COMPARISON OF TRADITIONAL AND MACHINE LEARNING PROGRAMS IN THE EVALUATION OF PROTEIN-LIGAND BINDING

Alexander Zatuchney, Blessing Anyangwe, Arushi Desai, Elizabeth Fishman, Kevin Jin, Kai Kim, Erin Kraus, Eugene Lee, Angelina Li, Bridget Liu, Nicholas Sardy, Aarna Tekriwal, Osariemen Unuigbe, Eric Zhu, and David Cincotta

Drew University, Madison, New Jersey, United States of America

ABSTRACT

Molecular docking, an in-silico method with widespread pharmacological applications, is used to predict the optimal conformation of a protein-ligand complex. Traditionally, it uses search-score algorithms that generate protein-ligand poses and calculate each pose's binding strength. More recently, artificial intelligence (AI) programs have been developed and trained with protein-ligand datasets. To compare the accuracy of these approaches in site-specific docking, a traditional program and a deep-learning (DL) program were tasked with docking a set of protein-ligand pairs. Upon comparison of the two programs' results, it was determined that they predict optimal binding conformations with similar accuracy.

KEYWORDS

Molecular docking, Protein-ligand binding, Deep-learning, DiffDock-Pocket, SeeSAR

1. INTRODUCTION

The objective of this project was to assess and compare the accuracy of traditional and deep-learning molecular docking technologies to determine the value of machine learning (ML) programs within the realm of molecular docking. It will utilize a curated dataset of protein-ligand ground truths to evaluate the unpublished site-specific ML molecular docking program Diffdock-Pocket compared to the commercial traditional molecular docking program SeeSAR v13.1.1.

2. LITERATURE REVIEW

2.1. Protein-Ligand Binding

Protein-ligand complexes were once thought to be static structures with perfect geometric complementarity, similar to a lock and key [1]. This, however, was later determined to be energetically implausible, especially for enzyme-substrate binding. Instead, these complexes continue to tighten after binding—a model called induced fit [2]. All protein-ligand interactions abide by principles of induced fit [1].

2.2. Definition and Significance of Molecular Docking

Molecular docking is an in-silico method in bioinformatics that computationally assesses the interaction of two or more molecules based on stability and strength [3]. Docking programs are used to predict the optimal binding conformation of a small protein-ligand complex given a ligand and its respective target protein.

Now commonly used in drug development, molecular docking has become a crucial tool in the medical field for curing diseases and treating symptoms. In particular, it can determine whether a drug of interest can bind to specific binding sites of a targeted protein in a conformation that allows it to fulfill its physiological purpose.

Additionally, molecular docking has several applications in the field of polypharmacology, where it is used to identify ligands that simultaneously bind to a pool of selected targets of interest. Docking has also been used to determine novel uses for chemical compounds with already optimized safety profiles, an approach called drug repositioning [4]. The widespread pharmacological applications of molecular docking have directed significant attention towards it.

2.3. Molecular Docking: Traditional Methods

Traditionally, molecular docking techniques rely on a search-score algorithm, which generates various protein-ligand binding conformations (poses) and assesses their binding strengths. A target structure must first be selected. A ligand or separate molecule is then selected for docking; its torsion is accounted for in the docking calculation.

In the actual docking process, a variety of search algorithms can be used to identify possible binding conformations. In fragmentation, the ligand is separated into fragments, individually bound to the macromolecule, and then repositioned and combined to find the most favorable position. Programs may also use smaller bound structures stored in their database to estimate the binding position of similar, larger molecules. Other processes, such as approaches based on the Monte Carlo algorithm, randomly insert ligands to determine the optimal position. A genetic algorithm-based approach utilizes a pose and then adds mutations, or random changes, which it then analyzes and ranks in terms of compatibility. Tabu searches analyze the failures of previously identified binding attempts to apply further restrictions in new binding simulations. Programs like “LUDI” and “DOCK” use a fragmentation-based method while “MCDOCK” uses a Monte Carlo approach and “AutoDock” and “GOLD” use a genetic algorithm [5].

