iCn3D: from Web-based 3D Viewer to Structure Analysis Tool in Batch Mode

Jiyao Wang[1*], Philippe Youkharibache[2], Aron Marchler-Bauer[1], Christopher Lanczycki[1], Dachuan Zhang[1], Shennan Lu[1], Thomas Madej[1,2], Gabriele H. Marchler[1], Tiejun Cheng[1], Li Chuin Chong[3], Sarah Zhao[4], Kevin Yang[5], Jack Lin[6], Zhiyu Cheng[7], Rachel Dunn[8], Sridhar Acharya Malkaram[9], Chin-Hsien Tai[2], David Enoma[10], Ben Busby[11], Nicholas Johnson[12], Francesco Tabaro[13], Guangfeng Song[1], Yuchen Ge[14]

[1]National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, 20894, USA

[2]National Cancer Institute, National Institutes of Health, Bethesda, MD, 20894, USA

[3]Bezmialem Vakif University, Istanbul, Turkey

[4]Cary Academy, Cary, NC, USA

[5]University of Maryland, College Park, MD, USA

[6]University of Washington, Tacoma, WA, USA

[7]University of California, Irvine, CA, USA

[8]University of British Columbia, Vancouver, BC, Canada

[9]West Virginia State University, Institute, WV, USA

[10]Noma Technology, Port Harcourt, Nigeria

[11]DNAnexus, Pittsburgh, PA, USA

[12]National Institute of Neurological Disorders and Stroke, National Institutes of Health, Bethesda, MD, 20894, USA

[13]European Molecular Biology Laboratory, Monterotondo, Italy

[14]Johns Hopkins University, Baltimore, MD, USA

*** Correspondence:**
Corresponding Author
wangjiy@ncbi.nlm.nih.gov

Keywords: iCn3D, web-based, 3D viewer, structure analysis, batch, Node.js, DelPhi, mutation

Abstract

iCn3D was originally released as a web-based 3D viewer, which allows users to create a custom view in a life-long, shortened URL to share with colleagues. Recently, iCn3D was converted to use JavaScript classes and could be used as a library to write Node.js scripts. Any interactive features in iCn3D can be converted to Node.js scripts to run in batch mode for a large data set. Currently the following Node.js script examples are available at https://github.com/ncbi/icn3d/tree/master/icn3dnode: ligand-protein interaction, protein-protein interaction, change of interactions due to residue mutations, DelPhi electrostatic potential, and solvent accessible surface area. iCn3D PNG images can also be exported in batch mode using a Python script. Other recent features of iCn3D include the alignment of multiple chains from different structures, realignment, dynamic symmetry calculation for any subsets, 2D cartoons at different levels, and interactive contact maps. iCn3D can also be used in Jupyter Notebook as described at https://pypi.org/project/icn3dpy.

1 Introduction

Since the release of GLmol and iview [1], WebGL-based structure viewers (e.g., 3Dmol [2], NGL [3; 4], Aquaria [5], iCn3D [6], LiteMol [7], and Mol* [8]) have been widely adapted in the structure biology field [9]. These web-based viewers have several advantages over stand-alone viewers (e.g., RasMol [10], VMD [11], Cn3D [12], JMOL [13], Chimera [14; 15], and PyMOL [16]). First, users can view 3D structures directly in web browsers without installation. Second, users' custom views could be shared with colleagues in a shortened link (or an image) with all data and command history. Third, these viewers are written in open-source JavaScript code and their features can be easily shared with each other.

iCn3D (I-see-in-3D) has all these features plus its own unique feature: all interactive structure analyses in iCn3D could be converted to analyses in batch mode using Node.js scripts, which call functions in iCn3D. This feature has been available since we modernized the code of iCn3D version 3 to use JavaScript classes.

iCn3D was first released in 2016 as a 3D structure viewer with synchronized 1D, 2D and 3D views and different annotations. The custom view can be shared in a life-long, shortened URL. Since then, we introduced interaction analyses [17] and converted three pieces of software into RESTful APIs. First, DelPhi [18] was licensed from Columbia University and was used to calculate electrostatic potentials for nucleotides, proteins, and ligands in iCn3D. Second, scap [19] was kindly provided by the Barry Honig group at Columbia University and was used to predict the side chain change due to mutation. Third, SymD [20] was kindly provided by Chin-Hsien Tai at the National Cancer Institute and was used to calculate protein symmetries dynamically [21].

We also added other features including the alignment of multiple chains from different structures, realignment, dynamic symmetry calculation for any subsets, 2D cartoons at different levels, and interactive contact maps. iCn3D can also be used in Jupyter Notebook.

2 Material and Methods

2.1 RESTful API based on DelPhi

The C++ code of DelPhi [18] was licensed from Columbia University and converted to a RESTful API "delphi.cgi" at National Center for Biotechnology Information (NCBI). To qualitatively illustrate the electrostatic potential and speed up the calculation, the linear Poisson-Boltzmann equation was solved. DelPhiPKa [22] is used to add hydrogens and partial charges to proteins and nucleotides. Open Babel [23] is used to add hydrogens to ligands. Antechamber [24] is used to add partial charges to ligands with the Gasteiger charge calculation method. The default grid size (n=65) and default salt concentration (0.15 M) can be changed. The pH is set at 7.0. The parameters of all RESTful APIs are described at https://www.ncbi.nlm.nih.gov/Structure/icn3d/icn3d.html#restfulapi.

