

RESEARCH

Learning RMSD to improve Protein-Ligand Scoring and Pose Selection

Rishal Aggarwal^{1,2} and David Ryan Koes^{1*}**Abstract**

Docking algorithms are an essential part of the Structure Based Drug Design (SBDD) process as they aim to effectively identify the binding poses of chemical structures at the target site. These algorithms are reliant on scoring functions that evaluate the binding ability of a ligand conformation. Typically, scoring functions are designed to predict the binding affinity of various poses at the target site. In this work, we design a novel approach where the scoring function attempts to predict the Root Mean Square Deviation (RMSD) of a pose to the true binding pose. We show that a convolutional neural network (CNN) can be trained to learn these RMSD values with high correlation between predicted and experimental values. Furthermore we show that this scoring function can improve pose selection performance when used in combination with orthogonal scoring functions like Autodock Vina.

Keywords: Structure Based Drug Design; Docking; Deep Learning; Convolution Neural Networks; Protein-Ligand Scoring

Introduction

Computational drug design plays a critical role in reducing time and cost requirements for drug design and development. One common approach, structure based drug design [1], utilizes 3-D structures and *in silico* methods for predicting if and how compounds bind at an identified receptor target site. A key element of these *in silico* methods are docking algorithms. Such algorithms traverse the conformational search space of a ligand to pinpoint the ideal binding orientation of the chemical structure. Scoring functions are used by docking algorithms to evaluate the binding ability of arbitrary poses at the respective protein target. These functions are used to single out potential active candidates from large chemical libraries (virtual screening). They are also used to rank different orientations of the compound based on their binding scores (pose selection). Therefore, the efficacy of docking algorithms are heavily reliant on the performance on these scoring function.

Previous methods for protein-ligand scoring vary from physics based calculations to statistical approaches. Physically inspired scoring functions [2–4] are modelled through theoretical knowledge, first principles and analysis of experimental data to compute

binding free energies. Knowledge based scoring functions [5–9] aim to derive statistically important interactions by analysing the frequency of favorable atom-pair interaction types from large datasets. Empirical scoring functions [10–15] use terms that are known to be important to binding and are weighted via statistical learning on experimental data. These methods are limited by our knowledge of relevant forces and features involved in molecular bonding.

Machine learning based models, on the other hand, are not necessarily bound by such constraints and have the potential to learn non linear relationships and capture binding features that are hard to model explicitly[16]. This has become an increasingly popular approach to model protein-ligand scoring functions. Random forests, support vector machines, and neural networks are some of the algorithms that have been previously used in this context[10, 17–25]. They may take as input the same descriptors that are used in empirical models, such as electrostatic interactions or molecular interaction fingerprints, which can limit the expressiveness of the model.

Deep learning [26] algorithms are an extension of neural networks that contain multiple hidden layers. These algorithms have been at the frontiers of research in natural language processing [27] and image processing [28]. Advances in deep learning have seen the evolution of architectures that can efficiently analyze images

*Correspondence: dkoes@pitt.edu

¹Department of Computational and Systems Biology, University of Pittsburgh, 3501 Fifth Ave, 15260 Pittsburgh, USA

Full list of author information is available at the end of the article

and graphs. These methods require minimal feature engineering on the inputs to learn their parameters from the data[29]. With the ever expanding availability of 3D structural data[30], adaptation of these architectures for protein-ligand scoring is a natural extension to maximize expressiveness and performance.

Convolutional neural networks (CNNs) [26] are deep learning architectures composed of multiple layers of convolutional filters. These algorithms are used for image recognition, classification, segmentation and various other computer vision tasks. These spatial filters identify essential features in images that provide information specific to the task. The complexity of features identified increase as we go further up the hierarchy of convolutional layers. The availability of 3D structural information enables the usage of CNN architectures for this purpose via voxelized representations. 3D representations allow for implicit spatial relationships to be extracted by the CNN model[31]. Previous works have used this method to predict the binding affinity[32]; classify actives and in-actives[33] as well as to detect protein target sites[34].

Graph convolutional networks (GCNs) [35] are deep learning architectures that are primarily designed to operate on graphs. These use spatial convolutional filters as well to analyze all parts of the graph. A graph is treated as set of node vectors and edges that can easily be represented by vector and adjacency matrices. Convolutional operations are applied to local neighborhoods of each node at every level of the network to learn suitable weights for the task. These algorithms have been previously applied on molecular graphs for learning molecular representations[36, 37], structure based virtual screening[38] and for classification of protein structures[39, 40]. While GCNs treat pairwise interactions explicitly, they are typically not able to model all spatial relationships that can exist in the structure.

