tinyIFD: A High-throughput Binding Pose Refinement Workflow Through Induced-fit Ligand Docking †

Darren J. Hsu ,*,‡ Russell B. Davidson ,¶ and Jens Glaser ‡

‡*National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA*

¶*Biosciences Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA*

E-mail: hsudj@ornl.gov

Abstract

A critical step in structure-based drug discovery is predicting whether and how a candidate molecule binds to a model of a therapeutic target. However, substantial protein side chain movements prevent current screening methods, such as docking, from accurately predicting the ligand conformations, and require expensive refinements to produce viable candidates. We present the development of a high-throughput and flexible ligand pose refinement workflow, called "tinyIFD". The main features of the workflow includes the use of specialized high-throughput, small-system MD simulation code `mdgx.cuda` and an actively learning model zoo approach. We show the application of this workflow on a large test set of diverse protein targets, achieving 70% and 78%

success rates for finding a crystal-like pose within top-2 and top-5 poses, respectively. We also applied this workflow to the SARS-CoV-2 main protease (M^{pro}) inhibitors, where we demonstrate the benefit of the active learning aspect in this workflow.

Introduction

The rapid spread of infectious diseases in a globalized economy necessitates rapid development of therapeutics, as has been painfully evident in the aftermath of the Covid-19 pandemic. Among the drug development pipeline, the crucial step of high-throughput virtual screening aims to identify ligands with high binding affinities toward a target receptor of interest. The conventional method used in many virtual screens of small molecule inhibitors is docking, whereby a ligand structure is modified to best optimize a scoring function that approximates the quality of receptor-ligand fit, typically keeping the receptor structure rigid.[1–7] In addition, docking aims to predict ligand binding structures that align with those seen in the bioassemblies. In cases where the receptor binding sites are rigid, docking can achieve high accuracy and demonstrate good docking power.[8] However, when the binding site of a receptor is flexible or when cross-docking, where a ligand is docked to the same protein with a different structure, is performed, docking results often suffer from the potential induced-fit effect: where the receptor accommodates the ligand structure by altering its own. Without refinements to the predicted binding poses, the accuracy of cross-docking is generally low.[9]

To alleviate the impact of freezing the receptor, some programs allow a limited number of torsional groups such as protein side chains from the receptors to rotate,[2,3] or use an ensemble of receptor conformers to approximate the conformational flexibility of the receptor.[10–12] Molecular dynamics (MD) can, in principle, provide a high accuracy description of the protein-ligand system with full account of conformational flexibility, albeit at the expense of relativity high computational cost.[13]

Some protocols rely on MD simulations and have shown high prediction accuracies, such as the IFD-MD.[14] However, the computational cost can be as high as 250 graphics processing

unit (GPU) hours per ligand, making them unsuitable for refining a large set of ligands. Therefore, a high-throughput method that models ligand-receptor interactions using MD is needed, one that would enable refinement of virtually screened docked ligand poses.

In this work, we developed a workflow for refining docked poses by utilizing a specialized small-system MD engine, `mdgx` from the `AmberTools`.[15] Our aim is to simulate just the residues around the binding site, thereby better sampling the conformational space that would be of interest. Because of the tiny simulation system and the induced-fit nature, we refer to this workflow as "tinyIFD". Originally designed for ligand parameter generation, the `mdgx` code utilizes individual streaming multiprocessors (SMs, 80 on an NVIDIA V100) on a GPU card to perform as many simulations in parallel, which generates a high throughput when system size is small enough. The size of the shared memory of the GPU card imposes a limit on the system size to 928 atoms, which amounts to the simulation of a ligand and about 65 residues surrounding it. We describe the strategy to prepare such a receptor core-ligand complex for simulations, measure the simulation throughput and show it only consumes modest computational resources.

With a large amount of aggregated simulation snapshots comes the need of predicting ligand poses that are favorable, hereby defined as less than 2.5 Å of symmetry-corrected heavy atom root-mean-square deviation (RMSD) to the target poses. To this end we developed an active learning approach where, for each receptor-ligand complex, we construct a classification model to predict whether each snapshot is favorable or not. During inference time for a new complex, we use a "model zoo" approach to select a fixed number of existing models based on a distance metric defined using Tanimoto similarity[16] of the ligand and Levenshtein distance[17] of the receptor string to those in the training set. When the experimental structure of a test set complex is determined, data from that complex graduate into the training set and help future predictions for the same protein family.

We show that the tinyIFD protocol improves the success rate of finding a favorable pose compared to simple docking. We demonstrate the concept of active learning by testing an

expanded set of SARS-CoV-2 main protease (M^{pro}) non-covalent ligands, and show that by including some systems of the protein family of interest in the training set, the success rates of the refinement increases significantly. Finally, we discuss some limitations of this workflow and provide potential workarounds.

Methods

tinyIFD Workflow Overview

The overall tinyIFD workflow is depicted in Figure 1. It is a straightforward and scalable workflow that can be utilized on personal computers with a single GPU card as well as in high-performance computing environments. The required packages of this workflow are all open-source, which include `python`, `openbabel`,[18] AutoDock Vina,[2] `RDKit`,[19] `mdtraj`,[20] `OpenMM`,[21] `AmberTools`,[15] Open Drug Discovery Toolkit (`oddt`),[22] `XGBoost`,[23] and `spyrmsd`.[24] A list of software versions can be found in Supporting Information (Section **??**).

The input to this workflow is a set of receptor PDB files, a set of ligand PDB files, and a job descriptor list that includes pointers to the files as well as the docking center. For each job, the receptor PDB file is first processed by `tleap`[15] to construct missing heavy atoms and protonate the protein. The protonated protein and ligand PDB files are converted to `.pdbqt` files with `openbabel`. AutoDock Vina is used to dock the ligand into the protein, taking in the docking center information from the job list. Note that this part can, in principle, be substituted with any docking program with minimal changes to the workflow.

For MD simulations, the ligand is parameterized with `Antechamber`[25] available from `AmberTools`, while the truncated protein is parameterized with `tleap`. Truncation of the protein is done with an in-house script taking into account the atom number limit (see below), which also changes the mass of Cα atoms to a special value, so as to fix these atoms in subsequent simulations. The receptor-ligand complexes are assembled with `RDKit` and saved as AMBER simulation input files (`.prmtop` and `.inpcrd` files). The complexes are

energy minimized using OpenMM and simulated with a modified version of `mdgx`.

