UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

AGRONÓMICA,ALIMENTARIA Y DE BIOSISTEMAS

MÁSTER EN BIOLOGÍA COMPUTACIONAL

DEPARTAMENTO DE BIOTECNOLOGÍA VEGETAL

CENTRO NACIONAL DE BIOTECNOLOGÍA

Integration of Pocket Analysis Software in Molecular Dynamics Trajectories for Structural Biology and Virtual Drug Screening in Scipion

TRABAJO FIN DE MÁSTER

Autor/a: Lobna Ramadane Morchadi

Tutor/a: María Garrido Arandia
Cotutor/a: Carlos Oscar Sorzano Sánchez

Junio de 2022

INDEX

ABSTRACT

Protein pocket and cavities study is a high relevant process in the Chemoinformatics field, especially, in areas such as Drug Discovery, where it is used to find regions of interest like ligand binding sites. Currently, multiple algorithms have been developed to exploit this data from protein structures. From these 3D structures a computational study can be performed not only to find these cavities (or pockets) but also to simulate the physiological conditions in which these are present through Molecular Dynamics simulations. These simulations allow to capture protein motion and flexibility, which is important as most of these cavities are changing and adapting depending on protein function and affinity. MDpocket is an open-source software for pocket identification and characterization along Molecular Dynamic simulations. This software requires of expertise knowledge and multiple previous and during execution steps. Meanwhile, Scipion is an integrative platform to perform friendly-user workflow in different fields, in this case, for Virtual Drug Screening and Structural Biology. With the aim to facilitate the use of the bioinformatics analysis with such software and provide new utilities in Scipion, a plug-in for MDpocket is developed to perform a complete pipeline just from the 3D structure of a protein. Also, program validation, analysis and visualization have been performed in Scipion with Pru p 3, a non-specific ligand transfer.

1. INTRODUCTION

Protein cavity detection and analysis provides highly significant information about the active sites for biological processes like protein-ligand binding and protein-protein interaction. In molecular graphics and modelization, the 3D structure of a given protein retrieved as a PDB (Protein Data Bank) file can be studied to computationally determine the localization and description of these cavities.

At the actual state of bioinformatics, the advance of the computational resources have allowed the design and development of multiple techniques and algorithms for the detection and characterization of pockets. From these molecular regions, we can obtain many properties to further our understanding in interacting regions and molecular interfaces. Thus, providing significant information for the design of complementary compounds, like active protein inhibitors or disruptors of protein-protein interactions.

Cavity properties of interest include not only biophysical and biochemical properties like electrostatics, H-bond properties, polarity etc., but also geometry such as depth, size and shape. All these factors combined are which enable the recognition and correct interaction of a ligand (small molecule) or protein to a target protein.

It is extremely useful to have computational methods to simulate biochemical processes like protein-ligand interaction to make laboratory experiences easier and save time and resources. However, this *in silico* studies have some difficulties related with the variety of suitable ligands, protein cavities and shape variations, along others…

<u>PROTEIN CAVITIES</u>

In the literature, different names appear refering to protein cavities depending on the shape and location of the protein surface: pockets, inner cavities, tunnels, clefts, grooves... (Figure 1). However, there is not a consensus definition of cavity, neither about their classification of cavities, and these terms like pocket, cavity, channel, tunnel and void are frequently used rather differently depending on the context, or even indifferently.

Mathematically, using the Theory of Convexity, a protein cavity can be defined as the connexion of its complement space inside its covex hull (Simões et al., 2017).

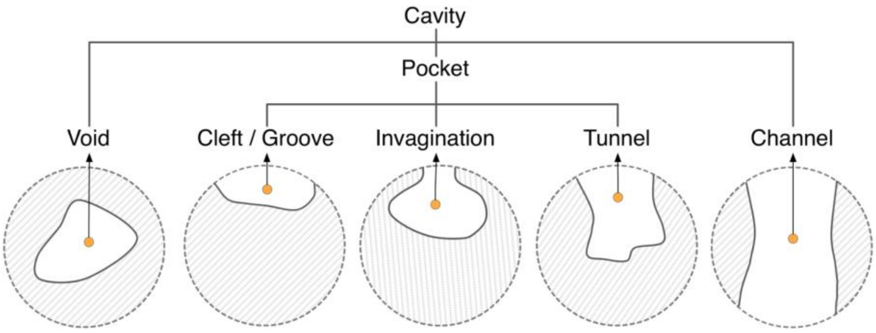


Figure 1: Cavity types (Simões et al., 2017)

1.1. Pocket analysis and detection in molecular dynamic trajectories

The physicochemical properties of the pocket, determined by the residues around it; the shape and localization in the protein establish the pocket functionality. The motion of the protein in the aqueous medium leads to the opening, adaptation and closing of pockets and regulates different processes like binding, molecule migration, conformational and function changes. For example, some cavities can exist permanently or transiently, like it occurs with tunnels and channels which transport compounds. This flexibility in the protein permits changes from small existing pockets to the formation of new ones, which may be related to a change in its function (Stank et al., 2017).

The structural flexibility of proteins is crucial for their adaptation to binding molecules, interaction specificity and other processes. Since proteins are dynamic systems, this dynamicity is important when developing and using methodologies for protein study (Arroyo-Mañez et al., 2011). Therefore, it is necessary to consider the internal motion when determining pockets and designing new binders. The exploration of protein flexibility enhances the chances of finding high-affinity interaction and identification of potential ligand/drug binding sites, thereby speeding the process of drug discovery (Wang et al., 2018; Barros et al., 2019).

Essentially, the study of protein cavities flexibility and characterization provides insight in fields like protein design and engineering, drug discovery and, moreover, enhances our understanding of molecular processes like ligand migration, protein-ligand interaction in atomic level, thus, contributing to our knowledge of biological processes (Perricone et al., 2018).

On the one hand, traditional approaches such as nuclear magnetic resonance (NMR) spectroscopy, X-ray crystallography provide high-valuable information as instants of the possible conformations of the protein, but not about processes like molecular recognition and binding. Additionally, other approaches like cryo-electron microscopy (Cryo-EM) and crystallization under high Xe pressure have been used to investigate the flexibility and dynamics of molecular systems, but these approaches are still time-consuming and expensive (Wang et al., 2018; Stank et al., 2017).

On the other hand, approaches based in computational methods like molecular dynamics (MD) simulations overcome the limitation of studying the motion of proteins. Currently, multiple MD simulation protocols are used to study protein dynamics on relatively short timescales thanks to the utilization of graphic processing unit (GPU). These devices enable the development of more affordable and efficient software to perform these simulations (Surpeta et al., 2020). Some of these software packages, which are nowadays available, are AMBER (CASE et al., 2005), CHARMM (Obst et al., 1998), NAMD (Phillips et al., 2005) and GROMMACS (Abraham et al., 2015).

The MD simulations of proteins are stored as trajectories: large, time-depending datasets of snapshots. When a trajectory is analysed, we can obtain information about the flexibility of cavities and the variation of its different conformations along the trajectory (Krone et al., 2011). This analysis can provide information that, initially, the static X-Ray or NMR structure cannot provide, like transient pockets, cryptic binding sites and allosteric sites, which improves our chances of exploiting the potential druggable sites (Durrant & McCammon, 2011).

Hence, the dynamics and structure of cavities of interest will be of high relevance for new drug design and development as to contribute to the Virtual Drug Screening Process (Wang et al., 2018).

1.2. Bioinformatic tools for pocket analysis and its applications

The understanding of processes related with protein binding requires the detection of these cavities where they take place. A computational estimation of its location is a powerful instrument for drug design, before the predicted interactions are validated with experimental laboratory work. For this purpose, many algorithms for predictions and identification of protein cavities have been developed. All these algorithms can be classified into 3 main categories:

- **Evolutionary algorithms**: They rely on multiple sequence alignment to find these cavities on a given surface.
- **Energy-based algorithms:** A computational calculation of the interaction energies between protein atoms and a small-molecule prove is done to detect the cavity.
- **Geometric algorithms:** They analyse the geometric properties of a molecular surface to detect the cavities and its depth.

Each approach has its drawbacks, for example, depending on the alignment quality we would have a better or worse performance of evolutionary methods; and energy-based rely on the quality of scoring function, filter and force-field used.

Focusing on geometry methods, these can be further divided in three main categories for cavity detection: grid-based, sphere-based and Voronoi-based. This classification can be extended, including Tesellation and consensus based, and Time-Varying (Simões et al., 2017) (Figure 2).

