Supplementary Information (SI) for Chemical Science.
This journal is © The Royal Society of Chemistry 2024

Supplementary Information

PharmacoNet: deep learning guided pharmacophore modeling for ultra-large-scale virtual screening

Seonghwan Seo[1] and Woo Youn Kim[123*]

[1]*Department of Chemistry, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea*
[2]*Graduate School of Data Science, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea*
[3]*HITS Inc, 28, Teheran-ro 4-gil, Gangnam-gu, Seoul 06234, Republic of Korea*

{shwan0106, wooyoun*}@kaist.ac.kr

July, 2024

Contents

1 Supplementary Figures and Tables

1.1 Fig. S1: Score distribution for hot spot detection.

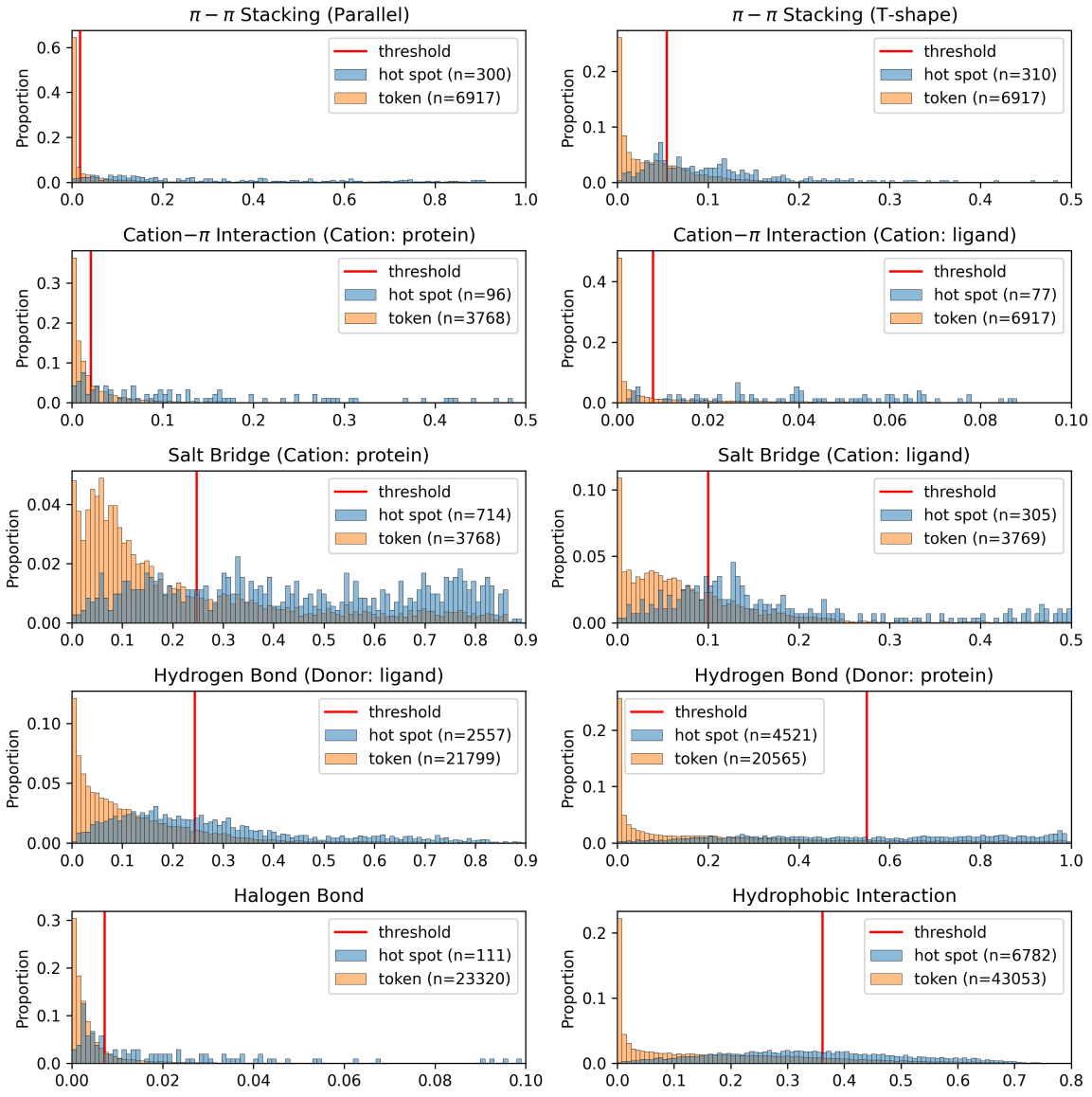

Figure S1: **Score distribution for hot spot detection.** The sigmoid score distribution of tokens in the validation set. The blue histograms mean the score distribution of hot spots, and the orange ones mean the score distribution of all tokens within cavities. The numbers in the parentheses indicate the number of tokens. The red line denotes the threshold for hot spot prediction.

