JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim

Application Note

DUBS: A Framework for Developing Directory of Useful Benchmarking Sets for Virtual Screening

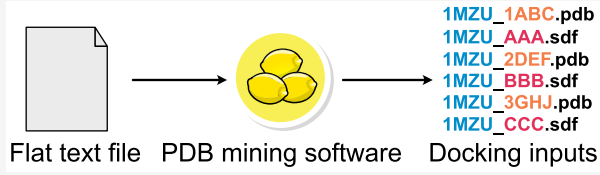
Jonathan Fine, Matthew Muhoberac, Guillaume Fraux, and Gaurav Chopra*

Cite This: https://dx.doi.org/10.1021/acs.jcim.0c00122

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Benchmarking is a crucial step in evaluating virtual screening methods for drug discovery. One major issue that arises among benchmarking data sets is a lack of a standardized format for representing the protein and ligand structures used to benchmark the virtual screening method. To address this, we introduce the Directory of Useful Benchmarking Sets (DUBS) framework, as a simple and flexible tool to rapidly create benchmarking sets using the protein databank. DUBS uses a simple input text based format along with the Lemon data mining framework to efficiently access and organize data to the protein databank and output commonly used inputs for virtual screening software. The simple input format used by DUBS allows users to define their own benchmarking data sets and access the corresponding information directly from the software package. Currently, it only takes DUBS less than 2 min to create a benchmark using this format. Since DUBS uses a simple python script, users can easily modify this to create more complex benchmarks. We hope that DUBS will be a useful community resource to provide a standardized representation for benchmarking data sets in virtual screening. The DUBS package is available on GitHub at https://github.com/chopralab/lemon/tree/master/dubs.



INTRODUCTION

Small molecule protein docking is one of many essential tools applied in virtual screening pipelines for drug discovery.[1−3] The docking protocol produces a 3D small molecule ligand pose inside the binding cavity of a protein, which is representative of the ligand conformation that is cocrystallized with the protein target. Energetics of binding site interactions of this pose are calculated to determine if the small molecule binds to the protein target, as well as, to estimate the binding affinity between the small molecule and the target. These calculations are used to virtually screen large libraries of synthetically feasible molecules to identify potential hits for specific targets.[4]

The virtual screening community has developed several benchmarking data sets to evaluate how well different docking methods perform at these vital tasks.[5−13] One of the popular benchmarking set is the Astex Diverse Benchmark[5] that is used to evaluate how well a given methodology can reproduce the crystal pose of a small molecule ligand in the bound (*holo*) form of a protein. Similarly, the PDBBind[14] and related CASF[11,15] data sets assess the ability of a docking method to produce and select a crystal-like pose for the small molecule ligand, and how well the methodology can rank the binding affinity of the small molecule. The most common metric used to identify a crystal-like pose is a root-mean-square deviation (RMSD) within 2.0 Å of the true crystal pose.[16,17] The Directory of Useful Decoys (DUD)[8] and Directory of Useful Decoys Enhanced (DUD-E)[9] data sets provide decoy ligands which may not bind to the associated target proteins to evaluate docking methods that can distinguish binders from nonbinders. Furthermore, the Pinc is Not Cognate (PINC)[12] benchmark is designed to evaluate how well a methodology can "cross dock" a given ligand. Cross docking refers to a process where a ligand is docked using the *holo* structure of a target protein crystallized with a different ligand. Similarly, the Holo-Apo Set 2 (HAP2)[13] benchmark is designed to measure the performance of methods to reproduce the crystal structure of a bound ligand using the *apo* (or unbound) form of the target protein. In addition to benchmarking sets derived from known protein structures, the docking community has produced "challenge sets" designed to rigorously validate a docking methodology in a blind manner. Two of these data sets are the Community Structure Activity Resource (CSAR)[18−21] and the Drug Design Data Resource (D3R)[22−24] using the data donated from the virtual screening community. Given the importance of these benchmarking sets as shown in the above works, it is clear that their continued development is important to improving and validating virtual screening pipelines.

