RESEARCH

A molecular docking integrated platform for virtual screening of bioactive compounds

✉Dany Naranjo Feliciano[1], Leobel Izquierdo Alvarez[1], Romel Calero Ramos[2,3],
Lilian Hernández Alvarez[1], Jesús Bouza Figueroa[2], Diego Enry Barreto Gomes[4],
Pedro A Valiente[5], Humberto Andrés Carrillo Calvet[3], Pedro Geraldo Pascutti[4]

[1] Centro Nacional de Sanidad Agropecuaria (CENSA)
Carretera de Tapaste y Autopista Nacional, CP 32700, San José de las Lajas, La Habana, Cuba
[2] Instituto Finlay de Vacunas
Ave 27, # 19805, La coronela, La Lisa, La Habana, Cuba
[3] Centro de Ciencias de la Complejidad, Universidad Nacional Autónoma de México,
CP 04510, México
[4] Instituto de Biofísica Carlos Chagas Filho, Universidade Federal do Rio de Janeiro (UFRJ), Brazil
[5] Centro de Estudios de Proteínas (CEP), Facultad de Biología, Universidad de la Habana, Cuba
✉dany@censa.edu.cu

ABSTRACT

The computer-aided drug design uses computational chemistry tools to discover, enhance or study drugs related to biologically active molecules. Here, we developed a molecular docking integrated and distributed platform (MODIP) for the virtual screening of bioactives compounds. MODIP is a webserver application, which downloads and prepares the compounds from the PubChem BioAssay database in a suitable format for docking with Autodock vina. The platform is able to screen in parallel chemotherapeutic receptors and its off-targets related proteins. MODIP can be run locally and is adaptable for running others public docking software and chemical databases.

Keywords: PubChem-BioAssay database, molecular docking, integrated and distributed platform

RESUMEN

Plataforma integrada de acoplamiento molecular para el cribado virtual de compuestos bioactivos. El diseño de fármacos asistido por computadora utiliza herramientas de la química computacional para descubrir, mejorar o estudiar los medicamentos relacionados con las moléculas biológicamente activas. Aquí desarrollamos una plataforma de acoplamiento molecular integrada y distribuida (MODIP) para el cribado virtual de compuestos bioactivos. MODIP es una aplicación de servidor web, que descarga y prepara los compuestos de la base de datos PubChem BioAssay en un formato adecuado para el acoplamiento molecular con el programa Autodock vina. La plataforma es capaz de examinar en paralelo los receptores quimioterapéuticos y sus blancos moleculares no específicos. MODIP se puede ejecutar localmente y es adaptable para ejecutar otros programas de acoplamiento público y bases de datos químicas.

Palabras clave: Base de datos PubChem-BioAssay, acoplamiento molecular, plataforma integrada y distribuida

Introduction

In concert with the continued need for improvements of in silico docking accuracies, the explosive growth of commercial and publicly available chemical databases requires computational techniques to efficiently implement docking protocols and rapidly screen millions of compounds in a timely fashion. The goal of docking is to identify, among the large number of possible orientations of a ligand within the binding site of the target, the one closest to the experimental structure of the complex [1]. This is done by using a mathematical function that accounts for the goodness of the coupling between ligand and target. Consequently, two key elements of the docking problem are: a) a good sampling method, and b) an accurate scoring function. Virtual screening (VS) is the extrapolation of docking to the case in which a large database of molecules is going to be processed [2].

Several docking programs have been developed for VS since the initial development of UCSF Dock, such as AutoDock [3], GOLD [4] and GLIDE [5]. There have been reports of the successful identification of lead compounds using docking-based VS methods [6-14]. To perform VS, the combination of various computational programs and molecular graphic environments is required. The integration of all these tools in a way that all of them are executed in a single graphic environment could reduce the difficulty of VS protocol implementations. However, few integrated software platforms are freely available today such as IVSPlat [15], WinDock [16], PyRx [17], MOLA [18], DOVIS [19] VSDMIP [20] and K-screen [21].

In this study, a VS web platform was developed by the integration of some of the methodologies applied to computer-aided drug design. Free tools (MySQL, Apache, Ruby, RoR, Bootstrap, NCBI Entrez Utilities, PUG SOAP web services and AutoDock tools) and open source software (AutoDock Vina and Open Bebel) were used to build the whole application. PubChem BioAssay is a database (www.ncbi.nlm.nih.gov/ pcassay) which stores chemical compounds used

1. Perola E, Walters WP, Charifson PS. A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. Proteins. 2004;56(2):235-49.

2. Shoichet BK. Virtual screening of chemical libraries. Nature. 2004;432:862-5.

3. Morris GM, Goodsell DS, Halliday RS, Huey R, Hart WE, Belew RK, *et al*. Automated docking using Lamarckian genetic algorithm and an empirical binding free energy function. J Comput Chem. 1998;19(14):1639-62.

