Predicting Correctness of Protein Complex Binding Orientations

Sarah Gurev
Stanford University
sgurev@stanford.edu

Nidhi Manoj
Stanford University
nmanoj@stanford.edu

Kaylie Zhu
Stanford University
kayliez@stanford.edu

1. Introduction

1.1. Motivation

Protein-protein interactions are integral to most biological processes. Scientists would like to know the structure of these protein complexes in order to understand the processes they are used for. Unfortunately, most protein complexes do not have an experimentally determined structure, and these structures will not be solved given the current experimental techniques for structure determination. Therefore, it is useful to be able to model the structure of protein complexes using the individual structures of each protein via docking (simulation) methods. However, numerous orientations can result from computational docking, and so being able to predict which of the simulated complexes are the correct orientation would be essential in order to use the modelled structures. As a result, we developed models that are capable of predicting whether the binding orientation of a protein-protein docking is correct.

1.2. Problem Formulation

Our project can be formulated as both a regression and classification task. Our inputs are Protein Data Bank (PDB) files that have energies and other information about a simulated protein complex as well as the position of all atoms in the complex. We also have as labels the root mean squared deviation (RMSD) of all atom positions in the simulated complex to the experimentally determined structure of each complex. A smaller RMSD signifies an orientation that is closer to the true binding orientation of the protein complex. When we frame the problem as a regression task, our SVM (regression) model takes features from the PDB file to output a predicted RMSD. Previous work in the field has shown that an RMSD of 0-1 is considered very good, 1-2 is good, and 2-4 is acceptable, thus we threshold our RMSD at 4 for our classification problem. Then, our classification model (either SVM or 3D CNN) outputs a binary prediction as to whether a binding orientation is correct.

2. Related Work

We build our SVM model after studying various related works of research such as Pairpred, which employs SVM methods to predict whether a pair of residues from two different proteins interact [2]. Another recent work uses SVMs to improve docking scoring functions in order to better predict binding affinity [16]. Currently, predicting whether a docking model is correct would be done via ZRANK [18] or ProQDock [3]. Similar to our SVM model, ZRANK uses electrostatics, van der Waals, and desolvation information to score docking predictions. However, they use a downhill simplex minimization algorithm to find the correct weights for each feature. ProQDock later builds on ZRANK's score model, and cleverly trains an SVM by combining different types of features that describe both the protein-protein interface and the overall physical chemistry. Our SVM model is based on the work from ProQDock, though narrowed down to four key features. ProQDockZ, a hydbrid of ProQDock and ZRANK, should be considered state-of-the-art in terms of finding correct protein-protein docking models.

With the advent of convolutional neural networks (CNNs) in recent years, we have seen no short of remarkable developments for their powerful performance in fields ranging from image recognition to bio-structural analysis, with constantly newer, deeper and better-performing CNN architectures. In 2015, ResNet-50 and ResNet-101 architectures were introduced for their relative ease to optimize and their higher accuracy gained from considerably increased depth. [13] Such leaps in advances in deep learning methods have greatly facilitated research in protein structural analysis, as knowledge and insights in the field of image recognition is adapted and extended to deal with three-dimensional bio-physio-chemical data and spatial features.

We, too, are inpsired by the multitude of recent endeavours turn to 3D CNN architecture for structure-based analysis of protein and other biomolecules. One such established literature documents EnzyNet, a 2-layer 3D-convolutional neural network classifier that predicts the Enzyme Commission number of enzymes based only on their voxel-based spatial structure. [1] While most conventional research to date explores relatively shallow 3D architectures, more con-

temporary algorithms employ much deeper 3D convolutional networks. For instance, one such model is successfully used to predict the ranking of model structures solely on the basis of their raw three-dimensional atomic densities, without any feature tuning [8]. Our paper draws inspiration from such research findings as well as the 3D ResNeXt architecture in order to find accurate predictions of correctness of protein complex binding orientations.

