

# Using Machine Learning to Improve Ensemble Docking for Drug Discovery

Tanay Chandak<sup>#</sup>, John P. Mayginnes<sup>#</sup>, Howard Mayes, and Chung F. Wong\*

Department of Chemistry and Biochemistry and Center for Nanoscience, University of Missouri-St. Louis, Saint Louis, MO 63121, USA

\*wongch@umsl.edu

## Abstract

Ensemble docking has provided an inexpensive method to account for receptor flexibility in molecular docking for virtual screening. Unfortunately, as there is no rigorous theory to connect the docking scores from multiple structures to measured activity, researchers have not yet come up with effective ways to use these scores to classify compounds into actives and inactives. This shortcoming has led to the decrease, rather than an increase in the performance of classifying compounds when more structures are added to the ensemble. Previously, we suggested machine-learning, implemented in the form of a naïve Bayesian model could alleviate this problem. However, the naïve Bayesian model assumed that the probabilities of observing the docking scores to different structures to be independent. This approximation might prevent it from achieving even higher performance. In the work presented in this paper, we have relaxed this approximation when using several other machine learning methods -- k nearest neighbor, logistic regression, support vector machine, and random forest – to improve ensemble docking. We found significant improvement.

Keywords: ensemble docking, machine-learning, k nearest neighbor, logistic regression, support vector machine, random forest, **protein kinases**

# these authors contribute equally

## Introduction

As rigid-receptor models can give high false negative rates in virtual screening, researchers have considered many ways to account for protein flexibility in docking small molecules to protein targets.<sup>1-10</sup> Ensemble docking has emerged as one of the most popular methods because it allows a large number of compounds to be screened with modest costs.<sup>1,2,6,11-18</sup> In this method, a structural ensemble for a receptor is generated only once and many compounds are docked to this same ensemble.

Although a fast method to take conformational flexibility into account, no rigorous theory connects the ensemble of docking scores of a compound to experimental measurements such as binding affinity. As a result, researchers have used the docking scores or docking ranks in a variety of ways to distinguish active compounds from inactive ones.<sup>2,19-22</sup> These methods are not yet satisfactory. One method could work well for one system but not for another, and the performance of classifying compounds into actives and inactives could decrease, rather than increase, when more structures are added to the ensemble.<sup>20,21,23</sup>

Recently,<sup>24</sup> we showed that a machine-learning approach implemented in the form of a naïve Bayesian model could alleviate these problems. The model starts by following two conditional probabilities for gauging the likelihood of a compound to be active or inactive:

$$P(\text{active}|\text{scores}) = P(\text{scores}|\text{active}) \times \frac{P(\text{active})}{P(\text{scores})}$$
$$P(\text{active})/P(\text{scores}) \prod_{i \in \text{conformers}} P(\text{scores}_i \vee \text{active}) \quad \text{Eq. 1}$$

$$P(\text{inactive}|\text{scores}) = P(\text{scores}|\text{inactive}) \times \frac{P(\text{inactive})}{P(\text{scores})}$$
$$P(\text{inactive})/P(\text{scores}) \prod_{i \in \text{conformers}} P(\text{scores}_i \vee \text{inactive}) \quad \text{Eq. 2}$$

where  $scores$  represents an ensemble of docking scores of a compound to an ensemble of structures of the receptor.  $P(scores_i \vee active)$  or  $P(scores_i \vee inactive)$ , the probability of obtaining a score,  $scores_i$ , of an active or an inactive for structure  $i$ , can be obtained by docking a set of known actives or a set of known decoys to the structure. The second equality in Eq. 1 or Eq. 2 results from the common assumption of naïve Bayesian. One can then calculate the log-odds for a compound to be active over inactive by

$$\text{log-odds(active/inactive)} = \log P(\text{active}) - \log P(\text{inactive}) + \sum_{i \in \text{conformers}} \log(P(score_i | \text{active}) / P(score_i | \text{inactive})) \quad \text{Eq. 3}$$

The naïve Bayesian approximation assumes the probability of observing the docking score of a compound to a structure is independent of the probability of observing the docking score to another structure. The probability of observing a set of scores for an active compound or an inactive one to an ensemble of structures is approximated by the product of the probabilities of the scores to individual structures. Although the naïve Bayesian model has already alleviated the problem of ensemble docking discussed above, this paper shows that the performance of classifying compounds can be improved further by relaxing the assumption of independent probabilities. We achieved this by using several machine-learning methods: k nearest neighbors, logistic regression, support vector machine, and random forest, which can ease this approximation more easily in practical applications. We first tested the methods thoroughly on the protein kinase Epidermal Growth Factor Receptor (EGFR). We then further substantiated the key findings by applying the methods to twenty more protein kinases: ABL1, AKT2, CDK2, CSF1R, JAK2, LCK, MAPK2, MET, MK01, MK10, MK14, MP2K1, PLK1, ROCK1, TGFR1, VGFR2, WEE1, BRAF, FGFR1, and IGF1R. We chose these protein kinases not only because of their significance in drug discovery but also because scientists have developed

databases for testing the performance of docking programs on these kinases. In particular, we used the Directory of Useful Decoys, Enhanced (DUD-E)<sup>25</sup>. The developers of this database designed it to include not only compounds that are easy for molecular docking but also difficult ones. In addition, it is easier for scientists to compare the performance of different docking methods or programs by using the same dataset such as DUD-E. The improved performance of the machine-learning models developed here should make them more effective in finding new drug candidates for these protein kinases from new chemical libraries or useful libraries that have not yet been extensively screened.

