1

VIEW-SC+: a web tool for integrating steric clash prediction, 3D visualization and information mapping in PDB structures

Jiawei Zhao[1,*]

[1] Department of Biology, Lund University, Lund, Sweden

Supervisor: Mauno Vihinen

* Corresponding author

E-mail: ji8842zh-s@student.lu.se (JZ)

1

Abstract

The Protein Data Bank is by far the most important database for the three-dimensional structural data of proteins predominantly based on crystallography. PON-SC, a PDB identification code-based program with promising accuracy, was developed by Vihinen Lab for predicting steric clashes caused by amino acid (AA) substitutions. However, 3D visualization for AA substitutions, along with data mapping and integration to relevant UniProt accessions, are no longer available at PON-SC. Therefore, we developed VIEW-SC+, a novel web tool based on PON-SC, JSmol, pdb2uniprot and APIs on several public protein databases. VIEW-SC+ consists of three major parts. The first part provides steric clash prediction for AA substitutions based on PDB identification code (PDB ID) or PDB files, together with PDB file download and residue-level variant mapping on equivalent UniProt accessions for individual substitutions. The second part offers users with interactive 3D visualization for general PDB structures and individual AAs in both original and variants in JSmol. The third part realizes protein data integration for both PDB and UniProt accessions from the GraphQL-based API of RCSB PDB. VIEW-SC+ is freely accessible to all at: https://structure-next.med.lu.se/view-scp.

Introduction

The Protein Data Bank (PDB) is the single global archive of experimentally determined three-dimensional (3D) structure data of biological macromolecules, especially proteins which were mainly determined using macromolecular crystallography (MX). Today, the PDB is universally regarded as a core data resource essential for understanding the functional roles that macromolecules play in biology and medicine [1].

If an AA substitution in three-dimensional protein structure represented by a PDB ID renders unsuitable local or global clashes in crystal structure, harmful functional effects are often followed [2]. Therefore, PON-SC, a program for identifying steric clashes caused by AA substitutions was developed by Protein Structure and Bioinformatics Group, Faculty of Medicine, Lund University [2]. PON-SC is currently available at http://structure.bmc.lu.se/PON-SC.

PON-SC utilizes backbone-dependent rotamer library [3] for testing the space of potential side chain conformations, BioPython package for parsing the input file in PDB format, STRIDE [4] for calculating φ and ψ torsion angles of AA backbones and accessibility of the side chains and KDTree algorithm from scikit-learn package [5] for preparing the structures for rotamer tests. The rotamer tests examine whether any of the common rotamers can be fitted into the protein structure [2]. PON-SC was found to have accuracy of 0.71 over five test datasets with specificity being observed higher than sensitivity, which is superior to earlier prediction methods [2].

UniProt is a freely accessible database of protein sequence and functional information, many entries being derived from genome sequencing projects. It contains a large amount of information about the biological function of proteins derived from the

research literature. Variations caused by AA substitutions in protein sequences, are based on reference sequences which are predominantly identified by UniProt accessions, e.g., D614G on the spike protein sequence P0DTC2 of SARS-CoV-2. Therefore, functional annotations of AA substitutions on a PDB ID would be possible if residue-level mapping to its equivalent UniProt accession is performed.

Nonetheless, 3D visualization for individual AAs and the equivalent substitutions in PDB structure, along with data mapping and integration to relevant UniProt accessions, are unavailable at PON-SC. Consequently, it is necessary to develop a novel web tool which integrates the features mentioned above.

Materials and methods

JSmol: an open-source HTML5 viewer for chemical structures in 3D

In order to realize three-dimensional visualization for general molecular structure and AA substitutions given PDB IDs on our web application, we assessed several relevant JavaScript libraries. JSmol, the HTML5 modality of Jmol was selected due to its highly flexible commands and comprehensive options in comparison to 3Dmol [6] and RasMol [7].

Jmol is reputed to be a highly functional viewer of three-dimensional molecular structures in biochemistry written in Java. It is cross-platform, free and open-source. JSmol is developed as the web modality of Jmol based on HTML5 Canvas, preserving all functionalities of Jmol **[8]**.