4. Jones G, Willet P, Glen RC, Leach AR. Development and validation of a genetic algorithm for flexible docking. J Mol Biol. 1997;267:727-48.

5. Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. J Med Chem. 2004;47:1739-49.

in experimental assays with different therapeutic targets; it is the main data source for MODIP. VS using MODIP is semi-automated due to the user must define before the receptor binding site cavity and the grid box size required for docking with Autodock Vina. The size and the exact position of the box should be defined by the user using external tools like Autodock/Vina plugin for PyMol [22] or AutoDock tools [23].

A study case for docking is the enzymatic activity cathepsin B. This is a potential terapeutic target for the control of the zoonosis caused by *Fasciola hepatica*. Furthermore the mammal proteins have being studied as therapeutics targets against cancer. Despite there are multiple assays of compounds evaluated to the rat, bovine and human cathepsin B, there is no information against the parasite enzyme. In this regard, we reported the results of "cathepsin B" as PubChem-BioAssay query; docked to human, bovine and *Fasciola hepatica* cathepsin B. Other selection criteria were that the three dimensional structures are close (high identity percentage), the catalytic pockets are open cavities and have being solved free or as complex, in the binding mechanism are not heteroatoms involucrate. Moreover, the compounds described as potential inhibitors do not have atoms incompatible with the Autodock vina scoring function.

Given these reasons, this work was aimed to implement a VS web platform of related proteins to update, manage and notify molecular docking results from the PubChem databases, for therapeutic targets and/or its homologous proteins or off targets.

*M*ethods

Platform design

MODIP is based on Linux/Apache/MySQL/Ruby/RoR/Bootstrap/Autodock Vina and NCBI Entrez Utilities and PUG SOAP web services. In the preparation of the system programming language Ruby (1.9.3) was used. Rails (3.2.8) served as a framework to develop web application. The transmission system database used was MySQL Server (5.5.38) in conjunction with Workbench (6.0.8) [24] as a visual interface to support the manipulation of data during the development phase. The molecular docking was made by AutoDock Vina program package (1.1.2-3) [25], together with Open Babel (2.3.2) [26] and several functionalities from the AutoDock Tools program (1.5.7) [27]. The visual interface design was developed through the web development platform bootstrap [28]. This interface provides the point of access to all system functions. Users have two categorizations within the web application: in a session or not. This role will determine the ability to query and access information related to the results of molecular docking.

Download engine

The Download Engine is the component where the connection with the services to download is built provided by PubChem SOAP (PUG SOAP and NCBI Entrez Utilities). Its' code is encapsulated in a library called download engine. This engine is also inserting data processed through the database module handles. This module has a mechanism to ensure that a download could not insert data that was previously or are being processed by another download. The *soap4r library* is used to develop client applications or server applications using SOAP [29]. Within this lies the *wsdl2ruby.rb* executable that converts all the information that a web service provides into a WSDL file with Ruby classes. In this project soap4r was used to establish a model based on objects PUG SOAP and NCBI Entrez Utilities services model. By running the download engine is filled the local database and the files related to the assays, with extension CSV, and the compounds in SDF, are obtained. These are in the server represented in two (SDF 2D) or three (SDF 3D) dimensions.

Receptors

VS can be based on prior knowledge of the three-dimensional structure of a target (nucleic acids or proteins), available on the Protein Data Bank (PDB) database through the website www.rscb.org/pdb/home/home.do. The methods for obtaining such a structure can be experimental or by homology modeling. Experimental methods are the most used: crystallography X-ray diffraction and nuclear magnetic resonance. The formats are stored in a *.pdb or *.ent with the information about the Cartesian coordinates of the atoms forming molecules. In this work, they were used as test the human and bovine "cathepsins b" receptors (pdb codes 1GMY: A and 1QDQ: A) as well as a model by homology modeling of the *F. hepatica* "cathepsin b" (with the *Shistosoma mansoni* cathepsin B as template, pdb code 3QSD). Input files for molecular docking were obtained through the script *prepare_receptor4.py* from the program AutoDock Tools.

Database

The data was taken from PubChem repository (pubchem.ncbi.nlm.nih.gov) for filling the local database and the construction of a local directory, where the files that establish the relation assay (CSV extension) and compounds (SDF extension) are stored. Of the three databases that comprise the *PubChem Bioassay* repository available at www.ncbi.nlm.nih.gov/pcassay and Compound available at www.ncbi.nlm.nih.gov/pccompound were initially taken into account, although not being applied for Substance.

