International Journal of Environment, Agriculture and Biotechnology
Vol-9, Issue-5; Sep-Oct, 2024
Peer-Reviewed International Journal
Journal Home Page Available: https://ijeab.com/
Journal DOI: 10.22161/ijeab

Recent Advances in the development of User-friendly Software Tools for Computational Protein Design, Modeling, and Molecular Docking

Magezi Joshua[1,2*], Semawule Syrus[2], Namuswe Magdalene[2], Janice Adaeze Nwankwo[3]

[1]The Key Laboratory of Industrial Biotechnology, Ministry of Education, Laboratory of Applied Microbiology and Metabolic Engineering, School of Biotechnology, Jiangnan University, China
[2]Makerere University, Uganda
[3]State Key Laboratory of Food science and Resources, National Engineering Research Center for Functional Foods, National Engineering Research Research Center of Cereal Fermentation and Food Biomanufacturing, Collaborative Innovation Center of Food Safety and Quality Control in Jiangsu Province, School of Food Science and Technology, Jiangnan University, Wuxi 214122, China
* Corresponding Author
magezijoshua183@gmail.com , magezijoshua183@outlook.com , joshua.magezi@students.mak.ac.ug

Received: 15 Sep 2024; Received in revised form: 11 Oct 2024; Accepted: 16 Oct 2024; Available online: 23 Oct 2024

Abstract— *Computational protein design, modeling, and Molecular Docking represent a group of vital in-silico methods employed in predicting protein sequences with desired functions, predicting protein structures, and several molecular interactions with proteins. The application of such in silico methods is seen in the screening of potential targets during new drug designs, the discovery of novel protein sequences that could play new and vital functions such as industrial processes, understanding of protein function by studying its residues, and understanding of the effects of position mutations to the structure and function of the protein. This means these tools find a lot more use in protein-related research. However, many scientists are basic computer users and not experts in utilizing sophisticated software involved in in-silico Protein methods. This limits the progress of mainly young researchers towards protein-related research ideas. This review therefore discusses the progress in the development of user-friendly tools for assisted Insilco protein design, modeling, and docking.*

Keywords— ***Computational protein design, molecular docking, in-silico methods, protein structure prediction, and user-friendly research tools.***

I. INTRODUCTION

Computational protein design (CPD) refers to Insilco methods employed in the design of sequence and molecular structure of proteins aimed towards a desired function in a wide range of fields such as protein design for harsh industrial processes, cellular signaling manipulation through the introduction of novel proteins in metabolic pathways, and improved protein stability and activity. (Enzymes)

On the other hand, Molecular docking refers to the application of computational tools whose algorithms predict binding modes and interactions proteins form with ligands, nucleic acids, and other protein molecules. Molecular docking performed with several ligands can help in the first screening of candidates that would bind effectively. Such information could be used to discover new drug compounds, which bind and inhibit protein activities involved in disease. Previously, scientists required years to perform protein manipulations for industrial and pharmaceutical uses; however, the present generation has seen rapid development and application of high-throughput computational technologies that have increased Biotechnological research. Computational protein design (CPD) and molecular docking are among these new game-changing technologies

that have provided time-saving alternative Biotechnological solutions.

A variety of computational tools utilized in computational protein design exist for mainly two roles; prediction and scoring of protein folding models, and denovo construction of proteins. (Obtaining new sequences of amino acids that are expected to fold in a desired pattern for a particular function) Protein modeling remains one of the crucial tools in CPD as it enables scientists to simulate various sequences and analyse the resulting structures while aiming toward a particular function and properties.

There are three types of modeling; Homology modeling, Threading, and Ab initio modeling. Homology modeling such as the SWISS model, depends on sequence alignments through blasting to fold sequences of interest. This highly depends on protein sequences stored in Protein Data banks and a 30% or more similarity could show chances of similar protein folding. Threading involves the use of known protein folds for proteins similar to the sequence of interest. Examples of such software include I-Tasser. Ab initio involves modeling a protein from only its amino acid sequence by calculating the most favorable energy conformations of the amino acids. However, it should be noted that in Ab initio, Protein Database information can still be used, for example in Rosetta, a common software that uses a stochastic approach based on Monte Carlo sampling to optimize a particular sequence. Different protein fragments stored in the Protein database are utilized during this modeling and these guide the folding of the protein sequence of interest taking into consideration that each fragment has a non-zero chance of being lined with the sequence of interest.

Denovo construction of proteins uses the energy function in the generation of the protein tertiary structure, designing of the amino acid identity, and side chain conformation at all residue positions. (Ben et al., 2021)

Molecular docking is also a form of modeling that involves the interaction of two or more molecules to give a stable adduct (complex). Molecular docking aims to predict the favored orientation of a molecular complex. (Protein-Protein, Protein-Ligand, or Protein-Nucleic acid) Several molecular docking software exist and these generally operate a scoring function that selects and ranks ligand conformation, orientation, and translation. Available docking programs have a common drawback where the protein surface is kept ridged which prevents flexibility as seen in induced fit mechanisms. (Serge & Igor, 2014)

Difficulties in molecular docking are largely due to the high number of degrees of freedom characterizing a complex system which increases time and cost of computational calculations. To overcome this challenge, several algorithms based on approximations are utilized such as rigid docking that considers only the three translational and three rotational degrees of freedom of protein and the docking molecules such as ligands, treating them as two separate rigid bodies. The most widely used algorithms at present allow docking molecules to fully explore conformational degrees of freedom in a rigid-body receptor.