1.2 Fig. S2: Comparison with molecular docking.

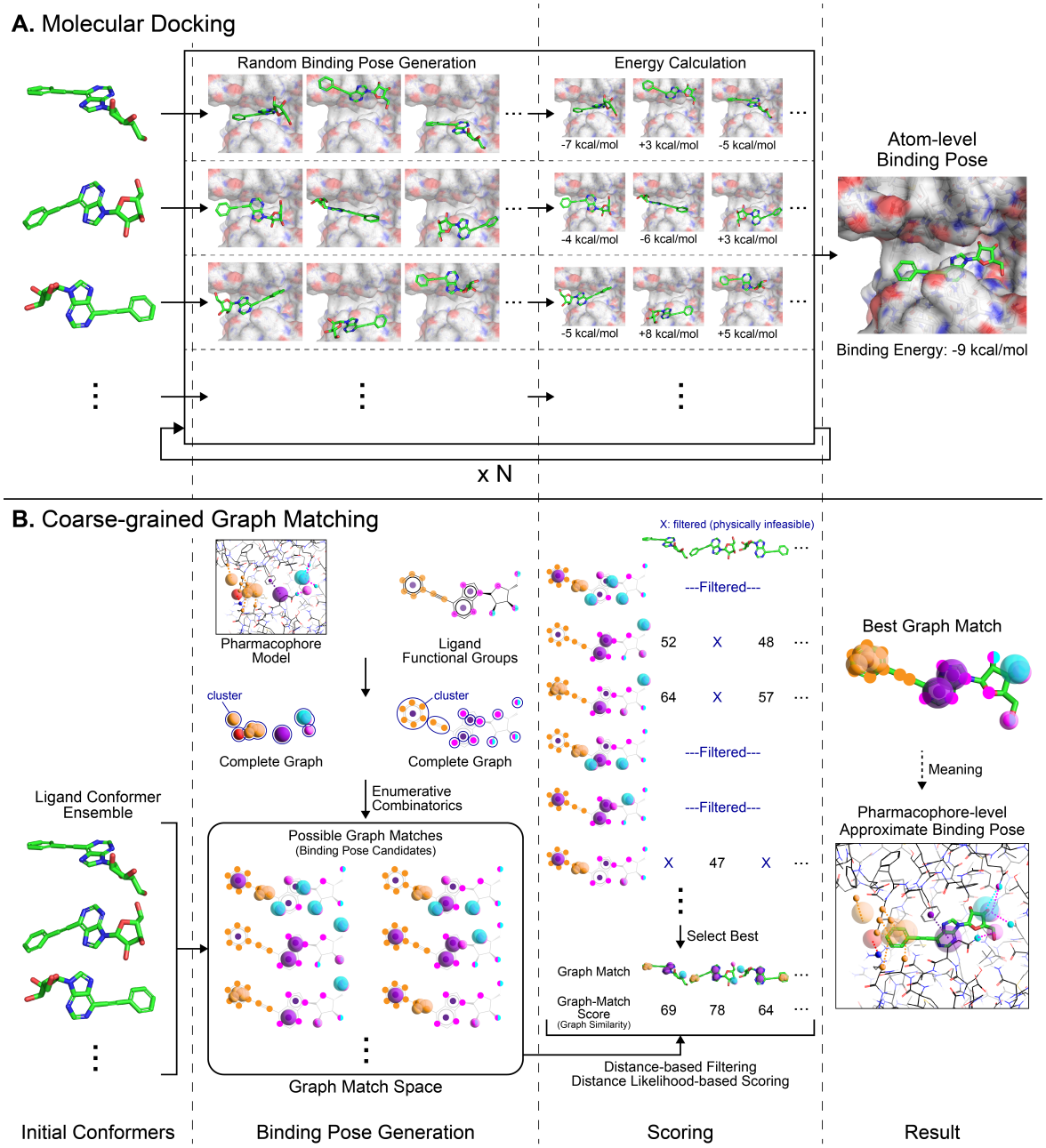

Figure S2: **Comparison with molecular docking. A.** The overall scheme of molecular docking. **B.** The overall scheme of graph matching.