2.2 RESTful API based on scap

The C++ code of Jackal/scap [19] was provided by the Barry Honig group and converted to a RESTful API "scap.cgi" at NCBI. The parameter "min" was set as "1". The scap program first iteratively samples all sidechain rotamers until convergence. The final lowest-energy conformation among all the conformations starting from the initial 3 conformations will then be minimized by refining the side-chain conformation with 2-degree rotation on each bond in the sidechain to search for lower energy conformations around the rotamer. The large rotamer library with the file "side_large_rotamer_format.txt" was used.

2.3 RESTful API based on SymD

The C++ code of SymD [20] was provided by the Chin-Hsien Tai and Byungkook Lee group and converted to a RESTful API "symd.cgi" at NCBI. The maximum allowed number of atoms is 30,000. A symmetry is considered found if the Z score is greater than 8. Only Cn and helical symmetries are considered.

2.4 Modernize iCn3D Code and Update Three.js

Before iCn3D version 3, two JavaScript objects "iCn3DUI" and "iCn3D" were used to contain all functions. In iCn3D version 3, we switched from ECMAScript 2009 (ES5) to

ECMAScript 2015 (ES6) and used JavaScript classes. Each class is in a separate file and can access the classes "iCn3DUI" and "iCn3D", which can access almost all functions. The class structure is described at https://www.ncbi.nlm.nih.gov/Structure/icn3d/icn3d.html#classstructure.

We also updated Three.js from version 103 to version 128, which uses JavaScript classes as well. Since the changes of both iCn3D and Three.js are significant, we kept the previous library names untouched, and used new library names for Three.js and iCn3D. The updated embedding method is described at https://www.ncbi.nlm.nih.gov/Structure/icn3d/icn3d.html#HowToUse.

2.5 Build iCn3D Source Code

All JavaScript classes are bundled with rollup (https://rollupjs.org/guide/en/). Then gulp (https://gulpjs.com) was used to set up all files for release in iCn3D GitHub page (https://github.com/ncbi/icn3d) and icn3d npm package (https://www.npmjs.com/package/icn3d).

3 Results

3.1 Interaction Analysis

iCn3D shows eight kinds of interactions between molecular structures: contact, hydrogen bond, salt bridge or ionic interaction, halogen bond, π-cation interaction, and π-stacking. Users can generate the interactions shown in Figure 1 in three steps: click the menu "Analysis > Interactions", select the two sets chain 6M0J_A and 6M0J_E in the popup window, and click "2D Interaction Network". Figure 1A shows the interacting residues between ACE2 (in pink) and SARS-CoV-2 spike protein (in blue) using the PDB structure 6M0J. The green, cyan, and gray dotted lines show the hydrogen bonds, salt bridges, and contacts, respectively. Figure 1B shows the 2D interaction network, where the nodes represent residues, and the lines represent interactions. The interaction includes multiple hydrogen bonds, two salt bridges and many contacts. The custom view can be reproduced with a life-long sharable link https://structure.ncbi.nlm.nih.gov/icn3d/share.html?oLwZzGL59izJeEBVA as described in Figure 1.

This interactive analysis can be converted to a Node.js script https://github.com/ncbi/icn3d/blob/master/icn3dnode/epitope.js by using the npm package icn3d (https://www.npmjs.com/package/icn3d). Users can simply run the command "node epitope.js 6M0J E A" in a command line to get the interacting residues between the chain E (spike protein) and chain A (ACE2) of the PDB structure 6M0J.

3.2 Mutation Analysis

iCn3D can show side chain mutations in 3D views. The side chain prediction was done with a RESTful API based on scap [19]. Users can show the change of interactions due to mutations by clicking the menu "Analysis > Mutation", inputting the mutation (such as "6M0J_E_501_Y" to indicate the mutation is at position 501 of the E chain of PDB 6M0J and the mutant is Tyr residue), and clicking "Interactions". Users can press the key "a" to toggle between the wild type and mutant. Figure 2A shows the mutant's 3D structure in the stick style. The spike protein is in blue and ACE2 is in pink. Figure 2B shows that the mutant Y501 (bottom panel) introduces one π-cation interaction (red line) and one π-stacking (blue line). The view can be reproduced with the sharable link as described in Figure 2. The interactive analysis can be converted to a Node.js script https://github.com/ncbi/icn3d/blob/master/icn3dnode/interaction2.js. Users can run the command "node interaction2.js 6M0J E 501 Y" to get the change of interactions with counts.

3.3 DelPhi Electrostatic Potential

iCn3D can also show electrostatic potential for structures. The electrostatic potential was calculated by a RESTful service based on the software DelPhi [18], which is licensed from Columbia University. Figure 3A shows the binding of the PH domain of phospholipase C Delta to IP_3 (or a membrane containing PIP_2). The binding region contains multiple positively charged residues (in blue). Figure 3B shows the electrostatic potential on the surface of the PH domain. Blue and red indicates +75 and -75mV, respectively. The binding region clearly has large positive electrostatic potential. The custom view of Figure 3B can be reproduced with a sharable link: https://structure.ncbi.nlm.nih.gov/icn3d/share.html?8WqQV4wir5oFx7aM9. The DelPhi electrostatic potential on the surface of chain A can be exported from a Node.js script https://github.com/ncbi/icn3d/blob/master/icn3dnode/delphipot.js with the command "node delphipot.js 1MAI A". Users could also choose to show an equipotential map (Figure 3C) instead of surface potential. The blue and red mesh show the +25 and -25 mV equipotential profile.