After simulations, the features used in the classification are calculated from the resulting trajectories using `cpptraj`[26] and an in-house script based on `mdtraj` Python package. These features are passed to a collection of classification models which outputs a weighted probability of a sample structure being a favorable pose. Starting from the structure with the highest probability, we pick distinct structures that are R Å RMSD to those already picked, where R is a hyperparameter to be tuned. These distinct poses are ranked by their predicted probabilities, taken as the best predicted poses, and returned to the user.

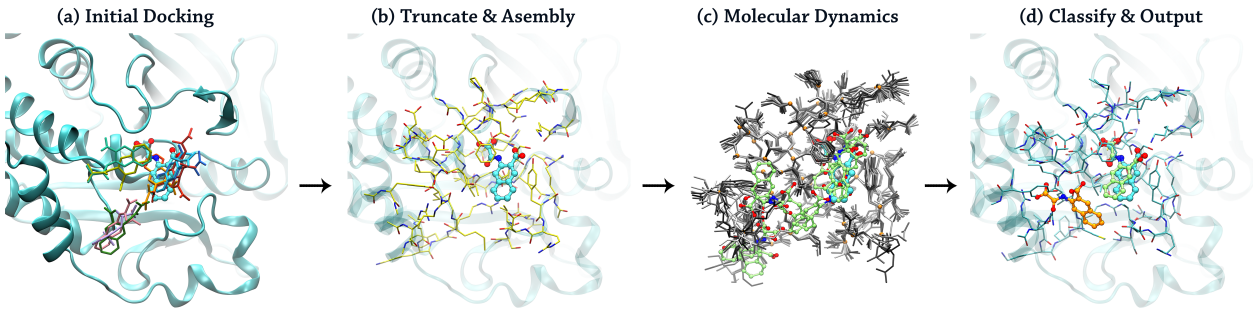


Figure 1: Workflow of tinyIFD using cross-docking of the ligand from 1C84 to 1WAX as an example. **(a)** Initial docking is performed by taking the ligand from 1C84 (cyan structure in ball-and-stick model) and docking to the protein structure of 1WAX (ribbons in cyan) with a box centered at the 1WAX ligand (not shown). The docking results are shown in thinner licorice models with varying colors. **(b)** Some protein residues (licorice models in yellow) of 1WAX are used to assemble the simulation system based on the distance to the 1WAX ligand (cyan ball-and-stick model), the number of which capped by the hardware limit of 928 atoms. To limit charged termini, bridging residues are included, and caps are added to the fragments. The nontrivial task of core truncation is done by iteratively changing an inclusion distance. **(c)** Molecular dynamics (MD) simulation with `mdgx.cuda` results in multiple snapshots (gray licorice models for protein residues, green ball-and-stick models for the ligand). Some are closer to the ligand ground truth (cyan). **(d)** The feature vector is calculated for each snapshot and used to classify the probability of the ligand being a favorable pose. From the most probable structure, clustering is performed by calculating the heavy-atom root-mean-squared distance to those already selected, resulting in distinct outputs (green and orange structures). In this case, the top-ranked pose is a favorable pose (green), close to the ground truth (cyan). The graphics were produced with Visual Molecular Dynamics (VMD).[27]

Generation of Docking Datasets

To develop the overall workflow, we first constructed a broad cross-dock dataset ("main" dataset) based on the list from Miller and coworkers,[14] where their induced-fit docking was developed. The dataset includes 369 cross-dock tasks spanning 40 protein families. For each docking task, the ligand whose poses are of interest is referred to as the "target" ligand. The target ligand is to be docked into the structure of a template protein. The template protein is ideally a *holo*-protein with a template ligand.

To test the developed workflow, we compiled a narrower dataset focusing on SARS-CoV-2 main protease (M^{pro}), which is referred to as the "M^{pro}" dataset. The active site of PDB entry 5R84 is set to be the target protein structure. 35 non-covalent M^{pro} ligands are selected to be docked to this active site.

All structure files were downloaded from the RCSB PDB database.[28] The respective PDB IDs are reported in the Supporting Information (section **??**).

Initial Rigid-body Docking

We utilized the the python binding of AutoDock Vina[2] to perform initial docking of the ligands with the following settings: The search center was set as the center of the template ligand. The side length of the cubic search box was set at 24 Å. The Monte Carlo (MC) search exhaustiveness, controlling the number of independent MC runs, was set to 32. Up to 20 poses were saved. The rest of the parameters, such as energy range of ligands and RMSD clustering radius, were left as default.

Protein Truncation

We cropped the receptor structures to their active site cores by choosing residues radially distributed closest to the docking center. A hard limit of 928 atoms is enforced by the `mdgx` program, as dictated by the size of GPU shared memory. Because of this sensitivity

to the number of atoms, the missing protein side chains and hydrogens were first repaired with AMBER's `tleap`. The truncation strategy was to include as many intact residues as possible by iteratively increasing the search radius, taking into account the number of atoms from the ligand. Truncating the receptor into fragments inevitably generated multiple chain breaks, or termini. If the termini of two fragments are less then two residues apart, they are merged as one longer fragment with bridging residues included. Otherwise, each fragment is capped with acetyl (ACE) and N-methylamide (NME) residues to minimize the number of charged groups, unless the fragment terminus is also the N- or C-terminus in the full receptor molecule, in which case no change is made. The coordinates of the heavy atoms in these capping residues were taken from the flanking residues, so as not to create steric clashes.

Pre-processing for MD Simulations

The short chains of the truncated core were assembled with `RDKit` and saved as AMBER simulation parameter (`.prmtop`) and input coordinate (`.inpcrd`) files. Finally, we changed the masses of Cα atoms in the parameter file to 9.999E+03 to inform the modified `mdgx` code that these atoms should be fixed in the simulations (see next section). We also saved a copy where the masses of the same atoms were set to zero, which fixed the atoms in OpenMM simulations. Note that the present protocol disables the discovery of extra binding sites, which is beyond the scope of this study. A workaround is provided in section .

For the ligands, the top-ranked pose from AutoDock Vina were parameterized with `Antechamber` from `AmberTools`, using the `AM1-BCC` charge model.[29] After parametrization, the receptor and ligands were assembled again with `RDKit`. Each complex was energy minimized with `OpenMM` before production runs.