RCSB PDB Data API and The Proteins API

RCSB PDB (Research Collaboratory for Structural Bioinformatics PDB) is a member of the wwPDB. It operates the US data center for the global PDB archive, and makes PDB data available at no charge to all data consumers without limitations on usage [9]. The RCSB PDB offers two ways to access data through application programming interfaces (APIs): REST-based API and GraphQL-based API. In order to fetch data with less redundancy, we selected the GraphQl-based API and assigned a sub-route on VIEW-SC+ for it.

The RCSB PDB route of VIEW-SC+ utilizes PDB IDs as index entry to the RCSB PDB Data API. RCSB PDB identifier and RCSB polymer entity identifiers are subsequently identified by index entry. Thus, each of the polymer identifiers is paired with PDB identifier as entry into the mapping query from PDB IDs to UniProt accession. Subsequently, each of the mapped UniProt accessions is utilized as entry into the query for fetching vital information about the UniProt accession.

The Proteins REST API is served by the European Bioinformatics Institute (EMBL-EBI), and provides searching and programmatic access to protein and associated genomics data such as curated protein sequence positional annotations from UniProtKB, as well as mapped variation and proteomics data from large scale data sources (LSS). Using the coordinates service, researchers are able to retrieve the genomic sequence coordinates for proteins in UniProtKB [10].

Information on variations on protein sequences, e.g., consequences and PubMed ID for papers of AA substitutions, are continuously updated by the Proteins REST API database. However, the notations for AA substitutions are based on UniProt accessions. Thus, it is necessary to perform residue-level mapping from AAs substitutions on PDB structure to its equivalent UniProt accessions. To achieve this, we utilized pdb2uniprot [11] as a transitional tool. Therefore, each of the positions

where AA substitutions occur in PDB ID queries from users is examined for possible equivalent variations in UniProt sequences.

React.js, Node.js and Typescript

In order to run PON-SC programs written in python on the server of VIEW-SC+, it is necessary to set up a back-end process in the daemon mode. Therefore, we selected Node.js, an asynchronous event-driven JavaScript runtime designed to build scalable network applications for this task [12]. Node.js is an evented I/O framework for the V8 JavaScript engine. It is intended for writing scalable network programs such as web servers. Node.js is similar in purpose to Twisted for Python, Perl Object Environment for Perl, and EventMachine for Ruby. Unlike most JavaScript, it is not executed in a web browser, but it is rather related to server-side JavaScript [13]. We selected Express.js, a fast, unopinionated, minimalist web framework [14] for server side request handling instead of the original http and https libraries of node.js due to better performance in speed and routing. JavaScript packages used for the back-end route of VIEW-SC+ at https://structure-next.med.lu.se/pon-scp are listed in Table 1.

Table 1. All JavaScript libraries used for server-side Node.js scripting in this project

Source page	Library	Version
https://yarnpkg.com/package/	axios	0.21.0
https://yarnpkg.com/package/	body-parser	1.19.0
https://yarnpkg.com/package/	cors	2.8.5
https://yarnpkg.com/package/	express	4.17.1
https://yarnpkg.com/package/	multer	1.4.2
https://nodejs.org/en/download/	node	v14.16.0
https://yarnpkg.com/package/	nodemailer	6.4.18
https://yarnpkg.com/package/	python-shell	2.0.3

In addition to server-side Node.js programs, development for client-side web application is another crucial part for VIEW-SC+. React.js, a popular JavaScript library

for building user interfaces [15], was selected by us for this task. React.js is declarative, component-based and easy to debug and develop new features. Another notable feature of React.js is the use of a virtual Document Object Model (virtual DOM), which significantly accelerates DOM rendering. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently [16]. JavaScript packages used for the front-end route of VIEW-SC+ at https://structure-next.med.lu.se/view-scp are listed in Table 2.

Table 2. All JavaScript libraries used for client-side React.js scripting in this project

