SoftwareX 12 (2020) 100606

Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

ParKVFinder: A thread-level parallel approach in biomolecular cavity detection

João Victor da Silva Guerra [a,b], Helder Veras Ribeiro Filho [a], Leandro Oliveira Bortot [a], Rodrigo Vargas Honorato [a,c], José Geraldo de Carvalho Pereira [a], Paulo Sérgio Lopes-de-Oliveira [a,b,*]

[a] Brazilian Biosciences National Laboratory (LNBio), Brazilian Center for Research in Energy and Materials (CNPEM), Campinas 13083-100, SP, Brazil
[b] Graduate Program in Biosciences and Technology of Bioactive Products, Institute of Biology, University of Campinas, Campinas 13083-862, SP, Brazil
[c] Faculty of Science–Chemistry, Bijvoet Center for Biomolecular Research, Utrecht University, Padualaan 8, 3584 CH, Utrecht, The Netherlands

ARTICLE INFO

Article history:
Received 1 April 2020
Received in revised form 21 July 2020
Accepted 6 October 2020

Keywords:
Cavity detection
Parallelization
Space segmentation
Spatial characterization
OpenMP

ABSTRACT

Biological processes are regulated mainly by the binding of small molecules into cavities distributed throughout the biomolecular structure. Computational tools to detect these cavities have an essential role in rational drug design. With the exponential availability of high-order 3D atomic structures and large sets of atomic models, the tools must balance accuracy and speed. In this sense, we developed parKVFinder, a parallelized software for geometry-based cavity detection. Here, we described its functionalities and presented its easy-to-use PyMOL plugin and command-line interface. Finally, we demonstrated parKVFinder by identifying an important HIV-1 protease cavity and compared it with other geometry-based programs.

Code metadata

Current code version	parKVFinder 1.0
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_153
Code Ocean compute capsule	None
Legal Code License	GNU General Public License v3.0
Code versioning system used	git
Software code languages, tools, and services used	C, OpenMP, python, Tk, PyMOL, TOML, FUTURE
Compilation requirements, operating environments & dependencies	Linux, OS X; GCC 6.5 (or later)
If available Link to developer documentation/manual	https://github.com/LBC-LNBio/parKVFinder/wiki
Support email for questions	paulo.oliveira@lnbio.cnpem.br

1. Motivation and significance

Biological processes are mostly regulated by the interaction between biomolecules, such as proteins and nucleic acids, and small molecules (ligands). To interact with these macromolecules, ligands generally bind into specific binding sites formed by solvent-exposed clefts or even into buried cavities that have proper physicochemical and geometrical characteristics to accommodate them [1,2]. Thus, the prospection of any type of

* Corresponding author at: Brazilian Biosciences National Laboratory (LNBio), Brazilian Center for Research in Energy and Materials (CNPEM), Campinas 13083-100, SP, Brazil.
E-mail addresses: joao.guerra@lnbio.cnpem.br (J.V.d.S. Guerra), helder.veras@lnbio.cnpem.br (H.V. Ribeiro Filho), leandro.bortot@lnbio.cnpem.br (L.O. Bortot), r.vargashonorato@uu.nl (R.V. Honorato), jose.pereira@lnbio.cnpem.br (J.G.d.C. Pereira), paulo.oliveira@lnbio.cnpem.br (P.S. Lopes-de-Oliveira).

https://doi.org/10.1016/j.softx.2020.100606

cavities allows the identification and characterization of putative ligand-binding sites, which play an essential role in the pipeline of rational drug discovery and design.

Given their low cost, speed and automation, *in silico* approaches are widely applied to detect cavities in 3D atomic biomolecular structures. In this context, three main approaches of computational algorithms are commonly applied: evolutionary-, energy- and geometry-based algorithms [1,3,4], and each has its own advantages and disadvantages. We previously developed KVFinder, a geometrical grid-and-sphere based method, which has advantages over evolutionary- and energy-based methods [5]. Since then, our software has been successfully used to prospect and describe ligand-binding sites, evaluating the shape and quantifying the volume of protein cavities [6–13].

In silico identification of cavities is based on 3D atomic structures which can be solved by X-ray crystallography, nuclear magnetic resonance or electron cryomicroscopy (cryo-EM). Advances in cryo-EM have increased the number of high-order biomolecular structures, composed of many domains or even entire biological particles [14–16]. Further, the development of computing resources has dramatically expanded the use of molecular dynamics (MD) simulation methods to understand the dynamic behavior of biomolecules in full atomic detail [17]. Together, these advances produce a large set of atomic models that can yield outstanding information when computationally analyzed.

In this scenario, computational biology tools have been adapting to the increasing amounts of structural data by using optimized routines. To cover this need, we developed parKVFinder, a parallelized software for biomolecular cavity detection. This new tool maintains the accurate grid-and-sphere based detection method used in the original implementation of KVFinder with faster subroutines guaranteed by thread-level parallelism. ParKVFinder is available as a new user-friendly PyMOL plugin with an intuitive graphical user interface (GUI) that allows users to configure custom parameters for cavity detection, result analysis and visualization. Advanced users can take full advantage of the tool via a new command-line interface (CLI), which can be incorporated into drug discovery and docking pipelines. Regarding KVFinder, the following new features were implemented: boundary definition, surface area estimation, and a new routine for the identification of residues that form the cavities. Here, we demonstrate parKVFinder by identifying an important cavity of the Human Immunodeficiency Virus type I (HIV-1) protease and estimating its volume over an MD simulation trajectory. Finally, we compared parKVFinder results and performance with other geometry-based programs.