## Methods

Docking: To test the new machine-learning models, we first used the docking results we obtained earlier for the Epidermal Growth Factor Receptor (EGFR).<sup>24</sup> In this work, we docked 832 active compounds and 35442 decoys to EGFR using AutoDock Vina.<sup>26</sup> These compounds came from the DUD-E dataset<sup>25</sup> designed for evaluating docking programs. As described earlier, we used 34 structures from the Protein Data Bank<sup>27</sup> for docking: entries 2RGP, 2J5F, 3VJO, 4G5J, 4JQ7, 5CAV, 4LI5, 5FED, 1XKK, 3BEL, 4ZJV, 1M14, 2GS2, 4I23, 4TKS, 5CNN, 4RIW, 3IKA, 4LQM, 4LL0, 2JIU, 2EB2, 5XDL, 2EB3, 4I24, 5Y9T, 4LRM, 2ITN, 2JIT, 4G5P, 5FEE, 2ITT, 3UG1, and 2GS7. We chose these structures by performing a sequence search on the Protein Data Bank<sup>27</sup> using the sequence of the wild-type protein. We selected structures with the wild-type sequence first. To add more structures, we also selected structures with single mutations. For these mutants, we modelled the structures back to the wild-type before docking. Details of the docking studies were described in reference <sup>24</sup>. After testing the methods for this protein kinase, we evaluated the key findings with twenty more protein kinases: ABL1, AKT2, CDK2, CSF1R, JAK2, LCK, MAPK2, MET, MK01, MK10, MK14 MP2K1, PLK1, ROCK1, TGFR1, VGFR2, WEE1, BRAF, FGFR1, and IGF1R. As we found that the

performance increased slower after a few structures had been added to the structural ensemble of a protein kinase, we included only eleven structures for these other protein kinases.

Measurement of performance: As in our study with the naïve Bayesian model, we calculated areas under the receiver operating characteristic curves<sup>28-32</sup>. The area under this curve (AUC) ranges from 0 to 1, with 1 giving the best possible model. A model with an AUC of 0.5 only performs as well as a random model does.

Training and test sets: We divided the actives and decoys of the DUD-E dataset<sup>25</sup> into three groups. We took each group in turn as a test set, with the remaining groups as training sets. The variations of AUCs from the three studies gave an idea on the statistical fluctuations of the results.

Feature representation: We represented each compound by a vector containing its docking scores to an ensemble of structures. For example, when all the 34 structures of EGFR were used, this vector had a length of 34. When fewer structures were used, the length of the vector decreased.

k nearest neighbors: In using this method, we first calculated the Euclidian distance between each compound in the test set and each compound in the training set. The compounds were then sorted according to the distance and the top k compounds with the shortest distance were recorded. The percentage of active compounds in these k compounds were then calculated and used to predict whether a compound was likely to be active. A larger percentage implied a higher probability to be active. For EGFR, we varied the value of k to find the k's that gave the best performance as measured by the AUCs. As  $k \approx 50$  gave good performance, we continued

to use this value for all the other protein kinases without individual optimization of this value for each protein kinase.

To reduce the amount of computing time, we also examined the use of smaller subsets of actives and decoys in the training sets. In choosing the subsets, we first performed K-means clustering and used only the compounds closest to the centroids of the clusters in the training sets. We chose K to represent a fraction of the structures in the training sets and we varied K to examine how the performance on classifying compounds varied with K. Such an analysis gives some ideas on how to strike a balance between computational time and performance in practical applications. It also gives insights into how large a training set needs to be to give useful performance.

We implemented the K-means clustering in java. We first randomly selected K compounds  $\{T_{1,\dots,K}\}$  to form K clusters. We then calculated the Euclidean distances between other compounds and each one of these K compounds. A compound was assigned to its closest cluster. The centroid of each cluster was then calculated. A new set  $\{T_{1,\dots,K}\}$  was then constructed from these centroids. Compounds were then re-clustered using the new  $\{T_{1,\dots,K}\}$ . This process was iterated until convergence.

#### Logistic regression:

We used scikit-learn<sup>33</sup> to perform logistic regression. It minimizes the cost function:

$$\frac{1}{2} \dot{w}^T \dot{w} + C \sum_i^n \log \left( \exp \left( -y_i (\dot{S}_i^T \dot{w} + c) \right) \right)$$

with respect to the weights that form components of the column vector  $\dot{w}$ . In our study,  $\dot{S}_i$  was a column vector containing the docking scores of compound  $i$ .  $y_i = 1$  if compound  $i$  was active

and = -1 if inactive.  $n$  was the number of compounds in a training set.  $c$  was the intercept of the logistic regression model.  $C$  controlled the strength of regularization. We used the default value of 1 for  $C$  as it already gave good AUCs.

### Support Vector Machine:

We used the implementation in scikit-learn<sup>33</sup> to perform this task. For EGFR, we first carried out a grid search to find a good combination of the gamma value of the radial basis functions and the C value that controlled to what extent misclassified points were penalized. We performed the grid search with  $C=0.1, 1, 10, 100,$  and  $1000$  and  $\text{gamma}=1, 0.5, 0.45, 0.4, 0.35, 0.3, 0.25, 0.2, 0.15, 0.1, 0.01, 0.001,$  and  $0.0001$ . The results presented were obtained by using  $C=100$  and  $\text{gamma}=0.2$ , which gave better performance. We also applied the “probability” option to give estimates for the probabilities of compounds to be active rather than only classified as actives or inactives. **For the other protein kinases, we continued to use  $C=100$  and  $\text{gamma}=0.2$  without individual optimization for each protein kinase.**

### Random Forest:

We used scikit-learn<sup>33</sup> to perform this task. For EGFR, we varied the parameter `n_estimators`, the number of trees in the forest, from 500 to 2000 but the performance was not sensitive to the choice as can be seen below. **Therefore, we used `n_estimators=750` for all the other protein kinases.**

## **Results and Discussions**

### EGFR

**We studied EGFR more thoroughly by studying the effects of changing various model parameters. We then evaluated the key findings further with the other twenty protein kinases.**

Figure 1 shows the performance of the k nearest-neighbor model as a function of k. The results from the three test sets give an idea on the fluctuation of the results with variation of the training and test sets. The AUCs started with a low value of ~0.6 at small k's but increased rapidly to > 0.8 when k exceeded ~20. All the three test sets reached a top AUC better than 0.86 at their optimal values of k. The optimal values for the three sets ranged from 73 to 229. Reasonable predictive models can be constructed by choosing a value of k within this range, as all the three test sets gave AUCs better than 0.85 in this range. These AUCs were significantly better than the value of 0.69 obtained from our earlier naïve Bayesian model for the same set of docking data for EGFR.<sup>24</sup>

As previous scoring models in ensemble docking showed a decrease, rather than an increase in performance as more structures were added, we repeated the analysis by using only a subset of the structures. Figure 2 shows that the k nearest-neighbor model, for k=100, gave poorer AUCs as structures were removed. Therefore, as the previous naïve Bayesian model showed, machine-learning can avoid the problems that the performance of classifying compounds could decrease with the increase in the number of structures included in the ensemble.

