D436–D442 *Nucleic Acids Research, 2009, Vol. 37, Database issue* *Published online 3 November 2008*
doi:10.1093/nar/gkn840

VIPERdb^2: an enhanced and web API enabled relational database for structural virology

Mauricio Carrillo-Tripp[1], Craig M. Shepherd[1], Ian A. Borelli[1], Sangita Venkataraman[1], Gabriel Lander[1], Padmaja Natarajan[1], John E. Johnson[1], Charles L. Brooks III[2] and Vijay S. Reddy[1,*]

[1]Department of Molecular Biology, The Scripps Research Institute, 10550 North Torrey Pines Road, La Jolla, CA 92037 and [2]Department of Chemistry, Biophysics Program, 930 N. University Ave., University of Michigan, Ann Arbor, MI 48109, USA

Received September 15, 2008; Revised October 13, 2008; Accepted October 14, 2008

ABSTRACT

VIPERdb (http://viperdb.scripps.edu) is a relational database and a web portal for icosahedral virus capsid structures. Our aim is to provide a comprehensive resource specific to the needs of the virology community, with an emphasis on the description and comparison of derived data from structural and computational analyses of the virus capsids. In the current release, VIPERdb^2, we implemented a useful and novel method to represent capsid protein residues in the icosahedral asymmetric unit (IAU) using azimuthal polar orthographic projections, otherwise known as Φ–Ψ (Phi–Psi) diagrams. In conjunction with a new Application Programming Interface (API), these diagrams can be used as a dynamic interface to the database to map residues (categorized as surface, interface and core residues) and identify family wide conserved residues including hotspots at the interfaces. Additionally, we enhanced the interactivity with the database by interfacing with web-based tools. In particular, the applications Jmol and STRAP were implemented to visualize and interact with the virus molecular structures and provide sequence–structure alignment capabilities. Together with extended curation practices that maintain data uniformity, a relational database implementation based on a schema for macromolecular structures and the APIs provided will greatly enhance the ability to do structural bioinformatics analysis of virus capsids.

INTRODUCTION

The VIrus Particle ExploreR database (VIPERdb) (http://viperdb.scripps.edu) is a relational database of manually curated icosahedral virus capsid structures (1). Initial motivation for creating VIPERdb came from the need to orient all the capsids in one icosahedral convention and generate their proper capsid assemblies by employing a single set of 60 icosahedral matrices that in turn facilitated their computational analysis (2–6). Recent conversion of VIPERdb into a web application by the addition of new interfaces to interact with the database, has brought an average of 60+ (and growing) unique visits a day from all over the world over the past year. The user access data/statistics can be viewed at any time through the ClustrMaps web service link at the bottom left corner of any VIPERdb web page. At the time of this writing, the database has grown to contain 256 entries that belong to 28 spherical virus families and 44 genera. The frequency of addition of new entries to the database, on average, is one structure per month that gets deposited at the RCSB-PDB (7). Structures are submitted with the same frequency by the users requesting to perform VIPER analysis, which are provided as a service to the virology community.

DATA CURATION

As described previously (1), initial data curation and data entry involves, (i) obtaining the PDB and mmCIF files of the entries from the RCSB-PDB site (http://www.rcsb.org/pdb/), (ii) uploading these files into VIPERdb using OpenMMS PDBase loader (http://openmms.sdsc.edu), (iii) locally developed programs and scripts are used to obtain the PDB-to-VIPER transformation matrix, generation of pictorial capsid illustrations, perform VIPER analysis using the CHARMM molecular program (8) and create Intra-family multiple sequence alignments (IFMSAs) and (iv) populate the database with the derived results.