3.4 Dynamic Symmetry Calculation

iCn3D can not only show the pre-calculated symmetry from PDB, but also calculate symmetry for any selected residues. The calculation is done by a RESTful API based on the software SymD [20]. Figure 4B shows the structure of PDB 3HUJ. Even though the whole structure has no symmetry, its subset of residues 1-178 in chain A has C2 symmetry as shown in Figure 4A. The symmetric parts are colored in red (same residue) or blue (different residues). The feature is available in the menu "Analysis > Symmetry > from SymD (Dynamic)". The custom view can be reproduced in a sharable link as describe in Figure 4.

3.5 Multiple Chain Alignment, and realignment

iCn3D can align multiple chains from different structures. Figure 5 shows an example with two structures. The master structure is the first PDB structure 1HHO. All other structures are aligned to it. Figure 5A shows the alignment with full chains using the VAST algorithm [25] based purely on geometric criteria. Figure 5B shows the alignment to a subset of residues 10-50 in the master structure. A sequence alignment to the set of residues is done, followed by the structure superposition of the coordinates of these residues. Figure 5C shows the alignment of predefined residues in each structure. The coordinates of the pre-defined residues are used for structure superposition. The sharable links are shown in Figure 5.

Users can also load multiple PDB files and use the "File > Realign Selection" feature to align multiple structures. The "on Sequence Alignment" option is similar to Figure 5B and uses a sequence alignment. The "Residue by Residue" option is similar to Figure 5C and uses predefined residues.

3.6 2D Cartoons at Chain, Domain, and Secondary Structure levels

Previously, iCn3D only showed “2D Interaction” for structure data from NCBI with the URL parameter “mmdbid=”. We added “2D Cartoon” for any data source such as from PDB with the URL parameter “mmtfid=”. Figure 6A shows the AlphaFold predicted structure for the UniProt ID A0A061AD48. iCn3D uses the URL parameter “afid=A0A061AD48” to get the AlphaFold predicted structure data from EMBL-EBI [26]. The high confidence parts are colored in blue and the low confidence parts are colored in yellow and orange. Figure 6B shows the 2D cartoon of the two domains in this protein: Nek and ATS1. The principal axes and size of the domains are used to determine the orientation and size of the ovals, respectively. The nodes can be dragged to modify the view of the cartoon and clicked to be selected in 3D.

Figure 6C shows the selected Nek domain. Figure 6D shows the 2D cartoon at the secondary structure level for the Nek domain by clicking the menu “Analysis > 2D Cartoon > Helix/Sheet Level”. The helices and sheets are projected from 3D to 2D and thus the 2D view is dynamic depending on the orientation of the 3D view. The helices and sheets can be dragged to rearrange the view and clicked to be selected in 3D. For a structure with multiple chains, the 2D cartoon at the chain level is useful too. The sharable link is shown in Figure 6.

3.7 Interactive Contact Map

iCn3D can also display interactive 2D maps for any subsets. Figure 7 shows the contact map of AlphaFold structure with UniProt ID A0A061AD48. The contacts within 8 angstroms of C-beta atoms are shown. The map has a scale of 0.04. If the scale is set as 1, the X- and Y-axes show residues as nodes with residue names and colors. The residue nodes and the contact points could be clicked to show the residues or contacts. This feature is available in the menu “Analysis > Contact Map”. It can be dynamically applied to multiple structures or a subset of the structure. The sharable link is shown in Figure 7.

3.8 Node.js Script Based on iCn3D

After we modernized iCn3D with JavaScript classes and released the npm package icn3d, we can use scripts to analyze a large set of structures in batch mode. A few Node.js scripts have been shown above. More examples are listed at https://github.com/ncbi/icn3d/tree/master/icn3dnode. We use the Node.js script “ligand.js” as an example to show the flow of the code. First, install the npm libraries "icn3d", "three", "jsdom", and "jquery". Then set up two variables used in all iCn3D functions: "me" is an instance of the iCn3DUI class, and "ic" is an instance of the iCn3D class. The class structure is described at https://www.ncbi.nlm.nih.gov/Structure/icn3d/icn3d.html#classstructure. Next, retrieve the coordinate data from NCBI. Finally, call the iCn3D function "ic.showInterCls.pickCustomSphere_base" to get the residues interacting with the ligand. Users now can use the command "node ligand.js [PDB ID] [three-letter ligand residue name]" to calculate the interaction between a ligand and a protein.