As the computational time increases with the number of compounds in the training sets, and large sets might not be available in the early phase of a project in drug discovery, we examined whether smaller training sets could still give useful performance. To test this, we repeated the calculations of the AUCs by using only a fraction of the compounds in the training set. To obtain a representative subset for each training set, we first performed K-means clustering and chose the compounds closest to the centroids of the clusters to be used in the new smaller training set. If we used only y% of the compounds, we performed K-means clustering in which K=~554 x y% for the actives and K=~23628 x y% for the decoys. (Because

we had ~544 actives and ~23628 decoys in each of the three training sets.) Figure 3 shows that one can still obtain AUCs over 0.80 when only 10 percent of the data were retained. Thus, in a practical application, one can start with a small training set to build a predictive model to guide drug discovery. If more accurate results are desired and as more compounds have been tested experimentally, one can enlarge the training set.

Figure 4 shows the receiver operating characteristic curves for the three test sets obtained by logistic regression and by support vector machine. Again, we obtained good performance in classifying compounds, with AUCs ranging from 0.82 to 0.86 for the three test sets for logistic regression and from 0.85 to 0.87 for support vector machine.

Likewise, random forest gave good results with AUCs ranging from 0.84 to 0.86 (Figure 5). The figure also showed that the results did not change much with the choice of the parameter n\_estimators in the scikit-learn implementation of the method. This parameter determines the number of trees included in the model. On the other hand, in the extreme of using only a single decision tree, we could obtain only low AUCs, under 0.62.

Similar to the k nearest-neighbor model, figure 6 shows that all the other machine-learning models classified compounds better as the ensemble included more structures of the receptor. Thus, our results consistently showed that machine-learning methods could fix the problem that increasing the number of structures in an ensemble could decrease rather than increase performance in ensemble docking.

As including fewer structures reduces computational time in ensemble docking, we also examined which structures might be more useful to be included if one could only afford to use a smaller structural ensemble. To achieve this, we removed one structure at a time from the

feature vector and examined to what extent the AUC would decrease. The five structures showing the largest impact were: 4I24, 4I23, 2J5F, 5FED, and 2ITN (PDB codes) for the k nearest-neighbor model (Figure 7), 1XKK, 2J5F, 4I23, 3UG1, and 4ZJV for the logistic regression model (figure S1), 2J5F, 4LQM, 1XKK, 4LI5, and 4I23 for the support vector machine (figure S2), and 2J5F, 5FED, 4I24, 4I23, and 4LL0 for the random-forest model (figure S3). Thus, 4I23 appeared most important because it appeared in the top five in all the four machine-learning models. 2J5F was next, appearing in three of the four models. 1XKK, 4I24, and 5FED appeared in two of the four models. These results are consistent with the previous results obtained from the naïve Bayesian model<sup>24</sup> when only one structure was used at a time. Four of these structures (4I23, 2J5F, 1XKK, and 4I24) are among the best five structures for both the present knock-one-structure-out analysis and the previous naïve Bayesian model utilizing only one structure. It is not surprising to see slight differences from the two analyses as they were performed differently. In the previous naïve Bayesian model, analysis was performed by including one structure at a time and reflected faithfully the contribution from that structure. On the other hand, the feature representation in the machine-learning models used in the present study made it less suitable to use one structure at a time, in which the feature vector contained only one component. Knocking out one structure at a time did not necessarily produce the same effects as including only one structure in the ensemble. Moreover, the training of machine-learning models did not necessarily produce unique results, as reflected from the different best structures identified by the four different machine-learning models used here.

Nevertheless, the five best structures identified by the different machine-learning models gave similar performance, suggesting that the choice of the best structures was not unique. This can be seen from Table 1 showing the performance, measured by AUCs, when the five best structures identified from the naïve Bayesian model utilizing only one structure or from the knock-one-structure-out model was used. Both showed comparable performance. The k

nearest-neighbor model performed the best with the random forest model having the next best performance. The AUCs reached just a little over 0.70, about 0.15 below the top AUCs obtained by using all the thirty four structures. Thus, if affordable, it is still advantageous to use more structures in the machine-learning enhanced ensemble docking presented in this paper if a higher predictive performance is desired. To further confirm that the best five structures had been identified, Table 1 also shows the results obtained by randomly selecting five structures. The AUCs were lower than those obtained by using all the five best structures identified by the naïve Bayesian model or by the knock-one-structure-out analysis.

To understand why some structures performed better, we analyzed the gaps between the averaged docking scores obtained by the actives and by the decoys (Table S1). The best 5 structures identified above were among the top ten structures that gave the largest gaps of averaged docking scores between the actives and the decoys. That structures giving the largest gaps performed better is also shown by the results in Table 1 when the five structures were selected randomly from only the twenty four structures outside of the top ten. The AUCs significantly decreased. Thus, the gaps of averaged docking scores between the actives and the decoys can provide a useful criterion to help select the best subset of structures for ensemble docking if one wants to reduce computational time on the expense of losing some accuracy.

#### *Twenty other protein kinases*

We performed studies on twenty other protein kinases to further confirm the key findings obtained for EGFR.