Molecular docking

The establishment of an order of the compounds according to the results of molecular docking of them with the system targets is one of the fundamental goals of the platform. This module is responsible for obtaining the necessary values for this ordering. Just like the download module, it is a distributed service accessible from Ruby druby: //localhost:1992. It uses the docking library as the main center for molecular docking. This is assisted by the Open Babel, AutoDock Vina programs, besides the script *prepare_ligand4.py* from the AutoDock Tools program. Figure 1 shows the procedure with a data set employed (cathepsin B as query).

*R*esults and discussion

MODIP infrastructure

The system consists of three distributed components and a web interface. The first component

6. Sousa SF, Cerqueira NM, Fernandes PA, Ramos MJ. Virtual screening in drug design and development. Comb Chem High Throughput Screen. 2010;13:442-53.

7. Makino S, Kuntz ID. Automated flexible ligand docking method and its application for database search. J Comp Chem. 1997;18(14):1812-25.

8. Bissantz C, Folkers G, Rognan D. Protein-based virtual screening of chemical databases. 1. Evaluation of different docking/scoring combinations. J Med Chem. 2000;43(25):4759-67.

9. Lyne PD. Structure-based virtual screening: an overview. Drug Discovery Today. 2002;7(20):1047-55.

10. Kitchen DB, Decornez H, Furr JR, Bajorath J. Docking and scoring in virtual screening for drug discovery: methods and applications. Nat Rev Drug Discov. 2004;3(11):935-49.

11. Kellenberger E, Rodrigo J, Muller P, Rognan D. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. Proteins: Structure, Function, and Bioinformatics. 2004;57(2):225-42.

12. Krüger DM, Evers A. Comparison of structure- and ligand-based virtual screening protocols considering hit list complementarity and enrichment factors. ChemMedChem. 2010;5(1):148-58.

13. Pereira JC, Caffarena ERI, dos Santos CN. Bosting docking based virtual screening with deep learning. J Chem Inform Model. 2016;56(12):2495-506.

14. Monsen RC, Deleeuw L, Maguire J, Dean WL, Gray RD, Chaires JB, *et al*. Structure-based drug discovery: computational virtual screening; 2017 [cited 2021 Jan 17]. Available from: https://uknowledge.uky.edu/ccs2/2017/posters/19/

15. Sun YX, Huang YX, Li FL, Wang HY, Fan C, Bao YL, *et al*. IVSPlat 1.0: an integrated virtual screening platform with a molecular graphical interface. Chem Cent J. 2012;6(1):2.

16. Hu Z, Southerland W. WinDock: structure-based drug discovery on Windows-based PCs. J Comput Chem. 2007;28(14):2347-51.

17. Wolf LK. New software and Websites for the chemical enterprise. Chem Eng News. 2009;87(45):31.

18. Abreu RMV, Froufe HJC, Queiroz MJRP, Ferreira ICFR. MOLA: a bootable, self configuring system for virtual screening using AutoDock4/Vina on computer clusters. J Cheminf. 2010;2:10.

19. Zhang S, Kumar K, Jiang X, Wallqvist A, Reifman J. DOVIS: an implementation for high-thoughput virtual screening using AutoDock. BMC Bioinformatics. 2008;9(1):126.

20. Gil-Redondo R, Estrada J, Morreale A, Herranz F, Sancho J, Ortiz AR. VSDMIP: virtual screening data management on an integrated platform. J Comput Aided Mol Des. 2009;23(3):171-84.

