PLANTAIN: Using an AI pose scoring function for fast and accurate molecular docking

Anonymous Authors[1]

Abstract

Molecular docking aims to predict the 3D pose of a small molecule in a protein binding site. Traditional docking methods predict ligand poses by minimizing a physics-inspired scoring function. Recently, a diffusion model has been proposed that iteratively refines a ligand pose. We combine these two approaches by training a pose scoring function in a diffusion-inspired manner. In our method, PLANTAIN, a neural network is used to develop a very fast pose scoring function. When docking, we parameterize the PLANTAIN scoring function on the fly and use BFGS minimization to optimize an initially random ligand pose. Using rigorous benchmarking practices, we demonstrate that our method achieves comparable accuracy to the state-of-the-art method while running 6 times faster. We release PLANTAIN publicly and hope that it improves the utility of virtual screening workflows.

1. Introduction

Proteins play many vital roles in the human body, and their functions can be modified by small molecules that bind to them. Designing such ligands for protein targets is an essential task in drug discovery. The 3D pose of a ligand in a protein binding site is critical for the compound's binding affinity, and thus its functional effect. Molecular docking aims to predict this pose for a given ligand and protein binding site. Docking is commonly used in structure-based virtual screening, wherein a large library of compounds are docked to a protein structure of interest. Their poses are used to score their binding affinity and the highest-scoring compounds are selected for experimental testing. Virtual screening promises to quickly discover binders for a target, but its utility is currently limited due to the low accuracy of docking algorithms.

1.1. Prior Work

Traditionally, docking has been framed as a problem of energy minimization (Trott & Olson, 2010; Friesner et al., 2004). The ground-truth pose is considered to be the pose that minimizes the potential energy of the protein-ligand complex. This energy is computationally intractable due to its quantum mechanical nature; thus, many scoring functions have been developed to approximate it. Unfortunately, these physics-based approaches need to make many assumptions in order to run quickly. Notably, they generally assume that the protein residues are inflexible, and only modify the ligand pose. This means the scoring functions are often very sensitive to the specific conformation of the given protein structure.

Recently, several methods have been proposed that use machine learning (ML) to directly predict ligand poses (Stärk et al., 2022; Lu et al., 2022). Notably, DiffDock (Corso et al., 2022) is a diffusion model over ligand poses. To predict a pose, DiffDock initializes the ligand in a random conformation and iteratively predicts updates to the translation, rotation, and torsional angles.

One disadvantage of these ML approaches is that they focus on the problem of whole-protein (or "blind") docking, implicitly combining the tasks of binding site prediction and ligand pose prediction. In practice, whole-protein docking makes the problem unnecessarily difficult; in most practical applications, researchers already know the binding site of their protein target and only want compounds that bind to that site (Yu et al., 2023). In this paper, therefore, we focus on the problem of predicting the ligand pose given the protein binding site.

1.2. A Diffusion-inspired Pose Scoring Function

We note that both traditional docking methods that rely on energy minimization and the diffusion-based DiffDock share some cursory similarities. Namely, both methods initialize a ligand in a random pose and permute the pose (by translating, rotating, and modifying torsional angles) to give the final prediction. Traditional methods explicitly minimize

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by *the 2023 ICML Workshop on Computational Biology*. Do not distribute.