2. Related work

Geometric approaches for cavity detection have been vastly employed, which has a wide range of different techniques [3,18]. Since these techniques are relatively simple, straightforward and do not rely on prior knowledge, the approach is the most recurrent in the literature [1,5]. Here, we briefly present a comprehensive classification on geometry-based cavity detection approaches, whose taxonomy include grid-, sphere-, tessellation- and surface-based techniques, and combinations of those [3,18].

Grid-based algorithms rely on a set of atoms, usually simplified by a hard sphere model, inserted in an axis-aligned 3D grid and a density map, i.e. a scalar field, in which each discrete point is an integer or a boolean value, that is used to collect empty voxels into cavities through a voxel clustering algorithm [2,3,18]. Based on this, it is possible to identify cavities in an automated procedure and represent a collection of data in a discrete point [1,3]; however, the main limitations are the grid-spacing and protein-orientation sensitivities [3,18]. Sphere-based algorithms employ a set of atoms, sometimes also simplified by a hard sphere model, and approximate a ligand by a single hard sphere, called probe, or even a full hard sphere model to identify cavities on the protein surface [2,18]. This technique brings the spatial extent of possible ligands to the cavity detection procedure and also detects any type of cavity. The main deficiency consists of finding and delineating cavity boundaries and mouth opening, without ambiguity [3,18]. Tessellation-based algorithms, based on the field of computational geometry, includes α-shapes, β-shapes and Voronoi-based methods; however, these subfamilies develop from the theory of α-shapes [3]. These methods overcome the dependency of accuracy and memory requirements on grid spacing and are accurate to identify cavities [18]; however, they do not directly rely on any molecular surface information. Specifically, while the α-shapes and Voronoi-based methods depends on atomic centers and constant-radius spheres, the β-shapes methods rely on varying-radius spheres to represent atoms. In addition, other limitations include the accuracy of not only identifying the correct location of the binding site, proper delineation of the cavity boundary, and number of surface atoms [3,18]. Surface-based algorithms depend on the analysis of the molecular surface model, such as van der Waals, solvent accessible, solvent excluded, ligand excluded surfaces, in contrast to the hard sphere model of the other algorithms [18]. With these surfaces defining the molecular interface and its surroundings, the cavities are defined with respect to a specific solvent or ligand, i.e., cavities accessible by this size of molecules [3,18]. These algorithms operate in an automated manner, as grid-based methods, and usually do not suffer from mouth opening ambiguity; however, these methods usually are not capable of identifying any type of cavity and sometimes not the entire extent of them [3].

In general, the combination of such techniques aims to capture the capabilities and correct or circumvent their individual deficiencies in order to obtain a more powerful technique, such as grid-and-sphere-based methods that are not orientation-sensitive as the grid-based ones [3]. Therefore, each technique has its own capabilities and deficiencies, which make each method best suited for certain applications.

3. Software description

Parallel KVFinder (parKVFinder) is open-source (GPL 3.0) software designed for detection and spatial characterization of any type of biomolecular cavity. ParKVFinder inserts the target biomolecule in a 3D grid divided by regular voxels and applies a dual-probe algorithm based on the theory of mathematical morphology [19,20].

3.1. Software architecture

ParKVFinder is written in C language and its PyMOL plugin GUI is written in Python 3. The code is tested in GNU/Linux x86-64 and macOS operating systems. A schematic diagram of parKVFinder operation is described in Fig. 1.

ParKVFinder can be executed via GUI or CLI, and users have access to a full set of customizable parameters, which are saved as a TOML file format (Fig. A.1). The cavity detection relies on a dual-probe system, a smaller sphere called Probe In, and a larger sphere called Probe Out, in which each probe scans the macromolecule structure by translating it over 3D-grid points to create a molecular surface. Explored grid points are limited by the van der Waals (vdW) radius of the macromolecule's atoms. The dual-probe system defines two molecular surfaces with different levels of accessibility, the cavity is defined as the region where there is no overlap between the surfaces. For the surface

