SOFTWARE

Open Access

GNINA 1.3: the next increment in molecular docking with deep learning



Andrew T. McNutt¹, Yanjing Li², Rocco Meli^{3,4}, Rishal Aggarwal¹ and David Ryan Koes^{1*}

Abstract

Computer-aided drug design has the potential to significantly reduce the astronomical costs of drug development, and molecular docking plays a prominent role in this process. Molecular docking is an *in silico* technique that predicts the bound 3D conformations of two molecules, a necessary step for other structure-based methods. Here, we describe version 1.3 of the open-source molecular docking software GNINA. This release updates the underlying deep learning framework to PyTorch, resulting in more computationally efficient docking and paving the way for seamless integration of other deep learning methods into the docking pipeline. We retrained our CNN scoring functions on the updated CrossDocked2020 v1.3 dataset and introduce knowledge-distilled CNN scoring functions to facilitate high-throughput virtual screening with GNINA. Furthermore, we add functionality for covalent docking, where an atom of the ligand is covalently bound to an atom of the receptor. This update expands the scope of docking with GNINA and further positions GNINA as a user-friendly, open-source molecular docking framework. GNINA is available at https:// github.com/gnina/gnina.

Scientific contributions: GNINA 1.3 is an open source a molecular docking tool with enhanced support for covalent docking and updated deep learning models for more effective docking and screening.

Keywords Molecular docking, Deep learning, Structure-based drug design

Introduction

The development of new drugs is a complex and timeconsuming process [27], requiring the evaluation of large numbers of compounds to identify those with therapeutic potential. Molecular docking, a *in silico* technique that models the 3D binding conformation of small molecules to proteins, is a key tool for accelerating this process [22]. Predicting the binding conformation of small molecules

*Correspondence:

¹ Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA, USA

⁴ Present Address: Swiss National Supercomputing Center (CSCS), ETH Zurich, 6900 Lugano, Switzerland to their target proteins enables prioritization of compounds for experimental testing along with enabling other *in silico*, structure-based methods such as lead optimization and binding affinity prediction.

One widely used, open-source molecular docking pipeline is GNINA [18], a fork of Autodock Vina [33] and Smina [13]. The docking workflow follows a conventional setup, where ligand conformational sampling is carried out via a set of Markov chain Monte Carlo (MCMC) chains that randomly perturb the ligand in the specified binding site. Following sampling, protein-ligand conformations are scored and ranked with the top poses output to the user. GNINA distinguishes itself from its predecessors by using convolutional neural network (CNN) scoring functions that work on an atomic density grid representation (i.e., a 3D "picture" of the complex) within the docking workflow [25]. The ligand poses from the MCMC chains are first minimized with



© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

David Ryan Koes

dkoes@pitt.edu

² Computational Biology Department, Carnegie Mellon University, Pittsburgh, PA, USA

³ Department of Biochemistry, University of Oxford, Oxford OX1 3QU, UK

respect to the Autodock Vina scoring function, and then rescored and ranked using the CNN scoring functions. An ensemble of CNN scoring functions of differing computational complexity is used to score the ligand poses, which enhances the binding pose prediction at the cost of additional computation. GNINA has performed well in prospective applications [14] and independent evaluations consistently find it outperforms Vina and achieves similar performance to commercial tools [7]. Recent works have also shown that the performance of GNINA can be further boosted through the use of multiple conformers of the small molecule [19].

We present incremental improvements to the docking pipeline resulting in GNINA 1.3. These changes include the introduction of covalent docking capabilities, retraining of the CNN scoring function on updated datasets for higher quality models, and the development of knowledge distilled CNN scoring functions for faster scoring. Furthermore, we establish GNINA as a platform to enable deep learning development in docking by integrating PyTorch as the supported deep learning framework. These enhancements expand the scope, accuracy, and computational efficiency of GNINA, further solidifying its position as a valuable, open-source tool in the pursuit of computationally developed therapeutics.