Most protein-ligand scoring functions until now aim to evaluate ligand orientations by either predicting the binding affinity or discriminating between active and inactive poses. In this paper we explore a novel approach of learning the Root Mean Square Distance (RMSD) of arbitrary poses to the true binding pose. We call this scoring function RMSDScore. We use a CNN for this purpose as it enables the calculation of spatial relationships that would be required for such a scoring function. Our initial hypothesis was that such a non-physical scoring function should have a smoother energy landscape that will enable more effective convergence and sampling during docking. In investigating this hypothesis, we provide the following contributions:

- We train a convolutional neural network to predict the RMSD of random poses to the true binding pose using fully differentiable atomic grids[41] and a process of iterative training. We attain a significant correlation between predicted and experimental values indicating that the RMSD is a learnable value.
- We show that this scoring function can be used to drive pose sampling and docking to get good poses. Furthermore, we show that this function can also be used for optimization of ligand conformations. However, we do not observe that the scoring function results in a smoother, less frustrated energy landscape.
- Nonetheless, we show that this scoring function can be used to effectively perform pose selection when applied in combination with an orthogonal scoring function like Autodock Vina[15].

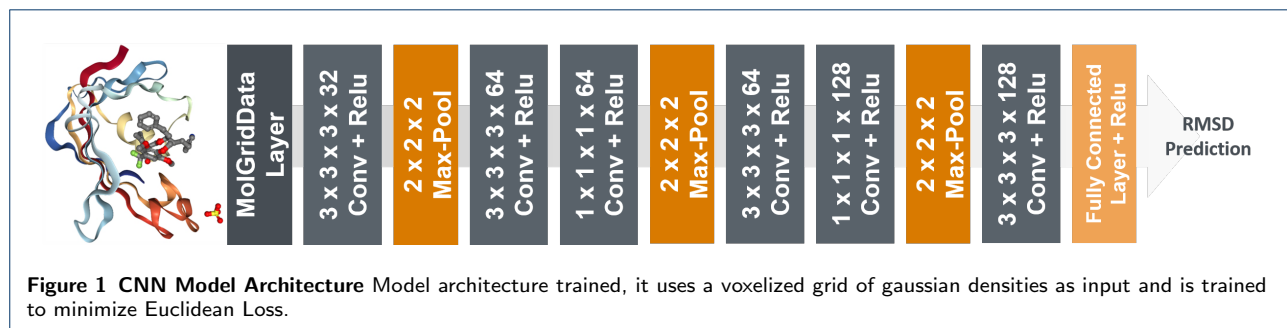
Methods

Data and Representation

The PDBbind database[42] contains a comprehensive collection of binding affinities for protein-ligand co-crystal structures present in the Protein Data Bank. It thus provides an high quality list of complexes that are present in PDB for analysis. The PDBbind database contains a subset of structures of better quality known as the refined set. This subset only contains structures that have an overall resolution better than or equal to 2.5Å and contain only non-covalently bonded complexes. We use the refined subset of PDBbind 2016 for our experiments.

For generating the dataset we redocked ligands of the PDBBind refined set using the Vina scoring function and the smina [10] tool. In addition, locally optimized crystal poses were added to the dataset to ensure at least one good pose (poses with an RMSD less than 2Å from the crystal pose) per complex was present in the dataset. Poses were labelled using the RMSD from the crystal pose using the `obrms` module in Openbabel[43]. To avoid over-optimistic results we performed clustered cross-validation [44] by clustering the complexes into 3 folds based on protein similarity. This was done so that the test sets did not contain targets similar to those in the training set.

The molecules were set onto a cubic grid for input. We used a 0.5Å resolution and 24Å edge length resulting in a cube of size 48x48x48. Atom types from smina were used to represent the different types of heavy atoms present in the complex (e.g., oxygens and nitrogens are further classified based on hydrogen bonding propensity and carbons are classified based on aromaticity). These were taken as the fourth dimension, analogous to color channels in an image, of



the atomic grid. We used the MolgridDataLayer described in [33, 41] for the atomic gridding. This layer also has the ability to do data transformations like rotations and translations on the fly during training. The atomic density is represented on the grid $g(d, r)$ using the following equation:

$$g(d, r) = \begin{cases} e^{-\frac{2d^2}{r^2}} & 0 \leq d < r \\ \frac{4}{e^2 r^2} d^2 - \frac{12}{e^2 r} d + \frac{9}{e^2} & r \leq d < 1.5r \\ 0 & d \geq 1.5r \end{cases} \quad (1)$$

Here d represents the distance of the grid point from the center of the atom and r is the radius of the atom. This provides a continuous density distribution of the atom from the atom center to 1.5 times its radius.