The performance of any new virtual screening or docking methodology is tested on at least a few of the above benchmarks to gauge advancement in the field. Unfortunately,

Special Issue: New Trends in Virtual Screening

Received: February 3, 2020
Published: July 8, 2020

there are as many ways to format the input structure files for docking as there are benchmarking sets since the input file format specifications are not followed diligently. Here, we explore various issues with several commonly used benchmarking sets including PDBBind[14] that is the most standards compliant and then list the issue with DUD-E,[9] PINC, Astex,[5] and HAP2[13] (see **Issues with current benchmarking sets** in the Supporting Information for more details). PDBBind has no serious issues with file formats as it provides ligands in both MOL2 and SDF file formats along and the receptor in the PDB format. The authors consider both the biological assembly (created through symmetry operations) and the symmetry operations that are used to recreate the unit cell of the crystal. However, in some cases, PDBBind does not consider chains that belong to other biological assemblies, and this choice is likely made on a per protein basis by manual curation (for example, chain C in 4W9I interacts with the ligand in chain A, but chain C is not considered). Although this choice is confusing for the reproduction of this benchmark, it does not present any systematic issues. The PDBBind benchmark also does not consider residues other than water and metal ions in the binding pocket. Although many docking programs remove these atoms by default, the decision to do so should be made by the authors of a given virtual screening method. Next, the DUD[8] and DUD-E[9] benchmarks remove the element type for atoms specified for the target proteins (a required component of the PDB file format as per http://www.wwpdb.org/documentation/file-format.php). This is easily addressed because the atom names are given and can be used to reproduce these columns. However, the DUD-E[9] benchmark does not contain columns that are used to denote the chain name for a given residue. This issue cannot be addressed without reference to the original protein databank structure. Another issue with the DUD-E benchmark is its explicit definition of protonation states through the nonstandard relabeling of Histidine residues. Although some docking methodologies ignore these, the explicit labeling of the protonation states may bias a given docking method and it should the decision of the developer or user of the method to protonate a given residue. Some discrepancies also exist between 3D coordinates of the structure provided by the DUD-E benchmark compared to PDB (for example, the ligand for 1L2S).

Additionally, the PINC[12] benchmark only provides files in the MOL2 format where the protein atom names have been removed, causing difficulties when converting these files back to the PDB format. The receptor input files are formatted as MOL2 files instead of the traditional PDB files. This, by itself, is not an issue as it is possible to convert between a properly formatted MOL2 file and a PDB file. However, the provided MOL2 files lack the atom names required by the PDB format, thus making a direct conversion impossible. These atom names are used by other docking programs to assign atom types, identify rotatable bonds for flexible docking, or used to assign charges for their respective scoring function. Therefore, a properly formatted copy of this benchmark would be useful to the docking community, but this cannot be done without aligning the protein to the reference and ensuring that the 3D coordinates do not change from the reference version. Finally, no reference inputs are provided for the Astex and HAP2 benchmarks and need to be rederived for any new method comparison. Since different methods have been used for cleaning the experimental structures, there are many versions of the Astex benchmark when evaluated in publications; and they disagree about how well a given docking or virtual screening method performs on this benchmark (compare Vina results in FlexAID[13] and rDOCK[25]). Unfortunately, adequate details are not provided for preparing these files for docking study and therefore reproducing a given result can become challenging. To support these benchmarking sets, the developers must support odd corner cases and other issues related to formatting which takes additional development time better spent improving their scientific methodology. The differences in structure of the target, decoys, and ligands results in errors associated with high quality evaluation of screening methods.[26] In general, attempts to use a nonstandard version of these benchmarks may lead to differences in reported performance compared to prior publications. To alleviate these issues, we have created the Directory of Useful Benchmarking Sets (DUBS) which aims to provide a framework for curated and standardized version of past and future benchmarking sets.

For DUBS, we chose to base our benchmarking framework on the highly standardized and widely accepted Macro Molecular Transmission Format (MMTF)[27] for input as these files are already curated by the RCSB organization and provide a standard way of representing atomic coordinates, inter residue bonding, intra residue bonding, and other important crystallographic information. Using MMTF data structures, the entire protein data bank data takes less than 10 gigabytes of space,[28] allowing users to easily keep a copy of the PDB on their local hard drive. Furthermore, this is small enough to fit into random access memory on modern day workstations, enabling incredibly quick processing times as compared to other formats. Our recently published Lemon framework[29] will be used for rapid processing by creating simple Python scripts for generating suitable inputs for docking software evaluation. These files are written using the Chemfiles[30] input/output library which supports reading/writing a variety of formats in a highly standards compliant manner. We have chosen to use the PDB file format for protein input, as this format is standard for use in the docking community, and the SDF file format as a preference for small molecule input, as this format allows for the storage of formal charge instead of partial charge. We did not want to include the partial charges as such parameters can bias the performance of a virtual screening methodology[31−33] and, therefore, should be handled with care for each docking simulation. Similar to the input file types, DUBS can output ligands (and proteins) in a variety of formats using the Chemfiles[30] input/output library. These file formats include SDF, Tinker XYZ, PDB, CML, ARC, CSSR, GRO, MOL2, mmCIF, MMTF, and SMI. DUBS can be used to standardize existing benchmarking data sets as well as rapidly create user defined new benchmarks for virtual screening applications.

