UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE CIÊNCIAS FARMACÊUTICAS

João Victor da Silva Guerra

Development of a computational platform for structural and functional characterization of biomolecules and binding sites

Desenvolvimento de plataforma computacional para caracterização estrutural e funcional de biomoléculas e sítios de ligação

CAMPINAS
2024

João Victor da Silva Guerra

Development of a computational platform for structural and functional characterization of biomolecules and binding sites

Desenvolvimento de plataforma computacional para caracterização estrutural e funcional de biomoléculas e sítios de ligação

Tese apresentada à Faculdade de Ciências Farmacêuticas da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciências, na área de Fármacos, Medicamentos e Insumos para a Saúde.

Thesis presented to the Faculty of Pharmaceutical Sciences of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Science, specializing in Pharmaceuticals, Medicines and Health Supplies.

Orientador: Prof. Dr. Paulo Sergio Lopes de Oliveira

Este exemplar corresponde à versão final da Tese defendida por João Victor da Silva Guerra e orientada pelo Prof. Dr. Paulo Sergio Lopes de Oliveira.

CAMPINAS
2024

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Ciências Médicas
Maristella Soares dos Santos - CRB 8/8402

Guerra, João Victor da Silva, 1993-
G937d Development of a computational platform for structural and functional characterization of biomolecules and binding sites / João Victor da Silva Guerra. – Campinas, SP : [s.n.], 2024.

Orientador: Paulo Sergio Lopes de Oliveira.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Ciências Farmacêuticas.

1. Biologia estrutural. 2. Ciência de dados. 3. Plataforma computacional. 4. Sítios de ligação. 5. Dinâmica molecular. I. Oliveira, Paulo Sergio Lopes de, 1969-. II. Universidade Estadual de Campinas. Faculdade de Ciências Farmacêuticas. III. Título.

Informações Complementares

Título em outro idioma: Desenvolvimento de plataforma computacional para caracterização estrutural e funcional de biomoléculas e sítios de ligação
Palavras-chave em inglês:
Structural biology
Data science
Computational platform
Binding sites
Molecular dynamics
Área de concentração: Fármacos, Medicamentos e Insumos para a Saúde
Titulação: Doutor em Ciências
Banca examinadora:
Paulo Sergio Lopes de Oliveira [Orientador]
Eduardo Xavier Silva Miqueles
Andre Luis Berteli Ambrosio
Gustavo Fernando Mercaldi
Felippe Mariano Colombari
Data de defesa: 22-03-2024
Programa de Pós-Graduação: Ciências Farmacêuticas

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-6800-442
- Currículo Lattes do autor: https://lattes.cnpq.br/5809550322159439

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE CIÊNCIAS FARMACÊUTICAS

Autor: João Victor da Silva Guerra

Orientador: Prof. Dr. Paulo Sergio Lopes de Oliveira

Tese aprovada em 22 de março de 2024

Comissão Examinadora:

- Prof. Dr. Paulo Sergio Lopes de Oliveira
  Centro Nacional de Pesquisa em Energia e Materiais (CNPEM)
- Dr. Eduardo Xavier Silva Miqueles
  Centro Nacional de Pesquisa em Energia e Materiais (CNPEM)
- Prof. Dr. Andre Luis Berteli Ambrósio
  Universidade de São Paulo (USP)
- Dr. Gustavo Fernando Mercaldi
  Centro Nacional de Pesquisa em Energia e Materiais (CNPEM)
- Dr. Felippe Mariano Colombari
  Centro Nacional de Pesquisa em Energia e Materiais (CNPEM)

A ata da defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Pós-Graduação da Faculdade de Ciências Farmacêuticas.

Campinas, 22 de março de 2024

Without data, you're just a person with an opinion. (William E. Deming)

Acknowledgements

My heartfelt gratitude goes to all those who supported me throughout the development of this project. Foremost, I want to express my heartfelt thanks to my parents, Roseli do Carmo Freitas da Silva and Mario Luiz da Silva Guerra, for their unwavering support across every stage of my academic and professional journey. I also want to extend special appreciation to my partner, Bruna Martins da Silva, whose steadfast support has been invaluable throughout all phases of this project.

A sincere thank you to my advisor, Dr. Paulo Sergio Lopes de Oliveira, for providing me with scientific and creative autonomy to shape this thesis. Above all, I am grateful for the opportunity to contribute to a research center of excellence under his guidance. My appreciation also extends to my colleagues and former colleagues at the Computational Biology Laboratory (LBC; *Laboratório de Biologia Computacional*)—Dr. José Geraldo de Carvalho Pereira, Dr. Helder Veras Ribeiro Filho, MSc. Luiz Fernando Giolo Alves, Dr. Mariana Bortoletto Grizante, Dr. Gabriel Ernesto Jara, Dr. Leandro Oliveira Bortot, MSc. Amauri Donadon Leal Junior, and Pablo Wesley de Aguiar e Silva. Their support during crucial moments, collaborative idea-sharing, and insightful suggestions have significantly contributed to the development of this project. In particular, I would like to acknowledge the contributions of Dr. Helder Veras Ribeiro Filho and Dr. José Geraldo de Carvalho Pereira in the development and planning of tools, as well as Dr. Gabriel Ernesto Jara and Dr. Leandro Oliveira Bortot for their guidance in molecular dynamics-related topics. My gratitude knows no bounds for all those who assisted me during this crucial phase of my academic journey.

Finally, I express my thanks to the Postgraduate Program in Pharmaceutical Sciences (PPGCF; *Programa de Pós-Graduação em Ciências Farmacêuticas*) at the Faculty of Pharmaceutical Sciences (FCF; *Faculdade de Ciências Farmacêuticas*) of the University of Campinas (UNICAMP; *Universidade Estadual de Campinas*), the National Laboratory of Biosciences (LNBio; *Laboratório Nacional de Biociências*), and the National Center for Research in Energy and Materials (CNPEM; *Centro Nacional de Pesquisa em Energia e Materiais*) for providing the necessary infrastructure and support for this work. Additionally, this study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil* (CAPES) – Finance Code 001 [Grant Number 88887.928702/2023-00], and grant #2018/00629-0, São Paulo Research Foundation (FAPESP). The opinions, hypotheses, and conclusions or recommendations expressed in this material are the responsibility of the author and do not necessarily reflect the views of FAPESP.

Resumo

Na atual era da ciência de dados, os campos da biologia estrutural e computacional têm se beneficiado significativamente do aumento da qualidade e acessibilidade da informação bioestrutural. Compreender a estrutura e a função das biomoléculas, juntamente com seus sítios de ligação, é crucial para desvendar processos biológicos e obter percepções valiosas na descoberta e desenho de fármacos. Apesar dessa importância, aprofundar-se nos mecanismos intrínsecos dessas interações e identificar potenciais sítios de ligação ainda é desafiador devido à natureza complexa das biomoléculas e à diversidade das interações. Diante dessa situação, há uma crescente demanda por ferramentas computacionais robustas que estudem de forma abrangente os sistemas biomoleculares. Neste contexto, apresentamos a KVFinder suite, uma plataforma computacional composta por ferramentas como parKVFinder, pyKVFinder, KVFinder-web, SERD e KVFinderMD. Cada ferramenta desempenha um papel específico, possibilitando não apenas a codificação, mas também a análise detalhada de biomoléculas e de seus sítios de ligação. De maneira importante, essas ferramentas oferecem a flexibilidade de utilizar suas unidades básicas em aplicações de ciência de dados e inteligência artificial. Ao longo deste trabalho, aplicações das ferramentas da KVFinder suite em estudos de caso terapêuticos demonstram sua eficácia. Essas aplicações, juntamente com novas caracterizações, proporcionam uma referência valiosa para a comunidade científica. Com suas diversas funcionalidades e capacidade de integração em aplicações de ciência de dados e inteligência artificial, a KVFinder suite tem o potencial de avançar na compreensão dos mecanismos de interação biomolecular e em estratégias terapêuticas inovadoras. Sua disponibilidade representa um recurso significativo, acelerando a pesquisa e ampliando o conhecimento na área, mantendo eficiência computacional.

Abstract

In the current era of data science, the fields of structural and computational biology have significantly benefited from the increasing quality and accessibility of biostructural information. Understanding the structure and function of biomolecules, along with their binding sites, is crucial for unraveling biological processes and gaining valuable insights into drug discovery and design. Despite this importance, delving into the intrinsic mechanisms of these interactions and pinpointing potential binding sites remains challenging due to the complex nature of biomolecules and the diversity of interactions. Given this situation, there is a growing demand for robust computational tools that comprehensively study biomolecular systems. Herein, we introduce the KVFinder suite, a computational platform comprising tools such as parKVFinder, pyKVFinder, KVFinder-web, SERD, and KVFinderMD. Each tool plays a specific role, enabling not only the coding but also the in-depth analysis of biomolecules and their binding sites. Importantly, these tools offer the flexibility of using their basic units in data science and artificial intelligence applications. Throughout this work, applications of KVFinder suite tools in therapeutic case studies showcase their effectiveness. These applications, along with new characterizations, offer a valuable reference for the scientific community. With its diverse functionalities and integration capabilities into data science and artificial intelligence applications, the KVFinder suite holds the potential to advance our understanding of biomolecular interaction mechanisms and innovative therapeutic strategies. Its availability represents a significant resource, accelerating research and advancing knowledge in the field while maintaining computational efficiency.

List of Figures

List of Tables

List of Abbreviations

3D Tridimensional

AI Artificial Intelligence

ADRP ADP-ribose phosphatase

API Application Programming Interface

Cα α carbon

Cβ β carbon

CAPES *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil*

CAPRI Critical Assessment of PRedicted Interactions

CEC *Congresso de Estudantes do CNPEM*

CHIKV Chikungunya Virus

CLI Command-Line Interface

CNPEM *Centro Nacional de Pesquisa em Energia e Materiais*

CSD Cambridge Structural Database

DFS Depth-First Search

DL Deep Learning

EEEV Eastern Equine Encephalitis Virus

FAPESP *Fundação de Amparo à Pesquisa do Estado de São Paulo*

FCF *Faculdade de Ciências Farmacêuticas*

GHECOM Grid-based HECOMi finder

GUI Graphical User Interface

HIV-1 Human Immunodeficiency Virus type 1

HPC High-Performance Computing

HTTP HyperText Transfer Protocol

KVFinderMD KVFinder for Molecular Dynamics analysis

LBC *Laboratório de Biologia Computacional*

LES Ligand Excluded Surface

LNBio *Laboratório Nacional de Biociências*

MAYV Mayaro Virus

MD Molecular Dynamics

MIMD Multiple Instruction, Multiple Data

ML Machine Learning

mmCIF macromolecular Crystallographic Information File

MRAE Mean Relative Absolute Error

ndarray N-dimensional array

Nsps Non-structural proteins

parKVFinder parallel KVFinder

PC Packing Coefficient

PDB Protein Data Bank

PDI Protein-DNA Interaction

Pi *Probe In*

Po *Probe Out*

PLI Protein-Ligand Interaction

PPGCF *Programa de Pós-Graduação em Ciências Farmacêuticas*

PPI Protein-Protein Interaction

PRI Protein-RNA Interaction

pyKVFinder Python-C parallel KVFinder

RE Relative Error

RMSD Root-Mean-Square Deviation

SARS-CoV-2 Severe Acute Respiratory Syndrome Coronavirus 2

SAS Solvent Accessible Surface

SERD Solvent-Exposed Residues Detection

SES Solvent Excluded Surface

SIMD Single Instruction, Multiple Data

SINV Sindbis Virus

SMP Symmetric Multiprocessing

SWIG Simplified Wrapper and Interface Generator

UNICAMP *Universidade Estadual de Campinas*

vdW van der Waals

VEEV Venezuelan Equine Encephalitis Virus

wwPDB Worldwide Protein Data Bank

Table of Contents

Chapter 1: Introduction

Biomolecules, such as proteins, nucleic acids, carbohydrates, and lipids, play crucial roles in various biological processes within organisms. Fundamental biological processes, including signal transduction, structural integrity, cell adhesion, and apoptosis, are regulated by biomolecular interactions [1–3]. These interactions are vital for unraveling biological processes and advancing pharmacological therapies [2]. However, investigating the intrinsic mechanisms of these interactions and potential binding sites proves challenging due to the complex nature of biomolecules and the myriad interactions among them.

These interactions involve receptors and ligands, spanning ions, such as iron and phosphate to macromolecules such as proteins, RNA, and DNA [4]. Molecular structures intricately fold, creating specific binding sites often nestled in cavities along the molecular surface, exposing morphological, topological and physicochemical patterns to accommodate specific ligands [2,5]. Receptor-ligand interactions, including protein-protein interaction (PPI), protein-ligand interaction (PLI), protein-RNA interaction (PRI), and protein-DNA interaction (PDI), arise from complementarity between the molecular surfaces of the interacting pair, limiting efficient interaction to a select few ligands with a given receptor [2, 6].

Given the paramount importance of biomolecular interactions, a comprehensive study of biomolecules and their binding sites is imperative for understanding biological processes and advancing pharmaceutical development. Computational methods for identifying binding sites and characterizing biomolecular interactions offer an effective alternative to experimental methods, providing detailed information [6]. Identifying binding sites poses a classification problem, aiming to determine whether a specific point on a biomolecule's surface functions as a binding site or not [1, 2, 6]. Advancements in computational resources and databases have led to the adoption of *in silico* methods for simulating biomolecular dynamics and implementing artificial intelligence (AI) applications to study biomolecular structures [7]. These structural data collectively provide fertile ground for data interpretation through automated protocols or data science applications, yet intensive data analysis requires efficient routines integrated with easily manipulated data structures.

In this context, the development of robust and comprehensive computational tools is crucial for the systematic study of biomolecular systems, accommodating various forms of user interaction. Foundational components for applications, programs, and/or automated protocols in computational biology, structural biology, machine learning (ML), and related fields are indispensable for the analysis of biomolecules and/or binding sites.

This research addresses this need by introducing the KVFinder suite, a comprehensive computational platform.

The KVFinder suite comprises five tools, each serving a specific purpose:

- **parKVFinder:** developed for the detection and characterization of any type of biomolecular cavity, integrated with a graphical plugin for PyMOL [8, 9];
- **pyKVFinder:** a Python package for detecting and characterizing cavities in biomolecular structures in automated protocols and data science applications [5];
- **KVFinder-web:** a web application with a simplified protocol for detecting and characterizing cavities in any type of biomolecular structure [10];
- **SERD:** a Python package for determining solvent accessibility of residues and representing biomolecular structures as graphs;
- **KVFinderMD:** a Python package for exploring the dynamics of binding sites in biomolecular structures.

The KVFinder suite is positioned to serve as a robust and comprehensive toolkit for the systematic study of biomolecular systems, offering applicability across diverse domains, ranging from structural biology to data science. Functioning as a versatile platform, it not only codes structural information through different data codings (e.g., volumetric representation, atomistic representation, graph representation) and relevant descriptors of binding sites (e.g., shape, volume, area, depth, hydrophobicity, composition) but also facilitates the analysis and characterization of this information. Furthermore, the basic units of the KVFinder suite are designed to be adaptable for applications in data science and AI.

Chapter 2: Objectives

This work aims to develop a computational platform known as **KVFinder suite**, with a focus on studying biomolecular systems within the domain of structural biology.

The specific objectives include: (1) Enhancement and development of descriptors for the KVFinder suite; (2) Implementation of data codings for biomolecules and binding sites; (3) Development of a tool tailored to automated protocols and data science applications; (4) Development of a web service for binding site analysis; (5) Development of a tool for assessing solvent accessibility; and (6) Development of a tool specifically for the analysis of molecular dynamics.

Chapter 3: Literature Review

In the forthcoming sections, we will delve into computational approaches for identifying binding sites, a spectrum of binding site characterizations and the current state of the art. Computational methods leverage the wealth of structural information from both bound and unbounded states to identify potential binding sites. Within the receptor-ligand complex, the binding phenomenon is intricately governed by the morphological, topological, and physicochemical patterns exhibited by native or transient binding sites, creating a high complementarity between the receptor and the ligand [1, 2, 8, 11]. Together, they contribute to a deeper understanding of molecular recognition within biological systems.

3.1 Binding Site Identification

Binding sites are found within a diverse array of cavities, yet a formal mathematical definition of biomolecular cavities remains elusive [6, 8]. Using the mathematical theory of convexity [6, 12], a cavity is defined as a connected component within the complement space of the biomolecule enclosed by its convex hull. As a topological entity, the connected components (i.e., cavities) are related to the first Betti number β_0 (i.e., the number of connected components) of the complement space. In 3D, biomolecules are represented as shapes, with connected components corresponding to Betti numbers β_i ($i = 0, 1, 2$) in 3D. Consequently, cavities are categorized based on their morphology as voids (0-ary), pockets (1-ary), and channels (2-ary) (Figure 3.1). Pockets can be further categorized into clefts, grooves, invaginations and tunnels.

In biomolecules, particularly proteins and nucleic acids, solvent-exposed clefts or buried cavities are crucial for ligand binding, which can ultimately regulate biological function [1, 2, 13]. The identification and characterization of these binding sites are fundamental for comprehending the intricate biomolecular interactions and their diverse functions. Morphological, topological and physicochemical properties at the contact interface, e.g., shape, volume, area, charge, hydrophobicity, solvation, and type of interactions, dictate the functions and interactions of biomolecules [1, 2, 8, 14, 15]. Identifying and sizing these cavities are initial steps in designing ligands based on protein structures [13]. Moreover, the structural and physicochemical characteristics of proteins are valuable in drug discovery and design [8].

In the current era of data science, the fields of computational and structural biology have significantly benefited from the increasing availability of biostructural data, as highlighted by [16]: "Structural biology meets data science". Advances in X-ray crystal-

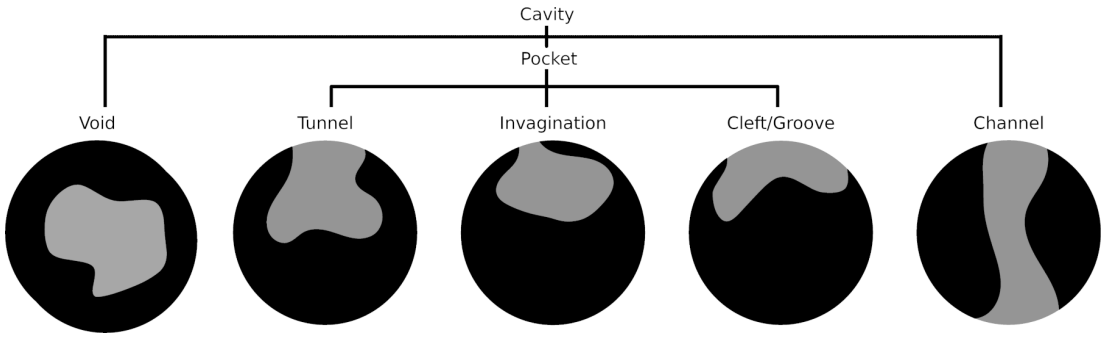


Figure 3.1: Types of biomolecular cavities. A biomolecular cavity (gray region) as a contiguous segment within the complement space of the biomolecule (black region), as defined by [12]. The categorization of these cavities follows an n-ary classification, where 'n' denotes the number of mouth openings. Specifically, a 0-ary cavity is identified as a void, representing an enclosed spatial configuration without external openings. In contrast, a 1-ary cavity, illustrated by a pocket, includes a single mouth opening. It is noteworthy that pockets encompass diverse forms like tunnels, invaginations, clefts, and grooves. Meanwhile, a 2-ary cavity, exemplified by a channel, features two mouth openings, facilitating the transit of molecular entities [6].

lography and electron microscopy techniques have expanded the determination of novel structures with respect to quality and size of these structures [17]. Biomolecules actively participate in various biological processes, such as protein folding, enzyme catalysis, DNA replication, DNA repair, cell signaling, virus-host interactions, and drug resistance. Dysregulation of these interactions is associated with numerous pathologies, including cancer, metabolic disorders, and pathogenic diseases [18]. For this reason, the identification and characterization of ligand-binding sites are the basis of rational structure-based drug discovery and design [1, 2]. Therefore, the imperative to identify targetable regions within the interaction area for new drugs becomes more apparent each day.

3.2 Binding Site Characterization

Biomolecular interactions (e.g., PPIs, PLIs, PRIs, and PDIs) arise from distinctive property fingerprints at native or transient binding sites, where the molecular recognition hinges on the morphological, topological and physicochemical complementarity between the target biomolecule and its binding partner [1,2,8,19]. Binding sites impose constraints on putative ligands, with each interaction triggering a distinct biological function. Thus, describing potential binding sites in terms of morphological, topological, and physicochemical characteristics becomes essential for rational drug discovery and design, and assessing binding site druggability [8, 13, 14, 19].

The morphological attributes of biomolecular cavities establish stringent constraints on the geometric profiles of ligands capable of interacting efficiently within them. High affinity between the binding site and potential binders relies on sufficiently large interaction interfaces—extensive surface areas within the biomolecular cavity. The specificity of the binding site is dictated by geometric restrictions, including shape, size, and burial extent [13, 20]. Importantly, the cavity volume must accommodate the potential ligand [21]. The shape complementarity between the receptor and the ligand plays a decisive role in the binding process, with small ligands typically binding to buried concave

sites on the molecular surface [2]. Empirical studies have revealed that the active site often constitutes the largest and deepest cavity in enzymatic proteins [20].

Topological characteristics of protein cavities contribute to functional analysis. The inference of the functionality of protein cavities has been achieved through comparison with homologous proteins, which exhibit homology and sequential similarity at the active site [22]. Conserved local structural patterns, such as the catalytic triads of serine proteases [23], exemplify these homologous features. In enzymes, these specific configurations assume preferential spatial arrangements to execute elementary steps of the catalytic reaction, influencing the molecular recognition process of ligands by active sites [1, 2, 24]. Describing a protein cavity according to the location of its different types of residues (e.g., hydrogen donors, electron receptors, hydrophobic contacts, and aromatics) becomes an insightful approach. Binding sites exhibit conserved amino acid sequences within protein families, offering valuable functional information [2]. Amino acid composition distinguishes enzymatic from non-enzymatic binding sites, revealing varying compositions across different proteins [8, 25].

Physicochemical characteristics play an essential role in selecting energetically favorable interactions. The synergy of van der Waals, hydrophobic, electrostatic, hydrogen bonding, and solvation interactions creates an energetically favorable environment for the binding process [2]. For instance, hydrophobicity, quantified by the partition coefficient P, influences the kinetic and dynamic characteristics of drug action [26]. It is modulated by the shape of the binding site and the exposed area of residues [2]. Various approaches generate distinct hydrophobicity scales for amino acids [27], such as Eisenberg & Weiss [28], Hessa & Heijne [29], Kyte & Doolittle [30], Moon & Fleming [31], Radzicka & Wolfenden [32], Wimley & White [33], and Zhao & London [34]. Conversely, the electrostatic potential, calculated by the Poisson-Boltzmann equation [35], estimates ligand anchoring points, interaction free energy, biomolecule stability, and average atomic forces.

Therefore, biomolecular interactions, shaped by morphological, topological, and physicochemical complementarity, form the foundation for understanding the biological function of biomolecules. While there is no simple and universally effective approach, characterizing binding sites based on their distincitve properties can lead to the identification of functionally relevant ones. As we unravel the intricate interplay of these properties, we gain essential insights into rational drug discovery and design, and the assessment of binding site druggability, marking a pivotal step toward advancing biomolecular research and therapeutic interventions [1, 2, 8, 13].

3.3 State of the Art

Over the past decades, various *in silico* have been developed for identifying binding sites in proteins to deepen our knowledge of a specific protein's function and for drug discovery and design [13]. However, only a few methodologies are applicable to other types of biomolecules, such as nucleic acids, carbohydrates, and lipids. The published computational approaches can be divided into three main categories: evolutionary, energetic, and geometric [4, 6, 8].

- **Evolutionary methods:** are based on the search for conserved residues in multiple sequence alignments and information from known binding site profiles;
- **Energetic methods:** identify binding sites based on the energetic interaction between the target biomolecule and a chemical probe, usually a chemical group;
- **Geometric methods:** identify cavities by analyzing the geometric characteristics of the molecular surface.

Each category of methods has its own advantages and disadvantages [1, 2, 6]. Evolutionary algorithms heavily depend on sequence information or databases of active binding sites, with the alignment procedure being a crucial step. Energetic methods rely on filtering procedures, force field parametrization, and applied scoring functions. In contrast, geometric methods are relatively simple and straightforward, requiring only structural data of the protein (i.e., the file in Protein Data Bank (PDB), XYZ, macromolecular Crystallographic Information File (mmCIF), or equivalent format), containing the Cartesian coordinates of atoms, easily accessible through the Worldwide Protein Data Bank (wwPDB). Once atom positions are available, geometry-based methods can represent any type of biomolecule [2, 4, 6]. While purely geometric methods are efficient in identifying all types of cavities in a target molecule, the challenge lies in discerning functionally relevant cavities. Nevertheless, it becomes possible to identify these relevant cavities (i.e., binding sites) for specific ligands by characterizing cavities based on well-chosen morphological, topological and physicochemical properties [1, 2, 8, 13].

In this context, geometric methods are the most widely used in the literature, as they are simple, direct, and do not require prior knowledge [2, 4]. Evolution-based methods are limited to proteins, as they depend on principles of biological evolution. Energy-based methods could apply to other biomolecule types but would require fine-tuning of force field parameters adapted to them. Given the distinct properties of nucleic acids, carbohydrates, and lipids compared to proteins, approaches that depend exclusively on geometric information (e.g., atom positions and radii) are more desirable.

3.3.1 Geometric Approaches

The detection of cavities through geometric approaches is widespread, encompassing various techniques [6, 9]. These techniques are simple, straightforward, and do not require prior knowledge, making them the most commonly used in the literature [2, 4]. In this context, we present a brief classification of geometric approaches for cavity detection (Figure 3.2), including techniques based on tridimensional (3D) grids, probes, surface, tessellation and their combinations [6, 9, 36].

- **Grid-based algorithms** (e.g., CavitySearch [37], POCKET [38], LIGSITE [39], POVME 3.0 [40]) represent a set of atoms as discrete points, typically employing a 3D grid aligned to the axes as a scalar field (i.e., a density map), where each point is assigned an integer or a boolean. These grid maps are used to cluster relevant empty space points (i.e., not belonging to the solute) into cavities using voxel clustering algorithms. Generally, these methods use simple data structures to represent

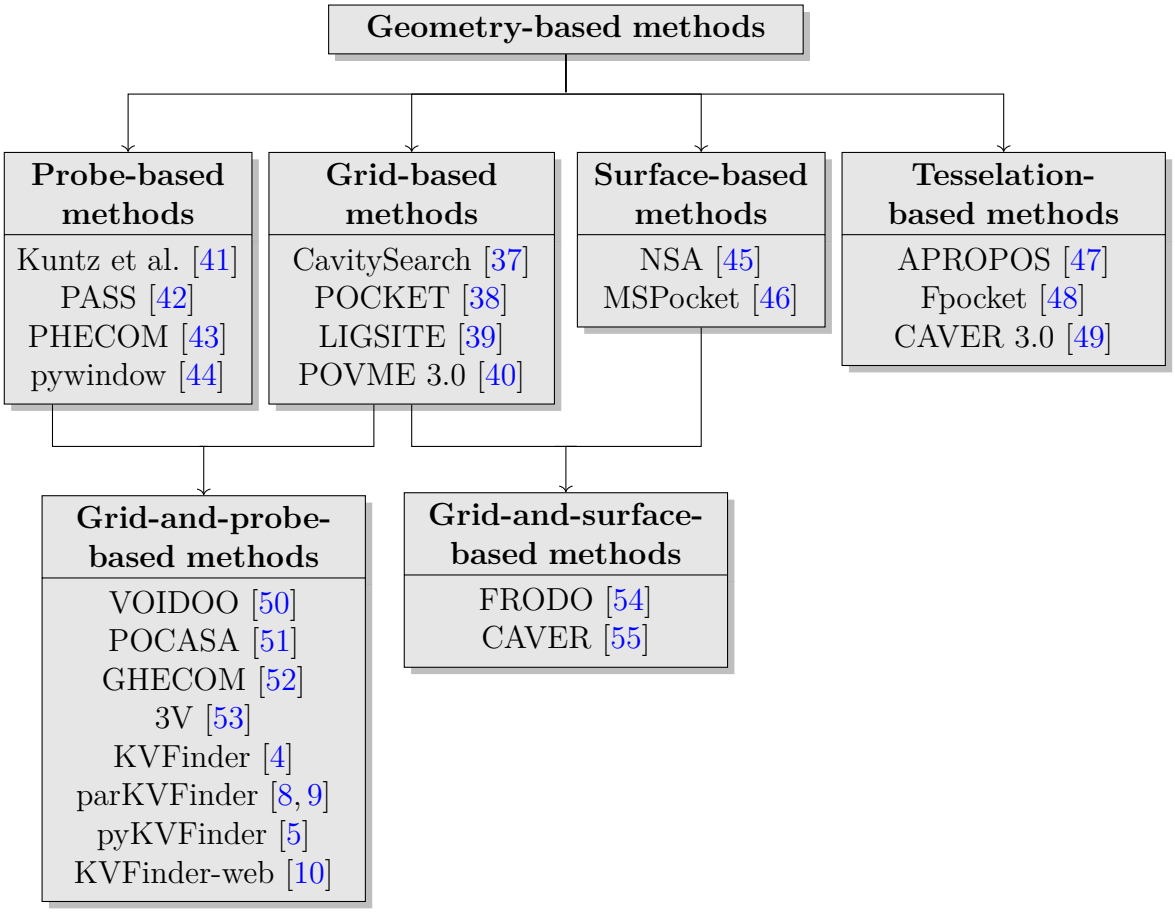


Figure 3.2: Classification of geometry-based methods.

a collection of data at a discrete point and identifying cavities automatically. However, geometric accuracy, computation time, and memory consumption are heavily influenced by the grid resolution (i.e., grid-spacing sensitivity). Additionally, these methods are rotationally variant, meaning that the orientation of a given molecule slightly affects cavity detection (i.e., orientation sensitivity);

- **Probe-based algorithms** (e.g., Kuntz et al. [41], PASS [42], PHECOM [43], pywindow [44]) use a set of atoms, taking into account their positions and van der Waals (vdW) radii, to model the molecular surface. The atoms are inspected by one or more probes, typically rigid spheres, to assess its accessibility levels. This approach is capable of detecting any type of cavity and is related to the spatial extension of potential ligands. However, it may struggle to find and unequivocally delineate the solvent-cavity boundary (i.e., mouth-opening ambiguity);

- **Surface-based algorithms** (e.g., NSA [45], MSPocket [46]) do not use a rigid sphere model but rather a molecular surface model (e.g., vdW surface, Solvent Excluded Surface (SES), Solvent Accessible Surface (SAS), and Ligand Excluded Surface (LES)), defining the molecular interface and its environment. Analysis of the molecular interface identifies cavities based on accessibility to a specific solvent or ligand. In this case, cavity detection occurs automatically, as in grid-based methods, but without mouth-opening ambiguity. However, in some cases, these algorithms

may struggle to detect all types of cavities and their complete extent;

- **Tessellation-based algorithms** (e.g., APROPOS [47], Fpocket [48], CAVER 3.0 [49]) employ computational geometry techniques (e.g., α-shapes, β-shapes, Voronoi diagrams, and Apollonius diagrams). α-shapes and Voronoi tessellation use atomic centers, representing atoms by constant-radius spheres. β-shapes and Apollonius-based methods rely on variable-radius spheres to represent atoms. These techniques explore the modeled atoms to detect cavities. Typically, they do not rely on information from the molecular surface to identify cavities but may struggle to detect the correct binding site location, delineate the solvent-cavity boundary, and define the number of surface atoms.

This classification illustrates that each technique has its own inherent strengths and weaknesses, rendering them more suitable for specific applications [6,9]. The cohesive combination of these techniques aims to leverage the capabilities of each and mitigate their individual shortcomings, resulting in more robust approaches. Within this broader context, two notable subcategories emerge:

- **Grid-and-probe-based methods** (e.g., VOIDOO [50], POCASA [51], GHECOM [52], 3V [53], KVFinder [4], parKVFinder [8,9]) combine the strengths of both grid- and probe-based methods. Analogous to grid-based methods, they rely on a scalar field defined at each grid point and predominantly use large probe spheres rolling on the vdW surface. These probes define cavities between the probe-generated surface and the molecular surface. This combined approach mitigates mouth-opening ambiguity and orientation sensitivity, common in grid-based methods [6]. The mouth-opening ambiguity is also known as cavity ceiling problem, which can be controlled by customizable probe sizes [4,6]. However, grid-spacing sensitivity remains, unless we use a grid spacing of at most half of the smaller probe to mitigate it. The usage of probes in grid-based methods follows three techniques, that are atom fattening (originating SAS or SAS-like surfaces), exemplified by VOIDOO [50], rolling probes of unequal radii on the vdW surface, as seen in POCASA [51], GHECOM [52], and 3V [53], and concentric probes of unequal radii at grid points, specifically implemented in KVFinder software [4];

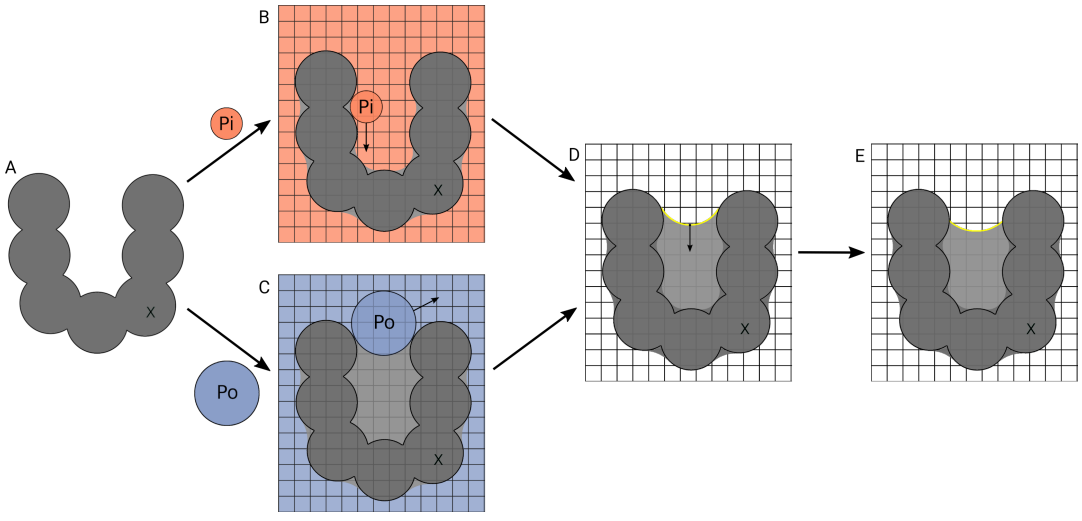
- **Grid-and-surface-based methods** (e.g., FRODO [54], CAVER [55]) combine the strengths of both grid- and surface-based methods. Similar to probe spheres, surfaces resolve the ambiguity issue in grid-based methods, particularly in defining cavity ceilings (and, consequently, mouth openings). These methods employ a scalar field in conjunction with a 3D grid, where the scalar field may be defined by various functions. In summary, the utilization of surfaces with grids overcomes common issues associated with grid-based methods, namely, orientation sensitivity and mouth-opening ambiguity. However, the challenge of grid-spacing sensitivity persists unless a grid spacing of at most half of the smaller probe.

In this context, the exploration of geometric approaches for cavity detection reveals a diverse landscape of techniques, each with its unique strengths and weaknesses. The

simplicity and accessibility of these methods have made them foundational in biomolecular research [2, 6]. This classification, encompassing grid-based, probe-based, surface-based, and tessellation-based algorithms, provides a comprehensive overview of the strategies employed in structural and functional characterization of biomolecules and binding sites [6]. The cohesive integration of these techniques, as seen in grid-and-probe-based or grid-and-surface-based methods, represents a strategic effort to overcome individual limitations and enhance the robustness of cavity detection methodologies. By acknowledging the strengths of each approach and strategically combining them, researchers aim to optimize the efficacy of computational platforms, paving the way for more nuanced and accurate structural and functional insights into biomolecules.

3.3.2 Well-established Computational Tools

The vast majority of cavity detection tools were originally developed for proteins (see Section 3.3.1). Although these algorithms exhibit robustness in describing and analyzing any molecular system (e.g., protein, DNA, RNA, inorganic material, supramolecular cage, etc.), so far, only a few tools have been applied to other biomolecular systems, excluding proteins, to assess structural characteristics such as the shape, volume, area and mouth openings. Our literature review led us to identify tools meeting specific criteria: applicability to any molecular system, availability as free software, comprehensive documentation, robust developer support, a well-conceived set of characterizations, and recognition within the scientific community. Guided by these stringent criteria, we spotlight four well-established cavity detection tools: parKVFinder, Fpocket, GHECOM, and CAVER tools.



Source: Reprinted with permission from [36]. Copyright 2023 American Chemical Society.

Figure 3.3: Cavity detection algorithm in parKVFinder. (A) A biomolecular structure X, composed of atoms modeled as rigid spheres with van der Waals radii, is inserted into a 3D grid. **(B)** The *Probe In* (Pi) probe traverses the surface of the structure, moving through grid points (orange). **(C)** Next, the *Probe Out* (Po) probe traverses accessible points in blue. **(D)** Cavity points (light gray) are defined as the difference between the accessible points of the probes. Points not reached by Pi (dark gray) define the SES (standard) or SAS, depending on the surface representation chosen by the user. **(E)** Finally, a distance-based removal procedure is applied to eliminate cavity points near the cavity-bulk boundary (yellow line).