Model Training

The CNN model used is shown in Figure 1. Our custom fork of Caffe [45] was used to train the model. Nesterov Accelerated Gradient Descent [46] was used as the optimizer with learning rate = 0.01, momentum = 0.9 and weight decay of 0.001. Batches of 50 samples were used at a time and the euclidean loss between predicted and true RMSD values were minimized. MolgridDataLayer was used for on-the-fly-data augmentation through application of random rotations and translations.

Minimization

With the usage of atomic grid, the gradients accumulated from backpropagation can be propagated further to the contribution of each individual atomic location. These gradients can thus be used to optimize the molecule accordingly to reach a local optima of the scoring function. The Broyden Fletcher Goldfarb Shanno algorithm (BFGS) algorithm, a multivariate optimization technique, as implemented in gmina, was used for optimization of the poses. The partial derivation of the scoring function f with respect to each individual atom can be easily calculated using the chain

rule in the following manner:

$$\frac{\partial f}{\partial \mathbf{a}} = \sum_{g \in \mathbf{G}_a} \frac{\partial f}{\partial g} \frac{\partial g}{\partial d} \frac{\partial d}{\partial \mathbf{a}} \quad (2)$$

Here g refers to the atomic grid of a single atom type and \mathbf{G}_a refers to the set of grids for all atom types. The first part of the chain rule is calculated via backpropagation and the remaining can easily be found using the density formula. (1).

Iterative Training

Since all the poses in the dataset were generated by docking using smina, it contained only viable poses that had no steric hindrance or instability. This bias in the dataset leads to bad performance in optimization and generation of poses by the CNN model. Therefore, we employed a strategy of iterative training[47] to sensitize the model to poses that are not typical of the results of docking, such as those with high steric hindrance. This is done by adding counterexample poses optimized by the model whose correct labels differ significantly from their predicted labels to the training set and retraining the model. We repeated this iterative procedure twice to get the final model. In the last iteration stratified sampling was done over a range of RMSD values to feed a uniform distribution of poses at different distances from the crystal pose to the model.

Results and Discussions

We evaluated the performance of the final RMSDScore model after iterative training in pose generation, pose selection and ligand optimization. The performance was compared to the Autodock Vina baseline and a CNN pose classification model (CNNScore)[31] that was trained to classify between low RMSD ($<2\text{\AA}$) and high RMSD ($>2\text{\AA}$) poses using a cross entropy loss. As with the RMSD model, the CNNScore model was trained with the iterations of counterexample generation. Evaluations are performed by merging the three test folds from clustered cross validation.

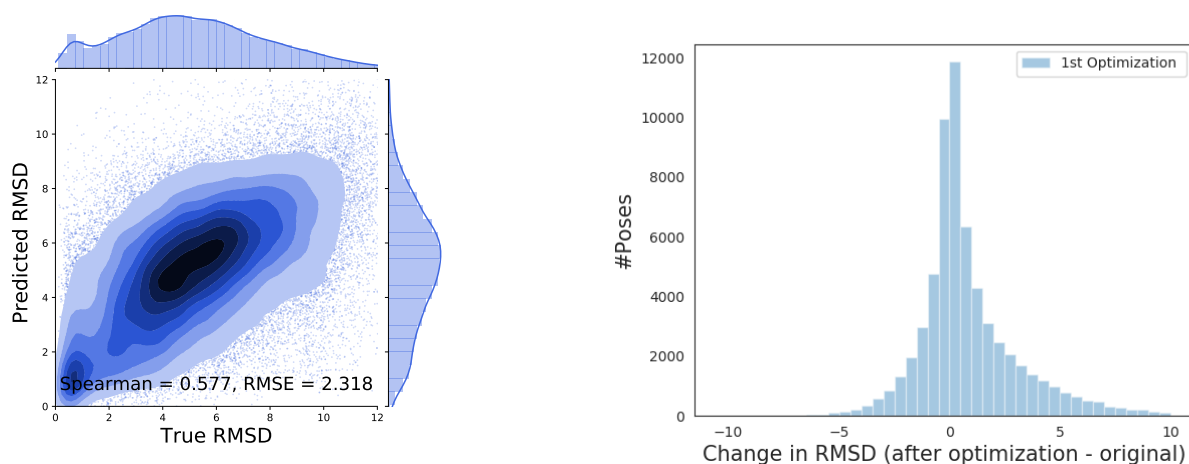


Figure 2 Initial Results test set results(left) and minimization(right)