3.9 iCn3D in Jupyter Notebook

We released icn3dpy at https://pypi.org/project/icn3dpy/ to make iCn3D work in Jupyter Notebook. Users just need to install icn3dpy with the command "pip install icn3dpy" and install one extension with the command "jupyter labextension install jupyterlab_3dmol". After launching Jupyter Notebook, three lines of script will launch the same view as in Figure 1. First, "import icn3dpy". Second, set up the view by specifying the PDB ID and command, which is the string after the "command=" in the expanded URL of https://structure.ncbi.nlm.nih.gov/icn3d/share.html?oLwZzGL59izJeEBVA from Figure 1: "view = icn3dpy.view(q='mmdbid=6m0j',command='line graph interaction pairs | !A !E | hbonds,salt bridge,interactions,halogen,pi-cation,pi-stacking | false | threshold 3.8 6 4 3.8 6 5.5; show selection; add residue number labels; set label scale 2.0|||{"factor":"0.6758","mouseChange":{"x":"0.000","y":"0.000"},"quaternion":{"_x":"-0.6336","_y":"0.3127","_z":"-0.3564","_w":"0.6114"}}')". Finally type "view" to see the interactive 3D view and 2D interaction network similar to Figure 1.

4 Discussion

With the release of more and more AlphaFold predicted structures and experimentally determined structures, it's crucial to be able to do structure analysis in batch mode. The iCn3D-based Node.js scripts fully leverage the power of iCn3D and can achieve any feature in iCn3D except graphical displays, which could be achieved by some Python code. The Python code to export 3D images from iCn3D is shown at https://github.com/ncbi/icn3d/blob/master/icn3dnode/batch_export_png.py.

Users can save their work in iCn3D by saving iCn3D PNG images with the menu "File > Save File > iCn3D PNG Image". These images not only contain the custom views, but also contain all data and command histories. They can be loaded back to iCn3D with the menu "File > Open File > iCn3D PNG Image". They can also be retrieved from the web to reproduce the saved views, e.g., https://www.ncbi.nlm.nih.gov/Structure/icn3d/full.html?type=icn3dpng&url=https://www.ncbi.nlm.nih.gov/Structure/icn3d/pdb/3GVU.png.

We showed three examples to convert software packages to RESTful APIs so that iCn3D can use them in AJAX calls seamlessly. To follow the FAIR (Findable, Accessible, Interoperable and Reusable) guiding principles [27; 28], bioinformatic tools had better provide RESTful APIs to make these tools accessible and reusable.

Our future plan includes linking iCn3D with multiple sequence alignment tools, allowing iCn3D to assemble different parts of structures, adapting WebGL2 in iCn3D, and more.

Conflict of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions

Philippe Youkharibache is the driving force for the development of iCn3D. He provides many designing ideas and consistently tests the features of iCn3D. As part of iCn3D team, Aron Marchler-Bauer, Christopher Lanczycki and Gabriele Marchler give many suggestions and test iCn3D features. Dachuan Zhang, Shennan Lu and Thomas Madej provide data for iCn3D from backend servers. Tiejun Cheng developed the Python code to export 3D images in batch mode. The following people contributed to the iCn3D feature of 2D cartoon during the ISMB Codeathon 2021: Li Chuin Chong, Sarah Zhao, Kevin Yang, Jack Lin, and Zhiyu Cheng. The following people contributed to the iCn3D features of dynamic symmetry and side chain prediction due to mutation during the STRIDES Codeathon 2020: Rachel Dunn, Sridhar Acharya Malkaram, Chin-Hsien Tai, and David Enoma. The following people contributed to using iCn3D in Jupyter Notebook during the ISMB Codeathon 2020: Ben Busby, Nicholas Johnson, Francesco Tabaro, Guangfeng Song, and Yuchen Ge.

Funding

The work of JW, PY, AM, CL, DZ, SL, TM, GHM, TC, and GS was supported by the Intramural Research Program of the National Library of Medicine at the National Institutes of Health/DHHS. Comments, suggestions and questions are welcome and should be directed to: info@ncbi.nlm.nih.gov. Funding to pay the Open Access publication charges for this article was provided by the Intramural Research Program of the National Library of Medicine at the National Institutes of Health/DHHS.

Acknowledgments

We would like to specially thank Barry Honig for licensing DelPhi and providing scap to us; Emil Alexov for his very helpful suggestions about using DelPhi; David Koes for his help in using iCn3D in Jupyter Notebook; Ravinder Abrol, Raul Cachau, Todd Smith and Sandra Porter for their feedback and suggestions during the development of iCn3D; and Ravinder Abrol and Allissa Dillman for organizing several codeathons related to iCn3D.

References

[1] H. Li, K.S. Leung, T. Nakane, and M.H. Wong, iview: an interactive WebGL visualizer for protein-ligand complex. BMC Bioinformatics 15 (2014) 56.

[2] N. Rego, and D. Koes, 3Dmol.js: molecular visualization with WebGL. Bioinformatics 31 (2015) 1322-4.

[3] A.S. Rose, A.R. Bradley, Y. Valasatava, J.M. Duarte, A. Prlic, and P.W. Rose, NGL Viewer: Web-based molecular graphics for large complexes. Bioinformatics doi: 10.1093/bioinformatics/bty419 (2018).

[4] A.S. Rose, and P.W. Hildebrand, NGL Viewer: a web application for molecular visualization. Nucleic Acids Res 43 (2015) W576-9.

[5] S.I. O'Donoghue, K.S. Sabir, M. Kalemanov, C. Stolte, B. Wellmann, V. Ho, M. Roos, N. Perdigao, F.A. Buske, J. Heinrich, B. Rost, and A. Schafferhans, Aquaria: simplifying discovery and insight from protein structures. Nature Methods 12 (2015) 98-99.