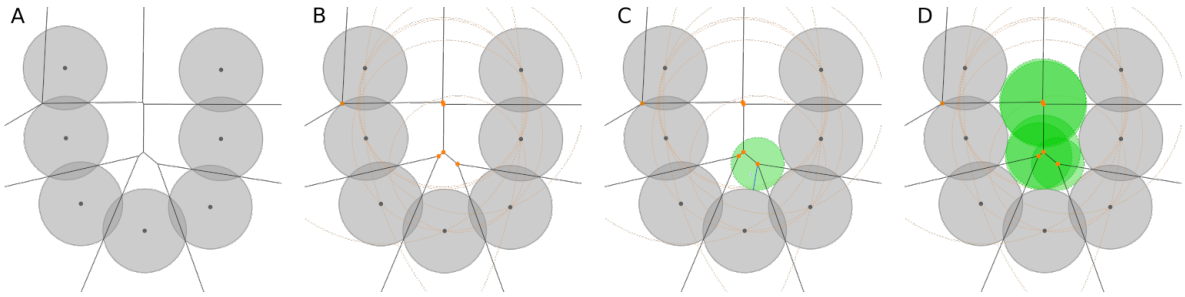
3.3.2.1 parKVFinder

parKVFinder [8,9], a grid-and-probe-based approach, detects and characterizes any type of biomolecular cavity, integrated with a graphical plugin for PyMOL [56]. The cavity detection algorithm (Figure 3.3) employs a dual-probe methodology based on mathematical morphology theory [57, 58]. Originally implemented in the KVFinder software [4], this algorithm involves two probes, a smaller probe (*Probe In*) and a larger probe (*Probe Out*), inspecting grid points to define cavities as the non-overlapping regions traversed by these probes.

It is noteworthy that the original KVFinder software [4], introduced in 2014, has been deprecated. However, subsequent implementations, namely parKVFinder [9], pyKVFinder [5], and KVFinder-web [10], have been developed to enhance computational performance and usability. Each of these tools addresses different scientific community demands in a flexible manner. Cavity characterization in these tools includes morphological descriptors (e.g., volume, area, shape, and depth), topological descriptors (e.g., interface residues surrounding the cavities and their classification—aliphatic, aromatic, polar uncharged, negatively charged, and positively charged—), and physicochemical descriptors (e.g., hydrophobicity). Further details on the parKVFinder can be found in Section 5.1.

3.3.2.2 Fpocket

Fpocket [48], a tessellation-based approach, employs the concept of α spheres, originally introduced by [13], for the detection of molecular pockets. The cavity detection algorithm (Figure 3.4) performs a comprehensive analysis to identify the whole set of α spheres within a given molecular structure, using the *qhull* package [59]. The process involves categorizing small alpha spheres situated inside the structure, large spheres outside the structure, and the spheres in between as corresponding to cavities. To refine the cavity selection, Fpocket eliminates any alpha sphere falling outside a customized range of minimum and maximum radii. The remaining α spheres are grouped into cavities based on proximity and neighborhood relationships, and cavities of poor interest (e.g., hydrophilic or small putative pockets) excluded from further analysis. Subsequently, the cavities are assessed using a set of dpocket descriptors [48], allowing the cavities to be ranked according to their putative binding affinity for small molecules.

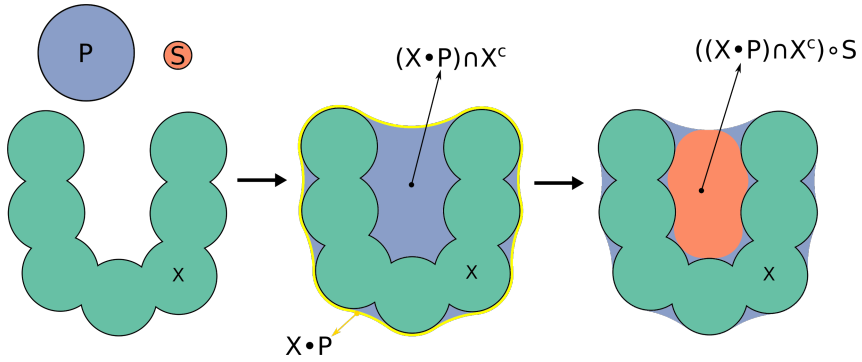


Source: Reprinted with permission from [36]. Copyright 2023 American Chemical Society.

Figure 3.4: Cavity detection algorithm in Fpocket. (A) Voronoi diagram of the atomic centers. **(B)** Voronoi balls (dotted orange circles) centered at Voronoi vertices (orange points). **(C)** Example of an α sphere (green sphere) centered at a Voronoi vertex (orange point) and grows until it becomes tangent to an atom. **(D)** Cluster of α spheres (green region) filling the binding site.

3.3.2.3 GHECOM

Grid-based HECOMi finder (GHECOM) [52], a grid-and-probe-based approach, identifies both deep and shallow pockets, employing an array of spherical probes. The cavity detection algorithm (Figure 3.5) combines fundamental erosion and dilation operations from mathematical morphology [57, 58] with different spherical probes to report openness-closedness of a target molecular shape, thereby revealing deep and shallow pockets (referred to as *multi-scale pockets*). Following that, a single-linkage clustering method groups pocket regions, allowing for the subsequent estimation of their respective volumes. GHECOM introduces a metric known as *pocketness*, which establishes a correlation between the volume and depth of points associated with individual residues or atoms. This metric serves as an informative indicator of their respective contributions to ligand interactions.



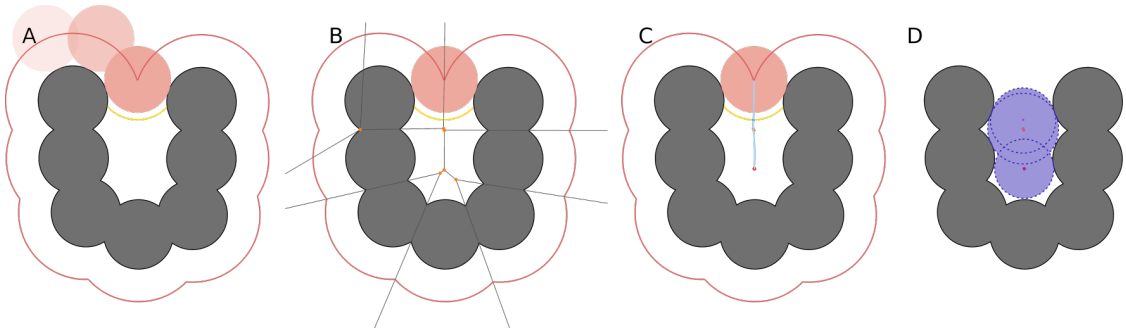
Source: Reprinted with permission from [36]. Copyright 2023 American Chemical Society.

Figure 3.5: Pocket detection algorithm in GHECOM. A biomolecular structure (green region; X) is enclosed by a spherical probe P (blue sphere), defining the region bounded by the yellow contour. Next, the intersection of X closed by P ($X \bullet P$) and the space outside the protein (X^c) defines the region not accessible to the probe P (blue region; $(X \bullet P) \cap X^c$). Subsequently, this region is opened by a spherical probe S (orange sphere), where P is larger than S. Finally, the pocket (orange region; $((X \bullet P) \cap X^c) \circ S$) is defined by the space outside the molecular shape not accessible to P but accessible to S. For multi-scale detection, different sizes of the spherical probe P are used.

3.3.2.4 CAVER tools

CAVER tools comprise a series of computational methods designed for tunnel and channel analysis within biomolecular structures. In the earlier version, CAVER [55], was a grid-and-surface-based approach for tunnel and channel calculation. Subsequently, CAVER 3.0 [49] replaced the axis-aligned grid with a Voronoi diagram approach, turning into a tessellation-based approach. The user-friendly interface, CAVER Analyst 2.0 [60], incorporates CAVER 3.0, providing visual assistance to users in tunnel and cavity calculations. The cavity detection algorithm (Figure 3.6) constructs a pseudo-Voronoi diagram of a given biomolecular structure. By identifying paths represented as graphs composed of Voronoi vertices and edges, CAVER 3.0 characterizes these paths as tunnels connecting cavities to the surrounding solvent. The resulting tunnels are further characterized by length, average radius, and bottleneck (mouth opening) radius. CAVER Analyst 2.0 complements these functionalities by identifying regions of empty space (i.e., cavities) within the biomolecular structure. Employing a similar approach to that described in the

KVFinder suite (Figure 3.3) and GHECOM (Figure 3.5), CAVER Analyst 2.0 determines regions accessible where a small probe can enter from outside, but a large probe cannot.



Source: Reprinted with permission from [36]. Copyright 2023 American Chemical Society.

Figure 3.6: Channel and tunnel detection algorithm in CAVER 3.0. (A) A molecular shape is probed by a spherical probe, called the *shell probe*, with a radius specified by the *shell radius* parameter, to define an outer surface SAS (red line). From it, a distance specified by the *shell depth* parameter is removed to define an inner surface (yellow line). **(B)** A pseudo-Voronoi diagram is constructed based on the molecular shape. Voronoi vertices (orange points) are used to create the central lines of the tunnel/channel. **(C)** A starting point (red point) is a user-defined parameter set as the center of mass of the molecular shape, and an endpoint (blue point) is set at the center of the inner surface. From the starting point, the central line passes through Voronoi edges and vertices to form a tunnel/channel to the outer surface and passes through the endpoint. **(D)** Spheres are fitted at all points of the central line, from the starting point to the endpoint, defining the bottleneck (mouth opening) radius along the tunnel and/or channel.

3.4 Computational Complexity

The effectiveness of cavity detection algorithms relies not only on their accuracy in identifying binding sites but also on their computational efficiency, particularly when dealing with biomolecular structures and large datasets. Computational complexity, a field studying algorithmic efficiency, involves analyzing the computational resources, such as time and space, required by an algorithm to solve a specific problem as a function of the input size. The goal is to understand how an algorithm's performance scales with increasing input sizes.

There are two primary aspects of computational complexity:

- **Time Complexity:** measures the time an algorithm takes to complete relative to the input size;
- **Space Complexity:** measures the memory (i.e., space) an algorithm needs to solve a problem based on the input size.

In computer science, both time and space complexity are often expressed using big-O notation ($\mathcal{O}$), also known as Bachmann-Landau notation. This notation describes the upper bound of an algorithm's running time and memory requirements, respectively, classifying algorithms based on how their performance grows relative to the input size.

In this scenario, the computational complexity of cavity detection algorithms is a critical aspect that influences their practical utility and scalability. Here, we briefly

delve into the computational complexity of the geometry-based approaches described in Section 3.3.1.

Grid-based algorithms (e.g., CavitySearch [37], POVME 3.0 [40]) rely on discretizing the molecular structure into a 3D grid. The computational complexity of these methods is inherently tied to the grid spacing and biomolecule size. The time complexity is typically $\mathcal{O}(n_v)$, where n_v represents the number of voxels, and it linearly increases with the number of voxels. In terms of user inputs, the number of voxels (Eq. 3.1) is expressed as follows:

$$n_v \colon (l_x, l_y, l_z, s) \simeq \left\lceil \frac{l_x}{s} \right\rceil \cdot \left\lceil \frac{l_y}{s} \cdot \right\rceil \left\lceil \frac{l_z}{s} \right\rceil \tag{3.1}$$

where l_x, l_y, l_z are the lengths of the target biomolecule along x, y, and z-axis, respectively, s is the grid spacing, and $\lceil \cdot \rceil$ is the ceiling function.

A finer grid provides a more detailed representation of the molecular surface but increases the number of grid points to be processed. Consequently, there exists a delicate balance between achieving sufficient resolution for accurate cavity detection and maintaining computational efficiency. The spatial complexity, referring to the memory requirements, is also $\mathcal{O}(n_v)$ as it scales with the number of voxels. Implementing strategies to optimize grid-based algorithms involves finding ways to reduce this linear relationship, such as through sparse grid representations or parallel processing.

Probe-based algorithms (e.g., PHECOM [43], pywindow [44]) introduce a distinct set of considerations, where the chosen probe size significantly influences precision and sensitivity in cavity detection. The time complexity of probe-based algorithms is often expressed as $\mathcal{O}(n_p \cdot n_a)$, where n_p is the number of probes, and n_a is the number of atoms. Conversely, spatial complexity depends solely on the number of atoms, denoted as $\mathcal{O}(n_a)$.

Surface-based algorithms (e.g., NSA [45], MSPocket [46]) focus on the molecular surface characteristics. The computational complexity of surface-based algorithms is associated with the surface representation and analysis techniques employed. These methods typically involve operations on the molecular surface mesh or point cloud, leading to complexities that depend on factors such as surface resolution, mesh complexity, and the nature of surface analysis. These complexities can range from linear (e.g., vertex transformation) to polynomial (e.g., convex hull computation), and optimizations often revolve around efficient mesh processing and surface characterization techniques.

Tessellation-based algorithms (e.g., Fpocket [48] and CAVER 3.0 [49]) leverage geometric techniques such as Voronoi diagrams and α spheres. The computational complexity here is intricately linked to the efficiency of these geometric operations. The time complexity of tessellation-based algorithms often involves complex geometric operations and is expressed as $\mathcal{O}(n_a \cdot \log n_a)$, where n_a is the number of atoms. Spatial complexity depends on the number of points (i.e., number of atoms) used in the geometric calculations, expressed as $\mathcal{O}(n_a)$.

Thus, systematic evaluation of the computational complexity inherent in a given algorithm presents their limitations but also unveils opportunities for optimization. The intrinsic complexity of biomolecular systems requires the development of efficient algorithms capable of handling large datasets. In this context, parallel computing emerges

as a promising avenue for enhancing the computational efficiency of computational biology algorithms, including tasks such as cavity detection, cavity characterization, solvent accessibility determination, and beyond.

3.5 Parallel Computing

Computational problems are typically divided into distinct parts, each comprising sets of instructions. In computing, two fundamental approaches exist: serial computing, which executes instructions sequentially, and parallel computing, which leverages multiple computational resources simultaneously (Figure 3.7). Parallel execution enables the simultaneous execution of instructions, thereby reducing computational time—a critical advantage in the context of modern computers featuring parallel architectures with multiple processors [61,62]. Despite its potential, parallel computing introduces new challenges that necessitate careful consideration. Coordinating parallel tasks, managing data synchronization, and minimizing communication overhead are critical concerns. Load balancing becomes pivotal to ensure uniform resource utilization, preventing bottlenecks that hinder performance gains. Additionally, designing algorithms that effectively parallelize tasks is a complex endeavor, requiring a deep understanding of both the problem domain and the intricacies of parallel architectures [62,63].

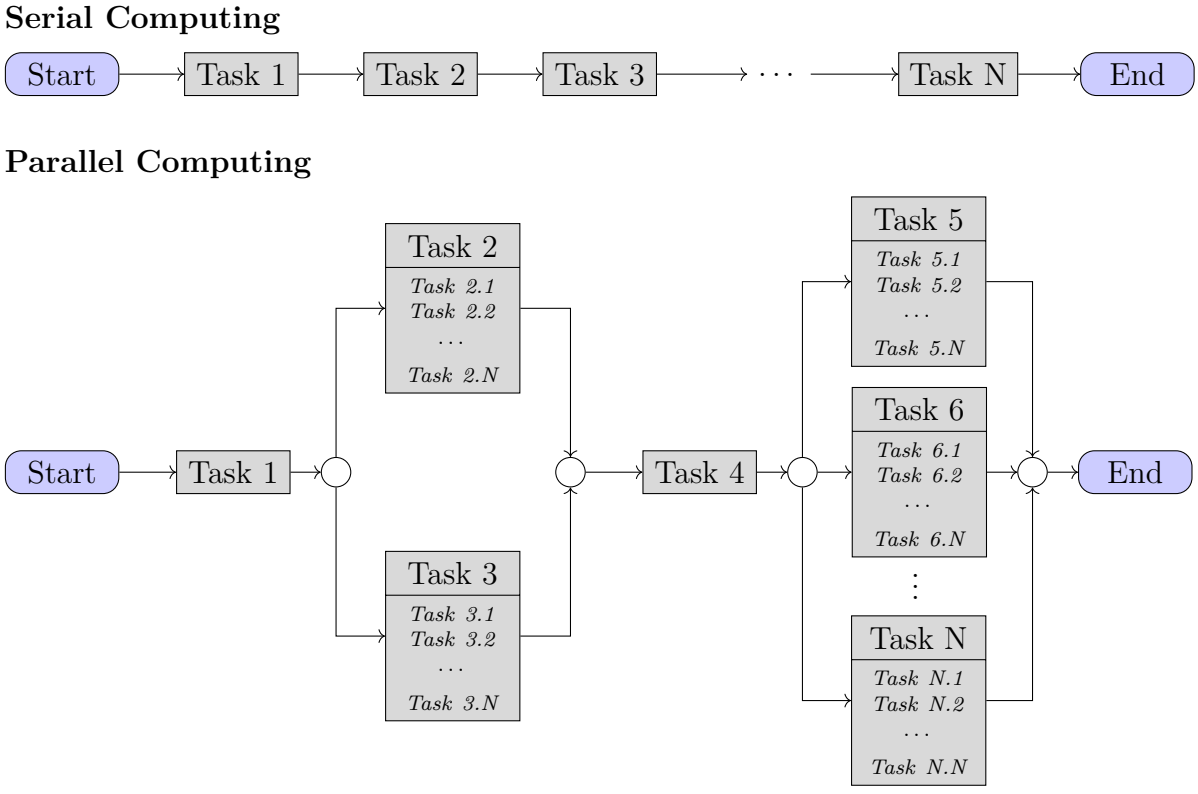


Figure 3.7: Diagram of serial and parallel computing.

Parallel computing marks a paradigm shift from traditional sequential approaches, employing multiple processors or computing units to simultaneously perform tasks. This paradigm has gained prominence in response to growing computational de-

mands, especially with the increasing complexity of data volumes and problem-solving requirements. Early parallel systems were tightly coupled, sharing memory and closely coordinating tasks. The evolution of parallel computing architectures progressed from Single Instruction, Multiple Data (SIMD) to Multiple Instruction, Multiple Data (MIMD) systems, reflecting a shift towards greater flexibility and efficiency. Diverse parallel computing architectures are designed to meet specific computational needs. Shared-memory architectures, such as Symmetric Multiprocessing (SMP), enable processors to access a common memory pool. In contrast, distributed-memory architectures, seen in clusters or grids, involve processors with their dedicated memory. Hybrid architectures combine these models, capitalizing on the strengths of both shared and distributed memory systems [63].

One key metric in evaluating parallel computing performance is speedup (Eq. 3.2), which is a relative measure used to analyze the efficiency gained by employing multiple processing elements compared to a single processor. The ideal speedup is linear, meaning that doubling the number of processing elements should ideally halve the computation time. However, achieving this ideal speedup is challenging, and many algorithms exhibit nearly linear speedup for a small number of processors, reaching an asymptotic behavior for a large number of processing elements [62, 63].

$$S\colon (T_1, T_p) \mapsto \frac{T_1}{T_p} \in [0, \infty) \tag{3.2}$$

where S is the speedup, T_1 is the execution time on a single processor, T_p is the execution time on p processors.

This performance gain can be predicted by two models (Figure 3.8): Amdahl's Law [64] and Gustafson's Law [65]. Amdahl's Law (Eq. 3.3), assumes the size of the computational problem is fixed, and the non-parallelizable fraction of the program is independent of the number of processors. Gustafson's Law (Eq. 3.4) addresses the deficiencies of Amdahl's Law, proposing that programmers tend to define the size of problems to exploit the computational power that becomes available as resources improve. In a way, Gustafson's Law redefines efficiency due to the possibility that limitations imposed by the sequential fraction of a program can be combated by an increase in available computational resources.

$$S_A\colon (f_p, N_p) \mapsto \frac{1}{(1 - f_p) + \frac{f_p}{N_p}} \tag{3.3}$$

$$S_G\colon (f_p, N_p) \mapsto (1 - f_p) + N_p \cdot f_p \tag{3.4}$$

where S_A is the speedup estimated by Amdahl's Law, S_G is the speedup estimated by Gustafson's Law, f_p is the parallelizable fraction of the code, and N_p is the number of available processors.

Following these rules, applications of parallel computing span various domains, from scientific simulations and data analytics to artificial intelligence. High-Performance Computing (HPC) clusters address intricate scientific problems, such as simulating climate models, conducting molecular dynamics simulations for drug discovery, or analyzing large-scale genomics data. On the other hand, parallel processing significantly accelerates

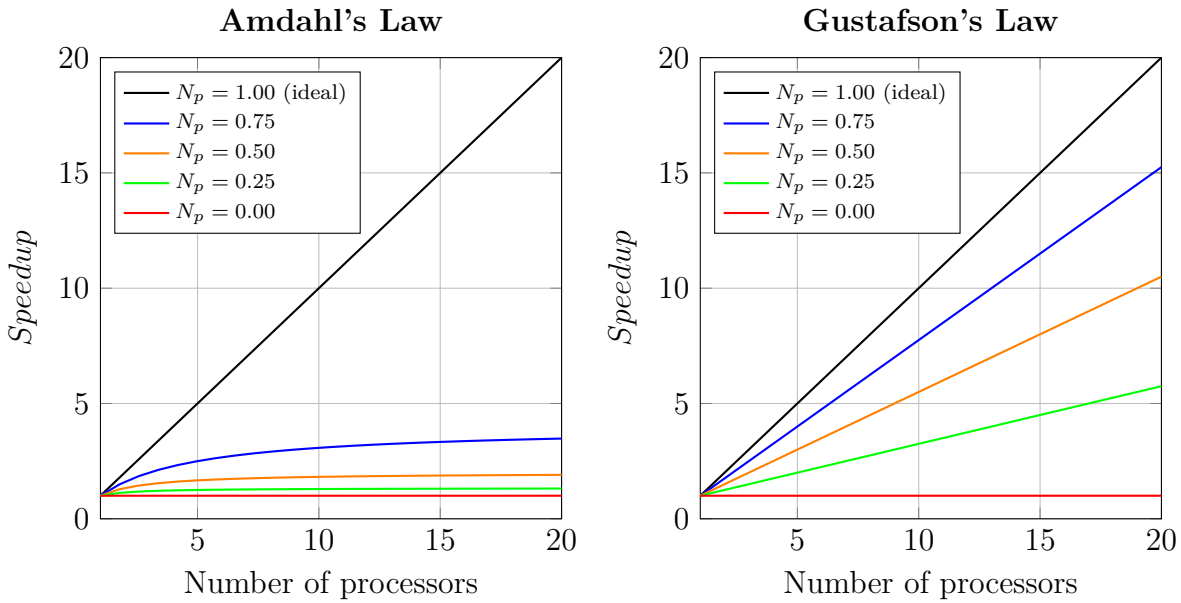


Figure 3.8: Speedup according to Amdahl's and Gustafson's Laws. The speedup is presented as a function of the number of available processors for different fractions of parallelizable code (N_p), considering the estimates of Amdahl's and Gustafson's laws.

tasks like training machine learning models for image recognition, natural language processing, and recommendation systems. Additionally, parallel algorithms play a crucial role in speeding up tasks like sorting, searching, and graph traversal, contributing to improved efficiency in various computational domains. The future of parallel computing is influenced by ongoing advancements in hardware, software, and algorithms. Emerging technologies, such as quantum computing, introduce new dimensions to parallelism. Ongoing research explores novel ways to harness parallelism efficiently, overcome scalability challenges, and integrate parallel computing into mainstream applications. In summary, parallel computing stands as a cornerstone in addressing the escalating demands for computational power. Its comprehensive application ensures the development of efficient computational platforms for different research fields.

Chapter 4: Data Coding in Structural Biology

Data coding, in essence, refers to the process of representing information or data using a specific set of symbols, codes, or conventions. This encoding allows for the efficient storage, transmission, and processing of data. In computer science, data coding can involve various techniques, such as character encoding (representing characters with numerical values), binary coding (using combinations of 0s and 1s to represent information), and other methods tailored to specific types of data.

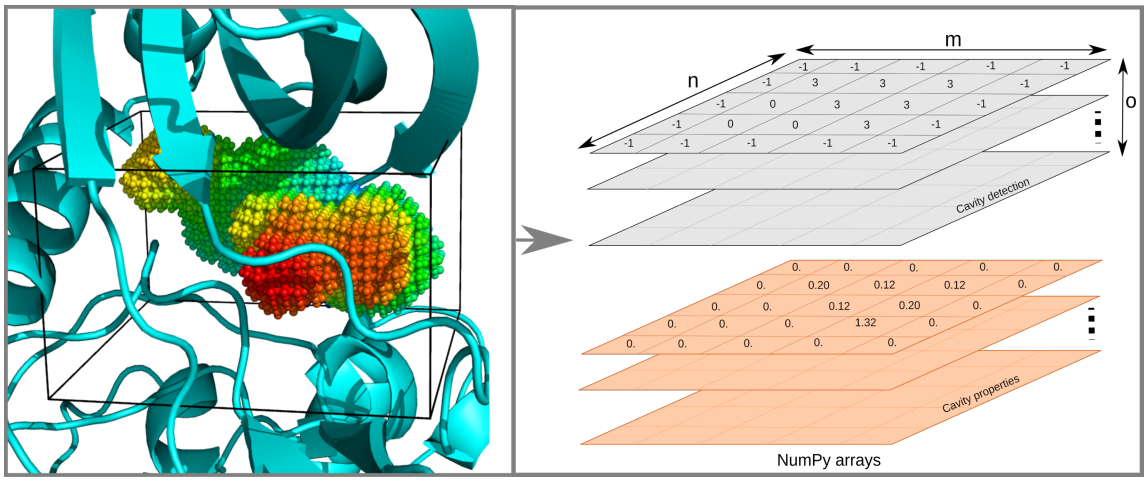
The data coding of biomolecules and binding sites plays a fundamental role in structural biology, enabling the computational representation, visualization and analysis of complex biological information [3]. This process converts biological information into formats that are understandable and suitable for processing by algorithms and programs in computational systems. Data coding involves assigning numerical or categorical codes to biological components, e.g., atoms, amino acids, and nucleotides, to describe occupancy, coordinates, forces, physicochemical characteristics, and/or structural properties.

In structural biology, data coding is important for performing advanced analyses, e.g., molecular modeling, molecular dynamics (MD) simulations, and interaction prediction. There are several applications that exemplify the biological data coding for computational analyses. For instance, in molecular modeling, a protein can be coded using one-letter codes to represent different amino acids, as seen in protein sequence representation, to predict protein folding, as applied in ESMFold [66], AlphaFold [67] and RosettaFold [68]. In MD simulation, the 3D coordinates of atoms or sets of atoms, along with vectors representing forces or other attributes, are used to represent the structure of a biomolecule, as in GROMACS [69] or AMBER [70] in atomistic-scale simulations and in CafeMol in coarse-grained molecular dynamics simulations [71]. Additionally, binding sites, once identified by computational algorithms, are coded in different ways for further analyses. Binding sites can be coded to represent the spatial and physicochemical features of a biomolecular region, such as the 3D grids used in the KVFinder suite [5, 8, 9, 36], or the 3D coordinates of atoms, as in Fpocket [48].

When working with computational models for the study of biomolecules, data coding is essential for the abstraction and computational representation of biological data. This abstraction is crucial for the development of computational tools aimed at studying biomolecular systems. Here, we present the data codings implemented for biomolecules and binding sites, i.e., volumetric representation, atomistic representation, and graph representation.

4.1 Volumetric Representation

Biomolecules and their binding sites can be represented in a 3D grid, organized into volumetric pixels, commonly known as voxels. This grid, akin to a matrix (e.g., a NumPy array), assigns values to each voxel, encapsulating a discrete point of data within the 3D grid. Each voxel, acting as a discrete data point, can contain multiple types of information, enabling a straightforward and efficient representation of various properties within a specific portion of space (Figure 4.1). In computer science, the 3D grid is a widely employed data structure in image processing and computer vision applications, including tasks such as image reconstruction, image segmentation, and object detection.



Source: Reprinted from [5]. Licensed under CC BY 4.0.

Figure 4.1: Grid representations. Based on a 3D grid with dimensions (m, n, o), each element corresponds to a region of cavity (>1), empty space (1), biomolecule (0), or solvent (-1). Additionally, properties are stored in the same data structure, corresponding to the property value in the region.

Among various molecular surface representations, the 3D grid composed of voxels is the simplest and most suitable for representing multiple properties under various conditions, as each voxel in the 3D grid can store multiple information, e.g., charge distribution, burial extent (depth), hydrophobicity, hydrogen-bond-forming regions and cavity identifier. Moreover, the 3D grid is an efficient data structure for storing and accessing values and/or attributes at a 3D position, enabling the performance of mathematical and logical operations within a 3D region. Its versatility makes it an ideal choice for the development of computational algorithms, leveraging parallel computing techniques to accelerate the processing of large datasets.

The computational complexity of grid-based algorithms heavily relies on the input size, i.e., the number of voxels. Higher density grids offer a more detailed spatial representation, which proves beneficial for in-depth studies but comes at the cost of increased computational requirements, i.e., increased memory consumption and poorer time performance. For cavity detection algorithms, efficiency is often assessed based on the linear scalability of time complexity with the number of voxels, deeming exponential scalability inefficient. Within this context, the voxel clustering algorithm emerges as the most time-intensive step, primarily due to its limited parallelizability.

In the parKVFiner software, the voxel length is a user-defined parameter, with default value of 0.6 Å, providing the flexibility to adjust this value based on the specific demands of their case studies. For instance, a smaller voxel length, such as 0.25 Å, is recommended for those seeking a more accurate morphological characterization of biomolecular cavities. The computational complexity is intricately tied to the number of voxels (n_v), denoted as $\mathcal{O}(n_v)$. User inputs determine the number of voxels (Eq. 4.1), calculated as follows:

$$n_v \colon (l_x, l_y, l_z, Po, s) \mapsto \left\lceil \frac{l_x + 2 \cdot Po + s}{s} \right\rceil \cdot \left\lceil \frac{l_y + 2 \cdot Po + s}{s} \right\rceil \cdot \left\lceil \frac{l_z + 2 \cdot Po + s}{s} \right\rceil \tag{4.1}$$

where l_x, l_y, l_z are the lengths of the target biomolecule along x, y, and z-axis, respectively, Po is the *Probe Out* size, s is the grid spacing, and $\lceil \cdot \rceil$ is the ceiling function.

Under typical conditions, the average dimensions of the 3D grid, denoted as $(\bar{m}, \bar{n}, \bar{o})$, are roughly $(120, 120, 120)$, along the x, y, and z axes, respectively. The computational complexity increases as the region of interest, dictated by the biomolecule and the probe size, influences the number of voxels, where the voxel-level analysis, inherent to the algorithm, amplifies computational demands.

Turning to spatial complexity, the number of voxels and the bit-depth of grid elements significantly influence memory usage, with the latter also impacting precision. The choice between int (32-bit integer) and double (64-bit floating-point) in the C-coded parKVFinder reflects a balance between memory efficiency and the precision required for accurate calculations. Spatial complexity scales linearly with the number of voxels (n_v), denoted as $\mathcal{O}(n_v)$. In parKVFinder, the grid for cavity and surface points identifiers is an integer grid (32-bit integer), while cavity properties, such as point depth and hydrophobicity, are stored in a double grid (64-bit floating-point).

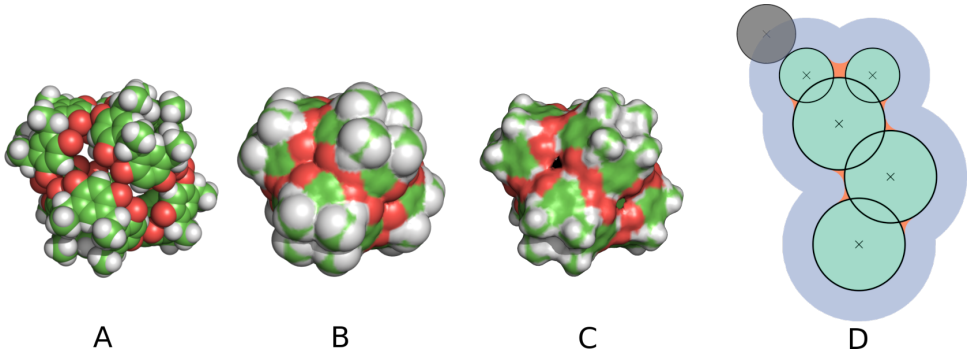
4.1.1 Molecular Surface Representation

The representation of molecular surfaces is a crucial step in the modeling and analysis of biomolecules. In this approach, biomolecules are described through a rigid sphere model, considering atomic 3D coordinates and radii to model the molecular surface. There are three commonly used mathematical formulations to represent molecular surfaces (Figure 4.2):

(A) **vdW surface:** models each atom as a hard sphere with a radius proportional to its vdW radius. The vdW surface is formed by combining these individual atoms;

(B) **SAS:** models the molecular surface accessible to a solvent molecule (e.g., water), which is approximated by a sphere;

(C) **SES:** is similar to SAS but considers the outer shell of the sphere instead of its center.



Source: Reprinted with permission from [36]. Copyright 2023 American Chemical Society.

Figure 4.2: Molecular surface representations. (A) vdW surface. **(B)** SAS. **(C)** SES. Images generated with PyMOL for the supramolecular cage (resorcin[4]arene-hexameric). **(D)** 2D schematic representation of molecular surfaces. The vdW surface (green) consists of atoms represented as green spheres. A spherical probe (gray), representing a solvent molecule, rolls over the atoms of the molecule to define SES and SAS. SES is defined by the vdW surface (green) and the space not reached by the spherical probe (orange). SAS is defined by the envelope reached by the center of the spherical probe (blue).

4.2 Atomistic Representation

In our exploration of structural and functional characterization, we now turn to atomistic representation, an alternative to the voxel-based approach. Rather than representing structural data through voxels in a 3D grid, biomolecules and their binding sites can also be portrayed through their atomistic representation (Figure 4.3). This representations are applied in tools of molecular dynamics simulations, such as GROMACS [69], AMBER [70], and CafeMol [71]. Here, atoms, residues, and/or nucleobases are modeled as hard sphere models in 3D coordinates (x, y, z), along with vectors (e.g., forces, velocities, and accelerations) and properties (e.g., mass, charge, and van der Waals radius).

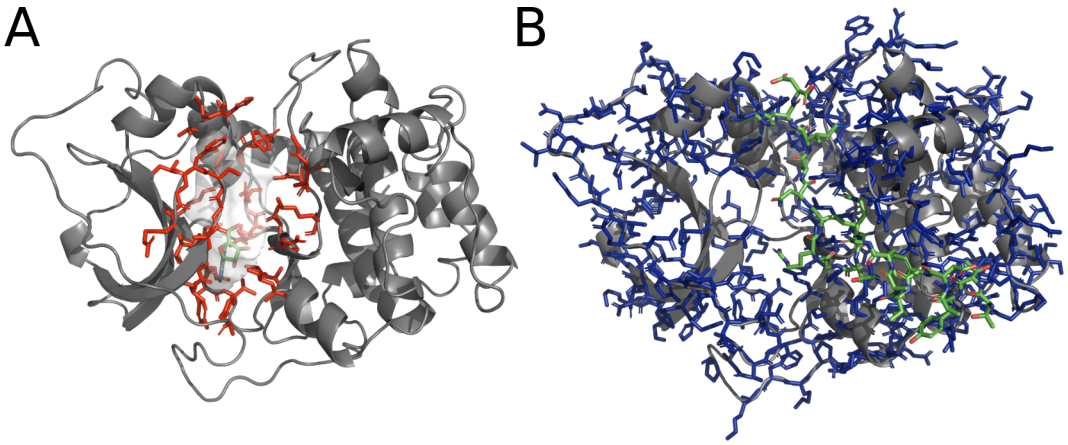


Figure 4.3: Atomistic representations. (A) Atoms and amino acids (*sticks* in green) forming the adenosine binding site (transparent surface). **(B)** Atoms and amino acids (*sticks* in blue) exposed to the solvent, excluding binding sites for small molecules, with an inhibitor bound (*sticks* in green). Images generated with PyMOL for a subunit of cyclic AMP-dependent protein kinase (PDB ID: 1FMO).

Yet, when delving effectively into interaction regions, a prerequisite emerges—the filtration of areas of interest, such as binding sites (Figure 4.3A) or solvent-exposed

regions (Figure 4.3B). This atomistic representation allows precise and in-depth analysis, focusing on interactions and structural features relevant to biological function. Additionally, atomistic information can be utilized in molecular docking studies, rational drug design, and prediction of molecular interactions. These applications can contribute to the development of new therapeutic compounds and aid in understanding the molecular mechanisms involved in biological processes.

Parallel to the challenges encountered in volumetric representation, the computational complexity of algorithms based on atomistic representations hinges on the input size—specifically, the number of atoms. Researchers must balance the level of detail required for their analyses with the computational resources available, opting for fine-grained (all-atom), coarse-grained or carbonα (backbone) models. Spatial complexity is linearly dependent on the number of atoms (n_a), denoted as $\mathcal{O}(n_a)$.

Within parKVFinder, the atomistic representation encodes each atom by its residue number (32-bit integer), chain identifier (char), residue name (4-char array), atom type (char), coordinates (64-bit floating-point), and vdW radius (64-bit floating-point). Concurrently, interface residues are represented by residue number (32-bit integer), chain identifier (char) and residue name (4-char array). In both instances, spatial complexity significantly diminishes compared to volumetric representation, given the fewer atoms or interface residues compared to voxels for the same target biomolecule. When applying atomistic representation, researchers adopt the biomolecule's perspective, in contrast to the cavity point-of-view inherent in volumetric representation.

4.3 Graph Representation

Expanding upon the atomistic representation, the utilization of graph-based models emerges as a robust representation for examining interactions and relationships within biomolecules. Graph theory techniques are widely applied across diverse scientific domains such as biology, chemistry, physics, computer science, and mathematics [72,73]. A cornerstone in discrete mathematics, it models the relationships and interactions between objects in a network, assisting in the analysis of biological interaction networks at various scales. In structural biology, graph theory has been utilized to investigate the structure, folding, stability, function, allostery and dynamics of proteins [74–77].