MD Sampling

The specialized MD engine `mdgx` within `AmberTools` was modified to interpret the special mass value as an instruction to freeze those atoms. The simulations were performed with

Generalized-Born (GB) implicit solvation, without periodic boundary conditions, and propagated with velocity Verlet integrator[30] with a 2-fs time step. Bond lengths are constrained with the RATTLE algorithm.[31] A Langevin thermostat is used to maintain a system temperature of 310 K.[32] For each pose, 20 independent, 8-ns simulations were performed, saving snapshots every 100 ps. The length of simulation is chosen such that simulations can be done in under two hours wall time. This eliminates the need for checkpointing on the Summit supercomputer where the simulations were limited to two hours. This resulted in an aggregation of, at most, 3.2 μs of simulation and 32,000 frames for a ligand. Users are free to increase the amount of sampling.

Feature Extraction from MD Trajectories

In order to single out the ligand motion, all of the snapshots collected in the MD sampling step were aligned to the target receptor structure, using only the coordinates of the receptor Cα as the reference. The ligand RMSD values were calculated with symmetry correction using `spyrmsd`.

In preparation of the classification, we calculated 46 features including several sets, focusing on different aspects of the system. A detailed explanation for how these terms are calculated are provided in the Supporting Information (Section **??**).

- Overall system energies (2): van der Waals (vdW) and GB solvation terms
- Ligand-centric energy terms (3): nonbonding vdW, 1-4 electrostatic, and nonbonding electrostatic energies of the ligand
- Interaction energies (2): Change in overall system vdW and electrostatic energies
- Solvent accessible surface area (SASA) terms (5): Change of SASA of entire complex and of the ligand; absolute SASA of ligand {C, S, P} atoms, ligand aliphatic C atoms, and ligand polar atoms

- Template-based terms (4): protein side chain motion, global and local; overlap of volume and pharmacophores between sampled poses and the template ligand
- Contact-based terms (30): hydrogen bond term between ligand and protein; contacts between protein {H, C, N, O} and ligand {H, C, N, O, S, halogen} atoms, as well as between protein S atoms and ligand {H, C, N, O, halogen} atoms

The inclusion of these features is inspired by previous research.[14,33,34] The GB solvation energy, interaction energies and SASA terms are reminiscent of the molecular mechanics generalized Born surface area (MM/GBSA) method.[35] The template-based terms are inspired from the IFD-MD work.[14] The contact-based terms resemble those from the random-forest scoring function (RF-score) implementation.[33] However, the score for each of the contact terms is determined using a distance-based function found in ChemScore,[36,37] similar to that implemented in Protein–Ligand Empirical Interaction Components (PLEIC),[34] instead of a numerical count per element pairs to allow for normalization (see next subsection). We note that more advanced features could be incorporated, such as the number of potential bridging water molecules[38,39] and excess chemical potential from molecular quasichemical theory,[40] provided they increase model performance and can be calculated in reasonable time for large number of individual snapshots.

XGBoost Pose Classifier Model Zoo

To avoid costly retraining of a large model, we employed an ensemble prediction paradigm. The strategy for ranking the sampled poses is to select a predetermined number (N) of pose classifiers from the "model zoo" and perform a weighted average of individual predicted probabilities of the pose being a favorable one. To build a model for a training system, we train a XGBoost classifier on the above feature set, using 2.5 Å RMSD as a cutoff for "favorable" ($<$ 2.5 Å RMSD to crystal pose) and "unfavorable" (otherwise) poses. To prepare the input feature set, we normalize the calculated features to a distribution with a mean of

0 and standard deviation of 1. This normalization is to the level of individual protein-ligand complexes, as the values from simulations of different complexes are generally not directly comparable. Furthermore, we remove samples with any normalized feature values beyond 5 standard deviations. During the development of this workflow, we observed that the data exhibit substantial class imbalance, where unfavorable poses far outweigh positive ones. To alleviate this problem, we included simulations directly starting from the crystal pose to increase the ratio of favorable / unfavorable samples. For each model, we associate a code string comprised of the 1-letter amino acid codes of the included protein residues and the caps, which are coded as a space. Finally, the ligand SMILES string of the complex is associated as an attribute of the model.

At predict time, a composite distance function is used to determine the weight of each model prediction based on the similarity of the test system to systems from which the models are built. The amino acid code string of the test system is first extracted, and the integer Levenshtein distances (L_{dist}) to all the models are calculated. The Tanimoto similarity scores (T_{sim}) of the test ligand and the ligands of the models are calculated as well. The composite distance between the new system and the model is calculated as $(L_{dist}+\epsilon)^n+k(1-T_{sim})$, where ϵ, n, and k are hyperparameters that will be tuned. N models with the lowest composite distances were picked. The predicted probability of each pose is then calculated as the weighted average of individual predicted probabilities. This approach allows the workflow to pick models of familiar systems for prediction, enables active learning, and reduces the computational cost of prediction to a fixed amount. Newly trained models can simply be placed in the model zoo when a ground truth crystal pose from the experiments becomes available, avoiding costly retrainings of a large model.

Results and discussion

MD Simulations

For each of the 369 docking tasks, up to 21 poses (docked pose and crystal pose for training) were combined with the protein, resulting in 7,322 protein-ligand poses. For each pose, 20 simulations of 8 ns were carried out, with a total of 146,440 individual MD runs. The simulations were performed on the Summit supercomputer, distributed over 1,836 GPUs on 306 nodes. Overall, the simulations took 110 minutes and generated 1.17 millisecond of aggregated simulations, resulting in 11.7 million snapshots.

Dataset Preparation and Splitting for Training and Testing

We constructed a cross-docking dataset from MD snapshots spanning 40 targets with 369 cases. This is similar to Schrödinger's dataset.[14] We will refer to this dataset as the "main dataset". The dataset was split horizontally,[41] regardless of protein families, to the training and testing sets with a ratio of 60%:40% to 221 and 149 cases, respectively, similar to what is done in Schrödinger's implementation.[14] Note that there exist other methods for splitting the dataset, such as per-target split or vertical split (assigning entire data from a target to either training or testing set). The models generally perform worst with a vertical split. In this work, we chose to split the dataset in a more realistic way, where available training data come from all possible targets. However, for completeness, as we test the unprecedented system from SARS-CoV-2 M^{pro} protein in subsection , we verify how this workflow can be utilized for a brand new target.

In order not to include extremely imbalanced structural pool, such as cases without any favorable poses, we set a threshold of 0.05% of favorable pose ratio, below which the training set is excluded. For the remaining cases where favorable pose ratio is less than 1%, we undersampled the unfavorable poses such that 1% of the poses are favorable. Each case was trained separately, resulting in 208 individual models in the model zoo.