2.4. Molecular Docking: Deep Learning Methods

More recently, machine learning-based molecular docking has been developed due to potential optimizations in time and ease of access. These deep-learning-based (DL) methods—which utilize reverse diffusion models—do not rely on conventional search-score algorithms and have improved processing speeds.

ML algorithms use training data to “learn” how to predict binding patterns; entirely based on included input data, they are “knowledge-based” rather than physics-based [6]. Since DL-based approaches rely entirely on training datasets, their docking performance on any new protein-ligand complex depends on its structural/compositional similarity to those used to train the program [7].

DL programs use scoring functions to evaluate and rank protein-ligand poses based on electrostatic, polar, nonpolar, hydrophilic, and hydrophobic interactions within the protein-ligand

complex, as well as how the complex interacts with its environment or other interacting "residues" [7].

2.5. Comparing Traditional Methods and Deep Learning Methods

Traditional methods and deep-learning methods for molecular docking possess characteristic advantages and disadvantages. For instance, traditional methods tend to specialize in molecular docking processes in which the target protein's binding pocket has already been provided alongside the ligand and the target protein (called "site-specific docking") [8]. However, many traditional docking methods experience complications in terms of accuracy and sampling space, especially when running particularly large protein-ligand complexes [7]. Furthermore, while traditional methods continuously rotate ligands and alter their torsion angles in the docking process, they generally fail to account for protein flexibility, undermining accuracy [4], [8].

DL methods, on the other hand, generally perform "blind-docking," a process that docks a ligand to an entire protein, which involves both pocket searching and site-specific docking. They tend to be very competent in locating a given protein's pocket (called "pocket searching") [8]. Moreover, the generative diffusion models used in deep-learning approaches are able to better emulate the stochasticity of biological processes [9]. However, since the performance of a deep-learning molecular docking program is dependent on the provided training dataset, they tend to be less consistent than traditional methods at generalization. In contrast, traditional docking programs are relatively consistent, especially when dealing with smaller protein-ligand complexes.

With the advent of new DL software, discussions have arisen comparing the accuracy of these programs to that of traditional programs. To compare the efficacy of traditional and deep-learning molecular docking technologies, selected programs of each type can be used to perform molecular docking on ligand-protein complexes with known optimal binding conformations. It is important to consider proteins and ligands distinct enough from those used to train the docking program when testing their docking ability. This minimizes the possibility of producing results that greatly overestimate the performance of DL programs.

An applicable traditional program for this comparison is BioSolveIT's SeeSAR 'Midas' (v. 13.1.1), a commercial program that performs site-specific docking. An applicable DL program for this comparison is DiffDock-Pocket—a currently unpublished site-specific docking program developed to be an extension of the standard DiffDock program [9]. Because DiffDock-Pocket, unlike most other DL programs, has a site-specific approach, it can be compared to traditional methods more effectively. DiffDock-Pocket uses random ligand poses and side chain conformations along with a reverse diffusion process to predict realistic binding conformations [9]. In general, it relies on known information about a ligand binding site to identify locations where a ligand most significantly impacts target protein structure, accounting for protein flexibility in these regions while keeping more static regions relatively fixed [9]. After inputting a ligand of interest within its target protein's binding pocket, DiffDock-Pocket repeatedly updates protein joint structures such that protein-ligand binding conformations become progressively more realistic [9]. Once docking has concluded, the program utilizes a confidence model to score and rank the generated protein-ligand conformations, allowing the most plausible conformations to be determined [9].

Due to the consistent, physical basis of the traditional docking programs, it is hypothesized that the traditional molecular docking program (SeeSAR) will be more accurate than its DL-based counterpart (DiffDock-Pocket).