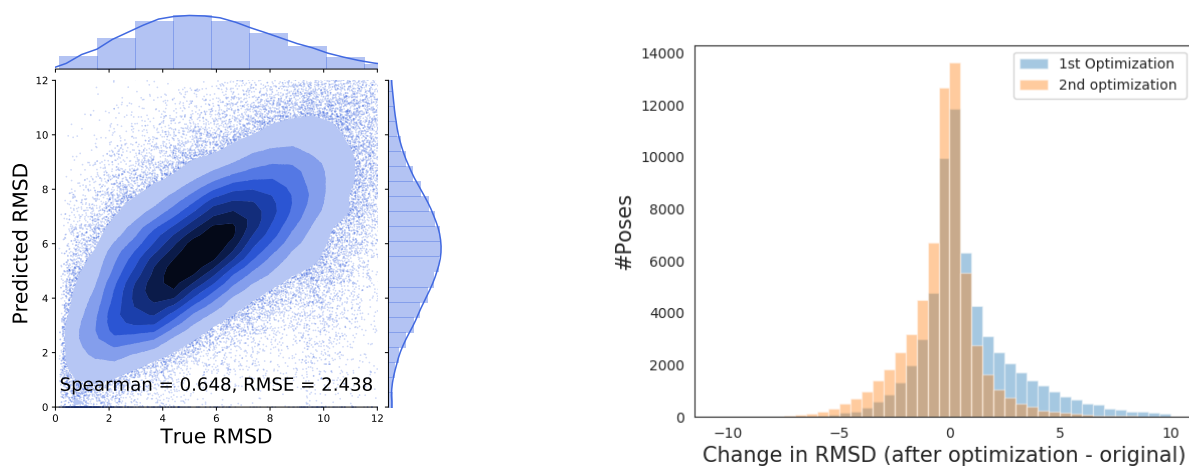


Figure 3 1st iteration results test set results(left) and minimization(right)

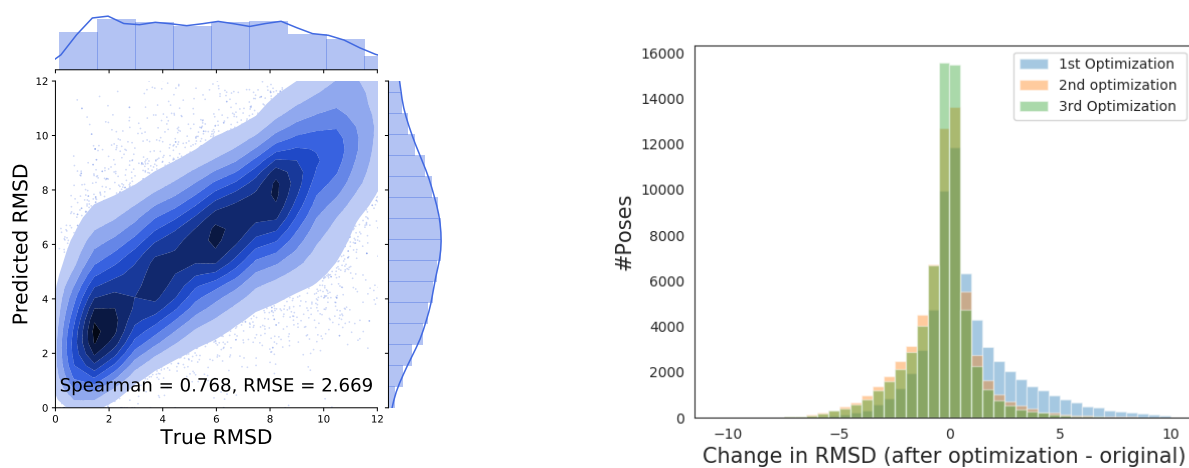


Figure 4 2nd iteration results test set results(left) and minimization(right)

Training Results

Three iterations of model training and subsequent minimization of docked poses were performed to get the final model for docking and pose selection. Figures 2,3,4 show the results of training and minimization of poses at each step of iterative training. Each iteration expands the training set with ligand poses generated using the previously trained model, resulting in different distributions of RMSD values. The histograms display the change in RMSD with respect to the true binding pose after optimizing a pose in the training set using the current iteration's model. A change greater than zero indicates that the optimized pose is farther from the correct binding pose (i.e., 'optimizing' the pose actually made it worse). These optimized, yet far from optimal, counterexamples are added to the training set for the next iteration. As more counterexamples are added, the distribution of the RMSD change shifts left, as is desired, although the average change does not go below zero (Table 1). Each iteration also results in an improvement in correlation between predicted and true values at the expense of worsening absolute error. The inclusion of stratified sampling when training the final model results in a Spearman correlation of 0.768 which indicates that RMSD is a learn-able value that could potentially be used for protein-ligand scoring and pose optimization.