A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (Figure 4.4) is a mathematical structure, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is a nonempty, finite set of vertices (also known nodes, point and junction), and $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ is a set of unordered pairs of distinct vertices of $\mathcal{V}$, whose elements $\{v_i, v_j\} \in \mathcal{E}$, where $v_i, v_j \in \mathcal{V}$, connects vertices v_i and v_j and termed edges (also known as line, arc, branch and link). Graphs can be directed (also known as digraph), where edges link two vertices symmetrically, or undirected, where edges link two vertices asymmetrically. Additionally, graphs can also be weighted, where each edge is assigned a weight or cost, which can represent a physical distance, a similarity measure, or any other property [72, 73, 78, 79]. Evolving from this, graph theory finds applications in modeling biological and chemical systems. In chemistry, molecular graphs capture the structural relationships between atoms and bonds, facilitating the study of chemical compounds

and reactions. Biomolecules, which are assemblies of atoms (vertices) connected by intramolecular and intermolecular interactions (edges), have also been extensively explored through graph theory [74, 80].

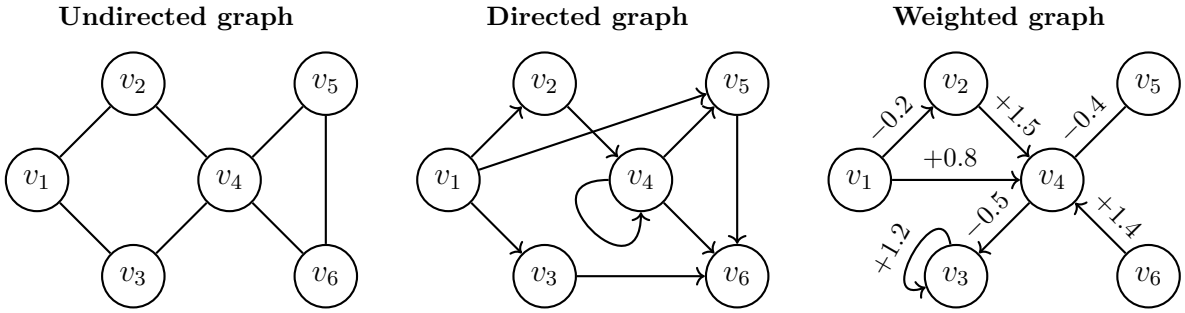


Figure 4.4: Pictorial overview of graphs $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

In this context, we introduce a graph-based representation for biomolecules using the 3D coordinates (x, y, z) of a biomolecular structure or complex that generates a residue-level graph, where vertices represent residues and edges represent interactions or some type of relationship between them. The construction of edges is based on customizable distance cutoffs between atoms, such as α carbon (Cα), β carbon (Cβ), or any other atom, which can be defined by the user (Figure 4.5). According to Critical Assessment of PRedicted Interactions (CAPRI) Round 28 [81, 82], a distance cutoff of 8 Å between any two Cβ atoms (or Cα for Gly) define interface residues, and a distance cutoff of 4 Å between any two atoms define native contacts. Expanding on the Cβ definition, a distance cutoff of 10 Å between any two Cα atoms to define interface residues. Thus, the default parameters set in the implementation are these distance cutoffs, establishing relationships or interactions between residues [74, 80]. Figure 4.5A shows an example of a residue-level graph generated from an adenosine binding site (Figure 4.3A), and Figure 4.5B shows an example of a residue-level graph generated from a solvent-exposed surface (Figure 4.3B).

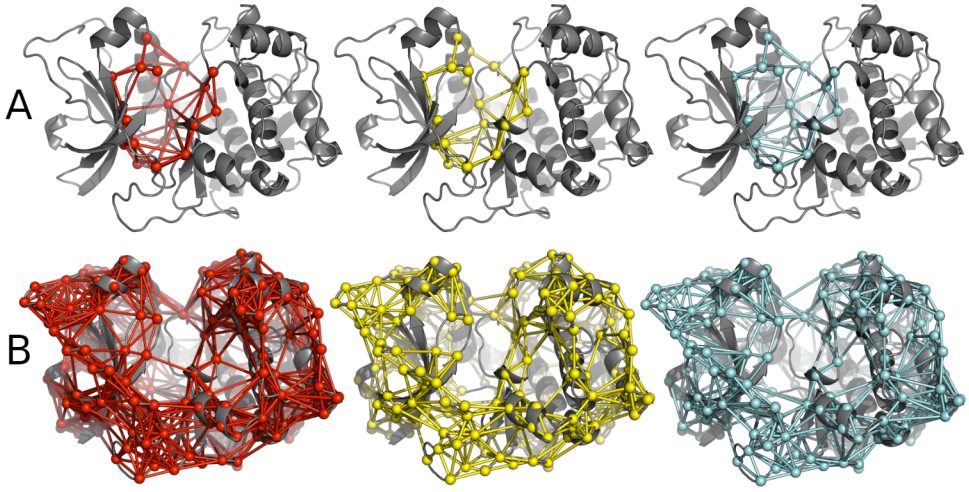


Figure 4.5: Graph representations. (A) Adenosine binding site as graphs with edges based on the distance cutoff of Cα (spheres and sticks in red), Cβ (spheres and sticks in yellow), and any atoms of the residues (spheres and sticks in cyan). **(B)** Solvent-exposed surface, excluding binding sites for small molecules, as graphs with edges based on the distance cutoff of Cα (spheres and sticks in red), Cβ (spheres and sticks in yellow), and any atoms of the residues (spheres and sticks in cyan). Images generated with PyMOL for a subunit of cyclic AMP-dependent protein kinase (PDB ID: 1FMO).

This graph-based representation allows a more efficient analysis of interactions and relationships between residues, providing a visually intuitive exploration of the structural and functional attributes of the biomolecule. Additionally, various properties and metrics can be calculated from the graphs, such as paths, distances, centrality, and other metrics that aid in understanding the structure and function of the biomolecule [73,74,80]. Since our primary focus is on the adjacency matrix, spatial complexity depends on the square of the number of vertices (i.e., number of residues—n_r), denoted as $\mathcal{O}(n_r^2)$. However, as we include attributes (weights) on vertices and edges, spatial complexity will depend on the size of this additional data. With this graph-based representation in hand, novel mathematical approaches from graph theory can be applied to deepen our understanding of biological function and provide crucial insights for the development of therapies and therapeutic interventions.

Chapter 5: KVFinder suite

The interactions between biomolecules play a crucial role in biological processes, involving entities ranging from small molecules such as ions and drugs to macromolecules such as proteins and nucleic acids. These receptor-ligand interactions (e.g., PPIs, PLIs, PRIs, and PDIs) take place at specific binding sites, which can be solvent-exposed clefts or cavities buried within the receptors. Morphological, topological, and physicochemical complementarity between ligands and receptors governs molecular recognition, limiting efficient interaction to a finite number of ligands. The identification and assessment of these regions are fundamental for understanding the tertiary structure of biomolecules and for the development of new drugs. To meet this demand, we have developed the computational platform **KVFinder suite**, which combines precise tools with high-performance processing, enabling the analysis of biomolecular experimental data and the comprehension of the biomolecular structure and function in biological systems.

The KVFinder suite consists of five computational tools that offer comprehensive functionalities for structural analysis and the study of biomolecular interactions. The tools included in the platform are: parKVFinder [8, 9], pyKVFinder [5], KVFinder-web [10], SERD, and KVFinderMD. Next, we will describe each of these tools, their main features, and their guideline applications.

5.1 parKVFinder

The parallel KVFinder (parKVFinder) [9] is an open-source tool, licensed under GPL v3.0, developed for the detection and characterization of any type of biomolecular cavity. Originating from the master's thesis entitled "Prospecção e caracterização de cavidades supramoleculares" [8], it presents itself as a refactored, optimized and parallelized successor of KVFinder [4]. The initial release, parKVFinder v1.0, was published in *SoftwareX* [9], with morphological (i.e., shape, volume, and area) and constitutional (i.e., interface residues surrounding cavities) characterizations. A notable addition was a novel algorithm for surface area estimation, based on Mullikin Verbeek voxel classification. Progressing further, parKVFinder v1.2.0 expanded its capabilities with additional morphological (i.e., depth), physicochemical (i.e., Eisenberg Weiss hydrophobicity [28]), and constitutional (i.e., residue frequency) characterizations [10].

The parKVFinder software is integrated with the PyMOL molecular viewer [56] through a graphical plugin. The original plugin, PyMOL parKVFinder Tools, was developed for PyMOL v1.8 [8, 9]. However, with the advent of PyMOL v2.0 by Schröndinger,

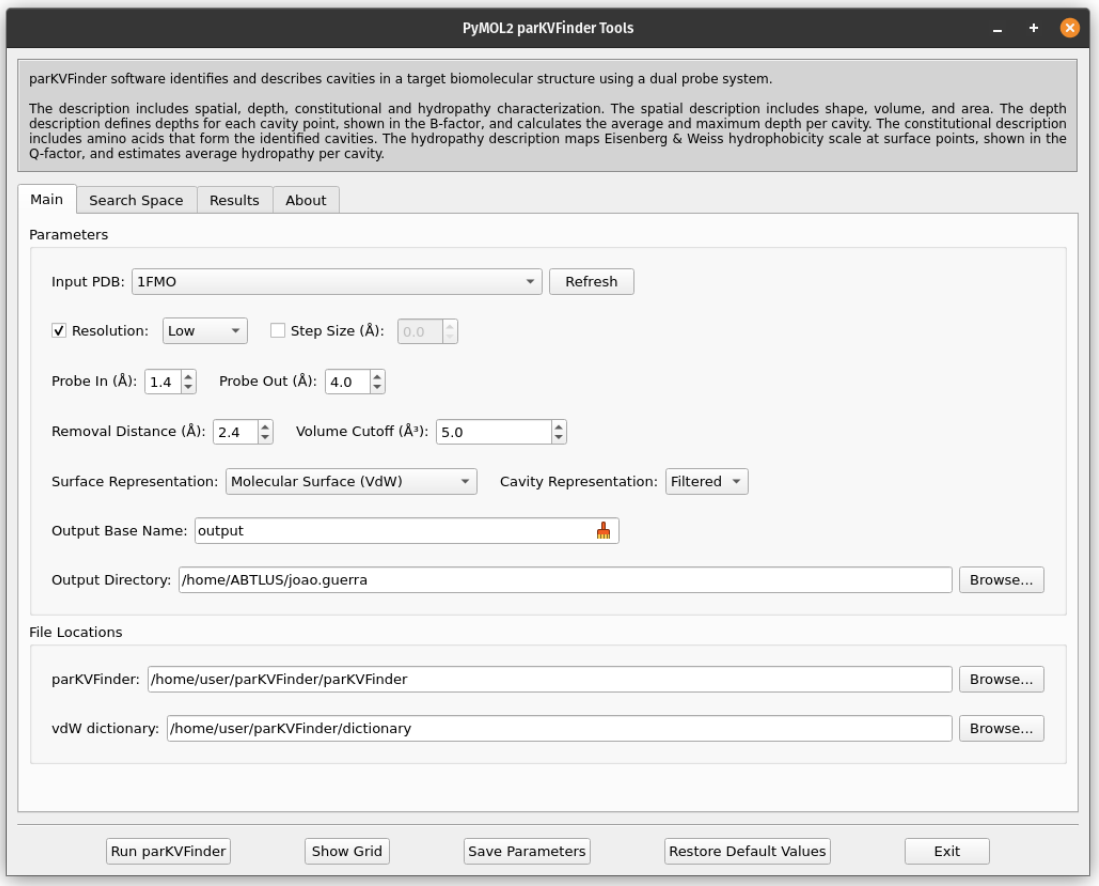


Figure 5.1: PyMOL2 parKVFinder Tools. The *Main* tab presents the options that directly control the cavity detection procedure, in which user can set *Probe In* and *Probe Out* sizes, grid spacing, removal distance, volume filter, surface representation and cavity representation. Space segmentation features are managed in the *Search Space* tab, allowing users to designate either the entire structure or custom search spaces. In box adjustment mode, an interactive box is drawn in the PyMOL viewer. Additionally, users can opt for ligand adjustment, limiting the search space around a defined ligand (PyMOL object) to a user-specified radius. After running parKVFinder, results (volume, surface area, depth, hydropathy, and interface residues) are interactively accessible in the *Results* tab, and the cavity PDB file seamlessly loads into the PyMOL viewer.

which discontinued support for the 1.8 version, a new plugin was developed—PyMOL2 parKVFinder Tools (Figure 5.1), written in Python3 with Qt interface. This update plugin incorporates new characterizations (depth and Eisenberg Weiss hydrophobicity) from parKVFinder v1.2.0. This new plugin provides an intuitive and easy-to-use graphical user interface (GUI), allowing users to customize parameters for the detection and characterization of cavities. Additionally, users can visualize identified cavities and their characteristics directly within the PyMOL environment (Figure 5.2). Alongside the GUI, parKVFinder also serves advanced users with a command-line interface (CLI), enabling task automation and program integration. The source code for parKVFinder and the PyMOL plugins is publicly available in the following repository: <https://github.com/LBC-LNBio/parKVFinder>.

Cavity detection and characterization routines in parKVFinder leverage the OpenMP library for parallelization, optimizing performance on multicore systems. Computational assessments on a set of 1000 protein domains (kv1000, <https://github.com/

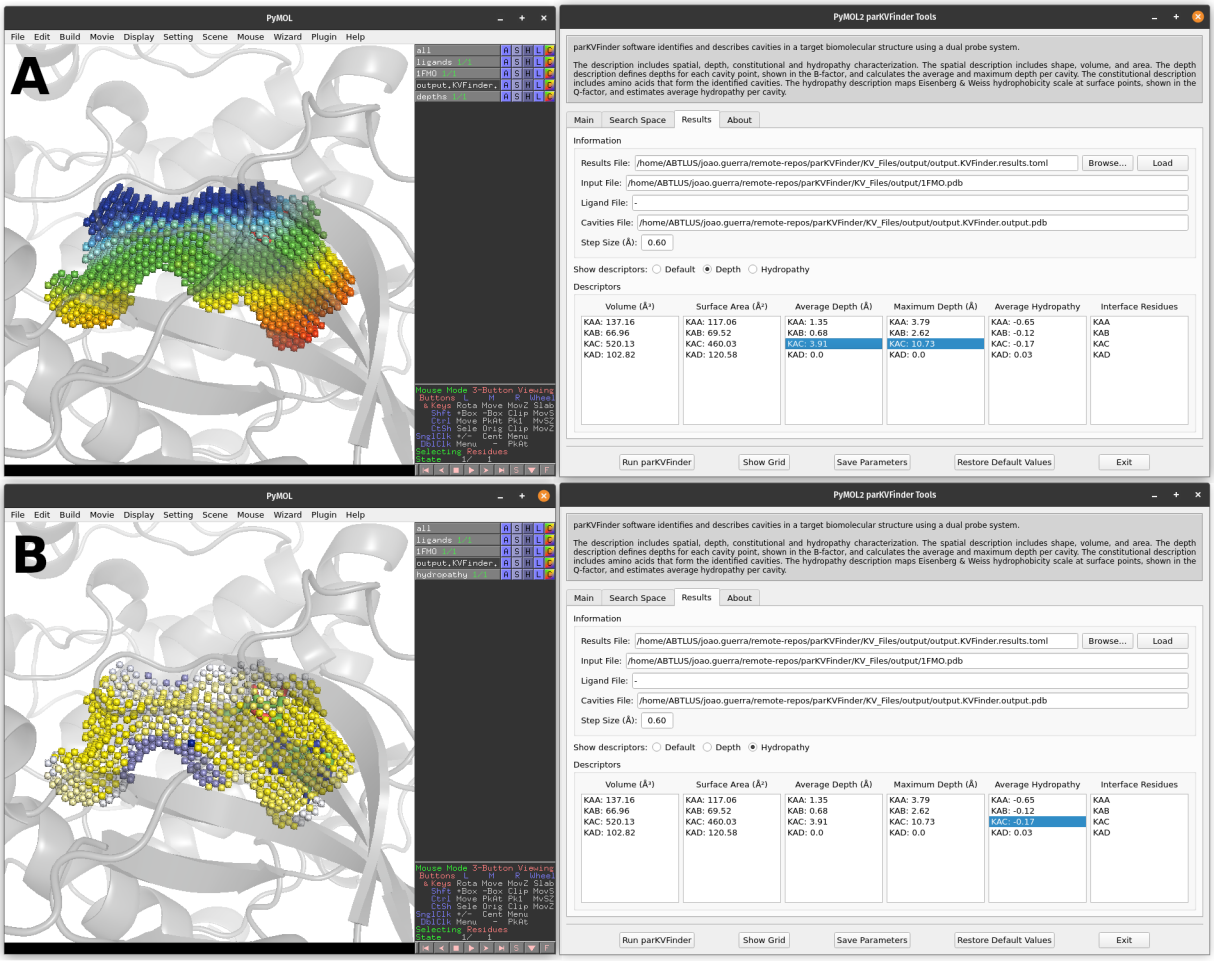


Figure 5.2: Interactive view of PyMOL2 parKVFinder Tools in PyMOL. Cavity characterization of the adenosine binding site in a protein kinase A (PDB ID: 1FMO). **(A)** Depth characterization: cavity points are colored by depth, ranging from superficial (blue points) to buried (red points). **(B)** Hydropathy characterization: surface cavity points are colored by Eisenberg Weiss hydrophobicity scale, from −1.42 (highly hydrophobic) to 2.6 (highly hydrophilic).

jvsguerra/kv1000>) revealed significant runtime reductions compared to KVFinder. The code optimization yielded a speedup of ∼1.5 times [8] compared to KVFinder, and the subsequent code parallelization in parKVFinder led to a speedup of ∼6.4 times in 24 OpenMP threads [9]. Overall, with 24-thread parallelization, parKVFinder demonstrated a remarkable ∼9.5 times reduction in runtime compared to KVFinder [8,9]. These calculations were performed on a computer with two 6-core (12 threads) 2.67GHz Intel Xeon CPU X5650 and 19 GB RAM, running Ubuntu 18.04 operating system.

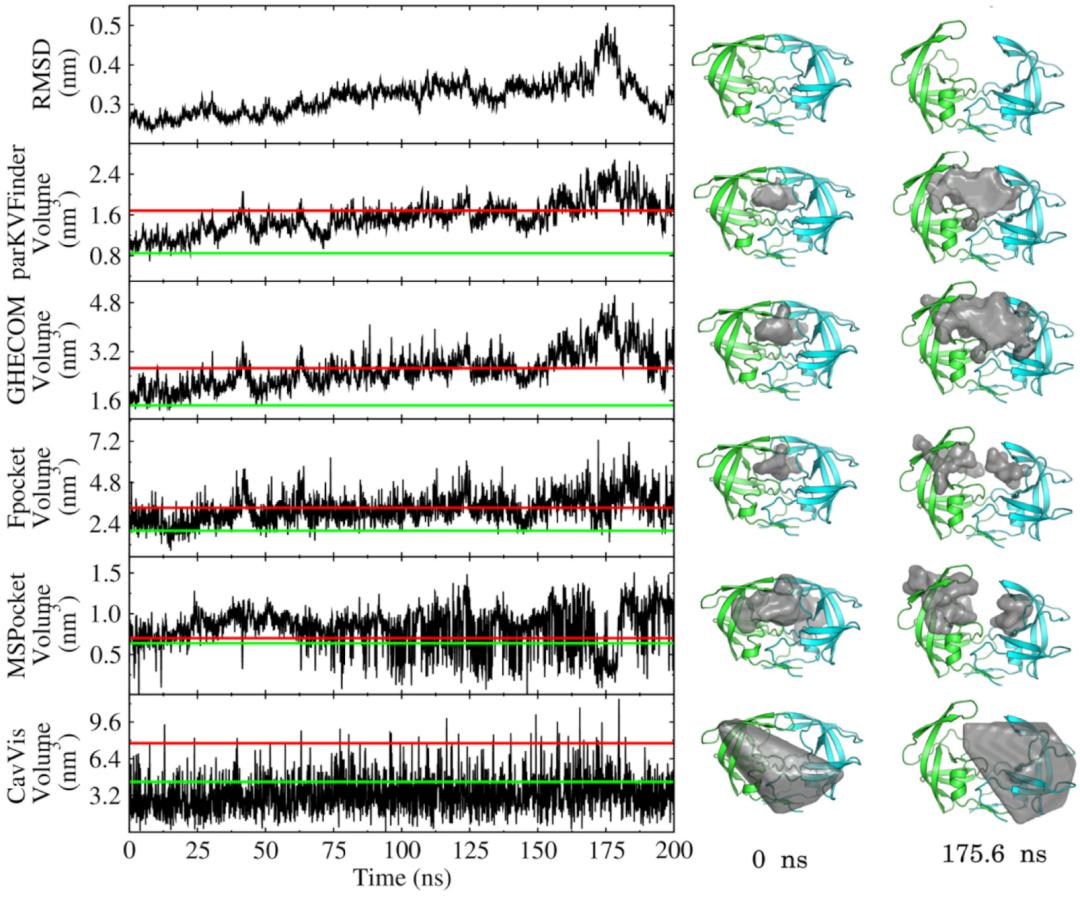
5.1.1 Case Studies

The parKVFinder was applied in two case studies published in scientific journals to investigate proteins of therapeutic interest. These analyses explored the dynamics of the flaps of the Human Immunodeficiency Virus type 1 (HIV-1) protease [9] and the hydrophobicity profile of binding sites in alphaviruses [83]. Next, we will describe each of these case studies in detail.

5.1.1.1 Molecular Dynamics of HIV-1 Protease

The active site of the HIV-1 protease represents a relevant therapeutic target for various antiretroviral drugs. This binding site exhibits significant complexity, which arises from the movement of β-hairpins, known as flaps, leading to structural and morphological variations. The volume of the binding site changes with the motion of these flaps, influencing substrate accessibility to the active site of the homodimer [84, 85].

In this study, the MD of the HIV-1 protease was investigated by identifying and evaluating the active site's volume during MD simulations. Our objective was to describe the dynamic movement of the flaps through the volume and shape of the active site cavity as conformational descriptors in MD simulations. To achieve this, 200 ns MD simulations were conducted using the GROMACS 2019.4 package [69], the AMBER99SB-ws force field, and the TIP42005s water model. The simulations initiated from the crystallographic structure of HIV-1 protease in the closed conformation [84], excluding the inhibitor present in the structure, cyclic urea.



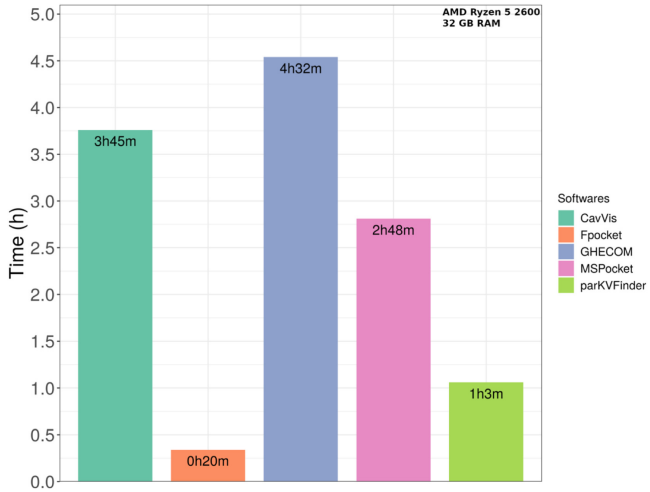
Source: Reprinted from [9]. Licensed under CC BY 4.0.

Figure 5.3: Volume of the HIV-1 protease active site over a 200 ns molecular dynamics simulation. The green and red lines indicate the cavity volume for the closed (PDB ID: 1HVR) and semi-open (PDB ID: 1HHP) states, respectively. Protein structures at the beginning of the simulation (0 ns) and the frame with the highest RMSD (175.6 ns) are shown as cartoons. The corresponding cavities detected by each tool are displayed as gray surfaces.

The active site volume was monitored throughout the MD simulations (Figure 5.3). Initially, the cavity volume corresponds to the closed conformation (green line).

After ∼25 ns, it increases, reaching the value corresponding to the semi-open conformation (red line) around 75 ns, indicating an opening process. Subsequently, the flaps separate further, and the cavity achieves its maximum volume at ∼175 ns before reverting to the more stable semi-open conformation. This dynamic volume change correlates well (Pearson correlation; $\rho = 0.72$) with the Root-Mean-Square Deviation (RMSD) of the Cα calculated from the closed conformation, providing an accurate depiction of the protein's conformational state throughout the simulation. It is noteworthy that RMSD describes global structural variations, while the estimated volume offers a direct metric of conformational changes in the active site, possibly associated with ligand accessibility.

The performance of parKVFinder [9] was benchmarked against other geometry-based methods (Figure 5.3), including GHECOM [52], Fpocket [48], MSPocket [46], and CavVis [86]. The correlation between the estimated volume by each program and the RMSD was evaluated, replicating the approach used for parKVFinder. The volume estimated by GHECOM ($\rho = 0.75$) correlates with the conformational state, similar to parKVFinder ($\rho = 0.72$), likely due to their shared use of grid-and-probe-based methods. However, the cavities identified by Fpocket ($\rho = 0.35$), MSPocket ($\rho = -0.24$), and CavVis ($\rho = 0.19$) did not show a satisfactory correlation with the conformational dynamics of the active site. Therefore, parKVFinder and GHECOM demonstrated high accuracy in describing the conformational state of the HIV-1 protease active site.



Source: Reprinted from [9]. Licensed under CC BY 4.0.

Figure 5.4: Computational time of the benchmarking methods.

Beyond accuracy, the computational time of the programs was also assessed (Figure 5.4). parKVFinder ($t = 1h03m$), employing 12 OpenMP threads, was at least four times faster than GHECOM ($t = 4h32m$), due to the multi-threaded subroutines implemented in parKVFinder. Furthermore, parKVFinder outperformed MSPocket ($t = 2h48m$) and CavVis ($t = 3h45$) in terms of computational time but did not surpass Fpocket ($t = 20m$), which uses Voronoi tessellation and α spheres methods, resulting in fast computations. However, this implementation of Fpocket proved less sensitive in the detailed description of the HIV-1 protease active site, inefficiently distinguishing the conformational states of the active site. These calculations were performed on a workstation

with a 6-core (12 threads) 3.4 GHz AMD Ryzen 5 2600 processor and 32 GB RAM, running Ubuntu 19.04 operating system. Therefore, considering both accuracy and performance, parKVFinder emerged as a robust option for the spatial detection and characterization of cavities in the case study of the HIV-1 protease [9].

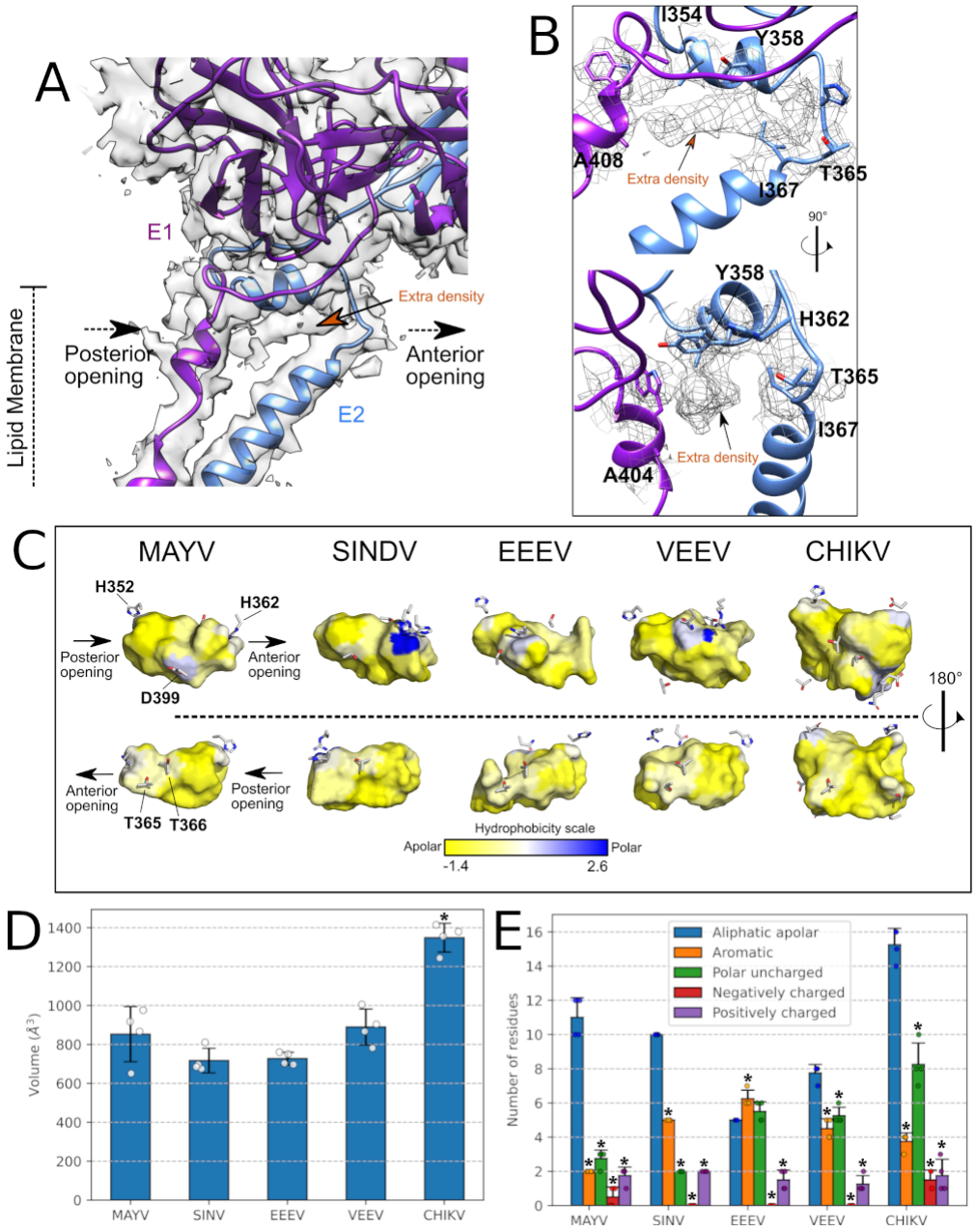
It is important to highlight that the details of the simulations and analyses are available in the article published in the *SoftwareX* [9].

5.1.1.2 Mayaro and Other Alphaviruses

Mayaro virus (MAYV) is an emerging arbovirus prevalent in Central and South America, associated with a debilitating and arthritogenic disease. The ectodmains E1 and E2 are essential transmembrane alphaviral proteins that form heterodimers. These heterodimers, organized into trimers, compose the spikes on the viral surface, extending through the lipid bilayer and interacting with the nucleocapsid proteins C. These spikes play a crucial role in binding to cellular receptors, cell internalization, and membrane fusion. The subsquent release of MAYV RNA into the cytoplasm triggers viral protein expression, replication, and the production of mature and infectious viral progeny. Given their central role, the E1 and E2 proteins of MAYV are key targets for vaccine development and antiviral drugs [83]. However, the lack of detailed structural information on these proteins has hindered the development of effective strategies to combat MAYV infection.

The central region between the E1 and E2 proteins forms a cavity occupied by an extra-long density, not accounted for by side-chain residues (Figure 5.5A and B). A similar density profile was previously observed in a cryo-electron map of the Sindbis virus (SINV), suggesting a hydrophobic phospholipid tail (C18), termed a *pocket factor*, might occupy this density and stabilize the hydrophobic pocket formed between E1 and E2 [87]. To gain a deeper understanding of the pocket environment and extract its chemical characteristics, we employed parKVFinder [9] to comprehensively characterize the MAYV pocket and compare it with pockets in other alphaviruses. The E1 and E2 ectodomains of MAYV (PDB ID: 7KO8), SINV (PDB ID: 6IMM), Eastern Equine Encephalitis virus (EEEV) (PDB ID: 6MX4), Venezuelan Equine Encephalitis virus (VEEV) (PDB ID: 3J0C), and Chikungunya virus (CHIKV) (PDB ID: 6NK5) were used for the detection and characterization of the hydrophobic pocket. In MAYV, the cavity between the E1 and E2 domains has a volume of ∼850 Å^3 (Figure 5.5D). This volume is quite similar across SINV, EEEV, and VEEV, while in CHIKV, the larger distance between E1 and E2 results in a larger volume.

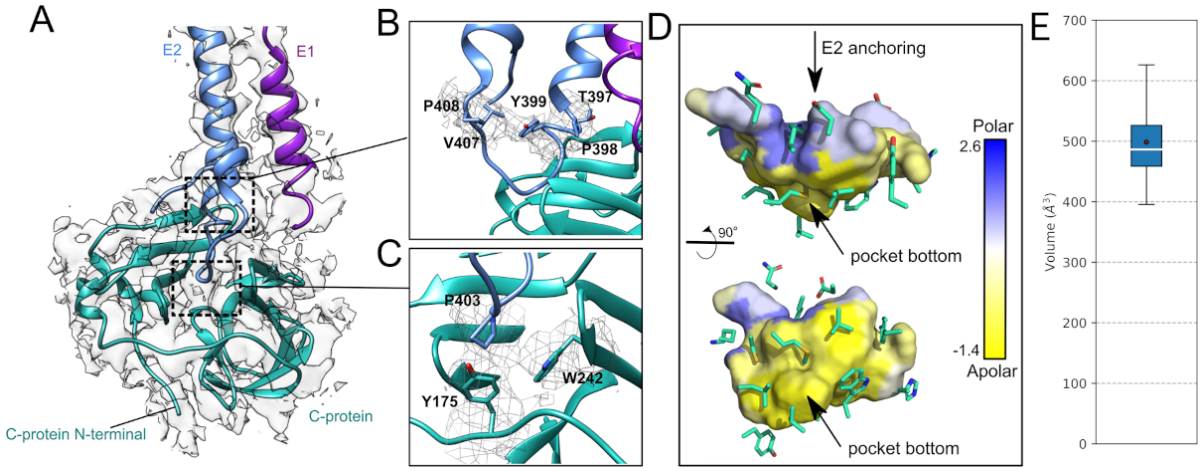
The hydrophobic nature of the alphavirus pocket is clearly observed by mapping the hydrophobicity surface (Figure 5.5C) and the number of apolar residues forming the core of the cavity (Figure 5.5E). The pocket density extends to polar residues, such as H362, T365, and T366 from the E2 domain at the posterior opening (Figure 5.5C and E), indicating that the molecule may have an amphipathic nature, such as a fatty acid. Notably, T365 and T366 maintain a structurally conserved position in other alphaviruses or are replaced by serine, an even more polar residue. At the posterior opening, another histidine (H352 from E2) helps close the pocket, with histidine residues in similar positions observed in most alphaviruses, except SINV. Together, these findings indicate that



Source: Adapted from [83]. Licensed under CC BY 4.0.

Figure 5.5: MAYV E1 and E2 transmembrane domains and the hydrophobic cavity. (A) 3D atomic model of MAYV fitted to the density map, showing the upper portion of the E1 and E2 TM helices. The E1-E2 intersection forms a cavity with anterior and posterior openings. The observed extra density within the cavity is indicated. **(B)** Detail of the extra density found in the head region of E1-E2 and the surrounding residues. The density map of MAYV is shown in mesh or surface representation. **(C)** Cavity detection in alphaviral structures. The cavity between the E1 and E2 domains is shown in surface representation and colored based on the Eisenberg Weiss hydrophobicity scale, using the residues that form the cavity. Only polar residues are represented as sticks. **(D)** Cavity volume for the four E1-E2 heterodimers (n = 4 independent heterodimer structures) in the asymmetric unit. One-way ANOVA with Tukey's multiple comparison test was used to compare the MAYV cavity volume with other alphaviruses (* indicates adj. $p < 0.01$ when compared to alphaviruses with MAYV). **(E)** Number of residues in each of the four E1-E2 heterodimers (n = 4 independent heterodimer structures) in the asymmetric unit, separated by classes. One-way ANOVA with Tukey's multiple comparison test was used to compare the number of residues in the apolar aliphatic class with the other classes in the same alphavirus species (* indicates adj. $p < 0.01$ when comparing the apolar aliphatic class with the other classes). All data are presented as mean values +/- SD. Apolar aliphatic: ALA, VAL, ILE, LEU, GLY, PRO; Aromatic: PHE, TYR, TRP; Uncharged polar: SER, THR, CYS, MET, ASN, GLN; Negatively charged: GLU, ASP; Positively charged: ARG, LYS, HIS.

alphaviruses have a consistent amphipathic cavity formed between the E1 and E2 domains in the outer membrane of the lipid bilayer. If the alphaviral pocket hosts a molecule, it would be chemically similar across different alphaviruses. The density map of MAYV suggests that the extra density may be occupied by a fatty acid, potentially enhancing interactions between E1 and E2. Consequently, the pocket emerges as a potential target for the development of antiviral compounds against MAYV and other alphaviruses through rational drug design.



Source: Adapted from [83]. Licensed under CC BY 4.0.

Figure 5.6: Interaction of MAYV capsid with the C-terminal domain of the E2 protein. (A) 3D atomic model of MAYV fitted to the density map obtained by cryo-electron microscopy. The C, E1, and E2 proteins are represented as cyan, purple, and blue cartoons, respectively. **(B)** The interaction of the TPY motif (residues T387, P398, and Y399) with the C protein. **(C)** Residues P403 and T402 of E2 and their interaction with the aromatic residues Y175 and W242 in the C protein. The density map of MAYV is shown in mesh representation. **(D)** Cavity detection in the C protein of MAYV, showing a hydrophobic environment at the bottom of the pocket and a polar and charged environment at the outer edges of the cavity. The cavity in the C protein that binds to the C-terminal domain of E2 is shown in surface representation and colored according to the Eisenberg hydrophobic consensus scale. The C protein residues surrounding the cavity are represented as sticks. **(E)** Boxplot of the cavity volume for the four capsids (n = 4 independent capsid structures) in the asymmetric unit. In the boxplot, the box represents the interquartile range (IQR) (67.5Å^3), the 75th percentile ($Q3$) (526.2Å^3), and the 25th percentile ($Q1$) (458.7Å^3). The central line indicates the median (486.3Å^3), and the mean (498.6Å^3) is indicated by a dot. The "whiskers" with a minimum value (395.7Å^3) and a maximum value (626.2Å^3) are determined using $Q1 - 1.5 \cdot IQR$ and $Q3 + 1.5 \cdot IQR$, respectively.