[6] J. Wang, P. Youkharibache, D. Zhang, C.J. Lanczycki, R.C. Geer, T. Madej, L. Phan, M. Ward, S. Lu, G.H. Marchler, Y. Wang, S.H. Bryant, L.Y. Geer, and A. Marchler-Bauer, iCn3D, a web-based 3D viewer for sharing 1D/2D/3D representations of biomolecular structures. Bioinformatics 36 (2020) 131-135.

[7] D. Sehnal, M. Deshpande, R.S. Varekova, S. Mir, K. Berka, A. Midlik, L. Pravda, S. Velankar, and J. Koca, LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data. Nat Methods 14 (2017) 1121-1122.

[8] D. Sehnal, S. Bittrich, M. Deshpande, R. Svobodova, K. Berka, V. Bazgier, S. Velankar, S.K. Burley, J. Koca, and A.S. Rose, Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. Nucleic Acids Res 49 (2021) W431-W437.

[9] P.W. Hildebrand, A.S. Rose, and J.K.S. Tiemann, Bringing Molecular Dynamics Simulation Data into View. Trends Biochem Sci 44 (2019) 902-913.

[10] R.A. Sayle, and E.J. Milner-White, RASMOL: biomolecular graphics for all. Trends Biochem Sci 20 (1995) 374.

[11] W. Humphrey, A. Dalke, and K. Schulten, VMD: visual molecular dynamics. J Mol Graph 14 (1996) 33-8, 27-8.

[12] Y. Wang, L.Y. Geer, C. Chappey, J.A. Kans, and S.H. Bryant, Cn3D: sequence and structure views for Entrez. Trends Biochem Sci 25 (2000) 300-2.

[13] M.E. Cass, H.S. Rzepa, D.R. Rzepa, and C.K. Williams, The use of the free, open-source program Jmol to generate an interactive web site to teach molecular symmetry. Journal of Chemical Education 82 (2005) 1736-1740.

[14] T.D. Goddard, C.C. Huang, E.C. Meng, E.F. Pettersen, G.S. Couch, J.H. Morris, and T.E. Ferrin, UCSF ChimeraX: Meeting modern challenges in visualization and analysis. Protein Sci 27 (2018) 14-25.

[15] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, and T.E. Ferrin, UCSF chimera - A visualization system for exploratory research and analysis. J Comput Chem 25 (2004) 1605-1612.

[16] L.L.C. Schrödinger, The PyMOL Molecular Graphics System. Version 2.0 (2015).

[17] P. Youkharibache, R. Cachau, T. Madej, and J. Wang, Using iCn3D and the World Wide Web for structure-based collaborative research: Analyzing molecular interactions at the root of COVID-19. bioRxiv (2020).

[18] A. Nicholls, and B. Honig, A Rapid Finite-Difference Algorithm, Utilizing Successive over-Relaxation to Solve the Poisson-Boltzmann Equation. J Comput Chem 12 (1991) 435-445.

[19] Z.X. Xiang, and B. Honig, Extending the accuracy limits of prediction for side-chain conformations (vol 311, pg 421, 2001). J. Mol. Biol. 312 (2001) 419-419.

[20] C.H. Tai, R. Paul, K.C. Dukka, J.D. Shilling, and B. Lee, SymD webserver: a platform for detecting internally symmetric protein structures. Nucleic Acids Res 42 (2014) W296-300.

[21] P. Youkharibache, Topological and Structural Plasticity of the Single Ig Fold and the Double Ig Fold Present in CD19. Biomolecules 11 (2021) 1290.

[22] L. Wang, M. Zhang, and E. Alexov, DelPhiPKa web server: predicting pKa of proteins, RNAs and DNAs. Bioinformatics 32 (2016) 614-5.

[23] N.M. O'Boyle, M. Banck, C.A. James, C. Morley, T. Vandermeersch, and G.R. Hutchison, Open Babel: An open chemical toolbox. J Cheminform 3 (2011) 33.

[24] J. Wang, R.M. Wolf, J.W. Caldwell, P.A. Kollman, and D.A. Case, Development and testing of a general amber force field. J Comput Chem 25 (2004) 1157-74.

[25] J.F. Gibrat, T. Madej, and S.H. Bryant, Surprising similarities in structure comparison. Curr. Opin. Struct. Biol. 6 (1996) 377-85.

[26] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zidek, A. Potapenko, A. Bridgland, C. Meyer, S.A.A. Kohl, A.J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A.W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, Highly accurate protein structure prediction with AlphaFold. Nature 596 (2021) 583-589.

[27] B. Mons, C. Neylon, J. Velterop, M. Dumontier, L.O.B.S. Santos, and M.D. Wilkinson, Cloudy, increasingly FAIR; revisiting the FAIR Data guiding principles for the European Open Science Cloud. Information Services & Use 37 (2017) 49-56.

[28] M.D. Wilkinson, R. Verborgh, L.O.B.D. Santos, T. Clark, M.A. Swertz, F.D.L. Kelpin, A.J.G. Gray, E.A. Schultes, E.M. van Mulligen, P. Ciccarese, A. Kuzniar, A. Gavai, M. Thompson, R. Kaliyaperumal, J.T. Bollema, and M. Dumontier, Interoperability and FAIRness through a novel combination of Web technologies. Peerj Comput Sci (2017).