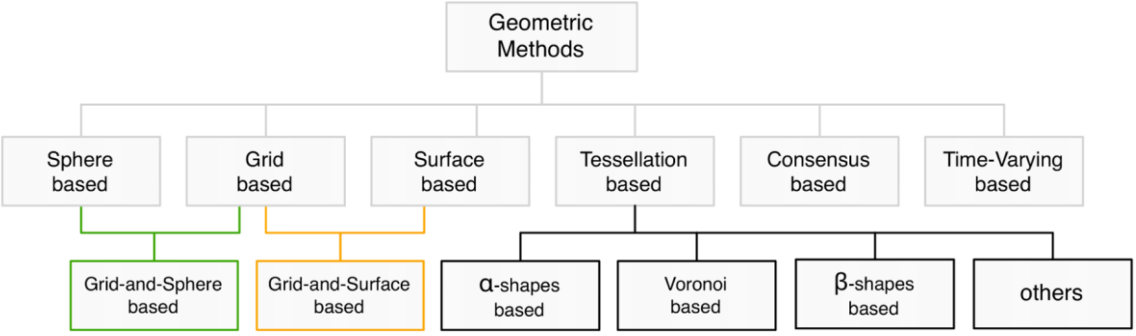


Figure 2. Taxonomy of geometry-based methods (Simões et al., 2017).

Most of the cavity detection algorithms have been developed to analyse static structures like CAVER (Petřek et al., 2006) and MOLE (Sehnal et al., 2013), which are based in Voronoi diagrams; LIGSITE (Hendlich et al., 1997) and PocketPicker (Weisel et al., 2007) are grid-based algorithms and Fpocket (Le Guilloux et al., 2009) is based in alpha-spheres and Voronoi diagrams.

However, as MD simulation are nowadays widely used in bioinformatics, and to capture cavity plasticity, efforts have been made in the development of algorithms to track the geometric evolution of molecular cavities along a MD trajectory. Some examples are EPOS (sphere-based) (*EPOS_BP – Ensemble of Pockets on Protein Surfaces with BALLPass* |, n.d.), Provar (sphere, Voronoi and grid-based) (Ashford et al., 2012), CAVER3.0 (Voronoi-based) (E et al., 2012), POVME3.0 (Grid-based) (Wagner et al., 2017) and MDpocket (Voronoi and grid-based) (Schmidtke et al., 2011).

1.2.1. MDpocket for pocket analysis and characterization

MDpocket is one of the first algorithms designed for protein pocket analysis and characterization of conformation ensemble of structures or a MD trajectory. This program provides a fast, free and open-source record of molecular binding sites, gas migration sites, transient pockets and suitable pockets for molecular docking.

The platform uses three programs: i) Fpocket: for protein cavity identification, ii) Dpocket: for cavity description and iii) Tpocket: for pocket scoring functions.

As MDpocket relies in Fpocket for pocket identification, it is a geometry based algorithm, specifically, it uses Voronoi tessellation, being a Voronoi vertex the center of an alpha-sphere. The radius of each alpha sphere is determined by the difference between the distance of a radius of a Voronoi ball (defined by a Voronoi vertex and the closest atom centers) and the radius of the atom (Figure 3b). These alpha-spheres are in contact and tangential to the surface atoms of the protein space. Finally, the cavity is returned as clusters of alpha spheres (Figure 3c).

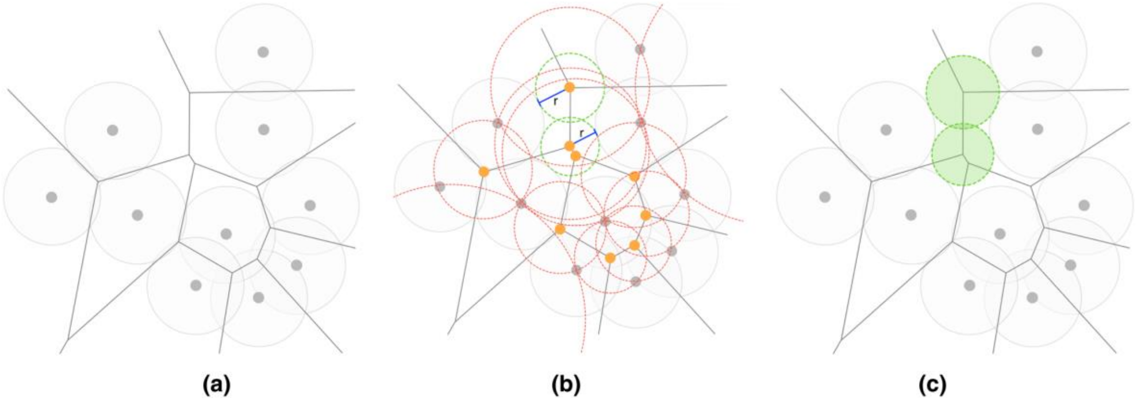


Figure 3. Detection method of Fpocket implemented in MDpocket: a) Diagram of Voronoi tessellation of protein atomic centers (grey points); (b) red circles are Voronoi balls and green circles correspond to alpha-spheres, also centerd at a Voronoi vertex (orange points). Blue line is the tangent line of the alpha-sphere to the surface atoms; (c) fulfilled cavity by clusters of alpha-spheres.

The algorithm is executed for each snapshot of the trajectory, generating normalized density and frequency maps for a structure trajectory, which indicates how many alpha-spheres are filling each cavity and how many times each pocket was open during the trajectory, respectively.

In order to generate the density map (ρ), the algorithm counts the number of alpha-spheres assigned to each point in a predefined spacial grid; and then this number is normalized by the number of snapshots in the trajectory. Meanwhile, in order to calculate the frequency map (φ), the algorithm calculates an occupancy binary parameter for each point in the grid, which is also normalized along the trajectory (Figure 4).

(a) $$\rho_i = \frac{1}{n}\sum_{i=1}^{n} d_{\mathrm{AS},i}$$ (b) $$\Phi_i = \frac{1}{n}\sum_{i=1}^{n} \delta_i$$

Figure 4. Equation for density (a) and frequency (b) map in which n is the number of snapshots, is number of snapshots standardized and δ is the occupancy parameter equals 1 or 0 which value depends is a α-sphere was assigned to that point previously (1) or not (0).

Moreover, a new feature of MDpocket is the characterization of selected pockets by a wide variety of descriptors like pocket volume, polarity score, density of the cavity, mean local hydrophobic density, number of alpha-spheres, etc. More than 20 descriptors can be tracked along the trajectory (further explanation in https://github.com/Discngine/fpocket/blob/master/doc/GETTINGSTARTED.md#pocket-descriptors) . After running MDpocket, a text file is provided with each descriptor per snapshot. In such manner, the software provides a record of the variation of the cavity properties along the trajectory.

1.3. SCIPION: an integrative platform and its applications

Scipion is a software framework developed by the researchers of the National Center of Biotecnology (CNB). It was developed as a platform for 3D electron microscopy (3DEM), and it includes a logical structure for bioinformatics workflows, programs and viewers. It is a free open source code available in Github (https://github.com/I2PC/scipion).

Scipion was firstly developed in the scientific context of addressing the issue of integration, interoperability and tracking of the entire workflow of 3DEM. For this, it provides a graphical user interface (GUI), which includes a workflow editor, allowing the modification and visualization of intermediate and final outcomes with visualization tools. One of the main goals of Scipion is traceability and tracking the entire pipeline, therefore, a database stores each program execution and parameters selected as well as the output created.

Finally, Scipion offers a unified interface throughout the integration of various 3DEM software packages, making 3DEM data processing easier, transparent and reaching a higher scope of users thanks to its user friendly GUI (de la Rosa-Trevín et al., 2016).

1.3.1. Scipion for Virtual Drug Screening

A similar scenario happens for Virtual Drug Screening (VDS), in which multiple software and bioinformatics tools are developed to perform specific tasks. To generate a VDS workflow, it is necessary to execute multiple programs, each of them with their own requirements, which makes the study complex and tedious.

Scipion is currently furthering its application by developing a new branch for chemoinformatics: Scipion-chem. This branch is dedicated to integrate software for VDS and structural biology. Some of the included programs are Rosetta, AutoDock (Trott et al., 2010) and Schrödinger for molecular docking; GROMACS and AMBER for molecular dynamics or P2Rank (Krivák et al., 2018) for pocket identification, among others.

Also, Scipion-chem can perform some more general tasks like downloading molecules files from webpages, file format conversion and protein structure prediction and modification. Regarding the Virtual Drug Screening workflow, some of the new features that are being added involve target and ligand preparation, identification of regions of interest like pocket and sequence of high conservation, and molecular dynamics.

2. OBJECTIVES

In this context, adding software for the analysis of dynamicity of pockets opens a way to new utilities in the platform for the VDS process. The main objectives can be summarized in:

- Use of operative systems and bioinformatics tools for specific pocket and cavity protein analysis in molecular dynamics simulations.

- Integration of MDpocket in Scipion-chem framework.

- Development of a pipeline for a continuous workflow in an integrative virtual drug screening and structural biology platform.

- Biological interpretation and validation of the results with a molecule.

3. MATERIAL Y METHODS

For the plug-in development in Scipion, several protocols and viewers have been created to integrate MDpocket features in Scipion-chem. As MDpocket is a Fpocket extension, the main software package was already included in the Fpocket plug-in (https://github.com/scipion-chem/scipion-chem-fpocket). The plugin is cloned from the git repository of Fpocket (https://github.com/Discngine/fpocket.git), and then all Fpocket package programs, including MDpocket, are automatically installed by Scipion.