On the other hand, alphavirus nucleocapsid proteins C are composed of two subdomains: a disordered N-terminal domain, responsible for binding to viral RNA (not observed in the MAYV density map), and a structured C-terminal domain that non-covalently binds to E2 proteins (Figure 5.6). The N-terminal region has lower sequence identity among alphaviruses and is reported as virus-specific [83]. The MAYV density map confirms the generally conserved structure of the C protein, forming two subdomains rich in beta sheets separated by a shallow cavity of ∼500 Å^3 (Figure 5.6E), wherein the C-terminal domain of the E2 protein non-covalently binds (Figure 5.6A and D). The bottom of the pocket is hydrophobic, while the upper part contains polar and charged residues (Figure 5.6D). The interface between the capsid and the E2 protein involves the TPY consensus motif (residues T387, P398, and Y399; Figure 5.6B), conserved within the *Alphavirus* genus. Interestingly, small molecules proposed to inhibit the interaction

between the capsid and the E2 protein contain heterocyclic rings, reinforcing the relevance of this type of contact for the capsid-E2 protein interaction and positioning this site as a potential drug target.

It is important to note that the complete results and details of the analyses are available in the article published in the *Nature Communications* [83].

5.1.2 Discussion

parKVFinder emerges as a powerful tool for detecting and characterizing biomolecular cavities, offering enhanced capabilities through integration with PyMOL—a user-friendly platform for visualization. The routine optimization and parallelization have significantly improved its performance, surpassing its predecessor, KVFinder [8, 9]. In the case study involving the MD of HIV-1 protease, parKVFinder demonstrated a remarkable capacity to accurately describe the conformational dynamics of the active site, outperforming other geometry-based methods both in accuracy and computational time. The investigation of the hydrophobic pocket within alphaviruses, particularly in MAYV, not only identifies on potential drug targets but also highlights shared structural features across alphaviruses. This consistency in the nature of the cavity underscores parKVFinder's utility in uncovering fundamental aspects of biomolecular structures.

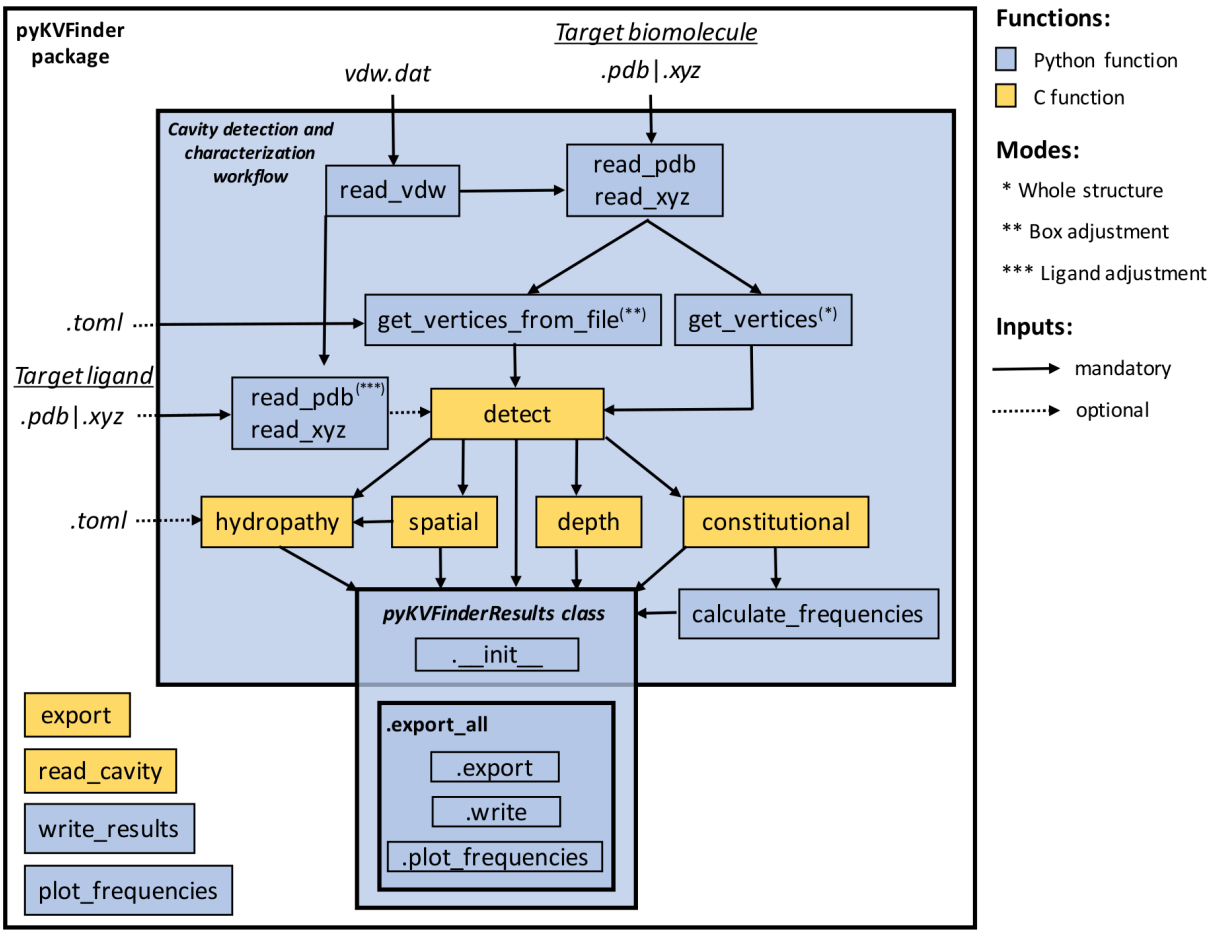
Despite the successful application of parKVFinder in MD simulations [9] and a comparative study [83], certain limitations became apparent in automated tasks and systematic binding site comparisons. Consequently, there arises a need for a more suitable tool tailored to data science applications, providing straightforward access to functions and data structures for efficient analysis. Nevertheless, it is important to acknowledge that parKVFinder still plays a significant role within KVFinder suite. Its contribution lies notably in optimizing detection and characterization parameters through the PyMOL graphical plugin (PyMOL2 parKVFinder Tools), leveraging its visual cues. These fine-tuned parameters seamlessly integrate into automated studies and systematic binding site comparisons. In essence, parKVFinder continues to be indispensable for conducting structural and functional analyses focused on individual biomolecular structures.

5.2 pyKVFinder

In data science, data-intensive cavity analysis requires efficient routines and algorithms built on easily manipulable data structures. Cavities identified by parKVFinder, like those in other well-known programs such as fpocket [48], GHECOM [52], and POVME 3.0 [40], are human-readable and easily displayed in molecular visualization programs. However, they lack suitable structuring for direct integration into automated protocols and data science applications. Addressing this need, we developed the Python-C parallel KVFinder (pyKVFinder) [5], an open-source Python package, licensed under GPL v3.0, for detecting and characterizing cavities in biomolecular structures within automated protocols and data science applications. Subsequently, pyKVFinder was published in *BMC Bioinformatics* [5] and released as pyKVFinder v0.2.5, with the current version being

v0.6.9. The source code is under continuous development and available at the following repository: https://github.com/LBC-LNBio/pyKVFinder.

pyKVFinder employs Simplified Wrapper and Interface Generator (SWIG) (<https://www.swig.org>) to generate Python bindings from the C library, allowing the extension of 3D grid operations, written in C, to the high-level programming language, Python. This integration combines the efficiency of C-compiled routines for grid operations with the flexibility, scalability, and ease of learning inherent in the interpreted nature of Python. By doing so, pyKVFinder can be imported as a package in the Python environment and users can decide to run the full cavity detection and characterization workflow through the *pyKVFinder.run_workflow* function or run pyKVFinder functions in a stepwise fashion (Figure 5.7). For further details on the functions of pyKVFinder package, please refer to the article published in *BMC Bioinformatics* [5] or the documentation page at <https://lbc-lnbio.github.io/pyKVFinder/>.



Source: Reprinted from [5]. Licensed under CC BY 4.0.

Figure 5.7: Diagram of cavity detection and characterization workflow using pyKVFinder package. The flowchart illustrates function calls and their dependencies for performing cavity detection and characterization with pyKVFinder package

In pyKVFinder, the target biomolecule is inserted in a regular 3D grid, stored as an N-dimensional array (ndarray) from the NumPy package [88]. To detect cavities, pyKVFinder uses a dual-probe algorithm, as shown in Figure 3.3, which scans the biomolecular structure for regions of inaccessibility (i.e., cavities). In addition to cav-

ity properties like volume, area, and interface residues stored as Python dictionaries, pyKVFinder calculates cavity depth and hydrophobicity. Both cavity points and these morphological and physicochemical properties are stored in ndarrays and can be visualized using Python molecular visualization packages (e.g., NGLView [89] and plotly [90]). Moreover, pyKVFinder can be integrated with various scientific packages and libraries (e.g., scikit-learn [91] and SciPy [92]) for mathematical calculations, statistical analysis, and 3D visualization using interactive interfaces (e.g., IPython, Jupyter, and JupyterLab notebooks). Thus, pyKVFinder facilitates complex analyses of biostructural data with protocols and algorithms within the Python ecosystem, serving as a building block for new applications in data science, rational drug design, and drug discovery. In essence, pyKVFinder provides a versatile means of detecting and characterizing biomolecular cavities and integrating this information into automated protocols and data science applications.

The computational performance of pyKVFinder was also assessed on kv1000 [9], presenting a considerably shorter runtime compared to parKVFinder, averaging 31% faster (Figure 5.8). The primary reason for this performance gain lies in the additional possibility to parallelize routines, i.e., atom insertion into the 3D grid in the detection function (i.e., *pyKVFinder.detect*), based on ndarrays. The most significant improvement was observed in proteins with over 2000 atoms, achieving a speedup of ∼4.3 times in proteins with 11000 atoms, benefiting the growing number of currently resolved high-order structures. For very small proteins (2000 atoms), which represent a smaller portion of available structures, pyKVFinder's performance gain was not significant or even lower than that of parKVFinder, mainly due to the Python reading of the target's PDB or XYZ file.

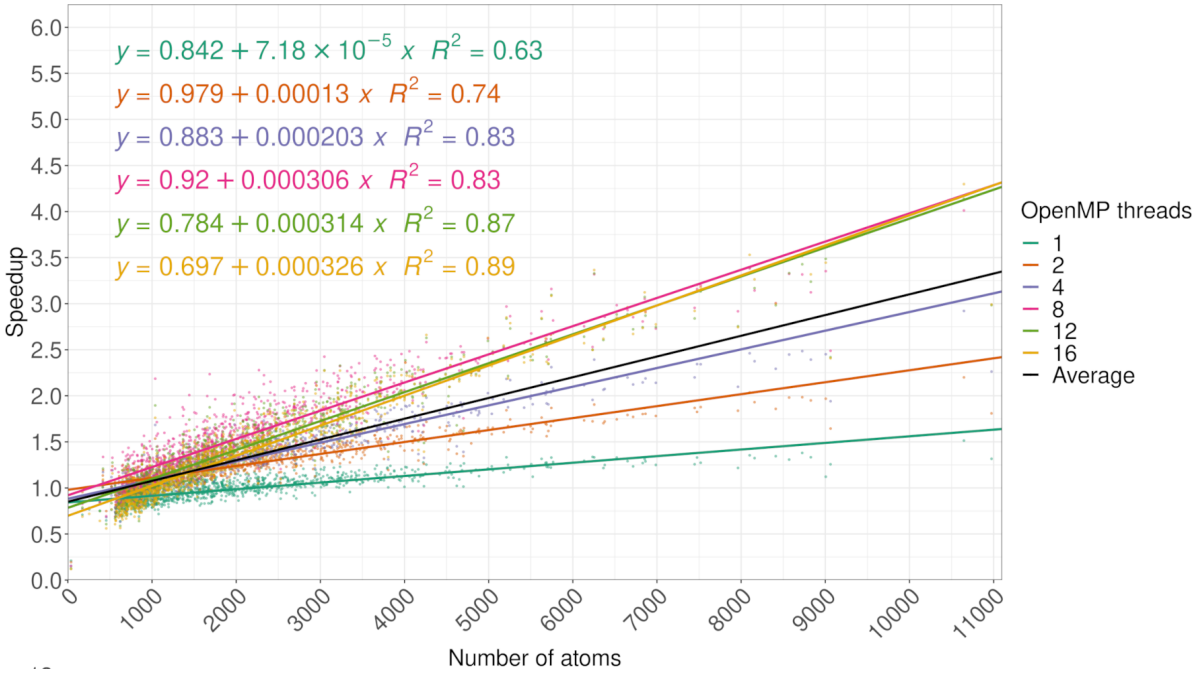


Figure 5.8: Speedup of pyKVFinder compared to parKVFinder. The speedup is the ratio of pyKVFinder's execution time to parKVFinder's execution time, applying the same number of OpenMP threads, for different numbers of atoms.

Even with the addition of new characterizations such as depth and hydrophobicity, pyKVFinder's performance was only reduced by an average of 5% (for depth) and 4% (for hydrophobicity), regardless of the number of threads used (Figure 5.9). Additionally, the scalability of pyKVFinder with an increasing number of threads, as well as the absolute time to perform cavity detection, is presented in Figure 5.9, following the behavior exhibited by parKVFinder [9].

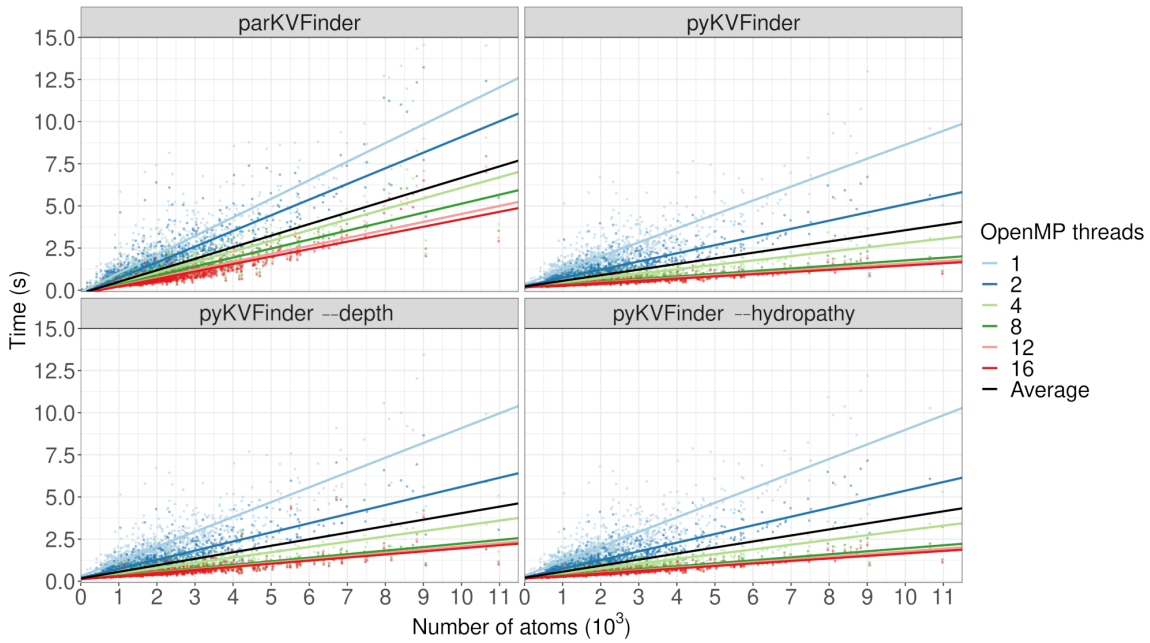


Figure 5.9: Computational time as a function of the number of atoms with different numbers of threads for parKVFinder and pyKVFinder. Top left panel: parKVFinder. Top right panel: pyKVFinder with default characterization (volume, area, and interface residues). Bottom left panel: pyKVFinder with default and depth characterizations. Bottom right panel: pyKVFinder with default and hydrophobicity characterizations.

Therefore, experienced users requiring scripting routines are encouraged to use pyKVFinder due to its enhanced performance, while newcomers should prioritize parKVFinder due to its monolithic behavior and ease of installation and execution.

5.2.1 Implementations of New Characterizations

Within the context of pyKVFinder, in collaboration with Dr. György Szalóki (Laboratoire Hétérochimie Fondamentale et Appliquée - Université Toulouse III Paul Sabatier - France), the scope of the KVFinder suite has been expanded to a new class of molecules called supramolecular cages. These cages are interconnected molecules that come together through non-covalent interactions, forming an internal cavity capable of encapsulating molecules or ions. The shape and size of the cavity are important parameters that can be easily determined by geometric algorithms, aiding in the rational design of supramolecular cages. In this context, new characterizations applicable to both cage and biomolecular contexts have been developed.

5.2.1.1 Molecular Volume Estimation

The *pyKVFinder.Molecule* class, introduced in pyKVFinder v0.5.0, enables users to model detailed molecular surfaces, following mathematical formulations shown in Figure 4.2, within the pyKVFinder framework. In this approach, molecules are inserted into a regular 3D grid, taking into account the vdW radii of each of the atoms in the molecule. Users can customize these radii through a configuration file (*vdw.dat*), as well as the surface representation, i.e.vdW surface, SES and SAS. Conveniently, users can represent the vdW surface by invoking *Molecule.vdw*, SES by invoking *Molecule.surface('SES')*, and SAS by invoking *Molecule.surface('SAS')* (Figure 5.10). Within the 3D grid, each voxel is assigned as a molecule (0) or solvent (1) point, allowing the estimation of the vdW volume by summing the voxels labeled as a molecule through the *Molecule.volume* method. The comprehensive implementation of these features for molecular modeling and characterization has been described and applied in an article published in the *Journal of Chemical Information and Modeling* [36].

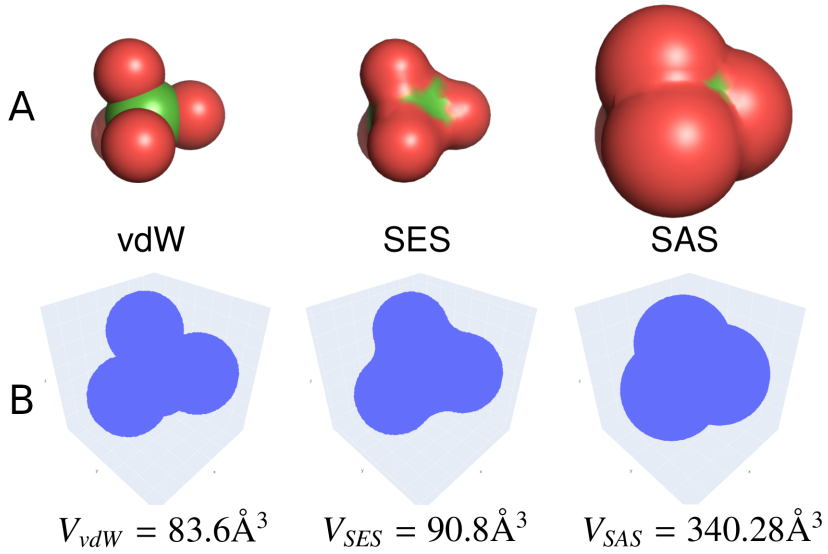


Figure 5.10: Molecular modeling and volume estimation of perchlorate (ClO_4). (A) Molecular surface of vdW (left panel), SES (center panel), and SAS (right panel) in the PyMOL molecular viewer. **(B)** Modeling and estimation of the vdW (left panel), SES (center panel), and SAS (right panel) molecular volume by pyKVFinder.

5.2.1.2 Opening Characterization

Understanding the characteristics of supramolecular cages, such as volume (Figure 5.11A) and openings (Figure 5.11B), which drive the encapsulation of reactive intermediates, is crucial for the rational design of new supramolecular cages with enhanced catalytic properties. In this regard, we have developed an opening characterization using pyKVFinder, allowing the identification of openings, determination of the area of these openings, and the largest spherical probe (i.e., atom) that can pass through each opening, as shown in Figure 5.11.

The dual-probe system uses integer identifiers for cavity points (1), biomolecule points (0), and solvent points (−1). After cavity segmentation, performed by the Depth-

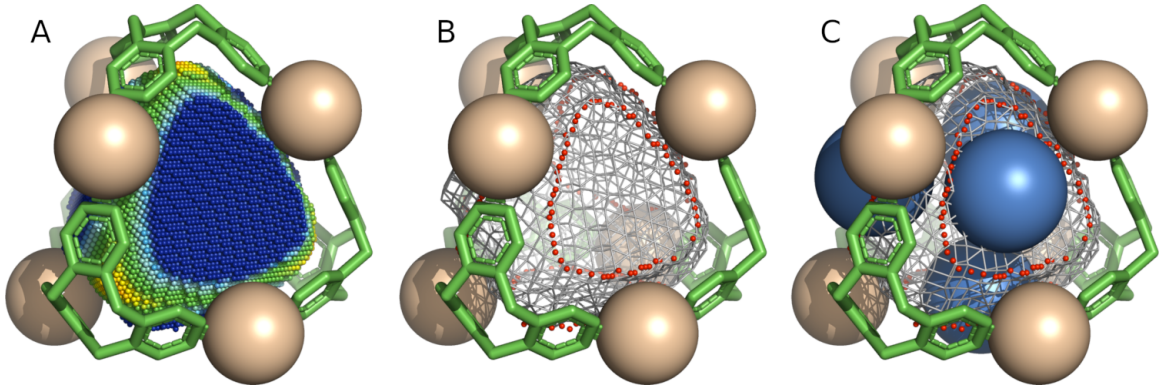


Figure 5.11: Supramolecular cage characterizations. (A) Volume and depth. Points are colored according to depth, with blue for shallower and red for deeper points. **(B)** Openings (red points) and opening area. **(C)** Largest spherical probe (blue sphere) accessible to the cage cavity.

First Search (DFS) algorithm, the cavity points are marked with values ≥ 2, and the cavity points that did not reach the volume cutoff are retained with the value 1. From this voxel classification, cavity points (≥ 2) located at a grid unit from a solvent point (-1), following the structuring element's rank 3 and connectivity 1 relationship (Figure 5.12), are identified as cavity-solvent boundary points, marked with the negative value of the corresponding cavity's numeric identifier, as described for depth characterization [5, 8]. From these boundary points, the surface area is calculated using the area estimation methodology of the KVFinder suite [8, 9], which corresponds to the opening area. Subsequently, the cavity-solvent boundary points located at a grid unit from a biomolecule point (0), following the structuring element's rank 3 and connectivity 1 relationship (Figure 5.12), are identified as opening points. At this stage, a new 3D grid is generated to accumulate the opening points, which are marked with the value 1, while the remaining points receive the value 0. Afterwards, the DFS algorithm is repourpouse to segment boundary points, to identify distinct openings, where opening points are marked with values ≥ 2 (Figure 5.11B), and openings with fewer points than a cutoff defined by the user are kept with the value 1. Finally, for each identified opening, the midpoint is calculated, and the largest sphere centered in this midpoint is fitted inside the respective boundary, defining the largest atom that can pass through this opening (Figure 5.11C).

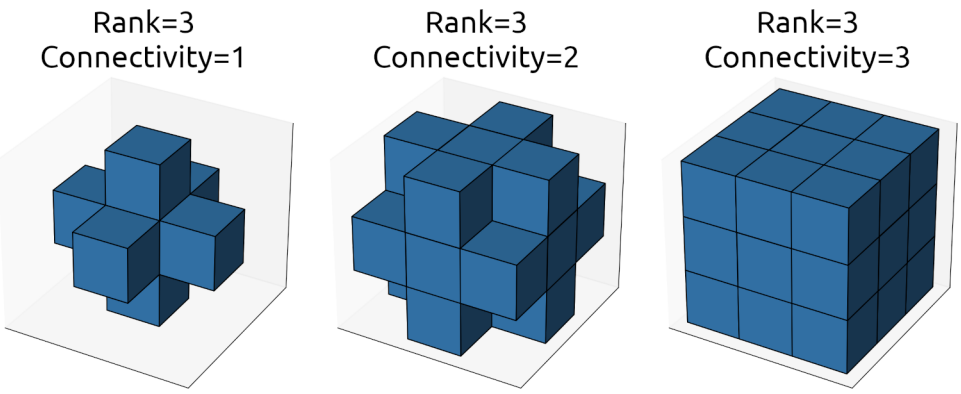


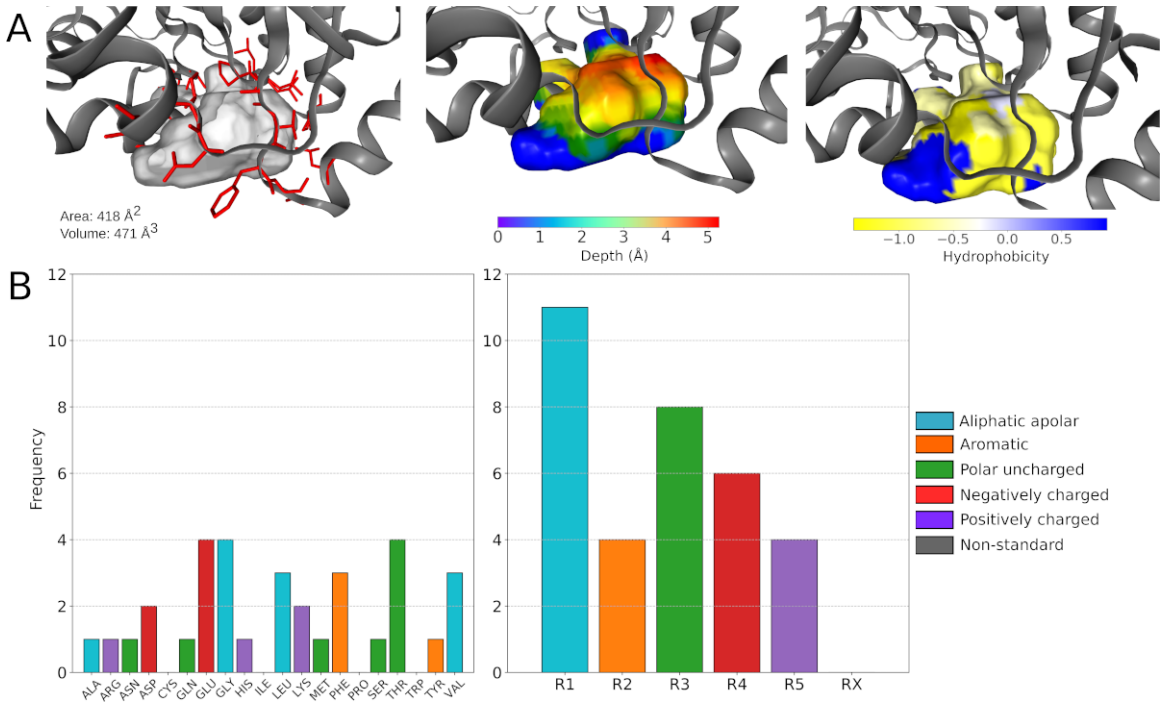
Figure 5.12: Structuring elements for spatial filters.

5.2.2 Case Studies

The pyKVFinder was applied in two case studies published in scientific journals to investigate proteins of therapeutic interest. These analyses explored the characteristics of cavities in proteins homologous to the ADP-ribose phosphatase (ADRP) domain of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) and the MD of the ADRP domain of SARS-CoV-2 [5]. Next, we will describe each of these case studies in detail.

5.2.2.1 SARS-CoV-2 and Homologous Proteins

Among the 15 non-structural proteins (Nsps) of the SARS-CoV-2, the ADRP domain of nsp3 protein, also known as the macrodomain, is noteworthy [93]. This domain has been under investigation to comprehend its exact functions in the coronavirus life cycle, as the ADRP domain recognizes ADP-ribose 1'-phosphate and appears to play a crucial role in virulence and innate immunity regulation against infection [94, 95]. Recent efforts have aimed to characterize the ADP-ribose substrate-binding site and evaluate this site as a potential target for antiviral drugs.



Source: Adapted from [5]. Licensed under CC BY 4.0.

Figure 5.13: Characterization of the ADRP substrate-binding cavity of SARS-CoV-2. (A) Characterizations of the substrate-binding site of the ADRP domain of SARS-CoV-2 (PDB ID: 6WEN). Left panel: Detected cavity represented as a gray surface and the surrounding residues as red sticks. The cavity area and volume are shown. Center panel: Cavity colored by depth. Right panel: Cavity colored by hydrophobicity using the Eisenberg and Weiss scale. **(B)** Bar chart of interface residue frequencies. Left panel: Amino acids. Right panel: Classes of amino acids.

In this context, we employed pyKVFinder to detect and characterize a cavity present in the ADRP protein of SARS-CoV-2, corresponding to the substrate-binding site (Figure 5.13A). After cavity detection, we characterized its volume, area, and the

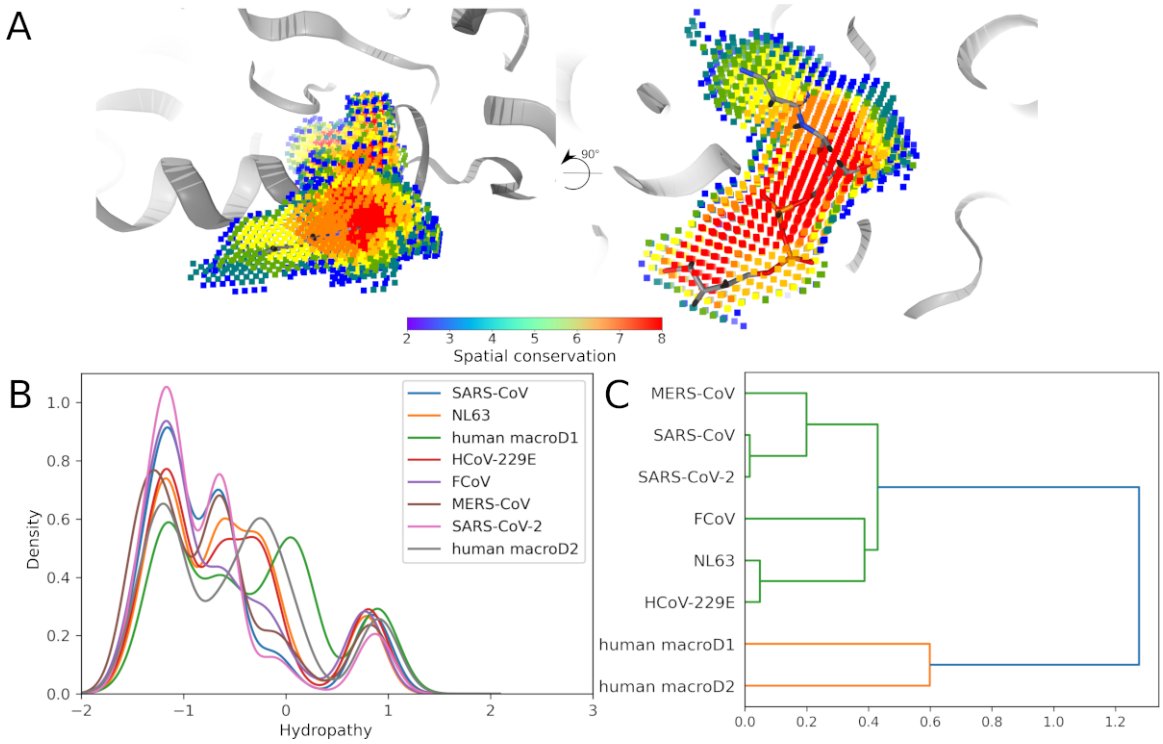
interface residues involved in the cavity (Figure 5.13A, left panel). It noteworthy that 3D visualization of the protein and cavity can be performed in the Jupyter notebook itself using the NGLView package [89]. However, users have the freedom to choose other molecular visualization tools (e.g., PyMOL [56], ChimeraX [96], NGL Viewer [97], or VMD [98]). Additionally, we inspected the substrate-binding cavity of ADRP in terms of depth (Figure 5.13A, center panel) and hydrophobicity (Figure 5.13A, right panel). These descriptions are relevant for drug development [99]. Despite being solvent-exposed in the apo form, the cavity has internal components (in red) that reach a more central portion of the ADRP beta-sheet (Figure 5.13A, center panel). Hydrophobicity analysis shows that the cavity's core is more hydrophobic (in yellow), with some polar residues at the edges (in blue), contributing to the rational design of more specific ligands.

As illustrated in Figure 5.13A, the ADP-ribose binding site forms a cleft between the α-helices of the ADRP domain, and key contacts involve residues from loop regions, explaining the flexibility of the pocket during substrate binding [93]. We then determined the composition of the residues forming the cavity and presented their frequencies in a histogram (Figure 5.13B), using the matplotlib library [100]. However, users are free to analyze the data and present results using their preferred graphical library.

To compare the composition of this ADRP binding cavity with that of other related proteins, we conducted the same analysis on seven other selected proteins based on structural homology and alignment using Dali [101]. The proteins related to the ADRP domain of SARS-CoV-2 (PDB ID: 6WEN, chain A) are: MERS-CoV (PDB ID: 5HIH, chain A), NL63 (PDB ID: 2VRI, chain A), HCoV-229E (PDB ID: 3EJG, chain A), FCoV (PDB ID: 3ETI, chain B), and human macrodomain proteins macroD1 (PDB ID: 2X47, chain A) and macroD2 (PDB ID: 6Y73, chain D). Structures in the apo form were realigned using the MUSTANG algorithm [102] of the YASARA program [103]. We assessed the conservation of the cavity across species by performing arithmetic operations (i.e., addition and division) on the boolean ndarrays representing detected cavities. As observed in Figure 5.14A, the core of the ADRP cavity (red points) is highly conserved among the analyzed species, being occupied by the ADP's diphosphate and ribose, as well as the second ribose linked to ADP in the substrate-bound form of ADRP. On the other hand, adenosine occupies a less conserved region of the cavity (blue points), indicating that the structure of this site undergoes changes in some species to accommodate the ADP-ribose substrate.

To compare the hydrophobicity of the cavity across species, we plotted a hydrophobicity distribution using the matplotlib library [100], as shown in Figure 5.14B. The distribution clearly reveals the hydrophobic nature of the pocket, widely shared among the ADRP substrate-binding cavities of coronaviruses. Interestingly, human proteins macroD1 and macroD2 seem to exhibit a less pronounced hydrophobicity distribution.

With pyKVFinder, we can calculate the frequency of residues composing the cavity. Using the SciPy library [92], we performed hierarchical clustering of these frequencies and presented the dendrogram in Figure 5.14C. In this dendrogram, we observe that the cavity of SARS-CoV-2 ADRP clusters with that of SARS-CoV, demonstrating high similarity between these betacoronaviruses. Next to them, we can observe another betacoronavirus, MERS-CoV. In turn, alphacoronaviruses NL63 and HCoV-229E, and feline



Source: Adapted from [5]. Licensed under CC BY 4.0.

Figure 5.14: Comparative study of the ADRP substrate-binding site of SARS-CoV-2 and related proteins. (A) Conservation analysis of the ADP-ribose binding site in the ADRP domain of SARS-CoV-2 (PDB ID: 6WEN, chain A), SARS-CoV (PDB ID: 2ACF, chain B), MERS-CoV (PDB ID: 5HIH, chain A), NL63 (PDB ID: 2VRI, chain A), HCoV-229E (PDB ID: 3EJG, chain A), FCoV (PDB ID: 3ETI, chain B), and human macrodomain proteins macroD1 (PDB ID: 2X47, chain A) and macroD2 (PDB ID: 6Y73, chain D). The cavity points detected in at least two structures are colored by conservation percentage. **(B)** Hydropathy profile. **(C)** Hierarchical clustering dendrogram of residue frequency. Pearson correlation metric was used to assess similarity, and complete linkage method was chosen as the linking method. All graphics and images were generated in a Jupyter notebook. Three-dimensional structure images were created using the NGLView package [89], while graphics were built using the matplotlib package [100].

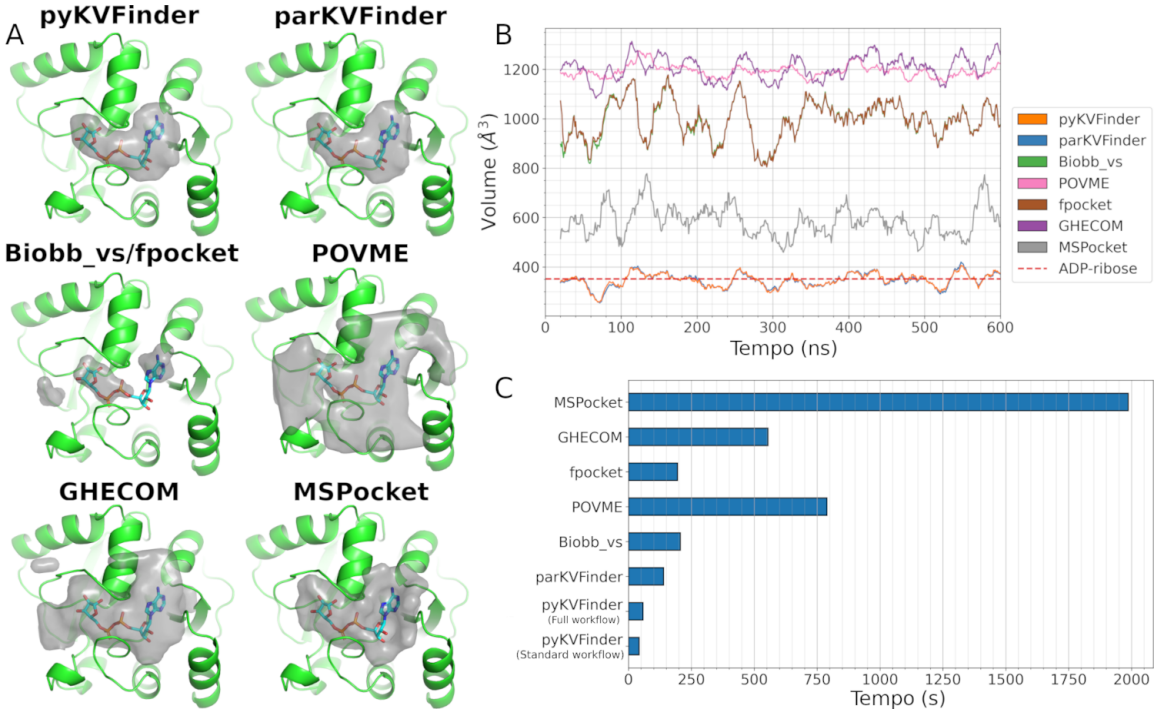
FCoV cluster together. Further away from the coronavirus macrodomain domains are the two human macrodomain proteins, macroD1 and D2. Despite the ADRP or macroD1/D2 cavities sharing the same substrate, ADP-ribose, these results indicate that the residue profile around these cavities follows evolutionary traces.