Therefore, two different approaches exist in Computational protein design and molecular docking; Deterministic and Heuristic algorithms. Deterministic approaches are reproducible, in which a continuous and consistent list of results is obtained. This means that, according to their energy function, one will find the best possible solution for a design problem. The only drawback is energy functions are not so accurate, require experimental validation and computational analysis can be time-consuming. (Chen et al., 2009, Frey et al., 2010)

Heuristic algorithms use stochastic processes to search the space described by the model and offer the advantage of being fast and cheaper compared to deterministic approaches. Because of this, many protein design programs use *heuristic* optimization algorithms to compute low-energy sequences in the search space described by the biophysical model. (Samish, 2009) However, this means Heuristic approaches do not maximally search the best solution models towards the global minimum and hence give varying results per computational run. (Gainza et al., 2016, Bonet et al., 2019) To solve this, A large number of candidate decoys are generated by Heuristic CPD workflows such as the Rosetta modeling suite which increases the sample space of generated models. (Alford et al., 2017)

This however results in the challenge of the need for user-friendly analysis tools which has seen many developers resorting to analysis scripts provided by third parties to correctly study such a large amount of data and increase biomolecular research. (Jaume et al., 2019) This review will therefore summarize the most used user-friendly tools for performing computational design protein, molecular docking, and analysis of CPD data.

User-friendly tools for analysis of Rosetta CPD data

Several computational software exists for the prediction of protein models and range from locally installed computer programs to web savers operated on supercomputers. Among these tools, the Rosetta modeling suite is one of the most commonly used software having both local and web-based operations. (Leman et al., 2019) Initially, it was primarily designed for protein structure prediction and folding, however to date, the Rosetta software includes Rosetta dock, docking small ligands to proteins, antibody

and immune system protein design, modeling membrane proteins, and Denovo design of proteins.

Despite its usefulness in scientific research, users especially scientists with low coding knowledge experience challenges with the Rosetta molecular modeling suite owing to the software being a command line-only application with approximately 1.7 million lines of c ++ code. This together with a large population of molecular decoys continued to limit research and therefore has led to a steady development of Graphical User Interfaces (GUI) that are user-friendly such as rstoolbox, pyrosetta, Rosettascripts, and Foldit suiting different use cases and workflow styles

PyRosetta Toolkit

PyRosetta is a Python-based Rosetta GUI for performing molecular modeling, Protein design, and analysis of Rosetta calculation results. (Adolf & Dunbrack, 2013) It is composed of the main window which allows the user to quickly specify protein regions and output options, perform quick analyses, and run standard protocols. (relaxing structures and regions) Additional functions can be found in the main menu. Visualization is aided by the PyMOL tool.

Main menu of PyRosetta shows a typical GUI. (The File Menu allows the user to load a structure from a PDB file or directly from the Protein Data Bank, prepare a PDB for use in Rosetta, save and load GUI sessions, and import or export a variety of Rosetta filetypes. The Options Menu allows the user to set the number of processors to use, setup the main score function, and interact with the Rosetta options system. The Visualization Menu allows the user to integrate modeling tasks directly with PyMOL using Rosetta's PyMOLMover. the Advanced Menu houses a variety of sub-windows and useful functions for analyzing Rosetta results. Four Rosetta-specific analyzers are implemented, including the Void Identification and Packing Analyzer (VIP), Packstat, InterfaceAnalyzer, and LoopAnalyzer). (Lewis & Kuhlman, 2011)

Python-based rstoolbox

Jaume et al (2019) developed a Python-based Rosetta silent (rs) toolbox to facilitate protein model analysis even with beginner-level coding skills. There are four functional modules within the Python library; rstoolbox.io (provides read and write functions for multiple data types), rstoolbox. Analysis (provides analysis functions responsible for scoring designed decoys), rstoolbox. Plot (Provides functions that aid in graphical generations to represent analysis results; such as Ramachandran plots), and rstoolbox. Utils. (Provides helper functions for data manipulation, design comparisons, and amino acid profile creation. In addition to the four functional modules, rstoolbox also contains decoy population data, position-specific scoring matrix (PSSM), and Rosetta fragments data which all together form a components module. Rosetta fragments data contains position and properties of fragments which is very important in Ab initio as the latter depends on fragments used during modeling. PSSM is a matrix containing information about the probability of a given amino acid occurring in each position as depicted from multiple sequence alignments. The design frame contains information on the decoy population such as energetic scores, sequence, and secondary structure.

This shows an advantage of the rstool box in increasing functional options that enable users to select desired decoys, with desired sequences and properties, together with choosing fragments that best represent a minimum energy.

With the plot functions, rstoolbox provides graphical representations which reflect on the properties of protein structural designs and fragments used in the case of Ab initio. Modeled protein designs have different dihedral angles present in polypeptide backbones and such angles directly relate to the stability of the protein. Analysis of such stability is aided by a Ramachandran plot which can show dihedral angle distributions of each amino acid present in a modeled structure. These tend to cluster in four different regions which can depict secondary protein structure. However, the clustering of amino acids disallowed regions shows the instability of a protein backbone owing to steric hindrances.