METHODS

Input Format. DUBS is designed to work using a specific input format that can incorporate variability in different types of information to develop user defined benchmarking sets. An overview of the DUBS pipeline is given in Figure 1 where an input file formatted in this specific manner is parsed using the provided python script that will be referred to as the parser, henceforth. Currently, the input format for DUBS is designed using five tags that is used by the parser to identify different types of information in the input file. This information is stored in nine dictionaries, which keep track of alignment of

https://dx.doi.org/10.1021/acs.jcim.0c00122
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

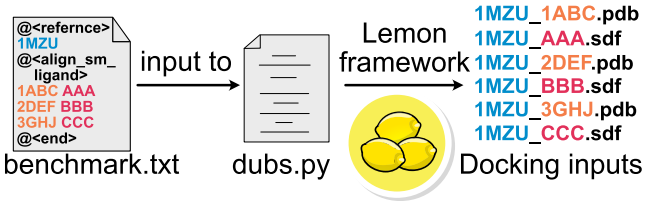


Figure 1. Overview of the DUBS software package where a flat text file is used to describe a given benchmarking set. A detailed description of this text file and the output of the dubs.py python script is given in the Methods section.

each benchmark protein to a reference protein (optional), and pairing of ligands to their respective protein. To better describe the input format and options, an example "block" of the input format used for DUBS is shown in Scheme 1 for hivproteasea2

Scheme 1. Input Format used for DUBS: Hivproteasea2 from the PINC Benchmarking Set

```
hivproteasea2
@<reference>
1MTB pinc_reference_files/hivproteasea2/1MTB.mmtf HPH
@<align_prot>
4HVP 2NC
2FGV NTB
1MTR PI6
2AID THK
@<align_sm_ligands>
3MXD K53
2FGU NTB
3EM3 478
3EM6 017
3OY4 017
3EKV 478
@<end>
```

from the PINC benchmarking set and the program options for each tag is provided in Scheme 2 for developing new benchmarking sets. The input files for several existing benchmarks, such as, PDBbind, Astex, PINC, HAP2, DUD-E, etc., are provided at https://github.com/chopralab/lemon/tree/master/dubs and as Supporting Information files. Additionally, there is a detailed DUBS user guide that explains the tag system and its usage in DUBS.

Scheme 2. Required and Optional Formatting Arguments for Both Input Files (a) Involving a Reference Protein and (b) Not Involving a Reference Protein[a]

a DUBS input format when a reference protein is used

```
Comments go here
@<reference>[1]
PDB_ID[3] reference_file_path[3] PBD_chemical_ID[4]
@<align_prot>[2]
PDB_ID[3] PBD_chemical_ID[3]
@<align_sm_ligands>[2]
PDB_ID[3] PDB_chemcial_ID[3]
@<end>[1]
Additional comments here
```

b DUBS input format when a reference protein is not used

```
Comments go here
@<no_align_sm_ligands>[2]
PDB_ID[3] PDB_chemical_ID[3]
@<end>[2]
Additional comments here
```

[a]Denoted are required tags (1), optional tags (2), required tag elements (3), and optional tag elements (4).

Algorithm Description. Some benchmarking sets such as PINC and HAP2 require the use of protein alignment as they are designed to evaluate docking performance on non-native protein conformations. To address this need, a customized version of the TM-align algorithm[34] is implemented in the Lemon framework[29] to allow for fast and accurate alignment between crystal structures of the same target protein. Briefly, this algorithm matches residues between the reference and nonreference crystal structures and attempts to find the affine transformation which minimizes the distance between the alpha carbons of the matched residues. In contrast to the original algorithm, the Lemon implementation incorporates the chain name of the residues in addition to their standardized residue ID. Additionally, this algorithm makes use of an optimized version of the Kabsch algorithm[35] to improve performance time. We also provide a method to access binding affinity information for specific protein−ligand pairs from Binding MOAD (Mother of All Databases).[36] Given a list of paired PDB IDs and three letter ligand codes as input, DUBS downloads the database information locally, parses the information, and accesses the binding affinity information for these pairs if it exists in the database. The binding affinity information includes the type (K_i, K_d, IC50), value, and units of the corresponding interaction.

Case Studies. Two case studies are presented in this work to show the similarity between benchmarking sets produced by DUBS versus originally published benchmarks. We obtained the original benchmarking sets from their respective Web sites (pdbbind.org.cn for PDBBind and dude.docking.org for DUD-E) and compared these structures to the versions produced by DUBS. To perform this comparison, we used the AutoDOCK Vina[37] scoring function to evaluate the scores of the native pose in the binding pocket for the two versions of the benchmarking sets. In addition, we calculated the difference in RMSD between the two sets using OpenBabel.[38]

Software Availability. The DUBS software is available for Python 3.6 and is compiled with gcc 6.3.0. It is freely available for installation via the Python Package Infrastructure (PyPI). All standardized formatted benchmarking sets are available for download directly from GitHub (https://github.com/chopralab/lemon/tree/master/dubs). The source code for the Lemon framework API and documentation are available on GitHub at https://chopralab.github.io/lemon/latest/.

