Workshop on Molecular Graphics and Visual Analysis of Molecular Data (2018)
J. Byška, M. Krone, and B. Sommer (Editors)

Mol*: Towards a common library and tools for web molecular graphics

D. Sehnal[1,2,4], A. S. Rose[3], J. Koča[1,2], S. K. Burley[3,5] and S. Velankar[4]

[1]CEITEC - Central European Institute of Technology, Masaryk University Brno, Kamenice 5, 625 00 Brno-Bohunice, Czech Republic
[2]National Centre for Biomolecular Research, Faculty of Science, Kamenice 5, 625 00 Brno-Bohunice, Czech Republic
[3]RCSB Protein Data Bank, San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA 92093, USA
[4]Protein Data Bank in Europe (PDBe), European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, Cambridgeshire, CB10 1SD, UK
[5]RCSB Protein Data Bank, Institute for Quantitative Biomedicine, Rutgers, The State University of New Jersey, Piscataway, NJ 08854, USA

Abstract

Advances in experimental techniques are providing access to structures of ever more complex and larger macromolecular systems. Web-browser based visualization and analysis of macromolecular structures and associated data represents a crucial step in gaining knowledge from these data. A common library and a set of tools for working with such macromolecular data sets would streamline this step. We present a project called Mol (/'mol-star/) whose goal is to provide a common library and a set of tools for macromolecular data visualization and analysis. The project includes modules for data storage, in-memory representation, query language, UI state management, and visualization; and tools for efficient data access.*

CCS Concepts

•***Computing methodologies*** → *Rendering; Graphics systems and interfaces;* •***Applied computing*** → *Bioinformatics;*

1. Introduction

Experimental methods to determine three-dimensional (3D) structures of biomolecules are being continuously improved and produce molecular complexes from models of different resolutions that span multiple scales and can be dynamic over the course of an experiment. By combining data from complementary experimental techniques (e.g., X-ray, NMR, 3DEM, SAS, chemical crosslinking) or integrative/hybrid methods (I/HM), 3D structural models of large macromolecular systems can be determined [BKM*17, SBS*15]. Such models include large macromolecular machines, dynamic assemblies, or genome architecture. Access, visualization and analysis of these structures is a central part of structural biology and structural bioinformatics. However, as macromolecular data sets grow ever more complex and larger, it becomes challenging to create software tools to access, visualize and manipulate them. The web platform, both mobile and desktop, is becoming an increasingly popular and important tool for performing these tasks.

Embracing recent advances in web browser technology provides the means for creating scalable molecular graphics and analysis tools with near-instant access to any available data. Web-based tools are platform-independent and require little or no local software installation, which makes them available to virtually everyone in both the scientific and non-scientific community, reaching an audience larger than ever before. Moreover, these technologies (most notably JavaScript, HTML, and WebGL) and their surrounding ecosystem (including NPM, Node.js, TypeScript, GitHub) offer good support for the development of modular libraries/components. Taken together, the web provides a unique opportunity to develop a common library and a set of tools for accessing, analyzing, and visualizing macromolecular data.

Here we introduce Mol* (*/'mol-star/*), an open source project supported/maintained by RCSB Protein Data Bank (PDB) [Ber00] and Protein Data Bank in Europe (PDBe) [MAA*18] with the goal of developing a common library and tools for web-based molecular graphics and analyses. The project includes modules for data storage, in-memory representation, a query language, UI state management, visualization, and tools for efficient data access. In developing Mol* as a collection of interoperable modules, we are creating an ecosystem that will support community contributions at all levels of the framework. Building on a common library can help contributors focus on innovative visualization and algorithms. It can offer visualization researchers and algorithm developers a means to make their research more widely available and used in practice. Furthermore, collaborative development can also help with anticipating and keeping up with developments in structural biology and related fields by pooling/sharing resources.

The purpose of Mol* is to enable web-based molecular visualization and analyses by providing a common library for quickly and efficiently building tools and services for the structural biol-

DOI: 10.2312/molva.20181103

ogy/bioinformatics community. Examples include showing experimental/validation related data for macromolecular models; displaying various annotations for macromolecular models providing biological context, including SCOP, PFAM, UniProt; the creation of visually interesting, engaging and interactive educational material; or visualizing results from structural bioinformatics or computer aided drug design algorithms.

The project builds on the code and knowledge the authors gained from developing web-based molecular viewers, analysis tools and compressed file formats, including the LiteMol Suite [SDV*17], the NGL Viewer [RH15, RBV*16], PatternQuery [SPS*15], and MMTF [BRP*17, VBR*17]. Mol* is developed as an open source project and hosted on GitHub (github.com/mol-star).

2. Mol* Overview

Successful web applications require the right tradeoff between network bandwidth/latency and client-side computations. This need is especially relevant to molecular visualizations and analyses as being data heavy and/or computation heavy. Mwalongo et al. describe four criteria to classify web-based visualization tools [MKRE16]. These are:

- Infrastructure (web service, grid-based, cloud-based, local).
- Data transfer (full, progressive, simplified).
- Rendering techniques (volume, raycasting, triangle mesh).
- Optimization targets (latency, bandwidth, rendering).

The goal of Mol* is to provide a framework that spans the server and the client so that computations can be performed using the same library on the server and the client depending on the optimization target.

2.1. Core Concepts

The core philosophy of Mol* is to provide means for high degree of interoperability with existing and future solutions. This is mainly achieved by: 1) supporting existing file formats (and extending on them where appropriate or required, e.g., mmCIF and BinaryCIF), and, 2) instead of creating a scripting language that would operate only within the Mol* ecosystem (similar to PyMOL or Jmol approach), defining a set of open domain specific languages (DSL) which Mol* only provides a reference implementation for. The basic concepts that expand on this idea are outlined below.

Common data/exchange formats A small set of data formats/layouts is used for all data exchange within Mol*. The BinaryCIF data format is utilized heavily. BinaryCIF is a binary encoding of the text based mmCIF that maintains the versatility and data layout of PDBx/mmCIF [WF09], while providing the compression ratios of MMTF [BRP*17]. The initial specification of the BinaryCIF data format is available at github.com/dsehnal/BinaryCIF.

Common query language To provide a common description of a wide range of structural queries the query language is abstracted into a domain specific language (DSL) called MolQL. It is possible to transpile selection expressions used by programs such PyMol, VMD, or Jmol into MolQL to enable backwards compatibility. The language runtime can be implemented in various programming languages, as MolQL only defines what queries should be available and what the results should look like. More information about MolQL, its initial specification and reference implementation are available at github.com/MolQL.

Common state description The state description of the viewer is abstracted into a DSL as well. The advantage of this approach is that the state description could be more easily parsed by other software (including desktop applications). This language could also have the potential to evolve into a standard for describing visualizations in publications. As with MolQL, we hope that the language will be implemented in other programming languages. We want to investigate the possibility of interpreting PyMOL, Jmol, VMD, etc. scripts to describe application state.

Data access tools Modern experimental and computation methods produce large amounts of data and it is often not feasible to transfer all this data at the same time due to bandwidth/memory constraints. To overcome these limitations we want to implement possible solutions, including sending subsets of the data, downsampled data, introducing new/better compression methods.

Ability to share code between client and server Some analyses can be run directly in the web browser and utilize the computational power of the user's computer. More demanding computations can be executed on the server.

Modularity Different libraries/modules/tools of Mol* should be usable separately. This ensures that Mol* could be partially integrated into existing projects.

High performance A high emphasis will be put on the performance of individual modules.

2.2. Project Structure

The project structure is separated into three main parts: 1) core modules, 2) data access tools, and 3) application modules; all are described below.

2.2.1. Core Modules

The core modules of Mol* provide basic data structures and algorithms that more complex applications can build upon.

Mol-data A set of common data structures for representing and accessing generic collections, tables, graphs and basic algorithms that use them.

Mol-io The parser and writer module for common molecular file/data formats. It additionally provides schemas (i.e. data layout descriptions) and helper methods to consume data from web-services (provided for example in JSON or CSV format). It also includes extensive support for the CIF and BinaryCIF formats.

Mol-math A collection of useful math and geometry related methods and algorithms, including basic linear algebra methods and more advanced algorithms like Delaunay triangulation.

Mol-model Data structures for representing molecular data, including atomic models, multi-scale I/HM models and volumetric data. It also includes efficient representations for trajectories, crystallographic and biological assemblies, and symmetries.

Mol-geometry Functionality to generate geometry for various visual representations of molecular data. Both LiteMol Viewer (litemol.org) and NGL Viewer (nglviewer.org) already support the creation and rendering of many representations including molecular surfaces (see Figure 1). Support for serialization of the data using CIF/BinaryCIF.

MolQL An implementation of the MolQL language specification. It includes transpilers with the ability to compile and execute selection expressions supported by PyMOL, VMD, or Jmol. An example of the expressiveness of MolQL can be found in Figure 2 which shows all clusters of alanine and lysine residues.

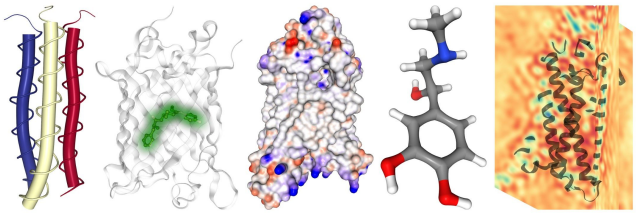

Figure 1: *Various molecular representations available in the NGL Viewer. From left to right: Tube and rope representations of a trimer of transmembrane helices; glowing particle effect around the chromophore in a green fluorescent protein (PDB entry 1GFL); electrostatic potential mapped on a molecular surface of rhodopsin (PDB entry 3PQR); HyperBalls [CVT*11] representation of an adrenaline molecule; two volume slices of the X-ray electron density of a rhodopsin structure (PDB entry 3PQR).*

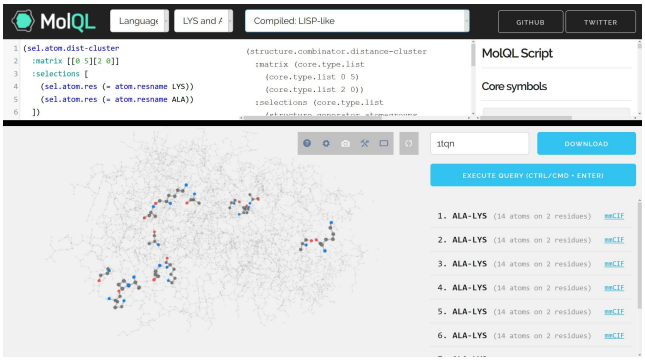

Figure 2: *MolQL Explorer window showing all pairs of ALA and LYS residues that are between 2 and 5 angstroms apart. The corresponding MolQL expression is* `(sel.atom.dist-cluster :matrix [[0 5] [2 0]] :selections [ (sel.atom.res (= atom.resname LYS)) (sel.atom.res (= atom.resname ALA))])`. *See molql.org for more examples.*

2.2.2. Data Access Tools

The philosophy of Mol* data access tools is that they can integrate/take advantage of the existing infrastructure where appropriate (e.g. "use 'text based' search to determine which structures to perform structural queries on using the ModelServer"). The response format of choice for the tools is JSON or CIF for smaller payloads and BinaryCIF for larger datasets.

ModelServer The ModelServer provides access to molecular 1D, 2D, and 3D (sub-)structure models of molecules. Sub-structures are described by the MolQL language. It has the ability to include additional data to mmCIF "on the fly", e.g. integrate primary PDB archival data from Chemical Component Dictionary (CCD, [WSF*15]), Protonation Variants Companion Dictionary (PVCD, [WSF*15]) and Biologically Interesting molecule Reference Dictionary (BIRD, [Ban18]). The server will include the support for trajectory data (i.e., the ability to directly access a specified range/sampling of snapshots; MDsrv at github.com/arose/mdsrv provides a basic implementation of this [TGGHR17]), and the ability to specify the shape/contents of the response using a GraphQL-like language [Fac18]. The CoordinateServer available at github.com/dsehnal/CoordinateServer is a partial implementation of this concept [SDV*17] (see Figure 3).

VolumeServer The VolumeServer provides access to volumetric data including density maps (for instance, from X-ray crystallography or cryo-electron microscopy experiments), spatial distribution data, output from electrostatic calculations, and high resolution light microscopy data. It works by utilizing adaptive downsampling (similar to how Google Earth works). The DensityServer available at github.com/dsehnal/DensityServer is an implementation of this concept and provides access to density maps from X-ray crystallography and cryo-electron microscopy experiments (see Figure 3).

AnnotationServer The AnnotationServer provides common data layout (derived from the mmCIF schema when possible) for various annotations that are relatable/mappable to molecular structures. It allows to aggregate multiple annotations and includes 3rd party annotations and an API for user-generated annotations. The server also allows to query annotations using MolQL, for example, "provide annotations relevant only for the specified atomic set", or "give me all atomic sets with the given annotation".

GeometryServer The GeometryServer allows streaming geometry related data by providing an interface for the Mol-geometry library.

ToolServer The ToolServer provides an API for unified access to various tools that provide data for web-based visualization. Possible applications are, for example, controlling modeling soft-

ware from a web-based viewer and stream the results on the fly or enabling access to different macromolecular/chemical simulation tools from a unified web-based user interface.

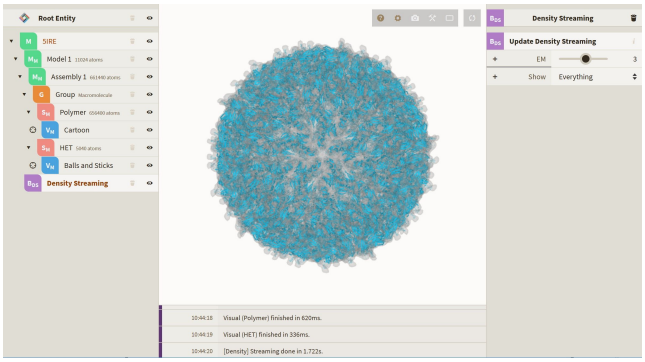

Figure 3: *Zika virus assembly and Cryo-EM data. Shows the Zika virus (PDB id 5IRE; assembly 1), including the underlying Cryo-EM data (EMD-8116) in LiteMol Viewer using the CoordinateServer and DensityServer. The data required for the transfer using the data delivery services is 1.03MB vs. 1620MB when downloading the raw data. Interactive version available at v.litemol.org/?example=zika-cryo-em.*

2.2.3. Application Modules

The user interface (UI) and user experience (UX) are an integral part of any web application. The goal of the Mol* project is not to solve UI/UX for all web-based applications dealing with molecular data. However, to help with designing good UI/UX several modules are provided:

Mol-task A module for abstracting computation trees which are built from tasks an application wants to perform. It includes built-in progress tracking and cancellation support which are essential for building responsive applications and providing a user friendly experience.

Mol-state A module for managing the state of an application over time as a graph. It includes native support for state saving and the ability to export the state as a set of DSL expressions. This will help backwards compatibility and interoperability with other tools, including desktop software.

Mol-gl A lightweight WebGL wrapper.

Mol-viewer A reference viewer that also serves as a showcase Mol* functionality.

3. Related Work

Comprehensive libraries to handle biological data exist in many languages, including projects like BioPython [CAC*09], BioJava [PYB*12], and BioPerl [Sta02], but no such library is available for the web platform (i.e., written in JavaScript). The BioJS [GGS*13] project provides an infrastructure, guidelines, and tools for creating reusable web component for representing biological data, however, it does not include a low-level library to build these components from.

The scientific and visualization communities in general are turning towards the web platform to enable on-demand interactive representations of the data. For example, the CesiumJS open source project offers a geospatial 3D mapping platform for creating virtual globes and visualizing visualizing dynamic data (cesiumjs.org). Alternatively, the Galaxy is an open source, web-based platform for data intensive biomedical research [ABvdB*16] (usegalaxy.org). It includes web-based visualization components that offer access to genomic data and analyses.

4. Conclusion and Outlook

We introduced Mol* and provided a high-level description of its modules. The initial prototype of Mol* and the working prototype of MolQL (available online for testing, molql.org) show the feasibility of creating and working with such a highly modular library. Moreover, the BinaryCIF format and two data delivery services, the CoordinateServer and the DensityServer (as components of the LiteMol Suite available at litemol.org; see Figure 3), are already in production use at PDBe (pdbe.org). The DensityServer and BinaryCIF is also supported in Jmol [Han10]. Likewise, the MMTF format and the NGL Viewer are in production use at RCSB PDB (rcsb.org).

By providing the means for mapping various data sources (for example RESTful services provided by wwPDB members) Mol* leverages/integrates existing scientific data and makes it easier to visualize and analyze molecular structures in the context of existing annotations.

To create a stable foundation for long term success, Mol* builds on proven technologies including mmCIF for an expressive molecular data model [WF09], TypeScript for the development of large (web) applications [Mic18], WebGL for hardware accelerated 3D rendering [Khr18] and the standards of the open web platform in general [Wor18].

5. Availability

Mol* is available under the MIT license, a permissive Open Source license, to facilitate code sharing and collaboration. The code is available on GitHub at github.com/mol-star.

6. Acknowledgments

This project was supported in part by the RCSB PDB which is jointly funded by the NSF, the NIH, and the US DOE (NSF DBI-1338415; PI: SK Burley). This research has been financially supported by the ELIXIR-EXCELERATE project, grant agreement no. 676559; ELIXIR CZ research infrastructure project (MEYS grant no. LM2015047); European Regional Development Fund-Project ELIXIR-CZ (no. CZ.02.1.01/0.0/0.0/16_013/0001777); Wellcome Trust (104948) and EMBL-EBI funding to PDBe.

References

[ABvdB*16] AFGAN E., BAKER D., VAN DEN BEEK M., BLANKENBERG D., BOUVIER D., ČECH M., CHILTON J., CLEMENTS D.,

CORAOR N., EBERHARD C., GRÜNING B., GUERLER A., HILLMAN-JACKSON J., VON KUSTER G., RASCHE E., SORANZO N., TURAGA N., TAYLOR J., NEKRUTENKO A., GOECKS J.: The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Research 44*, W1 (jul 2016), W3–W10. URL: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkw343, doi:10.1093/nar/gkw343. 4

[Ban18] BANK W. P. D.: The biologically interesting molecule reference dictionary (bird), 2018. URL: https://www.wwpdb.org/data/bird. 3

[Ber00] BERMAN H. M.: The Protein Data Bank. *Nucleic Acids Research 28*, 1 (jan 2000), 235–242. doi:10.1093/nar/28.1.235. 1

[BKM*17] BURLEY S. K., KURISU G., MARKLEY J. L., NAKAMURA H., VELANKAR S., BERMAN H. M., SALI A., SCHWEDE T., TREWHELLA J.: PDB-Dev: a Prototype System for Depositing Integrative/Hybrid Structural Models. *Structure 25*, 9 (sep 2017), 1317–1318. doi:10.1016/j.str.2017.08.001. 1

[BRP*17] BRADLEY A. R., ROSE A. S., PAVELKA A., VALASATAVA Y., DUARTE J. M., PRLIĆ A., ROSE P. W.: MMTF-An efficient file format for the transmission, visualization, and analysis of macromolecular structures. *PLOS Computational Biology 13*, 6 (jun 2017), e1005575. doi:10.1371/journal.pcbi.1005575. 2

[CAC*09] COCK P. J. A., ANTAO T., CHANG J. T., CHAPMAN B. A., COX C. J., DALKE A., FRIEDBERG I., HAMELRYCK T., KAUFF F., WILCZYNSKI B., DE HOON M. J. L.: Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics 25*, 11 (jun 2009), 1422–1423. URL: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btp163, doi:10.1093/bioinformatics/btp163. 4

[CVT*11] CHAVENT M., VANEL A., TEK A., LEVY B., ROBERT S., RAFFIN B., BAADEN M.: GPU-accelerated atom and dynamic bond visualization using hyperballs: A unified algorithm for balls, sticks, and hyperboloids. *Journal of Computational Chemistry 32*, 13 (oct 2011), 2924–2935. doi:10.1002/jcc.21861. 3

[Fac18] FACEBOOK: Graphql - a query language for your api, 2018. URL: https://graphql.org/. 3

[GGS*13] GOMEZ J., GARCIA L. J., SALAZAR G. A., VILLAVECES J., GORE S., GARCIA A., MARTIN M. J., LAUNAY G., ALCANTARA R., DEL-TORO N., DUMOUSSEAU M., ORCHARD S., VELANKAR S., HERMJAKOB H., ZONG C., PING P., CORPAS M., JIMENEZ R. C.: BioJS: an open source JavaScript framework for biological data visualization. *Bioinformatics 29*, 8 (apr 2013), 1103–1104. URL: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btt100, doi:10.1093/bioinformatics/btt100. 4

[Han10] HANSON R. M.: Jmol - a paradigm shift in crystallographic visualization. *Journal of Applied Crystallography 43*, 5 (oct 2010), 1250–1260. doi:10.1107/S0021889810030256. 4

[Khr18] KHRONOS GROUP: Webgl - opengl es for the web, 2018. URL: https://www.khronos.org/webgl/. 4

[MAA*18] MIR S., ALHROUB Y., ANYANGO S., ARMSTRONG D. R., BERRISFORD J. M., CLARK A. R., CONROY M. J., DANA J. M., DESHPANDE M., GUPTA D., GUTMANAS A., HASLAM P., MAK L., MUKHOPADHYAY A., NADZIRIN N., PAYSAN-LAFOSSE T., SEHNAL D., SEN S., SMART O. S., VARADI M., KLEYWEGT G. J., VELANKAR S.: PDBe: towards reusable data delivery infrastructure at protein data bank in Europe. *Nucleic Acids Research 46*, D1 (jan 2018), D486–D492. doi:10.1093/nar/gkx1070. 1

[Mic18] MICROSOFT: Typescript - javascript that scales, 2018. URL: https://www.typescriptlang.org/. 4

[MKRE16] MWALONGO F., KRONE M., REINA G., ERTL T.: State-of-the-Art Report in Web-based Visualization. *Computer Graphics Forum 35*, 3 (jun 2016), 553–575. doi:10.1111/cgf.12929. 2

[PYB*12] PRLIC A., YATES A., BLIVEN S. E., ROSE P. W., JACOBSEN J., TROSHIN P. V., CHAPMAN M., GAO J., KOH C. H., FOISY S., HOLLAND R., RIMSA G., HEUER M. L., BRANDSTATTER-MULLER H., BOURNE P. E., WILLIS S.: BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics 28*, 20 (oct 2012), 2693–2695. URL: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bts494, doi:10.1093/bioinformatics/bts494. 4

[RBV*16] ROSE A. S., BRADLEY A. R., VALASATAVA Y., DUARTE J. M., PRLIĆ A., ROSE P. W.: Web-based molecular graphics for large complexes. In *Proceedings of the 21st International Conference on Web3D Technology - Web3D '16* (New York, New York, USA, 2016), ACM Press, pp. 185–186. doi:10.1145/2945292.2945324. 2

[RH15] ROSE A. S., HILDEBRAND P. W.: NGL Viewer: a web application for molecular visualization. *Nucleic Acids Research 43*, W1 (jul 2015), W576–W579. doi:10.1093/nar/gkv402. 2

[SBS*15] SALI A., BERMAN H. M., SCHWEDE T., TREWHELLA J., KLEYWEGT G., BURLEY S. K., MARKLEY J., NAKAMURA H., ADAMS P., BONVIN A. M., CHIU W., PERARO M. D., DI MAIO F., FERRIN T. E., GRÜNEWALD K., GUTMANAS A., HENDERSON R., HUMMER G., IWASAKI K., JOHNSON G., LAWSON C. L., MEILER J., MARTI-RENOM M. A., MONTELIONE G. T., NILGES M., NUSSINOV R., PATWARDHAN A., RAPPSILBER J., READ R. J., SAIBIL H., SCHRÖDER G. F., SCHWIETERS C. D., SEIDEL C. A., SVERGUN D., TOPF M., ULRICH E. L., VELANKAR S., WESTBROOK J. D.: Outcome of the First wwPDB Hybrid/Integrative Methods Task Force Workshop. *Structure 23*, 7 (2015), 1156–1167. doi:10.1016/j.str.2015.05.013. 1

[SDV*17] SEHNAL D., DESHPANDE M., VAŘEKOVÁ R. S., MIR S., BERKA K., MIDLIK A., PRAVDA L., VELANKAR S., KOČA J.: LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data. *Nature Methods 14*, 12 (nov 2017), 1121–1122. doi:10.1038/nmeth.4499. 2, 3

[SPS*15] SEHNAL D., PRAVDA L., SVOBODOVÁ VAŘEKOVÁ R., IONESCU C.-M., KOČA J.: PatternQuery: web application for fast detection of biomacromolecular structural patterns in the entire Protein Data Bank. *Nucleic Acids Research 43*, W1 (2015), W383–W388. doi:10.1093/nar/gkv561. 2

[Sta02] STAJICH J. E.: The Bioperl Toolkit: Perl Modules for the Life Sciences. *Genome Research 12*, 10 (oct 2002), 1611–1618. URL: http://www.genome.org/cgi/doi/10.1101/gr.361602, doi:10.1101/gr.361602. 4

[TGGHR17] TIEMANN J. K. S., GUIXÀ-GONZÁLEZ R., HILDEBRAND P. W., ROSE A. S.: MDsrv: viewing and sharing molecular dynamics simulations on the web. *Nature Methods 14*, 12 (nov 2017), 1123–1124. doi:10.1038/nmeth.4497. 3

[VBR*17] VALASATAVA Y., BRADLEY A. R., ROSE A. S., DUARTE J. M., PRLIĆ A., ROSE P. W.: Towards an efficient compression of 3D coordinates of macromolecular structures. *Plos One 12*, 3 (2017), e0174846. doi:10.1371/journal.pone.0174846. 2

[WF09] WESTBROOK J., FITZGERALD P.: The PDB format, mmCIF formats, and other data formats. In *Structural Bioinformatics*, 2 ed. 2009, ch. 10, pp. 271–291. doi:10.1002/0471721204.ch8. 2, 4

[Wor18] WORLD WIDE WEB CONSORTIUM: Standards that define an open web platform, 2018. URL: https://www.w3.org/standards. 4

[WSF*15] WESTBROOK J. D., SHAO C., FENG Z., ZHURAVLEVA M., VELANKAR S., YOUNG J.: The chemical component dictionary: complete descriptions of constituent molecules in experimentally determined 3D macromolecules in the Protein Data Bank. *Bioinformatics 31*, 8 (apr 2015), 1274–1278. doi:10.1093/bioinformatics/btu789. 3