The studies of these extra proteins continue to support that our machine-learning models have solved a major defect observed with previous ensemble docking, in which the performance

in classifying compounds has decreased after adding just about three structures.<sup>20,21,23</sup> The results for the twenty protein kinases are shown in figures S4 to S23. They show that the performance of classifying compounds, measured by AUCs, increased as the number of structures in the ensemble was increased except for MK01, MK10, or ROCK1 in which the performance could decrease slightly. For FGFR1, the performance did not increase further after three structures were included, but the performance did not decrease when more structures were added.

The collective results for the twenty one kinases studied here also illustrate how the various machine-learning methods behave differently with respect to the dependence of their performance on the number of structures in the ensemble. The results for decision tree are included in the figures as a reference only. It was not a well-performing machine-learning method for the applications here, giving only low AUCs. Logistic regression was less sensitive to the number of structures included. It gave reasonable AUCs even when just a couple of structures were included, but its performance did not improve as much with increasing number of structures in the ensemble. The performance of random forest, k-nearest neighbor, and support vector machine were more sensitive to the number of structures. Their performance could be low at small number of structures, but increased substantially, and eventually exceeded the performance of logistic regression when more structures were added. Support vector machine performed slightly less well than k-nearest neighbor or random forest did.

Although including more structures increased performance, it also increased the amount of computational time for virtual screening. The results shown above for EGFR suggest that the gap between the averaged docking scores of known actives and inactives can serve as an indicator on which structures are more useful to be included if one can only afford to use a small structural ensemble in a project of large-scale virtual screening. We tested this idea further for

the other twenty protein kinases (Table S2). With four exceptions, including the best five structures yielded better results than using five randomly selected structures. Thus, although not fool-proof, the gap between the averaged docking scores of known actives and inactives can help a researcher to construct a small structural ensemble for less expensive virtual screening if the researcher is satisfied with losing some predictive reliability.

## Conclusions

In a previous perspective article,<sup>24</sup> we suggested that machine learning can improve the use of multiple docking scores from ensemble docking to improve virtual screening. It can fix the long-known problem that the performance of classifying compounds as active or inactive can decrease rather than increase when more structures of the receptor are added to the ensemble. Nevertheless, the naïve Bayesian model we used earlier assumed that the probabilities of observing docking scores to different structures were independent. This approximation might limit its ability to achieve even better performance. In this work, we have used several machine-learning models -- k nearest neighbor, logistic regression, Support Vector Machine, and random forest – to relax this approximation. We first tested these methods by applying them to study the protein kinase EGFR. The new machine-learning models have significantly improved performance over the previous naïve Bayesian model, by ~0.15 when measured by the area-under-receiver-operating-characteristic curve. The new models continued to show that the performance does not decrease with the increase of the number of structures included in the ensemble, a defect in previous models of ensemble docking. Furthermore, we have found that the gaps of the averaged docking scores between the actives and the inactives can provide a useful indicator to identify the structures that were most useful in distinguishing actives from decoys. This provides a useful criterion to help researchers choose a small structural ensemble to reduce computational time in a project of large-scale virtual

screening if they are willing to sacrifice some predictive reliability. Extending this study to twenty more protein kinases has further substantiated these findings.

### **Acknowledgements**

The U.S. National Institutes of Health (CA224033) has provided support for this work. The authors have no conflict of interest to declare.

## **Figure cations**

Figure 1: A plot of the area-under-receiver-operating-characteristics curve (AUC) versus k for the k nearest-neighbor model.

Figure 2: Performance measured by the area-under-receiver-operating-characteristics curve (AUC) increased with the number of structures in the ensemble for the k nearest-neighbor model.

Figure 3: Dependence of the performance of the k nearest-neighbor model on the size of the training set.

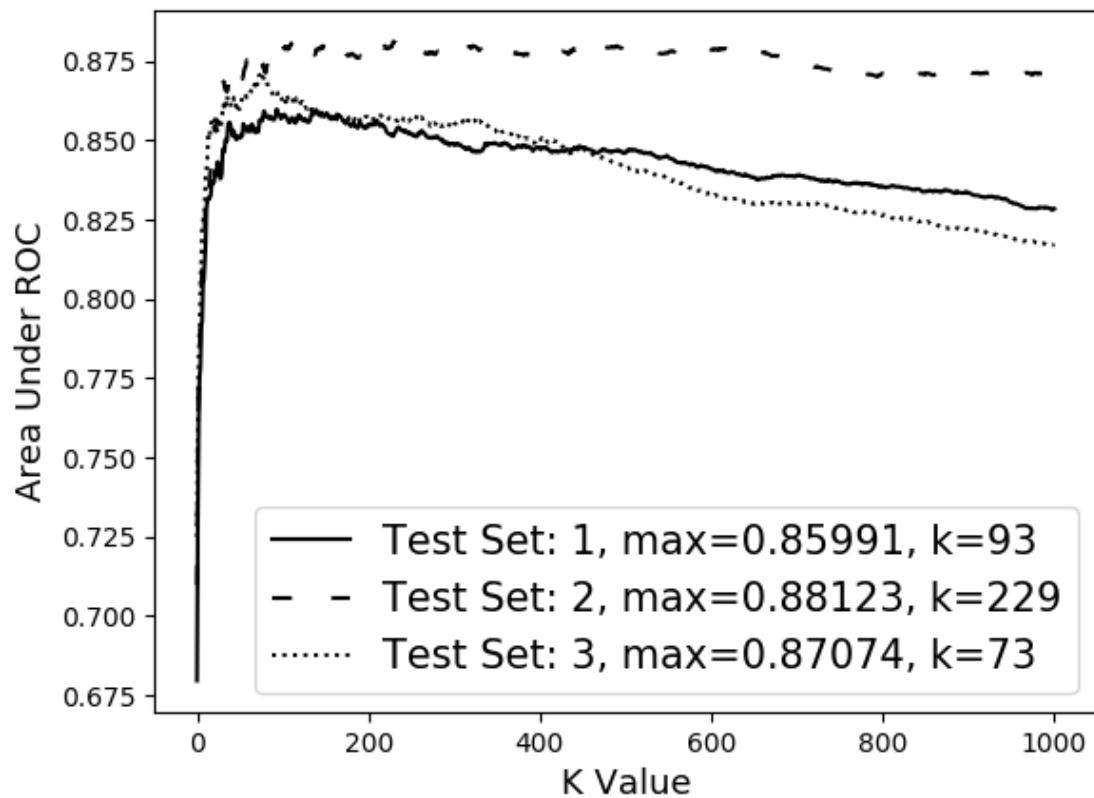
Figure 4: Receiver-operating-characteristics curves for the logistic regression model and the support vector machine. .