RESULTS AND DISCUSSION

Commonly Used Benchmarking Sets. We showcase the ability of DUBS to reproduce and standardize previously published benchmarking sets by creating the input files described above for six well established benchmarking sets that derive from the protein databank (see Listing 2−7 in the Supporting Information). We measured the amount of time required to generate these benchmarking sets when the Hadoop copy of the protein databank was stored in RAM and using a single CPU core. We generated the benchmarking inputs 50 times for each benchmarked to calculate a mean and standard deviation for each calculation. We measured the total time taken by the application as well as the "user" and "system" time, which represent the time spent by the application itself and the time taken accessing hardware/allocating memory, respectively. These results are shown in Figure 2 and indicate

https://dx.doi.org/10.1021/acs.jcim.0c00122
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

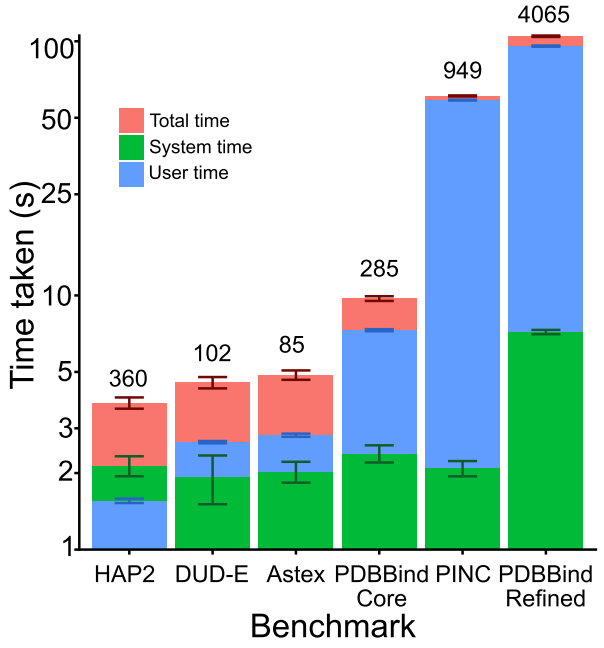


Figure 2. Timing results for the creation of five benchmark sets with the number of proteins indicated above each bar.

that DUBS is able to quickly produce a benchmarking set in less than 2 min. The most computationally expensive benchmark to generate is the PDBBind Refined set simply due to the number of ligands required to written as output files the software. The next most computationally expensive benchmark is PINC due to a large number of alignments that are required for the benchmark to be generated as there are 949 ligands as well as 60 proteins that needs to be aligned. It should be noted that the application took a greater proportion on its time in "user" mode for this benchmark than in "system" mode, compared to the PDBBind Refined set. These results show that DUBS can quickly recreate known benchmarking sets from the protein data bank. DUBS also has the ability to extract the binding affinity of a given ligand for a given entry but this feature is not discussed further.

To ensure that DUBS is recreating the benchmarking sets faithfully, we compared the AutoDOCK Vina score of the crystal poses of the PDBBind 2016 core set[11] using the protein structures provided by the original authors and the structures created by DUBS. We selected this scoring function because it relies on the positions of the heavy atoms in the ligand and protein and does not rely on the charges of the ligand and proteins assigned by the benchmarking set. As stated previously, it is not the goal of DUBS to provide atomic charges as this may bias a benchmarking set to perform better or worse for specific virtual screening methodology. The results of this comparison (Figure 3) shows that DUBS is able to recreate this benchmarking set faithfully. The structures produced by DUBS perfectly match the provided structures for 279 out of the 285 files provided by the original PDBBind