Implementation

Caffe replaced with PyTorch

GNINA 1.0 uses the venerable Caffe [12] C++ deep learning framework to implement its convolutional neural network scoring. Since the initial development of GNINA's CNN scoring model [25], more flexible, powerful, and popular deep learning frameworks have been released. Specifically, the PyTorch [23] framework has come to dominate the deep learning community, with more than 90% of models on the popular HuggingFace model sharing site being PyTorch exclusive. PyTorch, and the underlying PyTorch C++ backend, supports a robust ecosystem of developers and users and provides a flexible, auto-differentiation based approach that enables rapid prototyping and the development of sophisticated model architectures. With GNINA 1.3, Caffe has been replaced with PyTorch. This introduces no changes to typical usage, but makes it easier for advanced users to integrate their own PyTorch trained models into a conventional docking workflow and sets the stage for more substantive changes in future GNINA releases, such as augmenting the Monte Carlo sampling with deep neural network directed sampling [4, 7].

Retrained models

The CrossDocked2020 dataset [9] used for training of the GNINA CNN scoring functions has been updated to version 1.3 since the initial GNINA 1.0 models were trained. The updated version 1.3 addresses ligand and receptor misalignment problems and incorrect bond typing problems present in earlier versions (statistics of the updated datasets are provided in Table S1 and Figure S1). All models trained on CrossDocked2020 ReDocked2020, a redocked-only subset or of CrossDocked2020, [9] were retrained on the updated version of their corresponding dataset. Models input a 3D grid of Gaussian atom-type densities generated by the libmolgrid library [30]. All models are trained for two tasks: pose scoring and binding affinity prediction. The pose score is trained to classify if a pose is $\leq 2\text{Å}$ RMSD from the ground truth using a cross entropy loss function. The binding affinity is trained with a mean squared error loss between the predicted and ground-truth affinity that is hinged if the pose is inaccurate. Further training details and hyperparameters are provided in the supplement.

After retraining the models, we greedily selected an ensemble of models with the best performance on both the redocking and cross-docking tasks following the Default Ensemble selection procedure enumerated in McNutt et al. [18]. This results in an ensemble of three models compared to the default GNINA 1.0 ensemble, which has five models.

Knowledge distillation for faster screening

McNutt et al. [18] found that ensembles of CNN scoring functions always produced higher quality docked poses than a single CNN scoring function when used in the GNINA docking pipeline. However, utilizing an ensemble of CNN scoring functions incurs a greater computational cost than using a single CNN scoring function. This extra computational burden is especially egregious when running GNINA without a GPU (458 s and 72 s for the best ensemble and single model, respectively in GNINA 1.0), a common scenario when utilizing GNINA for high throughput screening. Knowledge Distillation (KD) is a technique to condense the knowledge of a large "teacher" model into a smaller "student" model, enabling faster inference with similar model performance [10]. Ensemble KD transfers the knowledge learned by multiple teacher models to a single student model by minimizing the discrepancy between the average representation of the teachers and the student [2, 32]. Ensemble KD can reduce the computational overhead of workflows that use an ensemble of large models without significantly impacting performance.

There are four different CNN models for molecular docking within GNINA that differ in their model architecture and training set [18]. The two architectures are "Default 2018", a linear CNN with five convolutional layers, and "Dense", which has twelve convolutional layers

organized into three densely connected blocks [11]. In addition to the full CrossDocked2020 dataset, models are also trained on a subset that consists of only redocked poses: ReDocked2020 [9]. Each CNN model has five variants that only differ in their training initialization (random seed). These five variants form an ensemble for each CNN model. We utilize ensemble KD to compress the ranking performance of the ensemble of five variants into a single student model with the same architecture (Fig. 1). Additionally, we consider one more ensemble of the CNN models: "All Default2018 Ensemble", consisting of all CNN models with the Default2018 architecture. Default GNINA docking only utilizes the pose score



Fig. 1 Knowledge distillation condenses the pose scoring power of the teacher ensemble into a single student model. The student model is trained to reproduce the pre-softmax pose score logits of the ensemble of teacher models and simultaneously trained on the ground truth pose and affinity labels. The student model is then used to rescore and rank poses in the GNINA docking pipeline to speed up docking

of the CNN models, therefore our distillation only considers the pose score with our KD loss being the sum of Kullback–Leibler (KL) divergence of the pre-softmax values of the pose score between the student and each teacher. The total training loss is a sum of the KD loss and the ground truth affinity and pose classification losses. Training is carried out on the same training dataset as the teachers. For the "All Default2018 Ensemble", we train the student on the CrossDocked2020 v1.3 dataset since this is largely a superset of the training datasets used for the

Default2018 models (CrossDocked2020, ReDocked2020, and PDBBind General v2016). This leads to the creation of 6 CNN scoring functions distilled from ensembles.