Figures

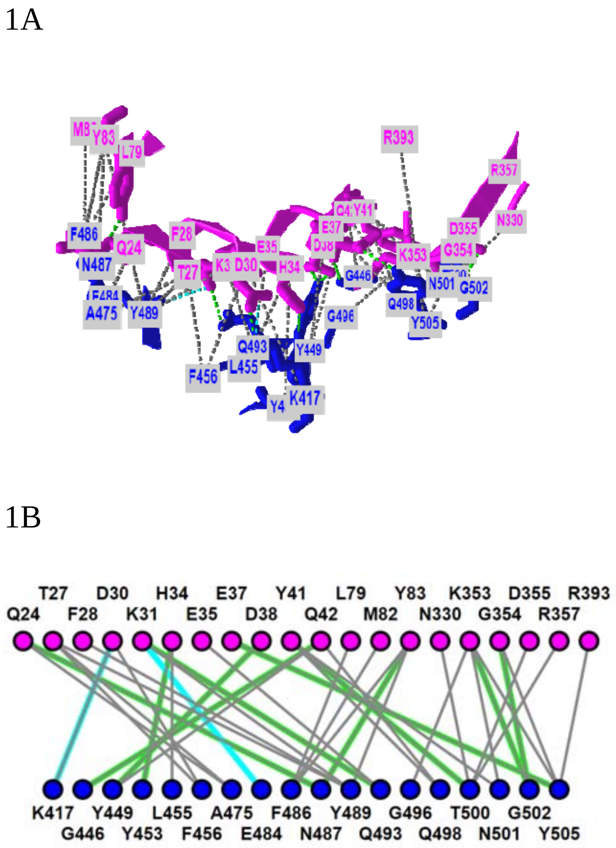


Figure 1. 3D and 2D views of the interaction between ACE2 (pink) and SARS-CoV-2 spike protein (blue) (PDB 6M0J). (A) 3D view of the interaction. Hydrogen bonds, salt bridges, and contacts are depicted as dotted lines and are colored in green, cyan, and gray, respectively. (B) 2D interaction network of the interaction. Each node and line can be clicked to show the corresponding residue and interaction in the 3D view. Sharable Link: https://structure.ncbi.nlm.nih.gov/icn3d/share.html?oLwZzGL59izJeEBVA

2A

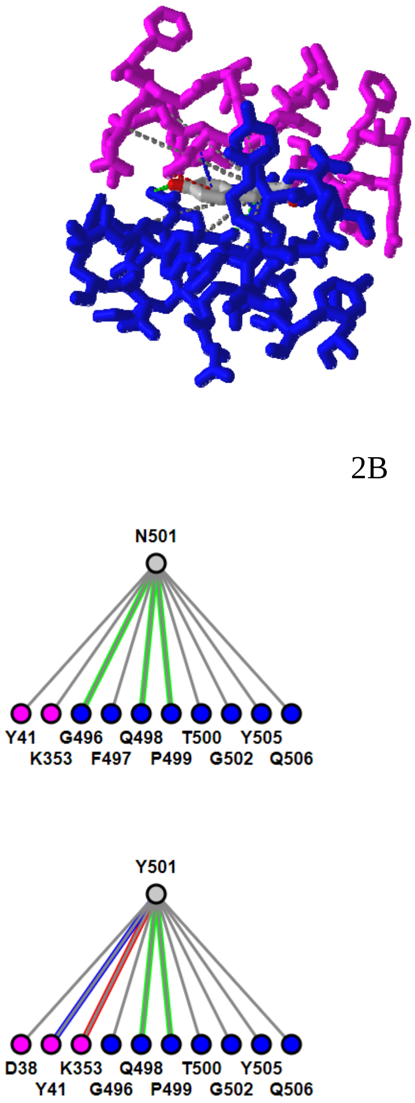
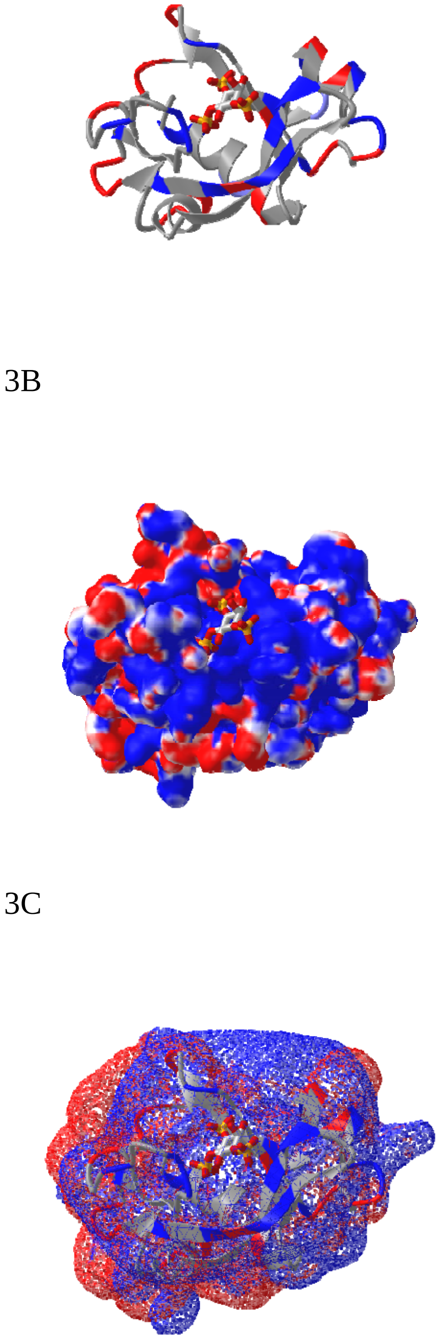