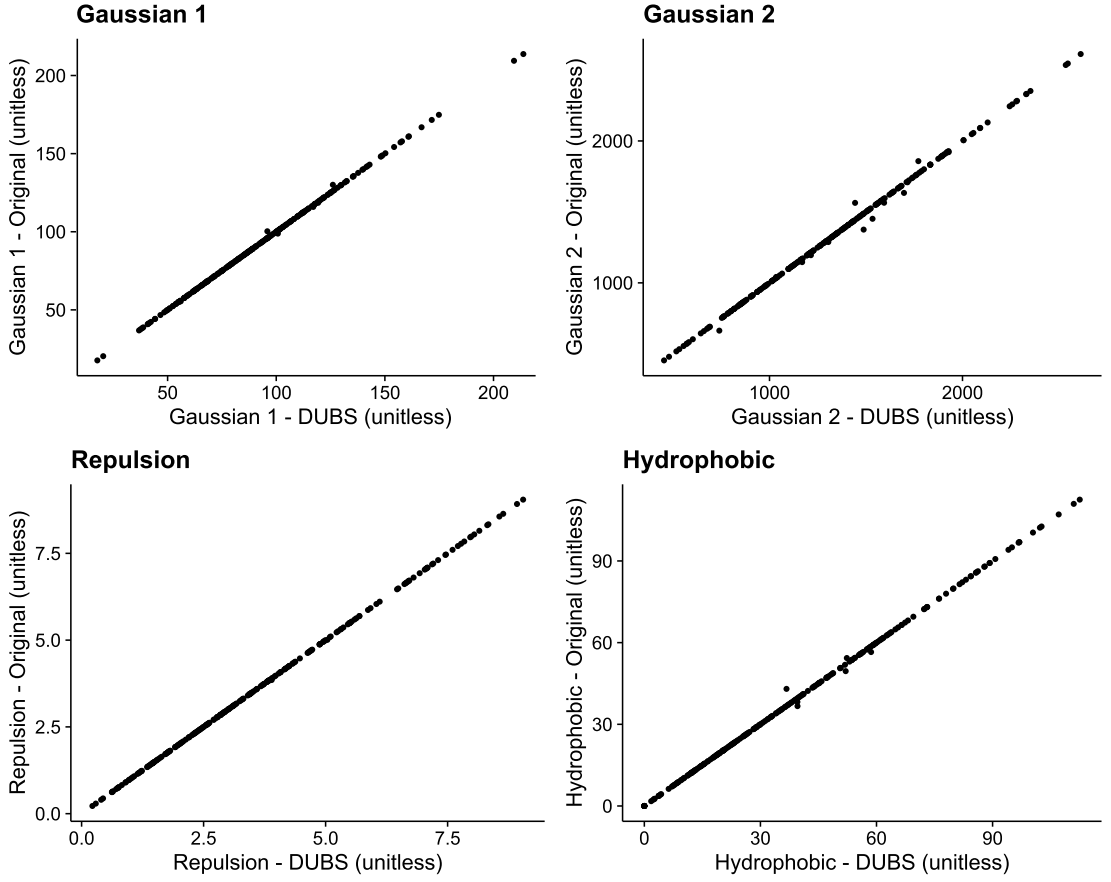


Figure 3. Comparison of the four components of the AutoDOCK Vina scoring function using structures provided by DUBS versus those provided by the original PDBBind 2016 Core set. The r^2 value for each comparison is greater than 0.99, indicating that DUBS is able to faithfully recreate this benchmarking set.

https://dx.doi.org/10.1021/acs.jcim.0c00122
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

authors. Specifically, the differences for two structures (4TMN and 5TMN) is related to how DUBS handles the symmetry of the crystal structures compared to PDBBind core set. By default, DUBS only considers the symmetry of the biological assembly in the unit cell, whereas PDBBind considers the symmetry of the crystal point group. Therefore, PDBBind contains an additional chain that is not included in the DUBS version. Since different benchmarking sets apply different cleaning, symmetry, and protonation operations on the experimental crystal structures, we think that none of these operations should be applied. This concept is discussed in detail for the DUD-E[9] benchmark in the next paragraph. Next, the remaining 4 (out of 6) differences between PDBBind and DUBS (4W9H, 4W9I, 4W9L, and 5C28) are due to the inclusion of an additional chain in the version provided by DUBS (chain L in the first three entries and chain B in the latter entry). DUBS considers all biological assemblies in the crystal structure, whereas PDBBind only considers one biological assembly. It should be noted that there is no difference between the native pose ligand RMSDs for the structures generated by DUBS versus those provided by the original PDBBind authors, therefore all differences in score are due to changes in how the protein is represented. However, an option in DUBS to remove other biological assemblies, even when they are in contact with the ligand, resolves these four differences.

A similar procedure was used to recreate the DUD-E benchmark using DUBS that revealed many issues with standardization of benchmarking sets. As an example, the DUD-E benchmark changes to the histidine residues to reflect their protonation states result in different residue codes for the affected residues. These residue codes are not recognized as histidine by some docking methods and are either ignored or not handled properly. Additionally, the DUD-E benchmark selectively includes residues such as Heme, ADP, NADP, etc., but removes other residues such as sugar molecules and polyatomic ions. These differences result in small deviations between the AutoDOCK Vina score in the original DUD-E benchmark compared to DUBS set in its normal configuration (see Figure S1 in the Supporting Information). It should be noted that DUBS can be modified to reproduce the DUD-E benchmark perfectly. However, these decisions will be different than those made for other benchmarks and will result in nonstandardization of benchmarking sets.