More details about the training and hyperparameters of the ensemble KD can be found in McNutt et al. [17].

Covalent docking

GNINA 1.3 provides a simple interface for covalent docking, as shown in Fig. 2. Instead of presuming a particular chemical reaction, GNINA expects the bound, covalent form of the ligand to be provided as input (as is the case with other programs [1, 3, 15, 34, 35]). The user then specifies the ligand atom, using a SMARTS expression, and a receptor atom, using the chain, residue ID, and atom name. If multiple ligand atoms match the SMARTS expression, all pairings of ligand and receptor atoms are evaluated, resulting in a corresponding expansion of the number of output poses. Given a pairing of receptor and ligand atoms, the ligand is re-positioned so that the ligand atom is within bonding distance of the receptor atom, the bond is created with a user configurable bond order (default of one), and the residue-ligand construct is treated as one flexible residue while docking. That is, the internal torsion angles are sampled and optimized during Monte Carlo



Fig. 2 Covalent docking with GNINA. The input ligand must be provided as conformation representative of the bound form the ligand, including any chemical modifications (e.g. epoxide ring opening). The covalent atom on the ligand is specified with a SMARTS expression; all matching atoms are evaluated. The covalent atom on the receptor is specified with the chain identifier, residue number, and atom name. Additional optional arguments refine the positions and treatment of the covalent bond

sampling and energy minimization, but no rigid body transformations are performed. For purposes of CNN scoring, which treats receptor and ligand atoms as having different types, the ligand atoms remain identified as ligand atoms. In order to position the ligand at a reasonable location, by default the OpenBabel [21] GetNewBondVector heuristic is applied to the receptor atom (after reducing the number of hydrogens) to identify a logical placement of the ligand covalent atom. Alternatively, this position can be manually specified. The OpenBabel method OBBuilder::Connect is then used rotate and translate the ligand such that the covalent ligand atom is positioned appropriately and the bonding geometry is reasonable. Optionally, the entire residue-ligand construct can be optimized using the UFF force field to further refine the bonding geometry.

Results

We enumerate the improved performance, both in terms of run-time, cross-docking pose prediction accuracy, and virtual screening of GNINA 1.3.

Docking runtime is reported as the average time to dock a protein-ligand complex, computed over a random 100 complex subset of the PDBbind core set (further detailed in the supplement). Pose prediction accuracy is measured via TopN, defined as the percentage of protein-ligand complexes where a ≤ 2 Å RMSD pose is found within the top N ranked poses. Virtual screening metrics are described in Sect. Virtual Screening.

Torch performance

Docking is often used for virtual screening of large libraries, which requires a scoring function that is fast without compromising accuracy. We benchmark the GNINA CNN models on a random 100 complex subset of the PDBbind core set v.2016 [29] to determine their computational cost (details of the benchmarking can be found in the supplement). Replacing the Caffe models with a PyTorch implementation of the same models produces no change in pose performance, but does result in a significant runtime performance improvement in CPU-only mode as shown in Fig. 3. Average docking time reduces from 129 s to about 30 s per complex when no GPU is used during docking. This is in part due to better support for multiprocessing in PyTorch. For our benchmarking we limited GNINA to using four cores, therefore the performance benefit is potentially even greater than shown in Fig. 3 for many-core systems (Figure S2).

Updated models

We consider the performance of our updated models both at pose prediction and virtual screening.