We also refer to established literature such as AtomNet [22], which uses a similar 3D CNN to predict protein-ligand bioactivity, and BIPSPI, which attempts to predict partner-specific protein-protein binding sites using classical machine learning [20]. While these pose related questions – and mirror our project in terms of the methodologies used in many ways – none of these investigate whether the orientation is correct from protein-protein interaction. As a result, the input to our 3D CNN model is significantly different from the sequence and structural features fed into BIPSPI's model. AtomNet, which predicts bioactivativy rather than binding orientation and works on protein-ligand interactions rather than protein-protein complexes, has the most similar input to their model as to our 3D CNN, whose inputs we will explain more in the following section. They also focus on the center of mass of the interaction and randomly rotate before making a voxelized cube. However, their voxels contain structural information beyond atom type, they only need one cube because ligand binding site will be smaller (and so it is not necessary to aggregate results), and they unfold their cube into a 1D vector.

3. Data

3.1. Dataset

Our dataset is a modified version of Docking Benchmark 5 (DB5) [21], which is a group of protein complexes commonly used for protein-protein interaction prediction. For each protein complex in DB5, we have 10,000 PDB files (positions for all atoms) that correspond to different minimized Haddock [10] dockings of the two proteins in the complex. We received this dataset from João Rodrigues in the Stanford Levitt Lab.

3.2. Data Preprocessing

We split the 135 complexes we have from DB5 into train, validation and test sets. Each complex is only in exactly one of the sets, with 70% of the complexes in train, 20% in validation and 10% in test. With high class imbalance due to our dataset containing significantly more negative examples (incorrect binding orientations) than positive, we choose to undersample the negative examples. Therefore, for each of the complexes, we include all positive examples and an equal number of negative examples in the corresponding training, validation, or test set, using an RMSD of 4 as the threshold for correct orientation (positive). Ultimately, the training set has 80% of the positives (and of the used data), the validation set has 15% and the test set has 5%. For our SVM model, we then extract four different features (electrostatics, van der waals, buried surface area, and solvation energy) from each PDB file and normalize each feature by the mean and standard deviation of the training data.

For our 3D CNN, we process the PDB files into cubic representations of the interface. We split the interface into cubes of 10 cubic angstroms, with 1 cubic angstrom voxels, giving a value of 0 (no atom at that location) or a number representing the atom present. To get the cube representations, we first remove cofactors if present and randomly rotate the complex (by multiplying all atoms by a rotation matrix with 3 randomly generated angles, first around the x-axis, then y-axis, and z-axis). We then find the center of mass of the alpha carbons of all residue pairs that interact between the two proteins. We then cluster these centers of mass into regions of 10 cubic angstroms, and the center of each cluster becomes the center of the cube. Overall, for our SVM and CNN inputs, our training set has 7812 samples and our test set has 1472 samples.

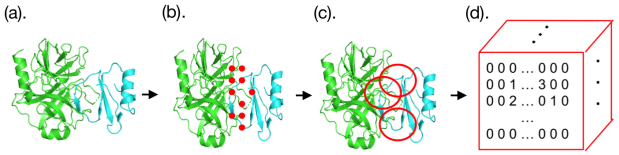

Figure 1. Make cubes pipeline: (a). Randomly rotate atom positions of docked protein complex, (b). Find center of the interaction, (c). Find cluster center and all atoms within radius, (d). For each cluster, make cubes with 1 cubic angstrom voxels that are 0 if no atom and number for atom type otherwise

4. Methods

4.1. Support Vector Machine (SVM)

For our baseline approach, we use a support vector machine (SVM), which is a supervised learning algorithm used to separate data by calculating the margin between data points [14]. Using the scikit-learn library for implementation [17], we parameterize our classifier with w, b and write our classifier as $h_{w,b}(x) = g(w^T x + b)$ where g(z) = 1 if z $\geq$ 0 and g(z) = -1 otherwise. The functional margin of the parameters with respect to training example $(x^{(i)}, y^{(i)})$ is $y^{(i)}(w^T x^{(i)} + b)$. The functional margin can be thought of as a measure of confidence in correct hypothesis so we want it to be large in scale. We use the following optimization problem, where the solution is the optimal margin classifier.

$$min_{w,b} \frac{1}{2}||w||^2$$
$$\text{such that } y^{(i)}(w^T x^{(i)} + b) \geq 1, i = 1, ..., m$$

We pass the four features we listed in the previous section, all of which serve to describe the favorability of a protein complex interaction, as inputs into the SVM, which then outputs a prediction of RMSD. We perform regression on RMSD values as well classification using a threshold of 4.0.