21. Tai D, Chaguturu R, Fang J. K-screen: A free application for high throughput screening data analysis, visualization, and laboratory information management. Comb Chem High Throughput Screen. 2011;14(9):757-65.
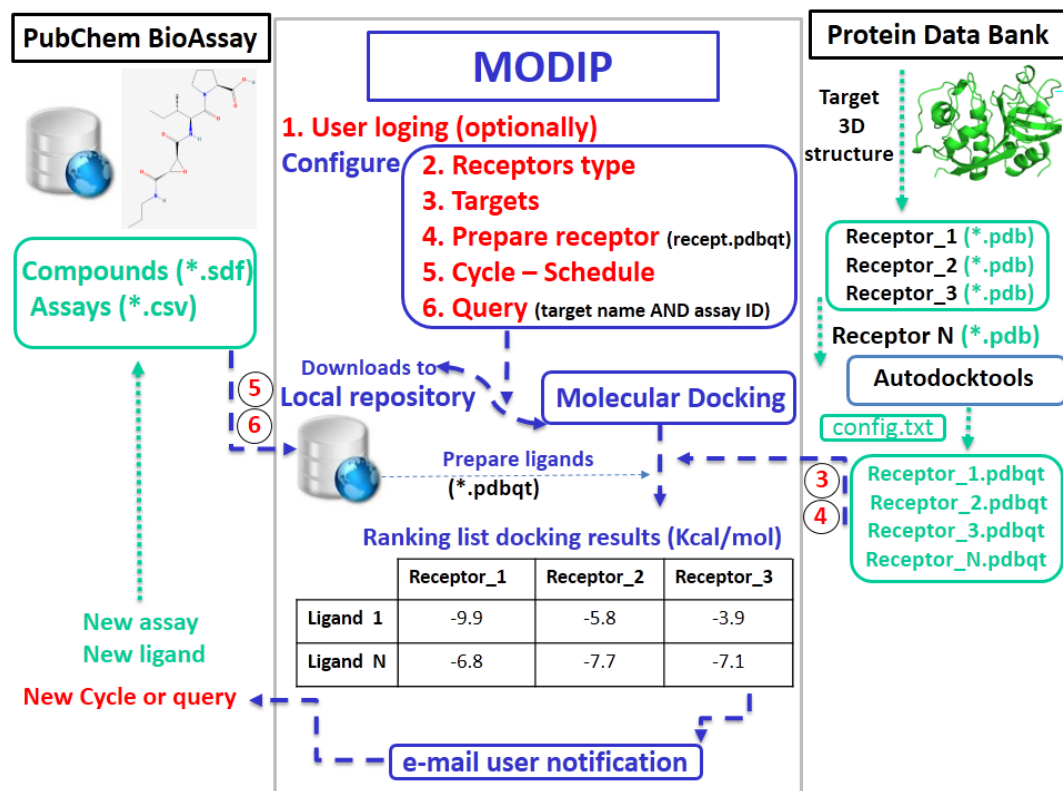
Figure 1. MODIP data set workflow. The colors of the components have the following meaning: red, steps followed by user; green, input data; blue, processes made by the platform. The user must login, configure the system and suggest the queries. Previously he must prepare the receptor.pdbqt archive for each of the three receptors and the conditions for the molecular docking provided on the config.txt archives (center and dimensions of the grid, energy range, etc.). As results, the user will receive an email with the sorted docking result list. The system will remain active in an iterative way while exist the update cycle execution functioning or will be done a new query.

handles the download requests made by users (download module). The second one performs the virtual screening of the chemical database using as receptor a specific therapeutic target (docking module). The third component is responsible for interacting with the local databases, avoiding the incorporation of duplicate records (database module). The web interface allows the access of the users to the systems functionalities and displays the stored data. In this interface are the *Query Manager* and *Docking Manager* classes, which provide the link to the download and docking modules respectively, coined as *Download Observer* and *Docking Observer*. These modules are responsible for managing the notifications generated during downloads and docking. Figure 2 shows the interaction among the main system components.

Notifications to the users

The user can operate the system through the web application, where all information relating assays and compounds are public (no credentials are required to access to them), except the table that indicates the order of the compounds according to the values of these molecular docking. Once the user utilizes the web application, it has the opportunity to log in. When it initiates the execution of a query, the web application

starts a notice service showing the discharge process associated with the query.

This functionality is achieved by *HTML5 Server-Sent Events (SSE)*. Due to the use of SSE, the platform should be accessed from a web browser that supports *HTML5*, otherwise it is not possible to view the download process of new compounds.

Download of assays and compounds

An important part of the system is the incorporation of new assays and compounds, through the implementation of downloads. These downloads are processed in parallel by the module downloads through a set of processes. The data required to run the download are serialized and sent over a *socket* to the processes. These data were serialized and deserialized with the help of methods *marshal_dump* and *marshal_load* of module Marshal respectively [30]. The processes use, within an infinite cycle, the *accept* method of the socket class to process downloads requests as assigned [30]. If it required that a download is processed but there are no processes available, the action is executed by the module itself through the creation of another execution thread with the *Thread* class [30].

The download module, besides being a distributed service, implements the observable pattern. From the web interface to run a query from the object class

22. Seeliger D, de Groot BL. Ligand docking and binding site analysis with PyMOL and Autodoc/Vina. J Computer-aided Mol Design. 2010;24(5):417-22.

23. Morris G, Huey R, Lindstrom W, Sanner MF, Bellew RK, Goodsell DS, *et al*. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. J Comp Chem. 2009;16:2785-91.

24. López J. Localización de equipos robados con Adeona. Todo Linux: la revista mensual para entusiastas de GNU/Linux. 2009;9(106):51-3.

25. Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading. J Comp Chem. 2010;31(2):455-61.

26. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open Babel: An open chemical toolbox. J Cheminformatics. 2011;3(1):33.

27. Sanner M, Huey R, Dallakyan S, Karnati S, Lindstrom W, Morris G, *et al*. AutoDock-Tools, version 1.4.5. The Scripps Research Institute, La Jolla, CA, USA; 2007.