XGBoost Model Zoo Performance Tuning

We carried out systematic tuning on the hyperparameters of the XGBoost models and the model zoo. The set of hyperparameters that resulted in the highest performance was listed in Table 1.

Table 1: Optimized values of the hyperparameters for the XGBoost model zoo

Meaning	Hyperparameter	Value
Number of decision trees in XGBoost models	n_estimators	200
Maximal depth of trees in XGBoost	max_depth	6
Regularization parameter in XGBoost	gamma	0.2
Fraction of features sampled by a tree	colsample_bytree	1.0
Fraction of samples sampled by a tree	subsample	1.0
Number of XGBoost models for ensemble prediction	N	20
Base distance in addition to Levenshtein distance	ϵ	0.1
Exponent of Levenshtein distance	n	-0.5
Scaling factor for Tanimoto Similarity	k	2.0
RMSD of clusters (Å)	R	2.5

tinyIFD Performance

As tinyIFD is a refinement method, it is helpful to establish a baseline performance of the AutoDock Vina. AutoDock Vina was able to generate a favorable docking pose, defined as having less than or equal to 2.5 Å heavy atom RMSD from the crystal structure, within the two, five and twenty top ranked guesses in 107, 144 and 209 of the 369 cases, yielding a success rate of 29%, 39%, and 57%, respectively. The values are low and comparable to the performance of GlideSP in Miller and coworkers' benchmark.[14]

The refinement workflow based on the AutoDock Vina docking results aims to improve the success rate by allowing the protein side chain to move freely. Indeed, with the ensemble prediction method, tinyIFD was able to predict 103, 115, and 118 of the 148 test cases, or 70%, 78%, and 80%, for the top two, five, and twenty guesses, respectively. All top-N success rates of tinyIFD are higher than the corresponding values achieved by AutoDock Vina indicating that by allowing the molecular system to evolve, more favorable poses can be sampled. A list of the docking and tinyIFD refinement results can be found in the Supporting

Information.

The breakdown of how tinyIFD was able to preserve AutoDock Vina successes and rescue failures is shown in Table 2. When Vina was able to find at least a favorable pose in the top-20 guesses, the tinyIFD workflow was able to generate a favorable pose within the top-2 guesses in 70 out of 81 cases (86 %) and within top-5 guesses in the to 77 out of 81 poses (95 %). In the 67 cases where Vina was unable to find a favorable pose, the workflow still found a favorable pose within top-2 and top-5 guesses in 49 % and 57 % of the cases. This demonstrate the remarkable capabilities of tinyIFD in both ranking and exploring favorable poses by incorporating large scale sampling.

Table 2: Results breakdown per Vina and tinyIFD results (N = 148).

		tinyIFD			
		Top 2	Top 5	Top 20	Failed
Docking	Top 2	36	3	1	0
	Top 5	17	2	0	0
	Top 20	17	2	1	2
	Failed	33	5	1	28

An example that showcases the impact of the induced-fit effect on the outcome of docking and refinement is the docking of ligand DT2, a triazolopyrimidine inhibitor, of the human cyclin-dependent kinase 2 (CDK2) from PDB 2C6K to 1PXI (Figure 2). A superposition of 2C6K ligand and 1PXI protein active site structures shows that the sulfonamide group of DT2 in 2C6K are in close proximity of the LYS89 in 1PXI, with a minimum distance of 1.01 Å between the ligand sulfur atom and the NZ of the lysine (Figure 2a). All 20 docked poses avoided the clash, but as a result none is a favorable pose (Figure 2b). The tinyIFD workflow resolved this problem and was able to sample and predict the crystal pose, by allowing the LYS89 to move out of the way of the ligand (Figure 2c).

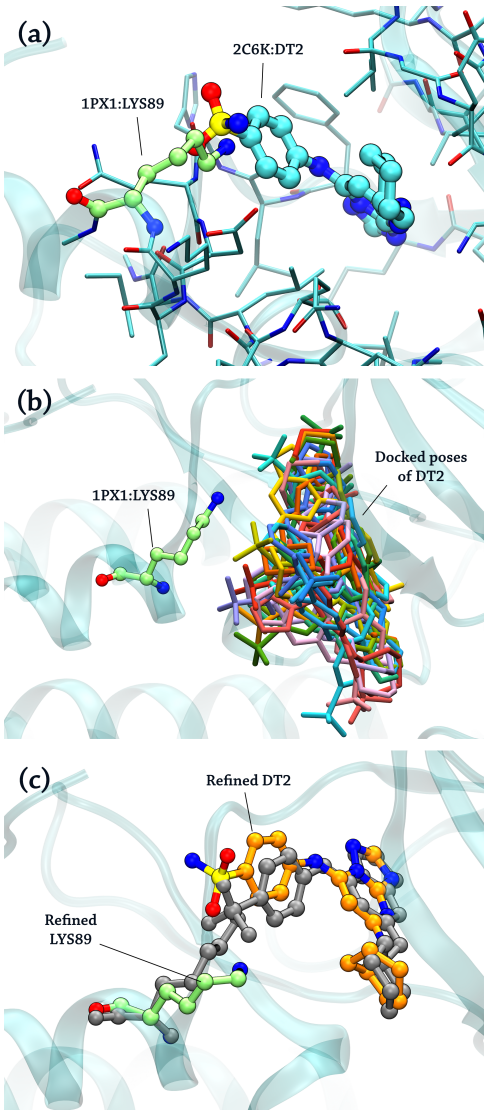


Figure 2: An example showing the importance of induced-fit effect in docking and refinement. **(a)** Structural superposition showing the steric clash of ligand DT2 of PDB 2C6K (ball-and-stick model in cyan) and LYS89 (green ball-and-stick model) of 1PXI within the active site (licorice models). **(b)** Docked poses of ligand DT2 (licorice models of various colors) avoid clashes with the LYS89 (green ball-and-stick model). **(c)** tinyIFD refined LYS89 and the DT2 poses, moving away the former (green ball-and-stick model) to allow for sampling of a near-native ligand pose of the latter (orange ball-and-stick model). The clashing structures from (a) are depicted with gray ball-and-stick models for reference.

Failure Analysis

We inspect how the tinyIFD workflow failed some of the test cases. The failed cases include those where tinyIFD predictions are worse than Vina docking results and where tinyIFD

cannot rescue docking results that failed (Table 2). Of the 34 cases that qualify as a failure, six cases are in the former group, while 28 of them are in the latter.