1.3 Fig. S3: Additional metrics on DEKOIS2.0 screening benchmark.

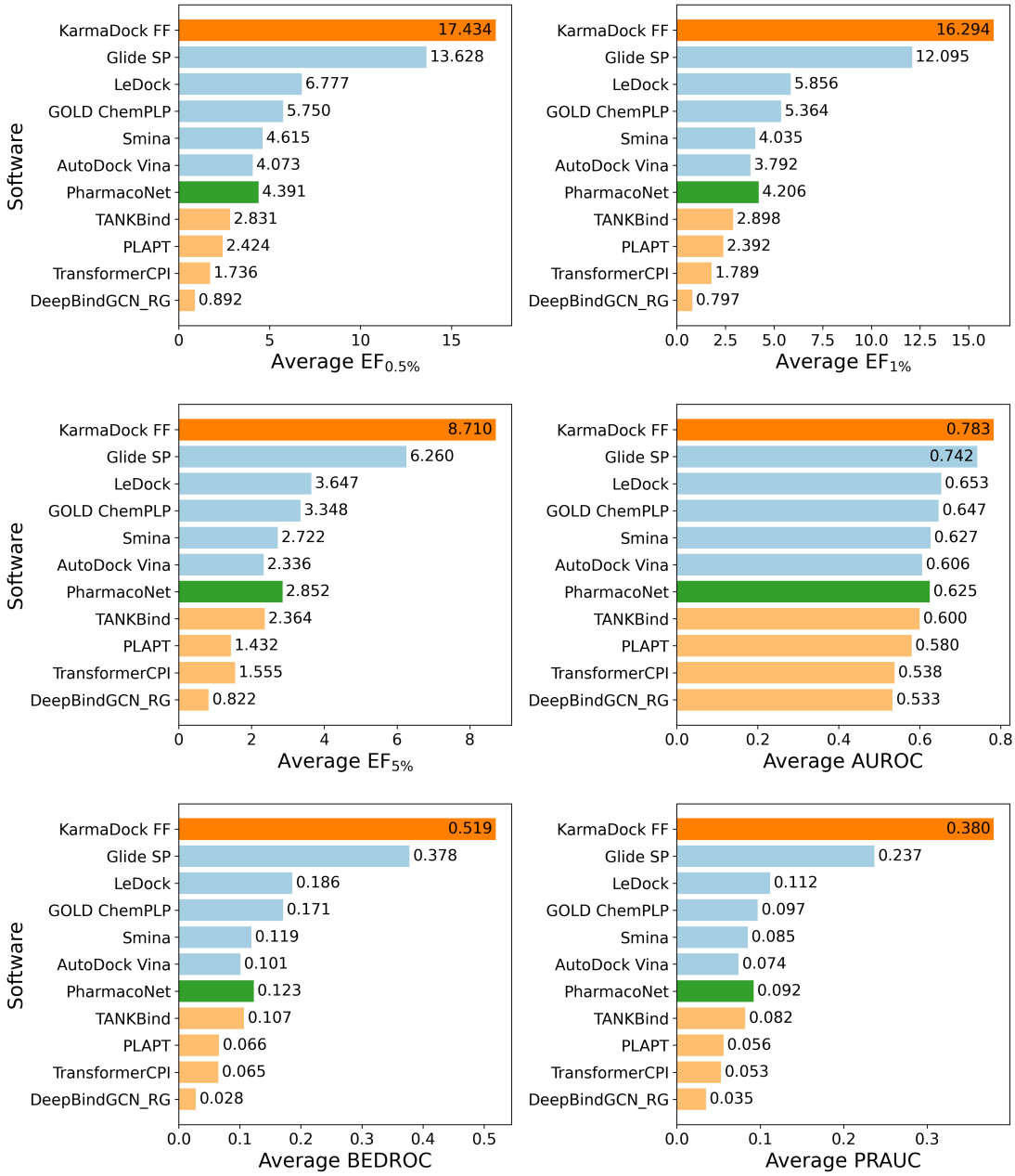

Figure S3: **DEKOIS2.0 benchmark study**. Average screening powers for conventional docking softwares (blue), docking-free DL scoring methods (light orange), DL-based docking method (deep orange), pharmacophore modeling-based methods (light green), and PharmacoNet (deep green).

1.4 Fig. S4: Protein sequence similarity between the training set and test sets.

Figure S4: **Sequence similarity** between 19,400 proteins in the PDBbind v2020 training set and 81 proteins in the DEKOIS2.0. The four proteins for the out-of-distribution case study (Fig. 2E) are highlighted in blue.

1.5 Fig. S5: Chemical structural similarity between the training set and test sets.

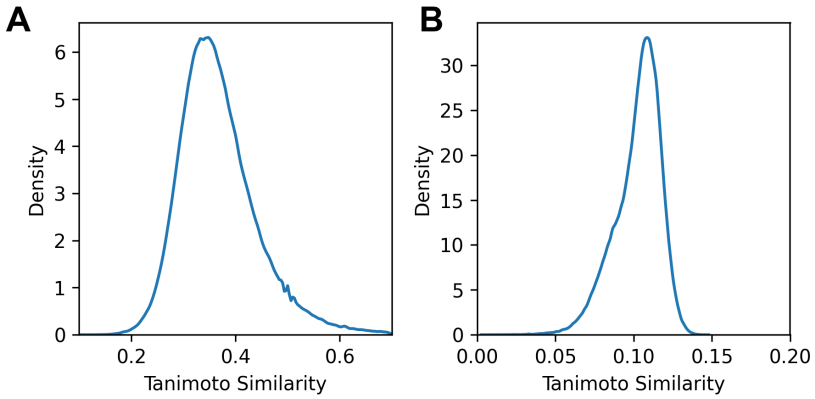

Figure S5: **Structural similarity** between ligands in the PDBbind v2020 training set and LIT-PCBA ligands used in the out-of-distribution case study (Fig. 2F). **A.** Distribution of the highest Tanimoto similarity to the training set. **B.** Distribution of the average Tanimoto similarity to the training set.

1.6 Table S1: Runtime benchmark with various docking methods

Table S1: **Runtime benchmark with various docking methods.** Average runtime per the prediction of conventional docking programs and several deep learning models. Since TANKBind is divided into two steps (1) DL-based scoring and then (2) binding pose prediction, we reported the runtime for scoring-only[2] (0.28 (s)) and for the whole process including binding pose prediction[1] (0.28 (s) + 0.44 (s)) separately. We used a single 32-core Intel Xeon Gold 6326 CPU @ 2.90 GHz for conventional docking programs and PharmacoNet.