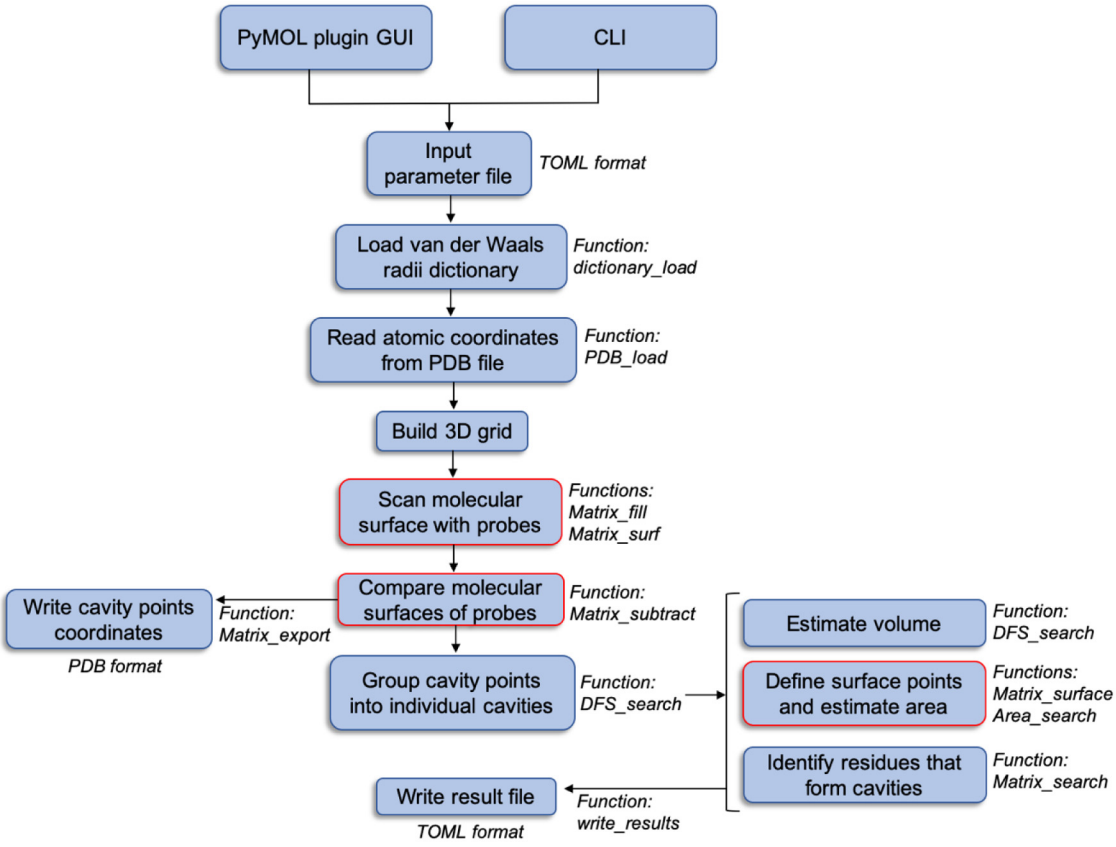

Fig. 1. Diagram of parKVFinder architecture. The flowchart presents parKVFinder execution steps from parameter file input until results files output. Main functions executed in each step are mentioned and boxes highlighted in red indicate steps with parallel subroutines.

definition, parKVFinder uses three main functions: *Matrix_fill* that fills the 3D grid with the probes In and Out, Matrix_surf that smooths the molecular surface generated by probes In and Out, and *Matrix_subtract* that compares the molecular surfaces of the probes to define the cavity points. Afterwards, a recursive depth-first search (DFS) algorithm is applied to connect points belonging to the same cavity, as described in [5], through the function *DFS_search*. At the same time, the volume for each cavity is estimated and a cavity PDB file is written by the *Matrix_export* function. For each cavity, parKVFinder identifies its surface points through a spatial filter, using the *Matrix_surface* function and then the surface area is estimated, using the *Area_search* function. Further, amino acids forming each cavity are determined by the function *Matrix_search*. All these spatial descriptors are written in a TOML format file (Fig. A.2).

ParKVFinder subroutines were submitted to extensive parallelization to improve their performance and reduce the computational time to prospect cavities. Alongside addressing all parallelizable functions, we focused the parallelization effort on two main functions that together consumed 70% of the total runtime in the previous version of the software: *Matrix_surf* and *Matrix_subtract*. A substantial improvement of performance has been achieved through the implementation of multiple threads, with OpenMP API creating a predefined amount of parallel threads, in which the 3D grid is subdivided into subsets and distributed among these threads. In addition, a pool of tasks scheme has been applied to ensure a balanced workload and better use of hardware resources.

3.2. Implementation details

ParKVFinder loads a customizable dictionary of vdW for each atom type in its respective residue, in which the default dictionary is the *parKVFinder/dictionary* file, using *dictionary_load* function. This dictionary is loaded in an array of singly linked list structures, which each index in the array is an integer corresponding to the residue three-letter code (e.g. ALA, GLY, VAL, etc.) of the *dictionary* file. Each node has a single head pointer, a string for the atom name and a float for its radius. Further, the software reads the atomic coordinates of the target structure into another singly linked list structure from an input PDB file, using the *PDB_load* function. Each node has a single head pointer, an integer for the residue number, characters for its chain and one-letter residue code (e.g. A, G, V, etc.), and floats for the atomic coordinates (x, y, z) and its radius. For both linked list structures, all nodes are allocated in the head end.

The 3D-grid is constructed based on the target structure dimensions and the Probe Out diameter, which uses a dynamic contiguous three-dimensional integer array. Each position represents a single point in space, confined within the boundaries of the grid and equidistant from each other by a grid spacing (h), which is defined by the user as an input parameter. In addition, each integer value of the grid corresponds to an occupied space (zero; i.e. target structure), an empty space (one) or a cavity space (non-zero integers; i.e. cavity identifiers). All parallelizable matrix operations (e.g. *Matrix_surf*, *Matrix_subtract*, *Matrix_filter*, *Area_search*) take advantage of the data independence between each grid position and each operation inside the grid. Based on this, the 3D-grid is divided in the predefined amount of parallel threads and each piece of data is allocated on a free node in the CPU to execute the matrix operations on it, which workload is

managed by a pool of tasks that dynamically allocate the data. Furthermore, steps of some matrix operations (e.g. *Matrix_filter*) are also independent and these steps occur concurrently.

Finally, the spatial characterization (i.e. volume, area and interface residues) are stored into a dynamic array with a struct for each identified cavity, which consists of floats for volume and surface area, and a singly linked list structure for the interface residues information, i.e. an integer for residue number and characters for its chain and one-letter residue code.

3.3. Software functionalities

3.3.1. Spatial characterization

ParKVFinder performs a spatial characterization, including volume, surface area, shape and interface residues that compose the cavities. For that, voxels marked with an integer identifier different from zero, for each cavity detected, are used. Volume is estimated as:

$$\widehat{V}_i = N_i \cdot h^3 \tag{1}$$

where $\widehat{V}_i$ is the volume of the cavity i, N_i is the number of voxels belonging to the cavity i and h is the grid spacing.