Figure 2. 3D and 2D views of mutation N501Y of SARS-Cov-2 spike protein (PDB 6M0J). (A) 3D view of the mutant with Asn at position 501 mutated to Tyr. The spike protein is colored in blue and ACE2 is colored in pink. Hydrogen bonds, contacts, π-cation interaction, and π-stacking are depicted as dotted lines and are colored in green, gray, red, blue, respectively. (B) Change of interactions due to mutation. The top panel shows the wild type, and the bottom panel shows the mutant. Sharable Link: https://structure.ncbi.nlm.nih.gov/icn3d/share.html?j5Y5CkMQPU2R6smf9

3A



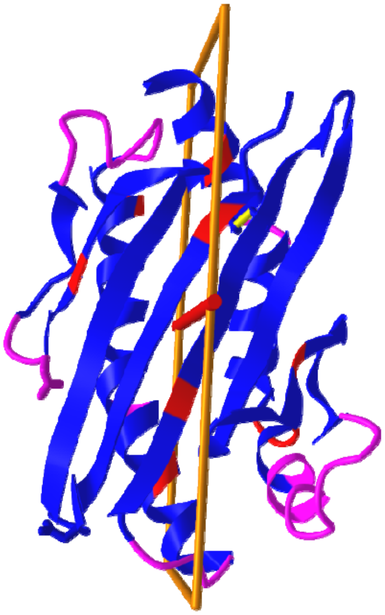
3B

3C

Figure 3. DelPhi electrostatic potential of membrane-binding PH domain (PDB 1MAI). (A) The PH domain is shown in ribbon style with positively and negatively charged residues colored in blue and red respectively. IP_3 is shown in stick style with three phosphate groups. (B) The electrostatic potential of the PH domain is shown on the surface. Blue indicates +75mV and red indicates -75 mV. Sharable link:

https://structure.ncbi.nlm.nih.gov/icn3d/share.html?8WqQV4wir5oFx7aM9. (C) The equipotential map of the PH domain. The blue and red meshes indicate +25 and -25 mV potential profile, respectively.

4A



4B

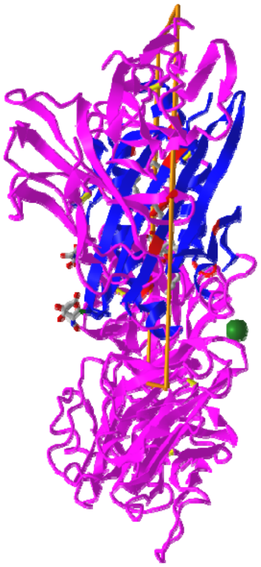
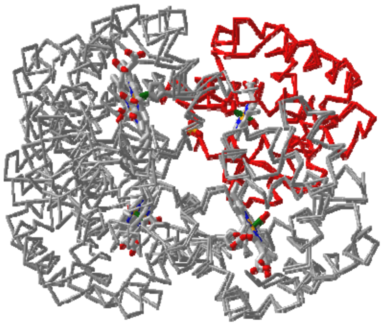
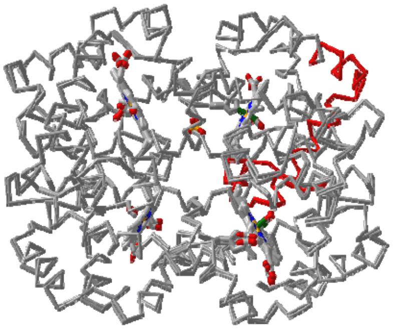


Figure 4. Dynamic symmetry calculation. (A) The subset of residues 1-178 in the A chain of PDB 3HUJ was used for symmetry calculation. The symmetric parts are colored in red (same residue) or blue (different residues). The rest are colored in pink. (B) The full structure of PDB 3HUJ with the symmetry in Chain A shown. Sharable link: https://structure.ncbi.nlm.nih.gov/icn3d/share.html?gNe7MemEb5vqYSba6.