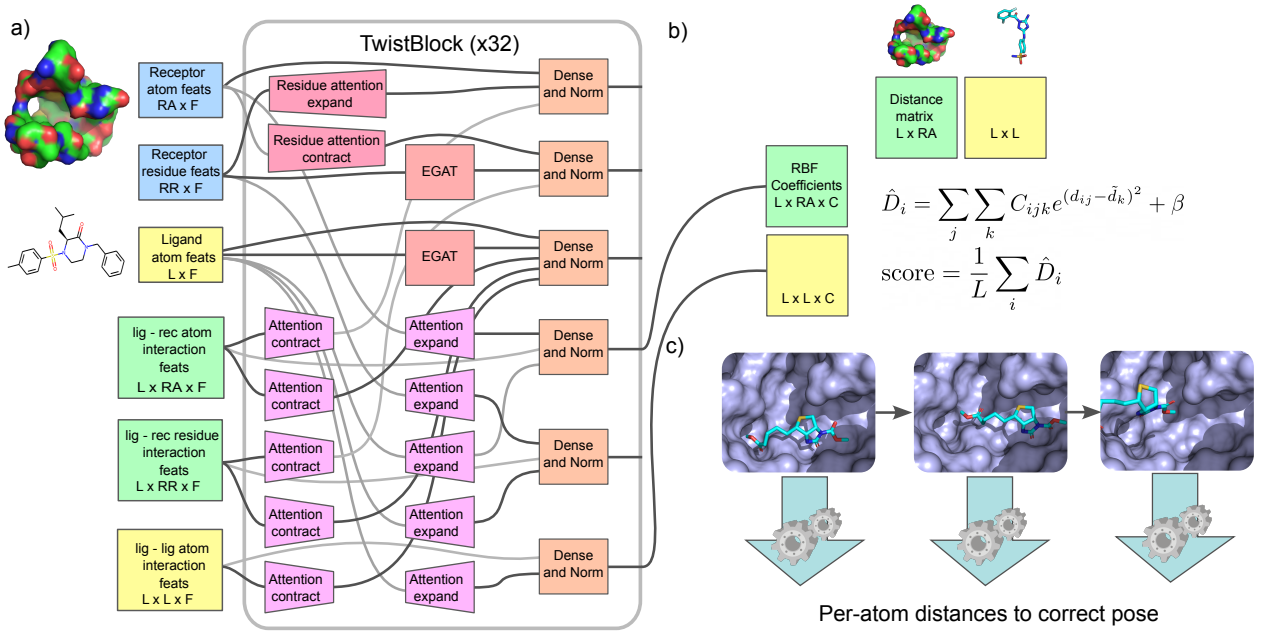

Figure 1. Overview of the PLANTAIN workflow. a) The Twister encoder uses the protein binding pocket and 2D ligand representation to produce coefficients for the scoring function. The encoder is only run once during inference. b) The scoring function uses the coefficients from the encoder, along with all the ligand-ligand and ligand-residue inter-atomic distances, to quickly score a proposed ligand pose. c) When training, we add Gaussian noise to the translations, rotations, and torsional angles of the ground-truth ligand pose. We use our model to predict each ligand atom's distance to its correct position.

an energy function, while the diffusion approach uses a neural network to propose new poses directly. Because DiffDock does not explicitly compute any pose score when generating, it requires a separate confidence model to rank the poses generated by the diffusion model.

We hypothesized that we could use diffusion-inspired training to train a pose scoring model that we then explicitly minimize when inferring. This approach is conceptually simpler than DiffDock. Additionally, by explicitly training on proteins and ligands from different crystal structures, we hoped that our learned scoring function would be less sensitive to the specific protein conformation than physics-based methods.

We validated this hypothesis by developing PLANTAIN (Predicting LigANd pose wiTh an AI scoring functioN). PLANTAIN uses a novel neural network that, given a protein binding pocket and a **2D** representation of the ligand, will compute coefficients for a scoring function that can then be quickly computed for a given **3D** ligand pose. This score function computes the predicted mean absolute error (MAE) of the current pose as compared to the correct pose. We then use this function to minimize the score of the ligand pose using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimizer (Nocedal & Wright, 1999). Using the rigorous benchmarking techniques outlined below, we show that our method performs comparably to GNINA (McNutt et al., 2021) (and outperforms DiffDock) while running significantly faster than both.

1.3. The Importance of Rigorous Benchmarking

Proper benchmarking of docking methods is difficult; it is very easy for hidden biases in the data to artificially inflate reported accuracy. We identify three best practices that should be followed to make docking benchmarks align as closely as possible with actual prospective docking runs.

First, one should not simply take a ligand out of a crystal structure and attempt to re-dock it to the protein from that structure. Protein residues are flexible and will change conformation upon binding to a particular ligand. Thus, we cannot assume knowledge of the exact locations of the protein pocket residues. The more realistic way to perform benchmarking is to dock a ligand to a *different* crystal structure of the same protein. This procedure is called cross-docking (Francoeur et al., 2020; McNutt et al., 2021).