28. Lerner RM. At the forge: twitter bootstrap. Linux J. 2012(218):6.

29. Vinoski S. Scripting JAX-WS IEEE Internet Computing. 2006;10(3):91-4.

Query_Observer is created and linked to the service in order to get the notifications it receives by the download engine. Each notification has a unique identifier that relates to the query from which it comes, so the notifications are identified and analyzed only by the object which establishes a link with the product of that query server. When this module starts its service, it takes two optional parameters; the first represents the size of the set of processes to create and the second, the "proxy" for the internet connection during download. If you do not specify a size for the whole process, it will not be created and there is no process available. For internet, it is assumed that a "proxy" is not required if this is not specified.

Data access

The access module database is responsible for inserting the assays and downloaded compounds as well as handling any request to the database by the download engine. This module uses *ActiveRecord* to connect to the underlying database system, its default configuration, *ActiveRecord* maintains a set of five connections to handle requests to the database. The connections can be used by only one thread at a time and each one served only one order. After the application, the connection is re-built (not necessarily immediately) to the connection, in order to be used again. If not any available connection is not found at the time of the application, *ActiveRecord* waits for a time interval (five seconds by default), trying to enabled a connection for use. If this does not occur, an error is thrown and the request is not processed. Due to the possibility of many download applications running simultaneously (by one or more users), the number of connections is increased to 20 and each connection enclosed in a box that tells *ActiveRecord* immediate return at the end of this application. This ensures having the largest number of connections available at all times.

Moreover, it is necessary to avoid downloading and storing data more than once. To achieve this, the download module uses a dictionary with the identifiers of the assay and compounds (AID's and CID's, respectively) being downloaded, but have not been inserted into the database. These dictionaries are handled by a monitor that prevents concurrency problems that are generated when two or more processes try to access these resources at the same time. In this way, when a process is about to download an assay or compound, the module verifies that it is not being processed (its identifier does not appear in the corresponding dictionary) and that it does not exist in the database. When the resource is inserted into the database, its identifier will be eliminated from the corresponding dictionary.

Database

The data were managed system with a relational database *MySQL*. The structure of the tables that comprise the database relied heavily on information taken from PUG web services SOAP and NCBI Entrez Utilities. A lot of data was obtained of these services, however, there were taken only those deemed relevant and interesting by the users to their investigations. It was used a numeric auto incremental value id as primary key in all tables.
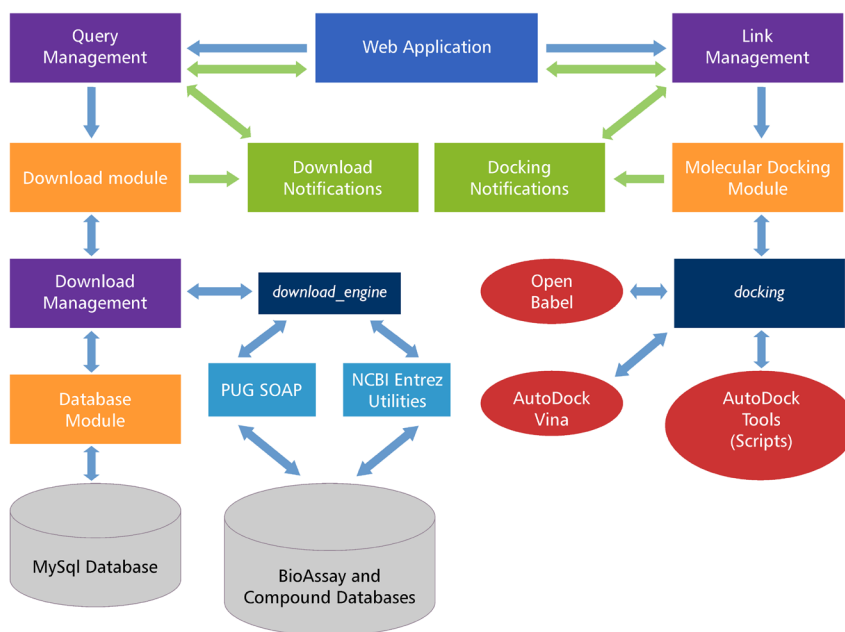

Figure 2. System components flow. The colors of the components have the following meaning: purple, connection components between two modules; orange, distributed services; green, notifications components; blue, download and molecular docking library; red, molecular docking programs; light blue, *PubChem* web services and gray, the local and *PubChem* databases. The green arrows indicate the flow of notifications while the blue ones indicate the communication among components.

Each foreign key id has the format *xxx_id*, where *xxx* is the singular name of the table to which it refers. This avoided extra settings establishing the navigation properties in the object model created by *ActiveRecord*.