Source page	Library	Version
https://yarnpkg.com/package/	@apollo/client	3.3.6
https://yarnpkg.com/package/	@apollo/react-hooks	4.0.0
https://yarnpkg.com/package/	axios	0.21.0
https://yarnpkg.com/package/	bootstrap	4.5.3
https://yarnpkg.com/package/	bootstrap-social	5.1.1
https://yarnpkg.com/package/	font-awesome	4.7.0
https://yarnpkg.com/package/	graphql	15.4.0
https://yarnpkg.com/package/	graphql-tag	2.11.0
https://yarnpkg.com/package/	jquery	3.5.1
https://yarnpkg.com/package/	js-file-download	0.4.12
https://yarnpkg.com/package/	react	17.0.1
https://yarnpkg.com/package/	react-dom	17.0.1
https://yarnpkg.com/package/	react-redux	7.2.2
https://yarnpkg.com/package/	react-router-dom	5.2.0
https://yarnpkg.com/package/	react-scripts	4.0.0
https://yarnpkg.com/package/	reactstrap	8.7.1
https://yarnpkg.com/package/	redux	4.0.5
https://yarnpkg.com/package/	redux-logger	3.0.6
https://yarnpkg.com/package/	redux-persist	6.0.0
https://yarnpkg.com/package/	redux-thunk	2.3.0

Nevertheless, since JavaScript is a dynamically typed scripting language, e.g., a variable initially bound to a number may be reassigned to a string, it is prone to unobserved errors during the development stage. Consequently, we utilized TypeScript as the scripting language for both server-side Node.js and client-side

React.js programs during the development stage. In order to reduce errors in JavaScript code, we selected TypeScript at the development stage. TypeScript is an open-source programming language developed by Microsoft by adding static type definitions to JavaScript [17]. Due its type inference and type checking, TypeScript enables more secured scripting in comparison to plain JavaScript. Type definition and JavaScript packages used during the development stage for VIEW-SC+ are listed in Table 3.

Table 3. All JavaScript or TypeScript libraries used during the development stage in this project

Source page	Library	Version
https://yarnpkg.com/package/	@graphql-codegen/cli	1.19.4
https://yarnpkg.com/package/	@graphql-codegen/typescript	1.19.0
https://yarnpkg.com/package/	@graphql-codegen/typescript-operations	1.17.12
https://yarnpkg.com/package/	@graphql-codegen/typescript-react-apollo	2.2.1
https://yarnpkg.com/package/	@testing-library/jest-dom	5.11.4
https://yarnpkg.com/package/	@testing-library/react	11.1.0
https://yarnpkg.com/package/	@testing-library/user-event	12.1.0
https://yarnpkg.com/package/	@types/body-parser	1.19.0
https://yarnpkg.com/package/	@types/cors	2.8.9
https://yarnpkg.com/package/	@types/express	4.17.9
https://yarnpkg.com/package/	@types/jest	26.0.15
https://yarnpkg.com/package/	@types/jquery	3.5.4
https://yarnpkg.com/package/	@types/lodash	4.14.168
https://yarnpkg.com/package/	@types/multer	1.4.5
https://yarnpkg.com/package/	@types/node	12.0.0
https://yarnpkg.com/package/	@types/nodemailer	6.4.0
https://yarnpkg.com/package/	@types/react	16.9.53
https://yarnpkg.com/package/	@types/react-dom	16.9.8
https://yarnpkg.com/package/	@types/react-redux	7.1.11
https://yarnpkg.com/package/	@types/react-router-dom	5.1.6
https://yarnpkg.com/package/	@types/redux-logger	3.0.8
https://yarnpkg.com/package/	concurrently	5.3.0
https://yarnpkg.com/package/	nodemon	2.0.6
https://yarnpkg.com/package/	serve	11.3.2
https://yarnpkg.com/package/	typescript	4.0.5
https://yarnpkg.com/package/	web-vitals	0.2.4

Description and new features

VIEW-SC+ is available at https://structure-next.med.lu.se/view-scp. Addtionally, source code of server-side Node.js and client-side React.js scripts is available at https://github.com/Orthologues/BINP39-Visualization-Project.

PON-SC prediction and its extensions

PON-SC prediction for steric clashes caused by AA substitutions is available at https://structure-next.med.lu.se/view-scp/home. Users are able to provide a query with one-letter code AA substitutions based on either PDB IDs or a PDB format file.

PON-SC prediction for steric clashes caused by AA substitutions is available at https://structure-next.med.lu.se/view-scp/home. Users are able to provide a query with one-letter code AA substitutions based on either PDB IDs or a PDB format file. Following a PON-SC query from client side, An https request is sent to a sub-route of https://structure-next.med.lu.se/pon-scp, which is the base URL to the Node.js server-side programs of VIEW-SC+. Thus, PON-SC programs are called as a python sub-process, whose predictions are recorded by server-side Node.js scripts and then responded back to the client-side web application.