3. METHODS

3.1. Dataset Formation

An ideal dataset to test molecular docking programs contains proteins that do not share similar characteristics to proteins in their dataset. Many existing datasets, however, do not meet this requirement, as non-homologous structures may still have similar binding pockets due to pockets' highly conserved nature. Corso et al. [10] sought to solve this issue when evaluating their DiffDock-L program. Many machine-learning protein dockers, like DiffDock and its derivatives, utilize the PDBBind dataset for their training data [11]. To test their model, Corso et al. utilized the Binding Dataset from the Mother of All Databases (MOAD), which has different selection criteria to form its dataset compared to PDBBind, to make a 189 protein dataset called Docking Generalization, more commonly referred to as "DockGen" [10]. Corso et al. found that existing ML programs had a large drop in accuracy with this dataset, showing that it is sufficiently different from their training datasets to use in generalization testing [10].

DockGen was used as a base and then reduced by removing protein-ligand pairs without structure quality assessments present in the RCSB Protein Data Bank (PDB) for a resulting dataset of 135 protein-ligand pairs. This was done to ensure the accuracy of the ground truth ligand positions in the protein.

3.2. Traditional Docking

Molecular docking was conducted with the proteins and ligands in the dataset using both SeeSAR v.13.1.1 and DiffDock-Pocket.

SeeSAR was used to predict the conformation of a ligand in its target protein. To do so, a protein of interest from the dataset was first loaded into SeeSAR. SeeSAR always loads the asymmetric unit of a protein, which means that the same biological assembly was used for the ground truth and DiffDock-Pocket for accurate comparison. After loading a protein, a list of ligands is provided. The first ligand with an identical name to that of the ligand of interest was extracted—while maintaining its binding pocket—and then added to molecule editing mode. Afterward, the ligand was deleted to allow the pocket to be unoccupied, and a SMILES code of the same ligand was added to the protein through Analyzer mode. Finally, standard docking was performed on the protein and the newly added ligand. Upon completion, a list of calculated poses and their estimated binding affinities is displayed. The pose with the greatest binding affinity was saved as a .sdf file. These files were then compared to the ground truth files found on PDB to calculate their Root Mean Square Deviation (RMSD), a quantity that describes the similarity between "two superimposed atomic coordinates" [10], [12].

3.3. Machine-Learning Docking

DiffDock-Pocket was also used to predict protein-ligand binding. The asymmetric structures of each protein were downloaded as .pdb files from PDB to match the biological assembly used in SeeSAR. The proteins were then "fixed" to remove all bound ligands, the aqueous solvent, and all other residues, eliminating any interferences to protein analysis. Then, the ligand of interest in the dataset was extracted from the PDB files using Python scripts to obtain the ground truth ligand and the subsequent protein with the ligand removed. DiffDock-Pocket ran the unoccupied proteins and exported .sdf files of the ligand that was used. The top-ranked ligand file was compared to the ground-truth ligand to produce an RMSD value.

3.4. Calculation of Data

Comparatively evaluating DiffDock-Pocket and SeeSAR required a shared metric that could quantify the accuracy of protein-ligand conformations generated by each program relative to ground truth ligand-protein complexes. This shared metric was Root Mean Square Deviation (RMSD), which "is the most commonly used quantitative measure of the similarity between two superimposed atomic coordinates" [12]. RMSD (see Figure 1) is calculated by measuring the distance between equivalent atoms in two superimposed structures. n represents the number of atom pairs and d_i is the distance between the two atoms. RMSD is 0 for identical structures and increases as the structures become more diverse. The RMSD values for each of the 135 protein-ligand pairs were calculated for both SeeSAR and DiffDock-Pocket using a Python-based program (See Appendix A for the link to the programs used). To calculate these values, the data outputted by both programs was compared to the ground truth ligand position/orientation. The similarity of the data (the ligand poses) produced by each program and the ground truth ligand poses produced the RMSD values. Values below 2Å were considered to be reliable while values below 1.5 Å were considered very reliable.

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}$$

Figure 1: Equation of Root Mean Square Deviation (RMSD), the metric used to evaluate generated protein-ligand complexes of both SeeSAR and DiffDock with ground truth protein-ligand complexes [12].