Novel Benchmarking Sets. In addition to the previously published benchmarks, we wanted to show how easy it is to create new benchmarking sets using the Lemon framework and DUBS. For this example benchmarking set, we created a simple script to identify all the small molecules in the PDB which interact with a Heme group (see original Lemon publication[29] for details). This script identified 1974 complexes that matches this criterion and created a DUBS input file without reference alignment. Of these complexes, 904 of them have unique small molecule ligands and we selected the complex with the lowest crystal structure resolution as representative of the small molecule/Heme protein complex. The resulting benchmarking set contains commonly studied complexes such as 1P2Y (nicotine in a CYPCAM) and 1R9O (flurbiprofen in CYP2C9) as these complexes are present in the Astex benchmark. In addition, it contains 6CIZ (a complex of abiraterone with CYP17A1) and 4NKX (progesterone with CYP17A1) which are not part of any benchmarking set. Using these structures, one could create a custom benchmarking set for measuring binding affinity toward various CYP enzymes. All steps in this methodology took less than 10 min, showing the power of the Lemon and DUBS pipeline. Depending on the user, several other complex but standardized benchmarking sets can be created using DUBS for future virtual screening applications.

CONCLUSION

In the work, we have presented a new software package, DUBS, for the creation of standards compliant benchmarking sets. DUBS uses a simple, flat text file as input to create docking input files for existing and new benchmarking sets for use in virtual screening within 5 min on modern hardware. Using this file format, we have reproduced six popular benchmarking sets and created an example of a new benchmarking set that can be used to evaluate how well docking methods can reproduce the binding mode of a compound in the presence of a Heme group. We believe that the DUBS framework will enable users to create new customizable and standardized benchmarking sets rapidly to foster a new era of useful virtual screening pipelines.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.0c00122.

Issues with current benchmarking sets, Figure S1, and Listings 1−7 (PDF)
DUBS user guide (PDF)
Various program input files (TXTs)

AUTHOR INFORMATION

Corresponding Author

Gaurav Chopra − *Department of Chemistry, Purdue University, West Lafayette, Indiana 47907, United States; Purdue Institute for Drug Discovery, Integrative Data Science Institute, Purdue Center for Cancer Research, Purdue Institute for Inflammation, Immunology and Infectious Disease, Purdue Institute for Integrative Neuroscience, West Lafayette, Indiana 47907, United States;* orcid.org/0000-0003-0942-7898; Email: gchopra@purdue.edu

Authors

Jonathan Fine − *Department of Chemistry, Purdue University, West Lafayette, Indiana 47907, United States*

Matthew Muhoberac − *Department of Chemistry, Purdue University, West Lafayette, Indiana 47907, United States*

Guillaume Fraux − *École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland;* orcid.org/0000-0003-4824-6512

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.0c00122

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported in part by a Purdue University start-up package from the Department of Chemistry at Purdue University, Ralph W. and Grace M. Showalter Research Trust award, the Integrative Data Science Initiative award, the Jim and Diann Robbers Cancer Research Grant for New

Investigators award and NIH NCATS ASPIRE Design Challenge awards to G.C., a Lynn Fellowship to J.F. and a Ross Fellowship to M.M. Additional support, in part by, a NCATS Clinical and Translational Sciences Award from the Indiana Clinical and Translational Sciences Institute (UL1TR002529) and the Purdue University Center for Cancer Research NIH grant P30 CA023168 are also acknowledged. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES

(1) Fine, J.; Konc, J.; Samudrala, R.; Chopra, G. CANDOCK: Chemical Atomic Network-Based Hierarchical Flexible Docking Algorithm Using Generalized Statistical Potentials. *J. Chem. Inf. Model.* **2020**, *60* (3), 1509−1527.

(2) Nedumpully-Govindan, P.; Jemec, D. B.; Ding, F. CSAR Benchmark of Flexible Medusa Dock in Affinity Prediction and Nativelike Binding Pose Selection. *J. Chem. Inf. Model.* **2016**, *56* (6), 1042−1052.

(3) Tanchuk, V. Y.; Tanin, V. O.; Vovk, A. I.; Poda, G. A New, Improved Hybrid Scoring Function for Molecular Docking and Scoring Based on AutoDock and AutoDock Vina. *Chem. Biol. Drug Des.* **2016**, *87*, 618.

(4) Lionta, E.; Spyrou, G.; Vassilatis, D.; Cournia, Z. Structure-Based Virtual Screening for Drug Discovery: Principles, Applications and Recent Advances. *Curr. Top. Med. Chem.* **2014**, *14* (16), 1923−1938.