Editable PON-SC query history of every user is cached in local disk storage at client-side. Each of the queried PDB IDs and PDB files is listed at the leftmost column on the web page. Once a PDB ID or a PDB format filename on the list is clicked, all its affiliated AA substitutions are classified into “good” (do not cause steric clash) ones and “bad” (cause steric clash) according to PON-SC prediction records. Subsequently, the client-side web application maps the “good” AA substitutions and the “bad” substitutions respectively to two columns on the web page. Each of the mapped substitution displays a link to download the descriptive PDB format file for the post-

substitution residue at its position. Downloadable PDB format files are all generated and transferred by the server-side Node.js and PON-SC programs on request of users. User interface for PON-SC prediction and PDB file download is designed as Fig. 1.

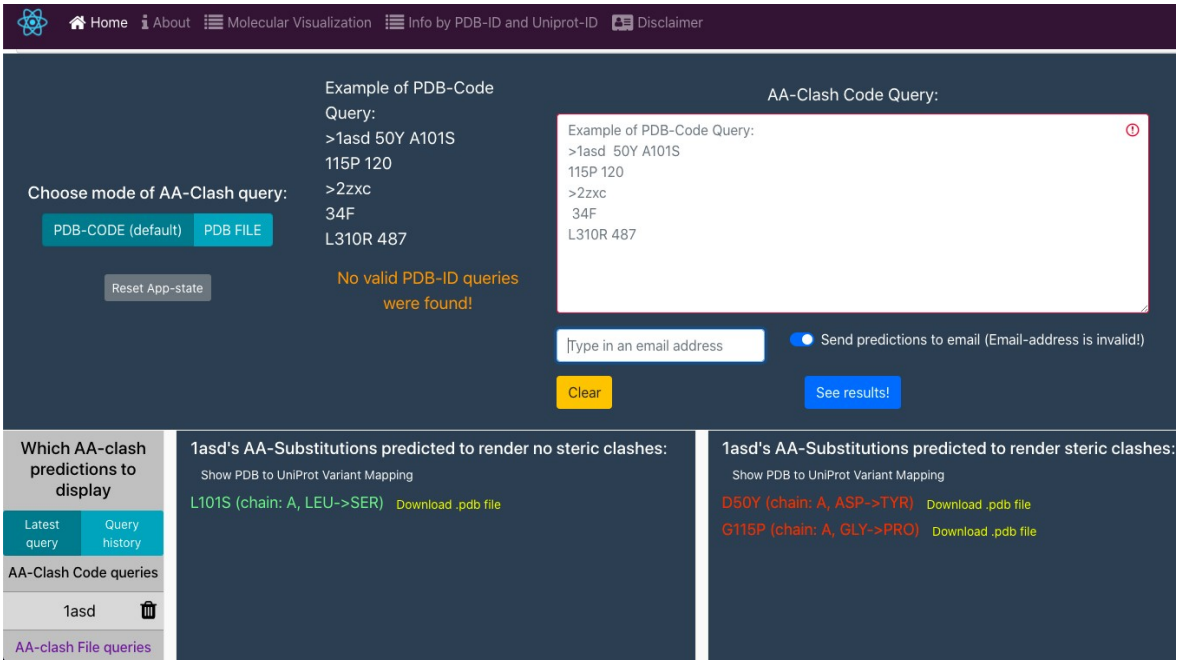

Fig 1. Example of PON-SC prediction and PDB file download

Additionally, once a PDB ID on the display list is selected, all positions of its affiliated AAs are examined for matches with possible equivalent variations on its corresponding UniProt accessions by a two-step mapping (pdb2uniprot [11] and Proteins API [10]). Once a match is found, variation data from Proteins API would be displayed on the web page. For example, if position 614 of PDB code *7jwy* (spike glycoprotein of SARS-CoV-2) is queried and then clicked by a user, position 614 of *7jwy* on its side chain A, B and C would be mapped to the 614th AA of UniProt accession *P0DTC2* by pdb2uniprot, then being matched with the D614G variation of SARS-CoV2 by Proteins API. The flowchart and user interface of residue-level PDB to UniProt variant mapping are shown in Fig. 2 and Fig. 3, respectively.