3.5. Protocol

The process of running the proteins on the two programs and calculating the RMSD values included various steps. First, .pdb files of a protein were downloaded from the PDB. A Python script was used to extract the corresponding ground-truth ligands from the .pdb files and convert them to .mol2 format. For DiffDock-Pocket, another script was used to remove hydrogens and add missing atoms. Both programs were run using the appropriate files. The "fixed" .pdb file and the ground-truth ligand in .mol2 format were inputted into DiffDock-Pocket. For SeeSAR, the protein data was collected through the program's built-in database and the proper ligand was inputted to the program in .pdb format.

The above procedure was followed for each protein in the dataset. Subsequently, the outputted .sdf files were gathered alongside the ground-truth ligands in .mol2 format, and Python code was written and used for calculating RMSD values for each program.

4. RESULTS

36 of the proteins analyzed did not output an RMSD value due to errors in at least one program. To better compare the two programs' ability to predict the optimal binding position, only the protein-ligand complexes able to be generated by both programs and evaluated by the Python program were analyzed. To visually analyze and compare the RMSD values calculated for SeeSAR and DiffDock-Pocket, two histograms (Figures 2a and 2b) were created through the SPSS program. The X-axis contains the range of RMSD values grouped into 7 bins. Each bin has a range of 1 angstrom, starting from 0 to 6 angstroms. Bin 7 contains the number of protein-ligand complexes that have an RMSD value above 6 angstroms. Table 1 represents the

percentage of proteins that ran on SeeSAR and Diffdock-Pocket that were successful and unsuccessful at docking.

Table 1: Percentage of proteins that ran on both programs that were successful and unsuccessful at docking.

	SeeSAR	DiffDock-Pocket
%Successful (≤2Å)	18.2%	15.2%
%Unsuccessful (>2Å)	81.8%	84.8%

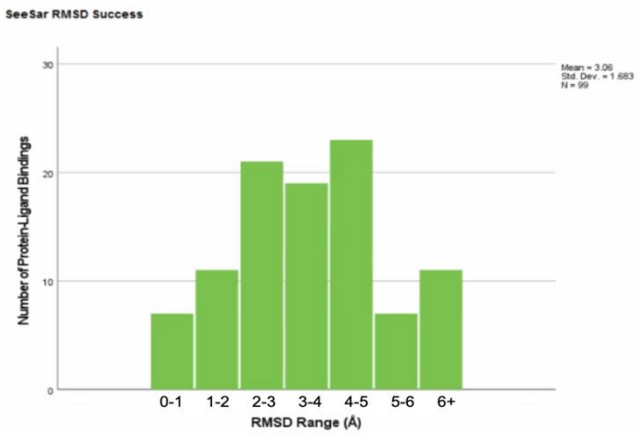

Figure 2a: Histogram displaying the RMSD values of the proteins that successfully ran through SeeSAR and the calculations.

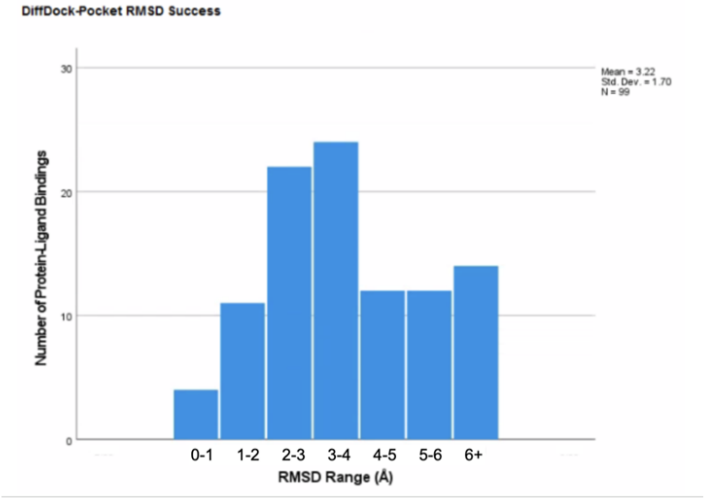

Figure 2b: Histogram displaying the RMSD values of the proteins that successfully ran through DiffDock-Pocket and the calculations.

