however, more examples are needed to determine the biological significance of relative importance of different residue types in different fragment-binding interactions.

We also analyzed the features given the highest weight in the SVC linear classifier model. The top 20 positive and negative coefficients for the features are shown in Figure 6. This analysis showed that the most important positive predictor of adenine binding was the presence of a ring system in the 5th shell, and the most important negative predictor was the presence of a hydroxyl group in the 5th shell.

4 Conclusions

Overall, no model showed consistent performance across residue centers and across fragments, suggesting that we need to take into account more information about context in order to produce a good classifier. In general, ligands, as well as fragments, associate with a binding pocket in a protein, in which many interactions across many residues help them to bind. A single FEATURE vector may be too limiting in what spatial features and orientation information it can represent. In the future, we will expand our classifier by scoring multiple microenvironments surrounding a binding pocket to get an aggregate score to predict binding.



(a) Positive Coefficients

(b) Negative Coefficients

Figure 6: These plots show the features with the highest absolute value coefficients for the adenine SVC linear classifier for valine residues.

Using regression algorithms, which we have shown to be more effective than classification algorithms, we can generate binding scores that can be easily aggregated across a cluster of neighboring microenvironments which may form a pocket. This will allow us to expand towards the ultimate goal of predicting the binding site of a ligand on a protein, given an experimentally known interaction.

In addition, the amount of data we had was a major constraint on the performance of our classifiers, as evidenced by the constant downward trend in test error shown in Figure 4. Despite our efforts to reduce overfitting with PCA and trying different regularization parameters, it appears that our models still have high variance. We had insufficient data for many residues, preventing us from training a model for those microenvironment types. As a result, we hope to obtain an expanded dataset with a reduced sequence similarity cut off and try training models on this. We are also confident that as the PDB grows at a rate of almost 10,000 new structures each year[4], we will be able to incorporate more data and vastly improve our models.

5 Acknowledgments

We would like to thank Russ Altman for his guidance on this project and Grace Tang for providing the FragFEATURE knowledge base.

6 References

[1] Wang, J.-C. & Lin, J.-H. Scoring functions for prediction of protein-ligand interactions. *Current pharmaceutical design* **19**, 2174–82 (2013).

[2] Hann, M., Leach, A. & Harper, G. Molecular Complexity and Its Impact on the Probability of Finding Leads for Drug Discovery. *Journal of Chemical Information and Modeling* **41**, 856–864 (2001).

[3] Hajduk, P. J. & Greer, J. A decade of fragment-based drug design: strategic advances and lessons learned. *Nature reviews. Drug discovery* **6**, 211–9 (2007).

[4] Protein data bank. URL `http://www.rcsb.org/pdb/home/home.do`. Accessed 2014-11-13.

[5] Halperin, I., Glazer, D. S., Wu, S. & Altman, R. B. The FEATURE framework for protein function annotation: modeling new functions, improving performance, and extending to novel applications. *BMC genomics* **9 Suppl 2**, S2 (2008).

[6] Tang, G. W. & Altman, R. B. Knowledge-based fragment binding prediction. *PLoS computational biology* **10**, e1003589 (2014).