For the four cases where Vina gave top-2 poses but tinyIFD did not, tinyIFD refinement of docking ligand 132 in 1GJ4 (thrombin) to 1C5N and ligand HFS from 2ETK (rho-associated kinase) to 2ESM placed the favorable pose at the third, while in the case of placing IIE from 2BQW (factor XA) in 2BOK, the favorable pose is ranked the fifth. Notably, in the case of ligand MYU from 2O64 (PIM1 kinase) to 1YI3, while the favorable pose dropped to the seventh, a pose with an RMSD of 2.75 Å is at the third rank. The RMSD filter implemented to avoid over-selection of the same ligand conformation may have absorbed some of the favorable poses into that selection.

Most failures of tinyIFD in refining the other 30 cases can be attributed to the low concentration or lack of favorable snapshots sampled in MD simulations. 22 cases have no favorable snapshots at all, while five have less than 1.5 %. These reflect the strong upstream dependency on docking quality of the tinyIFD workflow. The remaining failure cases, where MD simulations were able to sample a meaningful portion of favorable poses but the classification failed, illuminate the room for improvement of the tinyIFD, such as taking into account the halogen bonding and effect of metal ions in the ligand design for 3SHY,[42] and accounting for the drastically different ligand designs between template and target receptors. For example, the heat shock protein 90 structures, 1YET and 2BYI, exhibit completely different strategies: the ligand GDM in 1YET is a macrocycle[43] while the design for ligand 2DD in 2BYI specifically exploits the contact surface of Phe137.[44] Similarly, in docking 3CB from 1WSS (factor VIIa) to 2FLB, the target ligand 3CB is much larger than the template ligand 6NH and interacts with a different set of residues.[45,46]

Ensemble Prediction with Distance Weighting Is Better Than Individual Models

We compared the ability of individual models to predict favorable poses to that of ensemble models. As shown in Table 3, the weighted average method significantly outperforms other methods, including selecting N best models, defined as having the highest individual top-5 success rates, as well as selecting 20 random models. The weighted average method did especially well on identifying a favorable pose within the top 2 choice, which is important for lowering the cost of more expensive downstream simulations such as free energy-based methods by lowering the number of poses to be examined.

Table 3: Success rate of the model zoo using different selection strategies. For random selections, standard deviations are included. Values in parentheses are counts of successful refinements, out of a set of 148 cases.

Method	Top 2	Top 5	Top 20
Weighted average	70%	78%	80%
Best single model	49%	69%	79%
Best 5 models	50%	67%	76%
Best 10 models	53%	69%	76%
Best 20 models	53%	69%	77%
Random 20 models (7 trials)	$49 \pm 4\%$	$67 \pm 1\%$	$78 \pm 1\%$

Computational Cost

The main computational cost for tinyIFD is due to the MD simulations. Because the `mdgx` code can be run on an NVIDIA V100 GPU in parallel with up to 80 replicas, the 3.2 μs aggregated simulations can be done in about 9 GPU hours. The computational cost can be adjusted based on the amount of sampling desired, by changing simulation length or number of replicas per pose. Table 4 lists several possible scenarios of reduced sampling, showing no major degradation in performance. Of particular interest is reducing simulation lengths to 4 ns, as it directly cuts the time-to-solution by a half compared to the reference protocol.

Table 4: Performance of the model zoo using different selection strategies.

Method	Top 2	Top 5	Top 20
Reference	70%	78%	80%
10 replicas	69%	76%	78%
6 ns simulations	68%	77%	80%
4 ns simulations	68%	76%	80%

Application of tinyIFD to SARS-CoV-2 M^{pro} Docking

To demonstrate a typical use case when approaching a new target, we collected 35 crystal structures of the SARS-CoV-2 M^{pro} with available non-covalent inhibitors on the RCSB PDB database and applied the same protocol with the same parameters. We refer to this dataset as "M^{pro} dataset". All of the ligands were cross-docked into the 5R84 protein structure. The performances of different settings are listed in Table 5. AutoDock Vina in fact performed better than for the main dataset, achieving 37% and 49% top-2 and top-5 success rate, respectively.

We then tested the tinyIFD workflow, using all the models from the main dataset (353 models) as the initial model zoo. When we use only the models from the main dataset, the workflow performed slightly better at 40% and 57% top-2 and top-5 success rate, respectively. However, since there is almost always a complex structure available, in this case 5R84 and its own ligand, a more relevant metric would be the performance when the model from the self-dock task is included, where the top-2 success rates increased to 53%. If there are multiple available structures, the performance further increases to 67% and 70% top-2 success rates when two and four other models are included. The included models are constructed by applying the workflow for docking the ligands from 7AU4, 7RNK, 7B77, and 7RN4 to 5R84; these would represent the "incorporated test set" after experimentally determined structures became available.

As an example, refinement of cross-docked pose of ligand V1B from 7QBB to 5R84 is shown in Figure 3. AutoDock Vina did not produce a favorable pose for this case (best pose is 3.08 Å), but the tinyIFD workflow was able to rescue the docking failure and suggest

a pose that is 0.57 Å RMSD to the crystal pose with the inclusion of three 5R84-based models. Because of the (identical) protein sequence that gave a Levenshtein distance of 0, these models from the same protein are heavily weighted during the pose quality prediction stage. This demonstrates the use of active learning; when more data is available for a specific protein, the workflow naturally utilizes these more local data to predict the pose quality, and the performance increases. Note the reason that the performance seemed to be capped at 70% success rate is due to that, only 80% of the cases have MD snapshots of ligands in a favorable pose.

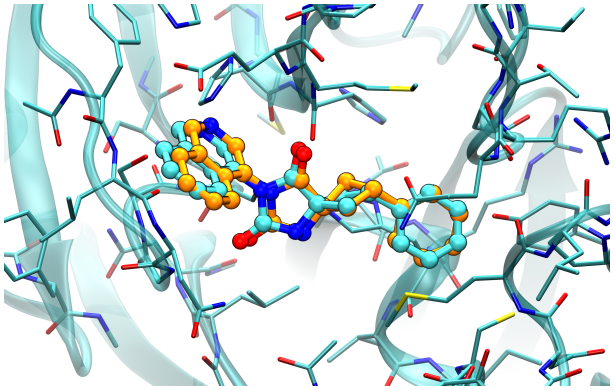


Figure 3: Docking ligand V1B from 7QBB to 5R84. The crystal and refined poses are shown in cyan and orange ball-and-stick models, respectively.