Methods	Type	Environment	Avg. Runtime (s)
AutoDock Vina	Atomistic pose prediction & Scoring	32-core CPU	14.823
Smina		32-core CPU	12.678
TANKBind[1]		Tesla A100	0.72
KarmaDock		Tesla V100	0.017
EquiBind	Atomistic pose prediction	GTX 1060	0.04
DiffDock		Tesla A100	10/40
TANKBind[2]	Scoring only	Tesla A100	0.28
DeepBindGCN-RG		RTX A4000	0.0075
PharmacoNet	Coarse-grained pose prediction & Scoring	32-core CPU	**0.00089**

1.7 Table S2: Hyperparameters

Table S2: **Hyperparameters.**

Hyperparameter	Value
Batch Size	8
Learning Rate	1e-4
Optimizer	AdamW
Weight Decay	0.05
Eps	1e-8
Betas	(0.9, 0.999)
LR Schedular	LinearLR (start: 1e-3, total iters: 1,000)
	+ StepLR (steps: 50,000, gamma: 0.2)
Data Augmentation	Random Translation (3.0 Å), Random Rotation
Model Hidden Dimensions	SwinV2-T
Model Size	31M
Input Voxelization Kernel	Gaussian
Output Voxelization Kernel	Binaray
Voxelization Resolution	0.5 Å
Voxelization Dimension	$(64 \times 64 \times 64)$

2 Additional Results

2.1 Comparison with structure-based binding conformation prediction methods

In this section, we compared the average runtimes of PharmacoNet against those of the docking programs on both the PDBbind core set [1] and the refined set when the initial conformers of each ligand were provided. Moreover, we also compared the speed of several non-structure-based DL models and DL-based molecular docking methods [2–6] on PDBbind time split test set proposed by [2]. For all benchmark tests, we considered 8 conformers for each ligand for a fair comparison with AutoDock Vina and Smina, which perform pose searches with a default exhaustiveness of 8. It should be noted that the unique NCI-aware scoring strategy of PharmacoNet makes it faster than DL-based binding conformation prediction methods and non-structure-based PLI prediction models running on GPUs (Supplementary Table S1†). While these DL models can be accelerated with efficient parallel computations on GPUs, PharmacoNet's absolute computational cost is very small. As a result, PharmacoNet was 19x faster than KarmaDock [5], the fastest model with binding pose prediction, and 8x faster than DeepBindGCN-RG [6], the fastest model without binding pose prediction, even though both models were evaluated on GPUs, whereas PharmacoNet was performed on a CPU.

3 Training details

The prediction of binary masks for each pharmacophore point is carried out using a 3D-CNN which is computationally expensive. As a result, we train the model by sampling up to 6 pharmacophore points for each complex in each training iteration. Furthermore, the number of protein hotspots (instances) is considerably smaller compared to the total number of FGs (tokens) present in a given binding site. Consequently, we sample up to 200 tokens for each complex, with priority given to protein hotspots, tokens within cavities, and remaining tokens to reduce the data imbalance. The hyperparameters are explained in Table SS2. The model training took about 48 hours on 4 NVIDIA RTX 3090 24GB GPUs.

4 Baseline of benchmark studies

Gold, LeDock, Glide SP. Since those commercial docking programs require licenses, we reused the numbers reported in Zhang et al. [5]

TransformerCPI, KarmaDock. We reused the numbers reported in Zhang et al. [5].

TANKBind, EquiBind, DiffDock. We reused the numbers reported in Corso et al. [4]. For the screening power of TANKBind, we reused the numbers reported in Zhang et al. [5].

DeepBindGCN-RG. Since DeepBindGCN-RG is implemented in an older version of torch geometric, we manually modified the names of parameters associated with some layers included in the saved model file.

AutoDock Vina. The data preparation process for molecular docking has a significant impact on performance. However, previous studies did not report detailed benchmark settings, which has led us to report the preparation and docking parameters for a reproducible benchmark comparison.

For proteins, We first removed hetero atoms in protein pdb files, then converted them to a pdbqt format with AutoDockTools [7]. For ligands, mol2 or pdb files were converted to pdbqt files with Open Babel [8] and AutoDockTools. We used processed ligand and protein pdbqt files for AutoDock Vina with an exhaustiveness of 8. The search box size is (30, 30, 30), and its center is the center of the reference ligand.