5A



5B



5C

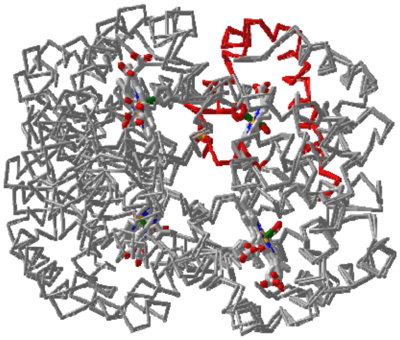
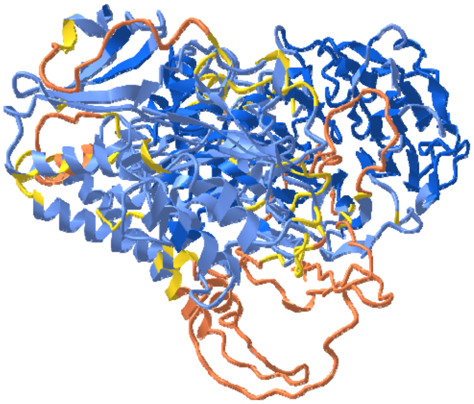
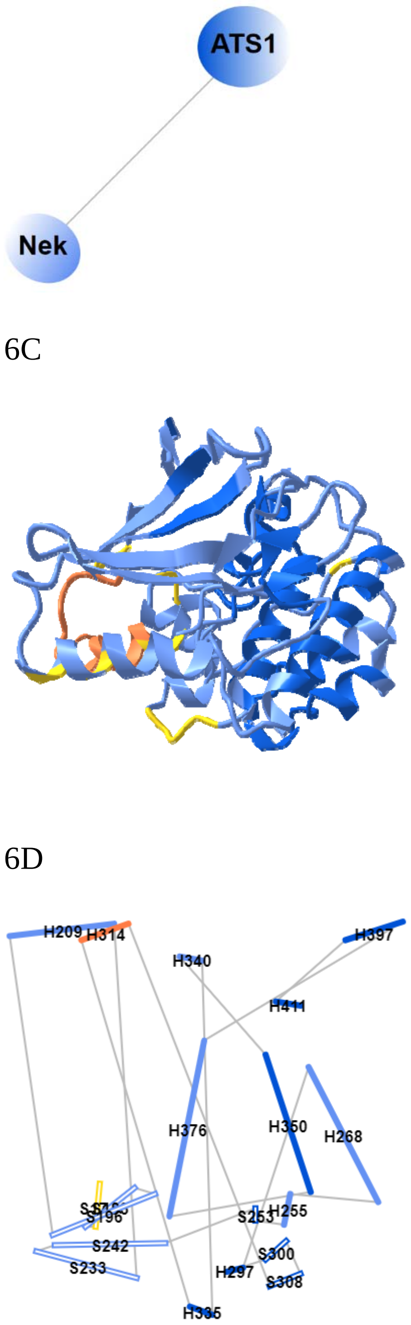


Figure 5. Multiple chain alignment of chain A of PDB 4N7N with chain A of PDB 1HHO. (A) Align with the full chains. The aligned part is colored in red. The sharable link is https://www.ncbi.nlm.nih.gov/Structure/icn3d/full.html?chainalign=1HHO_A,4N7N_A&resnum=&resdef=&showalignseq=1. (B) Align with a subset of residues 10-50 in the master chain A of PDB 1HHO. The sharable link is https://www.ncbi.nlm.nih.gov/Structure/icn3d/full.html?chainalign=1HHO_A,2HCO_A&resnum=10-50&resdef=&showalignseq=1. (C) Align with pre-defined residues: 50-100 in chain A of PDB 4N7N and residues 50-100 in chain A of PDB 1HHO. The sharable link is https://www.ncbi.nlm.nih.gov/Structure/icn3d/full.html?chainalign=1HHO_A,4N7N_A&resnum=&resdef=50-100+|+50-100&showalignseq=1.

6A



6B

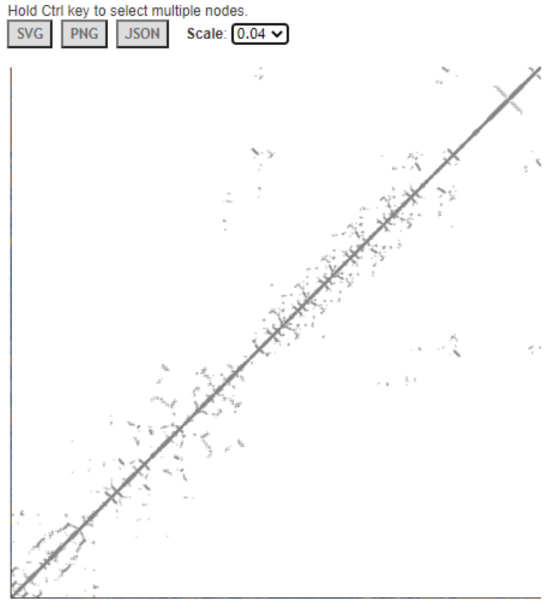


6C

6D

Figure 6. 2D Cartoons of an AlphaFold structure. (A) AlphaFold predicted structure for the UniProt ID A0A061AD48. The high confidence parts of the structure are colored in blue and the low confidence parts are colored in yellow and orange. (B) 2D cartoon in the domain level. There are two domains in the protein: Nek and ATS1. (C) The Nek domain of the structure. (D) 2D cartoon at the secondary structure level. Helices are solid cylinders and labeled as “H” plus the first residue number. Sheets are empty cylinders and labeled as “S” plus the first residue number. The sharable link for panels A and B is: https://structure.ncbi.nlm.nih.gov/icn3d/share.html?R8Wtf9hDsdo6sH5Z9

7


Hold Ctrl key to select multiple nodes.

Scale: 0.04

Figure 7. Contact map of AlphaFold structure with UniProt ID A0A061AD48. The distance threshold is 8 angstroms between C-beta atoms. The sharable link is https://structure.ncbi.nlm.nih.gov/icn3d/share.html?vTc6gkbCenTyLzt4A.