Further curation of virus entries is done by visually examining the gallery of images of the organization of subunits in the capsid within each individual family (Top Menu: Utilities > Gallery Maker) to see if there are

*To whom correspondence should be addressed. Tel: +1 858 784 8191; Fax: +1 858 784 8660; Email: reddyv@scripps.edu

© 2008 The Author(s)
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

any outliers based on the color-coding scheme of the subunits and the orientation of the reference asymmetric unit of the capsid with respect to the icosahedral axis. Once an outlier or an incomplete entry was identified, its coordinates are re-entered, checking each step for errors. One of the main reasons for the incorrect subunit color-coding and/or organization is due to inconsistencies in the chain labeling. In order to correct for such inconsistencies, a new check-point was added during data entry, listing all the existing chain-ids and providing an option to either retain or rename them with consistent chain-ids. By incorporating this step, problems with many of the entries were rectified. Another source of error in the data entry results from the incorrect location of the reference asymmetric unit. This is due to either the incorrect PDB-to-VIPER conversion matrix or the need to selectively transform certain segments after the PDB-to-VIPER transformation. Those entries with erroneous PDB-to-VIPER matrix are re-entered after referring to the original PDB and its BIOMT cards and incorporating the correct matrix. To correct for the rearrangement of selected chains another check-point was incorporated, where an option is made available to transform the selected chains using one of the standard icosahedral matrices. Another curation step has to do with finding inconsistencies, if any, in the estimated subunit–subunit association energies of analogous interfaces within a family of virus capids computed using CHARMM (8). In order to facilitate this, a GUI was created (Top Menu: Utilities > Family Association Energies), which allows for quick comparison of association energies of corresponding interfaces within each family to locate any outliers. To the best of our knowledge the current data in VIPERdb is correct and consistent.

INFORMATION PAGE (Info-Page)

Different sections of the Info-Page have been organized into different layers, each of which can be accessed through the Tabs on the left side of the browser: Biodata, Illustrations, 3D Icosahedral Asymmetric Unit (IAU), $\Phi-\Psi$ (Phi–Psi) Explorer, Annotations and Related Viruses (Figure 1a). The new format allows for an easy expansion of the Info-Page in the future by adding new layers (Tabs) to include new information, without clogging the view or having unmanageable page lengths. The header of the Info-Page shows the name of the virus, the title found on the original PDB file, and a count of how many times each particular entry has been accessed. A list of the most viewed entries can be found at the Statistics page of the web site (Top Menu: Main > Statistics), along with a list of the most recently deposited entries (based on their original PDB release date). A status box on the left side shows the current status of the sequence of HTTP requests. Contents and description of the individual tabs are given below.

BIODATA

Biodata tab is the default tab of the info-page, when an entry is accessed. It includes two images depicting the central icosahedral asymmetric unit and a full capsid, a link to download the primary sequence extracted from the protein structure (FASTA format), a link to download specific icosahedral lattice (cage) coordinates (PDB format) and a link to download the structure based intra-family multiple sequence alignments (IFMSAs) (9). In addition, a link to automatically load the asymmetric unit protein(s) coordinates into the STRAP application (10) is provided. STRAP in turn enables the generation of a full capsid or an oligomer of choice, by expanding the asymmetric unit using the 60 transformation matrices. The displayed/expanded set of coordinates can then be saved using the STRAP interface (drag-and-drop). In addition to being able to download directly the full or half capsid coordinates using the links in the Biodata tab, this provides an alternative way to obtain coordinates and avoid long delays in transferring big files over the internet, particularly when the user has access only to a slow network connection.

ILLUSTRATIONS

In this tab, three different representations of the capsid are presented. First, surface shaded representation of the capsid, generated using TexMol (11), is shown on the left. Organization of the subunits colored based on their type with respect to the icosahedral axes, generated using MOLSCRIPT (12), is shown in the middle. Lastly, arrangement of surface rendered subunits at various resolutions, generated using Chimera (13), with a pull down menu are shown on the right.