The definition of cavity surface points is essential for the assessment of shape and surface area. A spatial filter defines surface points as cavity points where at least one direct neighbor is a biomolecule point (Fig. A.3). The surface area estimation is based on the classification of surface voxels into six classes, as proposed by [21], and is calculated as:

$$\widehat{S}_i = \sum_{j=1}^{6} W_j \cdot N_{S,j,i} \cdot h^2 \tag{2}$$

where $\widehat{S}_i$ is the surface area of the cavity i, W_j is the weight of the voxel class j, and $N_{S,j,i}$ is the number of surface voxels j belonging to the cavity i. The weights are shown in Table A1.

The identification of interface residues that form the detected cavities provides a description of the residue composition and is based on the search for cavity points within a radius of each atom of the biomolecule. The search radius is the sum of the Probe In diameter and the vdW radius of the target atom. When an atom encounters a cavity point within the search radius, the residue of that atom is identified as an interface residue for that cavity.

3.3.2. Cavity boundary

A common problem in biomolecular cavities detection is the boundary definition [3]. To handle this issue, ParKVFinder features a new customizable parameter, called Removal Distance, which removes cavity points within a given distance from the cavity-bulk frontier. The decrease of this parameter makes it possible to detect more superficial cavities; however, the increase helps to segregate sub-pockets or even different cavities (Fig. 2). A schematic representation of the Removal Distance routine is shown in Fig. A.4.

3.3.3. Space segmentation

ParKVFinder also improves on the previous space-segmentation feature by allowing it over a large dataset, in a high-throughput friendly manner, with a single configuration file. The main issues concerning space segmentation are its resolution and subsequent memory overflow. The search space can be adjusted for a custom search box or a target ligand, or kept throughout the whole biomolecular structure. With this feature, the user can divide cavities into its sub-pockets and focus the analysis on a region of interest, such as a complex binding site, with detailed spatial definitions and descriptions. Furthermore, restricting the search space around a given area of interest achieves a higher resolution

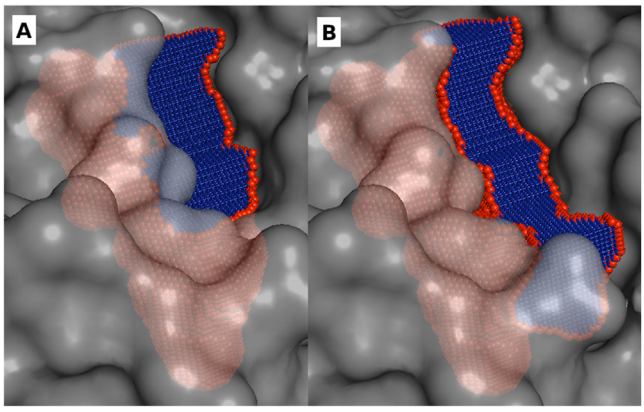

Fig. 2. Definition of cavity boundary. (A) Detection of the adenosine cavity on protein kinase A structure (PDB ID: 1FMO), using a 2.4 Å Removal Distance. (B) Same detection with 1.2 Å Removal Distance. Red dots represent surface cavity points and blue dots represent the remaining cavity points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

representation with a low computational cost. Custom search space parameters can be now defined via the CLI, with configuration files or explicitly setting it in the TOML parameters file, or interactively via the GUI as in KVFinder.

3.3.4. Usability features

The user experience provided to the user by other cavity detection software is often overlooked, we took extra steps to ensure the usability of our tool for both beginners and experts. ParKVFinder employs a fully customizable set of parameters that allows highly versatile cavity detection and spatial characterization, provided by a TOML parameters file with a simple human-readable key–value system. We also provide the users with the option of using a qualitative definition of the grid spacing by predefining values as the Resolution parameter: 0.25 Å for High, 0.5 Å for Medium and 0.6 Å for Low. Alongside these customizable parameters, parKVFinder presents a user-friendly GUI, available as PyMOL plugin (Fig. 3), and CLI, for Unix and macOS operating systems. Despite the same available options on both interfaces, the CLI is well-suited for fast, multi-model or MD analysis. A more detailed description of PyMOL plugin GUI and CLI is provided in the Supplementary Material.

4. Illustrative example

We demonstrate the use of parKVFinder to describe the conformational dynamics of a cavity that defines the active site of the HIV-1 protease, which is an effective therapeutic target. The HIV-1 protease catalytic cycle involves movements of β-hairpins, called "flaps", which control the accessibility of substrates to the active site of the homodimer [3,22]. While crystallographic structures show the flaps in the closed state [23] and in the semi-open state [24], MD simulations have investigated the dynamics of the opening flaps [22,25]. We use parKVFinder CLI to describe, for the first time, the movements of the HIV-1 protease flaps using the active site cavity volume as a conformational descriptor during MD starting from the closed state. The detailed simulation protocol is described in the Supplementary Material.