To demonstrate the functionalities and advantages of pyKVFinder, we conducted this study in a Jupyter notebook, executing step-by-step pyKVFinder functions. The notebook with the complete case study is available at <https://github.com/LBC-LNBio/pyKVFinder/blob/master/examples/conservation-analysis/conservation-analysis.ipynb>. A detailed description of this analysis is provided in the article published in the *BMC Bioinformatics* [5].

5.2.2.2 Molecular Dynamics of the ADRP Domain of SARS-CoV-2

In this context, the computational performance of pyKVFinder was evaluated in MD simulations of the ADRP domain of SARS-CoV-2 (PDB ID: 6W02, chain B) without its ligand, ADP-ribose, for a period of 600 ns, with frame extraction every

1 ns. This analysis was repeated with other well-known tools: parKVFinder v1.1.3 [9], POVME 3.0 [40], fpocket [48], GHECOM [52], MSPocket [46], and Biobb_vs [104]. All these methods successfully detected the substrate-binding site of the ADRP, where the shape and volume varied slightly during the MD simulation (Figure 5.15). The shapes of the cavities detected by pyKVFinder and parKVFinder precisely fit the original ligand in the binding site, similar to MSPocket (Figure 5.15A). Additionally, the volume calculated by pyKVFinder ($346.8 \pm 78.7 Å^3$) and parKVFinder ($346.5 \pm 79.3 Å^3$) closely relates to the volume of ADP-ribose ($351.1 Å^3$; molecular surface volume estimated by the YASARA program), which originally occupied the substrate-binding site in the crystallographic structure used in the MD simulations (Figure 5.15B).



Source: Reprinted from [5]. Licensed under CC BY 4.0.

Figure 5.15: Performance evaluation of benchmarking methods for the ADRP substrate-binding site detection. (A) Protein structures (shown in green cartoons) in frame 30 (with the lowest RMSD compared to the crystallographic structure) of the ADRP domain trajectory with corresponding detected cavities (gray surfaces) by each benchmarking method. **(B)** Total volume of cavities detected at the substrate-binding site of ADRP over a 600 ns simulation. The total volume is calculated in a window of 20 frames. The red dashed line indicates the molecular surface volume of the ADP-ribose molecule, which originally occupied the substrate-binding site of ADRP in the crystallographic structure (PDB ID: 6W02, chain B). **(C)** Elapsed time to detect and characterize the substrate-binding site of ADRP. The default protocol of pyKVFinder, as well as parKVFinder, detects cavities and applies morphological (e.g., volume, area, and shape) and constitutional (e.g., interface residues and their frequencies) characterizations. The complete pyKVFinder protocol includes the default protocol with depth and hydrophobicity characterizations.

In addition to accurately detecting biomolecular cavities, current tools must also perform perform fast cavity detection and characterizations. Therefore, we also assessed the elapsed time to execute these benchmarking methods (Figure 5.15C). pyKVFinder outperformed all analyzed methods in this aspect, even when applying the new characterization features of depth and hydrophobicity; the elapsed time of pyKVFinder increased

only by 36%, still outperforming other benchmarking methods. Furthermore, compared to parKVFinder, pyKVFinder was 3.3 times faster in detecting the ADRP binding site. The main reason for the performance gain is the additional possibility of parallelizing routines, i.e., atom insertion into the 3D grid in the detection function, based on ndarrays. Additionally, pyKVFinder's scalability, with an increase in the number of threads, follows the same behavior as parKVFinder [9]. Therefore, pyKVFinder offers a more flexible and efficient option for experienced users requiring large-scale applications, while parKVFinder is more suitable for beginners due to its simplicity of installation and use.

To demonstrate the functionalities and advantages of pyKVFinder, we conducted this study in a Jupyter notebook, executing step-by-step pyKVFinder functions. The notebook with the complete case study is available at <https://github.com/LBC-LNBio/pyKVFinder/blob/master/examples/md-analysis/md-analysis.ipynb>. A detailed description of this analysis is provided in the article published in the *BMC Bioinformatics* [5].

5.2.3 Discussion

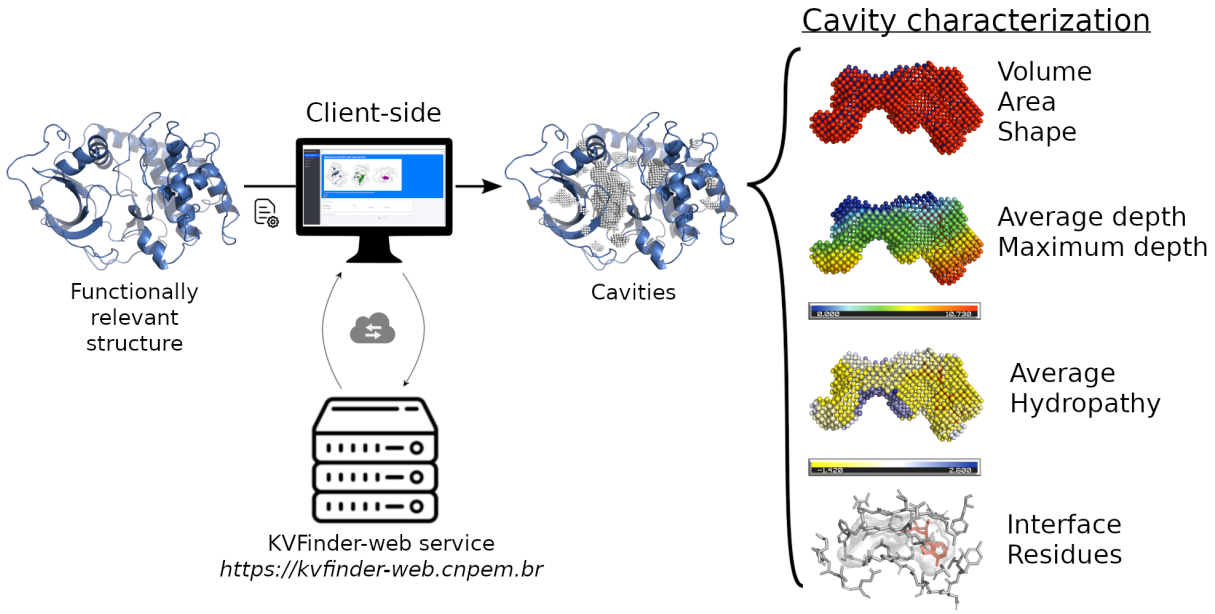
Despite each method having its own set of characterizations to be performed on the detected cavities, the cavity data structure is only accessible within the Python ecosystem in pyKVFinder, which provides ndarrays and dictionaries in Python. By providing an accessible and flexible data structure, pyKVFinder allows users to develop new cavity characterizations, as well as analysis protocols based on these data structures. For example, in a recent study conducted by [105], the cross-sectional area of cavities in related proteins was explored using pyKVFinder. This study demonstrated how the data structures provided by pyKVFinder can be used to delve deeper into cavity exploration and discover information relevant to drug development and understanding molecular interactions. Furthermore, the integration of pyKVFinder with the Python ecosystem expands the possibilities for data analysis and visualization, leveraging the robust scientific libraries available in Python language. This integration facilitates the implementation of advanced and customized analyses, enabling researchers to comprehensively explore the properties of biomolecular cavities and gain valuable insights.

In this way, pyKVFinder not only contributes to advancing research in drug discovery and rational drug design but also strengthens collaboration and knowledge sharing in the scientific community. By providing an accessible, flexible tool integrated into the Python ecosystem, pyKVFinder empowers researchers to explore biomolecular cavities more efficiently and effectively, driving the discovery of new therapeutic targets and the development of more effective drugs.

5.3 KVFinder-web

Recently, web services using HyperText Transfer Protocol (HTTP) protocols have become popular approaches in cloud computing environments, providing broad access to data and processing resources. Various web services have been introduced to detect

and/or characterize binding sites in biomolecules, including GHECOM [52], 3DLigandSite [106], CaverWeb [107], FpocketWeb [108], and MoloVol [109]. Compared to other tools for detecting cavities, parKVFinder stands out with its intuitive set of parameters and extensive testing in the literature for both detection and computational capabilities, offering precise and robust performance with any type of protein cavity, as shown previously [5, 8, 9]. Although other methods can also detect protein binding sites, each has its specific set of characterizations. parKVFinder distinguishes itself by combining morphological, topological, and physicochemical characterizations of binding sites, effectively assisting users in identifying functionally relevant cavities and studying the molecular recognition process.



Source: Adapted from [10]. Licensed under CC BY 4.0.

Figure 5.16: Representative scheme of the KVFinder-web workflow to detect and characterize cavities in functionally relevant structures.

Beyond the advancements in parKVFinder's performance and usability, the installation and configuration procedures of our cavity detection tool, alongside other independent tools, still present a significant barrier for users without the ideally required technical knowledge. Additionally, scientists, educators, and students may encounter limitations in their local computational resources, affecting the proper use of cavity detection and characterization methods.

In this scenario, we introduced KVFinder-web [10], an open-source web application for detecting and characterizing cavities in a wide range of biomolecular structures, including but not limited to proteins and nucleic acids. Subsequently, KVFinder-web was published in *Nucleic Acids Research* [10] and released as KVFinder-web v1.1.0, with the current version being v1.1.1. Accessible at <https://kvfinder-web.cnpem.br>, KVFinder-web operates on a standard client-server architecture, consisting of two independent components: a RESTful web service (KVFinder-web service) and a graphical web interface (KVFinder-web portal). To further enhance the user experience, we also provide a graphical plugin for PyMOL (PyMOL KVFinder-web Tools). Next, we will describe each of

these components in detail.

5.3.1 KVFinder-web portal

The KVFinder-web portal is an interactive graphical interface of KVFinder-web that provides users, especially those inexperienced, with an easy-to-use web application to run parKVFinder (v1.2.0) and analyze results through any web browser. Developed in R Shiny [110], the KVFinder-web portal offers a simple, direct, robust, and interactive protocol for cavity analysis and visualization, requiring only a biomolecule in PDB format or its corresponding PDB code.

The KVFinder-web portal was initially published in version v1.1.0 in the *Nucleic Acids Research* [10]. After collecting feedback from users, the graphical interface (Figure 5.17) has undergone improvements in version v1.1.1 to enhance user experience. Changes include revamping tab layouts, fixing critical bugs in the Retrieve Results tab, revising documentation for clarity, adding the LNBio/CNPEM logo, and aligning the color scheme with the LNBio institutional colors. These updates aim to provide a more polished and user-friendly KVFinder-web, ensuring visual consistency and addressing functional issues for a smoother user interaction. The source code is under continuous development and available at the following repository: <https://github.com/LBC-LNBio/KVFinder-web-portal>.

The interface provides the key functionalities of the KVFinder-web service, allowing users to upload a target biomolecule from a PDB file or provide the corresponding PDB code, and customize cavity detection parameters and execution modes (Figure 5.17B). Four cavity detection modes are available, offering options that best suit users' cavity analysis needs:

- **Whole structure (default)**: uses preset parameters to detect cavities across the whole biomolecular structure;
- **Whole structure (customized)**: allows users to customize the detection parameters for cavity detection across the whole biomolecular structure;
- **Around target molecule**: detects cavities around a chosen target molecule, focusing the search on the region surrounding that molecule, with customizable detection parameters;
- **Around target residues**: detects cavities within a custom box, defined by selecting specific residues within the target biomolecular structure, with customizable detection parameters.

Customizable detection parameters include *Probe In*, *Probe Out*, *Removal Distance*, and *Volume Cutoff* (Figure 5.17B). In brief, *Probe In* is a smaller spherical probe (in Å) that traverses the target biomolecule, defining its molecular surface (as default, defined by the radius of a water molecule - 1.4Å), while *Probe Out* is a larger spherical probe (as default, defined as a sphere of radius of 4.0Å) that traverses the target biomolecule, defining regions of inaccessibility. Thus, cavities are defined as regions accessible to *Probe*

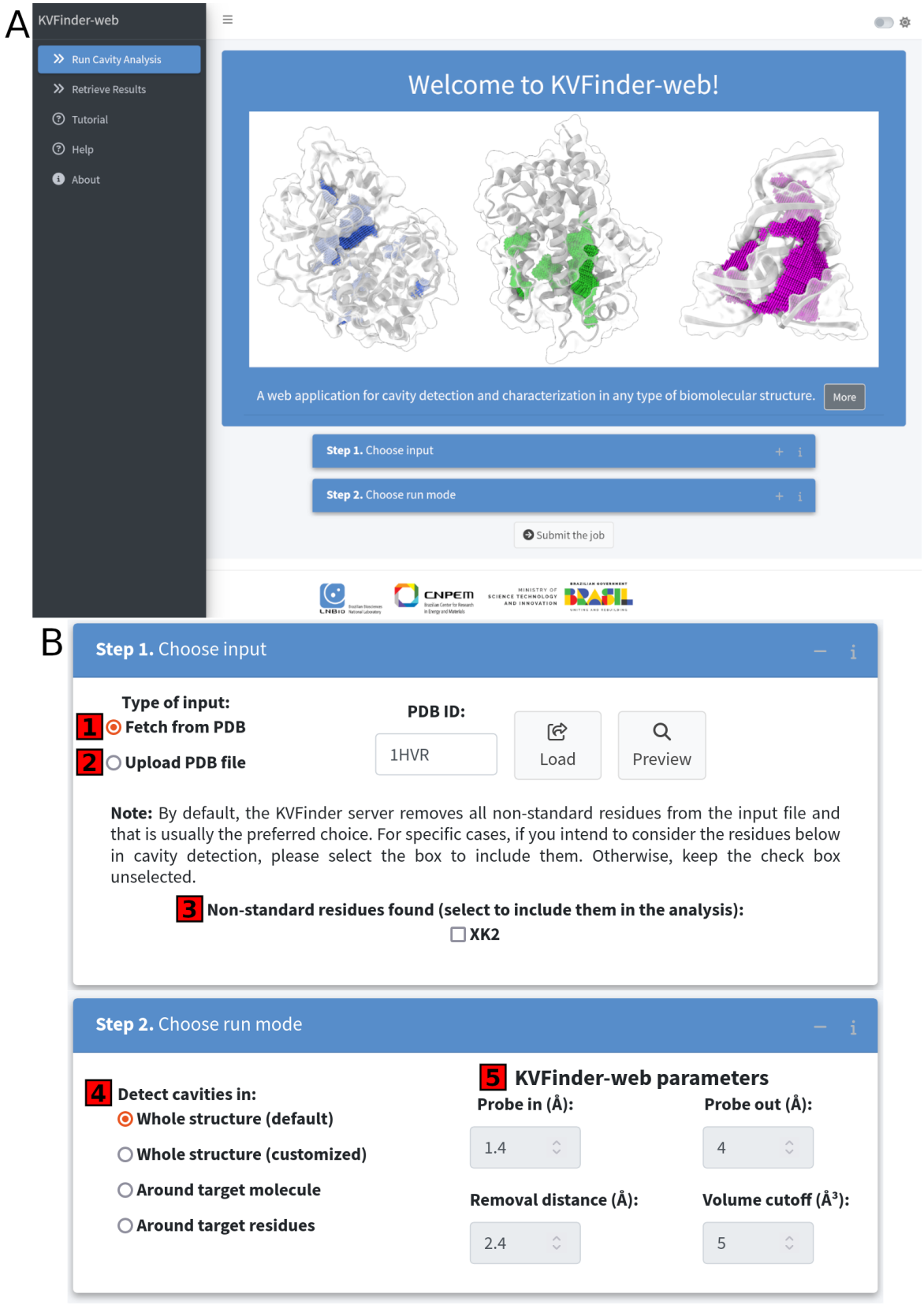
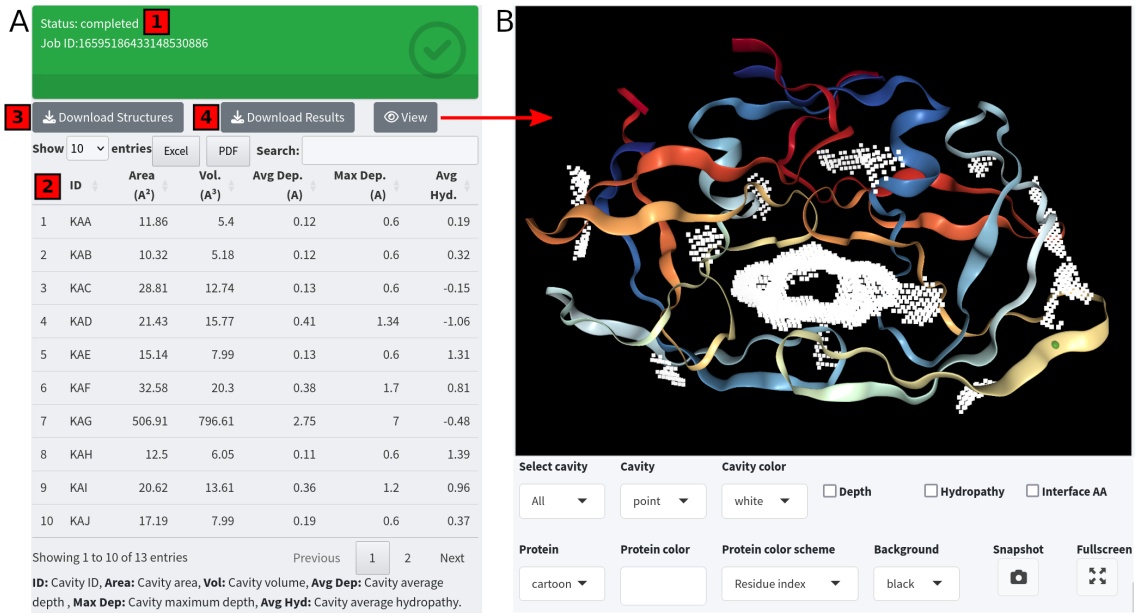


Figure 5.17: KVFinder-web portal. (A) Main page with the main tabs and sections for entering the target biomolecule and choosing the execution mode. **(B)** Detailed view of each step users must complete before submitting the target biomolecule for cavity analysis. The first step involves selecting the target biomolecule, which can be done by providing a PDB ID and searching the PDB database (1) or uploading a PDB file (2). After uploading the PDB, the KVFinder-web portal checks the PDB and informs about detected non-standard residues (3). In the next step, users must select a suitable execution mode (4) and customize, if necessary, the detection parameters (5).

In, which are usually more inclusive but not to *Probe Out*. *Removal Distance* is a distance (in Å) for removing cavity points from the cavity-border midpoint to delimit the outer limits of the cavity. *Volume Cutoff* is a cavity volume filter (in $Å^3$) to exclude cavities with volumes smaller than this limit, typically considered as functionally irrelevant cavities. For a more detailed explanation of each parameter, refer to the references [4, 5, 8–10].

Additionally, the graphical interface allows users to download and visualize results easily and interactively (Figure 5.18). Morphological (i.e., volume, area, and depth) and physicochemical (i.e., hydrophobicity) characterizations of each cavity are displayed in an interactive table, available for download in TOML format. A biomolecule viewer, powered by the graphical engine NGL for R (NGLVieweR [111]), displays the biomolecular structure with its cavities, available for download in PDB format, and allows various customizations, such as highlighting cavities and displaying interface residues around them.



Source: Reprinted from [10]. Licensed under CC BY 4.0.

Figure 5.18: Results visualization in KVFinder-web portal. (A) Results section of the KVFinder-web portal. The job status box (green: 'completed'; yellow: 'running' or 'queued'; red: 'cancelled') (1). Upon completion, the interface presents the results in a table, including volume, area, average depth, maximum depth, and average hydrophobicity of the cavities (2). Users can download a ZIP file containing the target biomolecule and PDB files of the cavities (3) or a TOML file with cavity characterizations (4). **(B)** The target biomolecule with cavities can be viewed by clicking the 'View' button, and users can customize the visualization of the biomolecule and cavities.

5.3.2 KVFinder-web service

The KVFinder-web service is a RESTful web service that employs parKVFinder (v1.2.0) to detect and characterize cavities in biomolecular structures, as described in [8–10]. Operating on a Web-Queue-Worker architecture (Figure 5.19), it handles HTTP requests and responses from the web interface, manages jobs, and executes parKVFinder on accepted jobs. Currently, KVFinder-web service is in version v1.1.1. The source code is under continuous development and available at the following repository: <https://github.com/LBC-LNBio/KVFinder-web-service>.

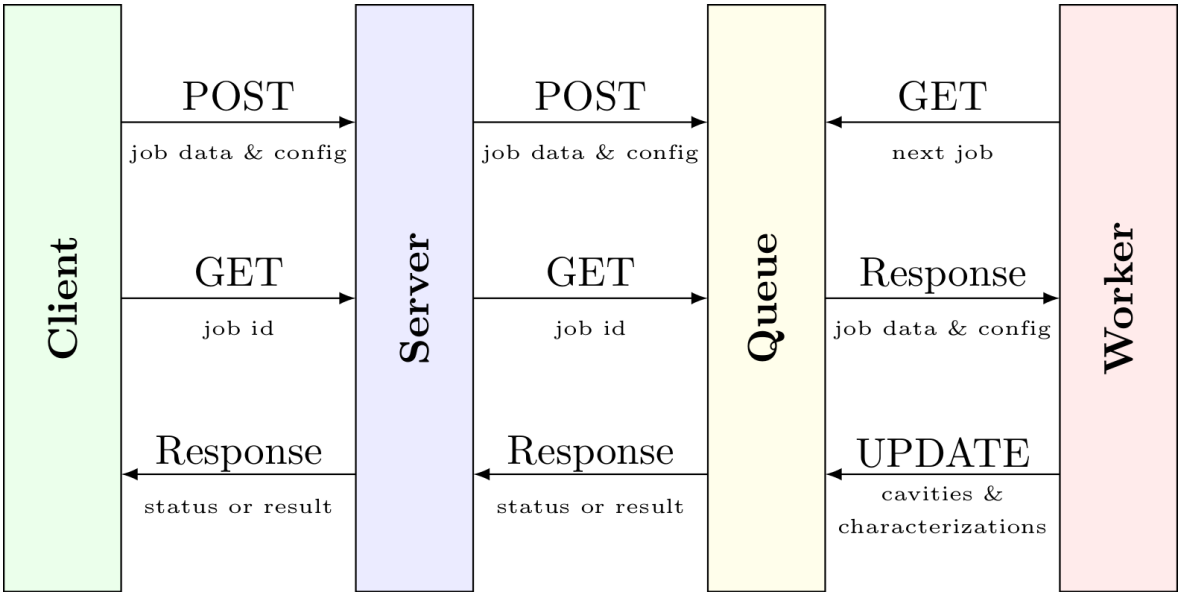


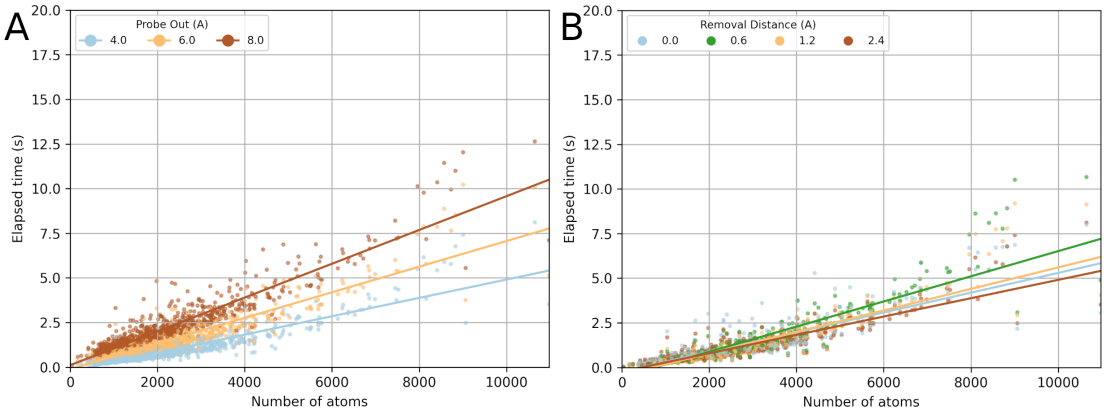
Figure 5.19: KVFinder-web architecture and communication.

The KVFinder-web service, written in Rust, comprises three modules: the *Web server*, the *Queue*, and the *Worker*. The *Web server* module, employing the Actix web framework (<https://actix.rs>), accepts job requests via HTTP POST with a JSON-based configuration file containing structural data (i.e., receptor and optionally ligand structures) and cavity detection parameters from parKVFinder. The *pdb* and *pdb_ligand* attributes store the target receptor and ligand structures, respectively, as PDB-formatted strings. Valid requests prompt the *Web server* to respond with a unique identifier, created using a hash function based on the incoming JSON data. In case of an invalid request, an HTTP error code with an error message is returned. The job ID creation through a hash function ensures that cached results are retrieved upon resenting a request.

Jobs accepted by the *Web server* are forwarded to the *Queue* module, which is an instance of the Ocypod job queue server (<https://github.com/davechallis/ocypod>), where they await processing by a *Worker* module. The *Worker* module interacts with the *Queue* module, requesting the next job for processing by parKVFinder software. To prevent resource exhaustion, parameters for cavity detection are either constrained or predefined. Upon job completion, the cavity analysis (cavities and their characterizations) is sent back to the *Queue* module, updating the job status and results. These results are then made accessible to clients through the *Web server* module.

Retrieving job results involves sending an HTTP GET request containing the job ID. The *Web server* then responds with the current job status, indicating whether it is "queued", "running", or "completed". Additionally, the corresponding results, if available, are included in the response. Each job remains cached in the queue for one day after completion. Depending on processing requirements, additional worker modules can be allocated to enable the simultaneous processing of multiple jobs. All modules of the KVFinder-web service are packaged into a Docker container [112], allowing for execution in both local or cloud computing environments. These modules are integrated into a Docker Compose file, ensuring easy deployment.

The computational performance of the KVFinder-web service was also evaluated on kv1000 [9], varying two crucial parameters associated with cavity detection: *Probe Out* and *Removal Distance* (Figure 5.20). Varying the *Probe Out* parameter creates a coarser molecular surface around the biomolecular structure, delineating the cavity-solvent boundary. Increasing the *Probe Out* reduces the degree of accessibility of the molecular surface and increases the calculation time in the KVFinder-web service. Conversely, the *Removal Distance* parameter removes cavity points close to the boundary, aiding in the identification of sub-cavities and surface cavities. The runtime does not exhibit a clear relationship with the *Removal Distance* parameter, as it is influenced by the size of the boundary and the number of cavities, not the number of atoms. Generally, the runtime increases linearly with the number of atoms in the target biomolecular structure within the 3D grid.



Source: Reprinted from [10]. Licensed under CC BY 4.0.

Figure 5.20: Effects of detection parameters on KVFinder-web service performance. Cavity detection was performed on kv1000 [9], varying **(A)** *Probe Out* sizes and **(B)** *Removal Distance*. Elapsed time relative the number of atoms in the target structure was recorded. Calculations were performed on a computer with an AMD Ryzen 7 1700 8-core, 3.0 GHz processor, 32GB of RAM, running Ubuntu 22.04 LTS.

5.3.3 PyMOL KVFinder-web Tools

For users familiar with PyMOL [56], the **PyMOL KVFinder-web Tools** (Figure 5.21), developed in Python3 and Qt, integrates the KVFinder-web service with the molecular visualization program. This user-friendly GUI allows customization of detection parameters for a target biomolecular structure and sends jobs to a configured KVFinder-web service (Figure 5.21A). Currently, PyMOL KVFinder-web Tools is in version v1.0.0. The source code is available at the following repository: <https://github.com/LBC-LNBio/PyMOL-KVFinder-web-Tools>.

Similar to the parKVFinder plugin for PyMOL [9], the search space can be customized to a specific box (box adjustment mode) and/or a radius around a ligand or target molecule (ligand adjustment mode), rather than identifying cavities across the entire biomolecular structure (whole protein mode). Upon successful submission, accepted jobs are routinely and asynchronously requested from the KVFinder-web service. After job completion, the plugin automatically processes the received data (e.g., cavities and

characterizations) into local files, making them available in the GUI. This graphical plugin functions similarly to the KVFinder-web portal; characterizations are displayed in lists (Figure 5.21B), and cavities are customized based on their characterizations in the PyMOL viewer (Figure 5.21C and D). However, jobs submitted in the KVFinder-web portal can be loaded into PyMOL KVFinder-web Tools, and vice versa.

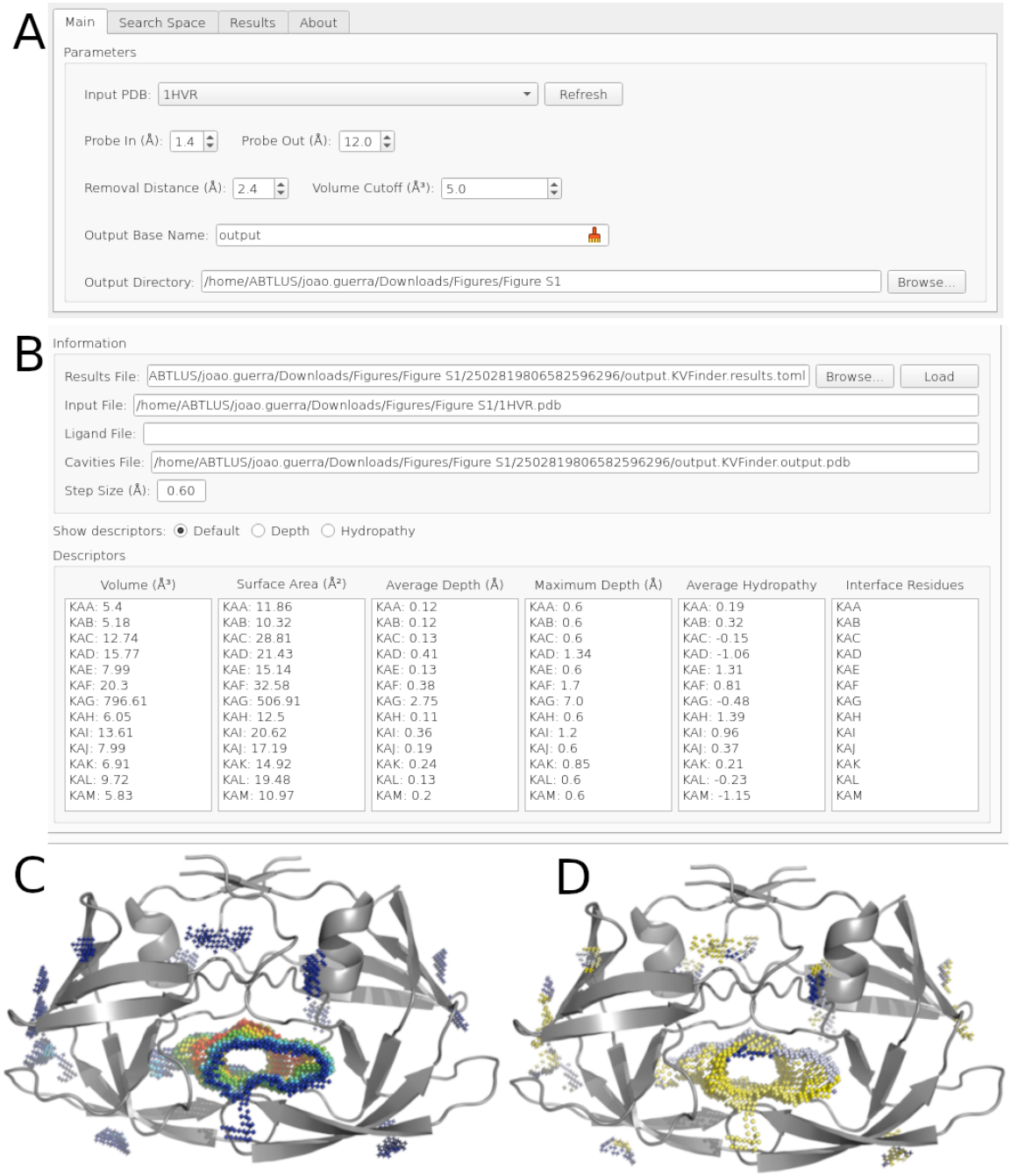


Figure 5.21: PyMOL KVFinder-web Tools. Cavity detection in the HIV-1 protease structure (PDB ID: 1HVR), with *Probe Out* set to 12 Å. **(A)** Main parameters tab containing detection parameters and molecular structures to explore. **(B)** Visualization tab displaying data received (cavities and characterizations) from the KVFinder-web service shown in the GUI. **(C)** Depth characterization and **(D)** Eisenberg Weiss hydrophobicity characterization, highlighting the active site (KAG cavity) in the GUI and PyMOL viewer.

5.3.4 Monitoring and Analytics

The monitoring and analytics of KVFinder-web are essential to ensure the web application's optimal functionality and deliver a high-quality user experience. This process provides valuable insights into user interaction, enabling maintainers and developers to tailor improvements and updates to user requirements. For a comprehensive approach, we implemented a monitoring process to track job execution and user behavior over time.

Job submission tracking provides valuable information into system usage patterns, helping identify peak demand periods and optimize resources for smooth and efficient user experience. Additionally, it helps in assessing user adherence to our platform and gauging the impact of new features and updates. From March 1, 2023, to January 31, 2024, KVFinder-web executed 1,320 jobs, averaging of ∼120 jobs per month (Figure 5.22). During the developmental phase from March 2023 to May 2023, when KVFinder-web was accessible only to internal users from CNPEM, it averaged ∼65 jobs per month. After its public release in June 2023, alongside its publication in *Nucleic Acids Research* [10], KVFinder-web had an average of ∼141 jobs per month, indicating user adherence and the publication's impact.

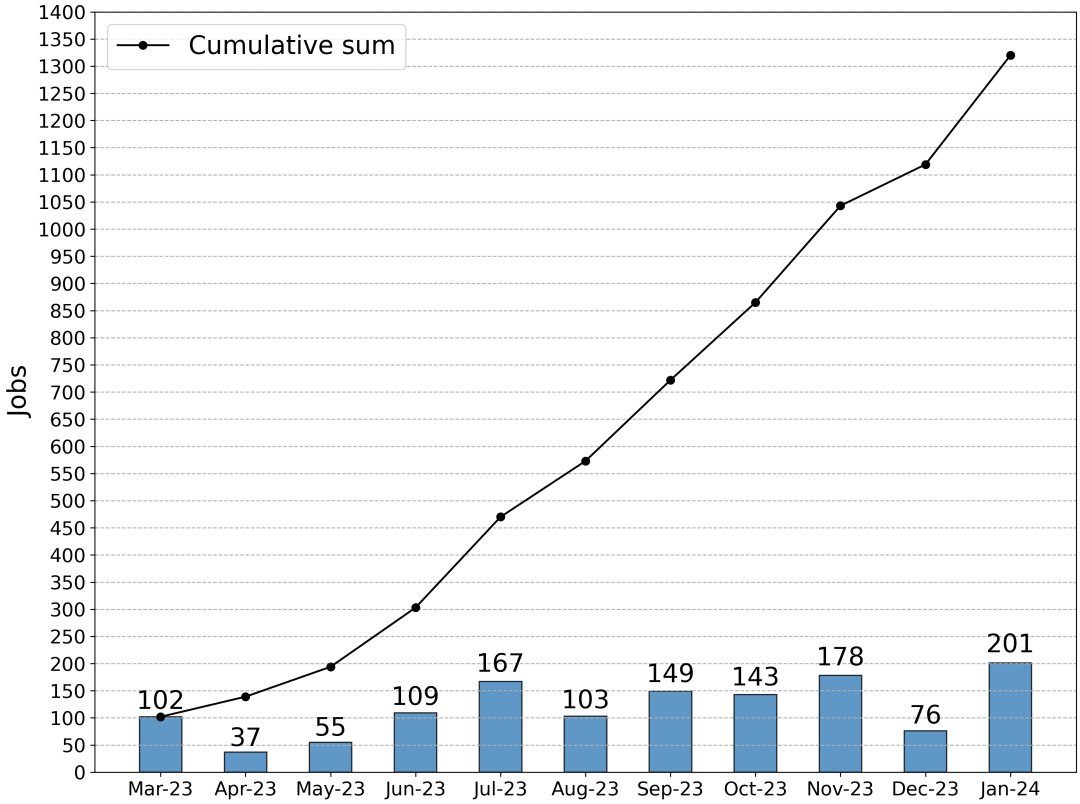


Figure 5.22: Jobs executed in KVFinder-web. Accepted jobs executed in KVFinder-web per month and their cumulative sum.

Understanding user behavior through analytics is crucial for KVFinder-web development and maintenance, guiding iterative improvements, optimizing interface design, identifying popular features, and ensuring a smooth and efficient user experience. Initially, Cloudflare (<https://www.cloudflare.com>) was explored for user behavior ana-

lytics, but its collected data did not provide the desired insights due to its broader focus on content delivery, security, and performance services. Its monitoring capabilities are more centralized around overall website health and security. From October 21, 2023 to November 20, 2023, Cloudflare recorded 1,130 visitors, mainly from the United States (38.6%), Netherlands (19.2%), Germany (9.6%), China (7.2%), Spain (1.8%), and Portugal (1.8%).