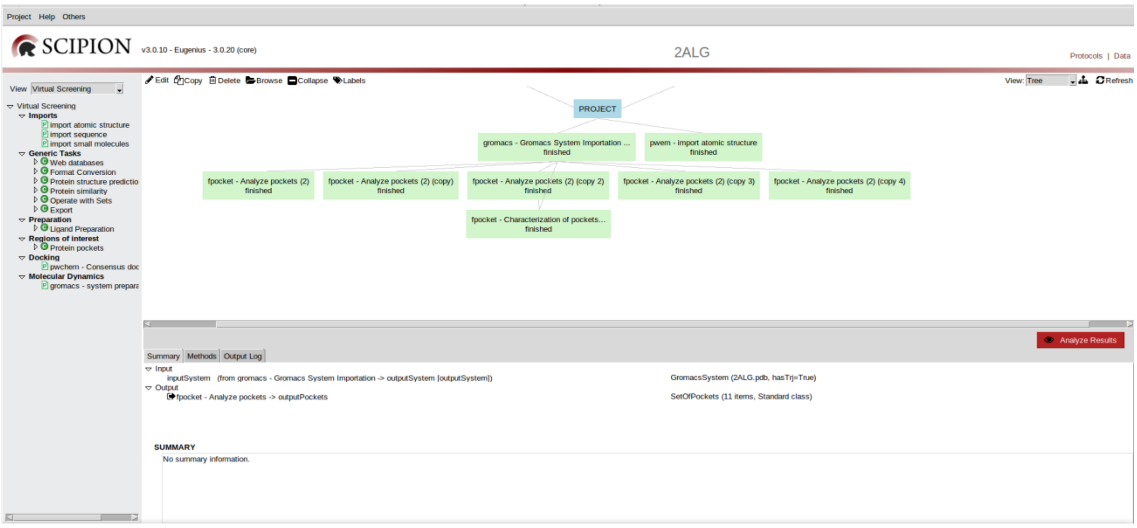
3.1. Scipion's GUI

Scipion main GUI workflow is adaptable to its applications. When executing Scipion, an interface editor is displayed (Figure 5). The left panel has a summary scheme of the type of protocols included. When selecting *View*, multiple branches of Scipion application are available. For “Virtual Drug Screening” selection, the list shows protocols available to perform tasks involved in the VDS process.

When the user wants to perform a specific task with the corresponding protocol, a window pops up to select a set of choices and parameters, characteristic to each software wrapped. For each task performed, the protocols are distributed as boxes in the main window.

By default, a tree view of the workflow is shown to the user. Each green box is a completed protocol, and they are connected by lines according to their input-output relationships. For each protocol, a *Summary* of the tasks performed, the software *Methods* and an *Output Log* are available in the inferior panel. While, in the superior part, protocols can be edited, copied, deleted or a user can browse among all the files generated during the protocol execution.

Finally, an "*Analyze Results*" button is available to display the results obtained in each protocol. This analysis can be a plot, a summary table or a visualization executed by other program.


Project Help Others
SCIPION v3.0.10 - Eugenius - 3.0.20 (core)
2ALG
Protocols | Data
View Virtual Screening
Virtual Screening
Imports
import atomic structure
import sequence
import small molecules
Generic Tasks
Web databases
Format Conversion
Protein structure predictio
Protein similarity
Operate with Sets
Export
Preparation
Ligand Preparation
Regions of interest
Protein pockets
Docking
pwchem - Consensus doc
Molecular Dynamics
gromacs - system prepara
Edit Copy Delete Browse Collapse Labels
View: Tree Refresh
PROJECT
gromacs - Gromacs System Importation ... finished
pwem - import atomic structure finished
fpocket - Analyze pockets (2) finished
fpocket - Analyze pockets (2) (copy) finished
fpocket - Analyze pockets (2) (copy 2) finished
fpocket - Analyze pockets (2) (copy 3) finished
fpocket - Analyze pockets (2) (copy 4) finished
fpocket - Characterization of pockets... finished
Analyze Results
Summary Methods Output Log

Input
inputSystem (from gromacs - Gromacs System Importation -> outputSystem [outputSystem]) GromacsSystem (2ALG.pdb, hasTrj=True)
Output
fpocket - Analyze pockets -> outputPockets SetOfPockets (11 items, Standard class)

SUMMARY
No summary information.

Figure 5. Window of Scipion GUI workflow for virtual drug screening. Project run for MDpocket pipeline finished.

3.2. Scipion Running Mode

Inside Scipion, each algorithm or program is wrapped in a protocol. This protocols have well-defined input and outputs as Scipion data objects, and a set of software-specific parameters.

Internally, these programs are defined as a Python class. The protocol classes in Scipion are structured in five parts: parameter definition, step lists, step function, validation and information functions, and other utile functions.

The parameter definition corresponds to the software-specific parameters that the user will select in the GUI's protocol. Step functions are defined functions by the developer to execute the software or call the external programs, while the utile functions contain additional functions for the correct execution of the program and, in general, conversion of files and formats for homogenization of input/output objects between protocols.

The validation and information functions are to provide useful information to the user and show errors. Finally, all these functions are integrated in the step list function to execute them in order to complete the protocol successfully.

Regarding Scipion's architecture, the main language of development is Python, to stick together all the different software integrated, but also some performance-critical functions of Scipion rely on C++ scripting. Scipion is constituted by different level modules, which interact with each other. One of the lower ones is the Mapper, which stores and retrieves from databases the objects used. Currently, Mapper uses SQLite (http://www.sqlite.org), serving as an internal database and intertwined with the host program ('Introducing SQLite', 2006; de la Rosa-Trevín et al., 2016).

3.3. Protocol Development

MDpocket has two main functionalities: pocket identification and pocket characterization along the MD simulation. For each functionality, a protocol and its respective viewer were created.

3.3.1. Protocol for Pocket Identification

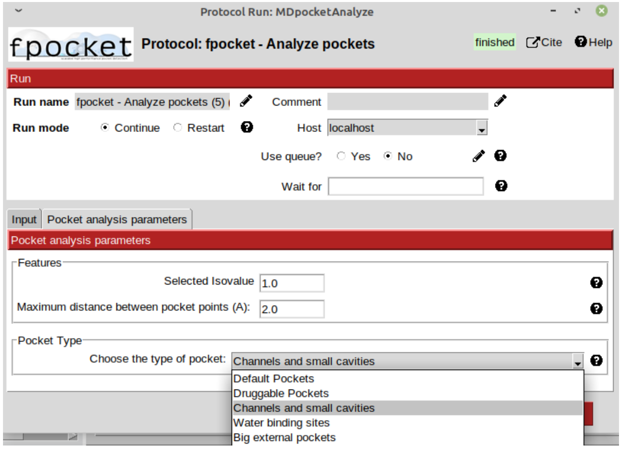
The first protocol which was implemented runs MDpocket using as input the selected files obtained from a MD simulation created with GROMACS or AMBER protocols. The parameters of this protocol are used to seek for specific cavity types; and also to perform additional tasks like creating an extra file with customized features by the user.

When selecting this protocol, a formulary window (Figure 6) is displayed to select parameters:

- **Pocket type**: a list of pocket types is displayed to be selected by the user. This list includes *Default Pockets* (small molecule binding sites), *Druggable Pockets* (high affinity for drug molecules), but also putative small cavities and channels, sterically water binding sites and external big pockets.
  With each selection, additional parameters define the minimum and maximum radius of the alpha-spheres of the pocket and how many of these a pocket must at least contain.
  From the chosen parameters, a command line is built with different flags that determine the behaviour of the algorithm. In consequence, different output files are generated (Table 1).

- **Selected isovalue (N):** isovalue threshold for the final output. This value is a threshold of the number of Voronoi vertices (or center of alpha-spheres) in a grid point for the density map grid file. When selecting this value, as general rule, the more conserved cavities are desired, the higher isovalue. On the contrary, the less conserved and for transient pockets the lower isovalue is introduced. This value is very important as it will determine the number of output pockets, which are provided as a pdb file: mdpoutout-N.pdb (Table 1).

- **Maximum distance pocket points (Å):** maximum distance in angstrom to consider two atoms corresponding to the same pocket cluster in the previous pdb file.
  In order to do so, first the coordinates of the atoms in contact with the pocket are extracted. The distance between each atom position and the closest one is calculated, when this distance is under the threshold, a cluster of atoms is created, in other words, a pocket. Each of these clusterized pockets are retrieved as a pdb file, having as output all the single pockets as independent pdb files needed for each pocket characterization.
  These functions automate and facilitate the pocket selection by the user, which was not performed by MDpocket.


Protocol Run: MDpocketAnalyze

Protocol: fpocket - Analyze pockets

Run

Run name: fpocket - Analyze pockets (5)

Comment:

Run mode: ● Continue ○ Restart

Host: localhost

Use queue? ○ Yes ● No

Wait for:

Pocket analysis parameters

Features

Selected Isovalue: 1.0

Maximum distance between pocket points (A): 2.0

Pocket Type

Choose the type of pocket: Channels and small cavities

Figure 6. Formulary window displayed to run MDpocket identification with the parameters chosen by the user.