Pose prediction

We consider two tasks: redocking and cross-docking. Redocking, removing a ligand from a complex structure and docking it back in place, provides an easily verifiable benchmark for molecular docking methods, while crossdocking represents a realistic use case of molecular docking: docking a ligand to a non-cognate receptor. For the cross-docking evaluations, we utilize the Wierbowski



19

without a GPU (note that the y-axis has different scales). Both the 1.3 Default Ensemble and the fast model sit on the Pareto-frontier of the docking accuracy and computational cost curve. Results for redocking performance are provided in Figure S6 and Table S7

et al. [36] cross-docking dataset. The redocking evaluations utilize the Posebusters benchmark set and the Astex diverse set as defined in Buttenschoen et al. [5]. Further dataset information is provided in Table S4. We find that all of the retrained models rank poses more accurately when cross-docking, but the retrained redock_ default2018 models are about the same at pose ranking for redocking (Figure S3 and S4). These improvements are due to the updated CrossDocked2020 dataset. We see additional improvements through ensemble knowledge distillation; while the distilled models are not as good as the full ensemble, they are better than any single un-distilled model (Table S5 and S6).

The updated default ensemble is composed of a retrained dense model, a knowledge distilled dense model, and a knowledge distilled crossdock_default2018 model (all models are trained on the full CrossDocked data set). We see in Fig. 4 that the new GNINA 1.3 Default Ensemble ranks cross-docked poses better than the 1.0 Default Ensemble for all N, increasing Top1 from 37% to 40%, and is faster with an average CPU-only time of 23 s compared to 30 s using the 1.0 Default Ensemble. However, redocking Top1 drops slightly on both datasets (Figure S5), decreasing from 69% to 67% on the Posebusters Benchmark set.

A new feature in GNINA 1.3 is a "fast" single model, the best performing Default2018 model. This model was distilled from the "All Default2018 Ensemble", which consists of all models trained using this architecture. This model is enabled with the command-line option –cnn=fast and is intended to be used during high-throughput screening. As shown in Fig. 3, the fast model has only slightly decreased TopN compared to the 1.0 Default Ensemble



Fig. 4 Cross-docking performance of the GNINA scoring functions on the Wierbowski et al. [36] dataset

when cross-docking, but is significantly faster with an average CPU-only time of 16 s, only 1.3s slower than using the Vina empirical scoring function and less than 1 s slower than when using a GPU (Table S7). We see a larger gap in performance between the 1.0 Default Ensemble and the fast model on redocking (Top1 of 69% and 64% for the 1.0 and fast model, respectively).

Virtual screening

Retrospective virtual screening results for GNINA 1.3 on the DUD-E [20] benchmark are shown in Fig. 5. Compounds are ranked using the pose score (CNNscore). We note that while there are known biases in the DUD-E benchmark that complicate evaluation of machine learned scoring function [6, 28], GNINA was not trained on DUD-E data and so is not directly effected by these biases. Both the area under the receiver operating characteristic (AUC) and the enrichment factor [24] at 1% (EF1%) are reported. EF1% measures the ratio of active compounds ranked in the top-1% of a virtual screen to a random selection of the database with the same size. As the enrichment factor is sensitive to class imbalances, we normalize by the best possible EF1% so the metric (denoted nEF1%) is comparable across targets [31]. GNINA 1.3 generally outperforms 1.0, with a median AUC and nEF1% of 0.78 and 0.27 compared to 0.75 and 0.25 for GNINA 1.0. GNINA 1.3 improves upon 1.0 for 68 of the 102 targets. The single, 'fast' 1.3 model has comparable AUCs to 1.0, but worse enrichment factors.

Covalent docking

To evaluate the new covalent docking feature in GNINA 1.3, we use a benchmark of 207 complexes from Scarpino et al. [26]. Use of this covalent redocking benchmark allows us to compare to previously evaluated approaches in Fig. 6. We consider two scenarios: default covalent docking where a generated conformer of the ligand is used with no additional positioning information, and docking the experimental conformer with a precisely specific location of the covalent ligand atom. This provides the expected range of performance depending on the amount of prior information available; results for in-between settings can be found in Figure S7. The success rate for GNINA ranges from the worst (36.2%) to the best (66.6%) depending on the settings used. Using the Vina scoring function results in significantly better performance than the CNN. This is unsurprising, as the CNN was not trained on any covalent complexes, and points to a common pitfall of applying models outside their domain of applicability. Using CNN scoring on this same benchmark but without covalent docking does outperform Vina scoring, with a 27.5% success rate compared to Vina's 15.8% (both of which are significantly