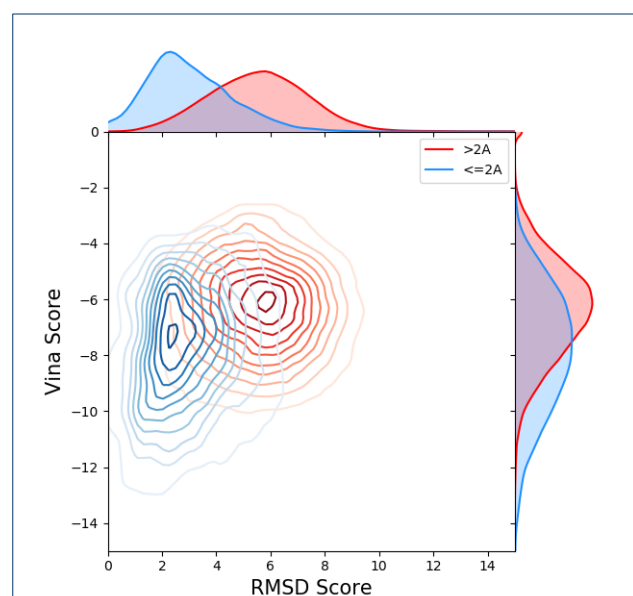


Figure 5 Score Distributions Distributions of RMSD and Vina scores shown as isolines for good (blue) and bad (red) poses

The isoline plot in Figure 5 displays the distributions of RMSD and Vina scores of poses in the test set of all folds. We can see from the marginal histograms that the RMSDScore provides a better separation between

good poses and bad poses as compared to the Vina scores. This is an indication that the RMSDScore has a potential to improve the pose scoring performance over the Vina baseline.

Pose Optimization

Here we evaluate the ability of the final RMSD trained model to locally optimize poses compared to the previously described CNNScore model. Docked poses for the complexes in the test set were optimized using the BFGS algorithm on the accumulated atomic grid gradients. The change in RMSD after optimization for poses that were initially within 4 Å of the crystal structure pose has been plotted in the Figure 6. The performance of both CNNScore and RMSDScore in minimization can be compared in the figure. Counter to our initial hypothesis that an RMSD trained model would provide an energy landscape more amenable to pose optimization, the RMSD model does not do better than the CNNScore model. Additionally, both models exhibit a strong preference for generating only a small change in model quality, suggesting the optimization process quickly converges to a local optima. The mean and standard deviation of change in RMSD from the true pose is reported in Table 1. For both models, the variance in the change in pose quality increases as the RMSD of the initial pose increases, indicating that less ideal poses are subject to more movement. However, neither method results in an average improvement in pose.

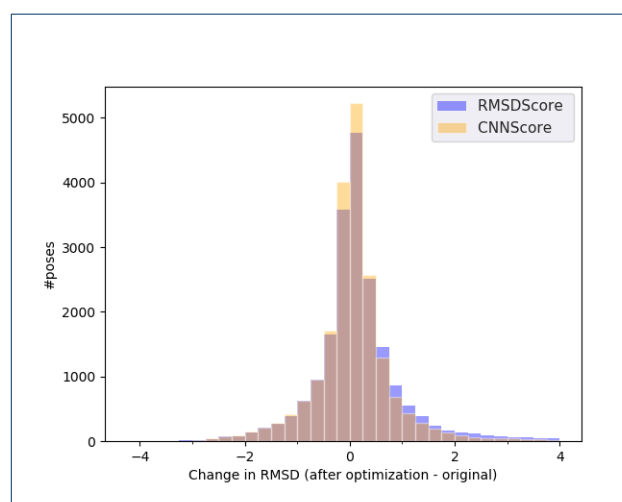


Figure 6 Minimization Results Change of RMSD to the true pose after optimizing poses that are within 4Å of the true pose using both the RMSDScore and CNNScore models.

Pose Generation

Here we evaluate the ability of the final RMSD trained model to globally optimize poses compared to the pre-

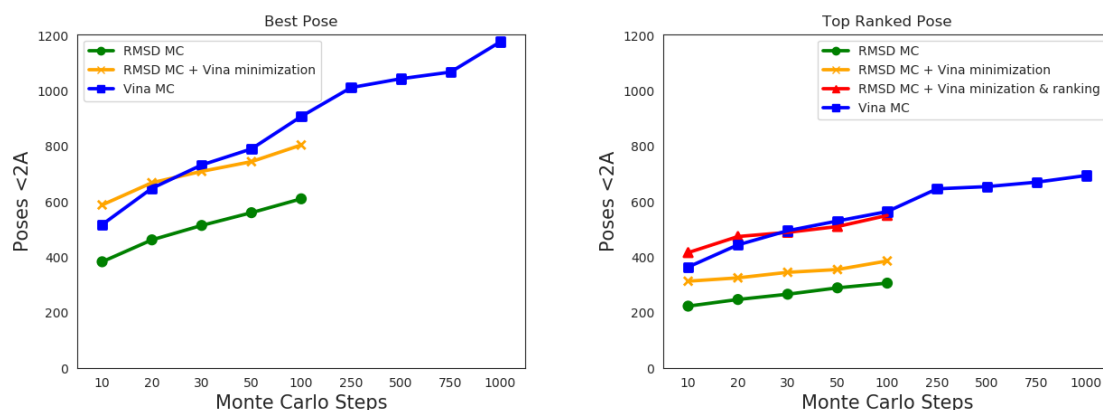


Figure 7 Pose Generation Results Number of Best Poses(left) and Top Ranked Poses(right) below 2 Angstrom from crystal binding pose