3.3.2. *Protocol for Pocket Characterization*

The second protocol involves running MDpocket on its second mode: characterization of pocket descriptors along the MD trajectory. In this second mode MDpocket makes an analysis of the region or cavity of interest from the pockets obtained in the previous step.

By default, MDpocket returns pdb files, with a pre-established isovalue (Table 1), to make this selection manually using a visualization program. This selection can be also done with the density or frequency map. But in both cases, pockets are returned as a unique set, not independent pockets.

As a result, when the user wants to make a selection of a unique pocket all are selected, so it must be done atom by atom, which makes the process tedious, time consuming and difficult for the user.

With the aim to ease this task and maintain the automation in Scipion, a clusterization step is performed, as explained above. This way each pocket is returned as a Scipion object, and used in this protocol to run MDpocket pocket characterization. For instance, the inputs needed are: the MD system, the pdb of the protein and the output of the previous protocol of pocket identification: a set of protein pockets.

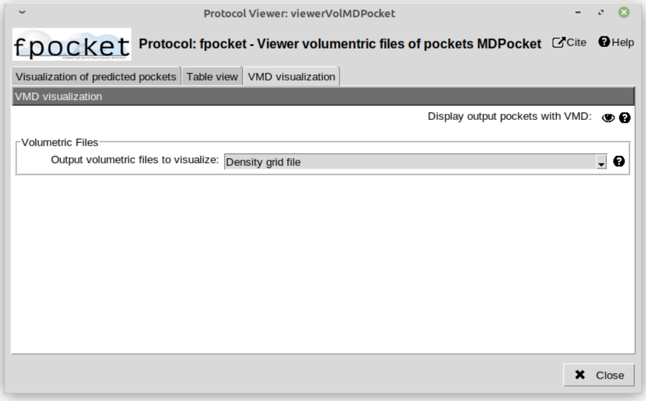
At the end, several files (Table 2) are obtained for plotting and visualization of the results.

3.3.3. Viewer of Volumetric Files and Pockets

When the protocols are finished, the results can be visualized. Several programs for visualization can be used, in this case, Pymol and VMD (Humphrey et al., 1996).

This viewer (Figure 7) corresponds to the results of the first protocol and has 3 tabs with different options:

1) **Visualization of the density and frequency maps.** It is available for the first protocol and both maps can be displayed with VMD. This program permits to change the isovalue (in Graphics> Representations > Isovalue) at the same time the pocket visualization of the maps varies, getting more or less conserved cavities.
2) **Visualization of the predicted pockets** (as Scipion objects): they can be observed with Pymol. When this option is selected, the pockets displayed correspond to the output generated by the isovalue selected in the protocol.
3) **Visualization of the output pockets table: it** displays the pockets with their respective attributes.


Protocol Viewer: viewerVolMDPocket
fpocket
Protocol: fpocket - Viewer volumentric files of pockets MDPocket
Cite
Help
Visualization of predicted pockets
Table view
VMD visualization
VMD visualization
Display output pockets with VMD:
Volumetric Files
Close

Output volumetric files to visualize: Density grid file

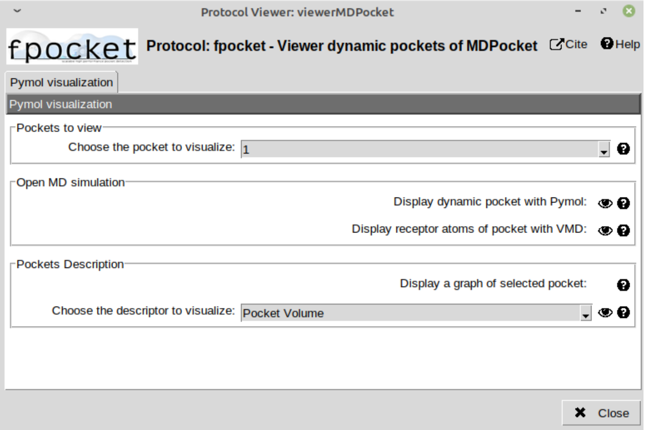
Figure 7. Viewer GUI to visualize results from protocol MDpocket *Pocket Identification.*

3.3.4. Viewer of Dynamic Pockets and Descriptors

A window formulary, like shown in Figure 8, is displayed in order to analyse the results of the *Characterization Protocol.* Several choices are offered to the user:

- **Choose the pocket:** each of the pockets obtained in the Identification Protocol is stored in a folder with its respective output files.
- **Display dynamic pocket with Pymol**: permits the visualization of the selected pocket along the MD trajectory, this way, the user can check how the pocket in the protein varies in position, density and volume along all the frames. A function to take the selected files, and create Pymol file to be executed is made.
- **Display receptor atoms of pockets with VMD:** in this case, the atom residues of the protein that interact sterically with the pocket can be also visualized, to see position and rotation of these atoms along the MD trajectory.
- **Choose the descriptor to visualize:** A variety of descriptors, like pocket volume, polar solvent surface area, number of apha-spheres, polarity, etc, is displayed as a graph of descriptor vs snapshots. When selecting the descriptor to view, a plot is drawn with the variation of the descriptor of the selected pocket along the MD trajectory.

  To plot this graph, a function based in Matplotlib is used, which retrieves the values of each descriptor type from a dictionary. This dictionary is built for each pocket object using the txt file obtained in this protocol. This function access to each folder of the obtained pockets and plots the respective graph.


Protocol Viewer: viewerMDPocket
fpocket
Cite
Help
Pymol visualization
Pymol visualization
Pockets to view
Open MD simulation
Display dynamic pocket with Pymol:
Display receptor atoms of pocket with VMD:
Pockets Description
Display a graph of selected pocket:
Close

Protocol: fpocket - Viewer dynamic pockets of MDPocket

Choose the pocket to visualize: 1

Choose the descriptor to visualize: Pocket Volume

Figure 8. Window of dynamic pockets visualization and plot of descriptors variation.

3.4. Additional Files

MDpocket is a versatile program that produces many output files that may be interesting for the user. For each protocol, a different set of outputs is obtained giving high valuable information to the user (Table 1 and 2).

Table 1. Input and output files for pocket identification with MDpocket

Identification Pockets Protocol		
Input files	Trajectory file	Input of the MD trajectory from Gromacs, AMBER, cpptraj…
	Trajectory format	dcd, xtc, netcdf, crd, crdbox, dtr, trr
	PDB file	Topology of the structure of the protein
Output files	mdpout_freq_grid.dx	Volumetric grid file with a measure of frequency the pocket was open during the MD trajectory
	mdpout_dens_grid.dx	Volumetric grid file with all alpha-spheres around each grid point
	mdpout_dens_iso_8.pdb	File with 3 or more Voronoi Vertices nearby in per snapshot
	mdpout_freq_iso_0_5.pdb	Grid points that are half of the trajectory overlapping with a pocket
	mdpout-1.0.pdb	Pocket file with a number of Voronoi vertices in the grid with a cut-off of the isovalue
	pocketFile.pdb	Pocket file clusterized

Table 2. Input and output files for pocket characterization with MDpocket

Characterization of Pockets Protocol		
Input files	Trajectory file	Input of the MD trajectory from Gromacs, AMBER, cpptraj…
	Trajectory format	dcd,xtc,netcdf,crd,crdbox,dtr,trr
	PDB file	Topology of the structure of the protein
	PDB selected pocket	Pocket file clusterized
Output files	mdpout_mdpocket.pdb	Voronoi vertices of the selected pocket for all the MD trajectory
	mdpout_mdpocket_atoms.pdb	Receptor atoms surrounding the selected pocket
	pocketFile_Modified.pdb	Pocket file standardized
	mdpout_descriptors.txt	All the descriptors of the selected pocket for each snapshot of the MD trajectory

4. RESULTS

In order to validate the correct implementation of the program MDpocket inside Scipion, a validation workflow is performed to detect and characterize the pockets of a well-described and structure known protein: peach protein Pru p3.

Peach Pru p 3 is a non-specific Lipid Transfer Protein (ns-LTPs). These proteins are a family of ubiquitous binding proteins in plants and they are known in medicine as relevant food allergens, especially, in Mediterranean countries (Pasquato et al., 2006).

The ns-LTPs are characterized for being small proteins, approximately 9KDa, and because of the non-specific ligand binding, being capable of carrying a wide type of amphipathic molecules like glycolipids, phospholipids, fatty acids and acyl-coenzyme A.

The structure of these type of proteins is formed by four packed helices and four disulphide bridges, that give high stability to the structure. The final structure of the protein shows a hydrophobic cavity that cross the whole molecule, and where the diverse ligands bind, specifically between the C-Terminal region and the three helixes (Pasquato et al., 2006).

It has been shown that the biochemical characteristics of the cavity along with its flexibility, allow different orientations of the ligands. Also, non-specific van derWaals (VdW) interactions between the protein residues and the ligands give the characteristic property of absence of specificity between ligand and these proteins, having a wide range of ligands.