Smina. We removed hetero atoms in protein pdb files. We used ligand sdf files and processed protein pdb files for Smina. Smina was then run with protein pdb files and ligand sdf files under the auto-box setting using the reference ligands and default exhaustiveness of 8. The size and center of the search box are determined with the auto-box setting of the reference ligand.

5 Graph matching algorithm

5.1 Clustering algorithm for ligand pharmacophore arrangements

We perform clustering for ligand pharmacophores in the same functional groups according to the following criteria:

- The cations, anions, H-bond acceptors, and H-bond donors in the same functional group are grouped.
- The aromatic ring and hydrophobic carbons in the same functional group are grouped.
- The hydrophobic carbons in the same carbon chain are grouped.

5.2 Clustering algorithm for pharmacophore model

We perform clustering for pharmacophore points accodring to the following process:

- The Cation-type points and Anion-type points in 1.5 Å are grouped, respectively. For each cation/anion-type point, near H-bond donor/acceptor-type points within 3.0 Å are grouped together.
- The Aromatic-type points in 1.5 Å are grouped. For each aromatic-type point, near HydrophobicCarbon-type points within 3.0 Å are grouped together.
- The H-bond donor/acceptor-type points in 3.0 Å are grouped.
- The Hydrophobic-type points in 3.0 Å are grouped.

6 Software details

ETKDG conformer generation. We used the ETKDG [9] version 3 (small ring) implemented in RDKit [10].

Protein-ligand interaction profiler. To detect non-covalent bonds from the structure of a protein-ligand complex in PDBBind v2020, we used the Protein-Ligand Interaction Profiler (PLIP) [11] and Open Babel [8]. For flexible and fast data processing during the model training, we reimplemented it while keeping the original PLIP rules except for the water bridge and the metal bridge.

PIGNet2. For fine-screening evaluation methods, we used PIGNet2 [12], the state-of-the-art scoring DL model. For PIGNet2, we used 10 Smina docking poses. The estimated binding affinity is an ensemble of predictions from 4 pre-trained models.

MolVoxel. We developed MolVoxel, a Python library with minimal dependencies to enable on-the-fly voxelization in various environments including cheminformatics and deep learning applications. Currently, it supports NumPy, Numba, and PyTorch (with CUDA support). Users can convert from the point clouds to voxel images with two kernels: Gaussian and binary. The MolVoxel is accessible at PyPI.

7 Virtual screening pipeline with PharmacoNet

In this section, we provide simple tutorials to use PharmacoNet. PharmacoNet is open-source and available at `https://github.com/SeonghwanSeo/PharmacoNet`.

7.1 Installation

```
# Clone the package from github
git clone https://github.com/SeonghwanSeo/PharmacoNet.git

# Create conda environments
conda create -f environment.yml
conda activate pmnet
pip install torch
pip install .
```

7.2 Protein-based pharmacophore modeling

```
# Use RCSB PDB Id
python modeling.py --pdb 3ug2 --cuda

INFO:root:Load PharmacoNet finish
INFO:root:Load result/3ug2/3ug2.pdb
INFO:root:A total of 2 ligand(s) are detected!
Ligand 1
- ID      : IRE (Chain: B [auth A])
- Center  : -0.184, 49.350, 20.022
- Name    : GEFITINIB

Ligand 2
- ID      : MES (Chain: C [auth A])
- Center  : 16.580, 45.572, 24.179
- Name    : 2-(N-MORPHOLINO)-ETHANESULFONIC ACID

INFO:root:Select the ligand number(s) (ex. 2 ; 1,2 ; manual ; all ; exit)
ligand number:1 # Enter the ligand number for binding site detection
Ligand 1
- ID      : IRE (Chain: B [auth A])
- Center  : -0.184, 49.350, 20.022
- Name    : GEFITINIB
INFO:root:Save Pharmacophore Model to result/3ug2/3ug2_B_IRE_model.pm
INFO:root:Save Pymol Visualization Session to result/3ug2/3ug2_B_IRE_model_pymol.pse

# Use custom protein file and reference ligand file
python modeling.py --protein <PROTEIN_PATH>
--ref_ligand <LIGAND_PATH> --prefix <PREFIX>

# Use custom protein file and xyz coordinates.
python modeling.py --protein <PROTEIN_PATH> --center <X> <Y> <Z>
--prefix <PREFIX>
```