Table 1 Optimization Results. Mean and standard deviation of change in RMSD after optimization

	CNNScore	RMSDScore
All Poses	0.09±0.94	0.22±1.07
Initial Pose <2Å	0.27±0.60	0.45±0.86
Initial Pose between 2Å and 4Å	0.02±1.03	0.14±1.13

viously described CNNScore model. Global optimization is performed using the iterated local search approach, as implemented in gnina, which is based off of Autodock Vina [15]. This algorithm combines Monte Carlo sampling steps with local optimization to reach the global minimum. At each step, random perturbations are performed along the degrees of freedom of the molecule pose. This is followed by local optimization using BFGS. The newly generated pose is accepted if it satisfies the Metropolis criteria. The new orientation always satisfies the criteria if it has a better score as compared to the previous pose. If it has a worse score, it is accepted with a probability that is dependant on the difference of scores between the two poses, with smaller differences having higher probabilities.

We evaluated the effectiveness of RMSDScore at pose generation on a single test fold. We redocked compounds to their cognate protein structure using a varying number of Monte Carlo steps. Figure 7 shows the result of pose generation for 1205 complexes present in the test set. We plot both the trend for sampling low RMSD (< 2Å) poses and for correctly ranking these poses as the amount of Monte Carlo sampling is increased. In all schemes we see that the number of low RMSD poses sampled increases with the amount of sampling, This shows that the RMSD scoring function can be used to drive pose sampling, but using RMSD-Score exclusively demonstrates the worst overall performance. If RMSDScore is used to guide the Monte

Carlo sampling with Vina used for the final pose refinement then a small improvement over Vina is initially observed. However, after a certain number of steps the Vina performance still remains the best. Furthermore, there is a high time requirement to dock poses with CNN models due to multiple forward and backwards passes during optimization. This makes it infeasible to carry out large numbers of sampling steps using the model. Thus, values only up to 100 Monte Carlo steps are reported for the RMSD scoring function.

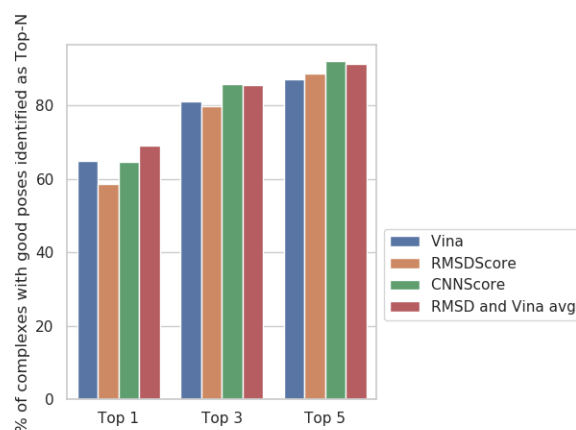


Figure 8 Pose Ranking The percentage of complexes with good poses that have a binding pose ranked as Top-1, Top-3 and Top-5 by different scoring methods

Pose Selection

Finally, since RMSDScore does not outperform Vina at pose optimization or sampling, we evaluate its ability

to correctly select low RMSD poses from Vina generated docked poses. Vina docked poses from the test set of all the models were ranked and the pose selection performance was compared. In addition, poses were also ranked using the average of Vina and RMSDScore. Figure 8 reports the results of pose ranking and selection results. The CNNScore function is the best at ranking top-3 and top-5 with approximately a 4-5% improvement over the Vina baseline. The RMSDScore function is worse than Vina at ranking the top pose but catches up to Vina for top-3 and outperforms it in ranking top-5. The average of the two scores, however, is much better at top pose selection with a 4-5% improvement over the Vina baseline. This shows the potential that RMSDScore has to improve orthogonal scoring functions like Autodock Vina.

Conclusions

In this work, we describe a novel approach for protein-ligand scoring using the RMSD of arbitrary poses from the true binding pose. We trained models to predict this RMSD with a correlation coefficient of 0.768 on the test set, demonstrating that RMSD is a learnable property of a pose. These models provide a greater separation between the distribution of low RMSD and high RMSD poses relative to Autodock Vina. We show that this scoring function can be used for pose optimization and pose generation, but is subject to time constraints and does not perform better than Vina at these tasks. However, when the RMSDScore function is used in combination with Autodock Vina it improves top pose selection performance.