Figure 5: A plot of the area-under-receiver-operating-characteristics curve (AUC) versus the number of trees, N Estimators, in the random forest model.

Figure 6: Dependence of the performance of the logistic regression model, the support vector machine, and the random forest model on the number of structures included in the ensemble.

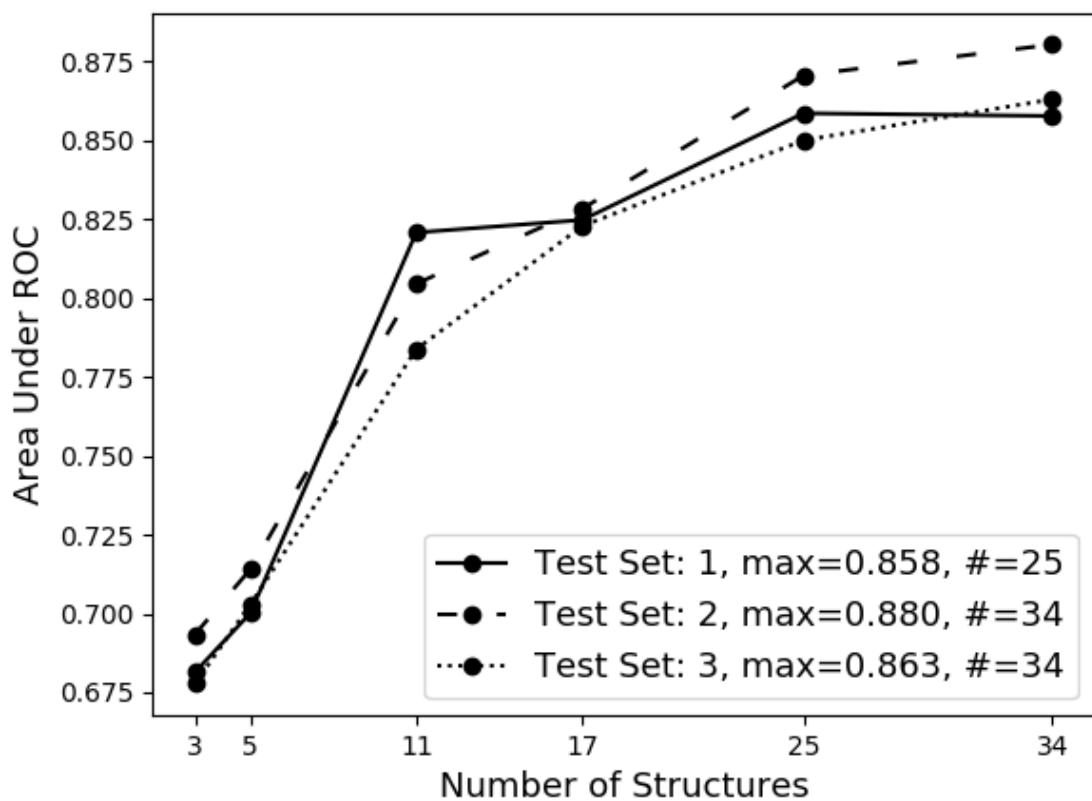
Figure 7: Influence of each structure on the performance of the k nearest neighbor model.

Figure 1



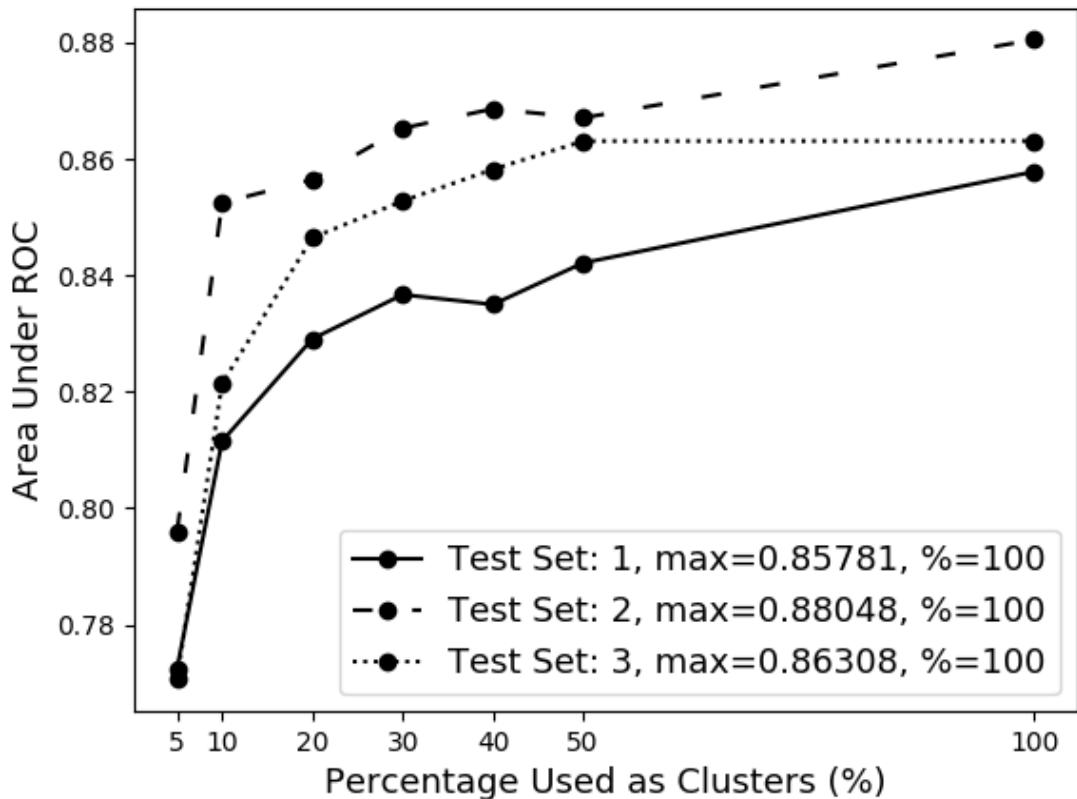
The three curves describe the results from three test sets to show the statistical variations of the results. The optimal AUC for each curve is shown along with the value of k that it occurred.