The cavity volume starts at the level that corresponds to the closed conformation. After ∼25 ns, the volume starts to increase and, at ∼75 ns, the volume reaches the level of the semi-open state, indicating that the cavity is opening during the simulation. After the 75 ns mark point, the flaps get even further apart,

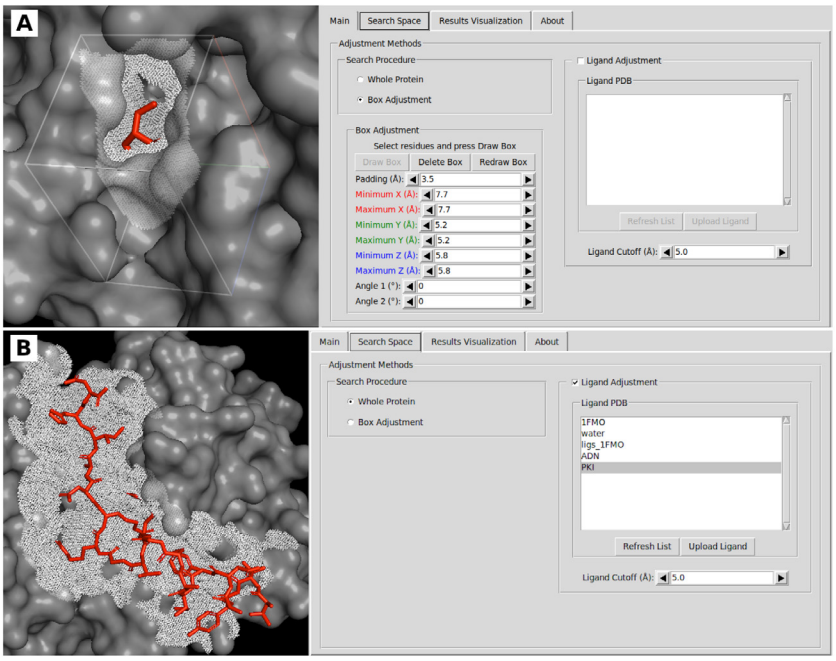

Fig. 3. PyMOL parKVFinder Tools GUI. (A) Box adjustment mode. Cavity detection set inside an interactive custom box around an adenosine ligand of a protein kinase A (PDB ID: 1FMO). (B) Ligand adjustment mode. Cavity detection set a 5 Å radius around a PKI molecule (PyMOL object) of a protein kinase A (PDB ID: 1FMO), with Removal Distance changed to 0 Å to detect more superficial points.

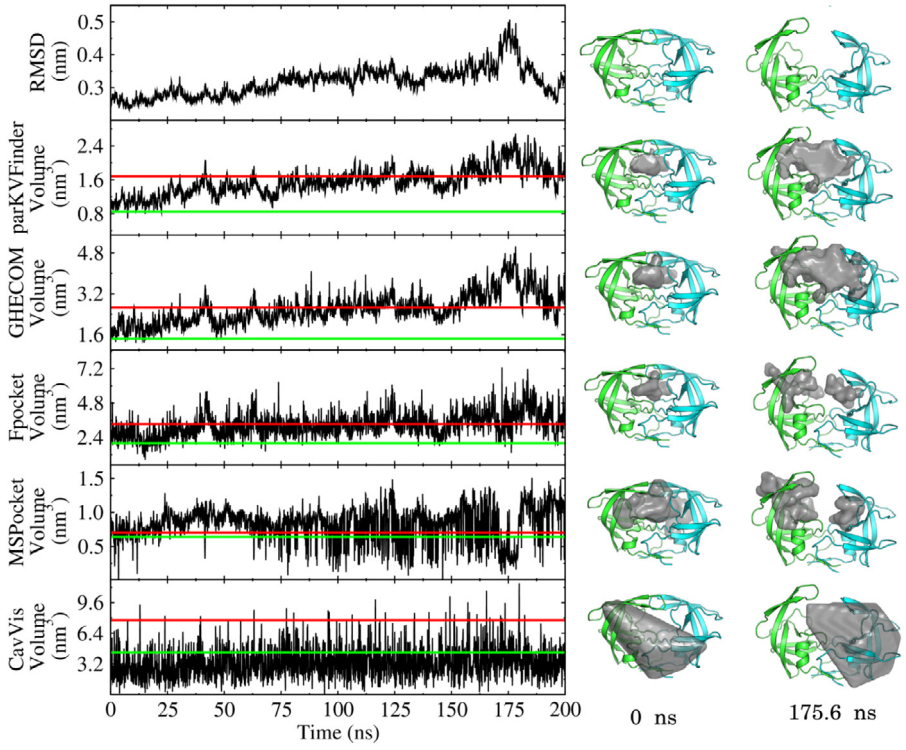

Fig. 4. Describing the conformational dynamics of the HIV-1 protease with the volume of its active site along a 200 ns simulation. The green and red lines indicate the cavity volume for the closed (PDB ID: 1HVR) and semi-open (PDB ID: 1HHP) states, respectively. The structures of the protein at the beginning of the simulation (0 ns) and at the frame which has the highest RMSD (175.6 ns) are shown as cartoons. The corresponding cavities detected by each software are shown as gray surfaces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with the cavity reaching its maximum volume at ∼175 ns before reverting to the more stable semi-open state (Fig. 4).