Second, it is important to ensure the method is tested on proteins unseen in the training set. Previous ML work has frequently used time-based splits to separate train and val-

idation/test data. While such splits ensure that the same protein-ligand complex is not present in both train and test splits, the same proteins can be present in both. ML methods can achieve deceptive performance by memorizing the kinds of interactions made by the protein in question. Additionally, proteins that are very similar to one another should be placed in the same data split. Thus, the most rigorous way to split the data is to first cluster proteins based on sequence (or, better yet, structural) similarity, and place different clusters in different splits (Kramer & Gedeck, 2010; Francoeur et al., 2020).

Finally, when docking to a defined binding site, additional bias can be introduced by the definition of the binding pocket. Prior work has often defined the pocket by drawing a box around the ligand with a certain amount of padding. However, the actual pocket might be larger, and we cannot assume knowledge of the specific location within the pocket. Thus, in this paper, we follow Brocidiacono et al. (2022) and define the binding site using x-ray crystallographic poses of all known ligands, not just the ligand in question. To our knowledge, this paper is the first work to report docking results using this more rigorous binding pocket definition.

2. Methods

2.1. Model Architecture

PLANTAIN consists of two parts: (i) a neural network encoder that uses a 3D protein pocket and a 2D ligand graph to produce coefficients, and (ii) a scoring function that uses these coefficients to rapidly evaluate candidate ligand poses. Following Autodock Vina (Trott & Olson, 2010), we use the BFGS optimization method to generate poses that minimize this function.

The encoder, which we call the Twister, is composed of 32 TwistBlocks, each of which updates the internal representation of the ligand atoms and bonds, the protein binding pocket at both the residue and atomic level, and interactions between the ligand atoms with themselves, the protein residues, and the protein atoms. After every other TwistBlock, there is a residual connection.

The Twister encoder takes as input the ligand molecular graph and a graph computed from the protein binding pocket. The nodes in the ligand graph are the heavy atoms, labeled with the element, formal charge, hybridization, number of bonded hydrogens, and aromaticity; the edges are the bonds, labeled with the bond order. Following others (Jing et al., 2021; Brocidiacono et al., 2022), the nodes in the protein graph are the residues (labeled with the amino acid), and an edge exists between them if their α-carbons are within 10 Å. The edges in this graph are annotated with the inter-α-carbon distances, encoded with Gaussian radial basis functions (RBFs). The model also requires all the receptor heavy atoms, annotated with both the atom element and residue amino acid. The model initializes its learned representation to embeddings from all these input features; the ligand-ligand and ligand-receptor interaction features are initialized to zeros. These representations are then fed into the TwistBlocks.

The architecture of each TwistBlock is shown in Figure 1. We use the `EGATConv` graph convolutions from DGL (Wang et al., 2020) to update the ligand atom and protein residue graph information. We use a variant of scaled-dot-product attention (Vaswani et al., 2017) to pass information between the residue-level and atomic-level representations of the protein pocket. We use this same attention mechanism to pass information to and from all the interaction representations. After all the graph convolutions and attention layers, we use dense layers followed by layer normalization (Ba et al., 2016) to finalize the update for each feature. We use a `LeakyReLU` activation after each operation outlined above.

We use the Twister encoder to produce coefficients C_{ijk} for each ligand-ligand and ligand-protein atom pair. We can then use these coefficients to score any ligand pose. For a given pose, we compute all the relevant inter-atomic distances d_{ij}, and use the following equation to predict the distance of each ligand atom i to its true position ($\hat{D}_i$).

$$\hat{D}_i = \sum_j \sum_k C_{ijk} e^{(d_{ij} - \tilde{d}_k)^2} + \beta \tag{1}$$

Here β is a learnable bias and $\tilde{d}_k$ are 24 RBF reference distances, evenly spaced from 0 to 32 Å. i indexes over ligand atoms, j indexes over both ligand and protein atoms, and k indexes over the RBFs.

When training, we predict the individual ligand atom distances $\hat{D}_i$ directly. When inferring, we use the mean predicted distance $\frac{1}{L}\sum_i \hat{D}_i$ as the global score for a given pose that we seek to minimize during inference.