Assays

The process of insertion of the data begins downloading the assays corresponding to a therapeutic target. Using NCBI Entrez Utilities Service, specifically the *ESearch* role, allowed the identification of assays related to a target using more expressiveness. The *ESearch* function receives two mandatory parameters that relate to the *PubChem* database, where it will make finding relevant and the query in question. The query has its own syntax, which can filter the information for all the indexed fields possessing the database. The *Einfo* function returns the indexed fields of a database if it is specified as a parameter. Importantly *ESearch* has another set of parameters divided into the following three groups:

Server files

Within this group are the *usehistory*, *WebEnv* and *query_key* parameters. Using these three parameters ensured an optimization in the search, because with them the server keeps track of the results of a query to be used in subsequent calls.

Recovery

The *retstart* and *retmax* parameters were used in obtaining the information by intervals. With *retstart* the starting index of data recovery is specified and with *retmax* the amount recovered.

30. Thomas D, Fowler C, Hunt A. Programming Ruby. The Pragmatic Programmers' Guide. Second edition. Pragmatic Bookshelf; 2005.

Date

The parameter *MinDate* specifies the oldest date that must have the results of the search; analogously *Max-Date* represents the top date. Thus, any query is filtered by the range between the two dates.

Invoking the function *ESearch* finally returns an object that contains the list of identifiers of the assay (AID's) associated with the query. After obtaining the IDs, proceed to perform the insertion into the database using the functions PUG SOAP[31].

Compounds

To download the compounds, we proceeded to parse the CSV file for each assay. As a result, the identifiers of each compound are recovered (CID's). Unlike assays, compounds have PUG SOAP functions that extract the information, so that this had to be retrieved from the referenced file.

The SDF extension can be processed by the Open Babel program, which carries information from one file to another SDF 3D*2 in mol2 and then accepted by the program package AutoDock Vina. Hence, this type of file is encoded in a readable format for all people.

PUG SOAP has the facility to download the compounds in two dimensions (file SDF 2D *2) and three dimensions (SDF 3D file). This work was priority downloading three dimensional compounds as they may be incorporated directly into the molecular docking strategies. Despite not being available in 3D, it is then downloaded 2D, allowing them to be taken after SDF 3D. As it is the case with the assays, not all the information provided by the compound file is part of the compounds table in the local database. SDF 3D and SDF 2D are nomenclatures only made in this paper to refer to SDF files whose coordinates are in two or three dimensions respectively.

Molecular docking

The molecular docking is responsible for obtaining values that establish the level of interaction between one or more chemical compounds represented in three dimensions and a therapeutic target (receptor). There are many criteria to obtain these values. In this platform, an energy criterion was used, with which the results of the interaction are measured in terms of energy expenditure [32]. One of the methods used to make the molecular docking involves an iterative process in which the target compounds are prepared by a set of programs. There are studies that grouped all these actions in one script responsible for all processing [33]. This approach provides better performance when the work is led by specialists in this branch. The present work was taken into account thus making the molecular docking in conjunction with a visual interface for easy access to this functionality.

Successful completion of this procedure in the system only requires the preparation of the compounds. AutoDock Vina receives its input parameters between two files that identify the receptor (target) and the ligand (compound) in PDBQT format. Although the system allows the user to insert the receptor directly in this format, the compounds are stored with SDF extension. This decision to keep the compounds represented as SDF is based on this representation as basic,

so obtaining PDBQT is done by conversion programs. Depending on how good the program convert formats, the better quality will end. Therefore, the incorporation of an improved performance that depends on the conversion can be achieved in the system by replacing the converter used.

To obtain a PDBQT from an SDF file, one must pass through two stages of conversion. First the Open Babel program was used to convert the SDF file to MOL2. This was used in the system by calling the operating system indicating its execution in another process [30]. However, as part of its distribution, Open Babel provides an interface for applications that can be used and integrated for various programming languages (including Ruby). With this facility all the features of Open Babel are encapsulated in a bookstore with the same source language to use the code.

Once the file is in MOL2 extension, the python script *prepared_ligand4.py* turns it to a PDBQT. Finished later this, the file is now ready to perform for AutoDock Vina docking. Other required parameters are those that identify the three-dimensional space (box) where the receptor-ligand interaction is performed. This space is defined by a center of Cartesian coordinates and dimensions of width, height and depth. Optionally you can specify the program number of conformations of the compound. Each shape represents a different position of the compound in the process of interaction with the target (change of its coordinates). In addition, you can specify a range of energy value representing the maximum energy difference between the best and worst conformation and the amount of chips to be used for docking. By default it uses all that are available. All these parameters are specified directly through the web interface. The output from AutoDock Vina is a PDBQT file with the energy values for each molecular conformation docking. The end result for a compound is the minimum of the energy values of this with the receptor.