This result was compared to the Cα RMSD calculated using the closed state as reference. The cavity volume profile obtained with parKVFinder roughly correlates to the RMSD profile (r=0.72), indicating that we are correctly describing the conformational state of the protein along the simulation. We emphasize that, although the RMSD is a global metric that describes global variations in the protease structure, the volume estimated by parKVFinder

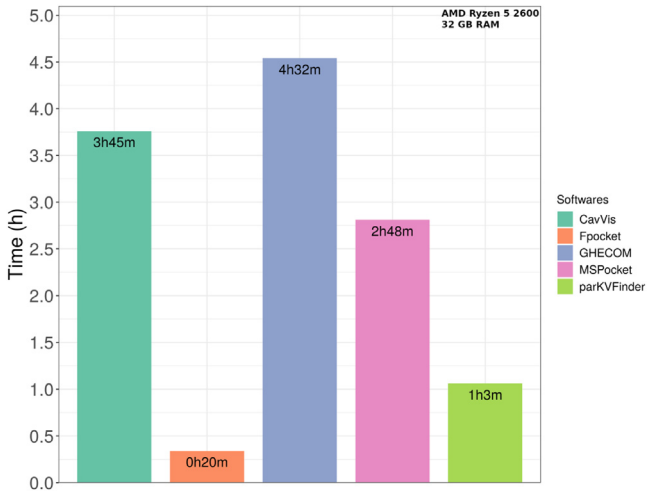

Fig. 5. Time performance of the benchmarking methods.

provides a direct measure of changes in the active site volume along the trajectory that may be directly related to the ligand accessibility.

The same analysis was repeated using standalone tools with volume characterization capabilities: CavVis [4], MSPocket [26], GHECOM [27] and Fpocket [28]. All the parameters and software versions are described in the Supplementary Material. Cavity volume estimated by GHECOM is also correlated to the conformational state of the protease during the simulation similarly to parKVFinder (r=0.75; Fig. 4), which may be due to both employ a grid-and-sphere based method. Cavities found by CavVis, MSPocket and Fpocket did not accurately correlate to the conformational dynamics of the protease binding pocket during the MD trajectory (r=0.19, −0.24 and 0.35, respectively). Thus, parKVFinder and GHECOM presented higher accuracy in describing the conformational dynamics of the active site of HIV-1 protease during the MD simulation.

Despite being able to accurately detect macromolecular cavities, current methods must also be able to perform fast detection and characterization, when dealing with a large set of models. We also evaluated the computational time used by these methods (Fig. 5). ParKVFinder and GHECOM, which presented the best accuracy in the study case, required quite different computational time. ParKVFinder performed at least four times faster than GHECOM and despite both being based on the same method, parKVFinder multithreading subroutines dramatically improved its performance. ParKVFinder speed also outperformed all other tools evaluated, except for Fpocket which ran three times faster. Fpocket uses a Voronoi tessellation and alpha spheres method, which is a fast method to be computed and, in principle, have higher geometrical accuracy than grid-based approaches; however, these methods has some limitations, including identification of the correct location of the binding site, boundary definition and volume and area calculations [3]. In our case study, Fpocket application of it was not as sensitive as the grid-based methods to finely describe cavities, e.g. shape description and calculations of volume and area. Despite its rapid cavity detection and volume characterization, it fails to differentiate the conformational states of the HIV-1 protease ligand-binding site. Taking accuracy and time performance together, parKVFinder has outperformed the benchmarking methods in the practical example of HIV-1 protease, presenting the robustness of its cavity detection, spatial characterization, and "fine-grained" parallelization procedures.

Furthermore, parKVFinder performance can be scaled up with more threads available in the CPU, making the tool HPC/HPA friendly and adequate for integration in large scale drug discovery pipelines (Fig. A.7). In addition, we evaluated the scalability of our software by executing it with a dataset, composed of a thousand unique protein domains, described in the Supplementary Material. The computational time against the number of atoms are shown in Fig. 6, in which we show the linear regression for different numbers of threads and the coefficient of determination (R^2) to address the quality of these linear adjustments. Based on Fig. 6, we observe that the slope reduces with the greater number of threads applied in parKVFinder, indicating that the time required to process each atom is also decreasing; hence, the performance is increasing with more available threads. In addition, the coefficient of determination is increasing with more threads available, showing that the behavior of the time-atoms relationship is becoming more linear with more threads.

5. Impact

The identification of ligand-binding sites in biomolecules is the main purpose of several research groups worldwide and is coveted by pharmaceutical industries since the detection of novel sites can lead to the structure-based development of new drugs. Ligand-binding sites can be cavities spatially distributed throughout the macromolecule structure and generally are located in shallow solvent-exposed regions or in void regions buried inside the macromolecule [3,29]. To prospect them, some software based on geometric approaches, such as KVFinder, CavVis, MSPocket, GHECOM, and Fpocket have been developed.

ParKVFinder is an open-source program designed to perform a fast detection of cavities in high-order macromolecules structures or a large set of them. The accuracy of the grid-and-sphere based method used in parKVFinder to identify ligand-binding sites was previously accessed in [5]. Here, we demonstrate a practical example of its capability and usability, using the HIV-1 protease as a case study. The cavity of HIV-1 protease active site is the target of several antiretroviral drugs, however its volume and shape vary according to its catalytic cycle. Despite the prior description of this cavity motion [22,25], none of these studies used a direct measure, such as cavity volume to characterize the state transition. With parKVFinder, we successfully detected the HIV-1 protease substrate-binding site and, by using the cavity volume as a state descriptor, our software was able to describe the cavity plasticity throughout an MD trajectory.

The ability of parKVFinder to detect volume changes in the HIV-1 protease cavity compared to the other programs is mainly derived from its intuitive set of customizable parameters, such as the size of Probe In and Out, and the Removal Distance. Further, this set helps the identification of different cavities, also the possibility to carry on steered cavity detection, enabling users to segregate cavities. Whilst GHECOM also uses the same grid-and-sphere based method as parKVFinder, it does not allow the user to alter cavity boundary and to segregate cavities, nor do spatial segmentation strategies, which might result in unnecessary calculations which in turn can be the culprit for its slower time performance.