Fig. 5 Virtual screening results on DUD-E for GNINA 1.3 compared with GNINA 1.0. Both the default scoring and the "fast" option are evaluated using (a) area under the ROC curve (AUC) and (b) normalized enrichment factor of the top 1%. Each data point corresponds to the performance of a specific, uniquely colored, DUD-E target



Fig. 6 GNINA covalent docking performance in terms of fraction of targets where the top ranked pose (darker shade) or any sampled pose (lighter shade) is within 2Å RMSD of the experimental structure. Error bars display the standard deviation across five docking runs initialized with different random seeds. Accuracy of other approaches is sourced from Scarpino et al. [26]

worse than enabling covalent docking). Overall, when using Vina scoring (-cnn_scoring=none), covalent docking with GNINA 1.3 is competitive with, but does not outperform, the state of the art.

Discussion

We present GNINA 1.3, an incremental improvement to the original GNINA software that lays the groundwork for more substantive future changes. GNINA now utilizes the PyTorch deep learning framework instead of Caffe, which allows quicker and easier integration with novel deep learning methods. Additionally, the switch to PyTorch reduces the computational cost of using the CNN scoring functions as shown in Fig. 3 and Table S7.

The built-in CNN scoring functions have been retrained on the most up-to-date version of the CrossDocked2020 dataset, which has increased the ranking performance on the cross-docking task. We find the retrained models show slightly reduced performance on redocking (Figures S5, S4), however, the CNN scoring functions still show superior ranking power to the Vina scoring function. The reduction in redocking performance is likely due to a reduction in the number of redocked poses in the CrossDocked2020 v1.3 dataset through filtering of problematic poses. Redocking is largely a synthetic benchmark for molecular docking as prospective drug discovery requires docking a ligand into a non-cognate receptor, so prioritizing improvements in cross-docking performance is a sensible strategy.

Finally, we utilized KD to reduce the computational burden of the highest performing CNN scoring functions without significantly reducing the pose ranking power of the models. Condensing CNN ensembles into a single model, in addition to the move to PyTorch, now enables an increase in Top1 cross-docking relative to Vina from about 25% to 36% with only a 1.5 s increase in average docking time without using a GPU. This will allow for much faster and cheaper screening of ultra-large libraries for drug discovery campaigns, like that in Li et al. [14] which docked 7 million compounds. Additionally, we now provide the option –cnn fast for high-throughput screening. This option is most appropriate for running many single-threaded docking jobs that will be followed by a rescreen of the top hits using the v1.3 Default ensemble to reduce the number of false positives. When ample compute or GPUs are available, the run-time performance improvement of this single fast model is likely not sufficient to justify a hierarchical screening strategy.

Due to the integration of PyTorch with GNINA we can now quickly develop new docking models and pipelines. In the future, we plan to add support for non-grid models such as Graph Neural Networks [8]. This development would allow direct comparison between CNN and GNN scoring functions with identical sampling strategies. We also plan to integrate newly developed deep neural network methods for sampling to replace or augment the Monte Carlo sampling currently provided in GNINA [7, 16]. These new sampling methods would provide an opportunity for improving binding site detection for whole protein docking, reducing the computational cost of sampling, and allowing for accurate docking to *apo* protein structures.

Availability and requirements

Project name: GNINA

Project home page: https://github.com/gnina/gnina Operating systems: Linux (Docker container available) Programming language: C++, CUDA Other requirements: CUDA, Open Babel 3 License: GPL2/Apache License Any restrictions to use by non-academics: None

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-025-00973-x.

Supplementary material 1.

Author contributions

All authors contributed to the development and evaluation of the software. All authors assisted in the preparation of this manuscript. All authors read and approved the final manuscript.