(5) Hartshorn, M. J.; Verdonk, M. L.; Chessari, G.; Brewerton, S. C.; Mooij, W. T. M.; Mortenson, P. N.; Murray, C. W. Diverse, High-Quality Test Set for the Validation of Protein-Ligand Docking Performance. *J. Med. Chem.* **2007**, *50*, 726.

(6) Weiss, D. R.; Bortolato, A.; Tehan, B.; Mason, J. S. GPCR-Bench: A Benchmarking Set and Practitioners' Guide for G Protein-Coupled Receptor Docking. *J. Chem. Inf. Model.* **2016**, *56* (4), 642−651.

(7) Wallach, I.; Lilien, R. Virtual Decoy Sets for Molecular Docking Benchmarks. *J. Chem. Inf. Model.* **2011**, *51* (2), 196−202.

(8) Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking Sets for Molecular Docking. *J. Med. Chem.* **2006**, *49*, 6789.

(9) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55* (14), 6582−6594.

(10) Li, Y.; Liu, Z.; Li, J.; Han, L.; Liu, J.; Zhao, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. *J. Chem. Inf. Model.* **2014**, *54* (6), 1700−1716.

(11) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *J. Chem. Inf. Model.* **2019**, *59* (2), 895−913.

(12) Cleves, A. E.; Jain, A. N. Knowledge-Guided Docking: Accurate Prospective Prediction of Bound Configurations of Novel Ligands Using Surflex-Dock. *J. Comput.-Aided Mol. Des.* **2015**, *29* (6), 485−509.

(13) Gaudreault, F.; Najmanovich, R. J. FlexAID: Revisiting Docking on Non-Native-Complex Structures. *J. Chem. Inf. Model.* **2015**, *55* (7), 1323−1336.

(14) Liu, Z.; Su, M.; Han, L.; Liu, J.; Yang, Q.; Li, Y.; Wang, R. Forging the Basis for Developing Protein-Ligand Interaction Scoring Functions. *Acc. Chem. Res.* **2017**, *50* (2), 302−309.

(15) Li, Y.; Han, L.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 2. Evaluation Methods and General Results. *J. Chem. Inf. Model.* **2014**, *54* (6), 1717−1736.

(16) Ruppert, J.; Welch, W.; Jain, A. N. Automatic Identification and Representation of Protein Binding Sites for Molecular Docking. *Protein Sci.* **1997**, *6* (3), 524−533.

(17) Jain, A. N. Scoring Noncovalent Protein-Ligand Interactions: A Continuous Differentiable Function Tuned to Compute Binding Affinities. *J. Comput.-Aided Mol. Des.* **1996**, *10* (5), 427−440.

(18) Smith, R. D.; Dunbar, J. B.; Ung, P. M. U.; Esposito, E. X.; Yang, C. Y.; Wang, S.; Carlson, H. A. CSAR Benchmark Exercise of 2010: Combined Evaluation across All Submitted Scoring Functions. *J. Chem. Inf. Model.* **2011**, *51* (9), 2115−2131.

(19) Damm-Ganamet, K. L.; Smith, R. D.; Dunbar, J. B.; Stuckey, J. A.; Carlson, H. A. CSAR Benchmark Exercise 2011−2012: Evaluation of Results from Docking and Relative Ranking of Blinded Congeneric Series. *J. Chem. Inf. Model.* **2013**, *53* (8), 1853−1870.

(20) Carlson, H. A.; Smith, R. D.; Damm-Ganamet, K. L.; Stuckey, J. A.; Ahmed, A.; Convery, M. A.; Somers, D. O.; Kranz, M.; Elkins, P. A.; Cui, G.; Peishoff, C. E.; Lambert, M. H.; Dunbar, J. B. CSAR 2014: A Benchmark Exercise Using Unpublished Data from Pharma. *J. Chem. Inf. Model.* **2016**, *56* (6), 1063−1077.

(21) Smith, R. D.; Damm-Ganamet, K. L.; Dunbar, J. B.; Ahmed, A.; Chinnaswamy, K.; Delproposto, J. E.; Kubish, G. M.; Tinberg, C. E.; Khare, S. D.; Dou, J.; Doyle, L.; Stuckey, J. A.; Baker, D.; Carlson, H. A. CSAR Benchmark Exercise 2013: Evaluation of Results from a Combined Computational Protein Design, Docking, and Scoring/Ranking Challenge. *J. Chem. Inf. Model.* **2016**, *56* (6), 1022−1031.