3D ICOSAHEDRAL ASYMMETRIC UNIT

In this layer, the 3D structure of the viral capsid protein(s) in the central icosahedral asymmetric unit can be explored interactively through a Jmol (http://www.jmol.org/) applet (Figure 2b). In this Tab, the protein(s) in the asymmetric unit are displayed in the standard VIPER convention (2), along with the corresponding icosahedral lattice. The default view shows the position of the protein(s) in relation to all the lattice symmetry axes in Cartesian space. The contents in the display window can be dragged, rotated and zoomed in or out. The control panel, on the right, offers options to show all the subunits (default) or selectively hide individual subunits, icosahedral lattice or XYZ axes, change background color, calculate and display the solvent accessible surface, color the proteins by chain-id (default) or secondary structure and change the proteins from cartoon (default) to space fill or trace representations. In order to explore the subunit interfaces in the assembled capsid, a script that generates (and removes) copies of the asymmetric unit around the 5-fold, 3-fold and 2-fold axes of symmetry is implemented. Jmol is an open source, highly extensible, application that will allow the easy inclusion of new *ad hoc* features to this layer. In addition to the controls described above, all the regular Jmol menus are always accessible by right clicking anywhere inside the applet. The display can be reset to

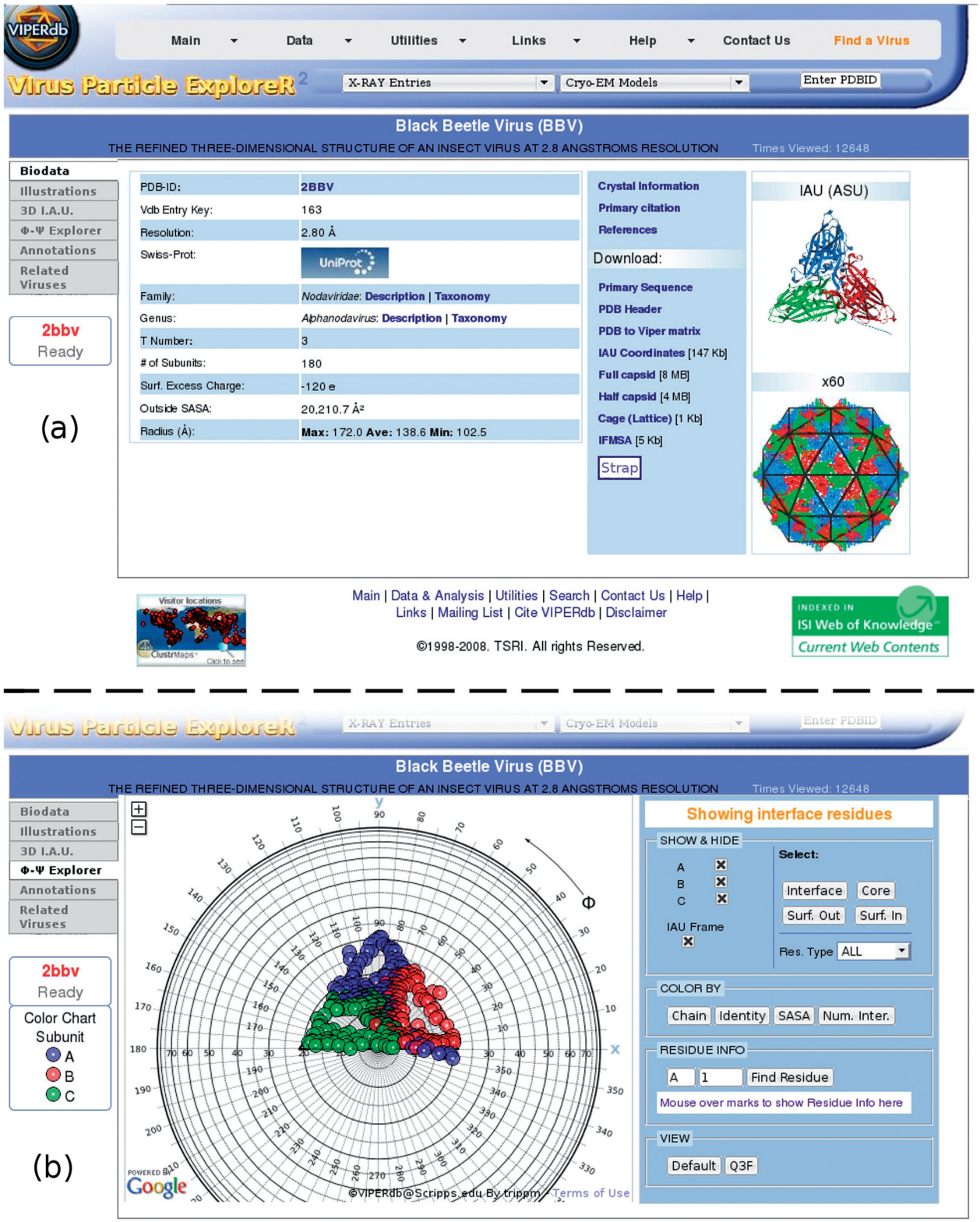

Figure 1. (**a**) Info-page of Black Beetle Virus (PDB ID 2bbv). (**b**) Default view of the Φ–Ψ Explorer Tab. See text for details.

the default view port at any time. A snapshot (static JPEG image) of the current view can be created, opening in a new window, which can be saved using the browser menu. This individual layer can also be opened in a new window using a 'liquid' layout, allowing the Jmol applet to expand and use all available space. This is a convenient feature for users with wide screens and high-resolution monitors.

