

SOFTWARE

Open Access



FPocketWeb: protein pocket hunting in a web browser

Yuri Kochnev and Jacob D. Durrant*

Abstract

Detecting macromolecular (e.g., protein) cavities where small molecules bind is an early step in computer-aided drug discovery. Multiple pocket-detection algorithms have been developed over the past several decades. Among them, *fpocket*, created by Schmidtke and Le Guilloux, is particularly popular. Like many programs used in computational-biology research, *fpocket* requires users to download and install an executable file. That file must also be run via a command-line interface, further complicating use. An existing *fpocket* server application effectively addresses these challenges, but it requires users to upload their possibly proprietary structures to a third-party server. The FPocketWeb web app builds on this prior work. It runs the *fpocket3* executable entirely in a web browser without requiring installation. The pocket-finding calculations occur on the user's computer rather than on a remote server. A working version of the open-source FPocketWeb app can be accessed free of charge from <http://durrantlab.com/fpocketweb>.

Keywords: Computer-aided drug discovery, Ligand-binding pocket, Fpocket, Browser app, Online resource, WebAssembly

Introduction

Proteins perform various cellular functions, ranging from mediating cell-signaling pathways to catalyzing vital chemical reactions. Many of these functions depend on interactions between protein receptors and endogenous small-molecule ligands that bind in cavities (pockets) typically located on protein surfaces. Carefully designed exogenous molecules can also bind in protein pockets, competing directly or indirectly with endogenous ligands and thus altering protein activity. Such molecules can serve as useful scientific tools (chemical probes) for revealing the basic biology of proper protein function. When properly optimized, they can also become pharmaceuticals that modulate the activity of disease-implicated proteins.

Structure-based computer-aided drug discovery (CADD) leverages known protein structures to identify

and design novel exogenous ligands. Central to these efforts is identifying the protein pocket where the ligand will bind. In many cases, the location of the ligand-binding pocket is well-characterized. For example, crystallographic or NMR structures may include a pocket-bound ligand, and biochemical experiments can identify pocket-lining catalytic residues that interact with endogenous small-molecule substrates. But often, the location of the ligand-binding pocket is uncharacterized.

In such cases, cavity detection programs can serve as valuable tools for pocket identification. These programs include (a) "geometric" approaches that search for cavities based on the positions of protein-receptor atoms, and (b) "energy-based" approaches that further consider the physicochemical properties of protein residues to identify sites that might energetically favor ligand binding [1]. Among the few programs that fall into the energy-based category, the open-source tool *fpocket* is particularly popular. *fpocket* is written in C, is relatively easy to use, and is notably fast.

As with many CADD programs, the original *fpocket* implementation requires users to (1) download an

*Correspondence: durrantj@pitt.edu

Department of Biological Sciences, University of Pittsburgh, 15260 Pittsburgh, PA, USA



executable program, (2) choose proper configuration parameters, (3) run the program from a Unix- or DOS-like command-line interface, and (4) separately analyze and visualize the program output. The *fpocket* authors have taken several measures to address these challenges. To facilitate output analysis and visualization, *fpocket* saves output files in formats compatible with VMD [2] and PyMOL [3], two popular visualization programs. And to enable use without the command line, the authors created a helpful web server so users can run pocket-hunting calculations in the cloud [4].

These efforts have substantially improved usability, but some challenges remain. VMD- and PyMOL-formatted output files enable more accessible analysis than text-based output. Still, users must separately download and install these third-party visualization programs, and each has a substantial learning curve. Similarly, the *fpocket* server app eliminates command-line use, but it requires users to upload their (possibly proprietary) structures to a third-party system. Furthermore, in principle, hosting such an app requires an extensive and difficult-to-maintain backend computing infrastructure. Long wait queues may be necessary if a remote resource becomes saturated with requests.

To build on these prior efforts to enhance *fpocket* usability, we created FPocketWeb, a WebAssembly (Wasm) enabled web app that runs *fpocket* entirely in users' browsers, without the need for other plugins or programs. Wasm-compiled code runs locally on the user's computer, so it does not depend on remote infrastructure to run complex calculations. Instead, a simple web server—or even a “thin server”—is sufficiently powerful to send the Wasm-compiled binaries to the user's browser.