Previous related work has found that with 5-fold cross validation and a radial basis function (RBF) kernel, the optimal C and γ values could be found using grid search over the following ranges to compromise between training error and margin: C from 2^{-15} to 2^{10} and γ from 2^{-10} to 2^{10} incrementing by log 2 [3]. Thus, we employ 5-fold cross validation and a grid search for our SVM experimentation using our training set with the same parameter value ranges.

4.2. ResNeXt 3D CNN

We additionally construct and investigate performance of a 3D CNN model on the cubic data we prepared as training input. The model is inspired by cutting-edge architecture ResNet, but utilizes 3D convolutions to better process and analyze three-dimensional data.

A homogeneous, highly-modularized architecture, ResNeXt is constructed by repetition of a building block that aggregates a set of transformations with the same topology. The model introduces group convolutions in ResNeXt blocks, as well as a new dimension called "cardinality" (which is the size of the set of transformations), an addition to factors such as dimensions of depth and width. [23] We employ ResNeXt, but with 3D convolutions and other modifications tailored to our task. In building the network, we build upon previous work and research relating to kinetics and video comprehension. [12] [7] [9] We replace the final fully connected layer with one that has a single output, after which we apply a sigmoid nonlinearity ($f(t) = \frac{1}{1+e^{-t}}$). Layer-wise details of our 3D ResNext model is delineated in Figure 2.

For a single example in the training set, we optimize the weighted binary cross entropy loss

$$\begin{aligned} L(X, y) \quad = \quad & -w_+ \cdot y \log p(Y = 1|X) \\ & -w_- \cdot (1 - y) \log p(Y = 0|X), \end{aligned}$$

where $p(Y = i|X)$ denotes the probability that the network assigns to the label i, $w_+ = |N|/(|P| + |N|)$, and $w_- = |P|/(|P|+|N|)$ where $|P|$ and $|N|$ are the number of correct and incorrect cases of protein complex binding orientations in the training set respectively.

5. Experiments

5.1. Evaluation Metric

We evaluate the accuracy of predicted RMSD values of protein orientations (regression) using R-squared statistic.

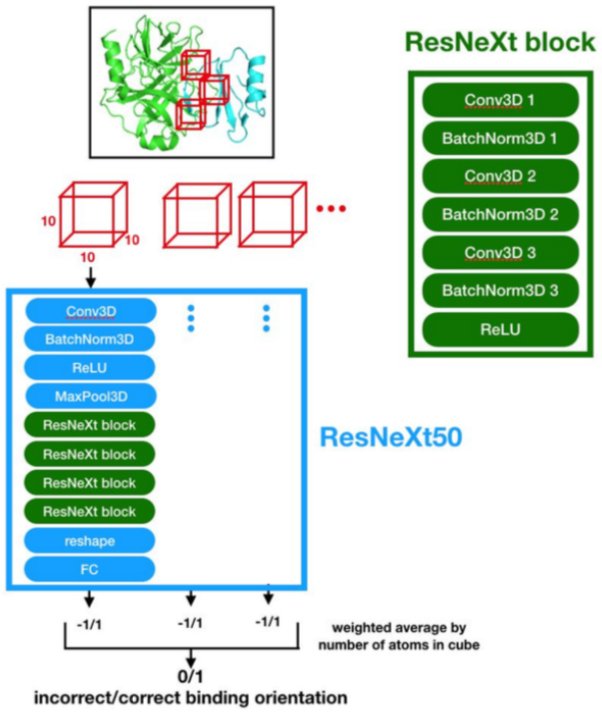

Figure 2. ResNeXt model architecture

R-squared, the coefficient of determination, is a statistical measure of how close the data is to the fitted regression line [6].

While at first, we frame the problem as a regression task, we later select a RMSD threshold of 4.0 and convert the problem into a classification task. For evaluating classification performance, we utilize ROC-AUC score and F1 score. ROC-AUC score is the Area Under the Receiver Operating Characteristic Curve from prediction scores, which is created by plotting the fraction of true positive rate (sensitivity) against the fraction of false positive rate (FPR), at various threshold settings [4] [5]. F1 score is the harmonic mean of the precision and recall [11]. These classification metrics reach their best values at 1 and worst values at 0.