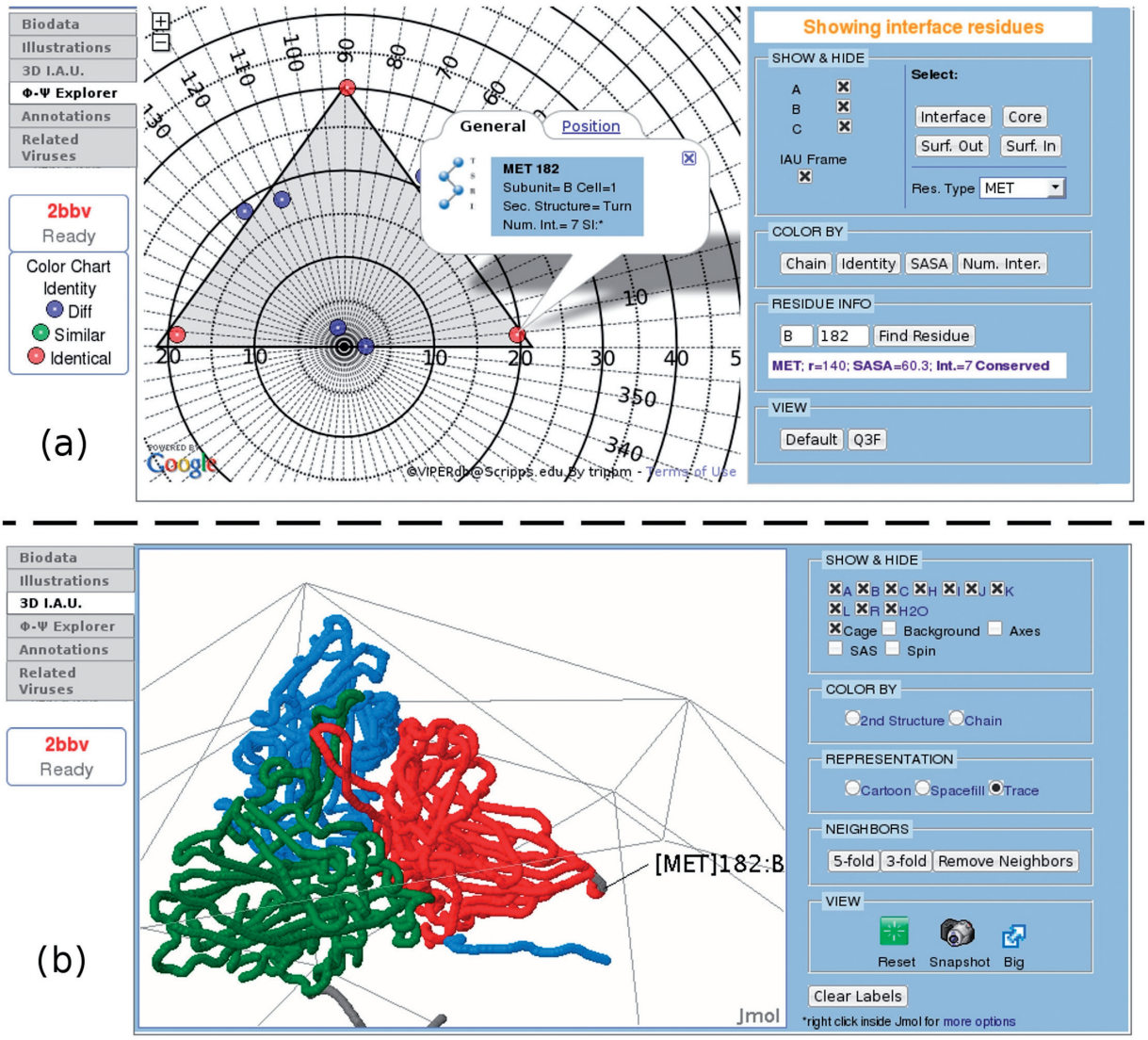

Figure 2. (**a**) Φ–Ψ Map of the interface residues of the Black Beetle Virus (PDB ID 2bbv) in the Q3F view, showing all the methionine residues color coded according to its sequence conservation (conserved among all nodaviruses are shown in red, 'Hot Spots'). Residue 182 of the B subunit is selected. (**b**) Protein structure in Cartesian space of the asymmetric unit, color coded according to subunit name (A: blue, B: red, C: green) and using the 'trace' representation. Residues are automatically displayed in the IAU tab, when they are selected in the Φ–Ψ map. See text for details.

Φ–Ψ EXPLORER

One of the main additions to VIPERdb in the current update is the inclusion of Φ–Ψ Maps (6) as a new layer (Figure 1b, 2a). This is a representation of proteins/residues in the IAU projected onto a unit sphere and then mapped onto a plane using an azimuthal orthographic transformation. Because of the spherical nature of icosahedral virus capsids, the Cartesian coordinates (x, y, z) of the center of mass of each residue in the IAU were represented in spherical coordinates (r, Φ, Ψ), where r is the magnitude of the vector **R** pointing to the center of mass of a residue, Φ is the angle between the x-axis and the projection of **R** into the xy-plane and Ψ is the angle between the z-axis and **R**. Each vector **R** is then transformed into a unit vector, leaving all points lying in the surface of a unit sphere. A 2D map can be created by making a polar azimuthal orthographic projection of the positions of all these points on the surface of the sphere. This map represents the view of a sphere (globe) seen from the top of the north pole down towards the equator, with the angle Φ starting at 0° at the x-axis on the right-hand side, growing counterclockwise up to 360° after a full circle, while the angle Ψ starts at 0° at the center of the map and grows in concentric circles up to 90° at the outer circle (sphere's equator).

Unlike other approaches, this method provides a unique advantage of mapping residue locations, onto the same area of IAU, irrespective of the capsid size or T number. Importantly, mapping of residues selectively located at the inter-subunit interfaces, allows the comparison of interactions between the capsids in a single family as well as to quantitatively estimate and visually assess the extent of the similarities (S-score) (6). This method also highlights the density and distribution of protein–protein interactions

with respect to the icosahedral symmetry axes in spherical space, as well as the presence and location of 'hot spots', when the extent of sequence conservation is taken into account. With this tool, answers to specific questions like 'Identify residues in the Black Beetle Virus that are conserved in the Nodavirus family and make contacts near the 5-fold symmetry axis' or 'What is the most abundant charged amino acid type on the outside surface among all the members of the Leviviridae family?' can be easily found.