Figure 2



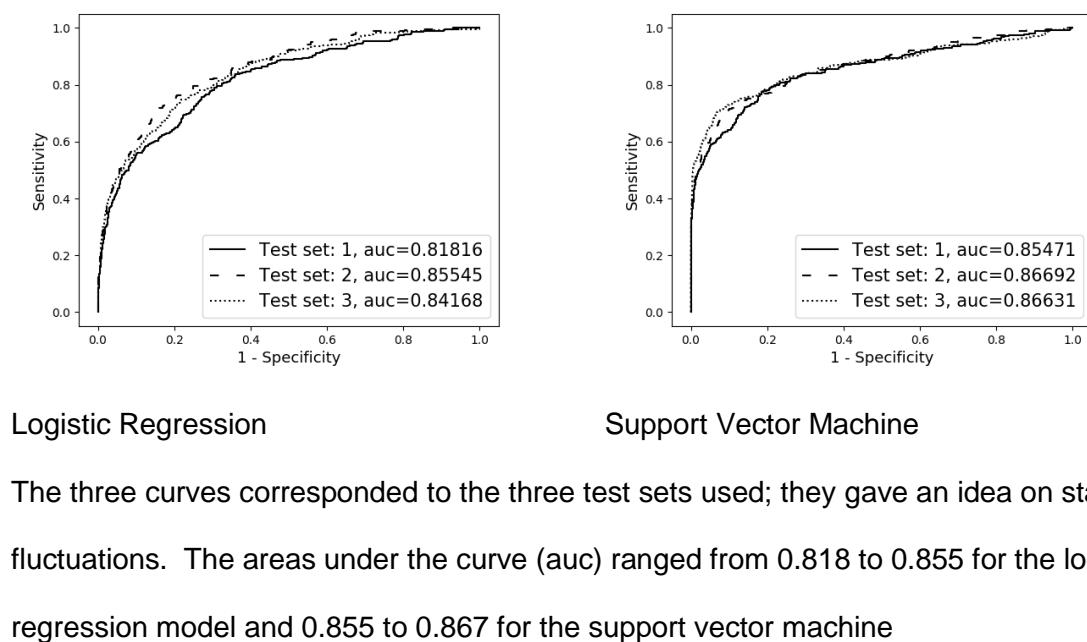
k=100

Figure 3



$k=100$ . Performance measured by the area-under-receiver-operating-characteristic curve. The size was described by the percentage of data in the DUD-E dataset used as training sets, containing ~544 actives and ~23628 decoys.

Figure 4

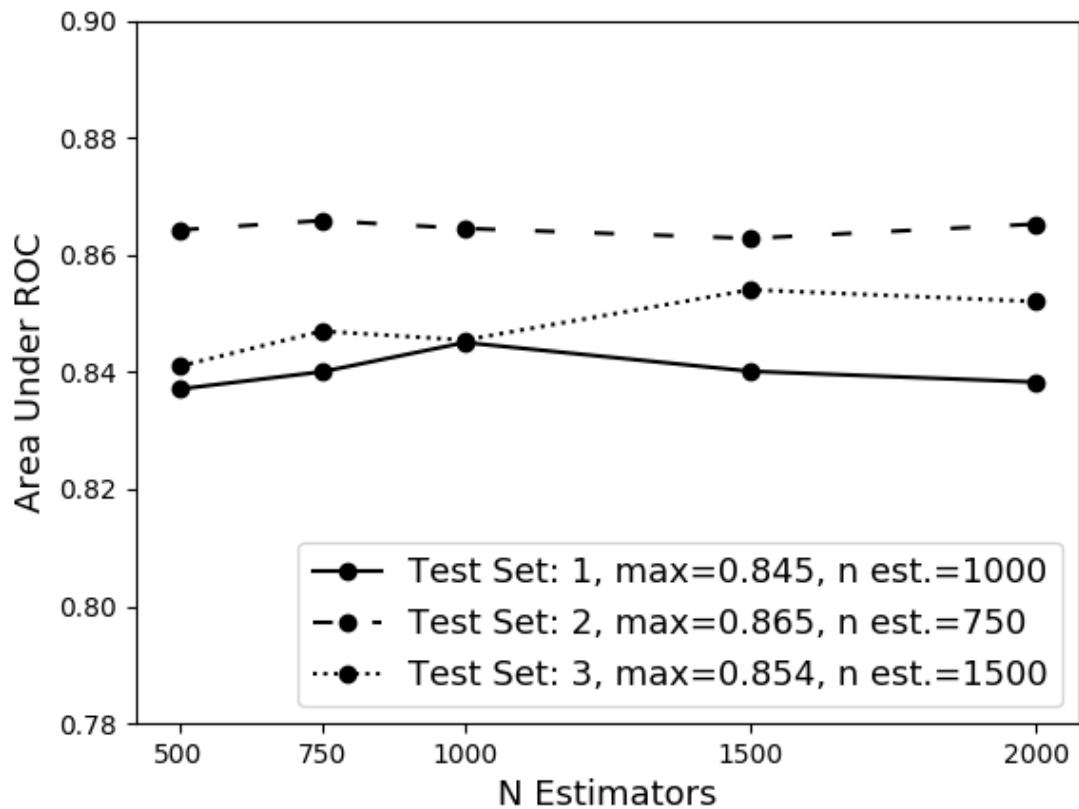


Logistic Regression

Support Vector Machine

The three curves corresponded to the three test sets used; they gave an idea on statistical fluctuations. The areas under the curve (auc) ranged from 0.818 to 0.855 for the logistic regression model and 0.855 to 0.867 for the support vector machine

Figure 5



The three curves describe the results from three test sets to show the statistical variations of the results. The optimal AUC for each curve is shown along with the value of N Estimators at which it occurred.