5.2. Support Vector Regression

Framing the problem as a regression task, we implement an SVM baseline. We experiment with different SVM hyperparameters and run a grid search for the optimal model performance. From preliminary experiments we see that the RBF kernel outperforms alternatives such as sigmoid, linear and polynomial. We then experiment with different C and γ values for our RBF kernel SVM. C serves as a regularization parameter in SVM, trading off correct classification of training examples with maximization of the margin of the decision function. The γ parameter defines the extent of influence of a single training example, also known as the

inverse of the radius of influence of samples selected by the model as support vectors. We present our regression results (train scores based on an average of 5-fold cross validation) in Table 1.

	Hyperparameters	Train	Test
R^2	$C = 4, \gamma = 32$	0.4445	0.171

Table 1. Results for SVM regression and corresponding hyperparameter values. Train scores were computed using 5-fold cross validation. C is the regularization parameter and gamma is the inverse of the radius of influence.

5.3. Support Vector Classification Baseline

Seeing that our regression baseline does not perform well, we instead propose the reframing of predicting correctness of protein binding orientations as a classification task, in an effort to improve our model performance. We proceed to run a grid search for the optimal SVM parameters (C and γ values) using RBF kernel SVM. We present our classification results (train scores based on an average of 5-fold cross validation) in Table 2. The model performs well on train but seems to overfit.

	Hyperparameters	Train	Test
F1 Score	$C = 2, \gamma = 32$	0.888	0.705
ROC-AUC	$C = 2, \gamma = 64$	0.950	0.876

Table 2. Results for SVM classification and corresponding hyperparameter values. C is the regularization parameter and gamma is the inverse of the radius of influence.

5.4. ResNeXt 3D CNN

Seeing as the SVM classification overfits on the train set, we experiment with a ResNeXt 3D CNN. We prepare cubic representations of the protein interface region. After passing in the 3D cubes into the CNN and outputting a prediction for each cube, we aggregate model output decisions for each cube in the docking (orientation) by computing a weighted average over all cubes along the interface, where the weight depends on the number of atoms present in the respective cube.

We experiment with various initial learning rates, batch sizes, and ResNeXt model depths. We decide to use learning rate reduced on plateau, which is common practice and known to help model optimization. We experiment with initial learning rates of $1e{-}3$, $1e{-}4$, and $0.5e{-}3$. We find that our results with initial learning rate $1e-3$ are poor so we reduce the value to $1e-4$ in an attempt to increase training performance. However, this model learns too slowly, so we employ an intermediate value of $0.5e-3$ which performs well. Thus, we use an initial learning rate of 0.0005 that is decayed by a factor of 10 each time the validation loss plateaus after an epoch, and pick the model with the lowest validation loss.

The mini-batch size is chosen to be 32, after empirically testing mini-batch sizes of 16, 32, and 64. We chose the cardinality value to be 32 which is commonly used and generally works well [23]. Adam optimizer, which computes adaptive learning rates on a per parameter basis, empirically works well in practice and is widely adopted in the deep learning community. Thus the network is trained end-to-end using Adam optimizer with standard parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) [15].

Lastly, we experiment with various model depths. We test ResNeXt50 model as well as a ResNeXt101 model and find that the ResNeXt50 model had better results. We figure that the ResNeXt101 model might be trying to learn a more complicated function over the data and overfits on the training data. Seeing that the ResNeXt50 model performs better, we decide not to experiment with larger model depths.

The results of the ResNeXt50 and ResNeXt101 models are provided in Table 3 and Table 4 respectively. Our best model is ResNeXt50, which has a F1 score of 0.929 and an ROC-AUC of 0.954 that, as expected, outperforms our SVM results. A normalized confusion matrix of this model is computed and provided in Figure 3.

	Hyperparameters	Train	Test
F1 Score	$lr = 0.5e-3, bs = 32$	0.956	0.929
ROC-AUC	$lr = 0.5e-3, bs = 32$	0.981	0.954

Table 3. ResNeXt50 Results for classification and corresponding hyperparameter values. lr is the initial learning rate and bs is the mini-batch size.

	Hyperparameters	Train	Test
F1 Score	$lr = 0.5e-3, bs = 32$	0.851	0.825
ROC-AUC	$lr = 0.5e-3, bs = 32$	0.903	0.864

Table 4. ResNeXt101 Results for classification and corresponding hyperparameter values. lr is the initial learning rate and bs is the mini-batch size.