Considering the ever-increasing number of structurally determined structures, fast and accurate cavity detection allows researchers to take full advantage of the atomic information. As demonstrated here, parKVFinder is a highly optimized, customizable and user-friendly tool capable of handling large volumes of data such as those generated by MD simulation.

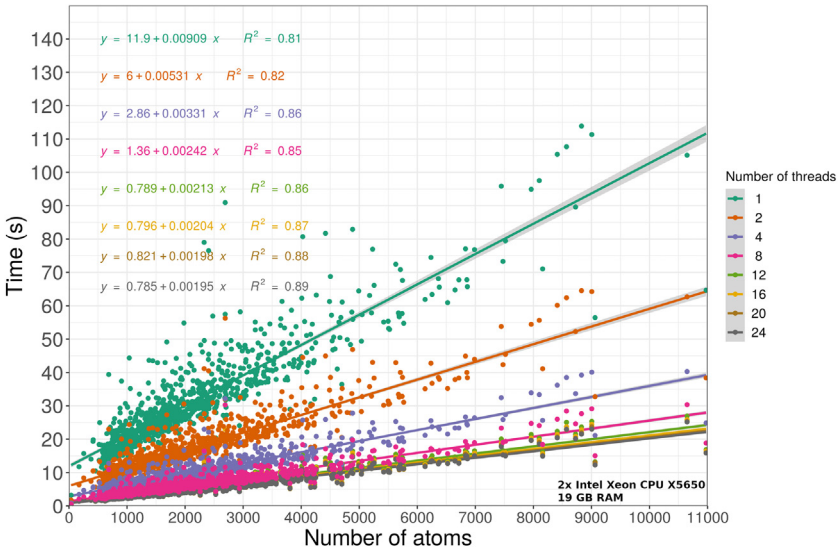

Fig. 6. Computational time against number of atoms per number of threads applied in parKVFinder parallelization.

6. Conclusion

ParKVFinder provides accurate, fast and efficient steered detection and spatial characterization of biomolecular cavities, with a multithreaded parallelization implemented with OpenMP. Cavity detection relies on a set of intuitive and customizable parameters, which users may interact through a GUI or a CLI. Our software also introduces a novel useful parameter, called Removal Distance, that allows users to alter the cavity boundary and, ultimately, generate a finer description of the detected cavities. ParKVFinder accuracy and speed were benchmarked against a set of geometrical cavity detection methods, using a practical example, yielding higher accuracy in correctly estimating the plasticity of the binding site over an MD trajectory. Finally, it presents a faster computation compared to benchmarking methods that successfully analyze the case study.

CRediT authorship contribution statement

João Victor da Silva Guerra: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing - original draft. **Helder Veras Ribeiro Filho:** Validation, Formal analysis, Writing - original draft. **Leandro Oliveira Bortot:** Investigation, Validation, Formal analysis, Writing - original draft. **Rodrigo Vargas Honorato:** Conceptualization, Writing - review & editing. **José Geraldo de Carvalho Pereira:** Conceptualization, Writing - review & editing. **Paulo Sérgio Lopes-de-Oliveira:** Conceptualization, Resources, Supervision, Writing - review & editing.

Declaration of competing interest

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

Acknowledgments

We thank the Brazilian Biosciences National Laboratory (LNBio), part of the Brazilian Center for Research in Energy and Materials (CNPEM) for accessibility to the Computational Biology Laboratory (LBC). This work was supported by the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) [grant number 2018/00629-0] and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) [grant numbers 300787/2019-7, 300036/2020-5 and 301360/2020-0].

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2020.100606.

References

[1] Henrich S, Salo-Ahen OMH, Huang B, Rippmann FF, Cruciani G, Wade RC. Computational approaches to identifying and characterizing protein binding sites for ligand design. J Mol Recognit 2009;23:209–19. http://dx.doi.org/10.1002/jmr.984.

[2] Sotriffer C, Klebe G. Identification and mapping of small-molecule binding sites in proteins: computational tools for structure-based drug design. Farm 2002;57:243–51. http://dx.doi.org/10.1016/S0014-827X(02)01211-9.

[3] Simões T, Lopes D, Dias S, Fernandes F, Pereira J, Jorge J, et al. Geometric detection algorithms for cavities on protein surfaces in molecular graphics: A survey. Comput Graph Forum 2017;36:643–83. http://dx.doi.org/10.1111/cgf.13158.

[4] Simões TMC, Gomes AJP. CavVis—A field-of-view geometric algorithm for protein cavity detection. J Chem Inf Model 2019;59:786–96. http://dx.doi.org/10.1021/acs.jcim.8b00572.

[5] Oliveira SH, Ferraz FA, Honorato RV, Xavier-Neto J, Sobreira TJ, de Oliveira PS. KVFinder: steered identification of protein cavities as a PyMOL plugin. BMC Bioinformatics 2014;15:197. http://dx.doi.org/10.1186/1471-2105-15-197.

[6] Moraes EC, Meirelles GV, Honorato RV, De Souza TDACB, De Souza EE, Murakami MT, et al. Kinase inhibitor profile for human Nek1, Nek6, and Nek7 and analysis of the structural basis for inhibitor specificity. Molecules 2015;20:1176–91. http://dx.doi.org/10.3390/molecules20011176.

[7] Patel H, Kukol A. Evaluation of a novel virtual screening strategy using receptor decoy binding sites. J Negat Results Biomed 2016;15. http://dx.doi.org/10.1186/s12952-016-0058-8.