From another point of view, these proteins have a high level of stability, being very resistant against denaturing agents and processes, like the proteolytic digestion. These properties, makes Pru p 3 a potent food allergen, as it has the capacity to reach the gastrointestinal immune system causing the response of specific IgE formation and to induce a systemic response (Dubiela et al., 2017).

4.1. Scipion workflow methodology

All the analysis was performed using different protocols of the software integrated in Scipion. The workflow can be summed as: I) protein preparation, II) creation of the MD simulation III) Pocket identification and characterization (Figure 9).

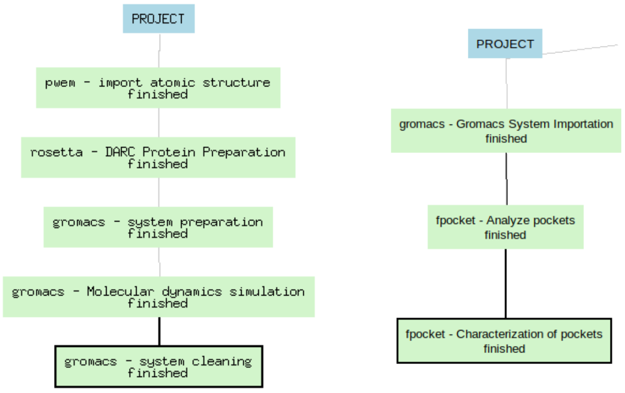

PROJECT
pwem - import atomic structure finished
rosetta - DARC Protein Preparation finished
gromacs - system preparation finished
gromacs - Molecular dynamics simulation finished
gromacs - system cleaning finished
PROJECT
gromacs - Gromacs System Importation finished
fpocket - Analyze pockets finished
fpocket - Characterization of pockets finished

Figure 9. Protocols performed in Scipion to perform the complete workflow for pocket analysis through MD trajectory.

For the protein preparation, the structure of Pru p 3 was retrieved from Protein Data Bank (ID 2ALG) with the protocol of *Import Atomic Structure.* This way, the structure was imported into the workflow just introducing the protein ID. For protein cleaning and chain selection, the protocol of *DARC Protein Preparation*, based in the Rosetta software was used. Heteroatoms like ligands and water molecules that appeared initially in the structure were removed, and selection of chain A of the protein was made.

In order to perform the MD simulation to get the trajectory of the molecule, protocols of GROMACS software were utilized. First, the protocol of *System Preparation* was run. The parameters selected for the Boundary box were by default orthorhombic for buffer distance to box, and buffer for system size type, the box size, as it is for a small protein, was of 1,5 nm. While for the charges, the main force field used was amber03 and tip3p for water force field, and Na+ and Cl- ions were added to the system.

Once the system was prepared, next GROMACS protocol used was *Molecular Dynamics simulation*, to obtain the trajectory. An energy minimization and equilibration was done in this step, to retrieve a 50 ns simulation. Afterwards, a *System Cleaning* protocol was performed to remove the water molecules from the MD trajectory, as it is necessary for MDpocket to run properly, since water molecules would be regarded as part of the system atoms and it would identify pockets between the water molecules of the cage, conducting to an erroneous analysis and not detection of actual the protein pockets.

It is important to mention, that to perform the previous GROMACS simulation the user must have a GPU unit on his computer. On the contrary, there is still the option to analyse MD trajectories with MDpocket by the importation of the files with the *Gromacs System Importation* protocol. Also, this protocol allows to work with other trajectories created outside Scipion, like for example from MoDel database (http://mmb.irbbarcelona.org/MoDEL/) (Meyer et al., 2010), which have more than 2000 MD trajectories that can be retrieved by the PDB ID of the protein.

Finally, a MD simulation of approximately 100.000 snapshots was obtained, to reduce the computational cost and time of MDpocket analysis, this trajectory was subsampled into a 5.000 snapshots one.

First protocol of MDpocket *Analyze pockets* was performed on this trajectory several times with different set of parameters:

- Set 1: Default isovalue of 1, 2Å distance for clustering pockets and default pockets identification.
- Set 2: Isovalue of 0.5, 2Å distance for clustering pockets and default pockets identification.
- Set 3: Isovalue of 3, 2Å distance for clustering pockets and default pockets identification.
- Set 4: Isovalue of 5, 2Å distance for clustering pockets and default pockets identification.
- Set 5: Isovalue of 0.5, 2Å distance for clustering pockets and druggable pockets identification.
- Set 6: Isovalue of 0.5, 2Å distance for clustering pockets and channels and big cavities detection.

To finish with, the pockets obtained identified in this first step were characterized with second protocol of *Characterization of Pockets.*

4.2. Results for Protocol MDpocket Analyse Pockets

From the previous workflow performed, in which different software wrapped in multiple protocols were used, several output files were obtained and analysed through viewers. These viewers display the molecules with visualization programs, like Pymol and VMD.

To start with, from first protocol of MDpocket, the density and frequency maps were obtained when running the program with default parameters and VMD was used to show the maps in Figure 10. Blue pockets correspond to the density map file, while orange ones to the frequency map file of the trajectory. As we can see, much more pockets are obtained in the density map as these are all the pockets identified during all the trajectory, were in contrast, orange pockets are those which were more present along the simulation. We can say that the orange regions correspond to the cavities of the pocket most frequently open and accessible during the trajectory. These regions correspond to the loop region between H1 and H2 and the side between H1 and H3, and third one between C-terminal coil and H4.

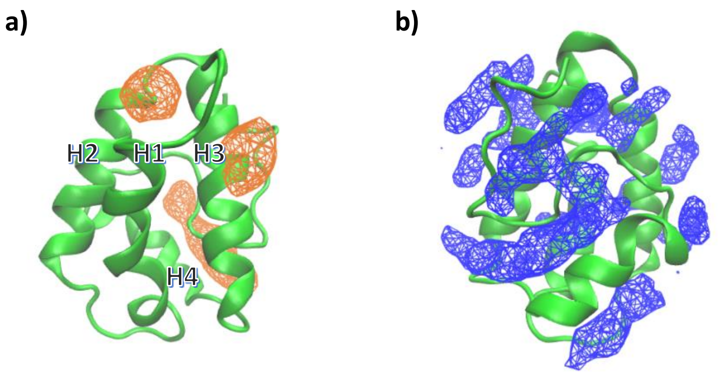


Figure 10. Frequency map (a) and density map (b) outputs with default parameters visualized with VMD. In the frequency maps helix enumeration is indicated.

For an initial exploratory view, these can be raw results to make a later adjustment of the parameters. To have a less permissive identification and not that many irrelevant pockets, it is important to choose a correct isovalue and to study carefully the density map to have more or less conserved cavities. VMD allows to change the isovalue iteratively, so the user can visualize cavity variation in the program.

As previously mentioned, the isovalue is the number of alpha sphere centers per snapshot in a 8 Å^3 cube in each grid point. The higher is this value, the denser or more conserved is this cavity, thereby getting conserved internal channels and pockets. On the contrary, the lower is this value, very superficial or transient binding sites are viewed.

Four sets with default cavity detection and isovalues: 0.5, 1, 3 and 5 were used to find the region of interest in the protein, it was found 16, 18, 10 and 7 pockets respectively (figure 11).

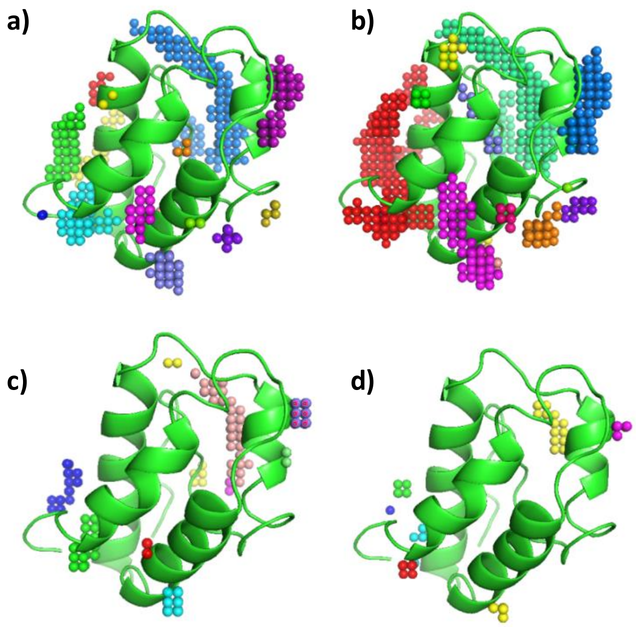


Figure 11. Pymol visualization of output pockets of Pru p 3 with the sets of isovalues 1 (a), 0.5 (b), 3 (c), 4(d) from protocol *Analyse Pockets.* Each colour cluster of spheres corresponds to a pocket.

To select the most convenient isovalue, superposition of density and frequency maps was done with VMD. It was shown that only when using 0.5 isovalue, we could see the inner pocket of interest between the 3 helix and C-Terminal region (red circle in Figure 12). Because of that, this was the isovalue chosen to study the different type of cavities and make the characterization.