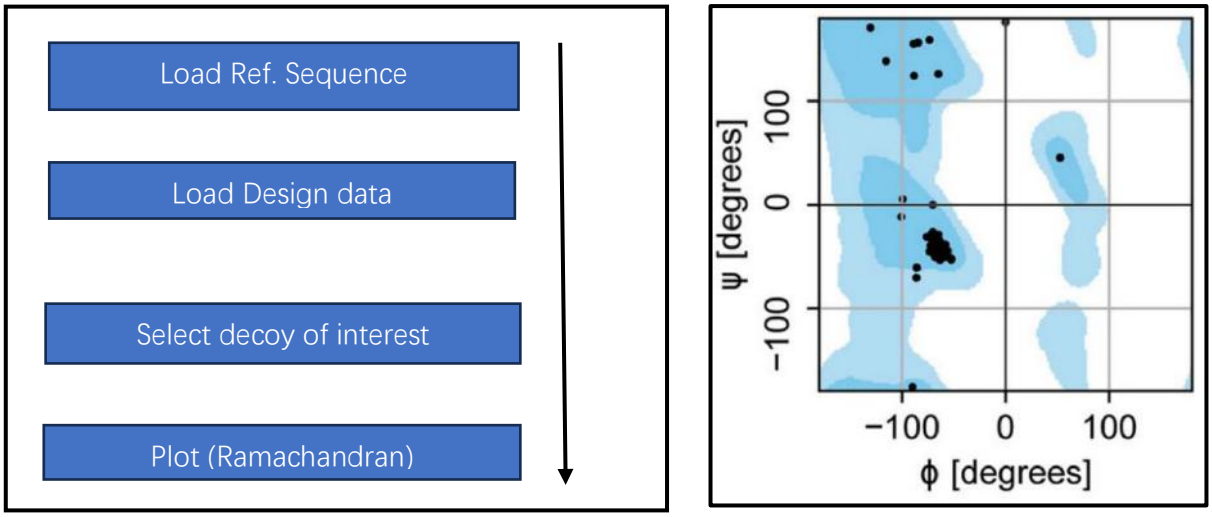

Fig. 1 shows a summary of procedures executed by commands while analyzing structural stability using rstoolbox. However, this can only be useful with already locally installed Rosetta software. With this, and with imported rstoolbox, loading of rosetta predicted design with input functions, .io results in the generation of data containers defined in the components module from which decoys of interest are selected for stability analysis via the Ramachandran plot.

The Ramachandran plot in Fig 1 represents a general form, in which the clustering of amino acids is limited to particular regions as shown. However, there are other forms of Ramachandran plots namely glycine, Proline, and Pre-proline that can also be generated by the toolbox.

The toolbox also offers easy-to-use analysis and plotting functions of novel protein structures predicted by Ab initio, by comparing local structure similarity of designed protein (target) with PDB fragments used during modeling as shown in Fig. 2

Interestingly, the rstoolbox facilitates interactive computational design approaches, meaning several rounds of design are performed and each generation of designs is used to guide the next one. The toolbox offers a variety of functions that facilitate this, for example; the selection of decoys with a particular mutation of interest or the generation of designs with a minimum number of mutations

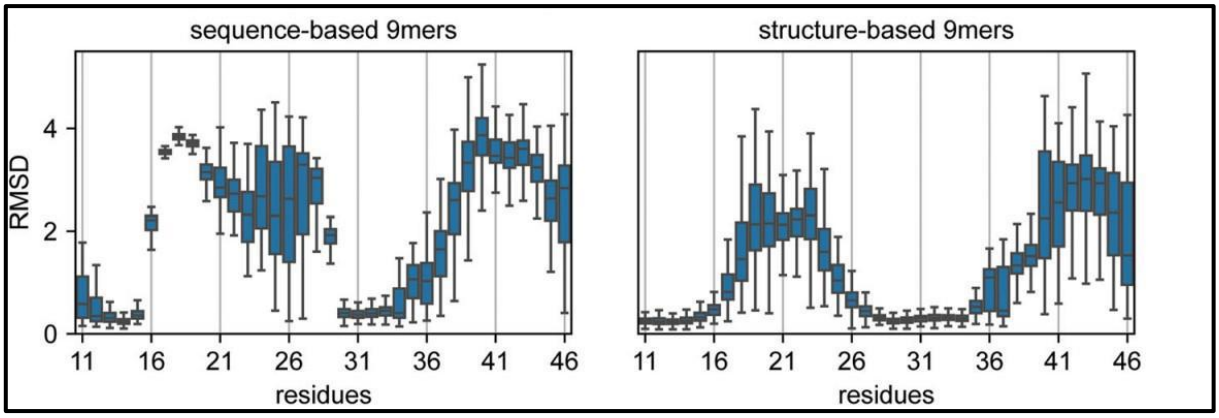

Fig. 2 demonstrates both sequence and structure-based fragments, and from the graphs, the root mean square deviation (RMSD) of structure-based 9mers is generally lower than that of sequence-based 9mers. (Jaume et al.,2019) This shows that there are fewer differences in amino acid residues at a particular position of the structure-based 9mers (fragment) and the target.

AMDock (Assisted molecular docking with AutoDock4 and AutoDockVina)-Python-based

A large number of docking programs have been developed during the last three decades, based on different search algorithms and scoring functions. Aiming to make these docking programs more user-friendly, especially to beginners, different graphical user interfaces (GUIs) have been developed to assist in the preparation of molecular systems, the execution of the calculations, and/or the analysis of the results. (Valdés et al., 2020)

AutoDock vina and AutoDock 4 are both AutoDock programs, a widely used free access docking software, and work similarly by pairing an empirically-weighted scoring function with a global optimization algorithm. However, the two differ in such a way that AutoDock Vina performs calculations of a gradient while seeking a local optimum. AutoDock 4 on the other hand uses stochastic approaches to generate random conformations for testing. (Chang et al., 2010)

AMDock includes both Auto Dock Vina and AutoDock 4 which labels it as a multi-platform tool. The tool further includes ADT scripts (Responsible for preparation of ligand and receptor files), Auto Ligand, (Harris et al., 2007) Open Babel, (Boyle et al., 2011) PDB2PQR, (Dolinsky et al., 2004) PyMOL, (Schrödinger, 2002) and AutoDock4Zn for proteins whose active sites contain Zinc ions.