The results indicate that DiffDock-Pocket can output poses with similar accuracy to that of computational programs such as SeeSAR. As shown in Table 1, both programs had a similar number of successful docking procedures. When looking specifically at the values that

successfully ran, SeeSAR had a success rate of 18.2% (18 proteins), while DiffDock-Pocket had a success rate of 15.2% (15 proteins). Although DiffDock-Pocket had a 3% lower success rate, the difference between the success rates is not large enough to conclude that SeeSAR was significantly better at predicting successful dockings. Additionally, as per Fig. 2a and Fig. 2b, the mean and standard deviation of the distribution of RMSD values vary minimally between DiffDock-Pocket and SeeSAR. More specifically, the difference in mean RMSD value is 0.16 and the difference in standard deviation is 0.017. Thus, the programs' performances are not different enough to discern which program is more accurate.

However, one of the major discrepancies in the data collected from the two programs is the number of protein-ligand combinations that could not be run: SeeSAR had a larger number of protein-ligand pairs for which an RMSD value could not be calculated than DiffDock-Pocket. Specifically, the number of proteins that were not able to be calculated from DiffDock-Pocket was 18 out of the 135 protein-ligand pair dataset (13.3% of the total), while SeeSAR was unable to run 31 proteins (23.0% of the total). Therefore, SeeSAR—while able to create a relatively high number of successful dockings—generates a lower number of appropriate binding structures for analysis than DiffDock-Pocket.

5. CONCLUSION

The purpose of this project was to evaluate the accuracy of machine learning-based Molecular docking programs in comparison to traditional programs. DiffDock-Pocket and SeeSAR had similar success rates and median error values. Though there is some slight variation in the magnitude of error, it is not significant enough to conclude that either program is more accurate. There is a possibility that the RMSD calculator used produced some error in its output, which may impact the results. However, whether or not DiffDock-Pocket had a significantly different success rate than SeeSAR, AI and ML programs are growing exponentially and there is a strong possibility that they will be able to be used for quick and accurate molecular docking in the future. As more training data is introduced into ML programs, the accuracy and precision of their predictions will improve.

User discretion should be used to determine which software is superior in a given situation. For a more automated experience, DiffDock-Pocket is preferable, as once the files are added in and run, the poses are automatically calculated, and it is not necessary for the user to sort through the various pockets. SeeSAR, on the other hand, has a manual interface that does not allow for easy automation. It is important to note that these programs both had similar running times. However, the hardware used was not powerful enough to determine whether this would still be true in commercial settings.

6. DISCUSSION

SeeSAR was initially expected to produce smaller RMSD values; however, in this experiment, SeeSAR was less reliable than anticipated—especially in comparison to the newer, deep-learning-based DiffDock-Pocket. However, SeeSAR was also more prone to human error, as obtaining RMSD values required several steps. DiffDock-Pocket, on the other hand, was automated.

DiffDock-Pocket performed similarly on the DockGen dataset when compared to other AI docking models. Corso et al. demonstrated that other models like GNINA, SMINA, and Equibind had a success rate of 10.6%, 17.5%, and 0.0% respectively [8]. Furthermore, the original DiffDock had a success rate of 6.0% and DiffDock-L had a one of 22.6%. DiffDock-

Pocket, thus, places around average in terms of success rate. However, DiffDock-Pocket had the lowest median error out of all the programs, likely due to its pocket-specific approach, showing that the program has promise for docking prediction.

6.1. Limitations

RMSD was the only metric used in this study to compare DiffDock-Pocket with SeeSAR since access to experimental data that could be collected from both programs was limited. Although it is a commonly used metric, solely relying on it limits the accuracy of comparison. Optimally, more than one metric should be compared to better evaluate molecular docking accuracy to provide a more multidimensional analysis of the differences in competency between the two docking approaches.