2.2. Training

Following DiffDock, our training procedure consists of taking the ground-truth ligand pose of each complex and adding increasingly large amounts of noise to its translation, rotation, and torsional angles. We desire that, at the final timestep, the ligand is an essentially "random" conformation within the pocket. We use 16 timesteps; at each timestep, we add Gaussian noise to the translation vectors, rotation vectors, and torsional angles.

Once we have the noised the ligand poses, we compute the distance of each atom to its position in the crystal structure. Our model predicts these distances, and we use the mean-squared-error (MSE) loss function when training. We note that, when training, our model is given the protein pocket

from a different crystal structure than the ligand cognate structure. This is done to ensure our model performs well on cross-docking tasks.

2.2.1. Dataset Preparation

In order to follow the best practices for docking benchmarking outlined above, we start with the cross-docked 2022 dataset (Francoeur et al., 2020). This dataset already contains ligand crystal poses aligned to the poses of cognate receptors, so we can test cross-docking performance. Additionally, we use the data splits and pocket definitions created by Brocidiacono et al. (2022) for the BigBind dataset. The pockets in the dataset are clustered according to their ProBiS similarity score, a measure of structural similarity between two pockets (Konc & Janežič, 2010). The binding pockets are defined to be the set of all residues within 5 Å of any ligand co-crystallized in that pocket.

2.3. Inference

To predict the ligand pose for a new protein-ligand complex, we start by running the Twister encoder to produce the score function coefficients from the pocket structure and 2D ligand graph. We then use RDKit (noa) to generate a 3D conformer and optimize it with the UFF force field (Rappe et al., 1992). To generate a possible pose, we randomize the torsional angles of the conformer and place the ligand in the center of the binding site, rotated randomly and translated by Gaussian noise. We then use the BFGS optimizer to compute translation, rotation, and torsional updates necessary to minimize the scoring function. We generate 16 poses with this method and return the one with the lowest score.

3. Results

We benchmarked PLANTAIN on our test set and compared the performance with GNINA. Following prior convention, a pose is considered to be correct if it is within 2 Å root-mean-square deviation (RMSD) of the correct pose. We also report the accuracy within 5 Å RMSD. To account for the large disparity in the number of ligands per protein in the dataset, we report the average accuracy per pocket in the dataset.

In order to compare our method to DiffDock, we created a subset of the test set without any proteins from the DiffDock training set. We then run all the methods, including DiffDock, on this set. We note that DiffDock is disadvantaged in this benchmark because it is not given any knowledge of the binding pocket.

The results are shown in tables 1 and 2. PLANTAIN performs almost as well as GNINA on 2 Å accuracy and beats it in 5 Å accuracy. Additionally, it runs 6 times as fast. Both PLANTAIN and GNINA beat DiffDock in terms of both accuracy and speed, likely because DiffDock uses the whole protein rather than the binding pocket.

Table 1. Performance of PLANTAIN and GNINA on the Cross-Docked test set.

Method	% <2 Å	% <5 Å	Mean Runtime (s)
GNINA	**32.9**	60.1	34
PLANTAIN	26.3	**66.6**	**5.2**

Table 2. Performance of PLANTAIN, GNINA, and DiffDock on the test set without the DiffDock training proteins.

Method	% <2 Å	% <5 Å	Mean Runtime (s)
GNINA	**34.4**	65.0	34
DiffDock	18.4	47.7	97
PLANTAIN	31.2	**77.3**	**5.2**

4. Discussion

In this paper, we present PLANTAIN, a docking method that uses a novel neural network encoder to generate coefficients for a very fast scoring function. This scoring function is used to optimize an initially random ligand conformation. We show, using best practices for benchmarking, that our method performs about as well as GNINA while running substantially faster.

There are many future directions to pursue. Most importantly, we desire to see how much PLANTAIN is able to assist in virtual screening. We plan on combining PLANTAIN pose prediction with GNINA's binding affinity prediction in order to test the effect of improved pose prediction on virtual screening benchmarks.

Additionally, the speed of PLANTAIN opens up many new possibilities. Previously, flexible docking (wherein some or all of the protein residues are also able to move) has been avoided because of performance concerns. It remains to be seen if PLANTAIN's fast scoring method is amenable to flexible docking.