Creating static images of these maps is an option, but we also wanted to take advantage of the new tools developed for web browsers in recent years. Mapping of residues of a spherical virus onto Φ–Ψ maps is analogous to mapping of countries/cities on the planet Earth's onto a 2D longitude-latitude world map. This led us to explore the option of taking an already existing interface used to dynamically interact with the maps of planet Earth through a web browser, and use it to display Φ–Ψ maps of the locations of residues forming the icosahedral virus capsid protein(s). One such application that met these requirements is the Google Maps project, which offers an open source Application Programming Interface (API) (http://code.google.com/apis/maps/). With a few modifications, we were able to adapt Google's API to our needs and use it, with a public key, on VIPERdb to implement interactive icosahedral virus Φ–Ψ maps. Additionally, the same method can be applied to any other systems that can be represented in two dimensions.

The Φ–Ψ map of a virus capsid in the VIPERdb can be accessed through the Info-Page under the Φ–Ψ Explorer Tab. The Φ–Ψ Map of Black Beetle Virus is shown in Figure 1b. When the browser opens the Info-Page, all the data associated with the generation of Φ–Ψ map is retrieved from the VIPERdb server through a web API (WAPI) function call and delivered to the client in XML format using AJAX technology. The map is then populated with one marker for each protein residue returned. The user interface is composed of the interactive Φ–Ψ map on the left and a control panel on the right. The map itself has all the features that a regular Google map has; it can be zoomed in or out and dragged with the mouse in any direction. Clicking on an individual marker (which represents a protein residue) will pop up an Info-Window with relevant information: amino acid type, residue number, type of secondary structure it belongs to, number of interactions made (if the residue is part of an interface), residue solvent accessible surface area (SASA, in $Å^2$) and the residue exact position in spherical coordinates (Figure 2a). The control panel is composed of several sections with each one providing specific set of options. The 'View' section offers the option to quickly switch from the current view to either the default view or the Q3F view, which is a zoomed in view down the quasi 3-fold axis of the reference asymmetric unit of the T = 3 capsids. The 'Residue Info' section offers the option to find a specific residue by entering its subunit and residue ids; an Info-Window will pop up indicating its location. The 'Show and Hide' section offers the option to independently show (on by default) or hide any of the individual subunits in the asymmetric unit as well as the IAU Frame. By default, all amino acid types are shown, but the user has the option to show only one kind of amino acid through the drop-down select box in this section. All residues in a protein are grouped in four categories based on their location: Interface, Core, Exterior Surface and Interior Surface (Carrillo-Tripp *et al.*, manuscript in preparation). Initially, the Interface residues are displayed, but any of the other three groups can be independently loaded into the map by clicking the corresponding button. The section 'Color by' offers the option to change the color-coding of the markers. The color code currently being used is displayed at the far left of the infobox. The default is to color the markers depending on the subunit they belong to. There is also an option to color the markers depending on the residues individual SASA values, or the number of interactions each residue makes (if the Interface residues are displayed). Another option is to color the markers by the sequence Identity, i.e. the primary sequence conservation among the members of the corresponding virus family (shown on the Biodata Tab). Previously calculated intra-family multiple sequence alignments (IFMSAs) based on the multiple structure alignments between all members of the family were added to the VIPER database so that they could be retrieved later via a WAPI function call. All the IFMSAs were computed with the package T-Coffee (9), using a consensus of several pairwise structure alignment libraries built with SAP (14), MUSTANG (15) and TM-align (16). The precomputed IFMSAs in clustalw format (17) can also be downloaded from the VIPERdb server, through a link in the Biodata Tab. Figure 2a shows the location of all the Methionine amino acids at the interfaces (in 2BBV), color coded by their Identity (conserved residues in red). It is striking to see that the 'hot spots', which are also the residues with the most number of interactions, are located precisely at each one of the symmetry axes. Every time a marker is clicked on the Φ–Ψ map, the corresponding residue with a label is automatically displayed in the Cartesian space that can be visualized and interacted with using the Jmol visualization tool in the 3D IAU Tab (Figure 2b).