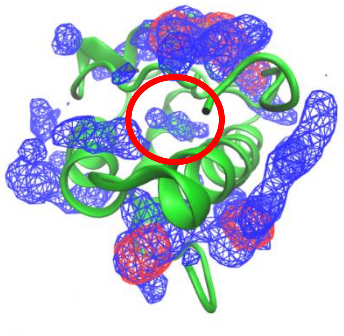


Figure 12. Visualization of overlapping density and frequency maps with VMD. Inner red circle to highlight the binding region of the protein to its ligands.

To study the non-specific affinity to diverse ligands of Pru p 3, two key parameters were chosen: an isovalue of 0,5 (inner cavity was shown) and the pocket type *Druggable pockets* (high-affinity binding sites) in which the exhaustive analysis and pocket characterization was made. Using this set of parameters, six of the total pockets were detected as the most druggable ones.

4.3. Results for Protocol MDpocket Characterize Dynamic Pockets

For each of the six pockets identified, the dynamic variation of the pocket and the atoms with which interact along the MD trajectory were studied with Pymol (Figure 13).

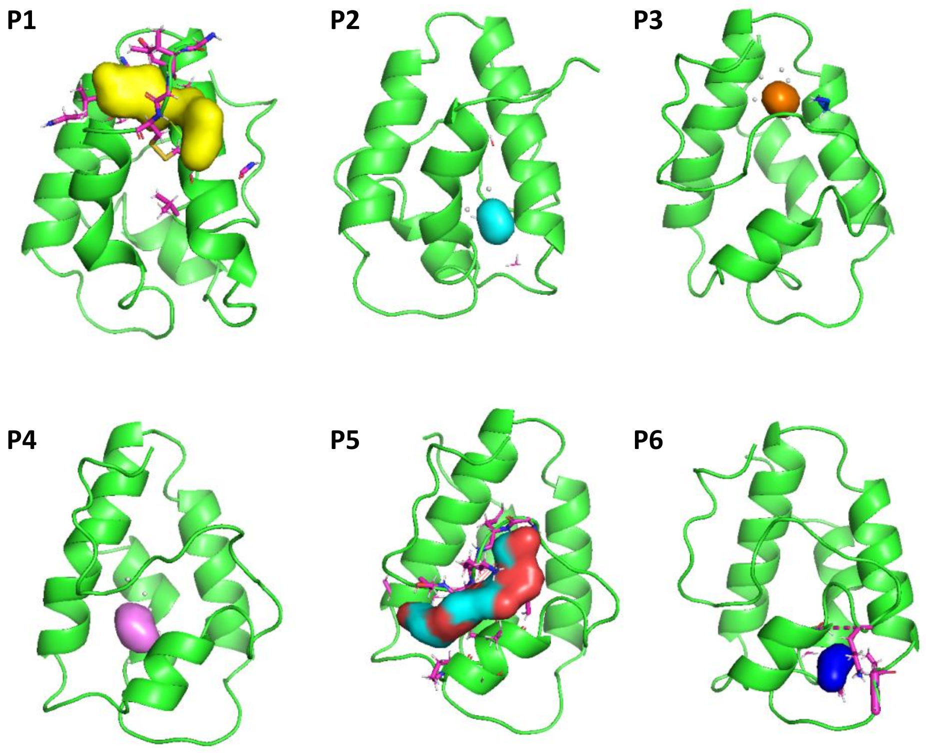


Figure 13. Pymol visualization of each of the six dynamic pockets identified in the previous protocol, with parameter isovalue 0.5 and find *druggable pockets.*

Pockets 1 was identified near the N terminal region of the protein and the helices H2 and H3, interacting with many residues of both substructures. This pocket varied along the trajectory splitting into two subpockets separated by the N-terminal coil. Pocket 2 was identified near the loop between helixes H1 and H2, interacting with few residues of the chains. Moreover, this pocket stayed with a low volume and appeared punctually along most of the trajectory.

Inside these helixes H2, H3 and the C-terminal regions, a very stable pocket appeared along the simulation (Pocket 3). While pocket 4, was identified above the C terminal and the 3 helix, near the center of the molecule (suggesting to be the binding region of the Pru p 3 in which we are interested).

Pocket 5 was the largest one, formed by all the C terminal loop of the protein and the helix H4. This pocket also had high flexibility, subspliting into smaller pockets throughout the trajectory. Finally, pocket 6 was detected between the loops of helix H2 and H3 and the helix 4 and characterized also as a low-stable pocket.

Exploratory results of the pocket descriptors were made for pocket 4 (P4) and 5 (P5) as these seem to be related with the lipid binding region of the protein.

Starting with P5, this pocket experimented changes in its volume and continuity along all the simulation. As it was pre-viewed in the density and frequency maps, this was one of the biggest and more frequently present pocket along the trajectory. In figure 14, we can see how the initial two small pockets bind into a longer one. This makes tunnel that crosses the whole surface of the protein in contact with the long terminal coil of the protein. For a more enriching approach, figure 14b and 14c shows a plot of some relevant descriptors values along the trajectory.

In the plot 14b, we can observe that at the beginning of the trajectory we have the lowest pocket volume values, as visually shows Pymol. However, for the rest of the simulation we have high values of the volume, being the maximum recorded 900Å^3 approximately. On the other hand, the mean local hydrophobic density (14c) seems to be related with the pocket volume and at the end of the trajectory, very high values are recorded. This values are important as this descriptor is correlated with the druggability as it corresponds to local densities of hydrophobic alpha sphere clusters in a cavity.

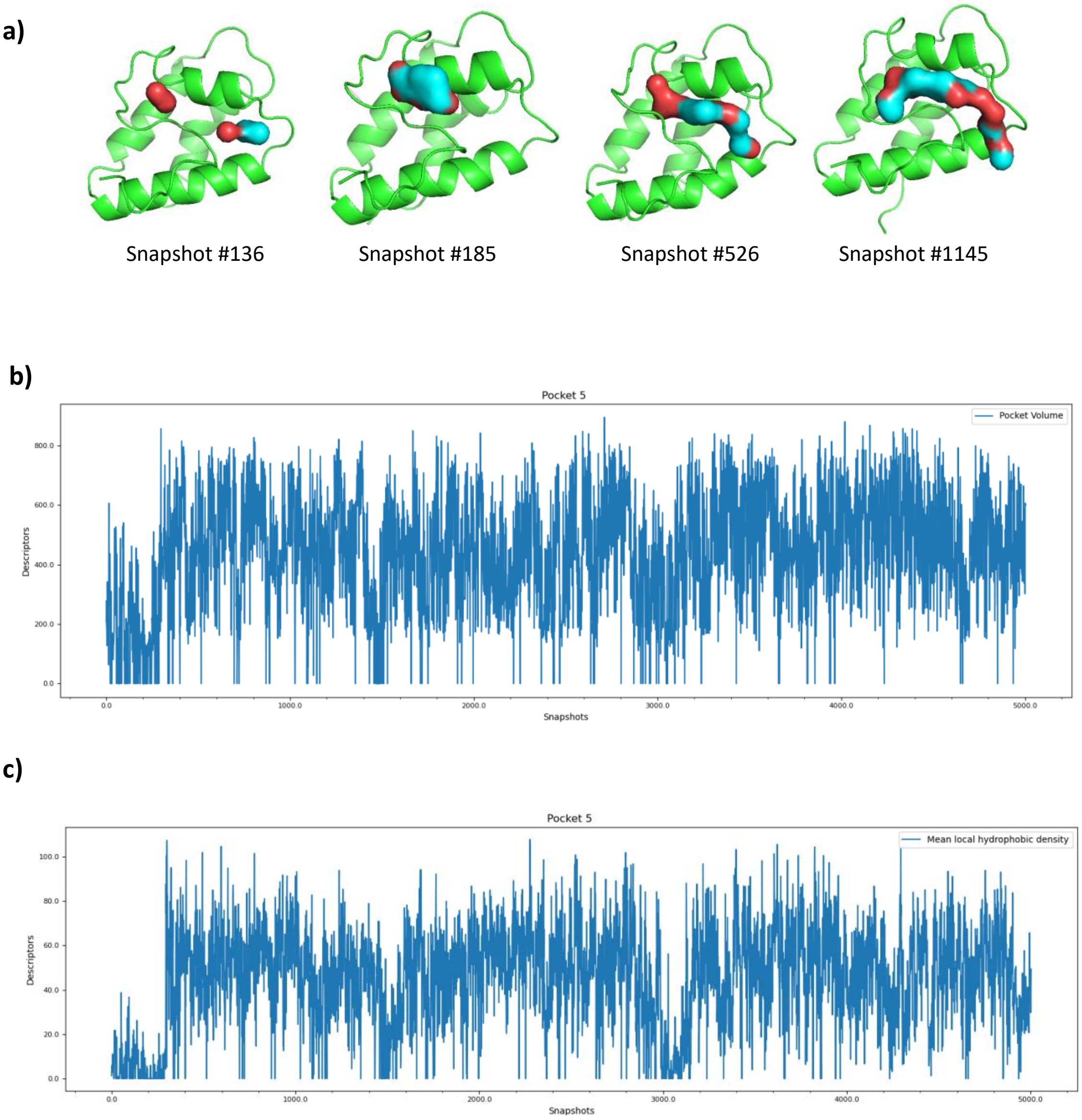


Figure 14. a) Sequence of dynamic pocket P5 variation along the MD trajectory visualized with Pymol; b) Pocket volume (angstrom) variation vs snapshots for the MD simulation; c) Mean local hydrophobic density variation along the trajectory.