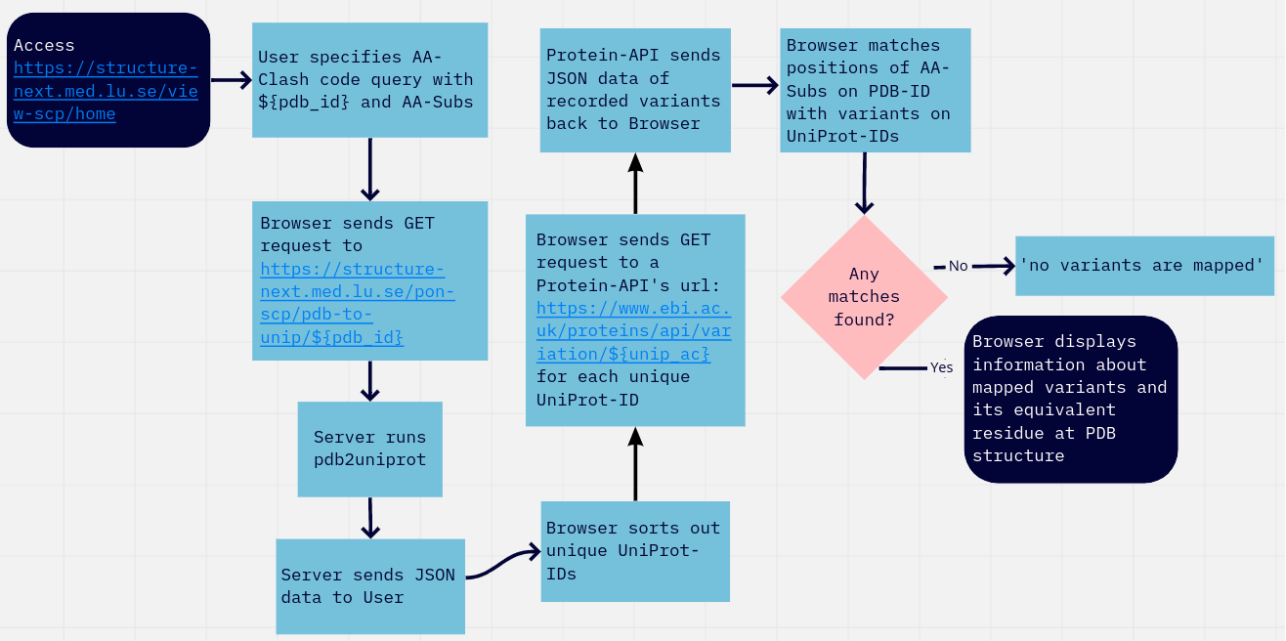

Fig 2. Pipeline of residue-level PDB to UniProt variant mapping

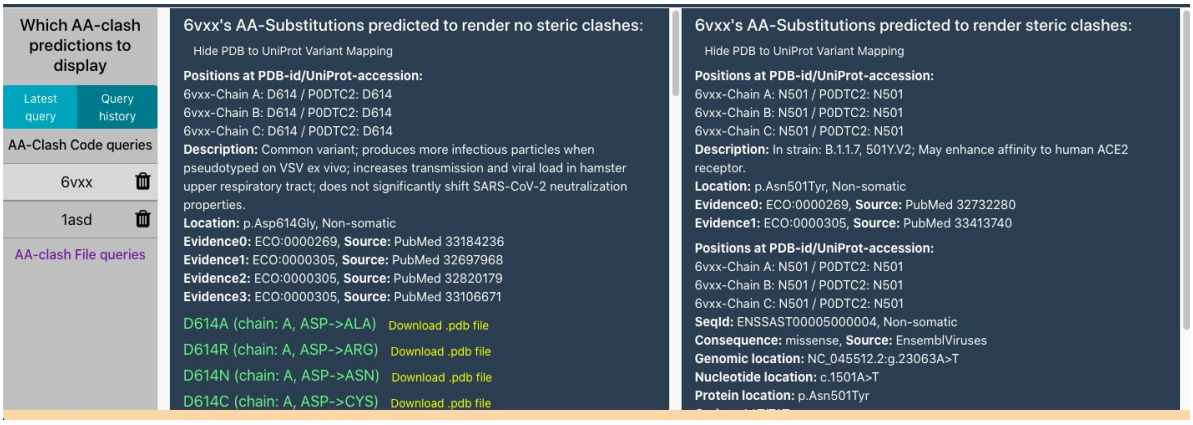

Fig 3. Example of PON-SC query to UniProt variant mapping

3D molecular visualization in JSmol

Three-dimensional molecular visualization in JSmol [8] is available at https://structure-next.med.lu.se/view-scp/mol-visualization. The leftmost column on the web page displays a list of all the queried PDB IDs for PON-SC prediction. In addition, it is possible to visualize extra queries combined with PDB IDs and AA substitutions which are independent of PON-SC queries. For each of the PDB IDs from PON-SC

queries, AA substitutions are classified into “good” and “bad” ones and then listed separately on two columns as mentioned above. By inserting Jmol commands on request of clicking events by users, multiple interactive visualization options are enabled in this web page.

Each of the AA substitution queries can be zoomed in, displaying AA in wild type or in variant type. For a zoomed-in AA, it is possible to display only atoms whose Euclidean distance to the AA is smaller than a specified numerical value of ångström. Additionally, it is possible to highlight atoms of a certain side chain or selected AAs, as well as display only the alpha-carbon backbone or the wireframe in structure. As in Fig. 4, the zoomed-in AA is the variant of a hypothetical variation at position 614 on side-chain A of PDB ID 6vxx. All the atoms of the zoomed-in AA are colored in red, while all the atoms that do not belong to the zoomed-in AA are colored in white.

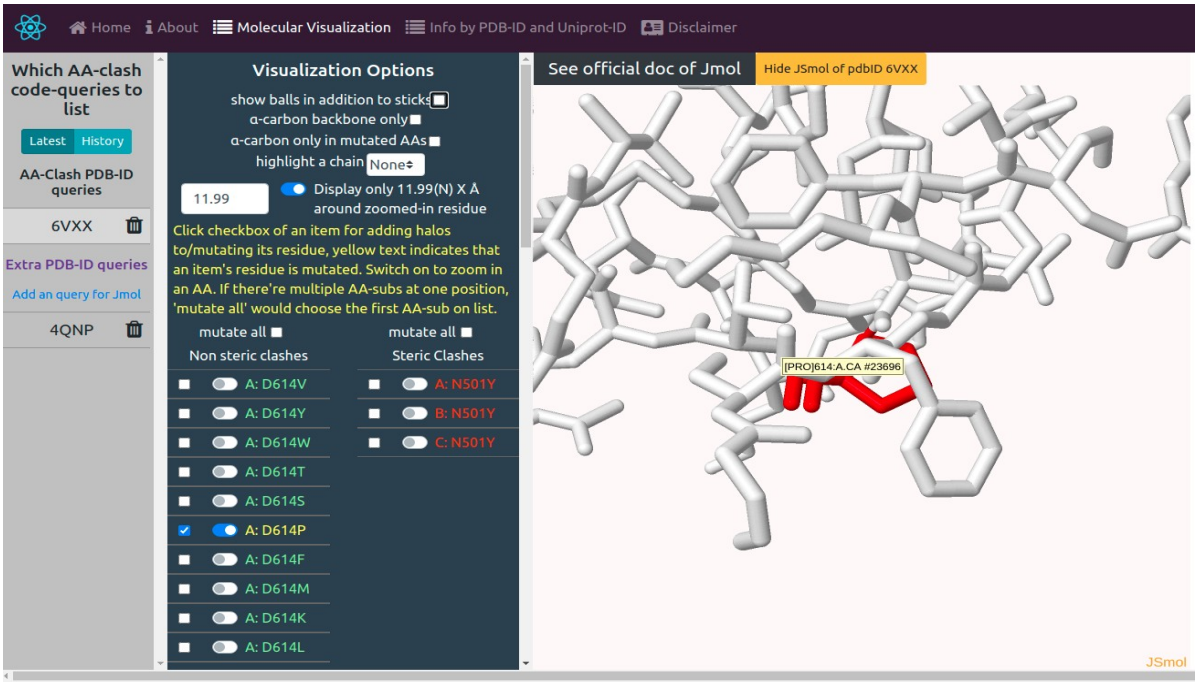

Fig 4. Example of 3D molecular visualization in JSmol

Summary for proteins based on RCSB PDB Data API

Information summary for PDB ID queries is available at https://structure-next.med.lu.se/view-scp/rcsb-info based on the GraphQL-based API at RCSB PDB [9]. The leftmost column on the web page serves the same task as the one on https://structure-next.med.lu.se/view-scp/mol-visualization. Additionally, it is also possible to submit extra PDB IDs which are independent of PON-SC queries.

Information summary of a PDB ID consists of its keywords, description, Pubmed ID, sample sequence length of its affiliated polymer entities, source organisms and host organisms etc. Additionally, each of the polymers is mapped to its most relevant UniProt accession and UniProt entry name. Both canonical PDBx sequences of the polymer entities and sequences of the mapped UniProt accessions are available for AA search by position. Extra links to PDB ID entry [18, 19, 20, 21] or mapped UniProt accession entry [22, 23, 24, 25, 26] at multiple online databases in proteomics for the queried PDB ID are also display on the web page. As in Fig. 5, RCSB-PDB information for PDB ID 4qnp is summarized.

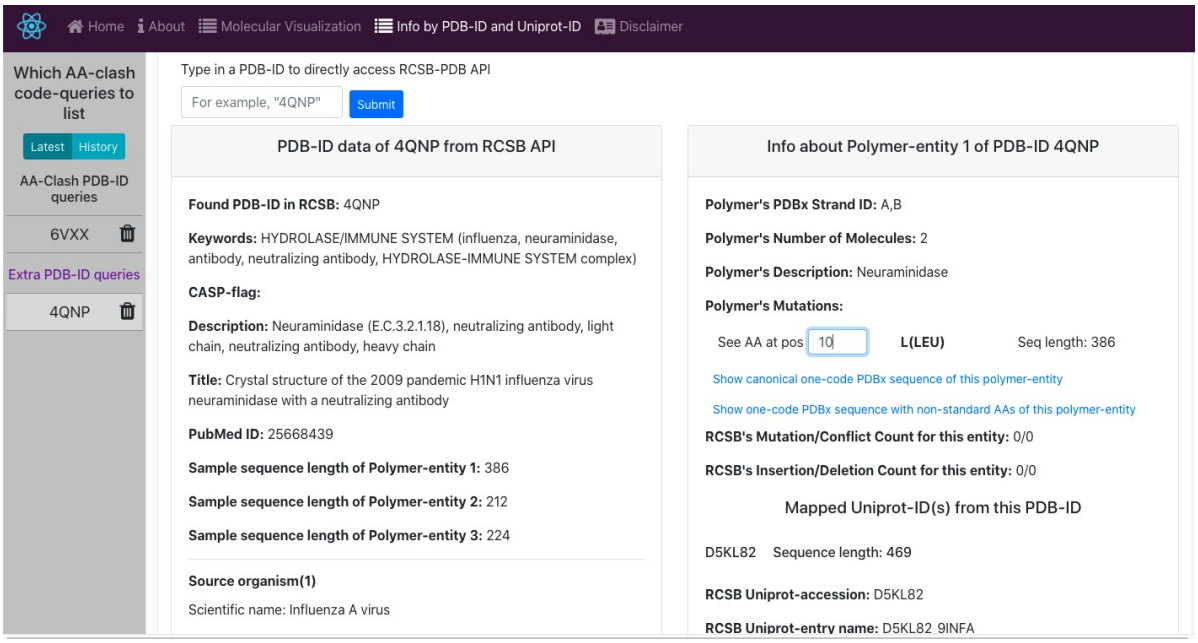

Fig 5. Example of Summary for proteins based on RCSB PDB Data API

Discussion

Potential improvements to VIEW-SC+

Several novel features are considerable for version 2.0 of VIEW-SC+.

First, in addition to a download link to the PDB file of each of the individual variant AAs, an alternative option to download the PDB file of the entire variant of its PDB ID could be added. B-Value of the atoms of variant AAs shall be set as 0 due to lack of experimental data. Furthermore, source code of PON-SC could be potentially modified to output the atoms that belong to the rotamer which is predicted to render steric clashes. Hence, these atoms would be colored in a distinct color in three-dimensional molecular visualization using JSmol.

Acknowledgements

I sincerely thank Mauno Vihinen for providing me with the idea of this web application, suggestions for new features and advice to multiple databases in proteomics. Additionally, I sincerely thank Jelena Čalyševa for source code of PON-SC.

References

1. wwPDB consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. Nucleic Acids Res. 2019;47(D1):D520-D528.

2. Čalyševa J, Vihinen M. PON-SC - program for identifying steric clashes caused by amino acid substitutions. BMC Bioinformatics. 2017;18(1):531. Published 2017 Nov 29.

3. Shapovalov MV, Dunbrack RL Jr. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. Structure. 2011;19(6):844-858.

4. Heinig M, Frishman D. STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. Nucleic Acids Res. 2004;32(Web Server issue):W500-W502.

5. Pedregosa, Fabian & Varoquaux, Gael & Gramfort, Alexandre & Michel, Vincent & Thirion, Bertrand & Grisel, Olivier & Blondel, Mathieu & Prettenhofer, Peter & Weiss, Ron & Dubourg, Vincent & Vanderplas, Jake & Passos, Alexandre & Cournapeau, David & Brucher, Matthieu & Perrot, Matthieu & Duchesnay, Edouard & Louppe, Gilles. (2012). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 12.

6. Rego N, Koes D. 3Dmol.js: molecular visualization with WebGL. Bioinformatics. 2015;31(8):1322-1324.

7. Sayle RA, Milner-White EJ. RASMOL: biomolecular graphics for all. Trends Biochem Sci. 1995;20(9):374.

8. Jmol: an open-source Java viewer for chemical structures in 3D. http://www.jmol.org/.

9. Burley SK, Bhikadiya C, Bi C, et al. RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. Nucleic Acids Res. 2021;49(D1):D437-D451.

10. Nightingale A, Antunes R, Alpi E, et al. The Proteins API: accessing key integrated protein and genome information. Nucleic Acids Res. 2017;45(W1):W539-W544.

11. pdb2uniprot [computer program]. Version 1.0.0. Hinxton, Cambridgeshire, UK: European Bioinformatics Institute; 2020, <https://github.com/mgalardini/pdb2uniprot>.

12. *About Node.js*, viewed 26 March 2021, <https://nodejs.org/en/about/>.

13. Lambert M. Surhone, Mariam T. Tennoe, and Susan F. Henssonow. 2010. Node.js. Betascript Publishing, Beau Bassin, MUS.

14. *Express 4.17.1, Fast, unopinionated, minimalist web framework for Node.js*, viewed 26 March 2021, <https://expressjs.com/>.

15. *React, A JavaScript library for building user interfaces*, viewed 26 March 2021, <https://reactjs.org/>.

16. *Refs and the DOM*, viewed 26 March 2021, <https://reactjs.org/docs/refs-and-the-dom.html>.

17. *Typed JavaScript at Any Scale*, viewed 26 March 2021, <https://www.typescriptlang.org/>.

18. *PDBsum, pictorial database of 3D structures in the Protein Data Bank*, viewed 26 March 2021, <https://www.ebi.ac.uk/thornton-srv/databases/cgi-bin/pdbsum/GetPage.pl?pdbcode=index.html/>.

19. *Protein Data Bank in Europe, Bringing Structure to Biology*, viewed 26 March 2021, <https://www.ebi.ac.uk/pdbe/entry/pdb/>.

20. *RCSB PDB, A Structural View of Biology,* viewed 26 March 2021, <https://www.rcsb.org/structure/>.

21. *Protein Data Bank in Europe, Bringing Structure to Biology, mapping API* , viewed 26 March 2021, <https://www.ebi.ac.uk/pdbe/api/mappings/>.

22. *UniProtKB*, viewed 26 March 2021, <https://www.uniprot.org/uniprot/>.

23. *Proteins API, variation*, viewed 26 March 2021, <https://www.ebi.ac.uk/proteins/api/variation/>.

24. PDBe-KB, *Protein Data Bank in Europe Knowledge Base, proteins*, viewed 26 March 2021, <https://www.ebi.ac.uk/pdbe/pdbe-kb/proteins/>.

25. *Welcome to Human variation Analysis (HUMA) platform*, viewed 26 March 2021, <https://huma.rubi.ru.ac.za/#proteins/fetch/>.

26. *VarSite – Disease variants and protein structure*, viewed 26 March 2021, <https://www.ebi.ac.uk/thornton-srv/databases/cgi-bin/VarSite/GetPage.pl>.