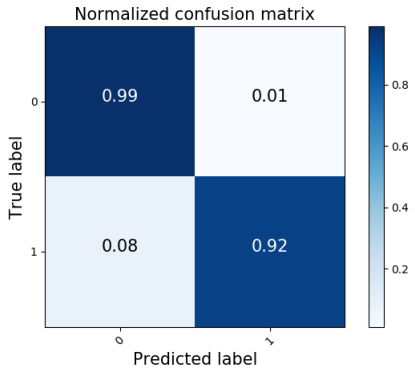

Figure 3. ResNeXt50 normalized confusion matrix where label is a binary value predicting correctness of protein binding orientation

To qualitatively evaluate our model performance, we used PyMOL which is molecular visualization software to create 3D visualizations of protein dockings. Figure 4 displays, on the left, an incorrect protein docking that was predicted as 0, which is what the CNN model should do. The right side of Figure 4 depicts a correct protein docking that was incorrectly predicted as 0, which demonstrates a protein docking that the CNN model misclassifies.

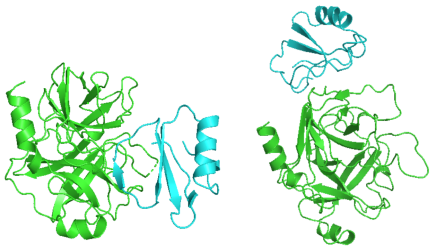

Figure 4. [Left] Incorrect protein docking with RMSD value 16.37 ($\geq$4.0) that was properly predicted as 0. [Right] Correct protein docking with RMSD value 0.5 ($\leq$4.0) that was misclassified and predicted as 0.

Until now, we train and evaluate the models on a balanced dataset that we sampled (explained in the Data Preprocessing section). Since the original dataset was unbalanced with more negative samples, in order to further evaluate our model performance, we determine the efficacy of our model on an unbalanced dataset. In real life, there will be many more negative than positive simulated examples, and so it is valuable to know how well our model predicts positive examples. Since we did not want to use all of the docking models from the complexes in our validation set (which would be a very large number of dockings and would take a lot of time to make the cubic inputs for), we subsampled from our test set negative and positive inputs that would maintain a near-original ratio of $10:700$ positive to negative samples. The performance of our best model (ResNeXt50 CNN) on an unbalanced dataset is as follows in Table 5 which indicates reasonable results.

	Hyperparameters	**Test**
F1 Score	$lr = 0.5e-3, bs = 32$	0.824
ROC-AUC	$lr = 0.5e-3, bs = 32$	0.850

Table 5. ResNeXt50 Results on unbalanced dataset. lr is the initial learning rate and bs is the mini-batch size. Test results based on trained model with performance seen in Table 3.

6. Conclusion

We consider the problem of predicting correctness of a protein binding orientation (docking). When formulated as a regression task (predicting RMSD values), we have poor performance with our SVM regression baseline. Thus in order to improve performance, we reframe the problem as a classification task, which achieved more promising results as expected. This SVM classification baseline seems to be overfitting on the training set so we consider a 3D CNN implementation that accepts 3D cubes of atom position data from the protein interface region. Using the model predictions on each cube, we compute a weighted average over all cubes in a respective protein orientation interface and output a final binary value predicting if the protein orientation is correct. Upon empirical analysis, our best model is a ResNeXt50 3D CNN, which has a F1 score of 0.929 that, as expected, outperforms our SVM results. We believe this model outperforms the ResNeXt101 model perhaps because the latter tries to learn a complicated function on the data and thus overfits on the training set.

7. Future Work

In the future, we would also like to evaluate precision (percentage of called positives that are true positives) and recall (percentage of positives that are called positive) on the unbalanced dataset [11]. It will also be interesting to determine from our ranked probabilities of being correct for the unbalanced dataset, the average rank of the top true positive. We can imagine this as how many structures would a scientist have to examine in order to get a true orientation.

Given our promising results with ResNeXt 3D CNN, in future work, we would like to experiment with a multi-channel 3D CNN and use more than just the atom type and position data to incorporate other attributes such as atom charge, specific atom type (such as alpha carbon). We would also like to experiment with different ensemble techniques and try to outperform our results.

With more time, it would also be fruitful to experiment with different ways of generating cubes to CNN input. We want to ensure that as much information of the binding interface is captured by the cubic representation. Therefore, it could be meaningful to vary how much overlap there is between the cubes. It could also be valuable to randomly rotate each cube rather than just each orientation in order to help reduce overfitting. Lastly, since more data is generally helpful, we could potentially oversample positive data (so that our original dataset is more balanced) by using data augmentation via molecular dynamics which would allow us to avoid undersampling our original dataset [19].