Subsequently, Microsoft Clarity (<https://clarity.microsoft.com>) emerged as a more suitable solution, offering a privacy-focused analytics tool that does not collect personal information (e.g., IP addresses or cookies). Leveraging Microsoft Clarity allowed for a deeper analysis of user interactions, offering valuable insights into navigation and engagement (e.g., click tracking, heatmaps, masked session recordings, and engagement metrics). Since its implementation from November 21, 2023 to January 31, 2024, KVFinder-web received 301 visitors, with 119 unique users from different countries, primarily United States(20.6%), India (15.9%), China (11.3%), Brazil (9.6%), and South Korea (5.6%) (Figure 5.23A). Notably, the release of KVFinder-web has broadened our user base beyond Linux users, as users from different operating systems, such as Windows and MacOS, are now adopting KVFinder-web for cavity analysis (Figure 5.23B). The average active session time was 6.7 minutes, with a variety of browsers being used (Figure 5.23C). Microsoft Clarity has consistently provided valuable insights into user behavior, aiding in refining the user experience and addressing potential pain points. The data collected indicates that KVFinder-web is democratizing of parKVFinder in the structural biology community, with users worldwide and an average of ∼141 jobs per month since its official release.

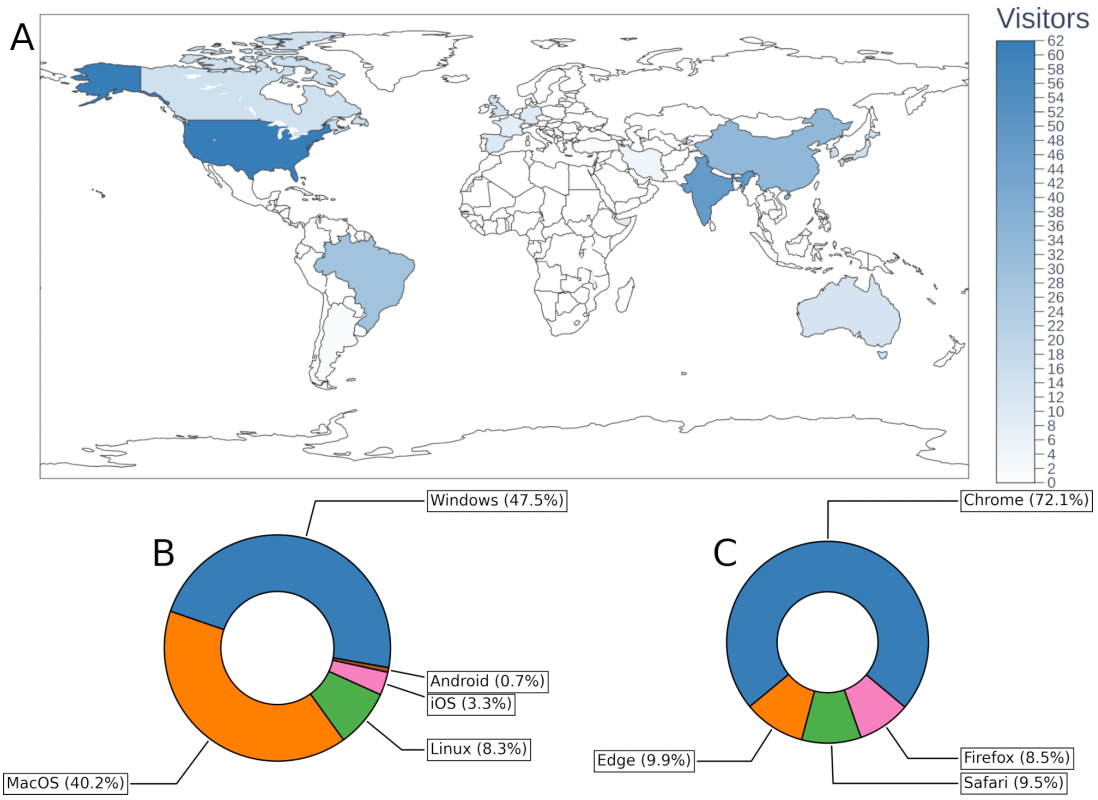


Figure 5.23: KVFinder-web user behaviour analytics. (A) Countries. **(B)** Operating systems. **(C)** Browsers.

Finally, a crucial consideration in this monitoring process is data privacy. Recognizing the sensitivity of user data, stringent measures have been implemented to uphold privacy standards. Importantly, no one in the maintainer and developer teams has access to data submitted to KVFinder-web, ensuring the utmost privacy for our users. Additionally, KVFinder-web does not collect any personal information (e.g., IP addresses or cookies), and all data collected is anonymized.

5.3.5 Case Studies

The KVFinder-web was applied in two case studies published in a scientific journal to investigate proteins of therapeutic interest. These analyses explored the characterization of the catalytic site of the HIV-1 protease and the morphological comparison of the structures of this protein deposited in the wwPDB. Next, we will describe each of these case studies in detail.

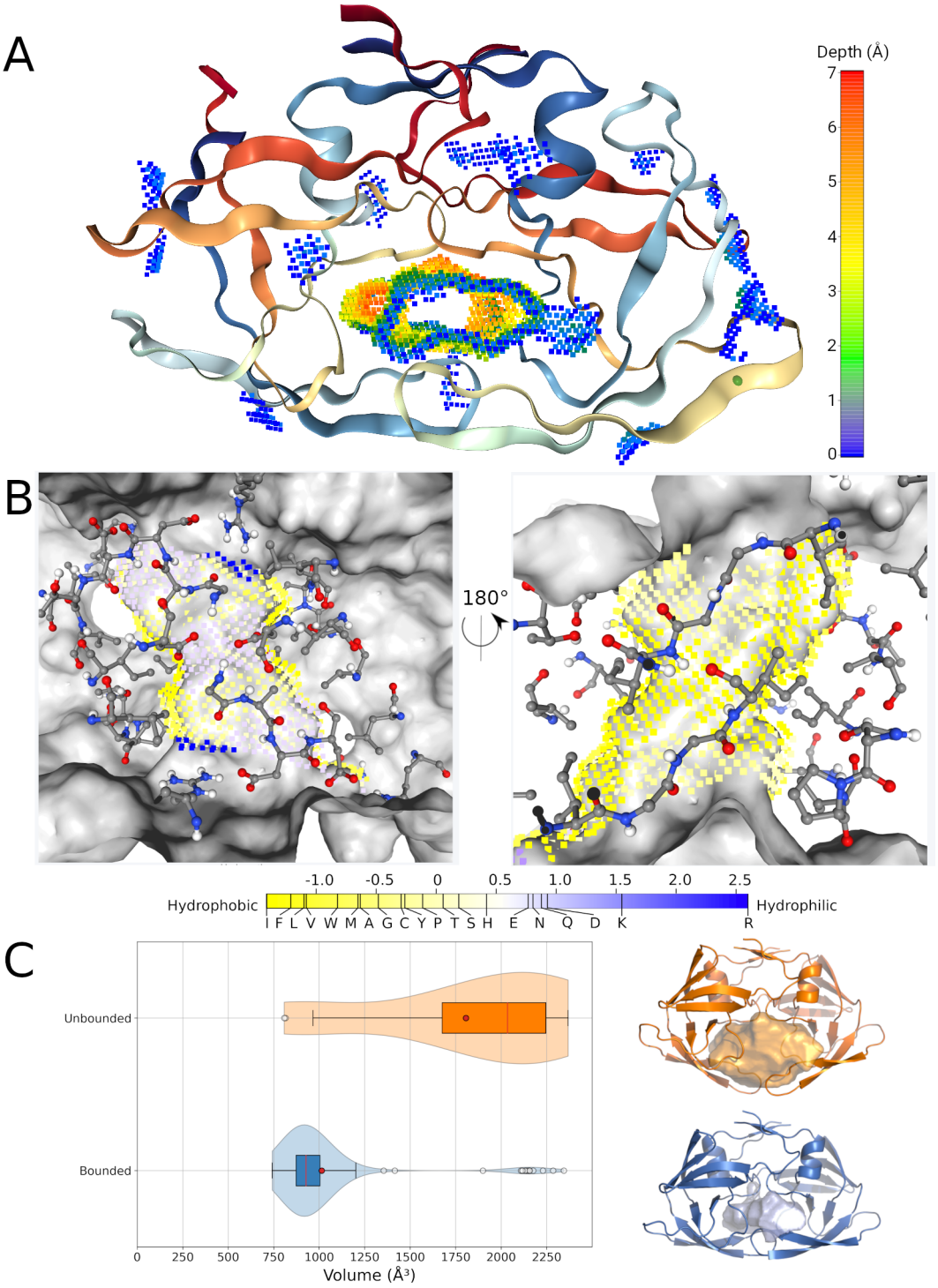
5.3.5.1 Characterization of the Catalytic Site of the HIV-1 Protease

Illustrating the capabilities of KVFinder-web (Figure 5.24), we conducted a comprehensive analysis of the catalytic site of the HIV-1 protease, bound to the carbonyl oxygen of cyclic urea [84], utilizing the KVFinder-web portal. This analysis successfully identified and characterized cavities across the biomolecular surface of the HIV-1 protease (Figure 5.24A). Beyond mere detection and shape definition, KVFinder-web provides detailed metrics, including volume, area, depth, and hydropathy of the identified cavities (Figure 5.18; KAG cavity). Usually, the active site is the largest and deepest cavity in the enzymatic protein [20], and this is also observed in the HIV-1 protease, as shown in Figure 5.18. In this sense, depth characterization can help researchers identify the active site across the molecular surface (Figure 5.24A).

As presented in Sections 5.1.1.2 and 5.2.2.1, hydropathy characterization gives valuable insights into the types of interaction and the water attractiveness of the binding site. In this scenario, an analysis of the hydrophobicity profile of the active site revealed a predominantly hydrophobic nature, particularly in the region near the catalytic aspartic acids (Asp-25 and Asp-25') β-hairpins, with limited hydrophilic regions around these catalytic residues, i.e.Asp-25 and Asp-25' (Figure 5.24B). As expected, subsites S1 and S1' (yellow region) demonstrated hydrophobicity, and subsites S2 and S2' predominantly exhibited hydrophobic characteristics (yellow region), with exceptions like Asp-29 and Asp-30 (blue region). Moreover, the hydrophobic portions of subsites S2 and S2', accommodating substrates P2 and P2', respectively, displayed a preference for aliphatic side chains [84, 113, 114].

5.3.5.2 Morphological Comparison of the Catalytic Site of HIV-1 Protease Structures

As discussed in Section 5.1.1.1, the HIV-1 protease is an effective therapeutic target, with its catalytic site being target of various antiretroviral drugs. The catalytic



Source: Reprinted from [10]. Licensed under CC BY 4.0.

Figure 5.24: Illustrative example of cavity detection and characterization in the HIV-1 protease. (A) Cavities detected throughout the HIV-1 protease (PDB ID: 1HVR). Cavity points are colored according to depth in a rainbow color scale. **(B)** Hydropathy mapped on surface cavity points in regions around the catalytic aspartic acids (left panel) and around the β-hairpins (right panel). The Eisenberg Weiss hydrophobicity scale ranges from -1.42 (highly hydrophobic) to 2.6 (highly hydrophilic). The protein is shown as a gray surface, and interface residues are shown as colored atoms in sticks. **(C)** Violin plot of the active site volume of HIV-1 protease structures from the RCSB PDB for structures with ligands bound at the active site (blue) and structures without ligands (orange). Structures with a median volume and the corresponding cavity are shown as a cartoon and surface model, respectively.

cycle hinges on the movements of the β-hairpins, controlling the substrate accessibility to the active site [84, 85]. Consequently, we conducted a comprehensive analysis of HIV-1 protease structures available in the PDB, comparing their cavities (refer to [10]). Thus, cavity volumes proved instrumental in clearly distinguishing ligand-bound and non-ligand-bound structures (Figure 5.24C), indicating geometric complementarity between the receptor and ligands, supported by physicochemical complementarity elucidated by the hydrophobicity profile.

5.3.6 Discussion

The KVFinder-web represents a significant advancement in the field of biomolecular structural and functional characterization, offering a user-friendly and efficient solution for cavity detection and characterization. The KVFinder-web portal, along with the PyMOL KVFinder-web Tools, provides an intuitive and user-friendly interface for users to perform cavity detection and characterization on any biomolecular structure. Both interfaces are designed with careful considerations to prevent overutilization of computational resources, ensuring a smooth and efficient operation of the KVFinder-web service. For users more familiarized with, the PyMOL plugin mirrors the key functionalities of the parKVFinder PyMOL plugin. The KVFinder-web service is a robust and scalable web service, providing a reliable and efficient computational infrastructure for cavity detection and characterization. Together, these components democratize the usage of parKVFinder software within the scientific community (e.g., scientists, educators, and students), eliminating barriers for users who may face challenges in employing computational tools independently.

In essence, KVFinder-web significantly simplifies the process of detecting and characterizing cavities, making it accessible even to less experienced users. Thus far, our data presented ∼141 jobs per month since its official release, indicating the adoption and utility of KVFinder-web in the scientific community. The success of KVFinder-web, as evidenced by its monitoring and analytics, highlights its effectiveness in achieving the goals of simplifying cavity analysis and promoting broader engagement within the scientific community.

5.4 SERD

Molecular recognition hinges on the accessibility of a ligand to the binding site of its corresponding receptor. The atoms exposed to the solvent in a target biomolecule represent the potential interaction points for ligands. However, the identification of protein-protein interfaces through conventional methods, such cavity detection, poses a challenge. These interfaces are often large, flat, and featureless, earning the classification of "undruggable". PPIs typically involve contiguous epitopes of one partner and a well-defined groove or series of specific small pockets [115], requiring the recognition of exposed residues to facilitate a more focused study of interaction hotspots in a target receptor, especially in molecular docking studies (e.g., protein-ligand docking and protein-protein docking).

In this scenario, we developed Solvent-Exposed Residues Detection (SERD) [116], an open-source tool licensed under GPL v3.0, designed to detect solvent-exposed residues of a target biomolecule (Figure 5.25). The algorithm employs a spherical probe, approximating a solvent or ligand molecule, to scan the target biomolecule and identify regions (i.e., residues) accessible to this spherical probe. This process mirrors the 3D grid scanning of the *Probe Out* in parKVFinder (Figure 3.3). Essentially, the probe size establishes a cutoff for the solvent- or ligand-accessible surfaces (Figure 5.25B), selectively picking residues beyond this cutoff (Figure 5.25C).

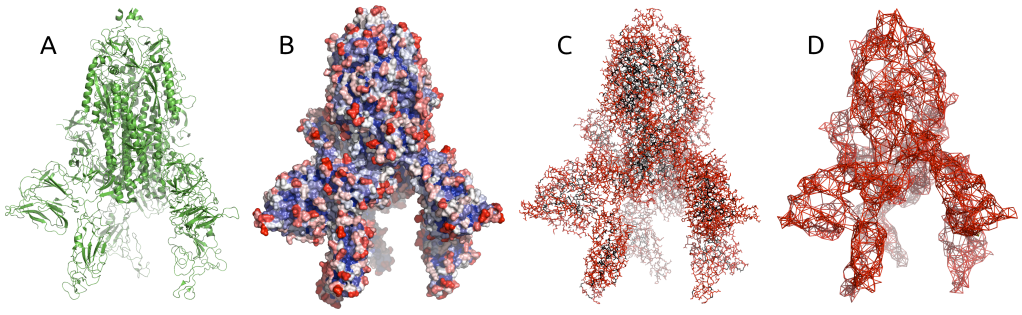


Figure 5.25: Solvent accessibility and graph-based representation in SERD. (A) SARS-CoV-2 Spike Glycoprotein (PBD ID: 7A98; green cartoon). **(B)** Solvent accessibility, from low accessibility (blue) to high accessibility (red). **(C)** Solvent-exposed residues (red sticks). **(D)** Graph-based representation of solvent-exposed residues. Edges are formed up to the limit distance of 10 Å between Cα (red).

Transitioning to a broader context, graph theory has been relevant in both biology and pharmaceutical sciences, applied in the study of protein structure, function, and evolution, as well as in MD analysis and receptor-ligand interaction networks [74,80,117]. As presented in Graphinity [117], a graph-based representation of protein-protein complexes, such as antibody-antigen complexes, can be harnessed in deep learning (DL) techniques for predicting experimental $\Delta\Delta G$. Moreover, graph-based representations can be employed in a broad range of machine learning techniques and data science applications, as exemplified in Section 4.3 and discussed in previous studies [73,74,80]. In this context, SERD further contributes by representing exposed residues in graph form using the NetworkX library [118]. This involves forming edges up to a specified distance between Cα, Cβ, or any atoms of the residue, optionally including these distances as attributes of the edges of these graphs, as illustrated in Figure 5.25D.

Expanding the scope beyond its initial purpose, the graph-based representations developed in SERD find applicability in binding sites detected by pyKVFinder (Figure 5.26). This topological representation takes into account intramolecular interactions to construct the edges of the graphs, offering novel perspectives for analyzing binding sites in biomolecules using graph theory.

In summary, SERD can represent biomolecular structures as graphs, from ligand-binding sites (e.g., cavities identified by the KVFinder suite) to biomolecular complexes (e.g., PPIs, PLIs, PRIs, and PDIs), as shown in Figure 4.5. To date, this tool has been applied in two collaborations at LBC: the Master's project of student Marcos Rogério Simões from PPGCF/FCF, entitled "Descrição e caracterização de sítios de ligação através de teoria dos grafos", studying the similarity between binding sites among kinase families, and the research project, led by Dr. Gabriel Ernesto Jara, explor-

ing antibody-antigen complexes through surface fragments, represented as graphs. Currently, SERD is in version v0.1.2. The source code is available at the following repository: <https://github.com/LBC-LNBio/SERD>.

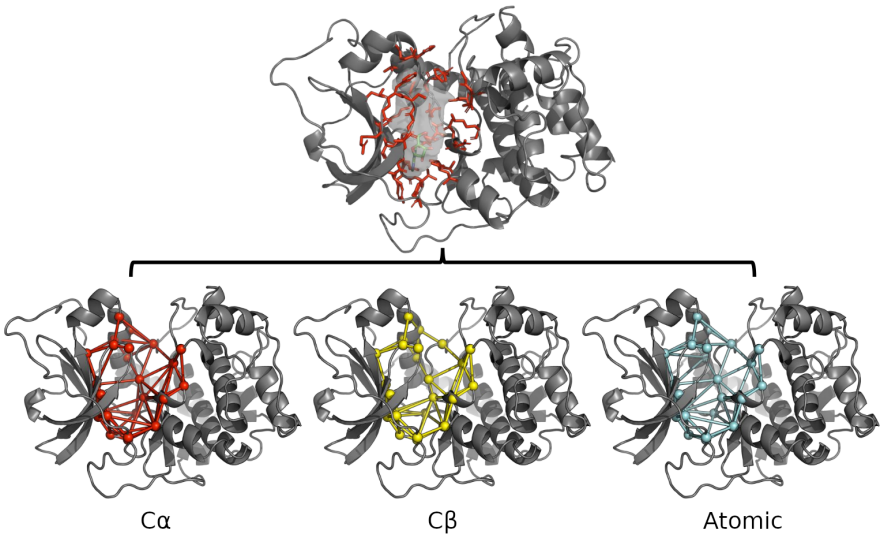


Figure 5.26: Graph-based representation of the adenosine binding site of protein kinase A (PDB ID: 1FMO). Upper panel: Binding site represented in pyKVFinder as a cavity (gray surface) and by the residues surrounding the cavity (red). Lower panel: Binding site represented as graphs. Edges are formed up to the limit distance of 10 Å between Cα (red), 8 Å between Cβ (yellow), and 5 Å between any atoms of different residues (cyan).

5.5 KVFinderMD

In specific scenarios, receptors utilize binding sites for the formation of the receptor-ligand complex, which are not easily identified in the unbound form. Certain biomolecular interactions (e.g., PPIs, PLIs, PRIs, and PDIs) rely on the intrinsic dynamics of the target receptor, where the classical lock-and-key model fails, and more recent binding models, e.g., induced fit and conformational selection, thrive. In summary, while the lock-and-key model implies a rigid active site, the induced fit model introduces flexibility in both enzyme and substrate, and the conformational selection model highlights the role of pre-existing enzyme conformations that can be selected by substrate binding [11] (Figure 5.27). In this context, MD simulations serve as a useful tool to comprehend molecular recognition process and, ultimately, biomolecular function.

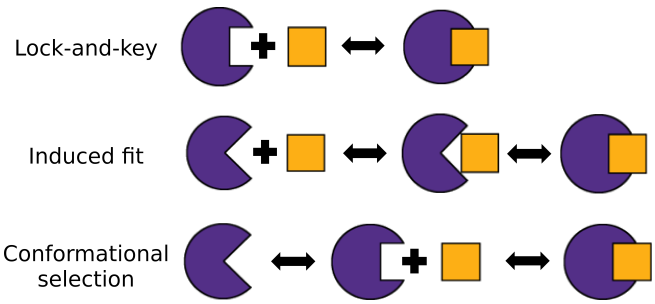


Figure 5.27: Schematic representation of binding models.

Recently, we developed pyKVFinder [5] as a foundational element for more complex applications, such as analyses of MD simulations. Building upon it, we developed a new tool, called KVFinder for Molecular Dynamics analysis (KVFinderMD), a Python package to explore binding site dynamics in biomolecular structures of interest (Figure 5.28). For the reading of binary data generated by MD simulation programs such as GROMACS [69], AMBER [70], and CafeMol [71], we employ the MDAnalysis package [119] to integrate the reading and processing of MD trajectory files (e.g., GRO, CRD, NC, DCD, XTC/TRR) and topology files (e.g., PSF, PRMTOP, GRO, PDB) into KVFinderMD. Given that the intrinsic dynamics of the biomolecule can alter the shape and properties of the binding site over time, KVFinderMD characterizes cavities in respect to volume, area, depth, hydrophobicity, and interface residues—properties crucial for describing the molecular recognition process (Figure 5.28A). Additionally, we implemented a protocol for the analysis of cavity conservation throughout MD (Figure 5.28A; bottom right panel), similar to the conservation analysis of the ADP-ribose binding site in the ADRP domain of SARS-CoV-2 and related proteins (Figure 5.14). Besides that, we also integrated the graph-based representation implemented in SERD (see Sections 4.3 and 5.4), considering Cα, Cβ, or atomic distances, to describe the binding site (Figure 5.28B). With it, cavity similarity can be determined by hierarchically clustering different cavity representations available in KVFinderMD (i.e., 3D grid, residue-level representation, and graph-based representation), allowing for tracking cavities throughout MD simulation.

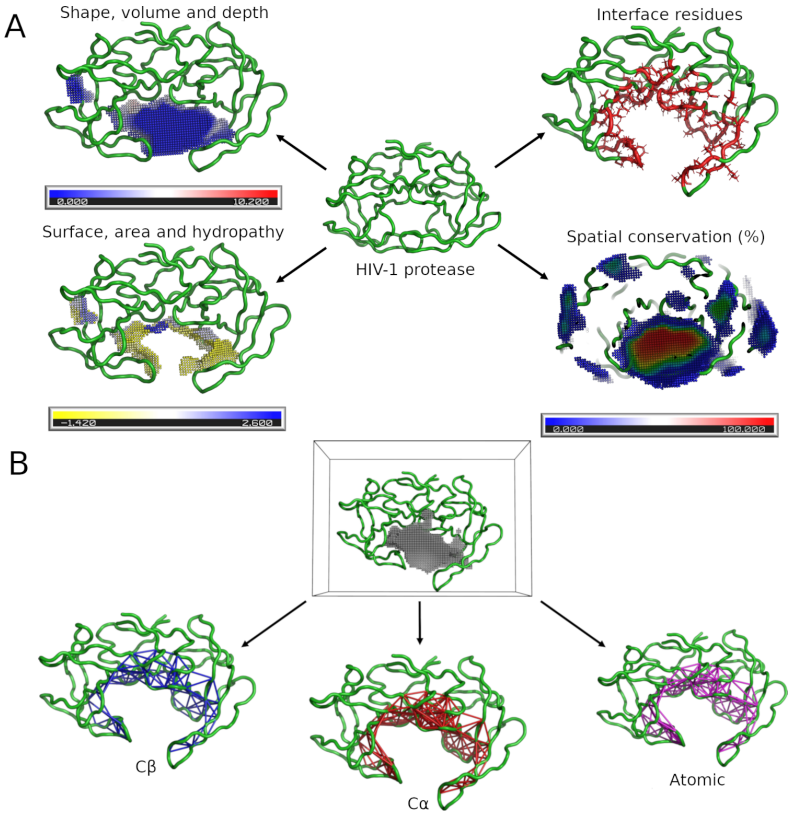


Figure 5.28: Detection, characterizations, and representations of cavities in molecular dynamics studies using the KVFinderMD tool. (A) Morphological, topological, and physicochemical characterizations applied in KVFinderMD. **(B)** Graph-based representation of cavities based on interface residues and their topological relationships. Edges are formed up to the limit distance of 8 Å between Cβ (blue), 10 Å between Cα (red), and 5 Å between any atoms of different residues (magenta).

5.5.1 Case Study

The KVFinderMD was applied in a case study to explore the cavity similarity of the HIV-1 protease throughout a MD simulation. This case study was presented at the *Congresso de Estudantes do CNPEM* (CEC), held between November 22 and 24, 2022, and received an award for the presentation entitled "KVFinderMD: a Python package to detect and describe binding sites in molecular dynamics trajectories".

5.5.1.1 Cavity Similarity of HIV-1 Protease throughout Molecular Dynamics Simulation

Computational tools for studying MD simulations face challenges in defining the continuity of cavities over time, mainly due to shape changes and the merging and splitting of cavities. Employing techniques to group cavities based on their topology enables the tracking of their continuity throughout MD simulations. To address this issue algorithmically, cavity similarity is calculated through a structural alignment. From the MD simulation of the HIV-1 protease, where we successfully described the conformational dynamics of the active site (see Section 5.1.1.1), we developed a structural cavity alignment methodology to determine cavity similarity throughout a MD simulation, using KVFinderMD. The methodology consists of three steps:

1. Detection and characterization of cavities in the HIV-1 protease throughout the entire MD simulation using KVFinderMD;

2. Representation of these cavities in various formats, including 3D grid representation, residue-level representation, and graph-based representation;

3. Application of hierarchical clustering by KVFinderMD to group cavities based on the similarity of their representations, exploring different affinity metrics.

In the MD simulation of the HIV-1 protease, we detected 672 cavities over 201 simulation frames, using *Probe Out* of 12 Å and *Volume Cutoff* of 50 Å^3. Subsequently, each cavity was treated as an independent data structure—represented in a 3D grid, residue-level representation, and graph-based representation (Figure 5.29). The 3D grid represents cavities as a boolean grid of dimensions (160, 126, 105), totaling $2.1 \cdot 10^6$ voxels. Cavity points are assigned with 1, while other points (biomolecule, solvent, and empty space) are marked as 0. For the residue-level representation, cavities are expressed as a square matrix depicting distances between constituent residues. In the graph-based representation, cavities are represented as a square matrix presenting contacts between constituent residues. These contacts are determined by the distance between the Cβ atoms of residues, employing a distance cutoff of 8 Å, the default metric of SERD. Both square matrices are ordered according to the protein sequence with 198 aminoacids, totalizing $3.9 \cdot 10^4$ points.

With these cavity representations, clustering analysis groups them based on their similarity. Hierarchical clustering, compared to other unsupervised clustering methods, yield more interpretable results, by providing a similarity relationship between cavity

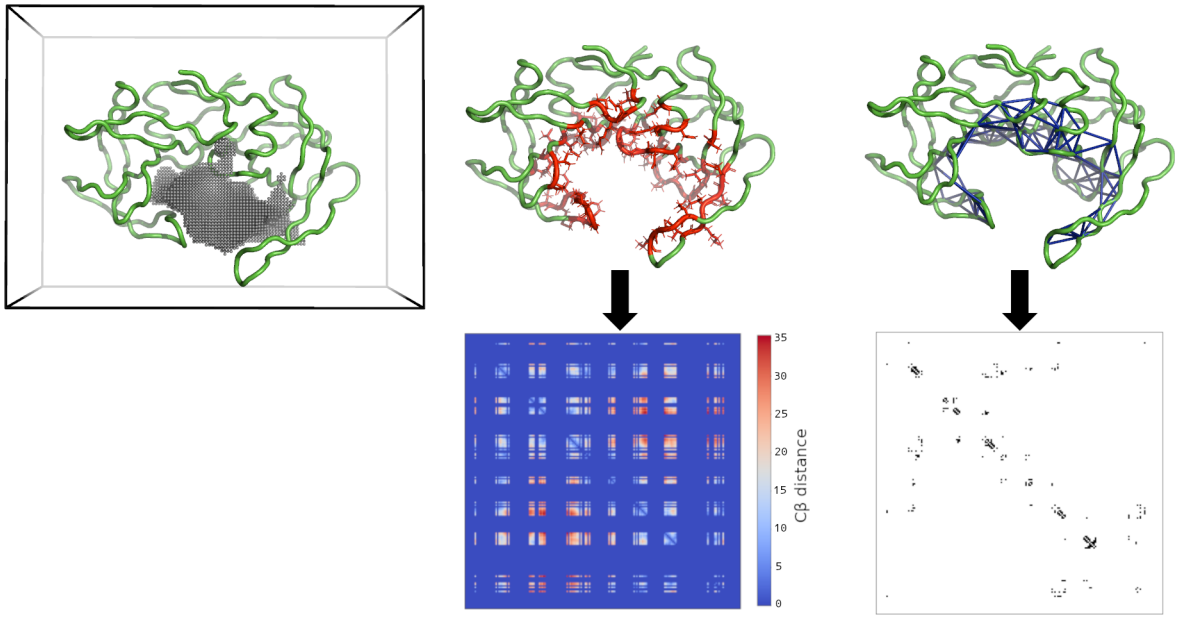


Figure 5.29: Cavity representation for the study of HIV-1 protease cavity similarity throughtout molecular dynamics trajectory. Representation of the active site cavity in 3D grid (left panel), where cavity points are identified by 1, while other points (biomolecule points, solvent points, and empty space points) are identified by the value 0. Residue-level representation of the active site (central panel), where we abstract cavities as a distance matrix between Cβ atoms of residues, ordered by the protein sequence. Graph-based representation of relationships between active site residues (right panel), where we abstract cavities as a contact matrix, ordered by the protein sequence.

representations. As there is no ground truth for an unlabeled cluster analysis (i.e., structural cavity alignment), the silhouette score (s; Eq. 5.1), proposed by Peter Rousseeuw [120], was used to evaluate the quality of the clustering. This metric measures the similarity of a cavity to its own group (cohesion) compared to other groups (separation), based on an affinity metric. Thus, achieving a silhouette score of 1 is optimal for cluster validity, while a score of -1 signifies poor clustering, and 0 implies clusters with overlapping boundaries.

$$s\colon (i) \mapsto \frac{(b(i) - a(i))}{max(a(i), b(i))} \in [-1, 1] \tag{5.1}$$

where a is the mean intra-cluster distance, b is the mean nearest-cluster distance, and i is a given cavity.

In simpler terms, 672 data structures (i.e., cavities) are clustered by hierarchical clustering, exploring appropriate affinity metrics for each cavity representation. Since the silhouette score cannot be compared between different affinity metrics, we compared affinity metrics based on the ideal number of clusters by maximizing the mean silhouette score over all cavity representations (Eq. 5.2), as proposed by Leonard Kaufmann and Peter Rousseeuw [121].

$$k \mapsto \underset{k}{\operatorname{argmax}}\, \tilde{s}(k) \in [2, \infty) \tag{5.2}$$

where $\tilde{s}(k)$ represents the mean $s(i)$ over all cavities of the MD simulation for a specific number of clusters k.

For non-negative real-valued vectors, we evaluated the following metrics available in the SciPy package [92]: correlation distance (ρ; Eq. 5.3), Bray-Curtis distance, Canberra distance, Chebyshev distance, Manhattan distance (also known as City Block distance), cosine distance (d_{cosine}; Eq. 5.4), Jensen-Shannon distance, and squared Euclidean distance. For boolean vectors, we evaluated the following metrics available in the SciPy package [92]: Dice dissimilarity (d_{dice}; Eq. 5.5), Hamming distance, Jaccard-Needham dissimilarity, Kulczynski dissimilarity, Rogers-Tanimoto dissimilarity, Russell-Rao dissimilarity, Sokal-Michener dissimilarity, Sokal-Sneath dissimilarity, and Yule dissimilarity.

$$\rho\colon (u, v) \mapsto 1 - \frac{(u - \bar{u}) \cdot (v - \bar{v})}{\|(u - \bar{u})\|_2 \|(v - \bar{v})\|_2} \in [-1, 1] \tag{5.3}$$

where $\bar{u}$ is the mean of elements in u, $\bar{v}$ is the mean of elements in v, and $x \cdot y$ is the dot product of x and y.

$$d_{\text{cosine}}\colon (u, v) \mapsto 1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2} \in [-1, 1] \tag{5.4}$$

where $u \cdot v$ is the dot product of u and v.

$$d_{\text{dice}}\colon (u, v) \mapsto \frac{C_{VF}(u, v) + C_{FV}(u, v)}{2C_{VV}(u, v) + C_{VF}(u, v) + C_{FV}(u, v)} \in [0, 1] \tag{5.5}$$

where C_{ij} is the number of occurrences of $u[k] = i$ and $v[k] = j$ for $k < n$.

Following this methodology for structural cavity alignment, cavities are grouped by hierarchical clustering, using the complete linkage method, optimizing the mean silhouette score to find the ideal number of clusters for each affinity metric. Then, the clustered cavities and their respective silhouettes scores of the affinity metrics with the highest number of clusters, for each cavity representation, are presented in Figure 5.30. The structural cavity alignment results are summarized in Table 5.1.

Table 5.1: Summary of structural cavity analysis in the molecular dynamics simulation of the HIV-1 protease.

	3D grid aligment	**Distance matrix alignment**	**Contact matrix alignment**
Clusters	10	15	16
Data size	$\sim 10^6$/cavity	$\sim 10^4$/cavity	$\sim 10^4$/cavity
Data type	Boolean (4-bit)	Float (32-bit)	Boolean (4-bit)

The hierarchical clustering of the 3D grid representation resulted in 10 groups with a mean silhouette score of ∼0.42. Due to data size ($\sim 10^6$ points per cavity), we did not explore different affinity metrics, and therefore, we only used correlation distance (Eq. 5.3). For the residue-level representation, we explored different affinity metrics, and cosine distance (Eq. 5.4) showed the highest number of groups among non-negative real-valued metrics (i.e., 15 groups) with a mean silhouette score of ∼0.54. For the graph-based representation representation, we explored different affinity metrics, and Dice dissimilarity

(Eq. 5.5) showed the highest number of groups among boolean metrics (i.e., 16 groups) with a mean silhouette score of ∼0.52.

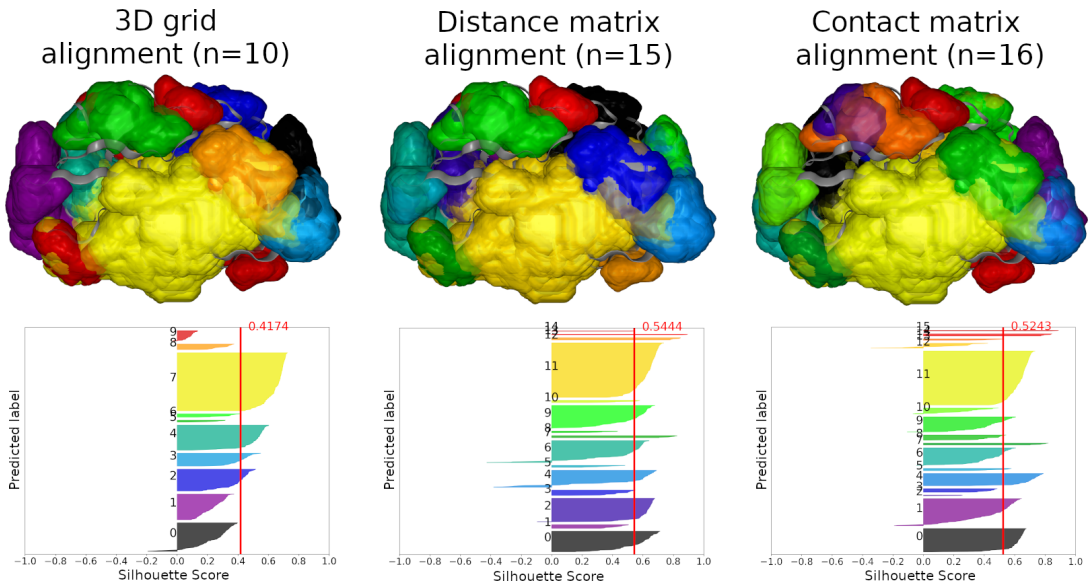


Figure 5.30: Hierarchical clustering of HIV-1 protease cavities. All cavities detected throughout the molecular dynamics were overlaid on the same structure and colored according to labels assigned by hierarchical clustering of 3D grid (left panel), distance matrix (central panel), and contact matrix (right panel). The silhouette score plot is presented for each sample according to the assigned label. The red line indicates the silhouette coefficient for hierarchical clustering.

Furthermore, the data size of the cavity representations highly influences the time complexity of the hierarchical clustering, which is $\mathcal{O}(n^3)$, where n represents the number of points. Thus, the 3D grid is the most computationally intensive ($\sim 10^6$ points per cavity), followed by the residue-level representation ($\sim 10^4$ points per cavity) and the graph-based representation ($\sim 10^4$ points per cavity). With data size simplification for the latter two representations, it was possible to explore different affinity metrics. Basically, the data size decreases by 100 times per cavity compared to the 3D grid structure. Therefore, considering we have 672 cavities in this case study, the size reduction is approximately 67,200 times. Conversely, the data type used by each cavity representation influences the memory usage for the structural cavity alignment. The 3D grid representation uses boolean (8-bit), while the residue-level representation uses float (32-bit floating-point) and the graph-based representation use boolean (8-bit). Thus, the graph-based representation uses less memory than the other representations, with a smaller data size. In summary, the 3D grid representation is the most computationally intensive, followed by the residue-level representation and the graph-based representation.

In summary, we successfully grouped cavities and tracked their continuity throughout the MD simulation, which simplified the comparison of characteristics between cavities over time. The use of distance matrices and contact matrices proved effective due to their ability to cluster cavities. Moreover, these representations were faster and simpler for clustering cavities over time in MD simulations since the hierarchical clustering algorithm has a time complexity of approximately $\mathcal{O}(n^3)$, and the data size decreases by 100 times per cavity compared to the original structure (i.e., 3D grid).