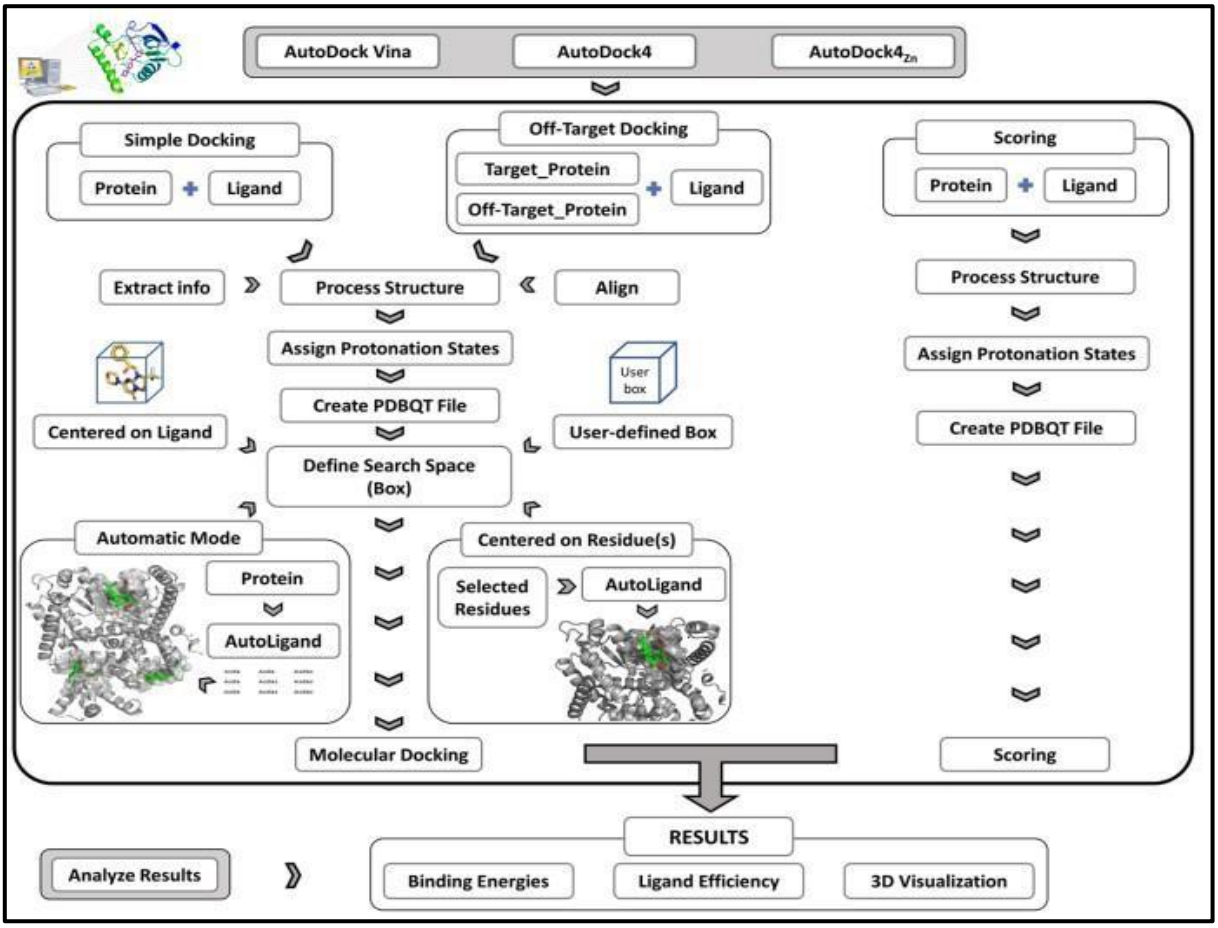

Fig 3: (Valdés et al., 2020) illustration of the workflow utilized by AMDock.

From **Fig 3**, AMDock consists of five tabs; Home, Docking options, Analysis, configuration, and Info. It is therefore much easier for the user to select the appropriate docking approach from the Home tab. (AutoDock Vina, AutoDock 4 engines and AutoDock 4_{Zn}.) Selection of any of these leads to an automatic shift to the Options tab in which simple docking (Protein and ligand), off-target docking (ligand with a protein having two ligand-receptor sites), or scoring can be performed which further increases the AMDock applications. After this, ADT scripts prepare the input files for docking. Another advantage of AMDock is that the user can define the pH under which docking is predicted. Molecular docking using Vina is typically conducted using the default box size and this is calculated based on the coordinates of the native ligand interacting with a protein of interest in the experimental structure. (Feinstein & Brylinski, 2015)

AMDock offers four different approaches to defining search box center and size; Automatic, Center, Hetero, and Box. In the automatic approach, the program uses Auto Ligand to predict possible binding sites and then generates a box with optimal dimensions cantered on each Auto Ligand object at each predicted binding site (Morris et al., 2009)

Center on Ligand involves the use of AutoLigand to generate an object with a volume similar to the ligand size, using it as a reference to the geometric center of the selected

residues. Then, a box with optimal dimensions is centered on the formed object.

Center on Hetero involves a box placed on the geometric center of an existing ligand (if the receptor was given in complex with a ligand). The box approach involves the box center and dimensions are defined by the user. Running docking involves just a click (run) and generated results are analyzed using the analyze results tab where Affinity, Estimated Ki values, and Ligand Efficiencies are listed for the different resulting complexes. Visualization of these results is performed using PyMOL software.

Shortcomings of AMDocking

Despite providing a novel, simple, and user-friendly workflow, some challenges have been identified in the software. It does not perform hydrated docking (A concept in which water molecules might not be completely displaced by ligand molecules, but some remain and contribute to the overall complex) and neither covalent docking which means that the docking approach by AMDock for protein ligands cannot be used to screen probable drug or active substances owing to system inability to take into account of chemical reactions leading to covalent bond formation. (Scarpino et al., 2020)

Flexible side chains and ligands are not implemented during structural manipulation which roughly means it is dependent on a local and key approach. This limits the number of possible docking complexes as a result of decreased degrees of freedom hence reducing computational predicting power. (Sheng, 2018)

Virtual Screening, a concept utilized so much in drug discovery is not supported by AMDock. In this approach, computational tools are used to analyze small molecule (ligand candidates) libraries to come up with a lead molecule having the highest score of binding. Owing to this approach not being incorporated, it limits the application of AMDock in results analysis for drug-related research. (Lavecchia & Di Giovanni, 2013)

PyMOL Visualisation & analysis software

PyMOL is another example of GUI that aids visualization of 3D data resulting from proteins, nucleic acids, small molecules, electron densities, surfaces, and trajectories. (Yuan et al., 2017) The role of PyMOL is this review will be limited to Visualization and analysis of protein models, docking data, molecular simulations, and role in drug design. It is an open and widely used Python-based software to create high-quality movies and images such as ribbons, cartoons, dots, spheres, surfaces, or lines while providing a wide range of other functions such as estimation of the distance between neighboring atoms, and differential representation of molecules. To date, PyMOL accommodates a wide range of Plugins with further increased applicability.

Protein structure analysis with PyMOL

To date, PyMOL accommodates a wide range of Plugins and so with increased applicability. For Protein structure analysis;

DSSP and Stride: These assign secondary structures to protein models and provide graphical user interfaces for coloring according to predicted secondary structures.

Mole: This assists in rapid and automatic location and characterization of channels, funnels, and pores in molecular structures.

PyANM: Allows users to build and visualize Anisotropic network models (Entropic models demonstrated as captures of Internal energy)

Protein-ligand modeling/Docking

This is also Protein-Ligand docking, and it involves several steps all supported by PyMOL. These include; protein homology modeling, protein preparation, ligand alignment, ligand preparation and finally docking. The most useful application of protein modeling is seen in new drug discoveries and involves the steps below, (Yuan, 2017)

Homology Modeling: Using this approach, a 3D model of the target protein sequence is generated using a homologous template. Plugins to assist in homology modeling within PyMOL include PyRosetta and PyMod which combines a majority of other homology modeling tools such as Modeller, PSI-BLAST, Clustal Omega, Muscle, and PSIPRED. The purpose of Homology modeling is to predict a 3D protein design that is required in docking.

Protein preparation: Protonation states of mainly Asparagine and Glutamine are so important because the amide residues at the end of the molecules tend to form up to four Hydrogen bonds that contribute so highly to Protein structure and stability. (Weichenberger et al., 2007)

With the Amber plugin, one can enhance predictions of the modeled protein structure. This comes with the addition of pKa information and optimization of conformations/protonation state of Asparagine, Histidine, and glutamine, refining the H-bond network and energy minimization.

Ligand Alignment: Proteins that bind a common ligand may have similarities and a conserved active site structure. PyMOL plugin LigAlign offers a reliable approach in which PDB files of proteins complex with the common ligand are obtained from a databank. It then aligns the protein complexes according to the minimum root mean square deviation (RMSD) and so enabling the user to identify

conserved structural patterns. This enables the user to identify and define ligand search space.

Ligand preparation: Any molecule imported into PyMOL can have its energy minimized by the Optimize plugin. (3D geometries, proper bond orders, accessible tautomer, and proper ionization states of molecules are altered towards minimum energy)

Protein-Ligand Docking: PyMOL can include Docking pie, a plugin that that provides an easy-to-use interface to four docking programs together with scoring functions; Smina, Autodock Vina, RxDock (Proteins and nucleic acids) and AutoDock for Flexible Receptors (AutoDockFR). (Ravindranath et al., 2015)

Virtual screening with PyMOL

Two types of virtual screening exist; Ligand and Structure-based virtual screening. Ligand-based virtual screening employs a concept of Scaffold hopping, in which different basic scaffolds can have similar or even better biochemical activities compared to an already-known compound. The PyMOL plugin Lisica utilizes this concept and aids scientists in using structural activity data from a set of known bioactive molecules to identify probable candidate compounds which reduces the amount of experimental validation. (PyMOL here is used as a compound screening tool) (Dilip et al., 2016)

On the other hand, Structural-based virtual screening involves the use of 3D data of protein targets obtained using Protein modeling or from NMR and X-ray methods to dock active compounds into the binding sites and then rank them using scoring functions. It is evident therefore that structure-based virtual screening is dependent on the 3D structure of the target protein and similar protein-ligand complexes present in a data bank. (Li & Shah, 2017)

Molecular dynamics (MD) simulations using PyMOL

Molecular dynamics simulation is an approach to obtain kinetic and thermodynamic characteristics of biomolecular structures. MD simulation software is therefore very useful in the establishment of macromolecular stability, identification of allosteric sites, elucidation of mechanisms of enzymatic activity, molecular recognition and properties of complexes with small molecules, association between proteins, protein folding, and its hydration.

Despite of application of MD simulations, most tools present are used in command line form and continue to challenge researchers with little computational skills. (Vieira et al., 2023)

MDBuilder is a PyMOL plugin that offers an easy-to-use GUI that assists researchers in building the starting structures for multiple popular MD simulation packages. Also, the Dynamics PyMOL plugin assists in MD simulation by using GROMACS tools to provide an easy-to-use GUI for molecules loaded directly to PyMOL.

Hydration site prediction

Water molecules that are bound to the active site of a protein can aid in substrate/ligand binding and so prediction of the properties and location of the hydration site is crucial in understanding protein function. PyMOL plugin WATsite provides analysis for hydration sites while providing an easy-to-use GUI. (Yang et al., 2017)

Many more plugins can be utilized in PyMOL software. With such a number, PyMOL is enhanced with several molecular functions that label it as a modern platform for computational drug design.

Protein-Ligand docking with Instadock

Compared to other docking platforms, Instadock provides one of the most user-friendly GUIs for docking and virtual screening. (Mohammad et al., 2021)

Several Docking programs have been released as GUIs for example; Raccoon, (Cosconati et al., 2010) PaDEL-ADV, PyRx, (allakyan $ Olson, 2015) AUDocker LE, (Sandeep et al., 2011) VSDocker, (Prakhov et al., 2010) DockingApp, (Di Muzio et al., 2017) MOLA, (Abreu et al., 2010) and DockoMatic (Bullock et al., 2010). However, these have several limitations to non-expert owing to being command line-based programs and their inability to handle all tasks required of a docking system. Such tasks include; Grid (configuration) file generation by ADT, conversion of import files to AutoDock standard format of PDBQT as they are not flexible with a variety of file formats, ligand preparation, and Visualisation. This means that the majority of earlier docking GUI still required individuals to perform manual preparation before docking and visualization.

However, InstaDock provides an easy-to-use and automated performance towards automatic detection of imported ligand and receptor files, preparation of the files, generation of search space configuration parameter files, (Specifies grid size, grid location, and atoms to be utilized in docking), conversion of ligand files into PDBQT and assigning appropriate atoms, if necessary, in just one click of a button. it also provides multiple docking parameters such as binding affinity, p*Ki*, torsional energy, and ligand efficiency for a compound and this also illustrates the diversity of the easy-to-use program. (Mohammad et al., 2021)

Alongside a single-click automated approach, InstaDock also provides a user-directed docking option by which a variety of standalone programs are provided in the tools section of the Interface which helps in virtual screening and other various Individual tasks such as receptor preparation, configuration generation, ligand preparation, file conversion, Hit identification, vina splitter, library splitter,

Inspection of receptor PDB files and visualization and complex generation.

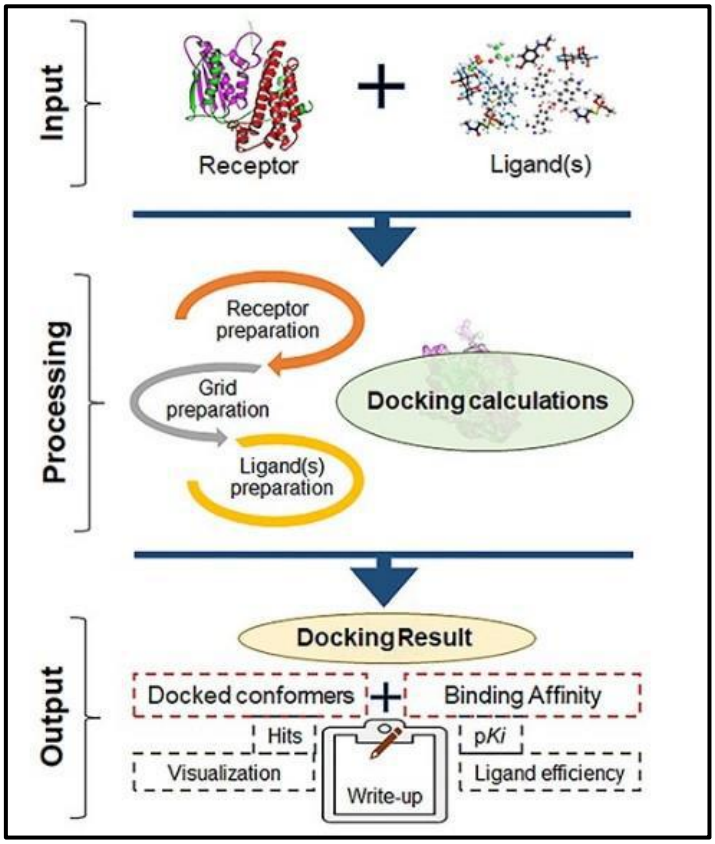

Fig. 4 (Mohammad et al., 2021) workflow of InstaDock illustrates the input of Receptor and Ligand files followed by automated processing of files and lastly output of docking results in a results folder containing different docked poses arranged according to binding affinity.

Limitations of computational protein modeling and molecular docking

Computational protein modeling energy functions that guide protein folding are aimed at improving computational power and therefore these functions only approximate resulting models. These approximations are due to the inability to accurately balance polar, and nonpolar interactions and solvation effects, particularly at the interfaces of proteins. (Kuhlman et al., 2019)

A single protein may display multiple activities as a consequence of having access to multiple protein structures either through allosteric regulation, post-translational modifications or other environmental factors. These multiple conformations are still a challenge to deal with in protein modeling.

The present mathematical protein modeling functions do not provide solutions to the prediction of protein activity; however, they just predict folding toward minimum energy. This leaves molecular docking (proteins with ligands or substrates for cases of enzymes) as the present method that can be used in predicting protein activity by specifying docking to amino acid residues expected to be involved in the desired biochemical process. However, it should be noted that the binding of a protein to a ligand does not mean it is activated or involved in a certain biochemical pathway. For example, binding of a substance to an enzyme does not only mean catalysis but also activation or inhibition. Also, binding of a transcription factor to DNA does not mean transcriptional activation but can also mean inhibition of transcription of DNA structural modifications. Therefore, finding a mathematical function that not only detects protein structure but also possible activity remains a challenge. (Del Río, 2021)

Bind of ligands to target proteins can sometimes follow synergism, where the binding of one alters the structure of the protein facilitating the being of the other. Currently, no molecular docking approach answers this, so there could remain a gap in the detailed understanding of how certain ligands affect proteins.

The existence of a variety of computational tools for molecular docking and protein modeling offers a wide range

of choices for scientists, However, this creates another problem; the need for a standardized tool or approach for testing and validation that will minimize the diversity of data from computational analysis.

Molecular docking utilizes 3D proteins that are already modeled towards their minimum conformations. It is with such stable conformations that prepared ligands are docked which means it will dock with the most stable conformations and not the others, However, this can be misleading as binding of ligands to even other conformations not necessarily having low energy could yield an overall stable and highly favorable complex. (Docking should not be static but flexible if accuracy is required)

AlphaFold: AI-driven Protein design (A solution to the folding problem)

Experimental X-ray and Nuclear Magnetic Resonance remain quite expensive and time-consuming for the determination of protein structure. Also, previous protein design approaches such as Rosetta and other user-friendly computational approaches could not achieve a reasonably high GDT meaning such computational approaches don't fully solve the folding problem which therefore led to further research utilizing machine learning as a probable solution.

AlphaFold is an artificial intelligence (AI) approach developed by Deepmind and fully depends on machine learning to solve the protein folding problem. This approach has produced an easy-to-use tool with good results that are very similar when compared to experimentally determined structures and it is better than any other prediction software with over 70% and 92% Global Distance Test (GDT) results from AlphaFold 1 (2018) and AlphaFold (2020) respectively.

AlphaFold carries three distinct steps towards protein folding prediction of an imported amino acid sequence of interest, first, it searches the Protein data bank (an important requirement for machine learning) to extract sequences with similarity with the input thereby generating a multiple sequence alignment (MSA) representation. It also pairs the Input sequence to form a pairwise alignment and searches structural databases for proteins with similar sequences. The templates extracted from the structural databases and pairwise alignment are utilized to generate a pairwise representation of the input and this represents the relationship of every pair of residues in the target protein. The second step occurs in the Evoformer and consists of two parts; the MSA representation tower and the pair representation tower which communicate with each other to yield a neural network that is responsible for the evaluation of residues of both pair representation and MSA data to create refined MSA and pair representations. The third and last step involves structural model generation utilizing both the refined representations to perform rotations and translations. The generated structural models can be recycled back to the Evoformer three or more times before yielding the final structure.

Local computational utilization of AlphaFold requires complicated computational power and a wide knowledge of the Linux operating system, which can be a limiting factor for most scientists. To solve this problem, AlphaFold can be accessed using cloud computing and a simple web interface available from the Phenix GUI. This offers options for uploading or protein sequences, and customizing (number of predicted models, template utilization from Protein Data Bank and MSA) all aimed at increasing accuracy by default. This shows an advantage of how even inexperienced scientists can perform highly accurate model prediction that is equally as accurate as experimental procedures. The interface continues to provide results with numeric indicators of model accuracy, such as plDDT (predicted local-distance difference test) values and PLE plots. (predicted aligned error, a confidence of pairwise orientation) (Mirdita et al., 2021; Jumper et al., 2021)

Therefore, the application of machine learning in Computational protein design represents a great achievement in the generation of highly accurate protein models that can be further utilized in other studies and fast improvement of the available size of data banks.

II. CONCLUSION

Structural data about proteins is increasingly growing and to deal with this, new analysis tools such as rstoolbox have to be designed to offer accessibility to users even with beginner-level coding experience, for analysis of CPD data and generation of novel protein sequences with new functions. There is however a challenge of developing software that is both highly accurate and easy to use. The only solution to this is the application of machine learning in computational protein design and analysis. This concept is seen in AlphaFold modeling software which is easy to use and produces models with over 90% GDT and this therefore leaves artificial intelligence as a future for an easy-to-use approach in computational protein design, modeling, and docking.

REFERENCES

[1] Abreu, R. M., Froufe, H. J., Queiroz, M. J., & Ferreira, I. C. (2010). MOLA: a bootable, self-configuring system for virtual screening using AutoDock4/Vina on computer

clusters. *Journal of cheminformatics*, *2*(1), 10. https://doi.org/10.1186/1758-2946-2-10

[2] Adolf-Bryfogle, J., & Dunbrack Jr., R. L. (2013). *The PyRosetta Toolkit: A Graphical User Interface for the Rosetta Software Suite. PLoS ONE, 8(7), e66856.* doi:10.1371/journal.pone.0066856

[3] Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K., Labonte, J. W., Pacella, M. S., Bonneau, R., Bradley, P., Dunbrack, R. L., Jr, Das, R., Baker, D., Kuhlman, B., Kortemme, T., & Gray, J. J. (2017). The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of chemical theory and computation*, *13*(6), 3031–3048. https://doi.org/10.1021/acs.jctc.7b00125

[4] allakyan S, Olson AJ. (2015) Small-molecule library screening by docking with PyRx. Chemical Biology. Springer, Humana Press, New York, NY, 243–50.

[5] Ben A Meinen, Christopher D Bahl. (2021) Breakthroughs in computational design methods open up new frontiers for *de novo* protein engineering, *Protein Engineering, Design and Selection*, Volume 34, gzab007, https://doi.org/10.1093/protein/gzab007

[6] Bonet, J., Harteveld, Z., Sesterhenn, F., Scheck, A., & Correia, B. E. (2019). rstoolbox - a Python library for large-scale analysis of computational protein design data and structural bioinformatics. *BMC bioinformatics*, *20*(1), 240. https://doi.org/10.1186/s12859-019-2796-3

[7] Boyle NMO, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. (2011) Open Babel: An open chemical toolbox. J Cheminform. 3:33. https:// doi.org/10.1186/1758-2946-3-33 .

[8] Bullock, C. W., Jacob, R. B., McDougal, O. M., Hampikian, G., & Andersen, T. (2010). Dockomatic - automated ligand creation and docking. *BMC research notes*, *3*, 289. https://doi.org/10.1186/1756-0500-3-289

[9] Chang, M. W., Ayeni, C., Breuer, S., & Torbett, B. E. (2010). Virtual screening for HIV protease inhibitors: a comparison of AutoDock 4 and Vina. *PloS one*, *5*(8), e11955. https://doi.org/10.1371/journal.pone.0011955

[10] Chen CY, Georgiev I, Anderson AC, Donald BR. (2009) Computational structurebased redesign of enzyme activity. Proc Natl Acad Sci U S A. 106(10): 3764–9.

[11] Weichenberger, C. X., & Sippl, M. J. (2007). NQ-Flipper: recognition and correction of erroneous asparagine and glutamine side-chain rotamers in protein structures. *Nucleic acids research*, *35*(Web Server issue), W403–W406. https://doi.org/10.1093/nar/gkm263

[12] Cosconati, S., Forli, S., Perryman, A. L., Harris, R., Goodsell, D. S., & Olson, A. J. (2010). Virtual Screening with AutoDock: Theory and Practice. *Expert opinion on drug discovery*, *5*(6), 597–607. https://doi.org/10.1517/17460441.2010.484460

[13] Del Río, G. (2021) Challenges in the Computational Modeling of the Protein Structure—Activity Relationship. *Computation 9*, 39. https://doi.org/10.3390/computation9040039

[14] Di Muzio E, Toti D, Polticelli F. DockingApp. (2017) a user friendly interface for facilitated docking simulations with AutoDock Vina. J Comput Aided Mol Des 31:213–8.

[15] Dilip, A., Lešnik, S., Štular, T., Janežič, D., & Konc, J. (2016). Ligand-based virtual screening interface between PyMOL and LiSiCA. *Journal of cheminformatics*, *8*(1), 46. https://doi.org/10.1186/s13321-016-0157-z

[16] Dolinsky TJ, Nielsen JE, McCammon, Baker NA. (2004) PDB2PQR: An automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. Nucleic Acids Res. 32:665 –7. https://doi.org/10.1093/nar/gkh381 .

[17] Feinstein, W. P., & Brylinski, M. (2015). Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets. *Journal of cheminformatics*, *7*, 18. https://doi.org/10.1186/s13321-015-0067-5

[18] Frey KM, Georgiev I, Donald BR, Anderson AC. (2010) Predicting resistance mutations using protein design algorithms. Proc Natl Acad Sci U S A. 107(31):13707–12.

[19] Gainza, P., Nisonoff, H. M., & Donald, B. R. (2016). Algorithms for protein design. *Current opinion in structural biology*, *39*, 16–26. https://doi.org/10.1016/j.sbi.2016.03.006

[20] Harris R, Olson AJ, Goodsell DS. (2007) Automated prediction of ligand-binding sites in proteins. Proteins. 70:1506 –17. https://doi.org/10.1002/prot .

[21] Jumper, J., Evans, R., Pritzel, A. et al. (2021) Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589.

[22] Kuhlman, B., Bradley, P. (2019) Advances in protein structure prediction and design. *Nat Rev Mol Cell Biol* **20**, 681–697. https://doi.org/10.1038/s41580-019-0163-x

[23] Lavecchia, A., & Di Giovanni, C. (2013). Virtual screening strategies in drug discovery: a critical review. *Current medicinal chemistry*, *20*(23), 2839–2860. https://doi.org/10.2174/09298673113209990001

[24] Leman, J. K., Weitzner, B. D., Lewis, S. M., Adolf-Bryfogle, J., Alam, N., Alford, R. F., Aprahamian, M., Baker, D., Barlow, K. A., Barth, P., Basanta, B., Bender, B. J., Blacklock, K., Bonet, J., Boyken, S. E., Bradley, P., Bystroff, C., Conway, P., Cooper, S., Correia, B. E., … Bonneau, R. (2020). Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nature methods*, *17*(7), 665–680. https://doi.org/10.1038/s41592-020-0848-2

[25] Lewis SM, Kuhlman BA (2011) Anchored design of protein-protein interfaces. PLoS ONE 6: e20872.

[26] Li, Q., & Shah, S. (2017). *Structure-Based Virtual Screening. Methods in Molecular Biology, 111–124.* doi:10.1007/978-1-4939-6783-4_5

[27] Mirdita, M., Ovchinnikov, S., Steinegger, M. (2021) ColabFold - Making protein folding accessible to all bioRxiv 2021.08.15.456425.

[28] Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., & Olson, A. J. (2009). AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, *30*(16), 2785–2791. https://doi.org/10.1002/jcc.21256

[29] Prakhov ND, Chernorudskiy AL, Gainullin MR. VSDocker. (2010) a tool for parallel high-throughput virtual screening

using AutoDock on windows-based computer clusters. Bioinformatics. 26:1374–5.

[30] Ravindranath, P. A., Forli, S., Goodsell, D. S., Olson, A. J., & Sanner, M. F. (2015). AutoDockFR: Advances in Protein-Ligand Docking with Explicitly Specified Binding Site Flexibility. *PLoS Computational Biology*, *11*(12), e1004586. https://doi.org/10.1371/journal.pcbi.1004586

[31] Samish I. (2009) Search and sampling in structural bioinformatics. *Structural Bioinformatics.* 207–236.

[32] Sandeep, G., Nagasree, K. P., Hanisha, M., & Kumar, M. M. (2011). AUDocker LE: A GUI for virtual screening with AUTODOCK Vina. *BMC research notes*, *4*, 445. https://doi.org/10.1186/1756-0500-4-445

[33] Scarpino, A., Ferenczy, G. G., & Keserű, G. M. (2020). Covalent Docking in Drug Discovery: Scope and Limitations. *Current pharmaceutical design*, *26*(44), 5684–5699. https://doi.org/10.2174/1381612824999201105164942

[34] Schrödinger L. (2002) The PyMOL molecular graphics system. Pérez, S., & Tvaroška, I. (2014). Carbohydrate-protein interactions: molecular modeling insights. *Advances in carbohydrate chemistry and biochemistry*, *71*, 9–136. https://doi.org/10.1016/B978-0-12-800128-8.00001-7

[35] Sheng-You Huang. (2018) Comprehensive assessment of flexible-ligand docking algorithms: current effectiveness and challenges, *Briefings in Bioinformatics*, Volume 19, Issue 5, September 2018, Pages 982–994, https://doi.org/10.1093/bib/bbx030

[36] Mohammad, T., Mathur, Y., & Hassan, M. I. (2021). InstaDock: A single-click graphical user interface for molecular docking-based virtual high-throughput screening. *Briefings in bioinformatics*, *22*(4), bbaa279. https://doi.org/10.1093/bib/bbaa279

[37] Valdés-Tresanco, M. S., Valdés-Tresanco, M. E., Valiente, P. A., & Moreno, E. (2020). AMDock: a versatile graphical tool for assisting molecular docking with Autodock Vina and Autodock4. *Biology direct*, *15*(1), 12. https://doi.org/10.1186/s13062-020-00267-2

[38] Vieira, I. H. P., Botelho, E. B., de Souza Gomes, T. J., Kist, R., Caceres, R. A., & Zanchi, F. B. (2023). Visual dynamics: a WEB application for molecular dynamics simulation using GROMACS. *BMC bioinformatics*, *24*(1), 107. https://doi.org/10.1186/s12859-023-05234-y

[39] Yang, Y., Hu, B., & Lill, M. A. (2017). WATsite2.0 with PyMOL Plugin: Hydration Site Prediction and Visualization. *Methods in molecular biology (Clifton, N.J.)*, *1611*, 123–134. https://doi.org/10.1007/978-1-4939-7015-5_10

[40] Yuan, S., Chan, H. C. S., & Hu, Z. (2017). *Using PyMOL as a platform for computational drug design. Wiley Interdisciplinary Reviews: Computational Molecular Science, 7(2), e1298.* doi:10.1002/wcms.1298