Figure 6

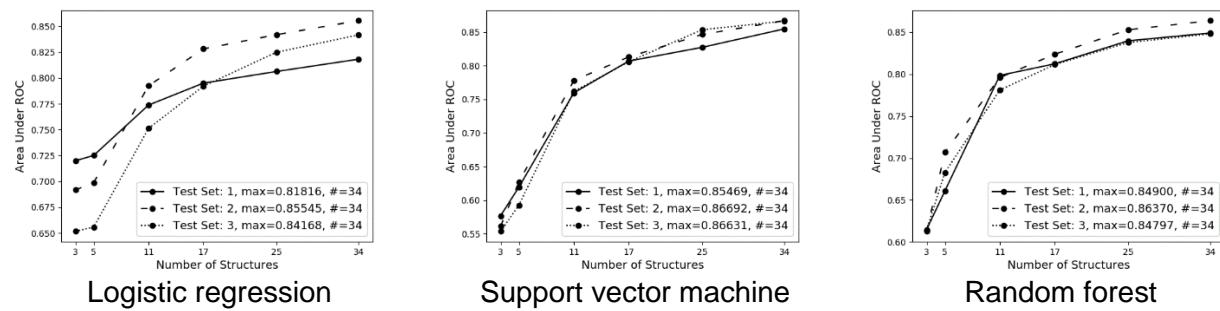
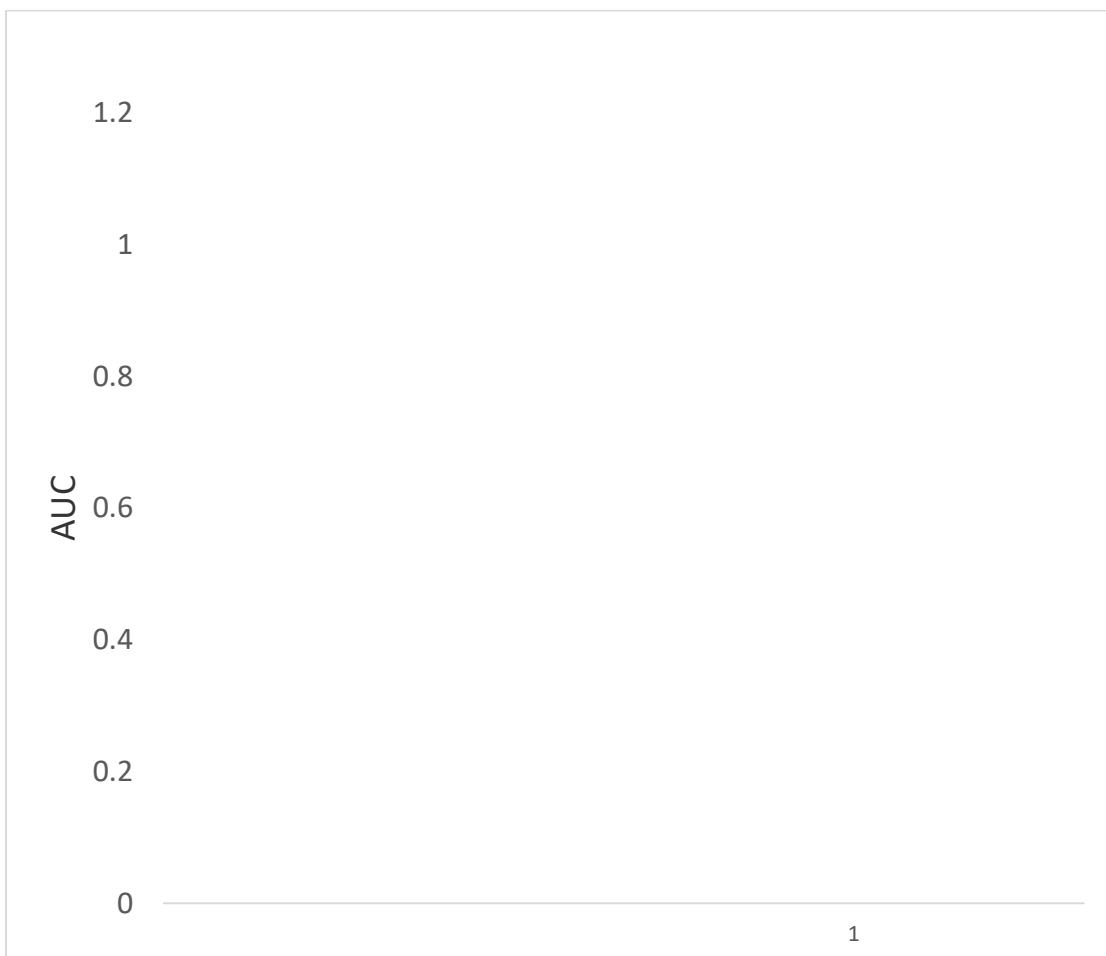


Figure 7



$k=100$ . Codes from the Protein Data Bank<sup>27</sup> are shown on the x-axis.

**Table 1: Area under Receiver Operating Curve (AUC) Obtained by Using Only Five Out of Thirty Four Structures of EGFR**

	Naïve Bayesian <sup>a</sup>	Knock-one-out <sup>b</sup>	Random from 34 <sup>c</sup>	Random from 24 <sup>d</sup>
k nearest neighbors	0.72	0.72	0.68	0.63
Random forest	0.69	0.71	0.66	0.60
Logistic regression	0.67	0.67	0.67	0.59
Support vector machine	0.62	0.63	0.61	0.60

a: Use the 5 structures that gave the best AUC when used individually in the naïve Bayesian model.<sup>24</sup>

b: Use the 5 structures that decreased the AUC the most when each was removed from the 34 structures used in the machine-learning-enhanced ensemble docking, see text for details.

c: use 5 structures randomly selected from the 34 structures of EGFR. This was done six times and the averaged AUCs were reported.

d: use 5 structures randomly selected from the 24 structures of EGFR that showed the least separation of gaps between the docking scores of actives and the docking scores of decoys shown in Table S1.