Additionally, solely using RMSD has limited applications when considering binding affinity. Though the accuracy of the position of the ligand is important, if the ligand is not likely to bind in the pocket, the data may not be relevant. SeeSAR can calculate binding affinity for generated poses, but DiffDock-Pocket does not. Notably, other ML programs like GNINA can predict binding affinity from a confidence score, but GNINA struggles with the DockGen dataset [9].

While the RMSD calculations for the DiffDock-Pocket outputs were relatively accurate, the SeeSAR RMSD values exhibited significant errors, likely due to programming errors. For instance, the ligand (TRP) in protein 3ZZS had a calculated RMSD of 2.76. In reality, this ligand has an RMSD of .533, as found on zhanggroup.org [13]. The error is likely due to a misalignment of atom order between the ground truth and SeeSAR files. However, since zhanggroup.org cannot be automated, the remaining ligands could not be efficiently compared using this resource.

Furthermore, analysis of protein-ligand interactions on SeeSAR was limited to completion on only Windows, whereas analysis on DiffDock-Pocket was limited to completion on only Mac (DiffDock-Pocket was not accessible on Windows). Because the operating system used was not standardized, it may be possible a given docking software would have produced different values if a different operating system had been used. Finally, using a more powerful computer product with high RAM storage capacity could produce more accurate results on both DiffDock-Pocket and SeeSAR.

Only one pocket per protein was considered for evaluation, even though many proteins in the dataset had more than one available pocket for a molecule to bind. Although both programs can evaluate multiple ligand binding pockets, to balance computational efficiency with the precision of results, only the top binding pocket for each protein was evaluated.

Moreover, repeatedly docking the same protein-ligand complex would occasionally yield different RMSD values during the study. Thus, the reliability of the methods used is likely questionable. Because SeeSAR is a commercial product, the probable causes of this variability are DiffDock-Pocket (due to its novelty) and the overall RMSD calculation script.

6.2. Future Research

Improving the quality and diversity of training data is a key aspect for further research, as large, well-curated datasets are essential for training AI models. Reducing bias and enhancing different models' abilities to generalize in creating results for more expansive data would lead to a significant boost in performance and reliability.

Another area of potential improvement is to include a more sophisticated statistical analysis like a T-test when comparing the RMSD results for both programs. 135 proteins were used in this study, which is not a statistically significant number from the full PDB database to perform a T-test. For future studies, a large enough dataset should be used to apply statistical analyses to help ensure that the results of evaluations are sound. Notably, as DockGen only contains 189 protein-ligand combinations, researchers should also seek to expand this dataset while maintaining its generalization-testing characteristics.

Furthermore, using a variety of RMSD calculators may reduce potential variation due to programming errors. Ensuring that the RMSDs are calculated correctly is imperative for future studies that use this metric to compare the accuracy of traditional and artificial intelligence docking programs.

Finally, future researchers could focus on hybrid approaches that combine both AI and traditional methods to increase the accuracy of molecular docking. Traditional approaches may be effective at providing the frameworks for molecular interactions, but they are limited in the ability to capture the full complexity of molecular behavior in particularly complex systems or compounds. AI-based methods can synthesize information and patterns from a vast array of data, but they are also limited in their specific data fields in terms of reliability. Combining the best of both approaches would enhance the overall capabilities of docking tools, potentially increasing their accuracy and efficiency.

AUTHOR CONTRIBUTIONS

All authors have contributed equally. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGEMENTS

Thank you to Mr. Hannes Stärk and Mr. Michael Plainer for teaching us about DiffDock and DiffDock-Pocket; to Drew University for providing the facilities used during the Governor's School of New Jersey program in the Sciences; to Dr. Cincotta for his guidance during our project; to Ian Weitz, Visvam Rajesh, Kaleb Yong, and Yeji Kim for assistance with data collection; and to Katerina Pouathas and Joel Moses for coordination and assistance. Finally, we would like to thank the Overdeck Family Foundation, Novartis Pharmaceuticals, and the New Jersey Office of the Secretary of Higher Education for the funding that has offered GSNJS to us and thousands more.

REFERENCES

[1] Nelson, David & Cox, Michael, (2017) “Lehninger Principles of Biochemistry, 8th edition”, Macmillan Higher Education, W.H. Freeman & Company.