(22) Gaieb, Z.; Liu, S.; Gathiaka, S.; Chiu, M.; Yang, H.; Shao, C.; Feher, V. A.; Walters, W. P.; Kuhn, B.; Rudolph, M. G.; Burley, S. K.; Gilson, M. K.; Amaro, R. E. D3R Grand Challenge 2: Blind Prediction of Protein−Ligand Poses, Affinity Rankings, and Relative Binding Free Energies. *J. Comput.-Aided Mol. Des.* **2018**, *32*, 1.

(23) Gathiaka, S.; Liu, S.; Chiu, M.; Yang, H.; Stuckey, J. A.; Kang, Y. N.; Delproposto, J.; Kubish, G.; Dunbar, J. B.; Carlson, H. A.; Burley, S. K.; Walters, W. P.; Amaro, R. E.; Feher, V. A.; Gilson, M. K. D3R Grand Challenge 2015: Evaluation of Protein−Ligand Pose and Affinity Predictions. *J. Comput.-Aided Mol. Des.* **2016**, *30* (9), 651−668.

(24) Gaieb, Z.; Parks, C. D.; Chiu, M.; Yang, H.; Shao, C.; Walters, W. P.; Lambert, M. H.; Nevins, N.; Bembenek, S. D.; Ameriks, M. K.; Mirzadegan, T.; Burley, S. K.; Amaro, R. E.; Gilson, M. K. D3R Grand Challenge 3: Blind Prediction of Protein−Ligand Poses and Affinity Rankings. *J. Comput.-Aided Mol. Des.* **2019**, *33* (1), 1.

(25) 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* (4), e1003571.

(26) Lagarde, N.; Zagury, J.-F.; Montes, M. Benchmarking Data Sets for the Evaluation of Virtual Ligand Screening Methods: Review and Perspectives. *J. Chem. Inf. Model.* **2015**, *55* (7), 1297−1307.

(27) Bradley, A. R.; Rose, A. S.; Pavelka, A.; Valasatava, Y.; Duarte, J. M.; Prlić, A.; Rose, P. W. MMTF—An Efficient File Format for the Transmission, Visualization, and Analysis of Macromolecular Structures. *PLoS Comput. Biol.* **2017**, *13* (6), No. e1005575.

(28) Rose, P. W.; Prlić, A.; Bi, C.; Bluhm, W. F.; Christie, C. H.; Dutta, S.; Green, R. K.; Goodsell, D. S.; Westbrook, J. D.; Woo, J.; Young, J.; Zardecki, C.; Berman, H. M.; Bourne, P. E.; Burley, S. K. The RCSB Protein Data Bank: Views of Structural Biology for Basic and Applied Research and Education. *Nucleic Acids Res.* **2015**, *43* (D1), D345−D356.

(29) Fine, J.; Chopra, G. Lemon: A Framework for Rapidly Mining Structural Information from the Protein Data Bank. *Bioinformatics* **2019**, *35*, 4165.

(30) Fraux, G. Chemfiles/Chemfiles: 0.9.2. *Zenodo*, 2020.

(31) Cho, A. E.; Guallar, V.; Berne, B. J.; Friesner, R. Importance of Accurate Charges in Molecular Docking: Quantum Mechanical/Molecular Mechanical (QM/MM) Approach. *J. Comput. Chem.* **2005**, *26* (9), 915−931.

(32) Tsai, K. C.; Wang, S. H.; Hsiao, N. W.; Li, M.; Wang, B. The Effect of Different Electrostatic Potentials on Docking Accuracy: A Case Study Using DOCK5.4. *Bioorg. Med. Chem. Lett.* **2008**, *18* (12), 3509−3512.

https://dx.doi.org/10.1021/acs.jcim.0c00122
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

(33) Bikadi, Z.; Hazai, E. Application of the PM6 Semi-Empirical Method to Modeling Proteins Enhances Docking Accuracy of AutoDock. *J. Cheminf.* **2009**, *1* (1), 15.
(34) Zhang, Y. I-TASSER Server for Protein 3D Structure Prediction. *BMC Bioinf.* **2008**, *9*, 40.
(35) Kabsch, W. A Solution for the Best Rotation to Relate Two Sets of Vectors. *Acta Crystallogr., Sect. A: Cryst. Phys., Diffr., Theor. Gen. Crystallogr.* **1976**, *32* (5), 922−923.
(36) Smith, R. D.; Clark, J. J.; Ahmed, A.; Orban, Z. J.; Dunbar, J. B.; Carlson, H. A. Updates to Binding MOAD (Mother of All Databases): Polypharmacology Tools and Their Utility in Drug Repurposing. *J. Mol. Biol.* **2019**, *431* (13), 2423−2433.
(37) 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**, *31* (2), 455−461.
(38) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminf.* **2011**, *3* (1), 33.

https://dx.doi.org/10.1021/acs.jcim.0c00122
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX