Computers in Biology and Medicine 147 (2022) 105695

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Protein structural bioinformatics: An overview

Vinícius de Almeida Paiva [a], Isabela de Souza Gomes [a], Cleiton Rodrigues Monteiro [a,b], Murillo Ventura Mendonça [c], Pedro Magalhães Martins [d], Charles Abreu Santana [d,e], Valdete Gonçalves-Almeida [f], Sandro Carvalho Izidoro [c], Raquel Cardoso de Melo-Minardi [d,e], Sabrina de Azevedo Silveira [a,*]

[a] *Department of Computer Science, Universidade Federal de Viçosa, Viçosa, 36570-900, Minas Gerais, Brazil*
[b] *Computer Center, Instituto Federal do Sudeste de Minas Gerais, Manhuaçu, 36905-000, Minas Gerais, Brazil*
[c] *Institute of Technological Sciences, Campus Theodomiro Carneiro Santiago, Universidade Federal de Itajubá, Itabira, 35903-087, Minas Gerais, Brazil*
[d] *Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, 31270 – 901, Minas Gerais, Brazil*
[e] *Department of Biochemistry and Immunology, Universidade Federal de Minas Gerais, Belo Horizonte, 31270 – 901, Minas Gerais, Brazil*
[f] *Computer Center, Campus Almenara, Instituto Federal do Norte de Minas Gerais, Almenara, 39900-000, Minas Gerais, Brazil*

ARTICLE INFO

Keywords:
Structural bioinformatics
Docking
Molecular dynamics
Molecular visualization
Structure prediction
Mutation
Atomic interactions
Binding site prediction

ABSTRACT

Proteins play a crucial role in organisms in nature. They are able to perform structural, catalytic, transport and defense functions in cells, among others. We understand that a variety of resources do exist to work with protein structural bioinformatics, which perform tasks such as protein modeling, protein docking, protein molecular dynamics, protein interaction, active and binding site prediction and mutation analysis. Nonetheless, they are generally spread all over different online repositories. For the students or professionals interested in working with protein structural bioinformatics, it may not be trivial to know what resources he/she should learn/use or where these could be accessed. Here, the main subareas in the field of protein structural bioinformatics are introduced with a brief description, and we point to and discuss several online resources, such as methods, databases and tools, in order to give an overview of this research field. Furthermore, we developed Protein Structural bioinformatics Overview (PreStO), a web tool available at http://bioinfo.dcc.ufmg.br/presto/, to organize and make it possible to retrieve these online resources based on a search term. We believe that this paper can be a starting point for potential bioinformaticians to trace a path that can be followed to build competencies and achieve knowledge milestones in the context of protein structural bioinformatics.

1. Introduction

Bioinformatics is a research field that aims to search for knowledge on biological data, more specifically biomolecules, through models and algorithms from Computer Science. It covers the collection, storage, retrieval, manipulation, and modeling of data for analysis, visualization, or prediction through the development of computational algorithms and strategies [1]. It also employs knowledge of Physics, Chemistry, Statistics, and Mathematics in the solution of biological problems, thus promoting the development of the various areas involved [2,3]. A variety of biological data, such as nucleotide sequence, gene expression, protein sequence, and protein structure, which give rise to omics data, have been generated at a fast pace, and demand integration, organization, and the development of reliable and accurate computational strategies to improve the understanding of the relationship between structure and function of biomolecules [4,5].

A natural question that arises is how to prepare human resources to work in such a broad field as bioinformatics. The International Society for Computational Biology (ISCB) has set a Curriculum Task Force that has published a set of reports and papers focused on the definition and refinement of curricular guidelines for training in bioinformatics. This work can be followed in the papers [6–8]. In a general manner, the idea was to define the core competencies, different types of users (profiles), and how these competencies match each type of user. This

[*] Corresponding author.
E-mail addresses: vinicius.d.paiva@ufv.br (V.d.A. Paiva), isabela.s.gomes@ufv.br (I.d.S. Gomes), cleiton.monteiro@ifsudestemg.edu.br (C.R. Monteiro), murillo.ventura1711@unifei.edu.br (M.V. Mendonça), pmartins@dcc.ufmg.br (P.M. Martins), chales.cas@gmail.com (C.A. Santana), valdete.almeida@ifnmg.edu.br (V. Gonçalves-Almeida), sandroizidoro@unifei.edu.br (S.C. Izidoro), raquelcm@dcc.ufmg.br (R.C.d. Melo-Minardi), sabrina@ufv.br (S.d.A. Silveira).

https://doi.org/10.1016/j.compbiomed.2022.105695
Received 19 October 2021; Received in revised form 1 June 2022; Accepted 2 June 2022
Available online 18 June 2022

was performed in an iterative process involving not only ISCB but also other education venues, including the Global Organisation for Bioinformatics Learning, Education, and Training (GOBLET).

Structural bioinformatics comprises data resources, algorithms, and tools for investigating, analyzing, predicting, and interpreting biomacromolecular structures. More specifically, we are interested in protein structural bioinformatics. Proteins are large and complex molecules that perform a myriad of functions in organisms. They are formed by the union of amino acids and can take on different sizes and shapes. Proteins perform much of the work in cells and are necessary for the structure, function, and regulation of tissues and organs in organisms [9]. Currently, there is a growing number of protein structures and sequences deposited in specialized databases, mainly provided by the advance in genomic sequencing technologies and methods for determining structures [10].

Initially, information about proteins was available in terms of their amino acid sequence. 2021 marks 50 years of the Protein Data Bank (PDB) [11], a catalog of all macromolecule structures we know to date. In 1971, the PDB was established at Brookhaven National Laboratory, led by Walter Hamilton, which comprises 7 structures. The genome sequencing of organisms and the rapid increase in the number of three-dimensional macromolecular structures available have given rise to structural bioinformatics. Structural Bioinformatics has two major goals: the creation of general-purpose methods for manipulating information about biological macromolecules and the application of these methods to solve problems in biology, to generate new knowledge [12].

Since the early years of PDB, much progress has been made in protein structure prediction, with emphasis on AlphaFold2 [13] in the CASP14 competition, which was able to predict protein domain structures with accuracy close to that of experimental methods [14]. AlphaFold2 was released with over 300k protein models and is scheduled to cover over 100 million proteins, which demands structural biology tools that can be applied on a proteome-wide scale thus posing new challenges and opportunities for these tools [15].

In recent years, the growing volumes of biological data have demanded scalable and data-driven bioinformatics models, algorithms, and tools. Thus, the need for life science scientists to develop basic bioinformatics skills has increased. Even experienced bioinformaticians need to update their knowledge and skills, so new algorithms and tools can be developed in response to the advances in science. However, basic data science, bioinformatics, and programming can be still relatively rare in life science curricula, and most biologists receive little or no formal training for the computational aspects of their field [16]. Another important aspect of bioinformatics is that finding appropriate material can be challenging as it is often scattered on the internet or hidden in its home institution.

To tackle this challenge, the scientific community proposed a set of rules named FAIR (Findable, Accessible, Interoperable, and Reusable) to support potential users in easily finding these digital objects [17]. Click2Drug is an online repository that provides a comprehensive list of protein structure tools and databases. On its website (https://www.click2drug.org/), users can find hundreds of works from the most diverse applications, mainly focused on drug design research. Zhang Lab (https://zhanggroup.org/) brings a series of tools, publications, and databases related to structural bioinformatics. Protein structure prediction, protein–protein interactions, and ligand-docking are some examples of the work carried out by the group. In bio.tools (https://bio.tools/), users find an extensive record of software, databases, and services related to bioinformatics. These resources can be found by searching for expressions or keywords and are described using accurate semantics and syntax. Datasets2Tools [18] consists of an online repository with thousands of analyses, databases, and computational tools. The website (https://maayanlab.cloud/datasets2tools) analyzes the resources according to some features, such as accessibility, interoperability, and reusability.

Nonetheless, despite the relevant contributions of the mentioned works, repositories that are focused on a specific topic in a subarea or bring a list of tools, without delving into them to some degree, do not provide a basic understanding of the field and its subareas and do not mention positive and negative aspects of each tool. Hence the potential user has no support in choosing an appropriate tool for a task of interest. In addition, some repositories are outdated and present resources that are no longer available, such as LISE [19], Pocketome [20], and OpenAstexViewer [21], which were not accessible as of the date of writing.

To overcome these challenges, in this work we present an overview of the main subareas of protein structure bioinformatics. In the next sections, we discuss docking, molecular dynamics, molecular visualization, structure prediction, mutation analysis, catalytic and binding site prediction, and databases. For each subarea, we present a brief introduction and definition followed by a set of widely used tools to address common tasks in the subarea. To bring as many examples of tools as possible, in addition to the works presented here, several other examples are described in the Supplementary Material. For each tool, in turn, we try to give an idea of how it works, with the intuition behind the proposed algorithm (when it is the case), as well as positive aspects and limitations, in addition to mentioning which other methods/tools it was compared with. It is important to point out that this paper does not present an exhaustive list of methods/tools in the field of structural bioinformatics of proteins, as it would be infeasible since there is a huge number of methods and tools available. We consider this set of tools a relevant starting point for those that are beginning in the field.

1.1. PreStO

We developed Protein Structural bioinformatics Overview (PreStO), an interactive visualization tool that organizes hierarchically or in a tabular manner all the resources presented in this paper, pointing to their papers' DOI and URL and providing a search tool that allows users to retrieve these resources. This tool is shown in Figure S1 (Supplementary Material), with a tree graph of the subareas and a wordle (cloud of words) created based on frequent words from titles and abstracts of each online resource described in the manuscript. Moreover, a table is available, which lists the title, abstract, DOI, and tool URL of all resources. The user may select a keyword from the wordle or submit a search term of interest. The web tool uses the Term Frequency–Inverse Document Frequency (TF–IDF) [22] to measure and rank the importance of a search term for each online resource, based on their titles or abstracts. Finally, the user can download the search result in BibTeX or CSV format.

2. Docking

The elucidation of the interactions between proteins and ligands, or between proteins, is very relevant to various fields of knowledge, such as the development of new drugs and elucidation of molecular recognition [23,24]. This stimulated the emergence of the technique of molecular docking, which is now widespread. It consists of calculating the best orientation that a molecule assumes to form a stable complex with a receptor. It is composed of two fundamental components: the sampling algorithm and the scoring function [25].

The sampling algorithm intends to generate different poses of the ligand arrangement inside the delimited simulation box. Such algorithms can essentially be classified into two categories: systematic and stochastic searches. The first seeks to exhaustively explore all the space allowed for simulation and has a high degree of freedom as a consequence. For such reason, the systematic search is a typical method for blind docking, whose main objective is to find the protein binding site. The second one, on the other hand, is a refinement search, through genetic algorithms, for example, and may not explore all the simulation space allowed. It is often used as local docking, when the binding site is already precisely known or after a blind docking step [24,25]. The pose generation is heavily affected by the simulation space and the flexibility of the molecules involved. With the evolution of graphics boards, the determination of poses using much more flexible molecules has become faster. Therefore, the docking efficiency in predicting the interaction poses has increased [23].

The scoring function, the other fundamental component of docking technique, aims to rank the conformations obtained by the sampling algorithm and predict the binding affinity between the two components. It is an equation that can estimate the thermodynamic properties of the complex arrangement and rank instability order. Currently, this aspect of the docking methodology is the bottleneck to define the best fit between two molecules, since it is a challenge to accurately calculate the affinity of several molecular groups [25]. The scoring functions can be of three sorts: empirical, force field and knowledge-based. Empirical functions estimate the affinity by adding important contact terms (H-bonds, for example) whose values have been previously determined. Force field ones estimate the affinity by calculating the parameters with a specific force field. The knowledge-based apply machine learning to predict the affinity [23].

Here, we present some tools that are commonly used for docking, summarized in Table 1. For protein–ligand, we describe GOLD, AutoDock Vina, and SwissDock, for protein–protein, we present ClusPro, pepATTRACT, HDOCK, and ZDOCK, and the hybrid tool HADDOCK (it can be used to perform protein–ligand and protein–protein docking). The distinction between the tools is commonly observed, due to the discrepancy in the complexity of the structures involved. The flexibility of the system components is the most important factor in this matter. Ligands, which are small molecules, have more freedom of translation, rotation and torsion than proteins, which have their movements restricted by a well-defined tertiary structure. Under this difference in the structures involved in docking, the scoring functions for affinity prediction and ranking are developed differently to best suit each scenario.

The molecular docking technique is widely used to predict molecular interactions. Several tools have been developed to perform this type of calculation, with certain specificities when it comes to a ligand or a peptide/protein. Based on the tools listed, we realize that the most classical tools for protein–ligand docking mostly run locally, since ligands are small molecules, which facilitates the positioning calculation. As for protein–protein docking, the tools developed tend to be web servers, which allows simulations to run more efficiently on more robust machines. In both types, there is a constant evolution of the scoring functions so that the energy description of the conformations is increasingly accurate.

It is important to emphasize that docking assays are usually used as an initial step in more complex simulations, such as molecular dynamics and metadynamics. Since it is a simulation performed in a vacuum and the molecular motions are restricted, its results may not represent reality with precision. However, these simulations are highly dependent on well-defined three-dimensional structures, mainly obtained by crystallographic methods. For this reason, it is possible to perform minimizations and quick molecular dynamics simulations prior to the docking step, to ensure the quality of the structures that were not defined with enough resolution [35].

It is important to mention that the quality of methods has increased over time due to initiatives as Critical Assessment of PRedicted Interactions (CAPRI), which conducted an independent assessment of docking techniques. Its main goal is to verify the quality of protein–protein docking of experimentally determined complexes to predict three-dimensional structures [36].

2.1. GOLD

Genetic Optimisation for Ligand Docking (GOLD) was one of the first pieces of software developed for automated protein–ligand docking that explores the full flexibility of the ligand and partial for the protein close to its active site. It uses a genetic algorithm to fit the ligand into the protein binding site, modifying its position, orientation, and conformation. GOLD passed through a lot of improvements during the years, one of them is the scoring function that, nowadays is composed of four ligand energy parameters: internal and external van der Waals, external H-bond, and internal torsion. Another component, internal H-bond, may be added if necessary [26]. Currently, GOLD is commercially available by The Cambridge Crystallographic Data Centre (CCDC) at https://www.ccdc.cam.ac.uk/solutions/csd-discovery/Components/Gold/.

2.2. AutoDock Vina

AutoDock Vina is a widely used docking program freely available for academic purposes, designed to be quicker and more accurate in binding prediction than its ancestor AutoDock4. It also has the advantage of calculating the grid maps automatically and generating a clean output for the user. A grid map is a set of regularly spaced points, used to segment the region of interest in the docking calculation. Vina can use multiple CPU cores, which is one of the reasons behind its speed-up. The positioning algorithm is based on iterated local search global optimizer, which is a combination of stochastic global and local optimization approaches. There is a gradient optimization method that uses the derivatives of the empirical scoring function among the ligand position, orientation, torsion tree and rotatable bonds. These parameters are set on an additional software, AutoDock Tools, previously developed to manage AutoDock files [27]. Vina has the limitation of lacking graphical user interface. Besides, it uses a very specific file format (pdbqt), which makes it difficult to visualize the results. AutoDock Vina can be downloaded at http://vina.scripps.edu/download.html.

2.3. SwissDock

SwissDock is a web service developed by the Swiss Institute of Bioinformatics, which is part of the tools designed in the SwissDrugDesign project. It is a protein–ligand docking platform available at http://www.swissdock.ch with a friendly user interface, which makes the tool more accessible for non-experts in molecular simulations and programming. Among the advantages of a web service, we can highlight that all docking processing and the visualization of the results occur on the server-side, which does not require great computational resources from the user. SwissDock works with the EADock DSS engine, which allows performing blind and local docking in four steps, involving evolutionary algorithm and energy calculation with CHARMM force field. The limitation of the tool is the restricted control over the simulation parameters. [28]. An example of the results page of the tool is presented in Figure S2.

2.4. ClusPro

ClusPro is a web server used for performing protein–protein docking, developed in 2004, but with constant updates to the present day. It is available at https://cluspro.org. The system only requires two protein structures in PDB format, and the user may or may not change the default settings of the simulation, which is completed within four hours. The docking calculation is divided into three steps. The first step consists of rigid-body docking based on Fast Fourier Transform (FFT) correlation to generate more than one billion possible conformations. From this set, the one thousand most stable poses are selected for the second clustering step to identify the most likely conformations. From there, thirty poses are selected to go through the last energy refinement step, using the CHARMM force field in the energy parameterization. Finally, the algorithm returns ten models that represent the best fit between the proteins. The scoring function used in the FFT correlation is based on structure interactions, in addition to energy terms, such as electrostatics and desolvation [29]. ClusPro is a recognized software by CAPRI, but its scoring function based on rigid body method is not as accurate as that of flexible methods [37].

Table 1
Summary of docking tools.

Name	URL	Features	Limitations	Reference
GOLD	https://www.ccdc.cam.ac.uk/solutions/csd-discovery/Components/Gold/	Partial flexibility close to protein active site Frequently improvements Internal H-bonds as parameter	Only supports protein–ligand docking Commercially available	[26]
AutoDock Vina	http://vina.scripps.edu/download.html	Open source Gradient optimization method Quicker and more accurate than AutoDock 4 Calculate grid maps automatically Can use multiple CPU cores	No GUI It uses pdbqt format Dependence of AutoDock Tools Only supports protein–ligand docking	[27]
SwissDock	http://www.swissdock.ch	Friendly user interface Web service Blind and local docking	Only supports protein–ligand docking Limited parameters control	[28]
ClusPro	https://cluspro.org	Web-server Frequently updates CAPRI validation	Only supports protein–protein docking Only supports PDB format Limited scoring function	[29]
pepATTRACT	https://bioserv.rpbs.univ-paris-diderot.fr/services/pepATTRACT/	Web service Does not require the protein binding site Precise results	Only performs peptide–protein docking 18 h of processing It does not run on GPU	[30]
HDOCK	http://hdock.phys.hust.edu.cn/	It can receive aminoacid sequence as input User friendly interface Allows the incorporation of experimental information Online molecular visualization Allows protein–RNA/DNA docking	Dependence of high-quality homologous complex templates	[31]
ZDOCK	http://zdock.umassmed.edu	User-friendly interface Editable scoring function Display input and output visualization with Jmol About 12 min of runtime	No clustering No post processing analysis included	[32]
HADDOCK	https://wenmr.science.uu.nl/haddock2.4/	Web server Allows all kind of docking Allows docking with more then 2 molecules Clustering and post processing analysis	Limited control for new users	[33,34]

2.5. pepATTRACT

pepATTRACT is a blind peptide–protein docking web service that is available at https://bioserv.rpbs.univ-paris-diderot.fr/services/pepATTRACT/. It does not require any information about the protein binding site and the peptide structure. pepATTRACT brings together several algorithms already developed by the research group. It works, initially, by performing a rigid docking, with the three most probable conformations of the peptide, for the putative determination of the binding site, using the ATTRACT algorithm. Then, a refinement with local docking is performed in two steps, which allows great peptide flexibility to explore as many conformations as possible. First, iATTRACT is used, and then a molecular dynamics stage is executed with AMBER. This refinement takes approximately 18 h to run, and the software does not integrate with GPU, even when it is possible, which we consider a limitation [30].

2.6. HDOCK

The HDOCK is a web server designed to perform protein–protein docking based on template modeling and structure prediction. The great advantage of this tool is that it can receive the three-dimensional structure in PDB format or the amino acid sequence as input. If the sequence is provided, HDOCK predicts the structure using MODELLER. From there, the hybrid docking strategy predicts the interaction between the chains, and experimental information can be incorporated to calculate the interaction and the scoring function. In the end, the top 100 poses are available for download, and the top 10 can be visualized on the web page. HDOCK is one of the first docking platforms to allow protein–RNA/DNA docking, which has its intrinsic scoring function. High dependency of high-quality homologous complex templates to execute hybrid docking is an issue faced by the platform [31]. It can be accessed on http://hdock.phys.hust.edu.cn/.

2.7. ZDOCK

ZDOCK Server is a user-friendly protein–protein docking web platform that allows editing the scoring function parameters and displaying input and output visualization. The process is divided into three steps: (1) the input, which can be carried out from an upload of the structure or the PDB id; (2) selection of contacting or blocking residues; and (3) calculation and results for visualization with JMol. However, the platform allows the user to download the results and use them in other analysis software. ZDOCK runtime is about 11.5 min and succeeds with CAPRI 'Acceptable' criteria. Eventually, the authors of the tool intend to develop clustering algorithms for the results and post-processing analysis on the webserver. ZDOCK is available on http://zdock.umassmed.edu [32].

2.8. HADDOCK

HADDOCK (High Ambiguity Driven biomolecular DOCKing) is an open web server available on https://wenmr.science.uu.nl/haddock2.4/ in which the user can perform protein–protein and protein–ligand docking and use a variety of molecules as input, including cyclic peptides and glycans. The platform allows docking calculating with two molecules up to 20. It has many interfaces, varying from Easy to Guru tier, that allow different levels of user control of the simulation parameters. The Easy tier is the most basic, which explains its limitations. The user can upload two structures, define active/inactive residues and submit to the platform by applying the default parameters for docking calculation. The full access and control can be obtained by the Guru tier, in which the user can modify over 500 parameters [33,34]. The result page, as observed on Figure S3, brings the summary of the complex pose clusters in several energy scores and the graphical comparison between clusters, using Bokeh Python library.

3. Molecular dynamics

Molecular dynamics is a technique of computational calculation in which Newton's classical laws of motion are extrapolated to molecular systems. To do so, the behavior of molecules is described by force fields that prescribe the spatial and energetic pattern temporally. In addition to this approximation, a cutoff is also established to determine the maximum interaction distance between two atoms to reduce the number of operations that the algorithm needs to perform [38]. These considerations make molecular dynamics a very efficient method for many applications, such as protein folding prediction, complex system stability, and inhibition potential of proteins with small ligands.

There are three types of molecular dynamics simulation: the conventional, which is used for several studies, such as protein folding, complex stability analysis, and structure refinement; the QM/MM (quantum mechanics/molecular mechanics) simulations, which can be used to simulate chemical reactions; and metadynamics, in which the free energy of systems can be predicted [39]. In all simulation cases, we employ algorithms for calculating the integration of the motion and energy equations, to obtain the vectors describing velocity, position, and force for all atoms in the system. These vectors are defined at small time intervals throughout the simulation, which allow obtaining the system's physical properties at the time interval considered in the full simulation [40]. Before any of these simulations, there is a preparation of the molecules, in which minimization steps are applied to ensure the quality and stability of the structures.

A critical factor for these simulations is the setting of equations of motion and energy, which are directly related to the force field selected for its parameterization. Several force fields have been developed with different resolutions, either expliciting all atoms or not or even working with entire groups. The choice for one force field or another should take into account the properties to be measured during the simulation, the level of detail of the interactions, atomic or interchain scale, for example, and the class of biomolecules. Neglecting to select the correct force field may result in low accuracy and long delays, as problems in the simulation are usually detected only when the simulation is finished, which can take up to months [41,42].

A force field is defined by Eq. (1), which represents the bonded, nonbonded interactions and specific parameters of the force field. Bonded interactions include intramolecular bonds, valence angles, rotation, and dihedrals. Nonbonded interactions involve intermolecular bonds, such as electrostatic, dispersion, and Pauli exclusion [43].

$$\begin{aligned} E_{total} = {} & \sum_{bonds} k_b(b-b_0)^2 + \sum_{angles} k_\theta(\theta-\theta_0)^2 \\ & + \sum_{dihedrals} k_\chi[1+cos(n\chi-\sigma)] \\ & + \sum_{nonbonded}\left(\varepsilon_{ij}\left[\left(\frac{R_{min,ij}}{r_{ij}}\right)^{12} - 2*\left(\frac{R_{min,ij}}{r_{ij}}\right)^{6}\right] + \frac{q_iq_j}{r_{ij}}\right) \\ & + E_{forcefield} \end{aligned} \tag{1}$$

CHARMM, Amber, GROMOS, and OPLS-AA, all of them developed by academic research groups, are the most common force fields used in molecular dynamics simulations. They are all present in the major molecular dynamics software, except for GROMOS, which was specially developed for GROMACS. Even though they were designed focusing on proteins, some modifications were added, and other classes can be described as well, such as nucleic acids, lipids, carbohydrates, and small molecules. The main differences between these force fields are related to nonbonded interaction and the "improper" dihedrals (this characteristic is specific for proteins, but it is estimated for the other biomolecular classes). For this reason, these parameters should be the criteria employed to decide which force field to use [43,44].

We discuss the main tools used in academia for performing molecular dynamics simulations, which are summarized in Table 2. In most of them, there is a great diversity of force fields available, and they allow more complex simulations, such as metadynamics, for example. Molecular dynamics is a technique that has been widely used to better understand the formation of complexes between macromolecules and ligands, as well as broader phenomena on a cellular scale. Both the software that allows this type of calculation and the force fields used are constantly evolving to bring the simulations closer to reality. Another aspect that has been incremented in software is the diversification of the simulations that can be performed, such as metadynamics and QM/MM, which expands the events that can be studied computationally. In addition, molecular dynamics simulations are computationally expensive. Therefore, powerful GPU processors are used to accelerate and parallelizing calculations of complex systems, and recent versions of this kind of software include GPU setup [45,46].

3.1. Amber

Amber [47,48] is a tool for performing molecular dynamics simulations created in the 1970s, which has been maintained by the development community to this day. This tool is a set of several programs that work together to configure, execute and analyze molecular dynamics simulations. To execute the simulation, Amber has several molecular dynamics force fields and simulation software packages, including source code and demo files.

Currently, Amber consists of three different molecular dynamics simulation tools: Sander, Pmemd, and Pmemd.cuda. In summary, Sander has been the primary platform for Amber's computing and development, to explore new features. Pmemd and Pmemd.cuda are more focused on maximizing performance, thus working to implement Sander features on high-performance architectures. Several force fields are supported by Amber, including pairwise amino and nucleic acid variants, fixed-charge protein force fields, Charmm force fields, Glycam series of force fields, polarizable force fields, and Amoeba force field.

The tool is available free of charge via the following web address: https://ambermd.org/. The web page contains a lot of important information about Amber, such as reference manuals, a description of force fields, and several tutorials, including installation and running information. It is also possible to find demonstration materials for educators on the website, where several examples and commented codes are made available, to provide an overview of relevant algorithms.

3.2. GROMACS

GROMACS, a software commonly used in the chemical area, is another example of a molecular dynamics simulation tool [50,51]. It was designed to work with biochemical molecules, such as protein, nucleic acids, and lipids, but it has also been used in non-biological systems, such as polymers, due to its efficiency in calculating non-bonded interactions, which generally account for most of the simulation. GROMACS comes with a variety of built-in force fields for molecules, including GROMOS53a6, OPLS, Encad, OPLS-AA/L, Amber99SB-ILDN, CHARMM27, among others. This tool was created to provide efficient modeling and has been developed through open-source and free software development, with a codebase generated through the sharing of infrastructure and contributions from several researchers.

The ease of use of the tool is a strength of GROMACS. First of all, it does not use any scripting language, just provides a simple interface. The simulations generated by the tool can be monitored as they are carried out, and the remaining time to finish the simulation is provided. In addition, GROMACS allows the users to select the accuracy and comprises a large set of tools for trajectory analysis, using lossy compression that can offer a compact way to store trajectory data. GROMACS is a free-of-charge tool that is available on its website through the link: https://www.gromacs.org/. At this address, a variety of information is provided, including features of the tool, funding, and support information. The website contains a link to extensive documentation of GROMACS, which supplies download addresses and release notes for various of its versions.

Table 2
Summary of molecular dynamics tools.

Name	URL	Features	Limitations	Reference
Amber	https://ambermd.org/	Free of charge Standalone software Usually applied in biomolecular simulations Several force fields available Many available tutorials	Only available for Linux	[47,48]
LAMMPS	https://www.lammps.org/	Free of charge Standalone software Available for Windows, Linux and macOS Run in parallel or single processor MPI and OpenMP parallel support	Does not allow interactively visualization Lacks output data plotting resources	[49]
GROMACS	https://www.gromacs.org/	Free of charge Standalone software Several force fields available GROMOS force field Many available tutorials	Only available for Linux	[50,51]
CHARMM	https://www.charmm.org/	Free of charge Standalone software Available for Linux and macOS MPI and OpenMP parallel support Usually applied in biomolecular simulations	Requires license for commercial use	[52]
NAMD	https://www.ks.uiuc.edu/Research/namd/	Free of charge Standalone software Available for Windows, Linux and macOS Many available tutorials Extensive documentation and training section	Does not support non-equilibrium MD Does not soppoort rigid bond between heavy atoms	[53]
Desmond	https://www.schrodinger.com/products/desmond	Commercial tool Standalone software Source code available for non-commercial use State-of-the-art GPU acceleration technology Focused on scalability	Only available for Linux	[54]
OpenMD	https://openmd.org/	Free of charge Open source project Standalone software Available for Windows, Linux and macOS MPI parallel support	–	[55]
ORAC	http://www1.chim.unifi.it/orac/	Free of charge Standalone software Scaling parallel support via OpenMP Focused on simulations at atomistic level complex systems	Only available for UNIX operating systems	[56–58]
AMMP VE	https://www.ddl.unimi.it/cms/index.php?Software_projects:AMMP_VE	Free of charge Standalone software Available for Windows and Linux Can be embedded in other programs	Lacks of support information and documentation	[59]
ACEMD	https://www.acellera.com/products/molecular-dynamics-software-gpu-acemd/	Free of charge Standalone software Parallel support Designed for GPUs	Needs a license to use all its resources Only available for Linux	[60]
Tinker	https://dasher.wustl.edu/tinker/	Free of charge Standalone software Available for Windows, Linux, macOS and Android Support very large molecular systems	Requires license for commercial use Does not focus on free energy calculations	[61]
Chemsol	https://laetro.usc.edu/doc/chemsol/cs-1.0/index.html	Free of charge Webserver Works with aqueous solutions	Lacks of support information and documentation	[62]
Abalone	http://www.biomolecular-modeling.com/Abalone/index.html	Free of charge Standalone software Several force fields available Education section available	Only available for Windows	[63]
YASARA dynamics	http://www.yasara.org/products.htm	Commercial tool Standalone software Available for Windows, Linux, macOS and Android Run simulations in aqueos solutions Shows simulations in real-time Allows screen interactions during simulations	–	[64,65]

3.3. CHARMM

CHARMM [52], an acronym for Chemistry at Harvard Macromolecular Mechanics, is a tool for molecular simulation with wide application in multi-particle systems with a set of energy functions, improved sampling methods, and implicit solvent models. It was developed for the study of biomolecules, such as peptides, proteins, carbohydrates, nucleic acids, small molecule ligands, and lipids, which can be used to solve problems of multi-particle systems.

The software provides computational apparatus, including molecular minimization, path sampling methods, dynamics, and free energy estimators. The calculations performed with the tool can be used for various functions and energy models. CHARMM contains a variety of analysis and model creation tools and can deliver high performance on a large group of platforms, including those using parallel computing and GPU.

The tool is available for free for academic use. Information about CHARMM is available at https://www.charmm.org/. The website provides a forum for discussion of various topics involving the tool, such as installation issues, discussion of parameters, and questions about molecular dynamics and chemistry. In addition, there is a section dedicated to the academic use of CHARMM, which covers aspects of the tool along with external packages that can be used. The user also has access to the CHARMM documentation, which contains a complete list and description of its modules.

3.4. NAMD

NAMD is another example of a molecular dynamics simulation program [53]. This tool features an efficient parallel implementation and offers scalable system performance, running on up to hundreds of thousands of processing cores. It is used to simulate large systems with millions of atoms, whose purpose is to enable biomedical research through practical supercomputing. NAMD is a tool that aggregates several state-of-the-art algorithms to perform simulations in thermodynamic sets, including CHARMM, AMBER, OPLS, and GROMOS biomolecular force fields. Like many other molecular dynamics simulation tools, NAMD is available as open-source, with no commercial use.

NAMD supports classical molecular dynamics simulations in explicit solvent, with periodic boundary conditions, in addition to coarse-grained models, in implicit solvent, with unrelated periodic boundary conditions (non-periodic and semi-periodic). Some characteristics can also be considered to attach external forces in the molecular simulations, similar to the Colvars module (where the user can set in calculations some variables as control parameters), flexible adaptation of structures to electron density maps, and methods for sampling acceleration.

Users can find NAMD at https://www.ks.uiuc.edu/Research/namd/, a website that contains information about the tool, including an overview, news, and announcements. In addition, it is possible to find a training session, with several development workshops. Besides, together with the documentation section, the user can obtain support for using the tool.

3.5. OpenMD

[55] proposed OpenMD, an open-source molecular dynamics engine that allows working with simulations of proteins, lipids, nanoparticles, transition metals, and several other systems, through the use of atoms with orientational degrees of freedom. OpenMD was implemented with a parallel computing approach, the Message Passing Interface (MPI), which makes it an efficient method to develop molecular dynamics simulations. The tool provides several trajectory analyses and utility programs. In addition, OpenMD supports various force fields and even allows users to define their own, using empirical energy functions.

OPENMD is available for free through the website https://openmd.org/, with plenty of information about the tool, including news and examples, as well as its related source code. Users can find a Documentation section where a manual and documentation code of the OpenMD are available. In addition, the website has a Community section, with discussions on various subjects involving OpenMD and molecular dynamics, both for common users and developers. OpenMD is available for Linux, Windows, and macOS operating systems.

3.6. ORAC

ORAC [56–58] is an open-access molecular dynamics simulation method that works with several complex molecular systems at the atomic level. Its main feature is the use of state-of-the-art molecular dynamics algorithms along with the flexibility to manipulate the most diverse types and sizes of molecules. In addition, ORAC has algorithms for biological systems with periodic boundary conditions, can perform simulations for computing electrostatic interactions, and allows the simulation of molecular systems in different thermodynamic ensembles.

ORAC was mainly developed using the Fortran language and is compatible with most compilers currently available. However, it is only available for UNIX-based systems. The recent versions present an implementation of parallel architectures, using MPI (Message Passing Interface) and OpenMP, which guarantees a faster and more efficient execution. In addition, some features were implemented, such as support for the fast switching double annihilation method (FS-DAM) [66,67], for the calculation of binding free energies in drug-receptor systems.

At the web address http://www1.chim.unifi.it/orac/, users can find information about ORAC, including links and instructions for downloading the current version as well as older releases. An extensive manual, with information about ORAC inputs, interactions, and simulations, is available to users to provide support for using the tool. In addition, from the ORAC website, there is a link to PrimaDORAC [68], a web interface that generates parameter files and topology for molecular dynamics from organic molecule coordinates.

3.7. Tinker

[61] proposed the Tinker method, a molecular mechanics and dynamics package for molecular modeling. Several parameter sets are found in the tool, including Amber, CHARMM, OPLS, and AMOEBA, among other force fields. Tinker has a series of algorithms for the most diverse applications, such as free energy calculations, continuum solvation treatments, vibrational analysis, reaction field treatment of long-range electrostatics, fitting charge, multipole, and polarization models, among many others.

The method, implemented in Fortran, uses a module concept and, together with dynamic memory allocation, allows working with very large molecular systems. To further improve its performance, Tinker uses a parallel approach, through OpenMP, in various aspects of its code. A new package, Tinker9 [69], was recently developed to provide performance on single NVIDIA GPU-based systems.

Tinker is available free of charge for academic and non-profit use through the website https://dasher.wustl.edu/tinker/. Before use, the tool must be downloaded and installed locally, on any machine with Windows, Linux, or macOS operating systems. Users will find many resources on the website, such as user and installation guides, Force Field Parameter Sets, and the source code of the tool.

3.8. YASARA dynamics

YASARA Dynamics [64,65] is software for molecular dynamics simulations, which is part of the YASARA tool package. This package consists of four stages: View (stage 1), Model (stage 2), Dynamics (stage 3), and Structure (stage 4). Each stage contains all the features of the previous stages, plus specific additional functions of a given application.

Stage 1 of the YASARA package (View) consists of a set of features for exploring macromolecular structures iteratively. Stage 2, YASARA Model, provides resources to explore, analyze and model macromolecules in a production environment. Stage 3, YASARA Dynamics, contains all the functionality of the previous stages and also has support for molecular simulations. Some features of YASARA Dynamics can be mentioned, such as treatment of long-range electrostatic interactions, parallel execution of the simulations, and calculation of energies and binding energies. Furthermore, an interesting feature of the tool is that it allows users to view the molecular simulation in real-time, unlike some methods that work like a black box. YASARA Dynamics makes available the use of its force fields, NOVA and YAMBER, besides allowing users to employ other known force fields such as AMBER. Stage 4, YASARA Structure, in addition to the molecular dynamics package (and all previous stages), contains functionalities for the identification and validation of macromolecular structures.

YASARA Dynamics, along with all other stages of the package, can be found via the web address: http://www.yasara.org/products.htm. The tool can be used on Linux, macOS, Android, and Windows operating systems. Only stage 1, YASARA View, is provided free of charge. It is necessary to purchase a license to use the other stages, including YASARA Dynamics. On the website, users can also find much more information about the YASARA package, as well as news and user support resources.

4. Molecular visualization

The advance of new experimental techniques and structure determination methods for obtaining atomic positions in three-dimensional space has contributed significantly to studies related to the functions of proteins. An increased number of 3D structures deposited in the PDB (Protein Data Bank) database enabled new studies [70]. Many of the so-called molecule viewers are freely available, and open-source enables open science aiming to learn about structures and physical–chemical phenomenons in proteins.

Molecule viewers are versatile and can be integrated with web platforms, mobile or stand-alone systems in studies involving structural similarities, mutations, drug design, and molecular docking, among others. In addition, graphical user interface (GUI) and simulations are critical for students entering the field of bioinformatics and computational biology [71]. The understanding of molecular structures is becoming more democratized as a result of the many molecule visualization software systems made available. Currently, in addition to high-quality images, the developers of such tools are also concerned with usability and offer several new features.

This constant evolution brings significant gains, especially when it comes to studies involving non-static and complex systems with several components. The concern with the graphical interface also allows the extrapolation of structural data interpretation for researchers who do not have as much contact with lines of code, thus greatly enriching research discussions. Nowadays, many systems are available freely or under a commercial license. In the next subsections, we present a relevant set of 3D molecule viewers largely used by the scientific community. A summary of the 3D viewer tools described in this section is presented in Table 3.

4.1. PyMOL

PyMOL is the most classical software when it comes to biomolecule visualization. It is open-source software for educational and commercial purposes that is distributed by Schrödinger, Inc. Written in C, C++, and Python, PyMOL can be installed on several operating systems, such as Unix and Windows, and through the tool's official website (http://www.pymol.org/), the user can obtain its license [72]. PyMOL can be used to produce both images and movies of biomolecular systems in various types of representations, as described in Figure S4. In addition, several plugins have been developed that can be installed and add several more features to the application, including protein and ligand modeling, PL docking, Gromacs-based molecular dynamics simulations, Dehydron (plugin that displays the hydrogen interactions of the backbone that are unprotected from water attacks, which compromises protein folding) and even QM/MM calculation [88].

The software consists, basically, of three interfaces: (a) Graphic User Interface (GUI) with menu bars, which allows the user to achieve several features of the visualization; (b) mouse controls that allow precise selections of molecule regions and movements for better structure positioning; and (c) command lines that can be used with no need of the two other options. The last one is a powerful way to modify the structure and achieve additional function combinations [89]. Due to the many uses of PyMOL, a support community has been developed, the PyMOLWiki (http://www.pymolwiki.org). There are tutorials, descriptions of features and plugins, image-making protocols, and answers to users' most frequently asked questions. The page is constantly updated with new information and features of the software.

4.2. VMD

VMD (Visual Molecular Dynamics) [73] is a software developed by the Theoretical and Computational Biophysics Group from the University of Illinois to allow visualization, modeling, and analysis of biomolecules, both static and in a simulated trajectory, using a built-in scripting system and three-dimensional graphics. A sample of the VMD interface is represented in Figure S5, in the Supplementary Material. We can mention some technical features of the tool, such as support for various computing platforms (MacOS X, Unix, or Windows), efficient management of memory usage, use of multicore processing, and GPU acceleration. Furthermore, the VMD has no limit on the number of atoms, residues, molecules, or trajectory frames, being limited only to the amount of memory available in the machine. The visualization features of the tool include multiple coloring and rendering methods, support for over 60 different file formats, support for using other visualization and simulation tools such, as NAMD [90], as well as extensions capable of allowing users to create their routines for molecular analysis.

VMD is distributed free of charge through its website https://www.ks.uiuc.edu/Research/vmd/, which also includes its source code. A range of information about the tool is available on the website, including news, announcements, development topics, and related publications. There is also a user support section, where the VMD documentation is available, with user guides, bug lists, and FAQs. A series of tutorials and manuals are available to users and include a very detailed walkthrough with descriptions and images of various problems in visualizing molecular structures. Examples of the use of VMD are also found in conjunction with other chemical or biological tools, such as AMBER [47,48], which works with molecular dynamics, and APBS [91], which performs electrostatic calculations.

4.3. Chimera

Chimera [74] is a molecular visualization tool written in Python, which is designed to be simple to use on several platforms and capable to produce classical graphics. The tool can be divided into two parts: core and extensions. The first one consists of a series of basic services and molecular visualizations, in addition to ensuring that the extensions can run robustly. Extensions allow for higher-level functionality of Chimera, are loaded via the tool's menu, and used when users access it, on-demand. We can mention some of the extensions used by Chimera such as Multiscale (useful to explore large molecular assemblies), Multalign Viewer (useful to align structures), ViewDock (integrated to DOCK, it helps to perform the screening of ligands), Movie (which allows visualization of molecular dynamics trajectories) and also Volume Viewer (to present three-dimensional information).

The authors developed a new version of the tool, ChimeraX [92], to work with larger structures while maintaining high performance. This new version contains some new features and advantages, such as high-performance manipulation and rendering of a large number of atoms, interactive ambient-occlusion lighting, and new windows, panels, and bars that allow users to navigate between the features of the tool more simply and clearly, plus a platform for virtual reality, made for Steam VR systems.

Table 3
Summary of the 3D viewer tools.

Name	URL	Features	Limitations	Reference
PyMOL	http://www.pymol.org/	Open-source Creating movies Cross-platform Support different input file formats Use command lines High quality images	Not on easy-of-use for new users Programming-knowledge for scripts No official support	[72]
VMD	https://www.ks.uiuc.edu/Research/vmd/	Open-source Cross-platform User support Support different input file formats GPU acceleration Creating movies	Does not work well representations of cyclic proteins Bug in the old "cartoon" representation	[73]
Chimera	https://www.cgl.ucsf.edu/chimera/	No-commercial free Cross-platform Creating movies Support other tools	No longer under active development No official support	[74]
NGL Viewer	http://proteinformatics.charite.de/ngl	Open-source Web application Allows the use of plugins	Support PDB, mmCIF, and GRO input file Default viewer PDB Not create movies	[75]
3Dmol.js	https://3dmol.csb.pitt.edu/	Open-source Web application JavaScript language Support different input file formats	Used in web project Not create movies	[76]
EzMol	http://www.sbg.bio.ic.ac.uk/ezmol/	No-commercial free Web application High quality images Easy-of-use for new users	Support PDB file only Not create movies	[77]
Schrödinger	https://www.schrodinger.com	Commercial Integrated computer platform Version free for academic purposes Support other tools User support High quality images	General purpose viewer Focused on drug design	[78]
MOE	https://www.chemcomp.com/Products.htm	Commercial Cross-platform Structure database associated GPU acceleration High quality images User support	General purpose viewer	[79]
Jmol	http://jmol.sourceforge.net/	Open-source Cross-platform Web and desktop application Support different input file formats	Java language specific	[80,81]
GLmol	https://www.glmol.com/	Open-source Web application JavaScript and GLSL languages	Used in web project Not create movies	[82]
DS Visualizer	https://www.3ds.com/products-services/biovia/	No-commercial free Integrated computer platform Cross-platform	General purpose viewer	[83]
Crystal Studio	http://www.crystal0studio.com/products.php	Commercial High quality images Structure database associated Support other tools	General purpose viewer	[84]
CueMol	http://www.cuemol.org	Open-source Web and desktop application Cross-platform High quality images Support other tools	No official support	[85]
YASARA View	http://www.yasara.org/products.htm}view	No-commercial free Integrated computer platform Cross-platform High quality images GPU acceleration Support different input file formats	Only standard ASCII characters	[86]
TextMol	https://cvcweb.oden.utexas.edu/cvcwp/software/texmol/	Open-source Cross-platform GPU acceleration High quality images	PDB and PQR formats only No official support	[87]

The tool is available for free through the link https://www.cgl.ucsf.edu/chimera/. On the website, users find a wealth of information about using Chimera, including its documentation, a download section, publications, datasets, and related software. In the documentation section, users have at their disposal guides and tutorials about the tool, visualization, and analysis videos, release notes, and a series of commands that can be used in Chimera. In addition, a gallery section is available, with an extensive collection of images and animations that can be made from the software, including RNA base models, protein interfaces, binding footprints, and wobble motion.

4.4. NGL viewer

Proposed by [75], the NGL Viewer is an open-source web application that uses WebGL to allow online molecular manipulation and visualization. The tool is one of the default viewers for Protein Data Bank (PDB) 3D view (https://www.rcsb.org/3d-view/) [11], which can be integrated into Jupyter Notebooks. The NGL Viewer was developed focused on memory efficiency, which was solved by parsing the input files quicker using the binary Macromolecular Transmission Format (MMTF). This advantage of the tool determined its option to be used in the PDB and has registered from small structures to representations of entire viruses, which demands a lot of memory space [93].

The NGL Viewer can load and display the main structural files type (PDB, mmCIF, and GRO) in a diverse way, such as ball & stick, cartoon, and surfaces. Moreover, it also loads molecular dynamics trajectories for analysis [75,94]. On the tool's web page (http://proteinformatics.charite.de/ngl), users can use various features of the viewer in their browser, without the need to install local software. The tool has menus and icons that allow users to view and manipulate structures easily and make use of plugins. It is also possible to find a documentation section, where users find information about the use of the tool and descriptions of objects, components, and representations used in the visualization of molecules with NGL.

4.5. 3Dmol.js

Proposed by [76], 3Dmol.js is a tool for molecular visualization in an online environment, which is an object-oriented library, WebGL based on JavaScript language. Among the features offered by the tool, we can highlight the different ways of representing surfaces, atoms, and chains, as well as the detail on demand when users hover the mouse over some part of the structure. We can also mention some other relevant features, such as support for various file formats (such as pdb, mol2, xyz, sdf, and cube), molecular surface parallel computation, as well as various structure visualization styles, such as cartoon, stick, line, and sphere. Besides the implementation in JavaScript, 3Dmol was also implemented in Python with some function reductions and can be used in Jupyter Notebook.

3Dmol.js can be accessed at https://3dmol.csb.pitt.edu/, where users can also download the tool's code. On the website, users can find examples of interactive visualization of the tool, such as protein structures and atomic characteristics. A notable facility concerning 3Dmol.js is that all the visualizations (including its several variations regarding style, color, and other characteristics) can be presented in the web browser itself, with no need to install local software. It is also possible to use library features through the Jupyter notebook, in addition to finding information about the development of web applications using a featured API. 3Dmol.js provides documentation, FAQ, and contact section, besides a teaching section, which allows students to learn about molecular structures through an active learning environment.

4.6. Jmol

JMol [80,81] is one of the oldest open-source viewers for chemical structures in 3D used by students and researchers around the world. It started being developed in the Open-Science project by Dan Gezelter and is based on Java language, which turns it compatible with different operating systems, such as Windows, Linux, and Mac. Its modules have different purposes: Jmol application (desktop), JSmol (web pages), and JmolViewer (Java applications). Jmol supports a wide variety of file formats for input, automatically identified: mol, cif, pdb, xyz, and some specific ones, such as ADF (Amsterdam Density Functional), Gaussian, and SDF (V2000 and V3000). For output, the user can download the generated image in jpeg, png, or ppm or export it into a pdf file.

Jmol can represent the whole periodic table and it accepts a variety of atoms representations, such as van der Waals radius or absolute size, ionic radius, and dots (spheres, stars, tetrahedra, or octahedra). For biological macromolecules, the tool allows the summarization of the structure in classic representation models, such as backbone, trace, ribbons, strands, and cartoon. Jmol is available on http://jmol.sourceforge.net/, where we can also find the software documentation.

4.7. DS visualizer

Discovery Studio Visualizer or DS Visualizer is part of a suite of programs able to simulate and share analyses of small and macromolecules. It is developed and distributed by Dassault Systemes BIOVIA (https://www.3ds.com/products-services/biovia/). It has many collaborations from the academic community, which makes it relevant for scientific research.

It is free software and provides a comprehensive collection of features to capture the specific nuances of research. It supports scientific research and uses known software and algorithms developed originally by the scientific community, such as CHARMM (molecular dynamic), MODELLER (3D modelling), DELPHI (electrostatic potential), ZDOCK (molecular docking), and DMol3 (electronic property).

In addition, the program suite is used for simulations and calculations, Ligand Design, Pharmacophore modeling, Structure-based Design, Macromolecule engineering, QSAR (quantitative structure–activity relationship), ADME (proprieties pharmacokinetics and pharmacology for absorption, distribution, metabolism, and excretion) and more. For more details of features, access [83].

4.8. YASARA view

YASARA View is a potent and free 3D viewer (http://www.yasara.org/products.htm}view) used to explore macromolecular structure interactively. Its architecture has an innovative engine for high-efficiency graphics and computation on modern GPUs enabling many structures to be loaded simultaneously. Therefore, it is possible to produce publication-quality ray-traced images including labels. Besides, it is possible to integrate programs or scripts in Python language to run the viewer.

Its important features include a cross-platform and easy installation (Windows, Linux, and macOS in the same directory, run directly from a USB stick), support for over 70 molecular file formats, download of PDB files from the RCSB, measurement (distances, angles, dihedrals), alignment of multiple proteins based on their structure or sequence, using a variety of methods, and Parallel (orthographic) and perspective projection [86]. For more details, access http://www.yasara.org.

5. Structure prediction

The number of known protein sequences has been increasing exponentially in recent years, and about 150 million entries were deposited in the UniProtKB database [95]. However, sequence information alone is not enough to understand the protein function, and protein structure information is crucial for this task. There is a gap between the growth rates of the sequential and structural information due to intrinsic difficulties and the costly nature of the experimental determination. For example, the Protein Data Bank [96], a database of protein structures,

Table 4
Summary of the structure prediction tools.

Name	URL	Features	Limitations	Reference
SWISS-MODEL	https://swissmodel.expasy.org/	Free of charge Webserver or standalone version User-friendly interface Build models at different levels of complexity	Only return a single result	[100,101]
Modeller	https://salilab.org/modeller/	Free for academic use Standalone program Can perform de novo modeling of loops Several models optimization	No GUI available	[102,103]
I-TASSER	https://zhanggroup.org/I-TASSER/	Free of charge (non-commercial use) Webserver Uses a multiple thread approach Give functional annotation information	Slower predictions compared to other methods	[104,105]
AlphaFold	https://alphafold.ebi.ac.uk/	Free for academic use Standalone program Machine Learning-based model	May not provide fold prediction in context-free scenarios	[13,106]
LOMETS	https://zhanggroup.org//LOMETS/	Free of charge Webserver Give functional annotation information Focused on give quicker response	Does not refine threading models	[107,108]
RaptorX	http://raptorx.uchicago.edu/	Free of charge Webserver or standalone version Deep learning-based method Inter-residue/inter-atom distance and orientation probability distribuition	Only provides contacts, not distances	[109]
MPACK	http://curie.utmb.edu/mpack/	Free of charge Standalone program Multiple-templates used for modelling Support for MOLMOL visualization	Do not allow gaps in the secondary structure region	[110–112]
CABS	http://biocomp.chem.uw.edu.pl/CABSfold/	Free of charge Webserver Outputs a corse grained trajectory of conformations Use Jmol representation	Small set of best scoring models Predictions can take up to 12 h	[113]
Rosetta	https://www.rosettacommons.org/software	Free for academic use Webserver or standalone version Deep learning-based method Can model multi-chain complexes	No GUI available Not recommended to use on Windows	[114]
QUARK	https://zhanggroup.org/QUARK/	Free of charge (non-commercial use) Webserver Suitable for proteins that do not have homologous templates	Cannot submit multiple jobs at once	[115]
ModPipe	https://salilab.org/modpipe/	Free of charge Open-source software Standalone program Almost no manual intervention is needed	Do not calculate profile–profile alignments Only builds a single model per alingment	[116]

contains about 170 thousand entries. Moreover, predicting a protein structure using the amino acid sequence as a starting point remains an unsolved problem in Bioinformatics.

Due to the importance of the theme, initiatives such as the Critical Assessment of Protein Structure Prediction (CASP) [97], have promoted the development of the protein prediction methods and provided standard evaluation and definitions. The core of protein structure prediction is the assumption that the protein's native state is the one with the lowest free energy [98]. Thus, the methods of protein structure prediction combine sampling of possible conformations with a ranking of these conformations through energy functions, aiming to find the lowest energy state.

Methods to predict protein structure can be labeled into two approaches: template-based and template-free [99]. When predicting a structure of a protein amino acid sequence, if there are related structures previously determined in PDB, template-based methods can use these structures as examples to model the target sequence. Otherwise, if related proteins are not found, template-free methods try to predict the protein structure directly from the protein sequence, using energy functions combined with conformational sampling. Table 4 summarizes the structure prediction tools described in this section.

Beyond the protein structure, the design of ligands that interact with these proteins is crucial for predicting binding on active and allosteric regions. Therefore, some software has been developed to generate the best structural conformation of small molecules. These systems take molecular formulas, in the case of simple molecules, or more detailed linear representations, such as SMILES, and arrange the atoms three-dimensionally, respecting the geometry of the molecular orbitals. Some examples of this kind of tool are LigPrep [117], ChemSketch [118], Avogadro [119], SCIGRESS [120] and ChemDraw [121]. Despite its relevance, this type of tool is still underdeveloped compared to tools specifically focused on biomolecules [119]. We discuss these tools in the Supplementary Material (Section 5).

5.1. Template-based modeling

This class of methods predicts 3D protein models using examples that share high sequence identity with a target protein. This approach is effective when the query shares at least 30% sequence identity with the protein examples. In general, the steps of standard template-based modeling include the selection of a suitable structural template, alignment of the query sequence to the template, and assembling the 3D protein model according to the target-template alignment [122].

5.1.1. SWISS-MODEL

SWISS-MODEL [100,101] is a web-based modeling system used to build a protein structure model based on homology modeling. This method aims to build a 3D protein structure model using experimentally determined structures of templates that share an ancestry with the target sequence. First, a target protein is used as input, and its sequence serves as a query to find evolutionarily related proteins contained in the SWISS-MODEL template library SMTL [123]. Then, templates are aligned against the target sequence to verify whether templates can represent conformational states or cover different locations of the query protein. Finally, a 3D protein model is generated for each selected template, transferring conserved atom coordinates according to the target-template alignment [101]. Figure S6 shows the SWISS-MODEL result page. Different template structures can be analyzed using the 3D viewer.

5.1.2. Modeller

Modeller also uses structures that share some sequence identity to model protein structures. The input to the Modeller is an alignment file of the query sequence with its respective templates, the atomic coordinates of the templates, and a script file [102]. Modeller uses satisfaction of spatial restraints to perform comparative protein structure modeling. These spatial limitations include stereochemical restraints, such as bond length and bond angles, restraints of distance and dihedral angles in the query sequence, and statistical knowledge for angle and inter-atomic distances, obtained from experimental protein structures. The spatial restraints are combined into an objective function that is optimized [103].

5.1.3. I-TASSER

Another successful modeling method, I-TASSER [104,105], constructs 3D protein models by interactive treading assembly simulations. The target protein is compared against a set of representative protein structures aiming to find possible folds. The conformation in I-TASSER is represented by a feature of Cα atoms and side-chain centers of mass. The reassembling process is conducted by replica-exchange Monte Carlo simulations. The energy function elements of this method include information about secondary structure properties, backbone hydrogen bonds, and correlations based on the structural statistics from the PDB library. The lowest free-energy conformations undergo a refinement step to remove steric clashes and refine global topology [105].

5.1.4. AlphaFold

In the recent CASP13, the participating methods achieve remarkable progress due to the adoption of deep learning and the exploration of co-evolutionary information in protein structure prediction. AlphaFold is a co-evolution method that achieves impressive results by CASP standards [106]. Co-evolution methods work by constructing multiple sequence alignments of proteins homologous to the target protein. Such a class of methods infers spatial proximity between residues by detecting mutations that occurred in the same evolutionary timeframes in response to other mutations. Another feature of AlphaFold came from applying convolutional networks, showing the potential of deep learning for protein structure prediction. In the coming years, the trend of improvements in template-based modeling is expected to continue, due to the increasing amount of available structures. More recently, AlphaFold2 [13] was developed, which brought improvements using deep learning architectures. AlphaFold2 achieved the highest accuracy in CASP14.

5.1.5. RaptorX

Proposed by [109], RaptorX is a tool for predicting the structure and function of proteins, freely available for non-commercial use at http://raptorx.uchicago.edu/. The method was developed taking into account cases in which the target sequence is distant from the related protein templates. For such, a profile-entropy scoring method was used, which analyzes the number of non-redundant homologs and template structures. Conditional random fields (CRFs), which incorporate various biological signals, are also used. Finally, RaptorX implements a multi-template threading procedure to use multiple templates to model a single target sequence. The results obtained by RaptorX in CASP9 indicated values similar or even superior to those of other tools in the area.

5.2. Template-free modeling

It is not always possible to find experimental structures that share similarities with the target sequence. In this scenario, methods *ab initio* and *de novo* are used where there is no information about the target sequence. Due to the lack of a structural template, these methods require conformational sampling and ranking criteria by which near-native conformations can be chosen as candidates.

The basis for template-free approaches is the Anfinsen's thermodynamic hypothesis [98], which claims that the native state is the one with the lowest free energy. Thus, protein structure prediction methods combine sampling of alternative conformations and scoring functions to rank the sampled conformations and identify the state with the lowest energy. However, the size of conformational space that must be searched grows exponentially, making the exhaustive search infeasible. Therefore, the exploration of the conformational space is guided by search algorithms that navigate through the energy landscape towards near-native conformations [99].

5.2.1. Rosetta

Rosetta is an *ab initio* approach that generates protein models by assembling small fragments taken from the PDB library. For conformational search, multiple rounds of Monte Carlo minimization are carried out, where each move is evaluated by a scoring function, and the move is accepted based on the Metropolis criterion, according to the energy difference between the original and the new conformation [114]. Rosetta's scoring function is a linear combination of terms, including physically based and statistically derived, which describes elements such as non-covalent interactions, residue solvation, and backbone torsion angles. Rosetta is a consolidated tool for protein model prediction that has been very successful for the free modeling targets in CASP experiments.

5.2.2. QUARK

QUARK is an *ab initio* protein structure prediction pipeline based on continuous fragment assembly that uses both physics-based energy and knowledge terms [115]. The target sequence is threaded through non-redundant high-resolution PDB structures, and at each residue position, structural fragments (1 to 20 residues) are generated [124]. The scoring function of the threading is composed of torsion angle, solvent accessibility, and secondary structure matches. Replica-exchange Monte Carlo simulations are performed to assemble the fragments into complete models.

5.2.3. ModPipe

ModPipe [116] is an automated pipeline software used to calculate protein structure models from sequences. The modeling performed by the tool is based on four steps: fold assignment, sequence–structure alignment, model building, and model evaluation. First, for a protein sequence given as input, potential templates are found, and then the alignment between templates and the input sequence is calculated. ModPipe is then executed, which generates results for all templates found. Although the standard ModPipe protocol is a template-free approach, it is possible to add a wrapper to the tool, making it run in template-based model. ModPipe is available for free at https://salilab.org/modpipe/.

6. Mutation analysis

Missense mutations are a specific type of genetic mutation characterized by single base-pair substitution that results in the change of one amino acid by another. It is commonly observed in nature and can modify important properties in proteins, such as conformation, stability, flexibility, drug resistance and protein-small molecule or antibody-antigen affinities. These variants (mutations) often contribute to the emergence of serious diseases, such as cancer, which corroborates the relevance of their analysis for the emergence of new therapies.

The identification or prediction of their effects is an important ally in the prevention or treatment of these disorders [125]. Analysis of changes in the properties of the substituted amino acids contribute to the identification of potentially critical mutations that becomes a problem of enormous relevance. Furthermore, many researches aim to evaluate the effects of a point mutation with varied purposes, including the identification of new drugs or treatments.

The technological advance has enabled many computational approaches to be combined with research for the prediction and analysis of the effects of missense mutations on proteins. Several systems, using different approaches to variant analysis, are currently available and can be used by researchers around the world. In this section, we listed the main computational tools to help to understand mutations in various aspects.

A summary of the tools and methods described in this section is presented in Table 5.

6.1. EVmutation

EVmutation combines prediction and data visualization techniques to explore mutational effects on proteins. According to [126], it is an unsupervised statistical method that models protein residue interaction, providing quantitative data on the effects of mutations. For this, it uses a “global probability” approach to analyze the interactions between all pairs of residues. In the work conducted by the authors, the results computed by the tool were compared with indicators from 34 experiments involving genetic variations, showing itself as a viable method that can be applied in different contexts and species of interest. However, the work also describes some limitations, such as trends coming from more recent families and with low diversity.

EVmutation presents itself as an interesting alternative for studies related to the analysis of mutations, making available for free through a web platform (http://evmutation.org/) a set of predictions for human proteins. Currently, 9935 entries are provided, which can be accessed individually or distributed in the following datasets: mutation effects, evolutionary couplings and sequence alignments.

6.2. DynaMut2

DynaMut is a web tool that uses Normal Mode Analysis (NMA) to assess mutational effects on protein stability and dynamics. In [127], NMA is described as a computational approach capable of exploring harmonic motions and providing information about structure–function relationships. Its application can also produce higher performance, considering the use of simplified structural representations that contribute to the reduction of computational cost. The DynaMut prediction model uses a Random Forest classifier with 10-fold cross-validation.

The experimental results obtained from DynaMut were compared with those of methods already consolidated for the study of mutations: I-Mutant, Maestro, DUET, SDM2, mCSM, ENCoM and FoldX. Although the study suggests the tool as a more suitable approach for predicting “destabilizing and stabilizing” mutations, the results demonstrate a lower Pearson correlation in relation to the I-Mutant2, DUET and mCSM methods.

DynaMut is freely available (http://biosig.unimelb.edu.au/dynamut2/) and offers a simple interface, providing users with three analysis options: single mutation (for processing a specific mutation), multiple mutations (for batch processing of a list of mutations) and NMA. The tool was developed using the Bootstrap framework version 3.3.7 (front-end) and the Python programming language, through the Flask framework version 0.12.2 (back-end).

6.3. PMut

PMut is a web tool for the annotation of pathological variants in proteins. Its first version was released in 2005, as a neural network-based classifier trained with a dataset extracted from SwissProt (https://www.uniprot.org). In [128], the authors present a new version of the tool, including the PyMut software package, through which users can prepare their own predictors for specific families of proteins. This creates an advantage for those who want to integrate these prediction features into their applications. Access to PMut is freely available at http://mmb.irbbarcelona.org/PMut, which also includes a tutorial with usage guidelines.

The new version of PMut (PMut2017) uses a Random Forest classifier, and its evaluation was performed using different approaches: 10-fold cross-validation, blind validations with SwissVar and ClinVar entries, and comparisons with specific genes. In a blind validation with SwissVar inputs, for example, PMut obtained greater accuracy than other methods such as SIFT and PROVEAN, but is inferior to LRT and PON-P2 methods.

The PyMut package included in this new version was developed as a Python 3 module and is based on consolidated libraries such as NumPy (for numerical computation), Pandas (for data management), and Scikit-learn (for machine learning). This module can be downloaded and installed locally. Its source code is available at https://github.com/inab/pymut and also at the official Python package repository at https://pypi.python.org/pypi/pymut.

6.4. SNPnexus

SNPnexus [129] is a web tool for the analysis of genetic sequence variation. It uses a Perl pipeline and a MySQL database to perform instant functional annotations. An advantage of the tool is that it provides information about annotations already available, as well as different examples that can help the assembly of new sequences. The portal includes a section with guidance on each available annotation category.

The SNPnexus architecture is structured in a request and response scheme that comprises two layers: access layer (with different search options and output formats to support the study of variants) and storage layer (composed of data sets for annotations and auxiliary data sets). SNPnexus is freely available at http://www.snp-nexus.org. Its use depends on filling out a form so that the data is processed and the results sent to the user.

6.5. Interactome INSIDER

INtegrated Structural Interactome and Genomic Data browser (Interactome INSIDER) is a web tool for the analysis and enrichment of mutations specifically in human diseases. It allows user to explore mutations of diseases already known and available in different databases, as well as mutations included by other users. It provides a diverse set of pre-computed mutations that can be accessed through the tool portal available at http://interactomeinsider.yulab.org.

As a prediction mechanism, Interactome INSIDER uses a framework called Ensemble Classifier Learning Algorithm to predict Interface

Table 5
Summary of mutation analysis tools.

Name	URL	Features	Limitations	Reference
EVmutation	http://evmutation.org/	Free of charge Webserver available Unsupervised statistical method	User guide not available Predefined input data Trends coming from more recent families and with low diversity	[126]
DynaMut	http://biosig.unimelb.edu.au/dynamut2/	Free of charge Webserver available Normal Mode Analysis (NMA) and machine learning method (Randon Forest)	Performance: lower Pearson coefficient than methods such as I-Mutant2	[127]
PMut	http://mmb.irbbarcelona.org/PMut	Free of charge Webserver available Machine learning method (Randon Forest)	Performance: lower accuracy than methods such as PON-P2	[128]
SNPnexus	http://www.snp-nexus.org/	Free of charge Webserver available Functional annotation	Its use depends on filling out a form, so that the data is processed and the results sent to the user	[129]
Interactome INSIDER	http://interactomeinsider.yulab.org/	Free of charge Webserver available Machine learning method (Randon Forest)	User guide not available	[130]
Metadome	https://stuart.radboudumc.nl/metadome/	Free of charge Webserver available Meta-domain based	Predefined input data—only those available in the own database	[131]
Mutfunc	http://www.mutfunc.com/	Free of charge Webserver available Provides an extensive database with pre-computed forecasts	Does not allow considering other types of mutation, such as variations in the number of copies Does not include many organisms with frequently studied mutations	[132]
ActiveDriverDB	https://www.ActiveDriverDB.org	Open-source Webserver available Based on information about post-translational modifications (PTMs)	The related study does not present comparisons with other tools with the same purpose	[133]
Condel	http://bg.upf.edu/condel	Free of charge Webserver available Consensus tool	Requires the user to login to access the features	[134]
PredictSNP	https://loschmidt.chemi.muni.cz/predictsnp/	Free of charge Webserver available Consensus tool	Allows you to use only sequences as input (FASTA format)	[135]
I-Mutant2.0	https://folding.biofold.org/i-mutant/i-mutant2.0.html	Free of charge Webserver available Machine learning method (SVM)	User experience: does not provide a modern interface	[136]

(*continued on next page*)

Residues (ECLAIR), which combines 8 independent classifiers based on the Random Forest classification algorithm from the scikit-learn library. ECLAIR was compared with other prediction methods (PIER, PINUP, SPPIDER, CPORT and PRISE) obtaining a performance as good or even a little better in some cases, such as accuracy and recall metrics, for example [130]. The tool does not optionally include a user guide.

6.6. *I-Mutant2.0*

I-Mutant2.0 is a web tool to predict changes in protein stability after single point mutations [136]. This tool is based on a support vector machine (SVM) and can receive protein structure or sequence data as input.

The I-Mutant2.0 classifier trained and tested the input using a cross-validation procedure. The dataset taken from the Thermodynamic

Table 5 (*continued*).

Name	URL	Features	Limitations	Reference
FATHMM	http://fathmm.biocompute.org.uk/	Free of charge Webserver available Markov Models-based	Little interactive features for presenting results	[137]
SIFT	https://sift.bii.a-star.edu.sg/	Free of charge Webserver available Evolutionary conservation-based	User experience: does not provide a modern and intuitive interface	[138]
Polyphen	http://genetics.bwh.harvard.edu/pph2/	Free of charge Webserver available Evolutionary conservation and structure-based	The related study does not present comparisons with other tools with the same purpose User experience: does not provide a modern interface	[139]
AUTO-MUTE	http://proteins.gmu.edu/automute	Free of charge Webserver available Stand alone (version 2.0) Statistical methods of classification and regression	User experience: does not provide a modern and intuitive interface For the local version (2.0), you need to install additional packages	[140]
MDPPM	–	Molecular dynamics and classification methods (Randon Forest and KNN)	Performance: increased processing time due to intensive computational simulations	[141]

Database for Proteins and Mutants (ProTherm) were able to correctly predict 80% for structures and 77% for sequences of the dataset.

I-Mutant2.0 is available through a web interface at https://folding.biofold.org/i-mutant/i-mutant2.0.html. For both inputs (structures and sequences), the tool gives the main result the value of the free energy change or only its sign. However, considering a better user experience, the web tool does not offer a modern interface.

6.7. SIFT

In [138,142], the authors present the SIFT (Sorting Intolerant From Tolerant) method, a tool for analyzing mutations in proteins through sequence homology. Its functioning is based on the premise that important amino acids will be conserved, and alterations in positions with a high degree of conservation tend to be intolerant of substitutions.

To predict the impacts of these substitutions on protein function, SIFT considers the position where the change occurred and the type of change. Thus, the probability that an amino acid is tolerated in one position is calculated from a pre-established sequence alignment. Substitution is given as deleterious if the normalized value of this probability is less than a cutoff point. For [142], a limitation of the method is that it does not apply structural data to assess the effects of substitutions.

SIFT provides a toolbox with six features for data entry: two batch tools, which provides predictions for multiple proteins and their substitutions, and another four tools that provide detailed predictions for a single protein, considering all substitutions or only selected ones. SIFT is currently available through a web platform at https://sift.bii.a-star.edu.sg/. Considering a better user experience, it does not provide a modern and intuitive interface.

6.8. Polyphen

Polyphen [139] is a web-based tool for predicting the effects of non-synonymous coding SNPs (nsSNPs) on protein structure and function. Polyphen may also be relevant in discovering the structural basis of mutations in diseases, enabling an understanding of their molecular cause.

It is a server dedicated to the automatic functional annotation of encoding nsSNPs, which receives an amino acid sequence as input and, from which it performs a series of actions (figure process). The PolyPhen server was applied to annotate all SNPs deposited in the HGVbase database. For the authors, the availability of this collection of annotated data could be useful in selecting nsSNPs for association studies based on candidate genes. However, the study does not present comparisons of the tool with others with the same purpose. The tool is available in its second version (PolyPhen-2) at http://genetics.bwh.harvard.edu/pph2/. An example of the results page of the tool is represented in Figure S7. Considering a better user experience, the tool does not provide a modern interface.

7. Interactions at atomic/residue level

Studies involving information about protein structures and their interactions with different types of molecules have significantly grown in recent years, but determining aspects associated with such interactions still face challenges [143]. Several experimental and computational techniques have been used to study interactions involving proteins, such as determining, evaluating, and understanding structures and protein interactions, this has a fundamental importance in molecular biology. Many proteins associate with other molecules to perform their functions and form essential complexes in a large number of cellular functions, such as cell signaling, proliferation, DNA repair, and immunity [144]. Three of the main interactions involving proteins will be described here, along with computational tools involving them: protein–protein, protein–ligand, and protein–peptide interactions.

Protein–protein interactions (PPI) are physical inter-chain contacts that lead to the forming of protein agglomerates as a result of a biochemical process in a cell. According to [145], the definition of PPI has to take into account two aspects that involve the interaction interface. The first one says that the interaction interface should be deliberate and not accidental, and the second aspect says that the interaction interface must be non-generic. Protein–protein interactions deal with a wide variety of biological processes, including cellular and metabolic interactions. In addition, PPIs play an important role in predicting protein function and the druggability of molecules [146].

Biological macromolecules, such as proteins, interact with a large number of molecules with a high degree of specificity and high affinity, called ligands, which can be defined as any molecule capable of binding to a protein that does not belong to the protein class. [147]. The interactions between these ligands and proteins are called protein–ligand interactions (PLI), the second type of protein interaction discussed here. Such interactions play a key role in enzyme catalysis, signal transduction, and several other biochemical processes [148]. Thus, understanding the aspects involving PLIs helps to understand protein functions, furthermore, can be linked to the development of new drugs, for instance [149].

The last interaction mentioned here is the protein–peptide. These interactions are present in several cells of living beings and play an important role in the protein–protein interaction network, in addition to participating in signaling and regulation [150]. Structural analysis of protein–peptide interactions indicates that most peptides, when binding, do not perform conformational changes in the protein, minimizing the entropy cost of the binding [151]. Peptides usually bind in the largest pockets available in proteins, being completely located in cavities, bound in pockets or form, at the surface of a protein, beta-strand interactions [152]. In the next subsections, we will describe some tools that work with protein–protein, protein–ligand, and protein–peptide interactions. Table 6 summarizes the tools presented in this topic.

7.1. *Protein–protein interaction methods*

7.1.1. *PrePPI*

[153] present a method for protein–protein interactions (PPI) prediction based on three-dimensional structural information. The developed tool, named PrePPI (predicting protein–protein interactions), combines structural information along with other functional aspects and it is available free of charge at https://bhapp.c2b2.columbia.edu/PrePPI/. The tool involves a few steps, when a pair of proteins is given as input, a sequence alignment is first used to find structural representatives. Afterward, a structural alignment is performed to find close and remote structural neighbors. When two pairs of neighbors of structural representatives form a pair reported in the PDB, this is defined as a template for modeling the interaction between the input proteins.

For the evaluation of the created protein–protein interaction model, some metrics are used (more specifically, five empirical scores) to analyze properties from individual monomer alignments to their templates. PrePPI uses a Bayesian network to combine structural and non-structural aspects, bringing more reliability in PPI predictions, as well as identifying more interactions compared to using a single source of information. The results obtained by the tool allow the use of homology models that can be used to study the close and remote geometric relationship between proteins. The tool facilitates the generation of experimentally testable hypotheses and allows the creation of a structural model for PPIs that play an important role in biological systems.

7.1.2. *ppiGReMLIN*

[154] proposed ppiGReMLIN, a tool for analyzing protein–protein interactions with a fine level of granularity at both the residual and atomic levels. The method is freely available at https://ppigremlin.github.io/pages/files.html and uses a graph-based strategy, where the nodes represent the atoms of the proteins (labeled according to their physicochemical properties) and the edges the non-covalent interactions (based on the physicochemical information of their proteins atoms and distance criteria). For each PDB entry, connected components are calculated and serve as the basis for clustering analyses. This step is performed with the Spectral Clustering algorithm [164] and is performed to find similar graphs for the next step of frequent subgraph mining, which is executed to find arrays of conserved structures.

To demonstrate the tool's ability to describe and find structural arrangements, at the atomic level, at protein–protein interfaces, the authors used two databases containing protein–protein complexes: a serine protease dataset and a BCL-2 dataset (details, including graphical analyses about these datasets can be found on the tool's webpage). Thus, it was possible to deduce that ppiGReMLIN is capable of detecting substructures at protein–protein interfaces on a large scale, which were compared with relevant residues and interactions found in the literature. Tests performed with the tool showed that ppiGReMLIN can find conserved structures automatically and the results for each of the datasets used ranged from 69% to 100% regarding the accuracy, with 100% of recall.

7.1.3. *mCSM-PPI2*

[155] proposed the development of the mCSM-PPI2 tool that predicts the effects of mutation in protein–protein interactions. Although we have already discussed aspects of mutation analysis in Section 6, we think it is pertinent to bring this tool to exemplify protein–protein interactions, as this is the context in which mutations are studied in [155]'s work. To create the optimized predictor, the tool uses a graph-based approach to be able to model the effects of variations, including several aspects, like inter-residue interaction network, complex network metrics, information about evolutionary and energetic terms. The mCSM-PPI2 models geometric and physicochemical properties of protein–protein interactions and can be applied in studies involving small molecules and protein structures.

The main component of the tool is the use of graph-based structural signatures (mCSM), which represent the context of wild-type residues. In this model, nodes represent atoms, and edges represent the interactions between nodes. The physicochemical information is coded according to the residual properties of the amino acids and the distance between atoms is described by their properties, defined in signatures. The tool also uses a machine learning technique (which uses six new features in the training stage) along with the graph-based signatures approach to explore the effects of mutations on protein–protein bonds.

The authors also implemented a web server (available at http://biosig.unimelb.edu.au/mcsm_ppi2/) to host mCSM-PPI2. The tool offers two services on its website: the first one is used to analyze the effects of user-specified mutations, and the second one performs the prediction of mutation effects in the protein–protein context. As a result, the website shows the whole protein binding environment, together with an interactive 3-dimensional viewer. Furthermore, a 2D viewer that shows non-covalent interactions of wild-type and mutant structures is provided.

7.2. *Protein–ligand interaction methods*

7.2.1. *PLIP*

The PLIP (protein–ligand interaction profiler) [165] is a web service (available at https://plip-tool.biotec.tu-dresden.de/plip-web/plip/index) used for identification and visualization of non-covalent contacts in the context of protein–ligand interactions, working with 3D structures and allowing in-depth analysis involving patterns of such interactions. The tool, which is free and open-source, offers automated high-quality images, and session files PyMOL to create custom images in addition to results files for data processing.

Identification and report of protein–ligand interactions are performed by the PLIP algorithm in four different steps: preparation, functional characterization, rule-based matching, and filtering of interactions. In the first stage, preparation, the structure is hydrogenated and its ligands and binding sites are identified. The functional characterization step includes the detection of hydrophobic atoms and

Table 6
Summary of the protein interaction tools.

Name	URL	Features	Limitations	Reference
PrePPI	https://bhapp.c2b2.columbia.edu/PrePPI/	Free of charge (PPI database) Based on structural information of proteins	Only predicted interactions are available, not the method	[153]
ppiGReMLIN	https://ppigremlin.github.io/pages/files.html	Free of charge Standalone program Based on structural information of proteins Graph-based method Only requires python environment	Does not calculate water mediated interactions	[154]
mCSM-PPI2	http://biosig.unimelb.edu.au/mcsm_ppi2/	Free of charge Webserver Graph-based method	–	[155]
PLIP	https://plip-tool.biotec.tu-dresden.de/plip-web/plip/index	Free of charge Webserver Offers publication ready images	Outputs limited to binary interaction fingerprints Few statistical analysis available	[156]
LIGPLOT	https://www.ebi.ac.uk/thornton-srv/software/LIGPLOT/	Free for academic use Standalone program Available for Windows, Linux and macOS	Requires license to use	[157,158]
nAPOLI	http://bioinfo.dcc.ufmg.br/napoli/	Free of charge Webserver Works in large-scale scenarios Graph-based method	Not designed to detect interactions in complexes from virtual screening	[159]
LeView	http://www.pegase-biosciences.com/leview-ligand-environment-viewer/	Free of charge Standalone software Available for Windows, Linux and macOS	Generates 2D images only	[160]
BINANA	https://durrantlab.pitt.edu/binana/	Free of charge Webserver or standalone program Only requires python environment	–	[161]
PoseView	https://proteins.plus/2ozrpose}view	Free of charge Webserver Generate structure diagrams from scratch Describe moties at atomic detail following IUPAC conventions	Lacks of statistical data	[162]
GalaxyPepDock	http://galaxy.seoklab.org/pepdock	Free of charge Webserver or standalone program Template-based docking approach Uses energy-based optimization	Does not perform predictions of peptides in isolation	[163]

acceptors/donors for halogen and hydrogen bonds, in addition to a search for aromatic rings and charge centers. In the next step, putative interacting groups are combined by applying geometric criteria. Finally, in the last step, filtering interactions are performed to eliminate redundant or overlapping interactions.

It is possible to apply the information brought by the web service in docking result evaluation, drug development, and repositioning and binding site similarity evaluation. As input, the webserver accepts a protein–ligand complex in PDB format and outputs 2D and 3D diagrams, visualization files, and various details involving interaction patterns for each binding site.

7.2.2. LIGPLOT

LIGPLOT is an algorithm for plotting protein–ligand interactions developed by [157]. The tool allows users to create two-dimensional representations of protein–ligand complexes from PDB input data files. The functioning of the algorithm can be summarized in a few steps. First, connectivity is calculated from 3D coordinates and hydrogen bond information, then rotatable bonds are identified and all ring groups are flattened. Finally, the structure is unrolled and cleaned up and a graphical representation is created through a postscript file. LIGPLOT is available at https://www.ebi.ac.uk/thornton-srv/software/LIGPLOT/ and it has an extensive operating manual session, where the users can learn about the tool's operation, as well as links to references and FAQ.

To improve LIGPLOT, [158] proposed the development of LigPlot+, a tool alleged to be a successor to the original algorithm and which is available on the following website: https://www.ebi.ac.uk/thornton-srv/software/LigPlus/. This new version also generates two-dimensional representations of protein–ligand interactions, like the first one, but with improvements. The first one is a new interface, developed using Java language, which offers diagram editing through click-and-drag mouse operations in an improved plotting environment. LigPlot+ also allows superposition of related diagrams, making it easier to visualize similar protein–ligand complexes and gives the users the option to visualize the representations in the PyMOL and RasMol tools.

7.2.3. nAPOLI

nAPOLI (Analysis of Protein–Ligand Interactions), proposed by [159], is a tool for analyzing protein–ligand interactions. The method works with the analysis of conserved interactions of protein–ligand complex datasets, along with interactive visualizations and reports of residues and atoms interactions. The approach brought by the authors consists of using bipartite graphs to model the protein–ligand interactions so that the atoms are represented by the nodes of the graph (characterized by their physicochemical properties) and the edges indicate the interactions between the atoms. Then, similar ligands are grouped to elucidate conserved interactions in groups of ligands. For this, the clustering algorithms GenerateMD [166] and Ward [167], both from ChemAxon, are used. Finally, superposition is used to find equivalences of residues and conserved interactions in various complexes.

The tool is available free of charge at http://bioinfo.dcc.ufmg.br/napoli/. On this web page, users are faced with two main menus: dataset submission and dataset analysis. In the first one, nAPOLI entries are inserted with information on the PDB ID of the proteins, chain, and information about the ligands, in addition to aspects about structural alignment. Still, in the dataset submission menu, users can even compose a new dataset and also define several parameters of the tool, which involve aromatic stacking, hydrogen bonds, and also hydrophobic, repulsive, and attractive interactions. In the dataset analysis menu, through the link generated when a project was previously submitted, users have access to various information about protein–ligand interactions. It is possible to graphically visualize analysis of interactions (Figure S8), perform filtering of ligands, and several other submission details. In addition, the website has a help section, where users can find detailed materials on all aspects of using nAPOLI, as well as descriptions of graphical analysis and interaction summary.

7.3. Protein–peptide interaction methods

7.3.1. GalaxyPepDock

GalaxyPepDock (http://galaxy.seoklab.org/pepdock) [163] is a web server capable of generating high-resolution complex structure models of protein–peptide binding by a similarity-based method. These models are built based on templates from experimentally known protein–peptide interaction structures and GALAXY [168] energy-based optimization that improves structural flexibility on the template-based search.

The method proposed by the authors consists firstly in selecting templates from the PepBind [169] database, according to similarity measures of protein structures and their interactions. For the construction of the model, for each template, 50 models of complex structures are generated using the GalaxyTBM [170,171] method, using both protein structure and peptide sequence alignment. For model optimization, distance controls between interactions are inserted in GALAXY energy, and finally, 10 structures with the best energy values are selected for each template and then refined through GalaxyRefine [168].

The website can present the generated structures using PyMOL and offers a download option. Along with the models, information about the binding sites and their estimated accuracy is also available. The average accuracy of GalaxyPepDock binding site residues identification is 75.4%.

8. Catalytic and binding site prediction

Currently, there is an increase in the number of structures and protein sequences deposited in specialized databases, being provided mainly by the advancement of genomic sequencing technologies and methods to determine their structure. Among the challenges involving proteins, we can highlight the prediction of their function, which increases the need for automatic and reliable approaches that are capable of performing such prediction [10].

Despite efforts to define and annotate the functions of proteins, the Pfam [172], a database that catalogs protein families, contains about 22% (3961) of all entries marked as proteins of unknown function. As another example, the Uniprot database, responsible for cataloging protein sequences, contains over 120 million entries. The amount of proteins that have been analyzed by experts is only less than a million [173].

A large number of biological processes, including cellular defenses, catalysis of enzymes, and signal transmission, depending on the interaction between proteins and small molecules. An enzyme can be defined as a protein molecule that acts as a catalyst in chemical reactions, regulating and synchronizing them. Therefore, it is crucial to develop methods that can be capable of identifying protein binding sites, contributing to studies on protein functions and new functional roles [174].

The function of a protein can be predicted by searching for enzyme binding sites. Enzyme binding sites are areas on the surface of an enzyme designed to interact with other molecules and they can be divided into two different parts: the substrate-binding site and the catalytic site. The first one recognizes the molecule on which the enzyme performs and the second one is a collection of two to six amino acids that performs the catalytic role [175].

There are some experimental methods performed in web labs that can identify catalytic and binding sites in proteins, however, they still face difficulties to be executed, due to problems related to cost, time, and automation of processes. These issues create the demand for computational techniques to grow even more, making efforts to create and improve prediction methods. These created techniques have been able to search and predict catalytic and binding sites, as well as their geometry, function, and various other information that can help different aspects of research in the area [176].

In the literature, there are a large number of computational methods that have been proposed to predict the binding and catalytic sites of a protein, providing a reduction in costs and time compared to experimental procedures. These methods can be grouped into three main categories: algorithms based on sequence, structure, or hybrid techniques. Table 7 summarizes catalytic and binding site prediction tools described in this section.

8.1. Methods for catalytic site prediction

8.1.1. GASS

As an example of methods for predicting catalytic sites, [177–179] propose GASS, a genetic algorithm that searches for active sites based on templates, which uses only distance information between residues. The search problem treated by GASS can be described as given a collection of N amino acids that make up the active site A of an enzyme of known function (template), and a protein B with M amino acids of unknown function, the method looks for the pattern A in B.

GASS defines an individual, used in his genetic algorithm, as a group of amino acids that make up candidate solutions to the problem. An individual is defined as a vector where each index contains information about a single amino acid. The method involves a population of candidate active sites in which genetic operators such as crossover and mutation act. The fitness function used by GASS is based on the Root Mean Square Deviation (RMSD) and calculates the distance between a template and an active candidate site. The web server can be found at https://gass.unifei.edu.br/. Figure S9 shows the GASS result page, with found active sites and matched templates.

The authors compared GASS with 17 other methods submitted in CASP 10 [193], for protein function prediction, where it was ranked fourth compared to the other methods, according to Matthew Correlation Coefficient (MCC) values. Other results also showed GASS effectiveness in identifying catalytic sites cataloged by the CSA with accuracy greater than 90% in several cases.

8.1.2. PINGU

An SVM-based method was proposed by [180], the algorithm named PINGU (PredIction of eNzyme catalytic residues using seqUence information), classifies sets of catalytic and non-catalytic residues, then filters these results through a post-processing method, leaving just the most probable catalytic ones. The train and the independent test data sets were selected from the CSA and the PDB, 650 enzymes for the training and 200 enzymes for the test dataset. The SVM model was chosen over Logistic Regression and Radial Basis Function Network as a better performance was presented using this type of data and inputs.

The PINGU algorithm outperformed other similar methods by using specific physicochemical residue attributes, together with an evolutionary conservation index. During the SVM step, the algorithm selects as

Table 7
Summary of the catalytic and binding site prediction tools.

Name	URL	Features	Limitations	Reference
GASS	https://gass.unifei.edu.br/	Free of charge Webserver Based on structural information of proteins Genetic Algorithm-based method	Dependent on precisely oriented residues	[177–179]
PINGU	–	Free of charge Webserver Based on sequence information of proteins SVM-based method	Predicts specific residue functions Data processing takes long time	[180]
iCataly-PseAAC	http://www.jci-bioinfo.cn/iCataly-PseAAC	Free of charge Webserver Based on sequence information of proteins Fuzzy KNN-based method	–	[181]
Chien and Huang method	–	Based on sequence and structural information of proteins SVM-based method	Method is not available via webserver or standalone program	[182]
GRaSP	https://grasp.ufv.br/	Free of charge Webserver Based on structural information of proteins Graph-based method	Binary results may be uninformative	[174]
FunFOLD	http://www.reading.ac.uk/bioinf/FunFOLD	Free of charge Webserver or standalone program Based on structural information of proteins Uses TM-Align method	Limited quality assessment features	[183]
DeepSite	https://www.playmolecule.com/deepsite/	Free of charge Webserver Based on structural information of proteins CNN-based method	No customize protein options Undocumented web API	[184]
TRAPP	https://trapp.h-its.org/	Free of charge Webserver Based on structural information of proteins	Identifies only specific pockets Slow running (Molecular Dynamics approach)	[185,186]
CAVER	https://loschmidt.chemi.muni.cz/caverweb/	Free of charge Webserver Based on sequence and structural information of proteins Uses TM-Align method	Tunnel length is sometimes shortened Works only with static structures	[187]
MED-SuMo	http://www.medit.fr/	Commercial software Standlaone program Based on structural information of proteins	GUI available only for Windows	[188]
eFindSite	http://www.brylinski.org/efindsite	Free of charge Standalone program Based on structural information of proteins Perform ligand-based virtual screening against identified pockets	Unbalanced base of templates	[189]
POVME	https://github.com/POVME/POVME	Free of charge Standalone program Based on structural information of proteins Integrate results into large data workflows	Limited when working with poorly understood proteins	[190]
SiteEngine	http://bioinfo3d.cs.tau.ac.il/SiteEngine/SiteEngine.html	Free of charge Standalone program Based on structural information of proteins Also uses physicochemical properties Can handle large protein structures quickly	Limited quality of biological predictions No implicit treatment of electrostatic potentials	[191]
SVILP_ligand	http://www.sbg.bio.ic.ac.uk/svilp_ligand/	Free of charge Standalone program Based on structural information of proteins SVM-based method	Method not tested on unbound structures Poor information available on the web address	[192]

many catalytic residues as possible, even at the cost of a high false-positive rate. This result is then filtered by the post-processing through S-SITE, a ligand-binding site predictor that works based on template recognition. This step allows for minimizing the false positives and leaves almost all of the catalytic residues. A boost of 16% in precision and 0.138 in MCC is marked after filtering.

8.1.3. iCataly-PseAAC

When using residue's physicochemical information, [181] proposes an interesting method called iCataly-PseAAC. The classification is made upon a preprocessed 21 × 20 pseudo amino acid composition matrix, where each line represents the amino acid position in a peptide, and each column represents the amino acid type. The gray-PSSM encoding is used in this case, to better represent the evolutionary conservation score data of each peptide and each residue within it. But the encoding itself does not have classification powers, thus the authors apply a Fuzzy K-NN to the prepared dataset to perform a nonparametric classification.

To assess the quality of the method, the authors compared the iCataly-PseAAC with two other predictors, CRpred [194] and EXIA [195], where the first is based on protein sequence information, and the second on protein structure and sequence conservation. The results were measured according to the accuracy in prediction, and the iCataly-PseAAC scores were greater than 87% in all three benchmark datasets, making the proposed method superior to the other two also evaluated.

The tool is available at the following web address: http://www.jci-bioinfo.cn/iCataly-PseAAC. On this page, the user can enter the query protein sequence in FASTA format or upload a batch prediction file. Authors report that predictions generally take about 20 min to run for each protein, and thus it is possible to receive prediction results via the user's email. Furthermore, users can find additional information about iCataly-PseAAC on the webpage, such as support data and a read me section.

8.2. Methods for binding site prediction

8.2.1. GRaSP

GRaSP [174] is a computational strategy that represents the residue environment (encoded by 2 shells of neighboring residues) through graphs to predict protein–ligand binding sites. The method uses machine learning to model, at the atomic level, the residue environment in the form of graphs. For each residue in a protein, some atom features used include topological and physicochemical properties. The interaction between them is represented as a graph, which is encoded as a vector. GRaSP is available for free at https://grasp.ufv.br/. In Figure S10, GRaSP presents the suggested ligands of the binding site, along with a 3D viewer.

The problem of binding site prediction was defined as follows: a characteristic vector is created for every single residue of a protein, based on 14 descriptors. These descriptors can be grouped into several levels, such as residue, atom, and interaction. Then, GRaSP builds a matrix that represents the entire set of proteins. In the experimental evaluation, six different datasets were used, as well as tests were performed comparing GRaSP to several state-of-the-art methods.

To evaluate the method, the authors used six different datasets, and GRaSP presented compatible or superior results. To illustrate, it ought to mention that GRaSP outperformed the other six methods that predict binding site residues. Also, the method ranked seventh among 17 CASP 10 methods (there was no statistical difference between the first 10 methods) and outperformed RaptorX-Binding, which is a method from CAMEO independent assessment similar to GRaSP. The method ranked second when compared to methods that predict pockets (that can potentially be binding sites) and it takes 10–20 s to calculate a binding site while the state-of-the-art method takes 2–5 h.

8.2.2. FunFOLD

The FunFOLD algorithm, proposed by [183], works with an automatic approach for the selection of residues and cluster identification. The method calculates binding sites, their residues, and other information about a target protein and its ligands. FunFOLD is based on the concept that templates that contain ligands, coming from the PDB, with the same fold as 3D models of the target protein, probably contain similar binding sites. FunFOLD uses the TM-Align method to superpose each of the structure templates that contain relevant ligands onto the 3d protein model.

The developed tool can be combined with existing fold recognition servers, using as input a list of templates and a 3D model of proteins. The authors also proposed the FunFOLD2 server [196], which can perform prediction of proteins from their sequences via structure. It uses connection sites prediction from FunFOLD, and quality assessment protocols from FunFOLDQA [197]. FunFOLD2 is available for use at http://www.reading.ac.uk/bioinf/FunFOLD, where users can download both FunFOLD and FunFOLDQA.

8.2.3. DeepSite

[184] propose a method called DeepSite for binding site prediction based on convolutional neural networks. The approach works by mapping protein structures from a computer vision perspective, in 3D images discretized into voxels. These voxels consist of a group of pharmacophoric properties at the atomic level and their occupations are defined according to the atoms of the protein, taking into account their excluded volume and some atomic properties, such as aromatic, hydrophobic, and metallic aspects. The tool is available at https://www.playmolecule.com/deepsite/, where the user can perform the prediction through a protein structure in PDB format and view the results through the WebGL viewer.

DeepSite makes use of a Deep Convolutional Neural Network (DCNN) composed of four convolutional layers, and every two of these layers, max-pooling, and dropout are used. At the end of the DCNN, there is a fully connected layer. All layers, except the last one, use the Exponential Linear Unit (ELU) activation function and the network output uses the sigmoidal activation function. For training and evaluation of the DCNN, a dataset formed by 7622 proteins from the scPDB database was used, where filters involving similarities of binding sites were applied to the structures.

To evaluate the proposed method, the authors used two different criteria. The first one evaluates the performance of predictions based on two metrics: distance to the center of the binding site (DCC) and discretized volumetric overlap (DVO). Both metrics are also used in two other prediction methods, fPocket and Concavity, as shown in [198] work, to compare them with DeepSite. In addition, the second criterion evaluates the method on different structural groups of proteins. Tests were performed using the SCOPe [199] dataset, which provides curated information on PDB structures.

8.2.4. TRAPP

The TRAnsient Pockets in Proteins (TRAPP) [185,186] is a tool for detecting pockets and sub-pockets that may have been generated by internal motion in proteins. Moreover, this tool provides resources for exploring the dynamics of protein binding sites. TRAPP is not a method focused on identifying all binding pockets in proteins but works to analyze physicochemical properties and spatial changes in specific pockets, which may have been generated from protein motion.

Three different modules make up the webserver: TRAPP Structure, TRAPP Analysis, and TRAPP Pocket. After defining parameters and user inputs, the three modules are executed in sequence. In the first module, TRAPP Structure, protein structures are generated, representing characteristics of the binding pocket, such as the diversity conformation. The second module, TRAPP Analysis, comes next and is responsible for providing tools for comparing binding pockets in protein structures. Finally, the TRAPP Pocket module locates pockets and identifies transient regions in protein structures.

TRAPP is available free of charge at https://trapp.h-its.org/ and can be run via a web application. Users can also download a desktop version of the tool, which comprises a command-line version of TRAPP but only for Linux systems. Although it is necessary to obtain a license to use TRAPP, there is currently no charge to acquire it, simply fill out a form with user information. Several tools for exploring binding sites can also be found on the webserver, as well as resources for analyzing the dynamics of binding sites. In addition, other information can be found on the TRAPP website, such as examples of work done by the tool, as well as documentation to help users.

8.2.5. CAVER

[187] proposed Caver Web, an interactive tool for the identification of protein tunnels and channels, in addition to allowing the analysis of ligand transport. The server is built using two different methods. For the detection of tunnels, the Caver 3.02 method [200] is used, a tool that provides high-quality results with fast calculations, and is based on the Voronoi diagram representation of protein structures. Ligand transport analysis is performed using the CaverDock 1.0 tool [201], which is based on molecular docking iteration along the tunnel and performs the analysis quickly and accurately.

Caver Web is available for free through its website https://loschmidt.chemi.muni.cz/caverweb/, which, in addition to the tool, also brings examples of using the method, as well as other tools of the group. The identifications and analyzes carried out by Caver are made entirely via the web, without the need to install software locally by the user. For the method execution, some steps must be followed. First, the user must select the protein structure to work on (PDB format) and also select the biological unit. In the second step, the starting point selection for tunnel detection is defined, according to several different modes, such as a pocket, catalytic pocket, ligands, sequence, and manual tweaking. Finally, in the third step, Caver settings and parameters are selected, such as shell radius and depth, clustering threshold, minimum probe radius, and also maximal distance.

After following all the steps described above, Caver Web is executed, calculating the protein tunnels. A results page is shown containing various analyses, including information about the executed job and tunnels. About the tunnels, several characteristics and details are described, such as bottleneck radius, length and curvature of the tunnel, list of all centerline spheres, and also the residues and atoms around the tunnel. In addition, the results page allows analysis of ligands transportation through the tunnel.

8.2.6. POVME

Proposed by [190], POVME (POcket Volume MEasurer) is a tool for ligand-binding pocket analysis in proteins. The method works with a grid-based representation of the binding pockets through voxels and performs the analysis through descriptions of the flexibility and shape of the cavity. Its latest version, POVME 3.0 [202], adds several features, such as methods for clustering and principal component analysis, identification of features through chemical coloring scheme, as well as new features for analyzing and comparing binding pockets.

Four steps are required to run POVME. First, the user must select the region of the binding pocket (using spheres and prisms). In the second step, an encompassing region is defined and a volume-grid file with equispaced points is also generated. Then, volume-grid points close to the protein atoms are deleted, leaving only the points that are in the binding pocket. Finally, in the last step, the volume of the binding pocket is calculated, according to the number of the remaining points. POVME is an open-source tool, developed in Python language and is only available in a standalone version, available for installation at https://github.com/POVME/POVME.

9. Databases

Data related to bioinformatics have been generated at a fast pace by researchers from all over the world. This increase in data has made it essential to use and interconnect (cross-references) databases for storage and analysis. Several types of databases differ in their structure (e.g. flat-file format and relational form), as well as in the management structures (e.g. object-oriented databases, data warehouses, and distributed databases). In addition, it is necessary to use a DBMS (database management system) to control the database [203].

Due to a large number of databases in bioinformatics, there is a lot of redundant and replicated data, making it difficult to search for information. The journal Nucleic Acids Research (NAR) can help with this search task. Every year, NAR produces a special issue reporting new and updated databases. In addition, there is a list of the URLs of databases that have been reported in these annual issues of NAR, called the Molecular Biology Database Collection (available from the NAR home page), where can be found data from gene expression and its regulation, genome structure, protein domains, and protein–protein interactions. Also, there are database centers, such as EBI (http://www.ebi.ac.uk) and NCBI (http://www.ncbi.nlm.nih.gov/) that have links to several databases involving different data such as sequence, structure and genome [203,204].

Bioinformatics databases can be classified according to the type of data they contain (e.g. sequence data, experimental data, structure data, and protein interaction data) [203]. However, the databases do not store the data itself. For example, an enzyme database may also contain information about catalytic residues (position in the sequence) and links to other databases with structural information. Next, some characteristics from important databases frequently used in bioinformatics will be presented. Table 8 summarizes the databases described in this section.

9.1. PDB

Established at Brookhaven National Laboratories (BNL) in 1971, The Protein Data Bank (PDB) is a database of biological macromolecular structures. In October 1998, the management of the PDB became the responsibility of the Research Collaboratory for Structural Bioinformatics (RCSB) [11]. In 2003, three organizations have established a collaboration to oversee the newly formed worldwide Protein Data Bank (wwPDB; http://www.wwpdb.org/): the Protein Data Bank Japan (PDBj) at the Institute for Protein Research in Osaka University, the Macromolecular Structure Database (MSD) at the European Bioinformatics Institute (EBI) and the Research Collaboratory for Structural Bioinformatics (RCSB). The goal was to maintain wwPDB publicly available to the global community [205,206].

Currently, PDB (https://www.rcsb.org/) has about 180,000 annotated structures, including nucleic acids, proteins, and large macromolecular complexes that have been determined using NMR, X-ray crystallography, and electron microscopy techniques. The PDB also provides several tools to help students and researchers. In addition to the search section, with various filters and features, there are also visualization tools (e.g. Mol* 3D Viewer, Protein Feature View), analysis methods (e.g. Pair Structure Alignment, Protein Symmetry, Statistics), and downloads tools. The PDB-101 section provides a series of information to help students and researchers explore the proteins and nucleic acid structures. There are videos, interactive animations, and a guide to understanding PDB data.

9.2. M-CSA

M-CSA (Mechanism and Catalytic Site Atlas) [213] is a new database formed by merging of databases CSA (Catalytic Site Atlas) [207,208,212] and MACiE (Mechanism, Annotation and Classification in Enzymes) [209–211]. The CSA database contained information about

Table 8
Summary of the database.

Name	URL	Features	Limitations	Reference
PDB	http://www.wwpdb.org/	About 180,000 annotated structures (nucleic acids, proteins, and large macromolecular complexes) using NMR, X-ray crystallography and electron microscopy techniques. It has several tools to help students and researchers (e.g. Mol* 3D Viewer, Protein Feature View, Pair Structure Alignment, Protein Symmetry, Statistics).	Need for pre-processing because in some old structures entries (X-ray crystallography) some atoms may be missing.	[11,205,206]
M-CSA	https://www.ebi.ac.uk/thornton-srv/m-csa/	Database with information about enzyme catalytic residues manually curated and annotation about enzyme mechanisms.	The homologous proteins section can show wrong catalytic residues.	[207,208] [209–211] [212,213]
MetalPDB	https://metalpdb.cerm.unifi.it/	Information on metal-binding sites detected in three-dimensional structures. The last release includes new contents and tools, statistical analyses involving protein families.	Limitations in the structure preview window.	[214]
BioLiP	https://zhanglab.dcmb.med.umich.edu/BioLiP/	Database semi-manually curated of ligand-protein interactions. Focusing on the biological relevance of the residues, contains 529,047 entries and some resources (COACH, Search, Browse and Download).	Some search options do not work. Unstable web server.	[215]
UniProt	https://www.uniprot.org/	The most complete compendium of all known protein sequence data (experimentally verified, or computationally predicted) and functional annotation. The databases are distributed in several formats.	–	[216]
SWISS-MODEL	https://swissmodel.expasy.org	Repository of automatically generated 3D models. Contains more than 400,000 high-quality models.	–	[217]
ProThermDB	https://web.iitm.ac.in/bioinfo2/prothermdb/	It contains protein information, structural information, experimental conditions, literature information and experimental thermodynamic data. method	More information about upload and download options is missing. More details and information are missing from the tutorial.	[218]
MEROPS	http://www.ebi.ac.uk/merops/	Manually curated peptidases information database integrated with their substrates and inhibitors. Includes a manually curated bibliography for an extensive number of entries (peptidases, clan, family, inhibitor and substrate).	–	[219]
SCOP2	https://scop2.mrc-lmb.cam.ac.uk/	It is an evolution of database SCOP, with the purpose of organizing and categorizing proteins according to their structural and evolutionary relationships.	–	[220,221]
LigBase	https://modbase.compbio.ucsf.edu/ligbase/	Database of ligand-binding sites of known structures aligned with related protein sequences.	The last update of the base was in 2002 when it contained approximately 50,000 ligand binding sites for small molecules found in PDB.	[222]

enzyme catalytic residues manually curated. Each entry in CSA was formed by a reference PDB entry, a list of catalytic residues with their chemical functions, the literature evidence, and the overall reaction. For each entry, there was also a list of homologous PDB structures. The database MACiE contained annotations about enzyme mechanisms. Its annotations also included PDB links, UniProtKB, CATH, and other databases. In addition, there were the roles of catalytic residues and cofactors.

The new M-CSA contains 538 entries (manually curated) where catalytic residues have been identified, but the complete mechanism is unknown, and 423 entries (manually curated) with detailed reaction mechanisms. The annotation was extended to 51,993 homologous PDB structures and over 5 million homologous sequences using the UniProtKB reference dataset.

The M-CSA home page (https://www.ebi.ac.uk/thornton-srv/m-csa/) contains a brief qualitative description of the main statistics, citation information, and additional navigation links. The browse page perhaps is the most catching resource once it gives the user a complete overall view of the database. This page shows a table with all the entries that can be sorted by any of the PDB, UniProtKB, CATH, and EC identifiers. Residues and cofactors also can be used to refine the search. When selecting a specific entry, the user can, among other information, access homologous proteins and their respective catalytic residues, as well as visualize the structure using Litemol [223]. Figure S11 shows part of the results of a search with the homologous of the 3NOS protein.

9.3. MetalPDB

Updated in 2018, MetalPDB [214] is a database providing information on metal-binding sites detected in three-dimensional structures. The current release includes new contents and tools, statistical analyses involving protein families, as well as 287,122 sites from 50,797 structures (there was a growth of 64% in the last 6 years).

In MetalPDB, the metal sites are stored as Minimal Functional Sites (MFSs), where each MFS is a set of atoms of the metal cofactor, the metal ligands, and any other residue within 5 Å from a ligand. In general, each MFS has at least one associated function among structural, electron transfer, catalytic, substrate, protection, regulatory, and transport. Beyond providing information about the functions or mechanisms of action of a metalloprotein, MFSs can be useful for predicting the role of 3D structures in the absence of experimental biochemical data.

In addition to offering several improvements to the web interface, the new version of MetalPDB (https://metalpdb.cerm.unifi.it/) includes some tools like MetalS 3, which is designed to search similar sites with a query site. Figure S12 shows part of the results of a search done in MetalPDB.

9.4. BioLip

The BioLiP [215] is a database that lists ligand-protein interactions, just as the PDB. BioLip site residues, however, often differ from the ones on PDB, since it is semi-manually curated and focuses on the biological relevance of the residues. The majority of ligand-binding site prediction methods use templates, generated by the alignment of the same function proteins. This method is very effective on the statistical and computational side, but not all ligands are biologically relevant.

The BioLiP database (https://zhanglab.dcmb.med.umich.edu/BioLiP/) contains the following basic resources: COACH, Search, Browse and Download. BioLiP contains 529,047 entries, these being: 109,998 PDB proteins; 57,059 DNA/RNA ligands; 25,960 peptide ligands; 146,969 metal ligands; 299,051 regular ligands; and 23,492 entries with binding affinity data. Due to its relevance, BioLiP is used and cited by many other methods and articles (e.g., GRASP, MionSITE and I-TASSER) [174,224–226].

9.5. UniProt

The recently updated UniProt [216] databases are a great source for biological, biomedical, and bioinformatics fields research. It aims at providing the most complete compendium of all known protein sequence data (experimentally verified, or computationally predicted) and functional annotation. All the reviewed and curated Swiss-Prot entries are combined by UniProt Knowledgebase (UniProtKB), with the unreviewed TrEMBL entries generated by in silico methods.

All UniProt databases are some of the most accessible, being distributed in numerous formats for download, such as XML, RDF, plain text, GFF, Excel tables, Tab-separated, and FASTA directly from the website (https://www.uniprot.org/). In addition, all data can be retrieved from an FTP interface (ftp://ftp.uniprot.org) and also through a public RESTful API capable of performing complex queries by the SPARQL API endpoint.

This extensible accessibility is proof that the UniProtKB fully supports the FAIR (Findable, Accessible, Interoperable, and Reusable) data principles. This summed to the scale of the databases, approximately 190 million (despite some sequence redundancy), and the fact that it is updated every eight weeks, makes UniProt a database of ultimate importance.

9.6. SWISS-MODEL

SWISS-MODEL Repository (SMR) [217] is an example of a database of automatically generated 3D models. The structural prediction of protein is based on its sequence, using homology modeling. These structure assessments are made using a repository of annotated protein structures and their sequences as templates, with a sequence identity of at least 30% with the input sequence.

The models built by this pipeline are added to the SMR every 15 min, and it tries to avoid redundancy by prioritizing the inclusion of longer sequences and the highest model quality (measured by QMEANS [227]). With almost 20% of SwissProt/UniProtKB entries, SMR contains more than 400,000 high-quality models.

SMR functionalities and the established models are accessible through the web page https://swissmodel.expasy.org/repository featuring a simple but complete search box. Queries to the repository or the pipeline can also be made programmatically through a RESTful API.

9.7. SCOP2

The original purpose of the Structural Classification of Proteins (SCOP) [220] database model was to organize and categorize proteins according to their structural and evolutionary relationships. It has been a reference database for structure annotation, but its tree-based model became unsuccessful in annotating some outlier proteins and some previously unknown relationships. So it was redesigned into the SCOP2 database [221].

SCOP2 (http://scop2.mrc-lmb.cam.ac.uk/) uses an acyclic graph-based categorization that allows different ramifications to the evolutionary and structural relationships. It is also an expert-curated database and uses the PDB as a source of protein structures. There are four main categories: protein relationships (which include structural, evolutionary, and ‘Other’), evolutionary events, protein types, and structural classes. SCOP2 also counts with the classical evolutionary levels from the original SCOP, Species, Protein, Family, and Superfamily.

10. Conclusion

Many resources such as methods, tools, and databases are available to perform main tasks in the context of protein structure bioinformatics. However, they are commonly scattered across different online repositories, making it not straightforward which topics should be learned/used and where these topics could be accessed. This task can be time-consuming, especially for those beginning in the field of bioinformatics.

In this paper, we covered the main subareas of protein structure bioinformatics, presenting, for each subarea, a succinct definition and a set of tools frequently used to address tasks in the subarea. Each tool is described in terms of the main ideas behind how it works or its algorithm, positive aspects, and drawbacks. As a complementary resource to the paper, we developed a website that allows users to retrieve the online resources described here by selecting a keyword from the wordle visualization or submitting a search term of interest. Our tool applies TF–IDF to measure and rank the importance of a search term for each resource covered in this paper, based on their title and abstract. The user can download the results in BibTeX or CSV format.

In future work, we intend to add to the proposed website the datasets, algorithms, tools, and online courses that we have been developing since 2014 in the context of the Structural Bioinformatics of Proteins (Babel) project, which involves 6 universities in the northeast, southeast and south of Brazil and was financed by Brazilian public funding institutions. This material was proposed to create a path in the area of structural bioinformatics of proteins to train our students during the project and also as a result of the research developed. We believe that making this resource available for free would be a significant contribution to society.

CRediT authorship contribution statement

Vinícius de Almeida Paiva: Conceived and planned the study, prepared the “Molecular Dynamics”, “Molecular Visualization”, “Structure prediction”, “Interactions at atomic/residue level” and “Catalytic and Binding Site Prediction” sections. **Isabela de Souza Gomes:** Prepared the “Docking”, “Molecular dynamics” and “Molecular Visualization” sections. **Cleiton Rodrigues Monteiro:** Prepared the “Mutation Analysis” section. **Murillo Ventura Mendonça:** Prepared the

"Interactions at atomic/residue level" and "Databases" sections. **Pedro Magalhães Martins:** Developed the PreStO web server. **Charles Abreu Santana:** Prepared the "Structure Prediction" section. **Valdete Gonçalves-Almeida:** Prepared the "Molecular Visualization" and "Mutation Analysis" sections. **Sandro Carvalho Izidoro:** Conceived and planned the study, prepared the "Catalytic and Binding Site Prediction" and "Databases" sections. **Raquel Cardoso de Melo-Minardi:** Conceived and planned the study. **Sabrina de Azevedo Silveira:** Conceived and planned the study and wrote the manuscript.

Acknowledgment

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 23038.004007/2014-82 - Grant 051/2013; Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG). The funding agencies had no role in study design, data collection, analysis and interpretation, decision to publish, or preparation of the manuscript. All authors read and approved the final manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compbiomed.2022.105695. In the supplementary material, further discussions of other tools on each topic are covered, along with some figures.