This functionality is present in the system in two different ways. A molecular docking can be performed by the user directly through a Web interface view; this allows setting the docking by selecting compounds that intervene in the prosecution and the receiving target.

The user can view the process in a similar way to what happened with queries, once the docking implementation begins, through notifications with "HTML5 Server-Sent Events" application, shown in figure 3. This functionality is performed in another way by downloads made by the engine of download. When finished downloading, the automatically downloaded to all compounds will have a simple molecular docking with the target of the query that produced the download belongs. The results are analyzed by performing an ordering of these novel compounds in relation to previously stored in the system. The final order to occupy the new compounds is sent in a notification via email to the user responsible for the query. This email also contains the highest scores (thresholds) of the target compounds.

Compounds ranking

The ranking of a compound in the list has to do with the specificity of the binding of this target to be taken

31. PubChem N NCBI PubChem [Online]. c2021 [cited 2001 Jan 17]. Available from: http://pubchemncbinlmnihgov/ pug_soap/pug_soap_helphtml

32. Meng EC, Shoichet BK, Kuntz ID. Automated docking with grid-based energy evaluation. J Comp Chem. 1992;13(4):505-24.

33. Sharma V, Pattanaik KK, Jayprakash V, Basu A, Mishra N. A utility script for automating and integrating AutoDock and other associated programs for virtual screening. Bioinformation. 2009; 4(2):84-6.
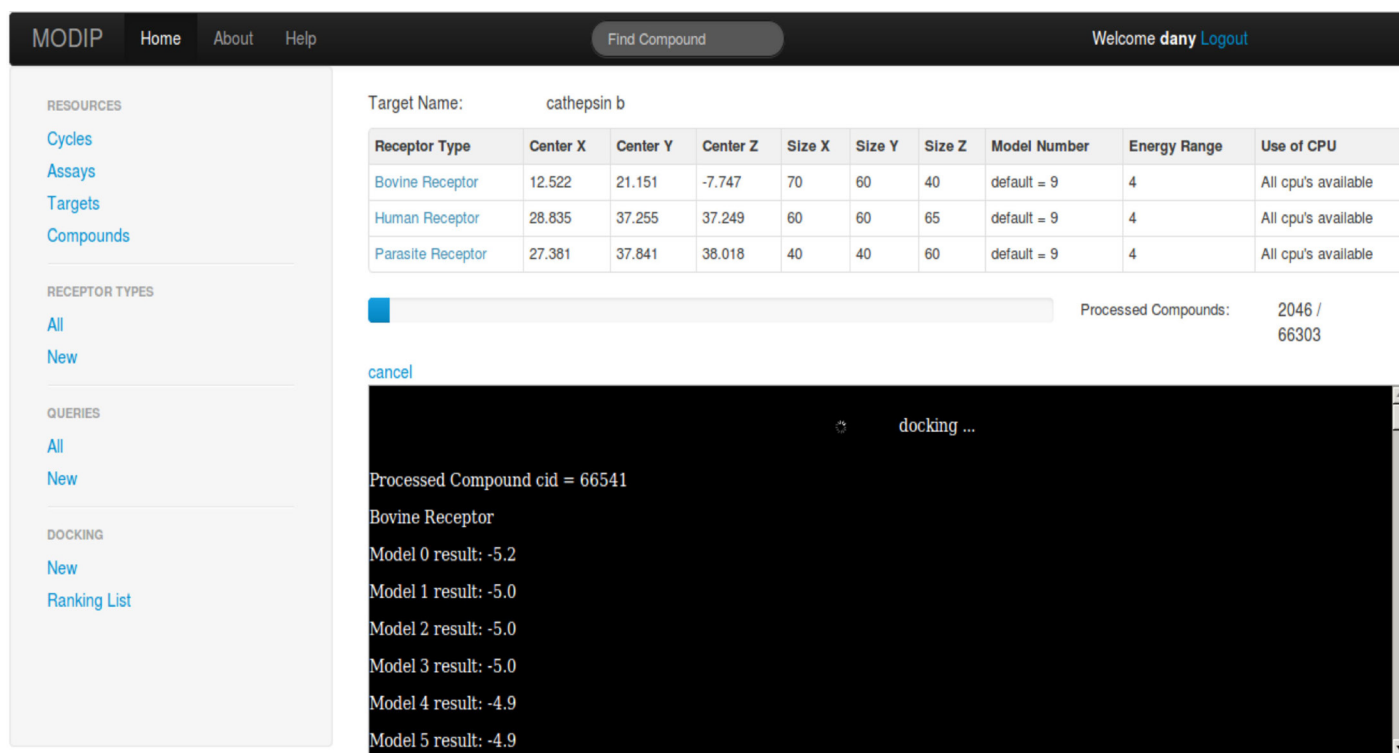
Figure 3. Molecular docking progress. In the top view shows the parameters of the receptors involved in the processing (bovine cathepsin B receptor). Details of the results are shown in the black area below the progress bar.