We release FPocketWeb under the terms of the permissive open-source Apache License, Version 2.0, which is compatible with *fpocket3*'s MIT license. The FPocketWeb source code [5] and browser app [6] are publicly available online without requiring registration.

Implementation

Wasm compilation

We downloaded the *fpocket3* codebase, written in the C programming language, in 2022 [7]. We then used Emscripten 3.1.4 to compile it to a Wasm module. The *fpocket* source code required only minor adjustments to be Wasm-compileable. To create the output directories and subdirectories, the original *fpocket* uses a system call to execute the *mkdir* command. This command is readily available on operating systems such as Linux, but it is not available in the browser environment. We therefore commented out the offending line before Wasm compilation.

We separately create the required directories on the virtual file system using JavaScript instead.

We also did not implement the full functionality of the VMD molfile plugin [8]. Specifically, our compilation excludes references to the netCDF library, which allows command-line *fpocket3* to load netCDF files. FPocketWeb therefore supports only PDB files.

FPocketWeb browser app

The FPocketWeb graphical user interface (GUI) allows users to quickly load protein structures into the browser's memory, specify FPocketWeb parameters, run calculations using the Wasm-compiled *fpocket* module, and visualize/download the results. To create the interface, we followed the same approach we have used previously [9–12]. The FPocketWeb interface is written in the Microsoft TypeScript programming language, which compiles to JavaScript. The open-source Vue.js framework [13] provides consistently styled components (e.g., buttons, text fields, etc.). Many of these components are derived from BootstrapVue [14], an open-source library that implements the Bootstrap4 framework [15]. We also adapted our existing molecular-visualization Vue.js component [9, 11, 12], which leverages the 3Dmol.js library [16]. Finally, we used Webpack [17] and Google Closure Compiler [18] to compile, assemble, and optimize the code for size and speed.

Results and discussion

FPocketWeb use

Input parameters tab

To run FPocketWeb, users need only visit the FPocketWeb website [6]. The “Input Parameters” tab then appears, which includes the “Input File” and “Advanced Parameters” subsections (Fig. 1A, B). In the “Input File” subsection (Fig. 1A), users can specify the protein file (PDB format) for pocket hunting. This file is opened locally, but it is never uploaded to any third-party server. Users can also click the “Use Example File” button to load *H. sapiens* heat shock protein 90 (Hsp90, PDB 5J2V [19]) for testing. Loaded files are displayed in the “PDB Preview” subsection (not shown) using the 3Dmol.js molecular viewer [16].

The “Advanced Parameters” subsection (Fig. 1B) allows users to specify the same parameters available through the *fpocket* command-line executable so they can fine-tune the underlying pocket-finding method. Those interested in further details should consult the original *fpocket* manuscripts [1, 4]. FPocketWeb initially hides these parameters

A Input File

PDB File:

B Advanced Parameters

Optional output parameters

Optional input parameters

Optional pocket detection parameters

C Visualization

Execution time: 1 seconds

D Detected Pockets

Pocket #	Display Color	Score	Druggability Score	Number Of Alpha Spheres	Volume	Show Details
1		0.377	0.808	98	698.828	<input type="checkbox"/>
2		0.222	0.004	44	524.250	<input type="checkbox"/>
3		0.090	0.001	23	137.013	<input type="checkbox"/>

« < 1 2 > »

Current Page: 1

Fig. 1 Select subsections of the “Input Parameters” and “Output” tabs. **A** The “Input File” subsection allows users to specify the protein for analysis. **B** The “Advanced Parameters” subsection allows users to fine-tune the pocket-hunting algorithm. **C** The “Visualization” subsection shows the detected pockets. **D** The “Deteted Pockets” subsection describes the properties of each pocket.

because most users will prefer to use the default values. Once ready, users can click the “Start FPocketWeb” button (Fig. 1C) to initiate the FPocketWeb run.

Output tab

FPocketWeb displays the “Output” tab once the calculations finish (Fig. 1). This tab includes the “Visualization” and “Detected Pockets” subsections. The “Visualization” subsection shows the protein and detected cavities (Fig. 1C). Initially, only the pocket with the highest *fpocket* score is shown. When running FPocketWeb on a device with a sufficiently wide screen, a list of the detected pockets appears on the right (not shown), allowing the user to easily toggle individual pocket visualizations on and off. The “Detected Pockets” subsection (Fig. 1D) contains a table with the detailed output for each detected pocket, one pocket per row. Initially, only the score, the druggability score, the number of alpha spheres, and the volume are displayed, but the “Show Details” toggle allows users to display all other *fpocket* metrics for each detected pocket. Users can also change the color used to visualize each pocket.

The “Output Files” subsection (not shown) allows users to view the FPocketWeb output PDB file, which includes the original protein structure and all the detected pockets. Users can press the “Download” button to save the file. Finally, the “Run Fpocket from the Command Line” subsection (also not shown) provides a code snippet so users can run *fpocket* from the command line with the same FPocketWeb parameters used in the browser.

Benchmarks and compatibility

We performed benchmark calculations on a Linux laptop (HP Pavilion 15.6”) running Fedora 34 (1.3 GHz Intel Core i7 processor and 16 GB 2400 MHz DDR4 memory) to compare FPocketWeb and command-line *fpocket* (Table 1). The two programs had comparable execution times and produced identical scores, druggability scores, numbers of alpha spheres, and volumes. After these initial benchmarks, we further tested FPocketWeb on the browser/operating-system combinations shown in Table 2. The app is compatible with desktop and mobile operating systems, as well as all major browsers (e.g., Chrome, Edge, Safari, and Firefox).

Example of use: TEM-1 β -lactamase

As a demonstration of use, we first considered TEM-1, a bacterial β -lactamase. TEM-1 contributes to penicillin and cephalosporin resistance in Gram-negative bacteria [24] by hydrolyzing the four-carbon ring common to all β -lactam antibiotics. Aside from the orthosteric pocket where the ring is hydrolyzed, a recently discovered allosteric cryptic pocket [25–28] is collapsed and hidden (PDB ID: 1FQG [29]) unless co-crystallized with

Table 1 FPocketWeb/*fpocket* benchmarks

PDB	Protein	FPocketWeb Time (s)	<i>fpocket</i> Time (s)	Score
1HWL [20]	HMG-CoA reductase	5.6 ± 0.095	5.1 ± 0.149	0.437
2P16 [21]	factor Xa	0.7 ± 0.032	0.6 ± 0.009	0.961
3LN1 [22]	COX-2	15.9 ± 0.514	16.0 ± 0.145	0.630
4LL3 [23]	HIV protease	0.4 ± 0.052	0.3 ± 0.015	0.862
4TZ4	Cereblon	5.2 ± 0.048	5.3 ± 0.075	0.301

Times are reported in seconds. FPocketWeb and *fpocket* produced the same scores; we report the scores associated with the top-ranked pockets. The calculations were performed on PDB files downloaded directly from the Protein Data Bank, without further processing.

Table 2 Browser compatibility

Browser	Operating system
Chrome 101.0.4951.41	macOS 12.1
Firefox 99.0.1	macOS 12.1
Safari 15.2	macOS 12.1
Chrome 101.0.4951.41	Windows 10 Enterprise
Firefox 99.0.1	Windows 10 Enterprise
Edge 100.0.1185.50	Windows 10 Enterprise
Chrome 100.0.4896.127	Android 12
Firefox 99.2.0	Android 12
Safari 15	iOS 15.4.1
Firefox 99.0	Ubuntu 20.04.4 LTS
Chrome 101.0.4951.41	Ubuntu 20.04.4 LTS

We have tested FPocketWeb on all major browsers and operating systems.

a bound ligand (PDB ID: 1PZP [28]). TEM-1 is often used to benchmark computational tools for cryptic-pocket identification [25, 30, 31].

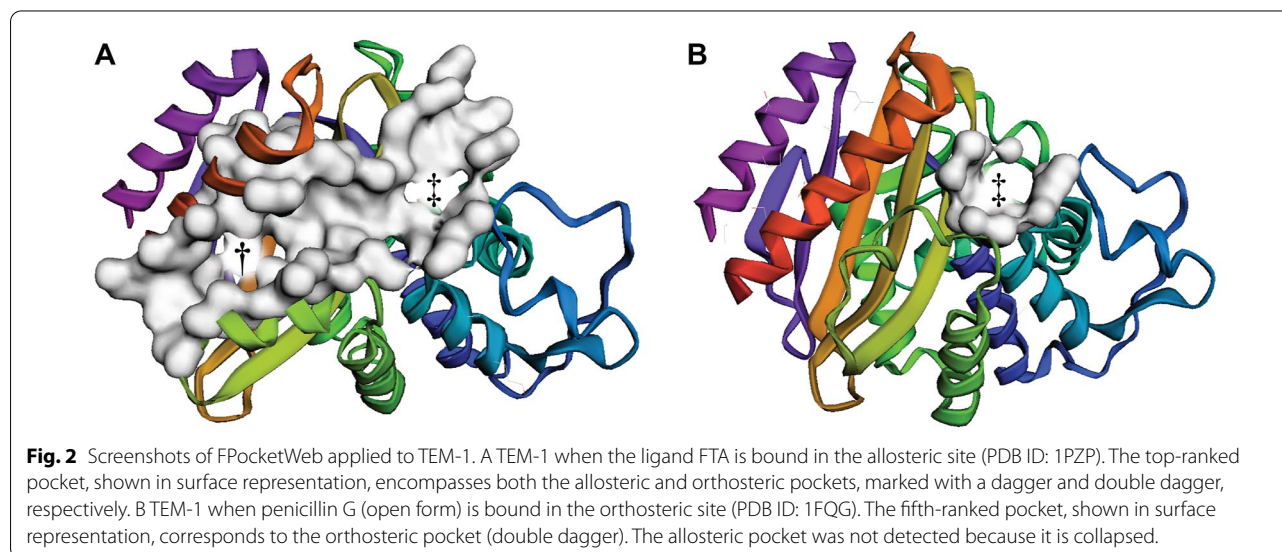
We applied FPocketWeb to TEM-1 crystal structures with (Z)-3-[(4-phenylazanylphenyl)amino]-2-(2 H-1,2,3,4-tetrazol-5-yl)prop-2-enitrile (FTA, PDB

ID: 1PZP [28]) and penicillin G (open form; PDB ID: 1FQG [29]) bound in the allosteric and orthosteric pockets, respectively. We first used FPocketWeb to remove all non-protein atoms before detecting pockets on the protein surfaces.

This analysis shows the dramatic impact that the allosteric inhibitor FTA has on the TEM-1 structure. The top-ranked pocket detected when we applied FPocketWeb to the FTA-bound structure (PDB ID: 1PZP [28]) encompassed the allosteric and orthosteric sites (Fig. 2A), with a total volume of 2161 Å³. In contrast, the allosteric pocket was not detected when we applied FPocketWeb to the penicillin-bound structure (PDB ID: 1FQG [29]) because the allosteric pocket was unoccupied and collapsed. Even the orthosteric pocket ranked only fifth (Fig. 2B, 350 Å³).

Example of use: influenza neuraminidase

As a second demonstration of use, we consider the enzymatic pocket of influenza neuraminidase (N1). N1 is an exosialidase that prevents the aggregation of newly formed viral particles on infected-cell surfaces [32]. The primary N1 enzymatic pocket [33, 34] is persistent even



in the absence of a bound ligand. But a flexible pocket-adjacent “150 loop” [35–37] enables a pocket extension (the 150-cavity) that is collapsed in many crystal structures. Others have designed N1 inhibitors that exploit this cavity [38].

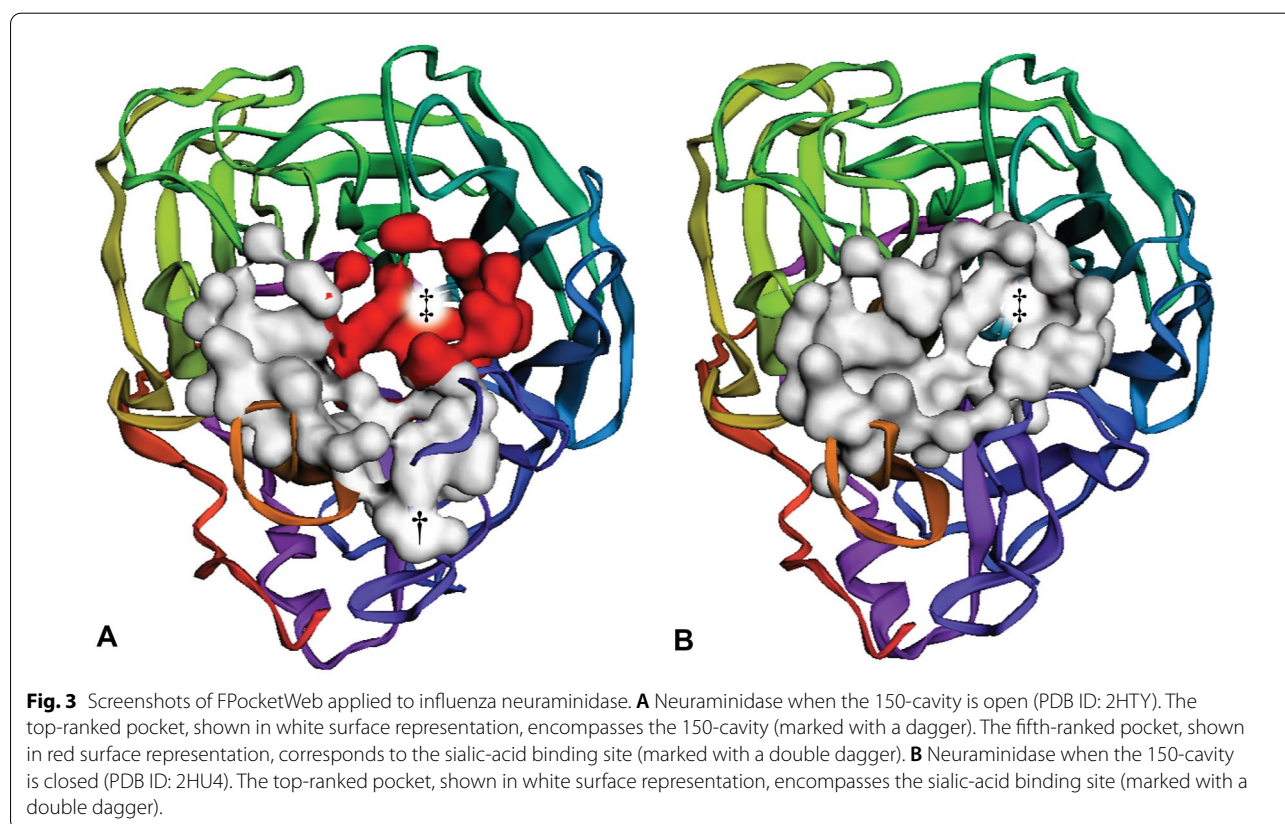
We applied FPocketWeb to N1 crystal structures in open (PDB ID: 2HTY [37]) and closed (PDB ID: 2HU4 [37]) 150-cavity conformations. We again removed any non-protein atoms before detecting pockets on the protein surfaces. The top-ranked pocket detected when we applied FPocketWeb to the open-150-cavity structure included the 150-cavity (Fig. 3A, marked with a dagger and shown as a white surface, 911 Å³). The fifth-ranked pocket corresponded to the known sialic-acid binding site (Fig. 3A, marked with a double dagger and shown as a red surface, 491 Å³). In contrast, when we applied FPocketWeb to the closed-150-cavity structure, the 150-cavity was not detected. The top-ranked pocket did include the known sialic-acid binding site (Fig. 3B, marked with a double dagger, 1142 Å³).

Limitations

In compiling *fpocket* to Wasm, we also compiled the related programs specified in the same makefile, including *mdpocket* (a tool for analyzing pocket conformations

sampled by molecular dynamics simulations), *dpocket* (a tool for extracting physicochemical pocket descriptors), and *tpocket* (a tool for evaluating cavity scoring functions). However, we subsequently focused our web-based implementation and testing on *fpocket* alone. Implementing *mdpocket* as a browser app is also appealing, but it would require loading an entire MD trajectory into the browser’s memory, which is not presently practical. For those interested in using an online version of *mdpocket*, we recommend the original *fpocket* creators’ useful server application, which allows users to upload their trajectory files and run *mdpocket* in the cloud [4].

Another limitation is that FPocketWeb only accepts files in the PDB format. The original *fpocket3* executable uses the molfile plugin [8] to load structure and trajectory files in many formats. In some cases (e.g., the netCDF format), the molfile plugin leverages other libraries with many dependencies. Given that PDB is the *de facto* standard for static protein structures and that FPocketWeb is not meant to work with trajectory files (i.e., it does not implement *mdpocket*), we opted to focus on the PDB format alone. Users with files in other formats can easily convert them to the PDB format using programs such as VMD [2] and Open Babel [39].



Conclusion

FPocketWeb is an easy-to-use pocket-finding tool that allows users to run the *fpocket* executable entirely in a web browser, without the need for explicit download and installation. The pocket-finding calculations occur on the user's computer rather than on a remote server, so FPocketWeb does not require substantial remote infrastructure, nor does it require users to upload their possibly proprietary structures. And yet it retains the usability of a web-based interface, and the identified pockets are visualized in the browser itself, without requiring third-party software such as VMD. In our tests of five proteins, FPocketWeb and the original *fpocket* had comparable run times and produced identical pocket scores.

We are hopeful that FPocketWeb's utility and usability will make it a broadly useful open-source tool for those interested in computational structural biology and computer-aided drug discovery. We release it under the Apache License, Version 2.0. The source code [5] and browser app [6] are available free of charge online, without requiring registration.

Availability and requirements

Project name: FPocketWeb.

Project home page: <http://durrantlab.com/fpocketweb-download>.

Operating system(s): Windows, macOS, Linux, Android, iOS.

Programming language: TypeScript, JavaScript, HTML, CSS, WebAssembly.

Other requirements: Any modern web browser.

License: Apache License, Version 2.0.

Any restrictions to use by non-academics: None.

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s13321-022-00637-0>.

Additional file 1. The FPocketWeb source code, version 1.0.1. See <http://durrantlab.com/fpocketweb-download> for the latest version.

Acknowledgements

We acknowledge the University of Pittsburgh's Center for Research Computing for providing valuable computer resources.

Author contributions

YK: Compiled the *fpocket3* codebase to WebAssembly, performed benchmark tests, helped create the FPocketWeb GUI, and contributed to the writing of the manuscript. JDD: Helped create the GUI, provided two examples of use, contributed to the writing of the manuscript, and provided research supervision. Both the authors read and approved the final manuscript.

Funding

This work was supported by the National Institute of General Medical Sciences of the National Institutes of Health [R01GM132353 to JDD]. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Availability of data and materials

The FPocketWeb source code is available free of charge from <http://durrantlab.com/fpocketweb-download>, and the web app can be accessed at <http://durrantlab.com/fpocketweb>. Both URLs are publicly available, without requiring registration. A copy of the source code is also provided as Additional file 1.

Declarations

Competing interests

Not applicable.

Received: 10 June 2022 Accepted: 4 August 2022

Published online: 26 August 2022

References

- Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: an open source platform for ligand pocket detection. *BMC Bioinformatics* 10:168. <https://doi.org/10.1186/1471-2105-10-168>
- Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. *J Mol Graph* 14:33–38. [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5)
- Schrödinger L (2015) The PyMOL Molecular Graphics System, Version, 1.8 edn
- Schmidtke P, Le Guilloux V, Maupetit J, Tuffery P (2010) fpocket: online tools for protein ensemble pocket detection and tracking. *Nucleic Acids Res* 38:W582–589. <https://doi.org/10.1093/nar/gkq383>
- Durrant J (2022) FPocketWeb-Durrant Lab. <https://doi.org/https://durrantlab.pitt.edu/fpocketweb-download>. Accessed 9 June 2022
- Durrant J (2022) FPocketWeb. <https://doi.org/http://durrantlab.com/fpocketweb>. Accessed 9 June 2022
- Discngine/fpocket (2022) <https://doi.org/https://github.com/Discngine/fpocket>. Accessed 2022
- Molfile Plugin Documentation (2021) University of Illinois at Urbana-Champaign. <https://doi.org/https://www.ks.uiuc.edu/Research/vmd/plugins/molfile/>. Accessed 28 April 2022
- Young J, Garikipati N, Durrant JD (2022) BINANA 2: characterizing receptor/ligand interactions in Python and JavaScript. *J Chem Inf Model* 62:753–760. <https://doi.org/10.1021/acs.jcim.1c01461>
- Durrant JD (2022) Prot2Prot: a deep learning model for rapid. Photorealistic macromolecular visualization. *bioRxiv*
- Green H, Durrant JD (2021) DeepFrag: an open-source browser app for deep-learning lead optimization. *J Chem Inf Model* 61:2523–2529. <https://doi.org/10.1021/acs.jcim.1c00103>
- Kochnev Y, Helleman E, Cassidy KC, Durrant JD (2020) Webina: an open-source library and web app that runs AutoDock Vina entirely in the web browser. *Bioinformatics* 36:4513–4515. <https://doi.org/10.1093/bioinformatics/btaa579>
- Vue.js—The Progressive JavaScript Framework. <https://doi.org/https://vuejs.org/>. Accessed 9 June 2022
- BootstrapVue. <https://bootstrap-vue.org/>. Accessed 9 June 2022
- Bootstrap: The most popular HTML, CSS, and JS library in the world. <https://doi.org/https://getbootstrap.com/>. Accessed 9 June 2022.
- Rego N, Koes D (2015) 3Dmol.js: molecular visualization with WebGL. *Bioinformatics* 31:1322–1324. <https://doi.org/10.1093/bioinformatics/btu829>
- webpack. <https://webpack.js.org/>. Accessed 9 June 2022

18. Closure Compiler. <https://doi.org/https://developers.google.com/closure/compiler>. Accessed 9 June 2022
19. Amaral M et al (2017) Protein conformational flexibility modulates kinetics and thermodynamics of drug binding. *Nat Commun* 8:2276. <https://doi.org/10.1038/s41467-017-02258-w>
20. Istvan ES, Deisenhofer J (2001) Structural mechanism for statin inhibition of HMG-CoA reductase. *Science* 292:1160–1164. <https://doi.org/10.1126/science.1059344>
21. Pinto DJ et al (2007) Discovery of 1-(4-methoxyphenyl)-7-oxo-6-(4-(2-oxopiperidin-1-yl)phenyl)-4,5,6,7-tetrahydro-1H-pyrazolo[3,4-c]pyridine-3-carboxamide (apixaban, BMS-562247), a highly potent, selective, efficacious, and orally bioavailable inhibitor of blood coagulation factor Xa. *J Med Chem* 50:5339–5356. <https://doi.org/10.1021/jm070245n>
22. Wang JL et al (2010) The novel benzopyran class of selective cyclooxygenase-2 inhibitors. Part 2: the second clinical candidate having a shorter and favorable human half-life. *Bioorg Med Chem Lett* 20:7159–7163. <https://doi.org/10.1016/j.bmcl.2010.07.054>
23. Kozisek M, Lepsik M, Grantz Saskova K, Brynda J, Konvalinka J, Rezacova P (2014) Thermodynamic and structural analysis of HIV protease resistance to darunavir—analysis of heavily mutated patient-derived HIV-1 proteases. *FEBS J* 281:1834–1847. <https://doi.org/10.1111/febs.12743>
24. Wang X, Minasov G, Shoichet BK (2002) Evolution of an antibiotic resistance enzyme constrained by stability and activity trade-offs. *J Mol Biol* 320:85–95. [https://doi.org/10.1016/S0022-2836\(02\)00400-X](https://doi.org/10.1016/S0022-2836(02)00400-X)
25. Bowman GR, Geissler PL (2012) Equilibrium fluctuations of a single folded protein reveal a multitude of potential cryptic allosteric sites. *Proc Natl Acad Sci USA* 109:11681–11686. <https://doi.org/10.1073/pnas.1209309109>
26. Grimsley JK, Calamini B, Wild JR, Mesecar AD (2005) Structural and mutational studies of organophosphorus hydrolase reveal a cryptic and functional allosteric-binding site. *Arch Biochem Biophys* 442:169–179. <https://doi.org/10.1016/j.abb.2005.08.012>
27. Cimermancic P et al (2016) CryptoSite: expanding the druggable proteome by characterization and prediction of cryptic binding sites. *J Mol Biol* 428:709–719. <https://doi.org/10.1016/j.jmb.2016.01.029>
28. Horn JR, Shoichet BK (2004) Allosteric inhibition through core disruption. *J Mol Biol* 336:1283–1291. <https://doi.org/10.1016/j.jmb.2003.12.068>
29. Strynadka NC et al (1992) Molecular structure of the acyl-enzyme intermediate in beta-lactam hydrolysis at 1.7 Å resolution. *Nature* 359:700–705. <https://doi.org/10.1038/359700a0>
30. Bowman GR, Bolin ER, Hart KM, Maguire BC, Marqusee S (2015) Discovery of multiple hidden allosteric sites by combining Markov state models and experiments. *Proc Natl Acad Sci USA* 112:2734–2739. <https://doi.org/10.1073/pnas.1417811112>
31. Oleinikovas V, Saladino G, Cossins BP, Gervasio FL (2016) Understanding cryptic pocket formation in protein targets by enhanced sampling simulations. *J Am Chem Soc* 138:14257–14263. <https://doi.org/10.1021/jacs.6b05425>
32. Shtyrya YA, Mochalova LV, Bovin NV (2009) Influenza virus neuraminidase: structure and function. *Acta Naturae* 1:26–32. <https://doi.org/10.1111/j.1747-0285.2011.01148.x>
33. Durrant JD, McCammon JA (2010) Potential drug-like inhibitors of Group 1 influenza neuraminidase identified through computer-aided drug design. *Comput Biol Chem* 34:97–105. <https://doi.org/10.1016/j.compbiolchem.2010.03.005>
34. Durrant JD, Bush RM, Amaro RE (2016) Microsecond molecular dynamics simulations of influenza neuraminidase suggest a mechanism for the increased virulence of stalk-deletion mutants. *J Phys Chem B*. <https://doi.org/10.1021/acs.jpcc.6b02655>
35. Amaro RE, Swift RV, Votapka L, Li WW, Walker RC, Bush RM (2011) Mechanism of 150-cavity formation in influenza neuraminidase. *Nat Commun* 2:388–394. <https://doi.org/10.1038/ncomms1390>
36. Amaro RE et al (2007) Remarkable loop flexibility in avian influenza N1 and its implications for antiviral drug design. *J Am Chem Soc* 129:7764–7765. <https://doi.org/10.1021/ja0723535>
37. Russell RJ et al (2006) The structure of H5N1 avian influenza neuraminidase suggests new opportunities for drug design. *Nature* 443:45–49. <https://doi.org/10.1038/nature05114>
38. Rudrawar S et al (2010) Novel sialic acid derivatives lock open the 150-loop of an Influenza A virus group-1 sialidase. *Nat Commun* 1:113–119. <https://doi.org/10.1038/ncomms1114>
39. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR (2011) Open babel: an open chemical toolbox. *J Cheminform* 3:33. <https://doi.org/10.1186/1758-2946-3-33>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