Table 5: Performance of the model zoo using different selection strategies. Docking has 35 cases and tinyIFD has 30 cases. The "2 other models" are from induced-fit docking of ligands from 7AU4 and 7RNK to 5R84, while the "4 other models" additionally include those of 7B77, and 7RN4.

Method	Top 2	Top 5	Top 20
Docking	37%	49%	51%
Models from main dataset	40%	57%	63%
+model from 5R84 self-dock	53%	57%	60%
+models from 5R84 and 2 others	67%	70%	70%
+models from 5R84 and 4 others	70%	70%	70%

Limitation of This Workflow

Summarizing and extending observations above, we discuss observed limitations of the workflow due to various required conditions to set up the simulations and when this workflow may be inapplicable.

Limited sampling due to fixed CA atoms

This code is less effective for refinements where there may be large CA atom movements, for example the docking and refinement of ligand 3EA of 2ATH to 2FVJ (Figure 4). In this case, the MET364-CA within the alpha helix in 2FVJ pushes inwards in the active site as the ligand does not have a steric interaction with the residue. However, in 2ATH the same CA atom moved 1.28 Å. It is likely because the simulation is done fixing the CA atoms, the system is unable to explore the part of the conformational landscape that includes favorable docking poses of 3EA.

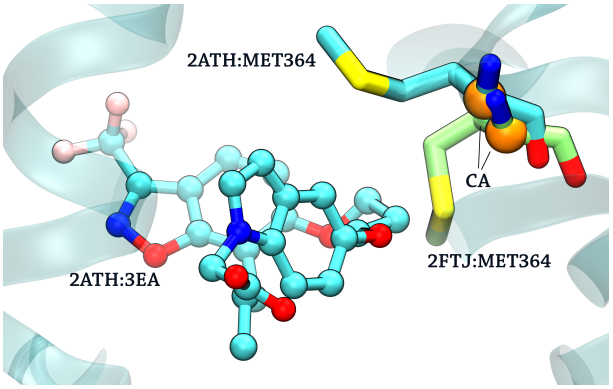


Figure 4: Failed refinement of ligand 3EA from 2ATH potentially due to fixed CA atoms.

The problem can potentially be addressed by allowing movements of the CA atoms. However, care must be taken as allowing all of the CA atoms to move may result in deformation of the artificially truncated system. A potential compromise could be to allow only CA atoms of the receptor-ligand interface residues to move, while keeping the rest of the CA atoms such as those of second shell residues and terminal residues frozen.

Large binding sites that are not fully occupied by the template ligand, exploring new binding sites, or generally when there is no information about the template ligands

Sometimes the template ligands do not occupy the entirety of the binding site, or in general the sizes of the template and target ligands are vastly different. This results in both (a) potential for systematically incorrect estimates of the template overlap feature and (b) a misrepresentation of included residues in the truncated system. An example is the failed case of docking 3CB from 1WSS to 2FLB mentioned above (Figure 5). A similar circumstance arises when only the *apo* receptor structure is available, or if new binding sites such as cryptic pockets are of interest, where the template overlap becomes undefined and truncation cannot be carried out.

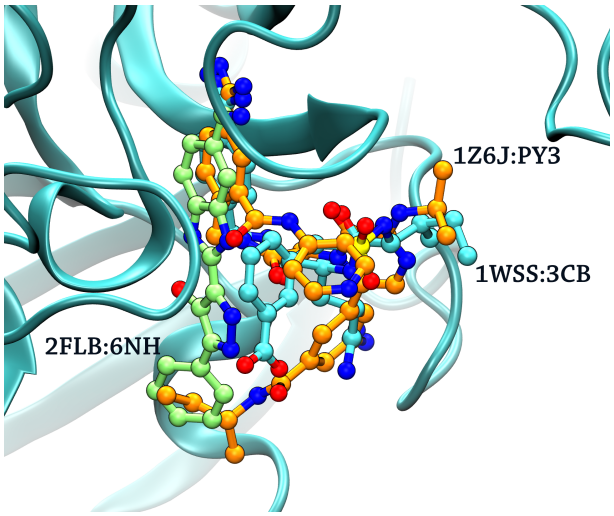


Figure 5: Demonstrating a better template selection. For the large ligand 3CB (blue) of 1WSS, selecting 1Z6J as the template with a similarly sized ligand PY3 (orange) would likely be better than 2FLB, which has a smaller ligand 6NH (green).

For cases where *holo* template structures are available, it is advisable to select one with a ligand similar in size to approximate the binding site shape. For example, in the factor VIIa protein family, a better choice of the receptor as the template for the larger 3CB ligand would be 1Z6J, which has a similarly sized ligand PY3 (Figure 5).[47] Alternatively, a way to circumvent these problems could be to carry out a short, normal MD simulation including

the entire (*apo*) receptor, and use one of the many binding pocket identification methods[48–51] to define a volume for the pocket. A dummy ligand can then be defined where the volume of dummy ligand atoms covers the desired pocket, which can be used to truncate the receptor structure, and the workflow proceeds as normal.

Upstream dependency on rigid-body docking results

The tinyIFD workflow employs AutoDock Vina as the provider of docking poses, serving as the initial guesses of the structures for the MD simulations. This places an upstream dependency of the workflow performance on the quality of the docking result. However, in principle, the workflow can take the output from any docking programs with appropriate conversions of the docking result, for example through the use of `OpenBabel`. Users are therefore advised to utilize programs best suited for the particular docking cases of interest.

While this workflow is designed to work with only open source codes, it is not limited to so. In particular, within Schrödinger's workflow, the Phase-Glide-Prime (PGP) steps are claimed to generate at least a favorable pose in 99% of the cross-docking cases within the top-20 guesses, although such poses are not reliably highly ranked.[14] The tinyIFD workflow can be adapted so that after MD sampling, these PGP-generated poses act as filters after the classification step, which could increase the performance of the workflow.

Conclusion

We have developed the tinyIFD workflow to carry out pose refinement based on docking results. The workflow enables modeling of induced-fit effects during docking, requires relatively low computational cost, and can refine docked poses to those seen in crystal structures. It also enables active learning when approaching new receptor families due to its ensemble prediction paradigm, achieving progressively better performance as new ground truth data become available.

Data Availability

The input files for this workflow, a csv file that contains docking center for each of the 369 cases, structural files including both apo and holo forms of the cleaned template receptor, and the target ligand are available on Zenodo server (DOI: 10.5281/zenodo.7401839). In addition, the tinyIFD workflow code, user manual, and analysis script for both the main dataset and M^{pro} dataset are available on GitHub (https://github.com/darrenjhsu/tiny_IFD). A separate set of structural coordinates for the M^{pro} dataset is also available in the same Zenodo deposit. All software dependencies of this workflow are open-source.