For pocket P4, the pocket volume was higher at the beginning of the simulation and less stable in the rest of the trajecoty, disappearing at some points or having a very low volume. On opposition to the previous one, this pocket was very small, with a maximum volume of 170 Å^3 along the MD simulation. Another interesting feature of this pocket, as it was located in the hydrophobic center of the protein was the Hydrophobicity Score, which was very high whenever the pocket was present. This suggests that, with the previous pocket P5, is related with the binding of lipid ligands and other hydrophobic molecules.

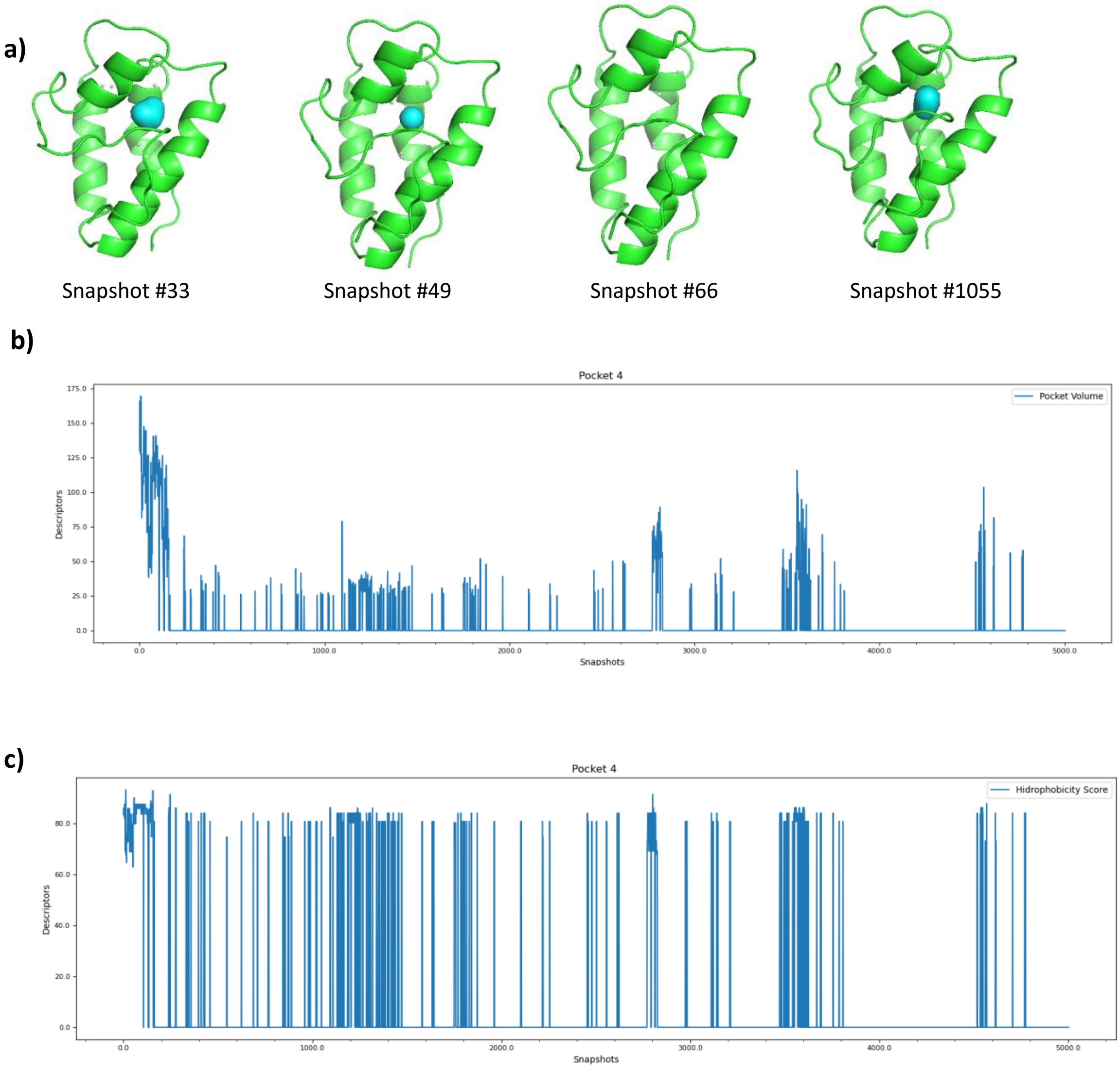


Figure 15. a) Sequence of dynamic pocket P4 variation along the MD trajectory visualized with Pymol; b) Pocket volume (angstrom) variation vs snapshots for the MD simulation; c) Hydrophobicity Score variation along the trajectory.

5. DISCUSSION

From the results obtained above, largest pocket has interesting features that may be related with the binding of Pru p 3 to lipidic ligands. These ligands of pru p 3 bind between the C-terminal loop and the 3 helices, as results show, this pocket (corresponding to P5) is very accessible and present in most of the conformations of the protein.

Also, its high mean local hydrophobicity suggests suitable conditions for the first interaction between Pru p 3 and its ligands. This very accessible and large binding site to its hydrophobic center enables to build unspecific VdW interactions with fatty acids, glycolipids and phospholipids.

In summary, a pocket with the proper physiochemical properties to bind small hydrophilic molecules was found near the hydrophobic center of the protein.

Another interesting pocket detected was pocket 3, as there is experimental evidence that this is the region where IgE forms unspecific interactions to Pru p 3 (Dubiela et al., 2017). It was shown that Pru p 3 epitopes that interact to this immunoglobulin correspond to the loop of helices 2 and 3 region and part of the non-structured C-terminal coil. Also, one of the main residues that participate in this interaction is asparagine (Asn35), which appears in the output as one of the atoms that interact with the pocket detected: P3.

Nevertheless, when running MDpocket with specific parameters to find inner tunnels or cavities, the result was not as expected, and the characteristic tunnel that crosses Pru p 3 was not shown. This suggest that we have to be cautious with our results and selected parameters. Moreover, this may also depend on the MD trajectory, as the reliability of the simulation directly determines the results of MDpocket.

Additionally, it is possible to track the physicochemical descriptors of each pocket along the MD trajectory, as well as, visualize the pocket variation during the simulation. In this context, the selection of a specific snapshot gives us an instant of the protein pocket with the desired conformation during the study. This is highly useful to capture the pocket flexibility to select this region of interest in a conformation of interest for further analysis, for example, for docking studies in VDS.

6. CONCLUSION

In conclusion, MDpocket was successfully integrated in Scipion-chem framework and a full pipeline to explore the pockets and cavities along MD simulations can now be performed.

Also, all the results can be visualized and analysed with Pymol and VMD. MDpocket can perform a qualitative and quantitative analysis, but the key is the set of parameters initially selected which may provide more or less accurate results.

As we have seen, the integration of algorithms that capture protein flexibility and plasticity provides more enriching results when performing any analysis. These simulations try to resemble the physiological conditions where proteins usually are, their study make results more realistic and accurate for cavity prediction.

A validation step was done, comparing experimental and prediction study, and results were very favourable as the binding cavities of ns-LTP Pru P 3 were identified with the MDpocket inside Scipion.

To sum up, it was possible to perform the complete pipeline to study regions of interest in a uniform platform and a simplified way, making the process traceable and more user-friendly to perform *in silico* studies with high quality and a basic knowledge in the field.

7. FUTURE PERSPECTIVE

Regarding the future of pocket and cavity detection along MD simulations, several algorithms have been developed combining conventional methods like grid-based with Machine and Deep Learning methods. These approaches seem to fill a gap in the field with the improvement of the already existing algorithms and development of new software integrating Artificial Intelligence to chemioinformatic classic methods. Moreover, the ability of GPU accelerators, new algorithms and large databases and libraries facilitates the development of other methods to exploit this data (Aggarwal et al., 2021; Karthikeyan & Priyakumar, 2022; Krivák & Hoksza, 2018).

Some of these algorithms are already integrated in Scipion, like P2Rank, that uses machine learning to predict ligand binding cavities. So from this side, Scipion-chem is still on growth but with promising impact in the research and experimental science community by the development of simple frameworks to work with interesting and cutting-edge software.