Thus we can conclude that models can be trained to predict RMSD and have the potential to improve protein-ligand scoring as indicated by Figure 5. However, they are not yet useful enough to guide pose ranking except when used in tandem with another independent scoring function.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work is supported by R01GM108340 from the National Institute of General Medical Sciences, by the University of Pittsburgh Center for Research Computing through the resources provided, and by a GPU donation from the NVIDIA corporation.

Author details

¹Department of Computational and Systems Biology, University of Pittsburgh, 3501 Fifth Ave, 15260 Pittsburgh, USA. ²Birla Institute of Technology and Science, Pilani, Vidya Vihar, 333031 Pilani, India.

References

- Lionta, E., Spyrou, G., K Vassiliadis, D., Cournia, Z.: Structure-based virtual screening for drug discovery: principles, applications and recent advances. *Current topics in medicinal chemistry* **14**(16), 1923–1938 (2014)
- Harder, E., Damm, W., Maple, J., Wu, C., Reboul, M., Xiang, J.Y., Wang, L., Lupyan, D., Dahlgren, M.K., Knight, J.L., et al.: Opls3: a force field providing broad coverage of drug-like small molecules and proteins. *Journal of chemical theory and computation* **12**(1), 281–296 (2016)
- Yin, S., Biedermannova, L., Vondrasek, J., Dokholyan, N.V.: Medusacore: an accurate force field-based scoring function for virtual drug screening. *Journal of chemical information and modeling* **48**(8), 1656–1662 (2008)
- Case, D.A., Cheatham III, T.E., Darden, T., Gohlke, H., Luo, R., Merz Jr, K.M., Onufriev, A., Simmerling, C., Wang, B., Woods, R.J.: The amber biomolecular simulation programs. *Journal of computational chemistry* **26**(16), 1668–1688 (2005)
- Huang, S.-Y., Zou, X.: Mean-force scoring functions for protein–ligand binding. In: *Annual Reports in Computational Chemistry* vol. 6, pp. 280–296. Elsevier, ??? (2010)
- Muegge, I., Martin, Y.C.: A general and fast scoring function for protein–ligand interactions: a simplified potential approach. *Journal of medicinal chemistry* **42**(5), 791–804 (1999)
- Gohlke, H., Hendlich, M., Klebe, G.: Knowledge-based scoring function to predict protein–ligand interactions. *Journal of molecular biology* **295**(2), 337–356 (2000)
- Zhou, H., Skolnick, J.: Goap: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction. *Biophysical journal* **101**(8), 2043–2052 (2011)
- Huang, S.-Y., Zou, X.: An iterative knowledge-based scoring function to predict protein–ligand interactions: II. validation of the scoring function. *Journal of computational chemistry* **27**(15), 1876–1882 (2006)
- Koes, D.R., Baumgartner, M.P., Camacho, C.J.: Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling* **53**(8), 1893–1904 (2013)
- Eldridge, M.D., Murray, C.W., Auton, T.R., Paolini, G.V., Mee, R.P.: Empirical scoring functions: I. the development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of computer-aided molecular design* **11**(5), 425–445 (1997)
- Böhm, H.-J.: The development of a simple empirical scoring function to estimate the binding constant for a protein–ligand complex of known three-dimensional structure. *Journal of computer-aided molecular design* **8**(3), 243–256 (1994)
- Wang, R., Lai, L., Wang, S.: Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of computer-aided molecular design* **16**(1), 11–26 (2002)
- Korb, O., Stutzle, T., Exner, T.E.: Empirical scoring functions for advanced protein–ligand docking with plants. *Journal of chemical information and modeling* **49**(1), 84–96 (2009)
- Trott, O., Olson, A.J.: Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry* **31**(2), 455–461 (2010)
- Li, H., Leung, K.-S., Wong, M.-H., Ballester, P.J.: The importance of the regression model in the structure-based prediction of protein–ligand binding. *Intelligence Methods for Bioinformatics and Biostatistics*, 219 (2015)
- Ballester, P.J., Mitchell, J.B.: A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics* **26**(9), 1169–1175 (2010)
- Ashtawy, H.M., Mahapatra, N.R.: Machine-learning scoring functions for identifying native poses of ligands docked to known and novel proteins. *BMC bioinformatics* **16**(S6), 3 (2015)
- Sato, T., Honma, T., Yokoyama, S.: Combining machine learning and pharmacophore-based interaction fingerprint for in silico screening. *Journal of chemical information and modeling* **50**(1), 170–185 (2010)
- Zilian, D., Sottriffer, C.A.: Sfcscoring: a random forest-based scoring function for improved affinity prediction of protein–ligand complexes. *Journal of chemical information and modeling* **53**(8), 1923–1933 (2013)
- Jorissen, R.N., Gilson, M.K.: Virtual screening of molecular databases