5.5.2 Discussion

The application of KVFinderMD illustrates the integration of tools within the KVFinder suite (i.e., pyKVFinder and SERD) to conduct a systematic analysis of cavities through an automated protocol. By employing hierarchical clustering algorithms and different affinity metrics across various representations (i.e., 3D grid, residue-level representation, and graph-based representation), we identified clusters of similar cavities, enabling the assessment of the temporal evolution of binding sites (i.e., cavity continuity). Additionally, we managed to reduce the data size required for analysis and simplify the comparison between cavities in MD simulations, showcasing the practical applicability of the KVFinder suite.

The analysis of cavity similarity in the HIV-1 protease throughout the MD simulation allowed us to track the evolution of these cavities and compare their characteristics over time, as depicted in Figure 5.3. This was achieved without human intervention, utilizing the automated features of KVFinderMD. However, it is important to note that, while KVFinderMD is a valuable tool for cavity analysis in MD simulations, there are some limitations to consider. For instance, the choice of affinity metrics and hierarchical clustering parameters can influence the obtained results. Therefore, careful evaluation and exploration of different configurations are essential for optimal outcomes.

Despite these limitations, KVFinderMD represents a significant advancement in the study of biomolecular interactions over time. ts ability to automate cavity analysis in MD simulations, coupled with other tools in the KVFinder suite, provides a comprehensive approach to investigate binding sites and understand their evolution. This can be valuable in the development of therapeutic strategies and the rational design of new drugs, especially in transient binding sites.

5.6 Benchmarking of Well-established Cavity Detection Tools

Cavity detection methods are usually only benchmarked on their ability to detect binding sites (i.e., qualitative assessment), and not on their ability to accurately characterize their volume (i.e., quantitative assessment). Quantitative descriptors still remains a challenge in the cavity detection field, because the "real" cavity volume in biological systems is not experimentally measurable. In collaboration with Dr. György Szalóki (Laboratoire Hétérochimie Fondamentale et Appliquée - Université Toulouse III Paul Sabatier - France), we developed a benchmarking protocol to evaluate the performance of cavity detection tools in a dataset of well-defined artificial supramolecular cages. This benchmarking protocol was applied to the well-established cavity detection tools, including KVFinder suite, Fpocket, GHECOM, and CAVER tools. The cavity detection benchmarking procedure, optimized detection parameters, and detailed results are available in the article published in the *Journal of Chemical Information and Modeling* [36]. The source code for the benchmarking protocol (v1.0.0) is available at the following repository: <https://github.com/LBC-LNBio/SMC-Benchmarking>.

The "real" cavity volume could be derived from the Rebek's rule of thumb [122], that states, in any biological or artificial host-guest system, the ratio of the guest and host's cavity is 0.55 ± 0.09, termed as packing coefficient (PC; Eq. 5.6), when guest encapsulation is only driven by weak intermolecular interactions (e.g., Keesom forces, Debye forces, and London dispersion forces). However, this rule of thumb is not applicable to systems with strong intermolecular forces (e.g., hydrogen bonds), where the packing coefficient can reach up to 0.70.

$$PC\colon (V_{guest}, V_{cavity}) \mapsto \frac{V_{guest}}{V_{cavity}} \in [0, 1] \tag{5.6}$$

where PC is the packing coefficient, V_{guest} is the guest vdW volume, and V_{cavity} is the host's cavity volume.

Within this context, there are no existing benchmark dataset to test the performance of cavity characterization methods. Thus, two benchmark datasets, comprising 22 well-known supramolecular cages from the Cambridge Structural Database (CSD) [123], have been selected from the supramolecular chemistry literature to evaluate the well-established cavity detection tools (see Section 3.3.2). Since topology and morphology of supramolecular cages differ from biomolecules, cavity detection parameters of the well-established tools (i.e., KVFinder suite, Fpocket, GHECOM and CAVER tools) had to be optimized for the supramolecular cages in both benchmark datasets. The benchmark datasets, Benchmark dataset 1 and Benchmark dataset 2, used in this study, which include the structural data files of each supramolecular cage and guest, are available at Zenodo: <https://doi.org/10.5281/zenodo.7702311>.

5.6.1 Benchmarking Dataset 1

The benchmark dataset 1 (Figure 5.31) comprises 13 X-ray structures of well-known supramolecular cages, with guest molecules being encapsulated in their void or void-like cavities, following the Rebek's rule of thumb. The guests molecules were removed from their cages and their vdW volumes were estimated using *volume* method from *pyKVFinder.Molecule* class (see Section 5.2.1.1). With it, the "real" cavity volume were calculated using the Rebek's rule of thumb. Then, the cavities of the supramolecular cages were detected and their volumes estimated for each cavity detection tool. As good evaluation metrics, the relative error (RE; Eq. 5.7) and the mean relative absolute error (MRAE; Eq. 5.8) were calculated between cavity volumes.

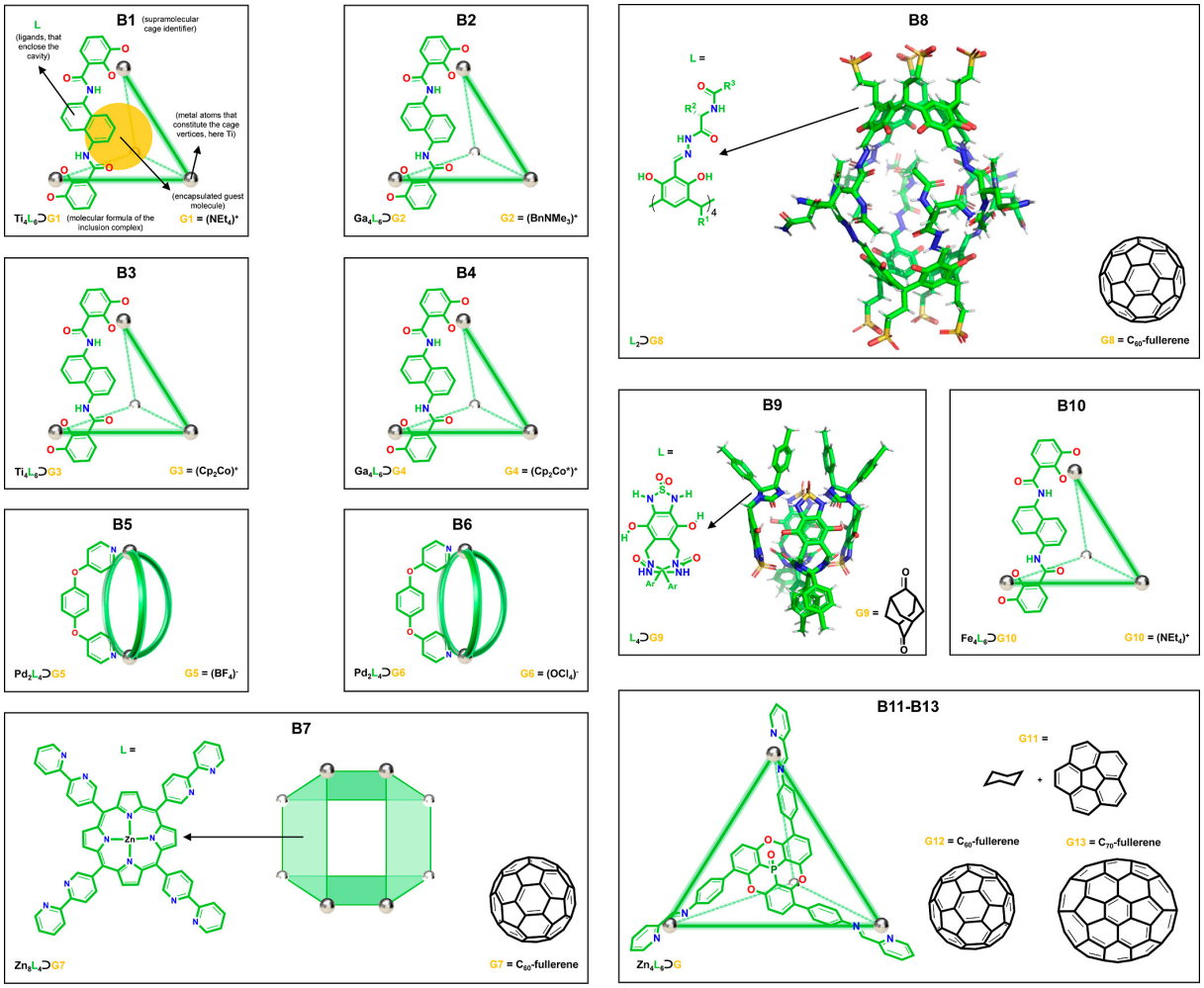
$$RE\colon (V, \hat{V}) \mapsto \frac{\hat{V} - V}{V} \in [-\infty, \infty] \tag{5.7}$$

where RE is the relative error, V is the "real" cavity volume, and $\hat{V}$ is the estimated cavity volume.

$$MRAE\colon (V, \hat{V}, i) \frac{1}{N_c} \sum_{i=1}^{N_c} \left| \frac{\hat{V}_i - V_i}{V_i} \right| \in [0, \infty] \tag{5.8}$$

where $MRAE$ is the mean relative absolute error, V_i is the "real" volume of cavity i, $\hat{V}_i$ is

the estimated volume of cavity i, and N_c is the number of cavities.



Source: Reprinted with permission from [36]. Copyright 2023 American Chemical Society.

Figure 5.31: Benchmark dataset 1. Each of these inclusion complexes features guest molecule(s) (G_i) within the cavity of a supramolecular cage (B_i). B1: $(NEt_4)^+ \subset Ti_4L_6$ (CSD ID: 718468) [124]. B2: $(BnNMe_3)^+ \subset Ga_4L_6$ (CSD ID: 718469) [124]. B3: $(Cp_2Co)^+ \subset Ti_4L_6$ (CSD ID: 718470) [124]. B4: $(Cp_2Co*)^+ \subset Ga_4L_6$ (CSD ID: 718471) [124]. B5: $(BF_4)^- \subset Pd_2L_4$ (CSD ID: 1862753) [125]. B6: $(OCl_4)^- \subset Pd_2L_4$ (CSD ID: 1862752) [125]. B7: $C_{60} \subset Zn_8L_4$ (CSD ID: 942782) [126]. B8: $C_{60} \subset Pd_2L_4$ (CSD ID: 1872778) [127]. B9: Adamantane-2,6-dione $\subset Pd_2L_4$ (CSD ID: 183906) [128]. B10: $(NEt_4)^+ \subset Pd_2L_4$ (CSD ID: 100947) [129]. B11: [Corannulene+Cyclohexane] $\subset Pd_2L_4$ (CSD ID: 2068665) [130]. B12: $C_{60} \subset Pd_2L_4$ (CSD ID: 2068666) [130]. B13: $C_{70} \subset Pd_2L_4$ (CSD ID: 2068667) [130].

The performance of each cavity detection tool was assessed by comparing the estimated cavity volumes with the "real" cavity volumes (Table 5.2). First, we analyzed the REs, that are associated with: (1) defining cavity boundaries, and (2) deviations in the PC relative to the Rebek's rule of thumb. To mitigate the first source of errors, benchmark dataset 1 exclusively contains supramolecular cages with well-defined void or void-like cavities, where delineating cavity boundaries is straighforward. Conversely, deviations from the Rebek's rule of thumb are more difficult to identify. Denser packings (i.e., $PC > 0.55$) in inclusion complexes can be reached by forces beyond weak dipole-dipole interactions (e.g., hydrogen bonds, π-π, and CH-π interactions), which favor strong host-guest association [124]. This results in a loss of entropy (i.e., restricted movement of the guest within

the cage), which is counterbalanced by the extra stabilization enthalpy. Consequently, negative RE is expected in such instances, resulting from to the overestimation of cavity volume. This trend can be clearly observed in the results of the KVFinder suite, Fpocket, and CAVER for B4–B9, B12, and B13 cages, with experimental evidence for H-bonding in B5, B6, and B9 [125, 128]. Moreover, considering cases with $(Cp_2Co*)^+$ (B4), C_{60} (B7, B8, and B12), and C_{70} (B13) as guests, additional π-π and CH-π interactions contribute strong host-guest associations. GHECOM, on the other hand, show a large negative RE for each supramolecular cage, suggesting that the calculation error becomes more significant. Conversely, MRAE values clearly show that KVFinder suite and Fpocket provide the most reliable cavity volumes.

Table 5.2: Performance of the well-established cavity detection tools in benchmark dataset 1. Estimated cavity volumes by the well-established cavity detection tools. The relative error are calculated using the "real" cavity volume as reference and can be found in paratheses. V: "real" cavity volume. V_{guest}: guest vdW volume. $V = V_{guest}/0.55$.

Cage	Guest	V_{guest}	V	KVFinder suite	Fpocket	CAVER	GHECOM
B1	$(NEt_4)^+$	150	273	283 (3.7%)	247 (-9.5%)	396 (44.8%)	175 (-35.9%)
B2	$(BnNMe_3)^+$	155	281	283 (0.8%)	279 (-0.6%)	339 (20.8%)	192 (-31.7%)
B3	$(Cp_2Co)^+$	137	248	269 (8.1%)	277 (11.6%)	335 (34.9%)	191 (-23.3%)
B4	$(Cp_2Co*)^+$	309	562	438 (-22.1%)	434 (-22.7%)	474 (-15.7%)	343 (-39.0%)
B5	$(BF_4)^-$	50	90	78 (-13.6%)	84 (-6.5%)	65 (-27.5%)	29 (-67.6%)
B6	$(OCl_4)^-$	53	96	81 (-15.6%)	82 (-14.3%)	79 (-17.4%)	64 (-33.7%)
B7	C_{60}	519	944	757 (-19.8%)	771 (-18.3%)	778 (-17.6%)	734 (-22.3%)
B8	C_{60}	512	930	731 (-21.4%)	805 (-13.5%)	682 (-26.7%)	704 (-24.3%)
B9	Adamantane-2,6-dione	141	257	155 (-39.5%)	181 (-29.5%)	173 (-32.5%)	159 (-38.0%)
B10	$(NEt_4)^+$	151	274	251 (-8.4%)	225 (-17.7%)	351 (28.1%)	161 (-41.1%)
B11	[Corannulene+Cyclohexane]	307	558	496 (-11.0%)	482 (-13.6%)	457 (-18.1%)	401 (-28.1%)
B12	C_{60}	524	954	737 (-22.7%)	627 (-34.2%)	742 (-22.2%)	606 (-36.5%)
B13	C_{70}	618	1123	872 (-22.3%)	811 (-27.7%)	1031 (-8.1%)	752 (-33.0%)
MRAE				**16.1%**	**16.9%**	**24.2%**	**35.0%**

Furthermore, the cavity volumes estimated with KVFinder suite and Fpocket closely approximate the "real" cavity volume, presenting a small spread in comparison to other cavity detection tools (Figure 5.32A). Overall, the general trend supports our strategy of using Rebek's rule of thumb as the optimal estimate for the "real" cavity volume within these supramolecular cages.

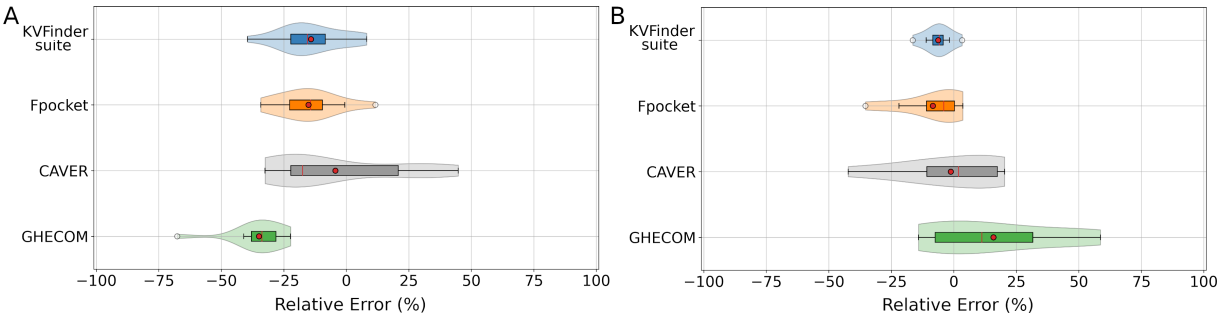
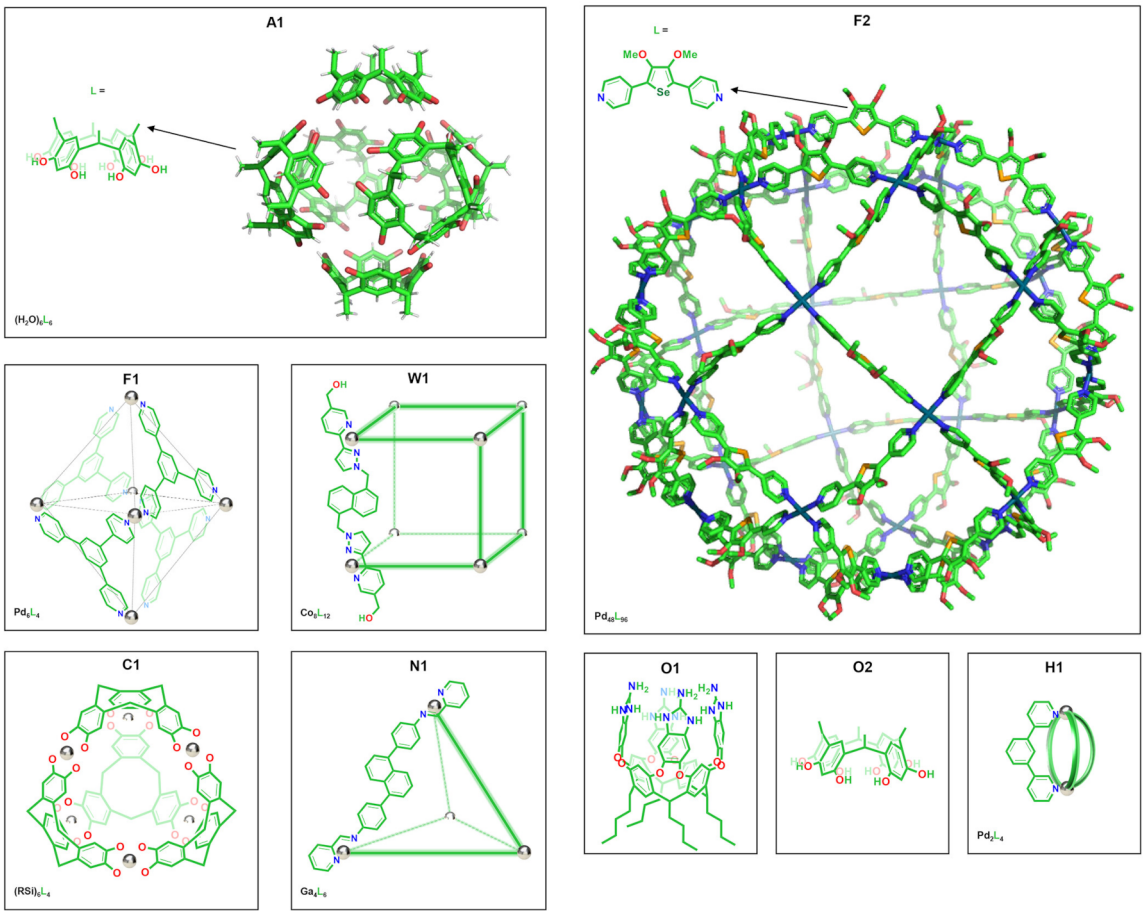


Figure 5.32: Boxplot of the relative error of the cavity volume in the benchmark datasets. **(A)** Benchmark dataset 1 with "real" cavity volume as reference. **(B)** Benchmark dataset 2 with the average volume of detected cavities as reference. In benchmark dataset 2, the cavity of O2 is excluded due to high uncertainty in detecting its cavity by some methods. Each cavity detection tool is presented with a violin plot depicting the probability density function of the relative error. The red line denotes the median, and the point indicates the mean relative errors.

5.6.2 Benchmarking Dataset 2

The benchmark dataset 2 (Figure 5.33) comprises 9 X-ray structures of well-known supramolecular cages, without the guest molecules, but with different topological and morphological features (i.e., cavity size, opening size, and shape).



Source: Reprinted with permission from [36]. Copyright 2023 American Chemical Society.

Figure 5.33: Benchmark dataset 2. A1: $(H_2O)_6L_6$ (CSD ID: 1207879) [131]. C1: $(RSi)_6L_4$ (CSD ID: 1892128) [132]. F1: Pd_6L_4 (CSD ID: 293777) [133]. F2: $Pd_{48}L_{96}$ (CSD ID: 1831430) [134]. H1: Pd_2L_4 (CSD ID: 768969) [135]. N1: Ga_4L_6 (CSD ID: 1541839) [136]. O1: Tetra(chloropropyl) 2-aminobenzimidazole cavitand (CSD ID: 2074472) [137]. O2: Resorcin[4]arene (CSD ID: 1207879) [131]. W1: Co_8L_{12} (CSD ID: 1416694) [138].

This dataset aims to evaluate the capabilities and limitations of the well-established cavity detection tools in detecting different types of cavities (Figure 3.1). However, since these supramolecular cages lack internal guests, the estimation of the "real" cavity volume via Rebek's rule of thumb is not feasible. Consequently, the average volume of detected cavities serves as the point of comparison for calculated volumes (Table 5.3), establishing an agreement between methods by mitigating discrepancies in volume data. According to Table 5.3 and Figure 5.32, KVFinder suite presented a small spread compared to other cavity detection tools, with the lowest MRAE value (6.3%). Fpocket and CAVER had a larger RE spread compared to KVFinder suite, while GHECOM tends to overestimate the cavity volume. This result is consistent with the previous benchmark dataset, where KVFinder suite provided the most reliable cavity volumes.

Table 5.3: Performance of the well-established cavity detection tools in benchmark dataset 2. Estimated cavity volumes by the well-established cavity detection tools. The relative error are calculated using the average cavity volume as reference and can be found in paratheses. $\bar{V}$: average cavity volume. (*) indicates that the cavity was not detected by the respective cavity detection tool.

Cavity type	Cage	$\bar{V}$	KVFinder suite	Fpocket	CAVER	GHECOM
Void	A1	1481	1399 (-5.6%)	1387 (-6.4%)	1778 (20.0%)	1362 (-8.1%)
Invagination	C1	603	558 (-7.5%)	617 (2.3%)	569 (-5.6%)	669 (10.9%)
Invagination	F1	520	463 (-11.0%)	483 (-7.2%)	481 (-7.6%)	655 (25.9%)
Invagination	F2	39723	37728 (-5.0%)	39077 (-1.6%)	22941 (-42.2%)	59147 (48.9%)
Invagination	H1	270	226 (-16.4%)	211 (-22.0%)	215 (-20.3%)	429 (58.6%)
Invagination	N1	496	488 (-1.8%)	495 (-0.3%)	543 (9.3%)	461 (-7.2%)
Invagination	O1	170	160 (-6.0%)	177 (3.6%)	199 (16.5%)	146 (-14.1%)
Cleft/Groove	O2	36	36 (0.0%)	0* (-100.0%)	0* (-100.0%)	0* (-100.0%)
Tunnel	W1	838	866 (3.3%)	542 (-35.3%)	1008 (20.3%)	936 (11.7%)
MRAE			**6.3%**	**19.9%**	**26.9%**	**31.7%**

5.6.3 Discussion

Overall, each cavity detection tool has distinct capabilities and limitations, influenced by the cavity detection method, its implementation, and the software interfaces used (Table 5.4). Broadly, grid-and-probe-based and tessellation-based methods had a good performance in detecting cavities and estimating cavity volume in our benchmarking protocol. Additionally, KVFinder suite can detect any type of cavities in supramolecular cages, including clefts and grooves (i.e., O2 shallow groove), a capability lacking in other cavity detection tools. Based on our assessment, KVFinder suite (i.e., parKVFinder, pyKVFinder, and KVFinder-web) outperformed other cavity detection tools in benchmarking, consistently providing volumes closest to the "real" and average volumes, with a small devitation compared to other tools (Figure 5.32).

Table 5.4: Qualitative assessment of well-established cavity detection tools.

Cavity detection tools	Software interface				Cavity types (Figure 3.1)				Reference
	GUI	CLI	API	Web	Voids	Invaginations	Channels/Tunnels	Clefts/Grooves	
KVFinder suite	×	×	×	×	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆	5,9,10
Fpocket	×	×	×	×	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆	NA	48,108
CAVER tools	×	×		×	⋆⋆	⋆⋆	⋆⋆	NA	49,60,107
GHECOM	×	×		×	⋆⋆⋆	⋆⋆	⋆⋆⋆	NA	52

NA: Not applicable.

All methods offer user-friendly installation, configuration, and execution, with accessibility through GUI, CLI, application programming interfaces (APIs), and web applications. GUIs and web applications are simpler for less-experienced users, aiding in intuitive analysis pipelines. However, they lack efficiency in handling large datasets, making CLIs and APIs more suitable for high-throughput analysis, MD simulations, ML, DL, virtual screening, and pipeline automation. While CLIs and APIs are efficient and integrable, CLIs may limit customization in pipeline automation. In contrast, APIs (e.g., pyKVFinder, and Fpocket) offer versatility, allowing users to build complex applications or integrate them with third-party scientific packages (e.g., NumPy [88], SciPy [92], scikit-learn [91], and matplotlib [100]). Notably, pyKVFinder provides accessible core data structures for user manipulation, enabling the development of new characterizations and applications built around them (see Section 5.2.1).

5.7 Perspectives

The structural biology community is expected to continue using the KVFinder suite for characterizing biomolecules and their binding sites across morphological, topological, and physicochemical features. The relevance of KVFinder suite is further emphasized by its application in a range of structural biology community studies, including both characterization and comparative investigations, as evidenced by references such as [105, 139–147].

Beyond conventional applications, there is a prospect for the KVFinder suite to find novel utility in data science applications within the structural biology community. The KVFinder suite's versatility is exemplified by its role in comparative studies presented in this thesis, such as investigations into Mayaro and other alphaviruses (Section 5.1.1.2) [8], SARS-CoV-2 and homologous proteins (Section 5.2.2.1) [9], and morphological comparisons of the catalytic site of HIV-1 protease structures (Section 5.3.5.2) [10]. Furthermore, its application in molecular dynamics simulations analysis, as demonstrated in studies on the dynamics of HIV-1 protease (Section 5.1.1.1) [8] and the ADRP domain of SARS-CoV-2 (Section 5.2.2.2) [9], highlights its potential in unraveling dynamic biomolecular processes. Yet, extending its original scope, the KVFinder suite also found users in the supramolecular chemistry community, as demonstrated by characterization of supramolecular cages (Section 5.2.1) and the benchmarking study (Section 5.6) [36].

As part of KVFinder suite, pyKVFinder has been utilized in exploring cavity cross-sectional area [105], molecular volume estimation (Section 5.2.1.1) [10], and cavity opening characterization (Section 5.2.1.2), serving as a guide for users to develop novel characterizations. Additionally, integrating pyKVFinder into DL, using cavity characterizations as input features, aligns with recent applications. Instances from the literature, such as Sfcnn [148], which is a scoring function model based on 3D convolutional neural networks for protein-ligand binding affinity prediction, LigVoxel [149], a deep convolutional neural network trained on experimental protein-ligand complexes for predicting spatial maps of ligand properties, and DeepDrug3D [150], a convolutional neural network capable of classifying binding sites in proteins through learning specific molecular interactions like hydrogen bonds, aromatic contacts, and hydrophobic contacts, provide inspiration for potential applications of pyKVFinder in deep learning models.

Looking ahead, future work involves algorithmic enhancements and novel characterizations. Upgrades to the cavity detection algorithm, aiming to mitigate grid-orientation sensitivity, the algorithm of cavity volume estimation, aiming to provide even more reliable cavity volumes, and the voxel-clustering algorithm, aiming to optimize the most time-consuming algorithm in the cavity detection workflow, enabling future GPU parallelization. In the context of novel characterizations, the KVFinder suite aims to integrate ligandability and druggability characterizations to the detected cavities, potentially serving as ranking scores for virtual screening applications. Ligandability, reflecting the feasibility of designing small ligands with high binding affinities, and druggability, assessing the suitability of a cavity for binding drug-like molecules, open avenues for exploring higher-level properties of ligands (ADME/T) [151]. This involves applying datasets, such as NRDLD from [152], to train machine learning models for distinguishing druggable and

non-druggable sites. Additionally, a protocol for identifying allosteric binding sites within the KVFinder suite would be desirable, involving the application of normal mode analysis, as demonstrated by AllogSigma2 [153] and ESSA [154]. Notably, the introduction of cavity skeletonization (morphological characterization) as a metric for defining cavity accessibility further enhances the KVFinder suite's characterization capabilities.

In essence, the KVFinder suite, having demonstrated its versatility and efficacy, stands poised for a future marked by continued applications across diverse research domains and ongoing advancements to meet evolving computational and analytical demands within the structural biology community.

Chapter 6: Conclusion

After a thorough evaluation of the computational tool requirements within the structural biology community, we successfully developed a computational platform known as KVFinder suite. This platform was designed for studies of biomolecular systems, providing comprehensive tools for coding and characterizing biomolecules and their binding sites. Comprising five computational tools, namely KVFinder-web, parKVFinder, pyKVFinder, SERD, and KVFinderMD, the KVFinder suite encompasses different demands and scopes of the structural biology community. Notably, this work has resulted in the publication of five articles, including four authored publications [5, 9, 10, 36] and a collaborative effort [83].

parKVFinder, an open-source tool, was developed for the detection and characterization of biomolecular cavities. It includes a graphical plugin integrated into the PyMOL molecular viewer, which offers an intuitive interface for exploring customizable parameters and visualizing the detected cavities and their characteristics. Although parKVFinder has limitations in automated applications and systematic comparisons of binding sites, it plays an important role in optimizing detection and characterization parameters through the PyMOL graphical plugin (PyMOL2 parKVFinder Tools), due to its visual features. These optimized parameters can be subsequently adopted in automated studies and systematic comparisons of binding sites.

KVFinder-web, an open-source web application of parKVFinder, is a user-friendly tool for detecting and characterizing cavities in any type of biomolecular structure. KVFinder-web aims to expand the use of parKVFinder in the scientific community, in addition to democratizing access and removing barriers for users who do not have technical knowledge to install and configure a computational tool, who have limited computational resources, or who wish to perform a simple and fast analysis. The web application is available at <https://kvfinder-web.cnpem.br>. KVFinder-web is a valuable tool for the structural biology community, as it provides a simple and intuitive interface for detecting and characterizing cavities in biomolecular structures.

pyKVFinder is an open-source Python package for detecting and characterizing cavities in biomolecular structures in automated protocols and data science applications. In addition to having the same functionalities as KVFinder-web and parKVFinder, pyKVFinder provides accessible and flexible data structures in the Python ecosystem, such as ndarrays and dictionaries. This allows users to develop new cavity characterizations and analysis protocols based on these data structures, facilitating efficient and effective exploration of biomolecular cavities and driving the discovery of new therapeutic targets and the development of more effective drugs. In collaboration with Dr. György

Szalóki, we expanded the cavity detection and characterization methodology to a new class of molecules called supramolecular cages. In addition, we developed new characterizations applicable to both cages and biomolecules, including molecular surface modeling in 3D grids, molecular volume estimation, and characterization of cavity openings. These implementations can serve as a guide for users to develop new characterizations.

Throughout the project, the computational performance and cavity detection capabilities of KVFinder suite tools (KVFinder-web, parKVFinder, and pyKVFinder) were continuously evaluated, showcasing their effectiveness and computational performance. Again, in collaboration Dr. György Szalóki, we benchmarked well-established cavity detection tools (e.g., KVFinder suite, Fpocket, GHECOM, and CAVER tools) in a dataset of well-defined artificial supramolecular cages, highlighting KVFinder suite as a reliable tool for detecting and characterizing cavities in biomolecular structures. Furthermore, our strategy of using the Rebek's rule of thumb as the optimal estimate for the "real" cavity volume within these supramolecular cages offers a solution to the current problem of missing gold standard reference data in the field of cavity detection.

Conversly, these cavity detection tools (e.g., parKVFinder, pyKVFinder and KVFinder-web) depend on 3D grid modeling and description. To cover a wider range of biomolecule encodings and binding sites, we developed the SERD tool. This tool expanded the possibilities of encoding in the KVFinder suite platform, including the topological representation of interface residues and the graph representation. Such flexibility allows the application of different encodings in studies of biomolecular systems.

Finally, KVFinderMD was developed as a Python package that allows the exploration of binding site dynamics in biomolecular structures of therapeutic interest. This tool exemplifies the ability of pyKVFinder to provide automated protocols for systematic analyses of binding sites. The analysis of cavity similarity in the HIV-1 protease throughout the MD simulation allowed us to track the evolution of these cavities and compare their characteristics over time. Using hierarchical clustering algorithms and different affinity metrics in different codings (i.e., 3D grid, residue-level representation, and graph-based representation), we identified groups of similar cavities. Furthermore, we were able to reduce the size of the data required for similarity analysis and simplify the comparison between cavities in MD simulations, demonstrating the practical applicability of the KVFinder suite.

In summary, the KVFinder suite simplifies the study of biomolecular systems of therapeutic interest, including the characterization of biomolecules and their binding sites, even for less experienced users. This has a direct impact on the search and rational design of drugs, as well as on the understanding of biomolecule structures.

Bibliography

1 SOTRIFFER, C.; KLEBE, G. Identification and mapping of small-molecule binding sites in proteins: computational tools for structure-based drug design. *Il Farmaco*, v. 57, n. 3, p. 243–251, 2002. ISSN 0014-827X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0014827X02012119>.

2 HENRICH, S. *et al.* Computational approaches to identifying and characterizing protein binding sites for ligand design. *Journal of Molecular Recognition*, v. 23, n. 2, p. 209–219, 2010. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jmr.984>.

3 KOZLÍKOVÁ, B. *et al.* Visualization of biomolecular structures: State of the art revisited. *Computer Graphics Forum*, Wiley, v. 36, n. 8, p. 178–204, nov. 2016. Disponível em: <https://doi.org/10.1111/cgf.13072>.

4 OLIVEIRA, S. H. *et al.* KVFinder: steered identification of protein cavities as a PyMOL plugin. *BMC Bioinformatics*, Springer Science and Business Media LLC, v. 15, n. 1, jun. 2014. Disponível em: <https://doi.org/10.1186/1471-2105-15-197>.

5 GUERRA, J. V. da S. *et al.* pyKVFinder: an efficient and integrable python package for biomolecular cavity detection and characterization in data science. *BMC Bioinformatics*, Springer Science and Business Media LLC, v. 22, n. 1, dez. 2021. Disponível em: <https://doi.org/10.1186/s12859-021-04519-4>.

6 SIMÕES, T. *et al.* Geometric detection algorithms for cavities on protein surfaces in molecular graphics: A survey. *Computer Graphics Forum*, Wiley, v. 36, n. 8, p. 643–683, jun. 2017. Disponível em: <https://doi.org/10.1111/cgf.13158>.

7 TUNYASUVUNAKOOL, K. *et al.* Highly accurate protein structure prediction for the human proteome. *Nature*, Springer Science and Business Media LLC, v. 596, n. 7873, p. 590–596, jul. 2021. Disponível em: <https://doi.org/10.1038/s41586-021-03828-1>.

8 GUERRA, J. V. *Prospecção e caracterização de cavidades supramoleculares*. Dissertação (Mestrado) — Universidade Estadual de Campinas, Instituto de Biologia, Campinas, SP, jun 2019. Disponível em: <https://hdl.handle.net/20.500.12733/1639705>.

9 GUERRA, J. V. da S. *et al.* ParKVFinder: A thread-level parallel approach in biomolecular cavity detection. *SoftwareX*, Elsevier BV, v. 12, p. 100606, jul. 2020. Disponível em: <https://doi.org/10.1016/j.softx.2020.100606>.

10 GUERRA, J. V. S. *et al.* KVFinder-web: a web-based application for detecting and characterizing biomolecular cavities. *Nucleic Acids Research*, Oxford University Press (OUP), maio 2023. Disponível em: <https://doi.org/10.1093/nar/gkad324>.

11 HOLYOAK, T. Molecular recognition: Lock-and-key, induced fit, and conformational selection. In: *Encyclopedia of Biophysics*. Springer Berlin Heidelberg, 2013. p. 1584–1588. Disponível em: <https://doi.org/10.1007/978-3-642-16712-6_468>.

12 LAY, S. *Convex Sets and Their Applications*. Dover Publications, Incorporated, 2013. (Dover Books on Mathematics Series). ISBN 9780486788265. Disponível em: <https://books.google.com.br/books?id=YU50swEACAAJ>.

13 LIANG, J.; WOODWARD, C.; EDELSBRUNNER, H. Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Protein Science*, Wiley, v. 7, n. 9, p. 1884–1897, set. 1998. Disponível em: <https://doi.org/10.1002/pro.5560070905>.

14 HUBBARD, S. J.; ARGOS, P. Cavities and packing at protein interfaces. *Protein Science*, Wiley, v. 3, n. 12, p. 2194–2206, dez. 1994. Disponível em: <https://doi.org/10.1002/pro.5560031205>.

15 BOHACEK, R. S.; MCMARTIN, C. Modern computational chemistry and drug discovery: structure generating programs. *Current Opinion in Chemical Biology*, Elsevier BV, v. 1, n. 2, p. 157–161, ago. 1997. Disponível em: <https://doi.org/10.1016/s1367-5931(97)80004-x>.

16 MURA, C.; DRAIZEN, E. J.; BOURNE, P. E. Structural biology meets data science: does anything change? *Current Opinion in Structural Biology*, Elsevier BV, v. 52, p. 95–102, out. 2018. Disponível em: <https://doi.org/10.1016/j.sbi.2018.09.003>.