Acknowledgement

The authors thank Dilipkumar N. Asthagiri for constructive discussions. CARES act funding to the Oak Ridge Leadership Computing Facility (OLCF) through DOE ASCR in support of this research is also acknowledged, as is the Laboratory Directed Research and Development Program at Oak Ridge National Laboratory (ORNL). This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

Supporting Information Available

(1) A list of software and the versions of which used in this manuscript, (2) detailed explanation of features extracted from MD snapshots for the classifiers, (3) a list of cross-docking cases used in the model zoo, (4) a list of cross-docking cases in the training set but excluded from the model zoo, (5) test set refinement results, and (6) a list of PDB entries used for the M^{pro} dataset are included in the Supporting Information:

References

(1) Trott, O.; Olson, A. J. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2009**, NA–NA.

(2) Eberhardt, J.; Santos-Martins, D.; Tillack, A. F.; Forli, S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *J. Chem. Inf. Model.* **2021**, *61*, 3891–3898.

(3) 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.; Shenkin, P. S. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.* **2004**, *47*, 1739–1749.

(4) Halgren, T. A.; Murphy, R. B.; Friesner, R. A.; Beard, H. S.; Frye, L. L.; Pollard, W. T.; Banks, J. L. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 2. Enrichment Factors in Database Screening. *J. Med. Chem.* **2004**, *47*, 1750–1759.

(5) Friesner, R. A.; Murphy, R. B.; Repasky, M. P.; Frye, L. L.; Greenwood, J. R.; Halgren, T. A.; Sanschagrin, P. C.; Mainz, D. T. Extra Precision Glide: Docking and Scoring Incorporating a Model of Hydrophobic Enclosure for ProteinLigand Complexes. *J. Med. Chem.* **2006**, *49*, 6177–6196.

(6) Verdonk, M. L.; Cole, J. C.; Hartshorn, M. J.; Murray, C. W.; Taylor, R. D. Improved protein-ligand docking using GOLD. *Proteins Struct. Funct. Bioinforma.* **2003**, *52*, 609–623.

(7) Ruiz-Carmona, S.; Alvarez-Garcia, D.; Foloppe, N.; Garmendia-Doval, A. B.; Juhos, S.; Schmidtke, P.; Barril, X.; Hubbard, R. E.; Morley, S. D. rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLoS Comput. Biol.* **2014**, *10*, e1003571.

(8) Shen, C.; Wang, Z.; Yao, X.; Li, Y.; Lei, T.; Wang, E.; Xu, L.; Zhu, F.; Li, D.; Hou, T. Comprehensive assessment of nine docking programs on type II kinase inhibitors: prediction accuracy of sampling power, scoring power and screening power. *Brief. Bioinform.* **2018**,

(9) Wang, Z.; Sun, H.; Yao, X.; Li, D.; Xu, L.; Li, Y.; Tian, S.; Hou, T. Comprehensive evaluation of ten docking programs on a diverse set of protein–ligand complexes: the prediction accuracy of sampling power and scoring power. *Phys. Chem. Chem. Phys.* **2016**, *18*, 12964–12975.

(10) Bottegoni, G.; Rocchia, W.; Rueda, M.; Abagyan, R.; Cavalli, A. Systematic Exploitation of Multiple Receptor Conformations for Virtual Ligand Screening. *PLoS One* **2011**, *6*, e18845.

(11) Sherman, W.; Day, T.; Jacobson, M. P.; Friesner, R. A.; Farid, R. Novel Procedure for Modeling Ligand/Receptor Induced Fit Effects. *J. Med. Chem.* **2006**, *49*, 534–553.

(12) Ferrari, A. M.; Wei, B. Q.; Costantino, L.; Shoichet, B. K. Soft Docking and Multiple Receptor Conformations in Virtual Screening. *J. Med. Chem.* **2004**, *47*, 5076–5084.

(13) Shan, Y.; Kim, E. T.; Eastwood, M. P.; Dror, R. O.; Seeliger, M. A.; Shaw, D. E. How Does a Drug Molecule Find Its Target Binding Site? *J. Am. Chem. Soc.* **2011**, *133*, 9181–9183.

(14) Miller, E. B. et al. Reliable and Accurate Solution to the Induced Fit Docking Problem for Protein–Ligand Binding. *J. Chem. Theory Comput.* **2021**, *17*, 2630–2639.

(15) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber biomolecular simulation programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.

(16) Tanimoto, T. T. *An Elementary Mathematical Theory of Classification and Prediction*; International Business Machines Corporation, 1958.

(17) Damerau, F. J. A technique for computer detection and correction of spelling errors. *Commun. ACM* **1964**, *7*, 171–176.

(18) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J. Cheminform.* **2011**, *3*, 33.

(19) Landrum, G. RDKit: Open-source cheminformatics. 2022; `https://www.rdkit.org`.

(20) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophys. J.* **2015**, *109*, 1528–1532.

(21) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Comput. Biol.* **2017**, *13*, e1005659.

(22) Wójcikowski, M.; Zielenkiewicz, P.; Siedlecki, P. Open Drug Discovery Toolkit (ODDT): a new open-source player in the drug discovery field. *J. Cheminform.* **2015**, *7*, 26.

(23) Chen, T.; Guestrin, C. XGBoost. Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. New York, NY, USA, 2016; pp 785–794.

(24) Meli, R.; Biggin, P. C. spyrmsd: symmetry-corrected RMSD calculations in Python. *J. Cheminform.* **2020**, *12*, 49.

(25) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A. Automatic atom type and bond type perception in molecular mechanical calculations. *J. Mol. Graph. Model.* **2006**, *25*, 247–260.

(26) Roe, D. R.; Cheatham, T. E. PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data. *J. Chem. Theory Comput.* **2013**, *9*, 3084–3095.

(27) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. Graph.* **1996**, *14*, 33–38.

(28) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.

(29) Jakalian, A.; Jack, D. B.; Bayly, C. I. Fast, efficient generation of high-quality atomic charges. AM1-BCC model: II. Parameterization and validation. *J. Comput. Chem.* **2002**, *23*, 1623–1641.

(30) Verlet, L. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.* **1967**, *159*, 98–103.

(31) Andersen, H. C. Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations. *J. Comput. Phys.* **1983**, *52*, 24–34.