8. BIBLIOGRAPHY

Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., & Lindahl, E. (2015). GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, *1–2*, 19–25. https://doi.org/10.1016/j.softx.2015.06.001

Aggarwal, R., Gupta, A., Chelur, V., Jawahar, C. V., & Priyakumar, U. D. (2021). DeepPocket: Ligand Binding Site Detection and Segmentation using 3D Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, acs.jcim.1c00799. https://doi.org/10.1021/acs.jcim.1c00799

Arroyo-Mañez, P., Bikiel, D. E., Boechi, L., Capece, L., Di Lella, S., Estrin, D. A., Martí, M. A., Moreno, D. M., Nadra, A. D., & Petruk, A. A. (2011). Protein dynamics and ligand migration interplay as studied by computer simulation. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, *1814*(8), 1054–1064. https://doi.org/10.1016/j.bbapap.2010.08.005

Ashford, P., Moss, D., Alex, A., Yeap, S., Povia, A., Nobeli, I., & Williams, M. (2012). Visualisation of variable binding pockets on protein surfaces by probabilistic analysis of related structure sets. *BMC Bioinformatics*, *13*, 39. https://doi.org/10.1186/1471-2105-13-39

Barros, E. P., Schiffer, J. M., Vorobieva, A., Dou, J., Baker, D., & Amaro, R. E. (2019). Improving the Efficiency of Ligand-Binding Protein Design with Molecular Dynamics Simulations. *Journal of Chemical Theory and Computation*, *15*(10), 5703–5715. https://doi.org/10.1021/acs.jctc.9b00483

Case, D. A., Cheatham, T. E., Darden, T., Gohlke, H., Luo, R., Merz, K. M., Onufriev, A., Simmerling, C., Wang, B., & Woods, R. J. (2005). The AMBER biomolecular simulation programs. *Journal of Computational Chemistry*, *26*(16), 1668–1688. https://doi.org/10.1002/jcc.20290

de la Rosa-Trevín, J. M., Quintana, A., del Cano, L., Zaldívar, A., Foche, I., Gutiérrez, J., Gómez-Blanco, J., Burguet-Castell, J., Cuenca-Alba, J., Abrishami, V., Vargas, J., Otón, J., Sharov, G., Vilas, J. L., Navas, J., Conesa, P., Kazemi, M., Marabini, R., Sorzano, C. O. S., & Carazo, J. M. (2016). Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy. *Journal of Structural Biology*, *195*(1), 93–99. https://doi.org/10.1016/j.jsb.2016.04.010

Discngine/fpocket. (2022). [C]. Discngine. https://github.com/Discngine/fpocket/blob/cd00b961e6ec0f06db50cb1995d762f31a995ce1/doc/GETTINGSTARTED.md (Original work published 2017)

Durrant, J. D., & McCammon, J. A. (2011). Molecular dynamics simulations and drug discovery. *BMC Biology*, *9*(1), 71. https://doi.org/10.1186/1741-7007-9-71

E, C., A, P., P, B., O, S., J, B., B, K., A, G., V, S., M, K., P, M., L, B., J, S., & J, D. (2012). CAVER 3.0: A tool for the analysis of transport pathways in dynamic protein structures. *PLoS Computational Biology*, *8*(10). https://doi.org/10.1371/journal.pcbi.1002708

EPOS_BP – Ensemble of Pockets on Protein Surfaces with BALLPass |. (n.d.). Retrieved 24 June 2022, from https://www-cbi.cs.uni-saarland.de/software/epos_bp-ensemble-of-pockets-on-protein-surfaces-with-ballpass/

Hendlich, M., Rippmann, F., & Barnickel, G. (1997). LIGSITE: Automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, *15*(6), 359–363. https://doi.org/10.1016/S1093-3263(98)00002-3

Introducing SQLite. (2006). In *The Definitive Guide to SQLite* (pp. 1–16). Apress. https://doi.org/10.1007/978-1-4302-0172-4_1

Karthikeyan, A., & Priyakumar, U. D. (2022). Artificial intelligence: Machine learning for chemical sciences. *Journal of Chemical Sciences*, *134*(1), 2. https://doi.org/10.1007/s12039-021-01995-2

Krivák, R., & Hoksza, D. (2018). P2Rank: Machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of Cheminformatics*, *10*(1), 39. https://doi.org/10.1186/s13321-018-0285-8

Krone, M., Falk, M., Rehm, S., Pleiss, J., & Ertl, T. (2011). Interactive Exploration of Protein Cavities. *Computer Graphics Forum*, *30*(3), 673–682. https://doi.org/10.1111/j.1467-8659.2011.01916.x

Le Guilloux, V., Schmidtke, P., & Tuffery, P. (2009). Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, *10*, 168. https://doi.org/10.1186/1471-2105-10-168

Meyer, T., D'Abramo, M., Hospital, A., Rueda, M., Ferrer-Costa, C., Pérez, A., Carrillo, O., Camps, J., Fenollosa, C., Repchevsky, D., Gelpí, J. L., & Orozco, M. (2010). MoDEL (Molecular Dynamics Extended Library): A Database of Atomistic Molecular Dynamics Trajectories. *Structure*, *18*(11), 1399–1409. https://doi.org/10.1016/j.str.2010.07.013

Obst, S., & Stote, R. H. (1998). Comment on Molecular Dynamics Simulations of Zinc Ions in Water Using CHARMM". *Journal of Molecular Modeling*, *4*(4), 145–146. https://doi.org/10.1007/s008940050077

Pasquato, N., Berni, R., Folli, C., Folloni, S., Cianci, M., Pantano, S., Helliwell, J. R., & Zanotti, G. (2006). Crystal Structure of Peach Pru p 3, the Prototypic Member of the Family of Plant Non-specific Lipid Transfer Protein Pan-allergens. *Journal of Molecular Biology*, *356*(3), 684–694. https://doi.org/10.1016/j.jmb.2005.11.063

Perricone, U., Gulotta, M. R., Lombino, J., Parrino, B., Cascioferro, S., Diana, P., Cirrincione, G., & Padova, A. (2018). An overview of recent molecular dynamics applications as medicinal chemistry tools for the undruggable site challenge. *MedChemComm*, *9*(6), 920–936. https://doi.org/10.1039/C8MD00166A

Petřek, M., Otyepka, M., Banáš, P., Košinová, P., Koča, J., & Damborský, J. (2006). CAVER: A new tool to explore routes from protein clefts, pockets and cavities. *BMC Bioinformatics*, *7*(1), 316. https://doi.org/10.1186/1471-2105-7-316

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R. D., Kalé, L., & Schulten, K. (2005). Scalable Molecular Dynamics with NAMD. *Journal of Computational Chemistry*, *26*(16), 1781–1802. https://doi.org/10.1002/jcc.20289

Schmidtke, P., Bidon-Chanal, A., Luque, F. J., & Barril, X. (2011). MDpocket: Open-source cavity detection and characterization on molecular dynamics trajectories. *Bioinformatics*, *27*(23), 3276–3285. https://doi.org/10.1093/bioinformatics/btr550

Schrödinger | Schrödinger is the scientific leader in developing state-of-the-art chemical simulation software for use in pharmaceutical, biotechnology, and materials research. (n.d.). Retrieved 24 June 2022, from https://www.schrodinger.com/

Sehnal, D., Svobodová Vařeková, R., Berka, K., Pravda, L., Navrátilová, V., Banáš, P., Ionescu, C.-M., Otyepka, M., & Koča, J. (2013). MOLE 2.0: Advanced approach for analysis of biomacromolecular channels. *Journal of Cheminformatics*, *5*(1), 39. https://doi.org/10.1186/1758-2946-5-39

Simões, T., Lopes, D., Dias, S., Fernandes, F., Pereira, J., Jorge, J., Bajaj, C., & Gomes, A. (2017). Geometric Detection Algorithms for Cavities on Protein Surfaces in Molecular Graphics: A Survey: Detection Algorithms for Cavities. *Computer Graphics Forum*, *36*(8), 643–683. https://doi.org/10.1111/cgf.13158

Stank, A., Kokh, D. B., Fuller, J. C., & Wade, R. C. (2016). Protein Binding Pocket Dynamics. *Accounts of Chemical Research*, *49*(5), 809–815. https://doi.org/10.1021/acs.accounts.5b00516

Surpeta, B., Sequeiros-Borja, C., & Brezovsky, J. (2020). Dynamics, a Powerful Component of Current and Future in Silico Approaches for Protein Design and Engineering. *International Journal of Molecular Sciences*, *21*(8), 2713. https://doi.org/10.3390/ijms21082713

The Rosetta Software | RosettaCommons. (n.d.). Retrieved 24 June 2022, from https://www.rosettacommons.org/software

Trott, O., & Olson, A. J. (2010). AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading. Journal of Computational Chemistry, 31(2), 455–461. https://doi.org/10.1002/jcc.21334

Wang, Y., Lupala, C. S., Liu, H., & Lin, X. (2019). Identification of Drug Binding Sites and Action Mechanisms with Molecular Dynamics Simulations. *Current Topics in*

Medicinal Chemistry, *18*(27), 2268–2277. https://doi.org/10.2174/1568026619666181212102856

Weisel, M., Proschak, E., & Schneider, G. (2007). PocketPicker: Analysis of ligand binding-sites with shape descriptors. *Chemistry Central Journal*, *1*(1), 7. https://doi.org/10.1186/1752-153X-1-7