17 BURLEY, S. K. *et al.* RCSB protein data bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. *Nucleic Acids Research*, Oxford University Press (OUP), v. 47, n. D1, p. D464–D474, out. 2018. Disponível em: <https://doi.org/10.1093/nar/gky1004>.

18 SCOTT, D. E. *et al.* Using a fragment-based approach to target protein-protein interactions. *ChemBioChem*, Wiley, v. 14, n. 3, p. 332–342, jan. 2013. Disponível em: <https://doi.org/10.1002/cbic.201200521>.

19 KRONE, M. *et al.* Visual analysis of biomolecular cavities: State of the art. *Computer Graphics Forum*, v. 35, n. 3, p. 527–551, 2016. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12928>.

20 LASKOWSKI, R. A. *et al.* Protein clefts in molecular recognition and function. *Protein science : a publication of the Protein Society*, v. 5, n. 12, p. 2438–52, 1996. ISSN 0961-8368. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/8976552/>.

21 STANK, A. *et al.* Protein binding pocket dynamics. *Accounts of Chemical Research*, American Chemical Society (ACS), v. 49, n. 5, p. 809–815, abr. 2016. Disponível em: <https://doi.org/10.1021/acs.accounts.5b00516>.

22 JUNCKER, A. S. *et al.* Sequence-based feature prediction and annotation of proteins. *Genome Biology*, Springer Science and Business Media LLC, v. 10, n. 2, p. 206, 2009. Disponível em: <https://doi.org/10.1186/gb-2009-10-2-206>.

23 DODSON, G. Catalytic triads and their relatives. *Trends in Biochemical Sciences*, Elsevier BV, v. 23, n. 9, p. 347–352, set. 1998. Disponível em: <https://doi.org/10.1016/s0968-0004(98)01254-7>.

24 THORNTON, J. M. *et al. Nature Structural Biology*, Springer Science and Business Media LLC, v. 7, p. 991–994, nov. 2000. Disponível em: <https://doi.org/10.1038/80784>.

25 CARLSON, H. A. *et al.* Differences between high- and low-affinity complexes of enzymes and nonenzymes. *Journal of Medicinal Chemistry*, American Chemical Society (ACS), v. 51, n. 20, p. 6432–6441, out. 2008. Disponível em: <https://doi.org/10.1021/jm8006504>.

26 MANNHOLD, R. *et al.* Calculation of molecular lipophilicity: State-of-the-art and comparison of LogP methods on more than 96, 000 compounds. *Journal of Pharmaceutical Sciences*, Elsevier BV, v. 98, n. 3, p. 861–893, mar. 2009. Disponível em: <https://doi.org/10.1002/jps.21494>.

27 HEIDEN, W.; MOECKEL, G.; BRICKMANN, J. A new approach to analysis and display of local lipophilicity/hydrophilicity mapped on molecular surfaces. *Journal of Computer-Aided Molecular Design*, Springer Science and Business Media LLC, v. 7, n. 5, p. 503–514, out. 1993. Disponível em: <https://doi.org/10.1007/bf00124359>.

28 EISENBERG, D.; WEISS, R. M.; TERWILLIGER, T. C. The hydrophobic moment detects periodicity in protein hydrophobicity. *Proceedings of the National Academy of Sciences*, Proceedings of the National Academy of Sciences, v. 81, n. 1, p. 140–144, jan. 1984. Disponível em: <https://doi.org/10.1073/pnas.81.1.140>.

29 HESSA, T. *et al.* Recognition of transmembrane helices by the endoplasmic reticulum translocon. *Nature*, Springer Science and Business Media LLC, v. 433, n. 7024, p. 377–381, jan. 2005. Disponível em: <https://doi.org/10.1038/nature03216>.

30 KYTE, J.; DOOLITTLE, R. F. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, Elsevier BV, v. 157, n. 1, p. 105–132, maio 1982. Disponível em: <https://doi.org/10.1016/0022-2836(82)90515-0>.

31 MOON, C. P.; FLEMING, K. G. Side-chain hydrophobicity scale derived from transmembrane protein folding into lipid bilayers. *Proceedings of the National Academy of Sciences*, Proceedings of the National Academy of Sciences, v. 108, n. 25, p. 10174–10177, maio 2011. Disponível em: <https://doi.org/10.1073/pnas.1103979108>.

32 RADZICKA, A.; WOLFENDEN, R. Comparing the polarities of the amino acids: side-chain distribution coefficients between the vapor phase, cyclohexane, 1-octanol, and neutral aqueous solution. *Biochemistry*, American Chemical Society (ACS), v. 27, n. 5, p. 1664–1670, mar. 1988. Disponível em: <https://doi.org/10.1021/bi00405a042>.

33 WIMLEY, W. C.; WHITE, S. H. Experimentally determined hydrophobicity scale for proteins at membrane interfaces. *Nature Structural &amp Molecular Biology*, Springer Science and Business Media LLC, v. 3, n. 10, p. 842–848, out. 1996. Disponível em: <https://doi.org/10.1038/nsb1096-842>.

34 ZHAO, G.; LONDON, E. An amino acid "transmembrane tendency" scale that approaches the theoretical limit to accuracy for prediction of transmembrane helices: Relationship to biological hydrophobicity. *Protein Science*, Wiley, v. 15, n. 8, p. 1987–2001, ago. 2006. Disponível em: <https://doi.org/10.1110/ps.062286306>.

35 HONIG, B.; NICHOLLS, A. Classical electrostatics in biology and chemistry. *Science*, American Association for the Advancement of Science (AAAS), v. 268, n. 5214, p. 1144–1149, maio 1995. Disponível em: <https://doi.org/10.1126/science.7761829>.

36 GUERRA, J. V. S. *et al.* Cavity characterization in supramolecular cages. *Journal of Chemical Information and Modeling*, American Chemical Society (ACS), maio 2023. Disponível em: <https://doi.org/10.1021/acs.jcim.3c00328>.

37 HO, C. M.; MARSHALL, G. R. Cavity search: an algorithm for the isolation and display of cavity-like binding regions. *Journal of computer-aided molecular design*, Springer, v. 4, p. 337–354, 1990.

38 LEVITT, D. G.; BANASZAK, L. J. Pocket: a computer graphies method for identifying and displaying protein cavities and their surrounding amino acids. *Journal of molecular graphics*, Elsevier, v. 10, n. 4, p. 229–234, 1992.

39 HENDLICH, M.; RIPPMANN, F.; BARNICKEL, G. Ligsite: automatic and efficient detection of potential small molecule-binding sites in proteins. *Journal of Molecular Graphics and Modelling*, Elsevier, v. 15, n. 6, p. 359–363, 1997.

40 WAGNER, J. R. *et al.* Povme 3.0: Software for mapping binding pocket flexibility. *Journal of Chemical Theory and Computation*, v. 13, n. 9, p. 4584–4592, 2017. PMID: 28800393. Disponível em: <https://doi.org/10.1021/acs.jctc.7b00500>.

41 KUNTZ, I. D. *et al.* A geometric approach to macromolecule-ligand interactions. *Journal of molecular biology*, Elsevier, v. 161, n. 2, p. 269–288, 1982.

42 BRADY, G. P.; STOUTEN, P. F. Fast prediction and visualization of protein binding pockets with pass. *Journal of computer-aided molecular design*, Springer, v. 14, p. 383–401, 2000.

43 KAWABATA, T.; GO, N. Detection of pockets on protein surfaces using small and large probe spheres to find putative ligand binding sites. *Proteins: Structure, Function, and Bioinformatics*, v. 68, n. 2, p. 516–529, 2007. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.21283>.

44 MIKLITZ, M.; JELFS, K. E. pywindow: Automated structural analysis of molecular pores. *Journal of Chemical Information and Modeling*, v. 58, n. 12, p. 2387–2391, 2018. PMID: 30199639. Disponível em: <https://doi.org/10.1021/acs.jcim.8b00490>.

45 Del Carpio, C. A.; TAKAHASHI, Y.; SASAKI, S. ichi. A new approach to the automatic identification of candidates for ligand receptor sites in proteins: (i) search for pocket regions. *Journal of Molecular Graphics*, v. 11, n. 1, p. 23–29, 1993. ISSN 0263-7855. Disponível em: <https://www.sciencedirect.com/science/article/pii/0263785593850039>.

46 ZHU, H.; PISABARRO, M. T. MSPocket: an orientation-independent algorithm for the detection of ligand binding pockets. *Bioinformatics*, v. 27, n. 3, p. 351–358, 12 2010. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btq672>.

47 PETERS, K. P.; FAUCK, J.; FRÖMMEL, C. The automatic search for ligand binding sites in proteins of known three-dimensional structure using only geometric criteria. *Journal of molecular biology*, Elsevier, v. 256, n. 1, p. 201–213, 1996.

48 GUILLOUX, V. L.; SCHMIDTKE, P.; TUFFERY, P. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, Springer Science and Business Media LLC, v. 10, n. 1, jun. 2009. Disponível em: <https://doi.org/10.1186/1471-2105-10-168>.

49 CHOVANCOVA, E. *et al.* Caver 3.0: A tool for the analysis of transport pathways in dynamic protein structures. *PLOS Computational Biology*, Public Library of Science, v. 8, n. 10, p. 1–12, 10 2012. Disponível em: <https://doi.org/10.1371/journal.pcbi.1002708>.

50 KLEYWEGT, G. J.; JONES, T. A. Detection, delineation, measurement and display of cavities in macromolecular structures. *Acta Crystallographica Section D: Biological Crystallography*, International Union of Crystallography, v. 50, n. 2, p. 178–185, 1994.

51 YU, J. *et al.* Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere. *Bioinformatics*, Oxford University Press, v. 26, n. 1, p. 46–52, 2010.

52 KAWABATA, T. Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins: Structure, Function, and Bioinformatics*, Wiley, v. 78, n. 5, p. 1195–1211, out. 2009. Disponível em: <https://doi.org/10.1002/prot.22639>.

53 VOSS, N. R.; GERSTEIN, M. 3v: cavity, channel and cleft volume calculator and extractor. *Nucleic Acids Research*, Oxford University Press (OUP), v. 38, n. Web Server, p. W555–W562, maio 2010. Disponível em: <https://doi.org/10.1093/nar/gkq395>.

54 VOORINTHOLT, R. *et al.* A very fast program for visualizing protein surfaces, channels and cavities. *Journal of Molecular Graphics*, Elsevier, v. 7, n. 4, p. 243–245, 1989.

55 PETŘEK, M. *et al.* CAVER: a new tool to explore routes from protein clefts, pockets and cavities. *BMC Bioinformatics*, Springer Science and Business Media LLC, v. 7, n. 1, jun. 2006. Disponível em: <https://doi.org/10.1186/1471-2105-7-316>.

56 SCHRöDINGER, L.; DELANO, W. *PyMOL*. Disponível em: <http://www.pymol.org/pymol>.

57 MATHERON, G. *Random Sets and Integral Geometry*. Wiley, 1974. (Probability and Statistics Series). ISBN 9780471576211. Disponível em: <https://books.google.com.br/books?id=bgzvAAAAMAAJ>.

58 SERRA, J.; SERRA, J. *Image Analysis and Mathematical Morphology*. Academic Press, 1982. (Image Analysis and Mathematical Morphology). ISBN 9780126372410. Disponível em: <https://books.google.com.br/books?id=BpdTAAAAYAAJ>.

59 BARBER, C. B.; DOBKIN, D. P.; HUHDANPAA, H. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, Association for Computing Machinery (ACM), v. 22, n. 4, p. 469–483, dez. 1996. Disponível em: <https://doi.org/10.1145/235815.235821>.

60 JURCIK, A. *et al.* CAVER analyst 2.0: analysis and visualization of channels and tunnels in protein structures and molecular dynamics trajectories. *Bioinformatics*, Oxford University Press (OUP), v. 34, n. 20, p. 3586–3588, maio 2018. Disponível em: <https://doi.org/10.1093/bioinformatics/bty386>.

61 FOSTER, I.; FOSTER, J. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*. Addison-Wesley, 1995. (Literature and Philosophy). ISBN 9780201575941. Disponível em: <https://books.google.com.br/books?id=r5JsQgAACAAJ>.

62 GRAMA, A. *Introduction to Parallel Computing*. Addison-Wesley, 2003. (Pearson Education). ISBN 9780201648652. Disponível em: <https://books.google.com.br/books?id=B3jR2EhdZaMC>.

63 MATLOFF, N. *Introduction to Parallel Computing*. University of California, 2012. Disponível em: <https://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>.

64 AMDAHL, G. M. Validity of the single processor approach to achieving large scale computing capabilities. In: *Proceedings of the April 18-20, 1967, spring joint computer conference on - AFIPS '67 (Spring)*. ACM Press, 1967. Disponível em: <https://doi.org/10.1145/1465482.1465560>.

65 GUSTAFSON, J. L. Reevaluating amdahl's law. *Communications of the ACM*, Association for Computing Machinery (ACM), v. 31, n. 5, p. 532–533, maio 1988. Disponível em: <https://doi.org/10.1145/42411.42415>.

66 LIN, Z. *et al.* Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, Cold Spring Harbor Laboratory, 2022.

67 JUMPER, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature*, Springer Science and Business Media LLC, v. 596, n. 7873, p. 583–589, jul. 2021. Disponível em: <https://doi.org/10.1038/s41586-021-03819-2>.

68 BAEK, M. *et al.* Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, American Association for the Advancement of Science (AAAS), v. 373, n. 6557, p. 871–876, ago. 2021. Disponível em: <https://doi.org/10.1126/science.abj8754>.

69 SPOEL, D. V. D. *et al.* Gromacs: Fast, flexible, and free. *Journal of Computational Chemistry*, v. 26, n. 16, p. 1701–1718, 2005. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20291>.

70 CASE, D. A. *et al.* The amber biomolecular simulation programs. *Journal of Computational Chemistry*, v. 26, n. 16, p. 1668–1688, 2005. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20290>.

71 KENZAKI, H. *et al.* Cafemol: A coarse-grained biomolecular simulator for simulating proteins at work. *Journal of Chemical Theory and Computation*, v. 7, n. 6, p. 1979–1989, 2011. PMID: 26596457. Disponível em: <https://doi.org/10.1021/ct2001045>.

72 FOULDS, L. R. *Graph Theory Applications*. 1. ed. New York, NY: Springer, 1995. (Universitext).

73 MAJEED, A.; RAUF, I. Graph theory: A comprehensive survey about graph theory applications in computer science and social networks. *Inventions*, v. 5, n. 1, 2020. ISSN 2411-5134. Disponível em: <https://www.mdpi.com/2411-5134/5/1/10>.

74 VISHVESHWARA, S.; BRINDA, K. V.; KANNAN, N. Protein structure: Insights from graph theory. *Journal of Theoretical and Computational Chemistry*, World Scientific Pub Co Pte Lt, v. 01, n. 01, p. 187–211, jul. 2002. Disponível em: <https://doi.org/10.1142/s0219633602000117>.

75 PAOLA, L. D.; GIULIANI, A. Protein contact network topology: a natural language for allostery. *Current Opinion in Structural Biology*, Elsevier BV, v. 31, p. 43–48, abr. 2015. Disponível em: <https://doi.org/10.1016/j.sbi.2015.03.001>.

76 HEAL, J. W. *et al.* Applying graph theory to protein structures: an atlas of coiled coils. *Bioinformatics*, Oxford University Press (OUP), v. 34, n. 19, p. 3316–3323, maio 2018. Disponível em: <https://doi.org/10.1093/bioinformatics/bty347>.

77 KANTELIS, K. F. *et al.* Graph theory-based simulation tools for protein structure networks. *Simulation Modelling Practice and Theory*, Elsevier BV, v. 121, p. 102640, dez. 2022. Disponível em: <https://doi.org/10.1016/j.simpat.2022.102640>.

78 BONDY, J. A.; MURTY, U. S. R. *Graph Theory with Applications*. New York: Elsevier, 1976.

79 BLACK, P. E. *DADS: The On-Line Dictionary of Algorithms and Data Structures*. [S.l.], set. 2020. Disponível em: <https://doi.org/10.6028/nist.ir.8318>.

80 MASON, O.; VERWOERD, M. Graph theory and networks in biology. *IET Systems Biology*, Institution of Engineering and Technology (IET), v. 1, n. 2, p. 89–119, mar. 2007. Disponível em: <https://doi.org/10.1049/iet-syb:20060038>.

81 JANIN, J. *et al.* Capri: a critical assessment of predicted interactions. *Proteins: Structure, Function, and Bioinformatics*, v. 52, p. 2–9, 2003.

82 GROUP, E.-P. *CAPRI: Critical Assessment of PRediction of Interactions - Round 28*. 2013. <https://www.ebi.ac.uk/msd-srv/capri/round28/round28.html>. Accessed: 30 October 2023.

83 RIBEIRO-FILHO, H. V. *et al.* Cryo-EM structure of the mature and infective mayaro virus at 4.4 {aa resolution reveals features of arthritogenic alphaviruses. *Nature Communications*, Springer Science and Business Media LLC, v. 12, n. 1, maio 2021. Disponível em: <https://doi.org/10.1038/s41467-021-23400-9>.

84 LAM, P. Y. S. *et al.* Rational design of potent, bioavailable, nonpeptide cyclic ureas as HIV protease inhibitors. *Science*, American Association for the Advancement of Science (AAAS), v. 263, n. 5145, p. 380–384, jan. 1994. Disponível em: <https://doi.org/10.1126/science.8278812>.

85 SOARES, R. O. *et al.* Unraveling HIV protease flaps dynamics by constant pH molecular dynamics simulations. *Journal of Structural Biology*, Elsevier BV, v. 195, n. 2, p. 216–226, ago. 2016. Disponível em: <https://doi.org/10.1016/j.jsb.2016.06.006>.

86 SIMÕES, T. M. C.; GOMES, A. J. P. CavVis—a field-of-view geometric algorithm for protein cavity detection. *Journal of Chemical Information and Modeling*, American Chemical Society (ACS), v. 59, n. 2, p. 786–796, jan. 2019. Disponível em: <https://doi.org/10.1021/acs.jcim.8b00572>.

87 CHEN, L. *et al.* Implication for alphavirus host-cell entry and assembly indicated by a 3.5{aa resolution cryo-EM structure. *Nature Communications*, Springer Science and Business Media LLC, v. 9, n. 1, dez. 2018. Disponível em: <https://doi.org/10.1038/s41467-018-07704-x>.

88 HARRIS, C. R. *et al.* Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>.

89 NGUYEN, H.; CASE, D. A.; ROSE, A. S. NGLview–interactive molecular graphics for Jupyter notebooks. *Bioinformatics*, v. 34, n. 7, p. 1241–1242, 12 2017. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btx789>.

90 INC., P. T. *Collaborative data science*. Montreal, QC: Plotly Technologies Inc., 2015. Disponível em: <https://plot.ly>.

91 PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

92 VIRTANEN, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020.

93 MICHALSKA, K. *et al.* Crystal structures of SARS-CoV-2 ADP-ribose phosphatase: from the apo form to ligand complexes. *IUCrJ*, v. 7, n. 5, p. 814–824, Sep 2020. Disponível em: <https://doi.org/10.1107/S2052252520009653>.

94 FEHR, A. R. *et al.* The conserved coronavirus macrodomain promotes virulence and suppresses the innate immune response during severe acute respiratory syndrome coronavirus infection. *mBio*, v. 7, n. 6, p. 10.1128/mbio.01721–16, 2016. Disponível em: <https://journals.asm.org/doi/abs/10.1128/mbio.01721-16>.

95 CLAVERIE, J.-M. A putative role of de-mono-ADP-ribosylation of STAT1 by the SARS-CoV-2 nsp3 protein in the cytokine storm syndrome of COVID-19. *Viruses*, MDPI AG, v. 12, n. 6, p. 646, jun. 2020. Disponível em: <https://doi.org/10.3390/v12060646>.

96 PETTERSEN, E. F. *et al.* Ucsf chimerax: Structure visualization for researchers, educators, and developers. *Protein Science*, Wiley, v. 30, n. 1, p. 70–82, out. 2020. Disponível em: <https://doi.org/10.1002/pro.3943>.

97 ROSE, A. S.; HILDEBRAND, P. W. NGL viewer: a web application for molecular visualization. *Nucleic Acids Research*, Oxford University Press (OUP), v. 43, n. W1, p. W576–W579, abr. 2015. Disponível em: <https://doi.org/10.1093/nar/gkv402>.

98 HUMPHREY, W.; DALKE, A.; SCHULTEN, K. VMD: Visual molecular dynamics. *Journal of Molecular Graphics*, Elsevier BV, v. 14, n. 1, p. 33–38, fev. 1996. Disponível em: <https://doi.org/10.1016/0263-7855(96)00018-5>.

99 BROSEY, C. A. *et al.* Targeting SARS-CoV-2 nsp3 macrodomain structure with insights from human poly(ADP-ribose) glycohydrolase (PARG) structures with inhibitors. *Progress in Biophysics and Molecular Biology*, Elsevier BV, v. 163, p. 171–186, ago. 2021. Disponível em: <https://doi.org/10.1016/j.pbiomolbio.2021.02.002>.

100 HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science &amp Engineering*, Institute of Electrical and Electronics Engineers (IEEE), v. 9, n. 3, p. 90–95, 2007. Disponível em: <https://doi.org/10.1109/mcse.2007.55>.

101 HOLM, L.; ROSENSTRöM, P. Dali server: conservation mapping in 3d. *Nucleic Acids Research*, Oxford University Press (OUP), v. 38, n. suppl_2, p. W545–W549, maio 2010. Disponível em: <https://doi.org/10.1093/nar/gkq366>.

102 KONAGURTHU, A. S. *et al.* MUSTANG: A multiple structural alignment algorithm. *Proteins: Structure, Function, and Bioinformatics*, Wiley, v. 64, n. 3, p. 559–574, maio 2006. Disponível em: <https://doi.org/10.1002/prot.20921>.

103 KRIEGER, E.; VRIEND, G. YASARA view—molecular graphics for all devices—from smartphones to workstations. *Bioinformatics*, Oxford University Press (OUP), v. 30, n. 20, p. 2981–2982, jul. 2014. Disponível em: <https://doi.org/10.1093/bioinformatics/btu426>.

104 ANDRIO, P. *et al.* BioExcel building blocks, a software library for interoperable biomolecular simulation workflows. *Scientific Data*, Springer Science and Business Media LLC, v. 6, n. 1, set. 2019. Disponível em: <https://doi.org/10.1038/s41597-019-0177-4>.

105 JEFFERSON, R. E. *et al.* Computational design of dynamic receptor—peptide signaling complexes applied to chemotaxis. *Nature Communications*, Springer Science and Business Media LLC, v. 14, n. 1, maio 2023. Disponível em: <https://doi.org/10.1038/s41467-023-38491-9>.

106 WASS, M. N.; KELLEY, L. A.; STERNBERG, M. J. E. 3dligandsite: predicting ligand-binding sites using similar structures. *Nucleic Acids Research*, Oxford University Press (OUP), v. 38, n. suppl_2, p. W469–W473, maio 2010. Disponível em: <https://doi.org/10.1093/nar/gkq406>.

107 STOURAC, J. *et al.* Caver web 1.0: identification of tunnels and channels in proteins and analysis of ligand transport. *Nucleic Acids Research*, Oxford University Press (OUP), v. 47, n. W1, p. W414–W422, maio 2019. Disponível em: <https://doi.org/10.1093/nar/gkz378>.

108 KOCHNEV, Y.; DURRANT, J. D. FPocketWeb: protein pocket hunting in a web browser. *Journal of Cheminformatics*, Springer Science and Business Media LLC, v. 14, n. 1, ago. 2022. Disponível em: <https://doi.org/10.1186/s13321-022-00637-0>.

109 MAGLIC, J. B.; LAVENDOMME, R. iMoloVol/i: an easy-to-use program for analyzing cavities, volumes and surface areas of chemical structures. *Journal of Applied Crystallography*, International Union of Crystallography (IUCr), v. 55, n. 4, p. 1033–1044, jun. 2022. Disponível em: <https://doi.org/10.1107/s1600576722004988>.

110 CHANG, W. *et al. shiny: Web Application Framework for R*. [S.l.], 2023. R package version 1.8.0.9000, https://github.com/rstudio/shiny. Disponível em: <https://shiny.posit.co/>.

111 van der Velden, N. *NGLViewR: Interactive 3D Visualization of Molecular Structures*. [S.l.], 2023. R package version 1.3.4. Disponível em: <https://github.com/nvelden/NGLViewR>.

112 MERKEL, D. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, v. 2014, n. 239, p. 2, 2014.

113 BRIK, A.; WONG, C.-H. HIV-1 protease: mechanism and drug discovery. *Organic &amp Biomolecular Chemistry*, Royal Society of Chemistry (RSC), v. 1, n. 1, p. 5–14, nov. 2002. Disponível em: <https://doi.org/10.1039/b208248a>.

114 WEBER, I.; AGNISWAMY, J. HIV-1 protease: Structural perspectives on drug resistance. *Viruses*, MDPI AG, v. 1, n. 3, p. 1110–1136, dez. 2009. Disponível em: <https://doi.org/10.3390/v1031110>.

115 JUBB, H. *et al.* Structural biology and drug discovery for protein–protein interactions. *Trends in Pharmacological Sciences*, Elsevier BV, v. 33, n. 5, p. 241–248, maio 2012. ISSN 0165-6147. Disponível em: <http://dx.doi.org/10.1016/j.tips.2012.03.006>.

116 GUERRA, J. V. *et al. SERD*. out. 2022. Disponível em: <https://github.com/LBC-LNBio/SERD>.

117 HUMMER, A. M. *et al.* Investigating the volume and diversity of data needed for generalizable antibody-antigen $\delta\delta$g prediction. *bioRxiv*, 2023. Disponível em: <https://www.biorxiv.org/content/early/2023/05/19/2023.05.17.541222>.

118 HAGBERG, A. A.; SCHULT, D. A.; SWART, P. J. Exploring Network Structure, Dynamics, and Function using NetworkX. In: VAROQUAUX, G.; VAUGHT, T.; MILLMAN, J. (Ed.). *Proceedings of the 7th Python in Science Conference*. Pasadena, CA USA: [s.n.], 2008. p. 11 – 15.

119 GOWERS, R. *et al.* MDAnalysis: A python package for the rapid analysis of molecular dynamics simulations. In: *Proceedings of the Python in Science Conference*. SciPy, 2016. Disponível em: <https://doi.org/10.25080/majora-629e541a-00e>.

120 ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, Elsevier BV, v. 20, p. 53–65, nov. 1987. ISSN 0377-0427. Disponível em: <http://dx.doi.org/10.1016/0377-0427(87)90125-7>.

121 KAUFMAN, L.; ROUSSEEUW, P. J. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990. ISSN 1940-6347. ISBN 9780470316801. Disponível em: <http://dx.doi.org/10.1002/9780470316801>.

122 MECOZZI, S.; REBEK, J. J. The 55 % solution: A formula for molecular recognition in the liquid state. *Chemistry - A European Journal*, Wiley, v. 4, n. 6, p. 1016–1022, jun. 1998. Disponível em: <https://doi.org/10.1002/(sici)1521-3765(19980615)4:6<1016::aid-chem1016>3.0.co;2-b>.

123 GROOM, C. R. *et al.* The Cambridge Structural Database. *Acta Crystallographica Section B*, v. 72, n. 2, p. 171–179, Apr 2016. Disponível em: <https://doi.org/10.1107/S2052520616003954>.

124 PLUTH, M. D. *et al.* Structural consequences of anionic host–cationic guest interactions in a supramolecular assembly. *Inorganic Chemistry*, American Chemical Society (ACS), v. 48, n. 1, p. 111–120, nov. 2008. ISSN 1520-510X. Disponível em: <http://dx.doi.org/10.1021/ic8012848>.

125 STEEL, P. J.; MCMORRAN, D. A. Selective anion recognition by a dynamic quadruple helicate. *Chemistry – An Asian Journal*, Wiley, v. 14, n. 8, p. 1098–1101, set. 2018. ISSN 1861-471X. Disponível em: <http://dx.doi.org/10.1002/asia.201801262>.

126 NAKAMURA, T. *et al.* A c60-templated tetrameric porphyrin barrel complex via zinc-mediated self-assembly utilizing labile capping ligands. *Journal of the American Chemical Society*, American Chemical Society (ACS), v. 135, n. 50, p. 18790–18793, dez. 2013. ISSN 1520-5126. Disponível em: <http://dx.doi.org/10.1021/ja4110446>.

127 EICHSTAEDT, K. *et al.* Self-assembly and ordering of peptide-based cavitands in water and dmso: The power of hydrophobic effects combined with neutral hydrogen bonds. *Chemistry – A European Journal*, Wiley, v. 25, n. 12, p. 3091–3097, jan. 2019. ISSN 1521-3765. Disponível em: <http://dx.doi.org/10.1002/chem.201805353>.

128 JOHNSON, D. W. *et al.* Solid-state and solution studies of a tetrameric capsule and its guests. *Angewandte Chemie*, v. 114, n. 20, p. 3947–3950, 2002. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/1521-3757%2820021018%29114%3A20%3C3947%3A%3AAID-ANGE3947%3E3.0.CO%3B2-X>.

129 CAULDER, D. L. *et al.* The self-assembly of a predesigned tetrahedral m4l6 supramolecular cluster. *Angewandte Chemie International Edition*, v. 37, n. 13-14, p. 1840–1843, 1998. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291521-3773%2819980803%2937%3A13/14%3C1840%3A%3AAID-ANIE1840%3E3.0.CO%3B2-D>.

130 YANG, Y. *et al.* A curved host and second guest cooperatively inhibit the dynamic motion of corannulene. *Nature Communications*, Springer Science and Business Media LLC, v. 12, n. 1, jul. 2021. ISSN 2041-1723. Disponível em: <http://dx.doi.org/10.1038/s41467-021-24344-w>.

131 MACGILLIVRAY, L. R.; ATWOOD, J. L. A chiral spherical molecular assembly held together by 60 hydrogen bonds. *Nature*, Springer Science and Business Media LLC, v. 389, n. 6650, p. 469–472, out. 1997. ISSN 1476-4687. Disponível em: <http://dx.doi.org/10.1038/38985>.

132 KAWAKAMI, Y. *et al.* Silane catecholates: versatile tools for self-assembled dynamic covalent bond chemistry. *Chemical Communications*, Royal Society of Chemistry (RSC), v. 55, n. 43, p. 6066–6069, 2019. ISSN 1364-548X. Disponível em: <http://dx.doi.org/10.1039/C9CC02103E>.

133 YOSHIZAWA, M.; TAMURA, M.; FUJITA, M. Diels-alder in aqueous molecular hosts: Unusual regioselectivity and efficient catalysis. *Science*, American Association for

the Advancement of Science (AAAS), v. 312, n. 5771, p. 251–254, abr. 2006. ISSN 1095-9203. Disponível em: <http://dx.doi.org/10.1126/science.1124985>.

134 FUJITA, D. *et al.* Self-assembly of tetravalent goldberg polyhedra from 144 small components. *Nature*, Springer Science and Business Media LLC, v. 540, n. 7634, p. 563–566, dez. 2016. ISSN 1476-4687. Disponível em: <http://dx.doi.org/10.1038/nature20771>.

135 LIAO, P. *et al.* Two-component control of guest binding in a self-assembled cage molecule. *Chemical Communications*, Royal Society of Chemistry (RSC), v. 46, n. 27, p. 4932, 2010. ISSN 1364-548X. Disponível em: <http://dx.doi.org/10.1039/C0CC00234H>.

136 RONSON, T. K.; MENG, W.; NITSCHKE, J. R. Design principles for the optimization of guest binding in aromatic-paneled feii4l6 cages. *Journal of the American Chemical Society*, American Chemical Society (ACS), v. 139, n. 28, p. 9698–9707, jul. 2017. ISSN 1520-5126. Disponível em: <http://dx.doi.org/10.1021/jacs.7b05202>.

137 ZHANG, H.-B. *et al.* Hydrophobic and metal-coordinated confinement effects trigger recognition and selectivity. *The Journal of Organic Chemistry*, American Chemical Society (ACS), v. 86, n. 13, p. 8873–8881, jun. 2021. ISSN 1520-6904. Disponível em: <http://dx.doi.org/10.1021/acs.joc.1c00794>.

138 CULLEN, W. *et al.* Highly efficient catalysis of the kemp elimination in the cavity of a cubic coordination cage. *Nature Chemistry*, Springer Science and Business Media LLC, v. 8, n. 3, p. 231–236, fev. 2016. ISSN 1755-4349. Disponível em: <http://dx.doi.org/10.1038/nchem.2452>.

139 MENDONcA, D. C. *et al.* An atomic model for the human septin hexamer by cryo-em. *Journal of Molecular Biology*, Elsevier BV, v. 433, n. 15, p. 167096, jul. 2021. ISSN 0022-2836. Disponível em: <http://dx.doi.org/10.1016/j.jmb.2021.167096>.

140 MERCALDI, G. F. *et al.* Discovery and structural characterization of chicoric acid as a sars-cov-2 nucleocapsid protein ligand and rna binding disruptor. *Scientific Reports*, Springer Science and Business Media LLC, v. 12, n. 1, nov. 2022. ISSN 2045-2322. Disponível em: <http://dx.doi.org/10.1038/s41598-022-22576-4>.

141 SANTHAKUMARI, P. R. *et al.* Variability in phenylalanine side chain conformations facilitates broad substrate tolerance of fatty acid binding in cockroach milk proteins. *PLOS ONE*, Public Library of Science (PLoS), v. 18, n. 6, p. e0280009, jun. 2023. ISSN 1932-6203. Disponível em: <http://dx.doi.org/10.1371/journal.pone.0280009>.

142 PREMETIS, G. E. *et al.* Structural and functional features of a broad-spectrum prophage-encoded enzybiotic from enterococcus faecium. *Scientific Reports*, Springer Science and Business Media LLC, v. 13, n. 1, maio 2023. ISSN 2045-2322. Disponível em: <http://dx.doi.org/10.1038/s41598-023-34309-2>.

143 CHEN, Z. *et al.* Cryo-em structures of human spca1a reveal the mechanism of ca 2+ /mn 2+ transport into the golgi apparatus. *Science Advances*, American Association for the Advancement of Science (AAAS), v. 9, n. 9, mar. 2023. ISSN 2375-2548. Disponível em: <http://dx.doi.org/10.1126/sciadv.add9742>.

144 KIM, D. *et al.* Domain swapping of the c-terminal helix promotes the dimerization of a novel ribonuclease protein from mycobacterium tuberculosis. *Protein Science*, Wiley, v. 32, n. 6, maio 2023. ISSN 1469-896X. Disponível em: <http://dx.doi.org/10.1002/pro.4644>.

145 GLASSER, N. R. *et al.* Accelerating the discovery of alkyl halide-derived natural products using halide depletion. *Nature Chemistry*, Springer Science and Business Media LLC, jan. 2024. ISSN 1755-4349. Disponível em: <http://dx.doi.org/10.1038/s41557-023-01390-z>.

146 LI, N. *et al.* A rumen-derived bifunctional glucanase/mannanase uncanonically releases oligosaccharides with a high degree of polymerization preferentially from branched substrates. *Carbohydrate Polymers*, Elsevier BV, v. 330, p. 121828, abr. 2024. ISSN 0144-8617. Disponível em: <http://dx.doi.org/10.1016/j.carbpol.2024.121828>.

147 RIVERA, K. G. *et al.* Antimicrobial peptide recognition motif of the substrate binding protein sapa from nontypeable haemophilus influenzae. *Biochemistry*, American Chemical Society (ACS), jan. 2024. ISSN 1520-4995. Disponível em: <http://dx.doi.org/10.1021/acs.biochem.3c00562>.

148 WANG, Y.; WEI, Z.; XI, L. Sfcnn: a novel scoring function based on 3d convolutional neural network for accurate and stable protein–ligand affinity prediction. *BMC Bioinformatics*, Springer Science and Business Media LLC, v. 23, n. 1, jun. 2022. ISSN 1471-2105. Disponível em: <http://dx.doi.org/10.1186/s12859-022-04762-3>.

149 SKALIC, M. *et al.* Ligvoxel: inpainting binding pockets using 3d-convolutional neural networks. *Bioinformatics*, Oxford University Press (OUP), v. 35, n. 2, p. 243–250, jul. 2018. ISSN 1367-4811. Disponível em: <http://dx.doi.org/10.1093/bioinformatics/bty583>.

150 PU, L. *et al.* Deepdrug3d: Classification of ligand-binding pockets in proteins with a convolutional neural network. *PLOS Computational Biology*, Public Library of Science (PLoS), v. 15, n. 2, p. e1006718, fev. 2019. ISSN 1553-7358. Disponível em: <http://dx.doi.org/10.1371/journal.pcbi.1006718>.

151 XU, Y. *et al.* Cavityplus: a web server for protein cavity detection with pharmacophore modelling, allosteric site identification and covalent ligand binding ability prediction. *Nucleic Acids Research*, Oxford University Press (OUP), v. 46, n. W1, p. W374–W379, maio 2018. ISSN 1362-4962. Disponível em: <http://dx.doi.org/10.1093/nar/gky380>.

152 COLEMAN, R. G.; SALZBERG, A. C.; CHENG, A. C. Structure-based identification of small molecule binding sites using a free energy model. *Journal of Chemical Information and Modeling*, American Chemical Society (ACS), v. 46, n. 6, p. 2631–2637, nov. 2006. ISSN 1549-960X. Disponível em: <http://dx.doi.org/10.1021/ci600229z>.

153 TAN, Z. W. *et al.* Allosigma 2: paving the way to designing allosteric effectors and to exploring allosteric effects of mutations. *Nucleic Acids Research*, Oxford University Press (OUP), v. 48, n. W1, p. W116–W124, maio 2020. ISSN 1362-4962. Disponível em: <http://dx.doi.org/10.1093/nar/gkaa338>.

154 KAYNAK, B. T.; BAHAR, I.; DORUKER, P. Essential site scanning analysis: A new approach for detecting sites that modulate the dispersion of protein global motions. *Computational and Structural Biotechnology Journal*, Elsevier BV, v. 18, p. 1577–1586, 2020. ISSN 2001-0370. Disponível em: <http://dx.doi.org/10.1016/j.csbj.2020.06.020>.