8. Acknowledgements

Thanks to João Rodrigues in the Levitt Lab for the dataset and advice as well as Raphael Townshend for his guidance on this project.

9. Contributions

All group members contributed to the SVM experiments and the writing of the paper. Sarah also extracted features for the SVM and processed the data (cubes) for the CNN, while Kaylie implemented the base code for model training and deployment. All group members worked on creating balanced data sets for the CNN and then worked on the 3D CNN architecture implementation to ensure that all project work was evenly spread out.

References

[1] A. Amidi, S. Amidi, D. Vlachakis, V. Megalooikonomou, N. Paragios, and E. I. Zacharaki. EnzyNet: enzyme classification using 3D convolutional neural networks on spatial representation. *arXiv e-prints*, page arXiv:1707.06017, July 2017.

[2] F. Amir, A. Minhas, B. J. Geiss, and A. B. hur (corresponding. Pairpred: Partner-specific prediction of interacting residues from sequence and structure, 2013.

[3] S. Basu and B. Wallner. Finding correct proteinrotein docking models using proqdock. *Bioinformatics*, 32(12):i262–i270, 2016.

[4] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145 – 1159, 1997.

[5] C. D. Brown and H. T. Davis. Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 80(1):24 – 38, 2006.

[6] A. C. Cameron and F. A. Windmeijer. An r-squared measure of goodness of fit for some common nonlinear regression models. *Journal of econometrics*, 77(2):329–342, 1997.

[7] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng. Multi-Fiber Networks for Video Recognition. *arXiv e-prints*, page arXiv:1807.11195, July 2018.

[8] G. Derevyanko, S. Grudinin, Y. Bengio, and G. Lamoureux. Deep convolutional networks for quality assessment of protein folds. *arXiv e-prints*, page arXiv:1801.06252, Jan. 2018.

[9] A. Diba, M. Fayyaz, V. Sharma, M. Mahdi Arzani, R. Yousefzadeh, J. Gall, and L. Van Gool. Spatio-Temporal Channel Correlation Networks for Action Classification. *arXiv e-prints*, page arXiv:1806.07754, June 2018.

[10] C. Dominguez, R. Boelens, and A. M. J. J. Bonvin. Haddock: a proteinprotein docking approach based on biochemical or biophysical information. *Journal of the American Chemical Society*, 125(7):1731–1737, 2003. PMID: 12580598.

[11] C. Goutte and E. Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In D. E. Losada and J. M. Fernández-Luna, editors, *Advances in Information Retrieval*, pages 345–359, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[12] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? *CoRR*, abs/1711.09577, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, July 1998.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[16] S. L. Kinnings, N. Liu, P. J. Tonge, R. M. Jackson, L. Xie, and P. E. Bourne. A machine learning-based method to improve docking scoring functions and its application to drug repurposing. *J Chem Inf Model*, 51(2):408–419, Feb 2011.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] B. Pierce and Z. Weng. ZRANK: reranking protein docking predictions with an optimized energy function. *Proteins*, 67(4):1078–1086, Jun 2007.

[19] A. Prez, G. Martinez, and G. De Fabritiis. Simulations meet machine learning in structural biology. *Current opinion in structural biology*, 49:139–144, 02 2018.

[20] R. Sanchez-Garcia, C. O. S. Sorzano, J. M. Carazo, and J. Segura. Bipspi: a method for the prediction of partner-specific protein–protein interfaces. *Bioinformatics*, page bty647, 2018.

[21] T. Vreven, I. H. Moal, A. Vangone, B. G. Pierce, P. L. Kastritis, M. Torchala, R. Chaleil, B. Jiménez-García, P. A. Bates, J. Fernandez-Recio, A. M. Bonvin, and Z. Weng. Updates to the integrated protein–protein interaction benchmarks: Docking benchmark version 5 and affinity benchmark version 2. *Journal of Molecular Biology*, 427(19):3031 – 3041, 2015.

[22] I. Wallach, M. Dzamba, and A. Heifets. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *CoRR*, abs/1510.02855, 2015.

[23] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016.

Code repository: https://tinyurl.com/yc25f9hv