[8] Najjar FMofidi, Ghadari R, Yousefi R, Safari N, Sheikhhasani V, Sheibani N, et al. Studies to reveal the nature of interactions between catalase and curcumin using computational methods and optical techniques. Int J Biol Macromol 2017;95:550–6. http://dx.doi.org/10.1016/J.IJBIOMAC.2016.11.050.

[9] Terada D, Voet ARD, Noguchi H, Kamata K, Ohki M, Addy C, et al. Computational design of a symmetrical β-trefoil lectin with cancer cell binding activity. Sci Rep 2017;7:5943. http://dx.doi.org/10.1038/s41598-017-06332-7.

[10] Salomon E, Schmitt M, Marapaka A, Stamogiannos A, Revelant G, Schmitt C, et al. Aminobenzosuberone scaffold as a modular chemical tool for the inhibition of therapeutically relevant m1 aminopeptidases. Molecules 2018;23:2607. http://dx.doi.org/10.3390/molecules23102607.

[11] Liou G, Chiang Y-C, Wang Y, Weng J-K. Mechanistic basis for the evolution of chalcone synthase catalytic cysteine reactivity in land plants. J Biol Chem 2018;293:18601–12. http://dx.doi.org/10.1074/jbc.RA118.005695.

[12] Im HN, Kim HS, An DR, Jang JY, Kim J, Yoon HJ, et al. Crystal structure of Rv2258c from Mycobacterium tuberculosis H37Rv, an S-adenosyl-l-methionine-dependent methyltransferase. J Struct Biol 2016;193:172–80. http://dx.doi.org/10.1016/j.jsb.2016.01.002.

[13] Mercaldi GF, Dawson A, Hunter WN, Cordeiro AT. The structure of a Trypanosoma cruzi glucose-6-phosphate dehydrogenase reveals differences from the mammalian enzyme. FEBS Lett 2016;590:2776–86.
[14] Subramaniam S. The cryo-EM revolution: fueling the next phase. IUCrJ 2019;6:1–2. http://dx.doi.org/10.1107/S2052252519000277.
[15] Shoemaker SC, Ando N. X-rays in the cryo-electron microscopy era: Structural biology's dynamic future. Biochemistry 2018;57:277–85. http://dx.doi.org/10.1021/acs.biochem.7b01031.
[16] Ho PT, Reddy VS. Rapid increase of near atomic resolution virus capsid structures determined by cryo-electron microscopy. J Struct Biol 2018;201:1–4. http://dx.doi.org/10.1016/j.jsb.2017.10.007.
[17] Hollingsworth SA, Dror RO. Molecular dynamics simulation for all. Neuron 2018;99:1129–43. http://dx.doi.org/10.1016/j.neuron.2018.08.011.
[18] Krone M, Kozlíková B, Lindow N, Baaden M, Baum D, Parulek J, et al. Visual analysis of biomolecular cavities: State of the art. Comput Graph Forum 2016;35:527–51. http://dx.doi.org/10.1111/cgf.12928.
[19] Ripley BD, Matheron G. Random sets and integral geometry. J R Stat Soc Ser A 1976. http://dx.doi.org/10.2307/2345196.
[20] Diggle PJ, Serra J. Analysis and mathematical morphology. Biometrics 1983. http://dx.doi.org/10.2307/2531038.
[21] Mullikin JC, Verbeek PW. Surface area estimation of digitized planes. Bioimaging 1993;1:6–16. http://dx.doi.org/10.1002/1361-6374(199303)1:1<6::AID-BIO3>3.0.CO;2-3.
[22] Soares RO, Torres PHM, da Silva ML, Pascutti PG. Unraveling HIV protease flaps dynamics by constant pH molecular dynamics simulations. J Struct Biol 2016;195:216–26. http://dx.doi.org/10.1016/j.jsb.2016.06.006.
[23] Lam PY, Jadhav PK, Eyermann CJ, Hodge CN, Ru Y, Bacheler LT, et al. Rational design of potent, bioavailable, nonpeptide cyclic ureas as HIV protease inhibitors. Science 1994;263:380–4. http://dx.doi.org/10.1126/science.8278812.
[24] Spinelli S, Liu QZ, Alzari PM, Hirel PH, Poljak RJ. The three-dimensional structure of the aspartyl protease from the HIV-1 isolate BRU. Biochimie 1991;73:1391–6. http://dx.doi.org/10.1016/0300-9084(91)90169-2.
[25] Hornak V, Okur A, Rizzo RC, Simmerling C. HIV protease flaps spontaneously open and reclose in molecular dynamics simulations. Proc Natl Acad Sci U S A 2006;103:915–20. http://dx.doi.org/10.1073/pnas.0508452103.
[26] Zhu H, Pisabarro MT. MSPocket: an orientation-independent algorithm for the detection of ligand binding pockets. Bioinformatics 2011;27:351–8. http://dx.doi.org/10.1093/bioinformatics/btq672.
[27] Kawabata T. Detection of multiscale pockets on protein surfaces using mathematical morphology. Proteins 2010;78:1195–211. http://dx.doi.org/10.1002/prot.22639.
[28] Le Guilloux V, Schmidtke P, Tuffery P. Fpocket: An open source platform for ligand pocket detection. BMC Bioinformatics 2009;10:168. http://dx.doi.org/10.1186/1471-2105-10-168.
[29] Stank A, Kokh DB, Fuller JC, Wade RC. Protein binding pocket dynamics. Acc Chem Res 2016;49:809–15. http://dx.doi.org/10.1021/acs.accounts.5b00516.