Overall, PLANTAIN improves the state of the art in pose prediction, and there are clear directions for future improvement. We hope that this method will be useful in future drug discovery scenarios. Full source code for training and evaluating PLANTAIN is available at http://github.com/ommitted/for/blind/review.

References

RDKit: Open-source cheminformatics. URL https://www.rdkit.org/.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer Normalization, July 2016. URL http://arxiv.org/abs/1607.06450. arXiv:1607.06450 [cs, stat].

Brocidiacono, M., Francoeur, P., Aggarwal, R., Popov, K., Koes, D., and Tropsha, A. BigBind: Learning from Nonstructural Data for Structure-Based Virtual Screening, November 2022. URL https://chemrxiv.org/engage/chemrxiv/article-details/6384ad70c5675357f89943c5.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking, October 2022. URL http://arxiv.org/abs/2210.01776. arXiv:2210.01776 [physics, q-bio].

Francoeur, P. G., Masuda, T., Sunseri, J., Jia, A., Iovanisci, R. B., Snyder, I., and Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *Journal of Chemical Information and Modeling*, 60(9):4200–4215, September 2020. ISSN 1549-9596. doi: 10.1021/acs.jcim.0c00411. URL https://doi.org/10.1021/acs.jcim.0c00411. Publisher: American Chemical Society.

Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P., and Shenkin, P. S. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *Journal of Medicinal Chemistry*, 47(7):1739–1749, March 2004. ISSN 0022-2623. doi: 10.1021/jm0306430. URL https://doi.org/10.1021/jm0306430. Publisher: American Chemical Society.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from Protein Structure with Geometric Vector Perceptrons, May 2021. URL http://arxiv.org/abs/2009.01411. arXiv:2009.01411 [cs, q-bio, stat].

Konc, J. and Janežič, D. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics*, 26(9):1160–1168, May 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq100. URL https://doi.org/10.1093/bioinformatics/btq100.

Kramer, C. and 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, November 2010. ISSN 1549-9596. doi: 10.1021/ci100264e. URL https://doi.org/10.1021/ci100264e. Publisher: American Chemical Society.

Lu, W., Wu, Q., Zhang, J., Rao, J., Li, C., and Zheng, S. TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction, June 2022. URL https://www.biorxiv.org/content/10.1101/2022.06.06.495043v1. Pages: 2022.06.06.495043 Section: New Results.

McNutt, A. T., Francoeur, P., Aggarwal, R., Masuda, T., Meli, R., Ragoza, M., Sunseri, J., and Koes, D. R. GNINA 1.0: molecular docking with deep learning. *Journal of Cheminformatics*, 13(1):43, June 2021. ISSN 1758-2946. doi: 10.1186/s13321-021-00522-2. URL https://doi.org/10.1186/s13321-021-00522-2.

Nocedal, J. and Wright, S. J. *Numerical optimization*. Springer, 1999.

Rappe, A. K., Casewit, C. J., Colwell, K. S., Goddard, W. A. I., and Skiff, W. M. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American Chemical Society*, 114(25):10024–10035, December 1992. ISSN 0002-7863. doi: 10.1021/ja00051a040. URL https://doi.org/10.1021/ja00051a040. Publisher: American Chemical Society.

Stärk, H., Ganea, O.-E., Pattanaik, L., Barzilay, R., and Jaakkola, T. EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction, June 2022. URL http://arxiv.org/abs/2202.05146. arXiv:2202.05146 [cs, q-bio].

Trott, O. and 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, January 2010. ISSN 0192-8651. doi: 10.1002/jcc.21334. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041641/.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention Is All You Need, December 2017. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs].

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., and Zhang, Z. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks, August 2020. URL http://arxiv.org/abs/1909.01315. arXiv:1909.01315 [cs, stat].

Yu, Y., Lu, S., Gao, Z., Zheng, H., and Ke, G. Do Deep Learning Models Really Outperform Traditional Approaches in Molecular Docking?, February 2023. URL http://arxiv.org/abs/2302.07134. arXiv:2302.07134 [cs, q-bio].