[2] Urry, Lisa; Cain, Michael; Wasserman, Steven; Minorsky, Peter; & Orr, Rebecca, (2021) “Campbell Biology, 12th Edition”, Campbell Biology, Pearson.

[3] Sharma, Aditi; Kunwar, Saloni; Agarwal, Vaishali; Singh, Chhaya; Sharma, Manish; & Chauhan, Neha, (2021) “Molecular Docking: An Explanatory Approach Structure-Based Drug Designing and Discovery”, *International Journal of Pharmacy and Pharmaceutical Sciences*, Vol 13, No. 6, pp6-12, https://www.researchgate.net/publication/352094390_MOLECULAR_DOCKING_AN_EXPLANATORY_APPROACH_IN_STRUCTURE-BASED_DRUG_DESIGNING_AND_DISCOVERY.

[4] Pinzi, Luca & Rastelli, Giulio, (2019) "Molecular Docking: Shifting Paradigms in Drug Discovery", *International Journal of Molecular Sciences*, Vol. 20, No. 18, https://doi.org/10.3390/ijms20184331.

[5] Xu, Xianjin; Huang, Marshal; & Zou, Xiaoqin, (2018) "Docking-based inverse virtual creening: methods, applications, and challenges"*BioPhys Rep.*, Vol. 4, No. 1, pp1–16, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5860130/.

[6] Balius, Trent & Rigby, Megan, (2022) "The use of Molecular Docking as a ligand discovery tool; Can machine learning help the pursuit for ligands?", *NCI Ras Dialogue Blog*, https://www.cancer.gov/research/key-initiatives/ras/news-events/dialogue-blog/2022/dock.

[7] Qing, Rui; Hao, Shilei; Smorodina, Eva; Jin, Daiv; Zalevsky, Arthur; & Zhang, Shuguag (2022), "Protein Design: From the Aspect of Water Solubility and Stability", *ChemRev*, Vol. 122, No. 18, pp14085–14179, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9523718/.

[8] Yu, Yuejiang; Lu, Shuqi; Gao, Zhifeng; Zheng, Hang; Ke, Guolin, (2023) "Do Deep Learning Models Really Outperform Traditional Approaches in Molecular Docking?", https://arxiv.org/pdf/2302.07134 (unpublished).

[9] Plainer, Michael; Toth, Marcella; Dobers, Simon; Stark, Hannes; Corso, Gabriele; Marquet, Celine; & Barzilay, Regina, "DIFFDOCK-POCKET: Diffusion for Pocket-Level Docking with Side Chain Flexibility", https://plainer.dev/assets/pdf/2023/DiffDock-Pocket.pdf (unpublished).

[10] Corso, Gabriele; Deng, Arthur; Fry, Benjamin; Polizzi Nicholas; Barzilay, Regina; & Jaakkola, Tommy, (2024) "Deep confident steps to new pockets: Strategies for docking generalization", *ArXiv*, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10925391/.

[11] Liu, Zhihai; Su, Minyi; Han, Li; Liu, Jie; Yang, Qifan; Li, Yan; Wang, Renxiao, (2017) "Forging the basis for developing protein–ligand interaction scoring functions", *Accounts of Chemical Research*, Vol. 50, No. 2, https://pubs.acs.org/doi/full/10.1021/acs.accounts.6b00491

[12] Abagyan, Ruben & Kufareva, Irina, (2012) "Methods of Protein Structure Composition", *Homology Modeling*, Vol 857, pp231-257, https://doi.org/10.1007%2F978-1-61779-588-6_10.

[13] Zhang, Yang & Bell, Eric, (2019) "DockRMSD: Docking Pose Distance Calculation", *Journal of Cheminformatics*, Vol. 11, No. 40, https://doi.org/10.1186/s13321-019-0362-7.

APPENDIX A

Github Link for Python RMSD Calculators

https://github.com/aarna-tekriwal/DD-Pocket-SeeSAR/tree/main