## References

1. Sørensen J, Demir Ö, Swift RV, Feher VA, Amaro RE. Molecular docking to flexible targets. In. *Methods in Molecular Biology*. Vol 12152015:445-469.
2. Wong CF. Flexible receptor docking for drug discovery. *Expert Opinion on Drug Discovery*. 2015;10(11):1189-1200.
3. Feixas F, Lindert S, Sinko W, McCammon JA. Exploring the role of receptor flexibility in structure-based drug discovery. *Biophys Chem*. 2014;186:31-45.
4. Wong CF, Bairy S. Drug design for protein kinases and phosphatases: Flexible-receptor docking, binding affinity and specificity, and drug-binding kinetics. *Curr Pharm Des*. 2013;19(26):4739-4754.
5. Lexa KW, Carlson HA. Protein flexibility in docking and surface mapping. *Q Rev Biophys*. 2012;45(3):301-343.
6. Antunes DA, Devaurs D, Kavraki LE. Understanding the challenges of protein flexibility in drug design. *Expert Opinion on Drug Discovery*. 2015;10(12):1301-1313.
7. Ivetac A, McCammon JA. Molecular recognition in the case of flexible targets. *Curr Pharm Des*. 2011;17(17):1663-1671.
8. Huang S, Zou X. Advances and challenges in Protein-ligand docking. *International Journal of Molecular Sciences*. 2010;11(8):3016-3034.
9. Wong CF, McCammon JA. Protein Simulation and Drug Design. *Adv Protein Chem*. 2003;66:87-121.
10. Wong CF, McCammon JA. Protein flexibility and computer-aided drug design. In: Cho A, K., Blaschke T, F., Insel P, A., Loh H, H., eds. *Annu. Rev. Pharmacol. Toxicol.* Vol 43. Palo Alto, California: Annual Reviews; 2003:41-46.

11. Wong CF, Kua J, Zhang Y, Straatsma TP, McCammon JA. Molecular docking of balanol to dynamics snapshots of protein kinase A. *Proteins*. 2005;61(4):850-858.
12. Osguthorpe DJ, Sherman W, Hagler AT. Exploring protein flexibility: Incorporating structural ensembles from crystal structures and simulation into virtual screening protocols. *J Phys Chem B*. 2012;116(23):6952-6959.
13. Korb O, Olsson TSG, Bowden SJ, et al. Potential and limitations of ensemble docking. *J Chem Inf Model*. 2012;52(5):1262-1274.
14. Lin JH. Accommodating protein flexibility for structure-based drug design. *Curr Top Med Chem*. 2011;11(2):171-178.
15. Fukunishi Y. Structural ensemble in computational drug screening. *Expert Opinion on Drug Metabolism and Toxicology*. 2010;6(7):835-849.
16. Lin JH, Perryman A, Schames J, McCammon JA. The relaxed complex method: accommodating receptor flexibility for drug design with an improved scoring scheme. *Biopolymers*. 2003;68:47-62.
17. Lorber DM, Shoichet BK. Flexible ligand docking using conformational ensembles. *Protein Sci*. 1998;7(4):938-950.
18. Knegtel RM, Kuntz ID, Oshiro CM. Molecular docking to ensembles of protein structures. *J Mol Biol*. 1997;266(2):424-440.
19. Ellingson SR, Miao Y, Baudry J, Smith JC. Multi-conformer ensemble docking to difficult protein targets. *J Phys Chem B*. 2015;119(3):1026-1034.
20. Bottegoni G, Rocchia W, Rueda M, Abagyan R, Cavalli A. Systematic exploitation of multiple receptor conformations for virtual ligand screening. *PLoS ONE*. 2011;6(5):e18845.
21. Barril X, Morley SD. Unveiling the full potential of flexible receptor docking using multiple crystallographic structures. *J Med Chem*. 2005;48(13):4432-4443.

22. Awuni Y, Mu Y. Reduction of false positives in structure-based virtual screening when receptor plasticity is considered. *Molecules*. 2015;20(3):5152-5164.
23. Abagyan R, Rueda M, Bottegoni G. Recipes for the selection of experimental protein conformations for virtual screening. *J Chem Inf Model*. 2010;50(1):186-193.
24. Wong CF. Improving ensemble docking for drug discovery by machine learning. *Journal of Theoretical and Computational Chemistry*. 2019;18(3).
25. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Directory of useful decoys, enhanced (DUD-E): Better ligands and decoys for better benchmarking. *J Med Chem*. 2012;55(14):6582-6594.
26. Trott O, Olson AJ. Software news and update AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem*. 2010;31(2):455-461.
27. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. *Nucleic Acids Res*. 2000;28:235-242.
28. Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*. 1982;143(1):29-36.
29. Hanley JA, McNeil BJ. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*. 1983;148(3):839-843.
30. Huang Z, Wong CF. Inexpensive Method for Selecting Receptor Structures for Virtual Screening. *J Chem Inf Model*. 2016;56(1):21-34.
31. Huang Z, He Y, Zhang X, et al. Derivatives of salicylic acid as inhibitors of YopH in *Yersinia pestis*. *Chem Biol Drug Des*. 2010;76(2):85-99.
32. Triballeau N, Acher F, Brabet I, Pin JP, Bertrand HO. Virtual screening workflow development guided by the "receiver operating characteristic" curve approach. Application to high-

throughput docking on metabotropic glutamate receptor subtype 4. *J Med Chem.*

2005;48(7):2534-2547.

33. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*. 2011;12:2825-2830.