(32) Loncharich, R. J.; Brooks, B. R.; Pastor, R. W. Langevin dynamics of peptides: The frictional dependence of isomerization rates of N-acetylalanyl-N'-methylamide. *Biopolymers* **1992**, *32*, 523–535.

(33) Ballester, P. J.; Mitchell, J. B. O. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics* **2010**, *26*, 1169–1175.

(34) Yan, Y.; Wang, W.; Sun, Z.; Zhang, J. Z. H.; Ji, C. Protein–Ligand Empirical Interaction Components for Virtual Screening. *J. Chem. Inf. Model.* **2017**, *57*, 1793–1806.

(35) Kollman, P. A.; Massova, I.; Reyes, C.; Kuhn, B.; Huo, S.; Chong, L.; Lee, M.; Lee, T.; Duan, Y.; Wang, W.; Donini, O.; Cieplak, P.; Srinivasan, J.; Case, D. A.; Cheatham, T. E. Calculating Structures and Free Energies of Complex Molecules: Combining Molecular Mechanics and Continuum Models. *Acc. Chem. Res.* **2000**, *33*, 889–897.

(36) 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. *J. Comput. Aided. Mol. Des.* **1997**, *11*, 425–45.

(37) Murray, C. W.; Auton, T. R.; Eldridge, M. D. Empirical scoring functions. II. The testing of an empirical scoring function for the prediction of ligand-receptor binding affinities and the use of Bayesian regression to improve the quality of the model. *J. Comput. Aided. Mol. Des.* **1998**, *12*, 503–19.

(38) Rossato, G.; Ernst, B.; Vedani, A.; Smieško, M. AcquaAlta: A Directional Approach to the Solvation of Ligand–Protein Complexes. *J. Chem. Inf. Model.* **2011**, *51*, 1867–1881.

(39) Ross, G. A.; Morris, G. M.; Biggin, P. C. Rapid and Accurate Prediction and Scoring of Water Molecules in Protein Binding Sites. *PLoS One* **2012**, *7*, e32036.

(40) Asthagiri, D. N.; Paulaitis, M. E.; Pratt, L. R. Thermodynamics of Hydration from the Perspective of the Molecular Quasichemical Theory of Solutions. *J. Phys. Chem. B* **2021**, *125*, 8294–8304.

(41) Wójcikowski, M.; Ballester, P. J.; Siedlecki, P. Performance of machine-learning scoring functions in structure-based virtual screening. *Sci. Rep.* **2017**, *7*, 46710.

(42) Xu, Z. et al. Utilization of Halogen Bond in Lead Optimization: A Case Study of Rational Design of Potent Phosphodiesterase Type 5 (PDE5) Inhibitors. *J. Med. Chem.* **2011**, *54*, 5607–5611.

(43) Stebbins, C. E.; Russo, A. A.; Schneider, C.; Rosen, N.; Hartl, F.; Pavletich, N. P. Crystal Structure of an Hsp90–Geldanamycin Complex: Targeting of a Protein Chaperone by an Antitumor Agent. *Cell* **1997**, *89*, 239–250.

(44) Brough, P. A.; Barril, X.; Beswick, M.; Dymock, B. W.; Drysdale, M. J.; Wright, L.; Grant, K.; Massey, A.; Surgenor, A.; Workman, P. 3-(5-chloro-2,4-dihydroxyphenyl)-Pyrazole-4-carboxamides as inhibitors of the Hsp90 molecular chaperone. *Bioorg. Med. Chem. Lett.* **2005**, *15*, 5197–5201.

(45) Kadono, S. et al. Structure of human factor VIIa/tissue factor in complex with a peptide-mimetic inhibitor: high selectivity against thrombin by introducing two charged groups in P2 and P4. *Acta Crystallogr. Sect. F Struct. Biol. Cryst. Commun.* **2005**, *61*, 169–173.

(46) Vijaykumar, D.; Sprengeler, P. A.; Shaghafi, M.; Spencer, J. R.; Katz, B. A.; Yu, C.; Rai, R.; Young, W. B.; Schultz, B.; Janc, J. Discovery of novel hydroxy pyrazole based factor IXa inhibitor. *Bioorg. Med. Chem. Lett.* **2006**, *16*, 2796–2799.

(47) Schweitzer, B. A.; Neumann, W. L.; Rahman, H. K.; Kusturin, C. L.; Sample, K. R.; Poda, G. I.; Kurumbail, R. G.; Stevens, A. M.; Stegeman, R. A.; Stallings, W. C.; South, M. S. Structure-based design and synthesis of pyrazinones containing novel P1 'side pocket' moieties as inhibitors of TF/VIIa. *Bioorg. Med. Chem. Lett.* **2005**, *15*, 3006–3011.

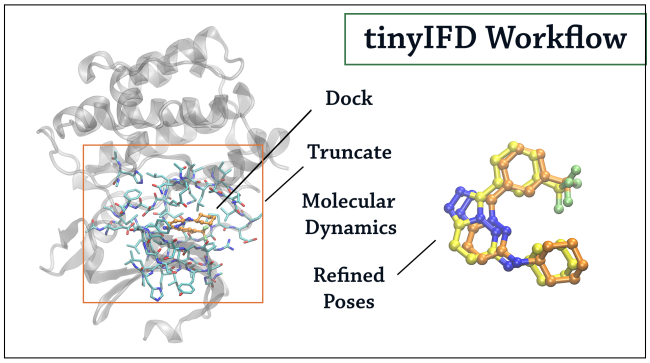
(48) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics* **2009**, *10*, 168.

(49) Broomhead, N. K.; Soliman, M. E. Can We Rely on Computational Predictions To Correctly Identify Ligand Binding Sites on Novel Protein Drug Targets? Assessment of Binding Site Prediction Methods and a Protocol for Validation of Predicted Binding Sites. *Cell Biochem. Biophys.* **2017**, *75*, 15–23.

(50) Wang, Y.; Lupala, C. S.; Liu, H.; Lin, X. Identification of Drug Binding Sites and Action Mechanisms with Molecular Dynamics Simulations. *Curr. Top. Med. Chem.* **2019**, *18*, 2268–2277.

(51) Kuzmanic, A.; Bowman, G. R.; Juarez-Jimenez, J.; Michel, J.; Gervasio, F. L. Investigating Cryptic Binding Sites by Molecular Dynamics Simulations. *Acc. Chem. Res.* **2020**, *53*, 654–661.

TOC Graphic


tinyIFD Workflow
Dock
Truncate
Molecular Dynamics
Refined Poses