to the order. The targets used as test set are divided by a set of receptors belonging to three categories. The categories of receptors in this example are: Human receptor, Bovine receptor and Parasitic receptor. To establish an order with a therapeutic target, each compound is assigned a triplet that contains the minimum values of molecular docking for each target receptor, grouped by category. Once these values have been computed, the system is performed through a comparison between objects that group CID values of the compounds and their respective panels. The CID value is used only to define an order if an order cannot be achieved by other means which will be mentioned below.

The comparison algorithm receives two objects O1 and O2 returning two values 1 or -1. The value 1 indicates that the object is greater than O2 O1, instead; the value -1 indicates that O1 must occupy a lower position than O2 in the final order. O1 is less if the absolute value of the subtraction of the first two values in its internal is larger than this same value in O2. In case of computing the same result for the two objects, the same procedure is considered but with the first and last value in the triads. If after these operations the same values remain, it returns -1 if the CID of O1 is less than the O2, and 1 otherwise. The CID is a unique field, so that the values are different for two different objects. If any of the above views subtraction cannot be made by the omission of any of the values of the triplet, the object with the missing value occupies a larger position in the order.

An important feature of this method is that depending on the location of the values within the triads the order of the compounds can change. This feature is taken into account in the system with the ability to specify what the user appears in the three receptor positions available to the system. The order of the receptors is the same that occupy in the triads. In the case of automatic molecular docking, the configuration taken by the system to fix the order was preset so that the receptor of the parasite is placed first, then the bovine receptor and finally the human. This order is to identify those compounds that inhibit the parasite and have the least effect on the cattle first and humans in second place. Figure 4 shows the order of the compounds for the following configuration of receptors: Parasite receptor, Human receptor and Bovine receptor.

The design of inhibitors of *F. hepatica* cathepsin B (initial data platform) can be based on the exploration of the structural differences between them and mammals. The high similarity between their binding sites makes available all information relating to inhibitors as the basis for the rational design of molecules with higher substrate specificity to parasitic proteases. Although the differences observed in this study did not result in significant appearance, the basis for more comprehensive studies that will define the true importance of these proteins as therapeutic target. The strength of this work resides in that this tool can be extrapolated to any therapeutic target in order to update methodologically the research for new drugs with three related receptors (by homology, paralogous, affinities in cross reaction, etc) at the same time.

Conclusions

VS and *in silico* drug design techniques have become one of the important processes for medicinal chemists

Figure 4. Compounds order. Cathepsin b target order, under the receptors configuration in the order: parasite, human and bovine. The red asterisks (*) remark the compounds selected as actives in one or different assays.

to conduct their rational design of novel scaffold compounds and can even predict their bioactivities before being synthesized. Furthermore, commercial software continues to expand on the core user interface, and new algorithms from both industry and academia are quickly into the high-end packages. Public domain packages are becoming more stable and offering functionality that rivals some of the commercial offering.

This platform has a relational database to store the relevant assay data of therapeutic targets and compounds ensuring referential integrity. The downloaded information is handled by a distributed service that works as a separate server. These are made using the NCBI Web services *Utilities Entrez and PubChem* PUG SOAP. From the web interface a user can schedule the execution of tasks to be performed in this service. Integrating programs of rational drug design yielded energy values for all compounds discharged to the system in three dimensions (SDF 3D). The users can select until six different criteria to sort the docking results, for a target with three receptors (e.g., the therapeutic target and two homologous proteins) and sets the order of specificity and substrate selectivity of a compound for different targets. In case that there were more receptors to evaluate, this number increases by the formula n!/(n-3)!. The system also has a deployed web application in Ruby on Rails allows PubChem similar to features in a local context. This provides a set of claims for downloading new data and performs molecular docking. As added value, it reports via email necessary for decision making. For these reasons, this Molecular Docking Integrated Platform is a useful and friendly tool for finding new prophylactic or therapeutic drugs.

MODIP platform is deployed and accessible at *modip.biof.ufrj.br:443* (at this moment it is only available as an internal platform for the LMDM: Laboratory for Molecular Modeling and Dynamics). The whole code for the platform is available at two public github repositories (https://github.com/r-calero/modip) and (https://github.com/LMDM/modip). It can be cloned and deployed at any data center under GNU Affero General Public License.

*A*cknowledgements

*C*onflicts of interest statement

The authors declare that there are no conflicts of interest.