References

[1] Nature, Bioinformatics (2021) https://www.nature.com/subjects/bioinformatics. [Online; access 10 August 2021].
[2] N.M. Luscombe, D. Greenbaum, M. Gerstein, Methods Inf. Med. 40 (4) (2001) 346–358.
[3] Mariaconcetta Bilotta, Giuseppe Tradigo, Pierangelo Veltri, Encyclopedia of Bioinformatics and Computational Biology, Elsevier, 2019, pp. 110–116.
[4] Frédéric Cazals, Tom Dreyfus, Bioinformatics 33 (7) (2016) 997–1004.
[5] Marnix H. Medema, Nat. Prod. Rep. 38 (2) (2021) 301–306.
[6] Lonnie Welch, Fran Lewitter, Russell Schwartz, Cath Brooksbank, Predrag Radivojac, Bruno Gaeta, Maria Victoria Schneider, PLoS Comput. Biol. 10 (3) (2014) e1003496.
[7] Lonnie Welch, Cath Brooksbank, Russell Schwartz, Sarah L. Morgan, Bruno Gaeta, Alastair M. Kilpatrick, Daniel Mietchen, Benjamin L. Moore, Nicola Mulder, Mark Pauley, et al., PLoS Comput. Biol. 12 (5) (2016) e1004943.
[8] Nicola Mulder, Russell Schwartz, Michelle D. Brazas, Cath Brooksbank, Bruno Gaeta, Sarah L. Morgan, Mark A. Pauley, Anne Rosenwald, Gabriella Rustici, Michael Sierk, et al., PLoS Comput. Biol. 14 (2) (2018) e1005772.
[9] Ozlem Keskin, Attila Gursoy, Buyong Ma, Ruth Nussinov, Chem. Rev. 108 (4) (2008) 1225–1244.
[10] Gunseli Bayram Akcapinar, Osman Ugur Sezerman, Biosci. Rep. 37 (2) (2017).
[11] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T.N. Bhat, Helge Weissig, Ilya N. Shindyalov, Philip E. Bourne, Nucleic Acids Res. 28 (1) (2000) 235–242.
[12] R.B. Altman, J.M. Dugan, Structural Bioinformatics, second ed., Wiley-Blackwell, 2009.
[13] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al., Nature 596 (7873) (2021) 583–589.
[14] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, John Moult, Proteins: Struct. Funct. Bioinform. 89 (12) (2021) 1607–1617.
[15] Mehmet Akdel, Douglas E.V. Pires, Eduard Porta Pardo, Jürgen Jänes, Arthur O. Zalevsky, Bálint Mészáros, Patrick Bryant, Lydia L. Good, Roman A. Laskowski, Gabriele Pozzati, et al., BioRxiv (2021).
[16] Teresa K. Attwood, Sarah Blackford, Michelle D. Brazas, Angela Davies, Maria Victoria Schneider, Brief. Bioinform. 20 (2) (2019) 398–404.
[17] Leyla Garcia, Bérénice Batut, Melissa L. Burke, Mateusz Kuzak, Fotis Psomopoulos, Ricardo Arcila, Teresa K. Attwood, Niall Beard, Denise Carvalho-Silva, Alexandros C. Dimopoulos, et al., PLoS Comput. Biol. 16 (5) (2020) e1007854.
[18] Denis Torre, Patrycja Krawczuk, Kathleen M. Jagodnik, Alexander Lachmann, Zichen Wang, Lily Wang, Maxim V. Kuleshov, Avi Ma'ayan, Sci. Data 5 (1) (2018) 1–10.
[19] Zhong-Ru Xie, Chuan-Kun Liu, Fang-Chih Hsiao, Adam Yao, Ming-Jing Hwang, Nucleic Acids Res. 41 (W1) (2013) W292–W296.
[20] Irina Kufareva, Andrey V. Ilatovskiy, Ruben Abagyan, Nucleic Acids Res. 40 (D1) (2012) D535–D540.
[21] OpenAstexViewer, OpenAstexViewer, 2022, http://www.openastexviewer.net/. [Online; accessed 18-April-2022].
[22] Karen Sparck Jones, J. Doc. (1972).
[23] S.F. Sousa, A.J.M. Ribeiro, J.T.S. Coimbra, R.P.P. Neves, S.A. Martins, N.S.H.N. Moorthy, P.A. Fernandes, M.J. Ramos, Curr. Med. Chem. 20 (18) (2013) 2296–2314.
[24] Gaoqi Weng, Junbo Gao, Zhe Wang, Ercheng Wang, Xueping Hu, Xiaojun Yao, Dongsheng Cao, Tingjun Hou, J. Chem. Theory Comput. 16 (6) (2020) 3959–3969.
[25] Zhe Wang, Huiyong Sun, Xiaojun Yao, Dan Li, Lei Xu, Youyong Li, Sheng Tian, Tingjun Hou, Phys. Chem. Chem. Phys. 18 (2016) 12964–12975.
[26] Gareth Jones, Peter Willett, Robert C Glen, Andrew R Leach, Robin Taylor, J. Mol. Biol. 267 (3) (1997) 727–748.
[27] Oleg Trott, Arthur J. Olson, J. Comput. Chem. (2009) NA.
[28] A. Grosdidier, V. Zoete, O. Michielin, Nucleic Acids Res. 39 (suppl) (2011) W270–W277.
[29] Dima Kozakov, David R. Hall, Bing Xia, Kathryn A. Porter, Dzmitry Padhorny, Christine Yueh, Dmitri Beglov, Sandor Vajda, Nat. Protoc. 12 (2) (2017) 255–278.
[30] Sjoerd J. de Vries, Julien Rey, Christina E.M. Schindler, Martin Zacharias, Pierre Tuffery, Nucleic Acids Res. 45 (W1) (2017) W361–W364.
[31] Yumeng Yan, Huanyu Tao, Jiahua He, Sheng-You Huang, Nat. Protoc. 15 (5) (2020) 1829–1852.
[32] B.G. Pierce, K. Wiehe, H. Hwang, B.-H. Kim, T. Vreven, Z. Weng, Bioinformatics 30 (12) (2014) 1771–1773.
[33] Rodrigo V. Honorato, Panagiotis I. Koukos, Brian Jiménez-García, Andrei Tsaregorodtsev, Marco Verlato, Andrea Giachetti, Antonio Rosato, Alexandre M.J.J. Bonvin, Front. Mol. Biosci. 8 (2021) 708.
[34] G.C.P. van Zundert, J.P.G.L.M. Rodrigues, M. Trellet, C. Schmitz, P.L. Kastritis, E. Karaca, A.S.J. Melquiond, M. van Dijk, S.J. de Vries, A.M.J.J. Bonvin, J. Mol. Biol. 428 (4) (2016) 720–725, Computation Resources for Molecular Biology.
[35] Jelisa Iglesias, Suwipa Saen-oon, Robert Soliva, Victor Guallar, WIREs Comput. Mol. Sci. 8 (5) (2018).
[36] Joel Janin, Kim Henrick, John Moult, Lynn Ten Eyck, Michael J.E. Sternberg, Sandor Vajda, Ilya Vakser, Shoshana J. Wodak, Proteins: Struct. Funct. Genet. 52 (1) (2003) 2–9.
[37] Israel T. Desta, Kathryn A. Porter, Bing Xia, Dima Kozakov, Sandor Vajda, Structure 28 (9) (2020) 1071–1081.e3.
[38] Saurav Goel, Xichun Luo, Anupam Agrawal, Robert L. Reuben, Int. J. Mach. Tools Manuf. 88 (2015) 131–164.
[39] David A. Case, Thomas E. Cheatham III, Tom Darden, Holger Gohlke, Ray Luo, Kenneth M. Merz Jr., Alexey Onufriev, Carlos Simmerling, Bing Wang, Robert J. Woods, J. Comput. Chem. 26 (16) (2005) 1668–1688.
[40] Dimitrios Vlachakis, Elena Bencurova, Nikitas Papangelopoulos, Sophia Kossida, Advances in Protein Chemistry and Structural Biology, Elsevier, 2014, pp. 269–313.
[41] Sereina Riniker, J. Chem. Inf. Model. 58 (3) (2018) 565–578.
[42] Pnina Dauber-Osguthorpe, A. Hagler, J. Comput. Aided Mol. Des. 33 (2018) 133–203.
[43] Olgun Guvench, Alexander D. MacKerell, Methods in Molecular Biology, Humana Press, 2008, pp. 63–88.
[44] Luca Monticelli, D. Peter Tieleman, Methods in Molecular Biology, Humana Press, 2012, pp. 197–213.
[45] Juekuan Yang, Yujuan Wang, Yunfei Chen, J. Comput. Phys. 221 (2) (2007) 799–804.
[46] Nikolay Kondratyuk, Vsevolod Nikolskiy, Daniil Pavlov, Vladimir Stegailov, Int. J. High Perform. Comput. Appl. 35 (4) (2021) 312–324.
[47] David A. Case, Thomas E. Cheatham, Tom Darden, Holger Gohlke, Ray Luo, Kenneth M. Merz, Alexey Onufriev, Carlos Simmerling, Bing Wang, Robert J. Woods, J. Comput. Chem. 26 (16) (2005) 1668–1688.
[48] Romelia Salomon-Ferrer, David A. Case, Ross C. Walker, Wiley Interdiscip. Rev.: Comput. Mol. Sci. 3 (2) (2013) 198–210.
[49] Steve Plimpton, J. Comput. Phys. 117 (1) (1995) 1–19.
[50] David Van Der Spoel, Erik Lindahl, Berk Hess, Gerrit Groenhof, Alan E. Mark, Herman J.C. Berendsen, J. Comput. Chem. 26 (16) (2005) 1701–1718.
[51] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, Erik Lindahl, SoftwareX 1 (2015) 19–25.

[52] Bernard R. Brooks, Charles L. Brooks III, Alexander D. Mackerell Jr., Lennart Nilsson, Robert J. Petrella, Benoît Roux, Youngdo Won, Georgios Archontis, Christian Bartels, Stefan Boresch, et al., J. Comput. Chem. 30 (10) (2009) 1545–1614.
[53] James C. Phillips, David J. Hardy, Julio D.C. Maia, John E. Stone, João V. Ribeiro, Rafael C. Bernardi, Ronak Buch, Giacomo Fiorin, Jérôme Hénin, Wei Jiang, et al., J. Chem. Phys. 153 (4) (2020) 044130.
[54] Kevin J. Bowers, David E. Chow, Huafeng Xu, Ron O. Dror, Michael P. Eastwood, Brent A. Gregersen, John L. Klepeis, Istvan Kolossvary, Mark A. Moraes, Federico D. Sacerdoti, John K. Salmon, Yibing Shan, David E. Shaw, SC '06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, 2006, p. 43.
[55] Patrick Louden, Hemanta Bhattarai, Suzanne Neidhart, Teng Lin, Charles F. Vardeman II, Christopher J. Fennell, Matthew A. Meineke, Shenyu Kuang, Madan Lamichhane, Joseph Michalka, et al., 2017.
[56] Piero Procacci, Hybrid MPI/OpenMP Implementation of the ORAC Molecular Dynamics Program for Generalized Ensemble and Fast Switching Alchemical Simulations, ACS Publications, 2016.
[57] Simone Marsili, Giorgio Federico Signorini, Riccardo Chelli, Massimo Marchi, Piero Procacci, J. Comput. Chem. 31 (5) (2010) 1106–1116.
[58] Piero Procacci, Tom A. Darden, Emanuele Paci, Massimo Marchi, J. Comput. Chem. 18 (15) (1997) 1848–1862.
[59] AMMP Program, Another Molecular Mechanics Program VE, 2012, https://www.ddl.unimi.it/cms/index.php?Software_projects:AMMP_VE. [Online; accessed 9-December-2021].
[60] Matt J. Harvey, Giovanni Giupponi, G. De Fabritiis, J. Chem. Theory Comput. 5 (6) (2009) 1632–1639.
[61] Joshua A. Rackers, Zhi Wang, Chao Lu, Marie L. Laury, Louis Lagardère, Michael J. Schnieders, Jean-Philip Piquemal, Pengyu Ren, Jay W. Ponder, J. Chem. Theory Comput. 14 (10) (2018) 5273–5289.
[62] J. Florián, A. Warshel, ChemSol, Version 2.1, University of Southern California: Los Angeles, 1999.
[63] Abalone Software, Abalone molecular simulations, 2021, http://www.biomolecular-modeling.com/Abalone/. [Online; accessed 9-December-2021].
[64] Elmar Krieger, Gert Vriend, Bioinformatics 30 (20) (2014) 2981–2982.
[65] Elmar Krieger, Gert Vriend, J. Comput. Chem. 36 (13) (2015) 996–1007.
[66] Francesca Nerattini, Riccardo Chelli, Piero Procacci, Phys. Chem. Chem. Phys. 18 (22) (2016) 15005–15018.
[67] Piero Procacci, Chiara Cardelli, J. Chem. Theory Comput. 10 (7) (2014) 2813–2823.
[68] Piero Procacci, J. Chem. Inf. Model. 57 (6) (2017) 1240–1245.
[69] Zhi Wang, Tinker9: Next Generation of Tinker with GPU Support, Washington University in St. Louis, 2021, accessed: Dec. 6, 2021.
[70] wwPDB consortium, Nucleic Acids Res. 47 (D1) (2018) D520–D528.
[71] Anne M. Brown, David R. Bevan, Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact, 2017, pp. 1–6.
[72] Schrödinger, LLC, The PyMOL molecular graphics system, version 1.8, in: PyMOL The PyMOL Molecular Graphics System, Version 1.8, Schrödinger, LLC, 2015.
[73] William Humphrey, Andrew Dalke, Klaus Schulten, J. Mol. Graph. 14 (1996) 33–38.
[74] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, T.E. Ferrin, J. Comput. Chem. 25 (13) (2004) 1605–1612.
[75] Alexander S. Rose, Anthony R. Bradley, Yana Valasatava, Jose M. Duarte, Andreas Prlić, Peter W. Rose, Bioinformatics 34 (21) (2018) 3755–3758.
[76] Nicholas Rego, David Koes, Bioinformatics (Oxford, England) 31 (2014).
[77] C.R. Reynolds, S.A. Islam, M.J.E. Sternberg, J. Mol. Biol. 430 (15) (2018) 2244–2248, cited By 48.
[78] Lingle Wang, Yujie Wu, Yuqing Deng, Byungchan Kim, Levi Pierce, Goran Krilov, Dmitry Lupyan, Shaughnessy Robinson, Markus K. Dahlgren, Jeremy Greenwood, Donna L. Romero, Craig Masse, Jennifer L. Knight, Thomas Steinbrecher, Thijs Beuming, Wolfgang Damm, Ed Harder, Woody Sherman, Mark Brewer, Ron Wester, Mark Murcko, Leah Frye, Ramy Farid, Teng Lin, David L. Mobley, William L. Jorgensen, Bruce J. Berne, Richard A. Friesner, Robert Abel, J. Am. Chem. Soc. 137 (7) (2015) 2695–2703.
[79] Molecular Operating Environment (moe), West, Suite 7 (2019) 2021.
[80] Angel Herráez, Biochem. Mol. Biol. Educ. 34 (4) (2006) 255–261.
[81] Bob Hanson, et al., 2008, URL: www.jmol.sourceforgenet.net.–2008.
[82] T. Nakane, 2014, Javascript.webglmol.sourceforge.jp.
[83] Discovery Studio Visualizer, Dassault Systèmes BIOVIA, San Diego, nd, 2020.
[84] John Rakovan, Rocks Miner. 93 (1) (2018) 60–64.
[85] R. Ishitani, T. Nakane, CueMol: molecular visualization framework, 2014.
[86] Elmar Krieger, Gert Vriend, Bioinformatics 30 (20) (2014) 2981–2982.
[87] Chandrajit Bajaj, P. Djeu, Vinay Siddavanahalli, A. Thane, IEEE Visualization 2004, 2004, pp. 243–250.
[88] Shuguang Yuan, H.C. Stephen Chan, Zhenquan Hu, WIREs Comput. Mol. Sci. 7 (2) (2017) e1298.
[89] L. Pan, S.G. Aller, Curr. Protoc. Mol. Biol. 2015 (2015) 19.12.1–19.12.47.
[90] James C. Phillips, David J. Hardy, Julio D.C. Maia, John E. Stone, João V. Ribeiro, Rafael C. Bernardi, Ronak Buch, Giacomo Fiorin, Jérôme Hénin, Wei Jiang, Ryan McGreevy, Marcelo C.R. Melo, Brian K. Radak, Robert D. Skeel, Abhishek Singharoy, Yi Wang, Benoît Roux, Aleksei Aksimentiev, Zaida Luthey-Schulten, Laxmikant V. Kalé, Klaus Schulten, Christophe Chipot, Emad Tajkhorshid, J. Chem. Phys. 153 (4) (2020) 044130.
[91] Nathan A. Baker, David Sept, Simpson Joseph, Michael J. Holst, J. Andrew McCammon, Proc. Natl. Acad. Sci. 98 (18) (2001) 10037–10041.
[92] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Elaine C. Meng, Gregory S. Couch, Tristan I. Croll, John H. Morris, Thomas E. Ferrin, Prot. Sci. 30 (1) (2021) 70–82.
[93] Peter W. Rose, Andreas Prlić, Ali Altunkaya, Chunxiao Bi, Anthony R. Bradley, Cole H. Christie, Luigi Di Costanzo, Jose M. Duarte, Shuchismita Dutta, Zukang Feng, Rachel Kramer Green, David S. Goodsell, Brian Hudson, Tara Kalro, Robert Lowe, Ezra Peisach, Christopher Randle, Alexander S. Rose, Chenghua Shao, Yi-Ping Tao, Yana Valasatava, Maria Voigt, John D. Westbrook, Jesse Woo, Huangwang Yang, Jasmine Y. Young, Christine Zardecki, Helen M. Berman, Stephen K. Burley, Nucleic Acids Res. 45 (D1) (2016) D271–D281.
[94] Alexander S. Rose, Peter W. Hildebrand, Nucleic Acids Res. 43 (W1) (2015) W576–W579.
[95] Alex Bateman, Maria-Jesus Martin, Sandra Orchard, Michele Magrane, Rahat Agivetova, Shadab Ahmad, Emanuele Alpi, Emily H. Bowler-Barnett, Ramona Britto, Borisas Bursteinas, et al., Nucleic Acids Res. (2020).
[96] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N. Bhat, Helge Weissig, Ilya N. Shindyalov, Philip E. Bourne, Nucleic Acids Res. 28 (1) (2000) 235–242.
[97] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, Anna Tramontano, Proteins: Struct. Funct. Bioinform. 86 (2018) 7–15.
[98] Christian B. Anfinsen, Biochem. J. 128 (4) (1972) 737.
[99] Brian Kuhlman, Philip Bradley, Nat. Rev. Mol. Cell Biol. 20 (11) (2019) 681–697.
[100] Konstantin Arnold, Lorenza Bordoli, Jürgen Kopp, Torsten Schwede, Bioinformatics 22 (2) (2006) 195–201.
[101] Andrew Waterhouse, Martino Bertoni, Stefan Bienert, Gabriel Studer, Gerardo Tauriello, Rafal Gumienny, Florian T. Heer, Tjaart A.P. de Beer, Christine Rempfer, Lorenza Bordoli, et al., Nucleic Acids Res. 46 (W1) (2018) W296–W303.
[102] Benjamin Webb, Andrej Sali, Curr. Protoc. Bioinform. 54 (1) (2016) 5–6.
[103] Benjamin Webb, Andrej Sali, Functional Genomics, Springer, 2017, pp. 39–54.
[104] Yang Zhang, BMC Bioinformatics 9 (1) (2008) 1–8.
[105] Jianyi Yang, Renxiang Yan, Ambrish Roy, Dong Xu, Jonathan Poisson, Yang Zhang, Nature Methods 12 (1) (2015) 7–8.
[106] Mohammed AlQuraishi, Bioinformatics 35 (22) (2019) 4862–4865.
[107] Sitao Wu, Yang Zhang, Nucleic Acids Res. 35 (10) (2007) 3375–3382.
[108] Wei Zheng, Chengxin Zhang, Qiqige Wuyun, Robin Pearce, Yang Li, Yang Zhang, Nucleic Acids Res. 47 (W1) (2019) W429–W436.
[109] Morten Källberg, Haipeng Wang, Sheng Wang, Jian Peng, Zhiyong Wang, Hui Lu, Jinbo Xu, Nat. Protoc. 7 (8) (2012) 1511–1522.
[110] Kizhake V. Soman, Catherine H. Schein, Hongyao Zhu, Werner Braun, Nuclease Methods and Protocols, Springer, 2001, pp. 263–286.
[111] Michel Sanner, Armin Widmer, Hans Senn, Werner Braun, J. Comput. Aided Mol. Des. 3 (3) (1989) 195–210.
[112] Thomas Schaumann, Werner Braun, Kurt Wüthrich, Biopolym.: Orig. Res. Biomol. 29 (4–5) (1990) 679–694.
[113] Andrzej Koliński, et al., Acta Biochim. Pol. 51 (2004).
[114] Julia Koehler Leman, Brian D. Weitzner, Steven M. Lewis, Jared Adolf-Bryfogle, Nawsad Alam, Rebecca F. Alford, Melanie Aprahamian, David Baker, Kyle A. Barlow, Patrick Barth, et al., Nature Methods 17 (7) (2020) 665–680.
[115] Dong Xu, Yang Zhang, Proteins: Struct. Funct. Bioinform. 80 (7) (2012) 1715–1735.
[116] Narayanan Eswar, Bino John, Nebojsa Mirkovic, Andras Fiser, Valentin A. Ilyin, Ursula Pieper, Ashley C. Stuart, Marc A. Marti-Renom, Mallur S. Madhusudhan, Bozidar Yerkovich, et al., Nucleic Acids Res. 31 (13) (2003) 3375–3380.
[117] Schrödinger Release. Schrödinger, LLC, New York, NY, 2019.
[118] ACD/ChemSketch, Advanced Chemistry Development, Inc., Build 29305 (2008).
[119] Marcus D. Hanwell, Donald E. Curtis, David C. Lonie, Tim Vandermeersch, Eva Zurek, Geoffrey R. Hutchison, J. Cheminform. 4 (1) (2012).
[120] J.J.P. Stewart, Fujitsu Limited: United States, 2009.
[121] PerkinElmer, ChemBioDraw, 2012.
[122] Lorenza Bordoli, Florian Kiefer, Konstantin Arnold, Pascal Benkert, James Battey, Torsten Schwede, Nat. Protoc. 4 (1) (2009) 1.
[123] Marco Biasini, Stefan Bienert, Andrew Waterhouse, Konstantin Arnold, Gabriel Studer, Tobias Schmidt, Florian Kiefer, Tiziano Gallo Cassarino, Martino Bertoni, Lorenza Bordoli, et al., Nucleic Acids Res. 42 (W1) (2014) W252–W258.
[124] Chengxin Zhang, S.M. Mortuza, Baoji He, Yanting Wang, Yang Zhang, Proteins: Struct. Funct. Bioinform. 86 (2018) 136–151.

[125] Gershon Celniker, Guy Nimrod, Haim Ashkenazy, Fabian Glaser, Eric Martz, Itay Mayrose, Tal Pupko, Nir Ben-Tal, Isr. J. Chem. 53 (3–4) (2013) 199–206.

[126] Thomas A. Hopf, John B. Ingraham, Frank J. Poelwijk, Charlotta P.I. Schärfe, Michael Springer, Chris Sander, Debora S. Marks, Nature Biotechnol. 35 (2017) W128–135.

[127] Carlos H.M. Rodrigues, Douglas E.V. Pires, David B. Ascher, Prot. Sci. 30 (1) (2021) 60–69.

[128] Carles Ferrer-Costa, Josep Lluis Gelpí, Leire Zamakola, Ivan Parraga, Xavier de la Cruz, Modesto Orozco, Bioinformatics 21 (14) (2005) 3176–3178.

[129] Abu Z. Dayem Ullah, Jorge Oscanoa, Jun Wang, Ai Nagano, Nicholas R. Lemoine, Claude Chelala, Nucleic Acids Res. 46 (W1) (2018) W109–W113.

[130] Michael J. Meyer, Juan Felipe Beltrán, Siqi Liang, Robert Fragoza, Aaron Rumack, Jin Liang, Xiaomu Wei, Haiyuan Yu, Nature Methods 15 (2018) W107–114.

[131] Laurens Wiel, Coos Baakman, Daan Gilissen, Joris A. Veltman, Gerrit Vriend, Christian Gilissen, Human Mutat. 40 (2019) W1030–1038.