Funding

This work is supported by R35GM140753 from the National Institute of General Medical Sciences and is supported in part by the University of Pittsburgh Center for Research Computing through the resources provided. RM was supported by funding from the Biotechnology and Biological Sciences Research Council (BBSRC) National Productivity Investment Fund (NPIF) [BB/S50760X/1] and Evotec (UK) via the Interdisciplinary Biosciences DTP at the University of Oxford [BB/MO11224/1].

Page 7 of 8

Data availability

No datasets were generated or analysed during the current study.

Declarations

Competing interests

The authors declare no competing interests.

Received: 22 October 2024 Accepted: 16 February 2025 Published online: 02 March 2025

References

- Abagyan Ruben, Totrov Maxim, Kuznetsov Dmitry (1994) Icm?a new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation. J Comput Chem 15(5):488–506
- Allen-Zhu Z, Li Y (2020) Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. arXiv preprint[SPACE]arXiv: 2012.09816
- Bianco Giulia, Forli Stefano, Goodsell David S, Olson Arthur J (2016) Covalent docking using autodock: two-point attractor and flexible side chain methods. Protein Sci 25(1):295–301
- Brocidiacono M, Popov KI, Koes DR, Tropsha A (2023) Plantain: diffusioninspired pose score minimization for fast and accurate molecular docking. ArXiv
- Buttenschoen M, Morris GM, Deane CM (2024) Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. Chem Sci 15:3130
- Chen L, Cruz A, Ramsey S, Dickson CJ, Duca JS, Hornak V, Koes DR, Kurtzman T (2019) Hidden bias in the dud-e dataset leads to misleading performance of deep learning in structure-based virtual screening. PloS ONE 14(8):e0220113
- Corso G, Jing B, Barzilay R, Jaakkola T et al (2023) Diffdock: Diffusion steps, twists, and turns for molecular docking. In International Conference on Learning Representations (ICLR 2023)
- Corso G, Stark H, Jegelka S, Jaakkola T, Barzilay R (2024) Graph neural networks. Nat Rev Methods Primers 4(1):17
- Francoeur PG, Masuda T, Sunseri J, Jia A, Iovanisci RB, Snyder I, Koes DR (2020) Three-dimensional convolutional neural networks and a crossdocked data set for structure-based drug design. J Chem Inf Model 60(9):4200–4215
- Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv preprint. arXiv:1503.02531
- Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4700–4708
- Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. arXiv preprint. arXiv:1408.5093
- Koes DR, Baumgartner MP, Camacho CJ (2013) Lessons learned in empirical scoring with SMINA from the CSAR 2011 benchmarking exercise. J Chem Inf Model 53(8):1893–1904
- ...Li F, Ackloo S, Arrowsmith CH, Ban F, Barden CJ, Beck H, Beránek J, Berenger F, Bolotokova A, Bret G, Breznik M, Carosati E, Irene CY, Chen AC, Corte DD, Denzinger K, Dong A, Draga S, Dunn I, Edfeldt K, Edwards A, Eguida M, Eisenhuth P, Friedrich L, Fuerll A, Gardiner SS, Gentile F, Ghiabi P, Gibson E, Glavatskikh M, Gorgulla C, Guenther J, Gunnarsson A, Gusev F, Gutkin E, Halabelian L, Harding RJ, Hillisch A, Hoffer L, Hogner A, Houliston S, Irwin JJ, Isayev O, Ivanova A, Jarrett AJ, Jensen JH, Kireev D, Julian KS, Koby B, Koes D, Kumar A, Kurnikova MG, Kutlushina A, Lessel U, Liessmann F, Liu S, Wei L, Meiler J, Mettu A, Minibaeva G, Moretti R, Morris CJ, Narangoda C, Noonan T, Obendorf L, Pach S, Pandit A, Perveen S, Poda G, Polishchuk P, Puls K, Pütter V, Rognan D, Roskams-Edris D, Schindler C, Sindt F, Spiwok V, Steinmann C, Stevens RL, Talagayev V, Tingey D, Oanh V, Patrick WW, Wang X, Wang Z, Wolber G, Wolf CA, Wortmann L, Zeng H, Zepeda CA, Zhang KYJ, Zhang J, Zheng S, Schapira M (2024) Cache

challenge #1: targeting the wdr domain of Irrk2, a parkinson's disease associated protein. bioRxiv. https://doi.org/10.1101/2024.07.18.603797

- London N, Miller RM, Irwin JJ, Eidam O, Gibold L, Bonnet R, Shoichet BK, Taunton J (2014) Covalent docking of large libraries for the discovery of chemical probes. Biophys J 106(2):264a
- Lu W, Zhang J, Huang W, Zhang Z, Jia X, Wang Z, Shi L, Li C, Wolynes PG, Zheng S (2024) Dynamicbind: predicting ligand-specific protein-ligand complex structure with a deep equivariant generative model. Nat Commun 15(1):1071
- McNutt A, Li Y, Francoeur P, Koes D (2024) Condensing molecular docking cnns via knowledge distillation. ChemRxiv. https://doi.org/10.26434/ chemrxiv-2024-0jh8g
- McNutt AT, Francoeur P, Aggarwal R, Masuda T, Meli R, Ragoza M, Sunseri J, Koes DR (2021) Gnina 1.0: molecular docking with deep learning. J Cheminform 13(1):1–20
- McNutt AT, Bisiriyu F, Song S, Vyas A, Hutchison GR, Koes DR (2023) Conformer generation for structure-based drug design: How many and how good? J Chem Inf Model 63(21):6598–6607
- Mysinger MM, Carchia M, Irwin JJ, Shoichet BK (2012) Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. J Med Chem 55(14):6582–6594
- O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) Open babel: an open chemical toolbox. J Cheminform 3(1):33
- 22. Paggi JM, Pandit A, Dror RO (2024) The art and science of molecular docking. Ann Rev Biochem 93:389–410
- Paszke A, Gross S, Chintala S, Chanan G, Yang E, de Vito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch. NIPS 2017 Autodiff Workshop
- 24. Pearlman DA, Charifson PS (2001) Improved scoring of ligand- protein interactions using owfeg free energy grids. J Med Chem 44(4):502–511
- Ragoza M, Hochuli J, Idrobo E, Sunseri J, Koes DR (2017) Protein-ligand scoring with convolutional neural networks. J Chem Inf Model 57(4):942– 957. https://doi.org/10.1021/acs.jcim.6b00740
- 26. Scarpino A, Ferenczy GG, Keserű GM (2018) Comparative evaluation of covalent docking tools. J Chem Inf Model 58(7):1441–1458
- Schlander M, Hernandez-Villafuerte K, Cheng CY, Mestre-Ferrandiz J, Baumann M (2021) How much does it cost to research and develop a new drug? a systematic review and assessment. Pharmacoeconomics 39:1243–1269
- Sieg J, Flachsenberg F, Rarey M (2019) In need of bias control: evaluating chemical data for machine learning in structure-based virtual screening. J Chem Inf Model 59(3):947–961
- Su M, Yang Q, Du Y, Feng G, Liu Z, Li Y, Wang R (2018) Comparative assessment of scoring functions: the casf-2016 update. J Chem Inf Model 59(2):895–913
- Sunseri J, Koes DR (2020) Libmolgrid: graphics processing unit accelerated molecular gridding for deep learning applications. J Chem Inf Model 60(3):1079–1084
- 31. Sunseri J, Koes DR (2021) Virtual screening with gnina 1.0. Molecules 26(23):7369
- 32. Tian Y, Krishnan D, Isola P (2019) Contrastive representation distillation. arXiv preprint. arXiv:1910.10699
- Trott O, Olson AJ (2010) Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem 31(2):455–461
- Verdonk ML, Cole JC, Hartshorn MJ, Murray CW, Taylor RD (2003) Improved protein-ligand docking using gold. Proteins Struct Funct Bioinform 52(4):609–623
- 35. Vilar S, Cozza G, Moro S (2008) Medicinal chemistry and the molecular operating environment (MOE): application of QSAR and molecular docking to drug discovery. Curr Top Med Chem 8(18):1555–1572
- Wierbowski SD, Wingert BM, Zheng J, Camacho CJ (2020) Cross-docking benchmark for automated pose and ranking prediction of ligand binding. Protein Sci 29(1):298–305

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.