7.3 Virtual screening

```
# Unzip example library file
tar -xf examples/library.tar

# Run virtual screening
python screening.py -p ./result/3ug2/3ug2_B_IRE_model.pm
--library ./library --out ./output.csv --cpus 4
```

8 Python interface of PharmacoNet

For application in various machine learning and cheminformatics applications, PharmacoNet provides the simple Python API. PharmacoNet provides three functionalities: 1) automated protein-based pharmacophore modeling 2) pharmacophore-based virtual screening 3) pre-trained protein representation extraction.

8.1 Pharmacophore modeling

```
from pmnet.module import PharmacoNet

# Load PharmacoNet
pmnet = PharmacoNet(<device>) #('cuda' or 'cpu')

# Pharmacophore Modeling with reference-ligand information
model = pmnet.run(<protein_path>, <ref_ligand_path>)
# pharmacophore Modeling with center
model = pmnet.run(<protein_path>, center=<center>)
# save pharmacophore model (.pm/.json formats)
model.save(<save_path>)
```

8.2 Ligand scoring for virtual screening

```
from pmnet import PharmacophoreModel

# Load Pharmacophore Model
model = PharmacophoreModel(<pharmacophore_model_path>)

# Ligand Scoring
model.scoring_file(<ligand_sdf_file>)
```

8.3 Protein feature extraction

```
from pmnet.module import PharmacoNet

# Load PharmacoNet
pmnet = PharmacoNet(<device>, 0.5)

# From reference-ligand information
out = module.feature_extraction(<protein_path>, <ref_ligand_path>)
# From center
out = module.feature_extraction(<protein_path>, center=<center>)

multi_scale_feature_maps, hotspot_infos = out
```

References

[1] Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59 (2):895–913, 2018.

[2] Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *International conference on machine learning*, pages 20503–20521. PMLR, 2022.

[3] Wei Lu, Qifeng Wu, Jixian Zhang, Jiahua Rao, Chengtao Li, and Shuangjia Zheng. Tankbind: Trigonometry-aware neural networks for drug-protein binding structure prediction. *Advances in neural information processing systems*, 35:7236–7249, 2022.

[4] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi S. Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=kKF8_K-mBbS`.

[5] Xujun Zhang, Odin Zhang, Chao Shen, Wanglin Qu, Shicheng Chen, Hanqun Cao, Yu Kang, Zhe Wang, Ercheng Wang, Jintu Zhang, et al. Efficient and accurate large library ligand docking with karmadock. *Nature Computational Science*, 3(9):789–804, 2023.

[6] Haiping Zhang, Konda Mani Saravanan, and John ZH Zhang. Deepbindgcn: Integrating molecular vector representation with graph convolutional neural networks for protein–ligand interaction prediction. *Molecules*, 28(12):4691, 2023.

[7] Ruth Huey, Garrett M Morris, and Stefano Forli. Using autodock 4 and autodock vina with autodocktools: a tutorial. *The Scripps Research Institute Molecular Graphics Laboratory*, 10550(92037):1000, 2012.

[8] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):1–14, 2011.

[9] Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12):2562–2574, 2015.

[10] Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.

[11] Sebastian Salentin, Sven Schreiber, V Joachim Haupt, Melissa F Adasme, and Michael Schroeder. Plip: fully automated protein–ligand interaction profiler. *Nucleic acids research*, 43(W1):W443–W447, 2015.

[12] Seokhyun Moon, Sang-Yeon Hwang, Jaechang Lim, and Woo Youn Kim. Pignet2: a versatile deep learning-based protein–ligand interaction prediction model for binding affinity scoring and virtual screening. *Digital Discovery*, 3(2):287–299, 2024.