[132] Omar Wagih, Marco Galardini, Bede P. Busby, Danish Memon, Athanasios Typas, Pedro Beltrao, Mol. Syst. Biol. 14 (2018).

[133] Michal Krassowski, Marta Paczkowska, Kim Cullion, Tina Huang, Irakli Dzneladze, B.F. Francis Ouellette, Joseph T. Yamada, Amelie Fradet-Turcotte, Jüri Reimand, Nucleic Acids Res. 46 (D1) (2017) D901–D910.

[134] Abel Gonzalez-Perez, Núria López-Bigas, Am. J. Human Genet. 88 4 (2011) 440–449.

[135] Jaroslav Bendl, Jan Stourac, Ondrej Salanda, Antonín Pavelka, Eric D. Wieben, Jaroslav Zendulka, Jan Brezovsky, Jiří Damborský, PLoS Comput. Biol. 10 (2014).

[136] Emidio Capriotti, Piero Fariselli, Rita Casadio, Nucleic Acids Res. 33 (suppl_2) (2005) W306–W310.

[137] Hashem A. Shihab, Julian Gough, David N. Cooper, Peter D. Stenson, Gary Barker, Keith J. Edwards, Ian N.M. Day, Tom R. Gaunt, Human Mutat. 34 (2013) 57–65.

[138] Pauline C. Ng, Steven Henikoff, Nucleic Acids Res. 31 (13) (2003) 3812–3814.

[139] Vasily Ramensky, Peer Bork, Shamil Sunyaev, Nucleic Acids Res. 30 (17) (2002) 3894–3900.

[140] M. Masso, I.I. Vaisman, Protein Eng. Des. Select. 23 (8) (2010) 683–687.

[141] Matthew D. McCoy, John Hamre, Dmitri K. Klimov, M. Saleet Jafri, Biophys. J. 120 (2) (2021) 189–204.

[142] Prateek Kumar, Steven Henikoff, Pauline C. Ng, Nat. Protoc. 4 (7) (2009) 1073–1081.

[143] Albert C. Pan, Daniel Jacobson, Konstantin Yatsenko, Duluxan Sritharan, Thomas M. Weinreich, David E. Shaw, Proc. Natl. Acad. Sci. 116 (10) (2019) 4244–4249.

[144] George R. Bickerton, Alicia P. Higueruelo, Tom L. Blundell, BMC Bioinformatics 12 (1) (2011) 1–15.

[145] Javier De Las Rivas, Celia Fontanillo, PLoS Comput. Biol. 6 (6) (2010) e1000807.

[146] V. Srinivasa Rao, K. Srinivas, G.N. Sujini, G.N. Kumar, Int. J. Proteom. 2014 (2014).

[147] Tjelvar S.G. Olsson, Mark A. Williams, William R. Pitt, John E. Ladbury, J. Mol. Biol. 384 (4) (2008) 1002–1017.

[148] Adrian Schreyer, Tom Blundell, Chem. Biol. Drug Des. 73 (2) (2009) 157–167.

[149] Xing Du, Yi Li, Yuan-Ling Xia, Shi-Meng Ai, Jing Liang, Peng Sang, Xing-Lai Ji, Shu-Qun Liu, Int. J. Mol. Sci. 17 (2) (2016) 144.

[150] Nir London, Barak Raveh, Ora Schueler-Furman, Homology Modeling, Springer, 2011, pp. 375–398.

[151] Nir London, Dana Movshovitz-Attias, Ora Schueler-Furman, Structure 18 (2) (2010) 188–199.

[152] Robyn L. Stanfield, Ian A. Wilson, Curr. Opin. Struct. Biol. 5 (1) (1995) 103–113.

[153] Qiangfeng Cliff Zhang, Donald Petrey, Lei Deng, Li Qiang, Yu Shi, Chan Aye Thu, Brygida Bisikirska, Celine Lefebvre, Domenico Accili, Tony Hunter, et al., Nature 490 (7421) (2012) 556–560.

[154] Felippe C. Queiroz, Adriana M.P. Vargas, Maria G.A. Oliveira, Giovanni V. Comarela, Sabrina A. Silveira, BMC Bioinformatics 21 (1) (2020) 1–25.

[155] Carlos H.M. Rodrigues, Yoochan Myung, Douglas E.V. Pires, David B. Ascher, Nucleic Acids Res. 47 (W1) (2019) W338–W344.

[156] Sebastian Salentin, Sven Schreiber, V. Joachim Haupt, Melissa F. Adasme, Michael Schroeder, Nucleic Acids Res. 43 (W1) (2015) W443–W447.

[157] Andrew C. Wallace, Roman A. Laskowski, Janet M. Thornton, Protein Eng. Des. Select. 8 (2) (1995) 127–134.

[158] Roman A. Laskowski, Mark B. Swindells, LigPlot+: multiple ligand–protein interaction diagrams for drug discovery, 2011.

[159] Alexandre V. Fassio, Lucianna H. Santos, Sabrina A. Silveira, Rafaela S. Ferreira, Raquel C. de Melo-Minardi, IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (4) (2019) 1317–1328.

[160] Ségolène Caboche, J. Cheminform. 5 (1) (2013) 1–7.

[161] Jacob D. Durrant, J. Andrew McCammon, J. Mol. Graph. Model. 29 (6) (2011) 888–893.

[162] Katrin Stierand, Patrick C. Maaß, Matthias Rarey, Bioinformatics 22 (14) (2006) 1710–1716.

[163] Hasup Lee, Lim Heo, Myeong Sup Lee, Chaok Seok, Nucleic Acids Res. 43 (W1) (2015) W431–W435.

[164] Andrew Y. Ng, Michael I. Jordan, Yair Weiss, Advances in Neural Information Processing Systems, 2002, pp. 849–856.

[165] Melissa F. Adasme, Katja L. Linnemann, Sarah Naomi Bolz, Florian Kaiser, Sebastian Salentin, V. Joachim Haupt, Michael Schroeder, Nucl. Acids Res. (2021).

[166] D. Szisz, Chemical hashed fingerprint, 2021, https://docs.chemaxon.com/display/docs/chemical-hashed-fingerprint.md. [Online; access 26 August 2021].

[167] Lawrence A. Kelley, Stephen P. Gardner, Michael J. Sutcliffe, Protein Eng. Des. Select. 9 (11) (1996) 1063–1065.

[168] L. Heo, H. Park, C. Seok, Nucl. Acids Res. 41 (Web Server issue) (2013) W384–388.

[169] Arindam Atanu Das, Om Prakash Sharma, Muthuvel Suresh Kumar, Ramadas Krishna, Premendu P. Mathur, Genom. Proteom. Bioinform. 11 (4) (2013) 241–246.

[170] Junsu Ko, Hahnbeom Park, Lim Heo, Chaok Seok, Nucleic Acids Res. 40 (W1) (2012) W294–W297.

[171] Junsu Ko, Hahnbeom Park, Chaok Seok, BMC Bioinformatics 13 (1) (2012) 1–8.

[172] Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R. Eddy, Aurélien Luciani, Simon C. Potter, Matloob Qureshi, Lorna J. Richardson, Gustavo A. Salazar, Alfredo Smart, et al., Nucleic Acids Res. 47 (D1) (2019) D427–D432.

[173] UniProt Consortium, Nucleic Acids Res. 47 (D1) (2019) D506–D515.

[174] Charles A. Santana, Sabrina de A. Silveira, João P.A. Moraes, Sandro C. Izidoro, Raquel C. de Melo-Minardi, António J.M. Ribeiro, Jonathan D. Tyzack, Neera Borkakoti, Janet M. Thornton, Bioinformatics 36 (Supplement_2) (2020) i726–i734.

[175] Torsten Schwede, Manuel C. Peitsch, Computational Structural Biology: Methods and Applications, World scientific, 2008.

[176] Medhavi Mallick, Ambarish Sharan Vidyarthi, et al., Curr. Bioinform. 6 (4) (2011) 444–449.

[177] Sandro C. Izidoro, Raquel C. de Melo-Minardi, Gisele L. Pappa, Bioinformatics 31 (6) (2015) 864–870.

[178] Sandro Izidoro, Anisio M. Lacerda, Gisele L. Pappa, Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015, pp. 905–910.

[179] Joao P.A. Moraes, Gisele L. Pappa, Douglas E.V. Pires, Sandro C. Izidoro, Nucleic Acids Res. 45 (W1) (2017) W315–W319.

[180] Priyadarshini P. Pai, S.S. Shree Ranjani, Sukanta Mondal, PLoS One 10 (8) (2015) e0135122.

[181] Xiao Xuan, Meng-Juan Hui, Zi Liu, Wangren Qiu, J. Membr. Biol. 248 (2015).

[182] Yu-Tung Chien, Shao-Wei Huang, BioMed Res. Int. 2013 (2013) 802945.

[183] Daniel B. Roche, Stuart J. Tetchner, Liam J. McGuffin, BMC Bioinformatics 12 (1) (2011) 1–20.

[184] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S. Rose, Gianni De Fabritiis, Bioinformatics 33 (19) (2017) 3036–3042.

[185] Daria B. Kokh, Stefan Richter, Stefan Henrich, Paul Czodrowski, Friedrich Rippmann, Rebecca C. Wade, TRAPP: A tool for analysis of tra nsient binding pockets in proteins, 2013.

[186] Antonia Stank, Daria B. Kokh, Max Horn, Elena Sizikova, Rebecca Neil, Joanna Panecka, Stefan Richter, Rebecca C. Wade, Nucleic Acids Res. 45 (W1) (2017) W325–W330.

[187] Jan Stourac, Ondrej Vavra, Piia Kokkonen, Jiri Filipovic, Gaspar Pinto, Jan Brezovsky, Jiri Damborsky, David Bednar, Nucleic Acids Res. 47 (W1) (2019) W414–W422.

[188] Olivia Doppelt-Azeroual, Fabrice Moriaud, A. Stewart Adcock, Francois Delfaud, Infect. Disord. - Drug Targets 9 (3) (2009) 344–357.

[189] Michal Brylinski, Wei P. Feinstein, J. Comput. Aided Mol. Des. 27 (6) (2013) 551–567.

[190] Jacob D. Durrant, César Augusto F. de Oliveira, J. Andrew McCammon, J. Mol. Graph. Model. 29 (5) (2011) 773–776.

[191] Alexandra Shulman-Peleg, Ruth Nussinov, Haim J. Wolfson, J. Mol. Biol. 339 (3) (2004) 607–633.

[192] Lawrence A. Kelley, Paul J. Shrimpton, Stephen H. Muggleton, Michael J.E. Sternberg, Protein Eng. Des. Sel. 22 (9) (2009) 561–567.

[193] Tiziano Gallo Cassarino, Lorenza Bordoli, Torsten Schwede, Proteins: Struct. Funct. Bioinform. 82 (2014) 154–163.

[194] Tuo Zhang, Hua Zhang, Ke Chen, Shiyi Shen, Jishou Ruan, Lukasz Kurgan, Bioinformatics 24 (20) (2008) 2329–2338.

[195] Yu-Tung Chien, Shao-Wei Huang, PLoS One 7 (10) (2012) e47951.

[196] Daniel B. Roche, Maria T. Buenavista, Liam J. McGuffin, Nucleic Acids Res. 41 (W1) (2013) W303–W307.

[197] Daniel B. Roche, Maria T. Buenavista, Liam J. McGuffin, PLoS One 7 (5) (2012) e38219.

[198] Ke Chen, Marcin J. Mizianty, Jianzhao Gao, Lukasz Kurgan, Structure 19 (5) (2011) 613–621.

[199] Naomi K. Fox, Steven E. Brenner, John-Marc Chandonia, Nucleic Acids Res. 42 (D1) (2014) D304–D309.

[200] Eva Chovancova, Antonin Pavelka, Petr Benes, Ondrej Strnad, Jan Brezovsky, Barbora Kozlikova, Artur Gora, Vilem Sustr, Martin Klvana, Petr Medek, Lada Biedermannova, Jiri Sochor, Jiri Damborsky, PLoS Comput. Biol. 8 (10) (2012) e1002708.

[201] Jiří Filipovič, Ondřej Vávra, Jan Plhák, David Bednář, Sérgio M Marques, Jan Brezovskỳ, Luděk Matyska, Jiří Damborskỳ, IEEE/ACM Trans. Comput. Biol. Bioinform. 17 (5) (2019) 1625–1638.

[202] Jeffrey R. Wagner, Jesper Sørensen, Nathan Hensley, Celia Wong, Clare Zhu, Taylor Perison, Rommie E. Amaro, J. Chem. Theory Comput. 13 (9) (2017) 4584–4592.

[203] M.J. Zvelebil, M.J.Z. Jeremy O. Baum, M. J, M. Zvelebil, J.O. Baum, Understanding Bioinformatics, Garland Science, 2008.

[204] Michael Y. Galperin, Xosé M. Fernández-Suárez, Daniel J. Rigden, Nucleic Acids Res. 45 (D1) (2016) D1–D11.

[205] Helen Berman, Kim Henrick, Haruki Nakamura, Nat. Struct. Mol. Biol. 10 (12) (2003) 980.

[206] Helen Berman, Kim Henrick, Haruki Nakamura, John L. Markley, Nucleic Acids Res. 35 (suppl_1) (2006) D301–D303.

[207] Gail J. Bartlett, Craig T. Porter, Neera Borkakoti, Janet M. Thornton, J. Mol. Biol. 324 (1) (2002) 105–121.

[208] Craig T. Porter, Gail J. Bartlett, Janet M. Thornton, Nucleic Acids Res. 32 (suppl_1) (2004) D129–D133.

[209] Gemma L. Holliday, Gail J. Bartlett, Daniel E. Almonacid, Noel M. O'Boyle, Peter Murray-Rust, Janet M. Thornton, John B.O. Mitchell, Bioinformatics 21 (23) (2005) 4315–4316.

[210] Gemma L. Holliday, Daniel E. Almonacid, Gail J. Bartlett, Noel M. O'Boyle, James W. Torrance, Peter Murray-Rust, John B.O. Mitchell, Janet M. Thornton, Nucleic Acids Res. 35 (suppl_1) (2006) D515–D520.

[211] Gemma L. Holliday, Claudia Andreini, Julia D. Fischer, Syed Asad Rahman, Daniel E. Almonacid, Sophie T. Williams, William R. Pearson, Nucleic Acids Res. 40 (Database issue) (2012) D783–D789.

[212] Nicholas Furnham, Gemma L. Holliday, Tjaart A.P. de Beer, Julius O.B. Jacobsen, William R. Pearson, Janet M. Thornton, Nucleic Acids Res. 42 (D1) (2013) D485–D489.

[213] António J.M. Ribeiro, Gemma L. Holliday, Nicholas Furnham, Jonathan D. Tyzack, Katherine Ferris, Janet M. Thornton, Nucleic Acids Res. 46 (D1) (2017) D618–D623.

[214] Valeria Putignano, Antonio Rosato, Lucia Banci, Claudia Andreini, Nucleic Acids Res. 46 (D1) (2017) D459–D464.

[215] Jianyi Yang, Ambrish Roy, Yang Zhang, Nucleic Acids Res. 41 (D1) (2012) D1096–D1103.

[216] A. Bateman, M.J. Martin, S. Orchard, M. Magrane, R. Agivetova, S. Ahmad, E. Alpi, E.H. Bowler-Barnett, R. Britto, B. Bursteinas, H. Bye-A-Jee, R. Coetzee, A. Cukura, A. Da Silva, P. Denny, T. Dogan, T. Ebenezer, J. Fan, L.G. Castro, P. Garmiri, G. Georghiou, L. Gonzales, E. Hatton-Ellis, A. Hussein, A. Ignatchenko, G. Insana, R. Ishtiaq, P. Jokinen, V. Joshi, D. Jyothi, A. Lock, R. Lopez, A. Luciani, J. Luo, Y. Lussi, A. MacDougall, F. Madeira, M. Mahmoudy, M. Menchi, A. Mishra, K. Moulang, A. Nightingale, C.S. Oliveira, S. Pundir, G. Qi, S. Raj, D. Rice, M.R. Lopez, R. Saidi, J. Sampson, T. Sawford, E. Speretta, E. Turner, N. Tyagi, P. Vasudev, V. Volynkin, K. Warner, X. Watkins, R. Zaru, H. Zellner, A. Bridge, S. Poux, N. Redaschi, L. Aimo, G. Argoud-Puy, A. Auchincloss, K. Axelsen, P. Bansal, D. Baratin, M.C. Blatter, J. Bolleman, E. Boutet, L. Breuza, C. Casals-Casas, E. de Castro, K.C. Echioukh, E. Coudert, B. Cuche, M. Doche, D. Dornevil, A. Estreicher, M.L. Famiglietti, M. Feuermann, E. Gasteiger, S. Gehant, V. Gerritsen, A. Gos, N. Gruaz-Gumowski, U. Hinz, C. Hulo, N. Hyka-Nouspikel, F. Jungo, G. Keller, A. Kerhornou, V. Lara, P. Le Mercier, D. Lieberherr, T. Lombardot, X. Martin, P. Masson, A. Morgat, T.B. Neto, S. Paesano, I. Pedruzzi, S. Pilbout, L. Pourcel, M. Pozzato, M. Pruess, C. Rivoire, C. Sigrist, K. Sonesson, A. Stutz, S. Sundaram, M. Tognolli, L. Verbregue, C.H. Wu, C.N. Arighi, L. Arminski, C. Chen, Y. Chen, J.S. Garavelli, H. Huang, K. Laiho, P. McGarvey, D.A. Natale, K. Ross, C.R. Vinayaka, Q. Wang, Y. Wang, L.S. Yeh, J. Zhang, P. Ruch, D. Teodoro, Nucl. Acids Res. 49 (D1) (2021) D480–D489.

[217] S. Bienert, A. Waterhouse, T.A. de Beer, G. Tauriello, G. Studer, L. Bordoli, T. Schwede, Nucl. Acids Res. 45 (D1) (2017) D313–D319.

[218] Rahul Nikam, A. Kulandaisamy, K. Harini, Divya Sharma, M. Michael Gromiha, Nucleic Acids Res. 49 (D1) (2020) D420–D424.

[219] Neil D. Rawlings, Alan J. Barrett, Paul D. Thomas, Xiaosong Huang, Alex Bateman, Robert D. Finn, Nucleic Acids Res. 46 (D1) (2017) D624–D632.

[220] Antonina Andreeva, Eugene Kulesha, Julian Gough, Alexey G Murzin, Nucleic Acids Res. 48 (D1) (2019) D376–D382.

[221] Antonina Andreeva, Dave Howorth, Cyrus Chothia, Eugene Kulesha, Alexey G. Murzin, Nucleic Acids Res. 42 (D1) (2013) D310–D314.

[222] Ashley C. Stuart, Valentin A. Ilyin, Andrej Sali, Bioinformatics 18 (1) (2002) 200–201.

[223] David Sehnal, Mandar Deshpande, Radka Svobodová Vařeková, Saqib Mir, Karel Berka, Adam Midlik, Lukáš Pravda, Sameer Velankar, Jaroslav Koča, Nature Methods 14 (12) (2017) 1121–1122.

[224] Liang Qiao, Dongqing Xie, Anal. Biochem. 566 (2019) 75–88.

[225] Jacquelyn McCullough, Petra Fey, Ryan J. Rahman, Morgan Wallace, Seeta Morey, Kyle Sahlberg, Ethan McGonagle, Danielle Hess, Chance Hatfield, Michaela-Romina Sarmiento, Jordi Velasquez, Richard H. Gomer, MicroPubl. Biol. (2021).

[226] Arian R. Jamasb, Ben Day, Cătălina Cangea, Pietro Liò, Tom L. Blundell, Methods in Molecular Biology, Springer US, 2021, pp. 263–288.

[227] P. Benkert, M. Biasini, T. Schwede, Bioinformatics 27 (3) (2011) 343–350.