- using a support vector machine. *Journal of chemical information and modeling* **45**(3), 549–561 (2005)
22. Deng, W., Breneman, C., Embrechts, M.J.: Predicting protein–ligand binding affinities using novel geometrical descriptors and machine-learning methods. *Journal of chemical information and computer sciences* **44**(2), 699–703 (2004)
23. Durrant, J.D., McCammon, J.A.: Nnscore: a neural-network-based scoring function for the characterization of protein–ligand complexes. *Journal of chemical information and modeling* **50**(10), 1865–1871 (2010)
24. Chupakhin, V., Marcou, G., Baskin, I., Varnek, A., Rognan, D.: Predicting ligand binding modes from neural networks trained on protein–ligand interaction fingerprints. *Journal of chemical information and modeling* **53**(4), 763–772 (2013)
25. Durrant, J.D., McCammon, J.A.: Nnscore 2.0: a neural-network receptor–ligand scoring function. *Journal of chemical information and modeling* **51**(11), 2897–2903 (2011)
26. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *nature* **521**(7553), 436–444 (2015)
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, pp. 5998–6008 (2017)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
29. Sunseri, J., King, J.E., Francoeur, P.G., Koes, D.R.: Convolutional neural network scoring and minimization in the d3r 2017 community challenge. *Journal of computer-aided molecular design* **33**(1), 19–34 (2019)
30. Berman, H.M., Bourne, P.E., Westbrook, J., Zardecki, C.: The protein data bank. In: *Protein Structure*, pp. 394–410. CRC Press, ??? (2003)
31. Hochuli, J., Helbling, A., Skaist, T., Ragoza, M., Koes, D.R.: Visualizing convolutional neural network protein–ligand scoring. *Journal of Molecular Graphics and Modelling* **84**, 96–108 (2018)
32. Jiménez, J., Skalic, M., Martínez-Rosell, G., De Fabritiis, G.: K deep: Protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling* **58**(2), 287–296 (2018)
33. Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J., Koes, D.R.: Protein–ligand scoring with convolutional neural networks. *Journal of chemical information and modeling* **57**(4), 942–957 (2017)
34. Pu, L., Govindaraj, R.G., Lemoine, J.M., Wu, H.-C., Brylinski, M.: Deepdrug3d: Classification of ligand-binding pockets in proteins with a convolutional neural network. *PLoS computational biology* **15**(2), 1006718 (2019)
35. Bacciu, D., Errica, F., Micheli, A., Podda, M.: A gentle introduction to deep learning for graphs. *arXiv preprint arXiv:1912.12693* (2019)
36. Kearnes, S., McCloskey, K., Berndl, M., Pande, V., Riley, P.: Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design* **30**(8), 595–608 (2016)
37. Duvenaud, D.K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., Adams, R.P.: Convolutional networks on graphs for learning molecular fingerprints. In: *Advances in Neural Information Processing Systems*, pp. 2224–2232 (2015)
38. Torng, W., Altman, R.B.: Graph convolutional neural networks for predicting drug–target interactions. *Journal of Chemical Information and Modeling* **59**(10), 4131–4149 (2019)
39. Zamora-Resendiz, R., Crivelli, S.: Structural learning of proteins using graph convolutional neural networks. *bioRxiv*, 610444 (2019)
40. Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks. In: *Advances in Neural Information Processing Systems*, pp. 6530–6539 (2017)
41. Sunseri, J., Koes, D.R.: libmolgrid: Gpu accelerated molecular gridding for deep learning applications. *arXiv preprint arXiv:1912.04822* (2019)
42. Wang, R., Fang, X., Lu, Y., Yang, C.-Y., Wang, S.: The pdbbind database: methodologies and updates. *Journal of medicinal chemistry* **48**(12), 4111–4119 (2005)
43. O’Boyle, N.M., Banck, M., James, C.A., Morley, C., Vandermeersch, T., Hutchison, G.R.: Open babel: An open chemical toolbox. *Journal of cheminformatics* **3**(1), 33 (2011)
44. Kramer, C., Gedeck, P.: Leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets. *Journal of chemical information and modeling* **50**(11), 1961–1969 (2010)
45. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 675–678 (2014)
46. Dozat, T.: Incorporating nesterov momentum into adam (2016)
47. Ragoza, M., Turner, L., Koes, D.R.: Ligand pose optimization with atomic grid-based convolutional neural networks. *arXiv preprint arXiv:1710.07400* (2017)