ANNOTATIONS AND RELATED VIRUSES

Annotations tab provides links to various derived results generated using VIPER analysis using CHARMM (8). These include (i) accessible surface profiles, (ii) contact tables, a list of residue pairs that interact at various (unique) subunit–subunit interfaces, (iii) association energies of unique subunit interfaces, estimated based on the buried surface areas (2), (iv) Q-scores, quantitative measure of quasi-equivalence (5) and finally links to the oligomer generator, map-a-residue and secondary structure information calculated using STRIDE (18).

The related viruses tab provides links to all the related members in the same genus and family of the selected entry.

WEB APPLICATION PROGRAMMING INTERFACE

In order to expand the interoperability of the VIPERdb server with other web applications, a new application

programming interface, accessible via HTTP calls and written in the server-side scripting language PHP, has been designed as an addition to the previous VIPERdb interfaces (MySQL, Perl, etc.). A growing number of functions are being developed to support a wide range of possible requests. The result of each function call is customizable by a series of available options, for example, the output format (currently supported: CSV, HTML, XML, JSON, PDB, FASTA). These functions, together with a Javascript library (http://labs.adobe.com/technologies/spry/), are being actively used to support some of the added features to the VIPERdb web site, like the PDB-ID suggest box in the Top Menu, the Virus Name suggest box in the Search page (Top Menu: Find a Virus), or the—sortable by column—tables in the Family and T-number Index pages (Top Menu: Data > Family Index, and Top Menu: Data > T-Number Index, respectively). The current state and documentation of the WAPI can be found on its own web page (Top Menu: Utilities > Web API), which includes descriptions and lists of all the available options for each function and examples on how to use them.

MULTIPLE STRUCTURE AND SEQUENCE ALIGNMENT TOOLS

A mash up implementation of VIPERdb with the web application Top-Match (19,20) has been done (Top Menu: Utilities > Structure Alignment), in which the user can select any two viral capsid proteins and submit their structures to the server for structural alignment. In addition, specific single chains can be selected for both the query and the target structures. The result of the structure alignment will be displayed on a 3D interactive Jmol applet (http://www.jmol.org/), along with the corresponding sequence alignment below it. When there is a need to do a multiple sequence alignment (MSA) of more than two structures, the suggested option to do the multiple structure alignment using the application STRAP (10) is available. Several structures can be loaded into STRAP (not necessarily from the same family nor T number) simply by dragging links from any page in the VIPERdb web site (Info-Page, Family Index Page, T-number Index Page, etc.) into the STRAP application window. STRAP will automatically load the corresponding protein(s) coordinates and generates an MSA based on the multiple structure alignment when instructed by the user.

OTHER UTILITIES AND SERVICES

In order to consolidate the already existing and growing community around VIPERdb, a discussion forums section has been implemented (http://www.phpbb.com/) (Top Menu: Help > Discussion Forums). The intention of the forums is to provide a way for users and developers to interact in a more direct way, interchanging any ideas related to icosahedral viruses. A series of tutorials in the form of screen casts (Top Menu: Help > Tutorials) are being developed as educational tools, and also to aid new users exploring all the tools and features of the VIPERdb site.

ACKNOWLEDGEMENTS

We gratefully acknowledge the help and guidance of Prof. Chandrajit Bajaj of Computer visualization center, University of Texas at Austin on using the molecular graphics program TexMol, Dr Tom Goddard from University of California at San Francisco on using the molecular graphics program Chimera, Prof. Robert M. Hanson from St Olaf College for his guidance on the scripting interface of Jmol, Dr Christoph Gille from the Medical Faculty of the Humboldt University Berlin for his help in coupling VIPERdb-STRAP and Dr Cedric Notredame from the Center for Genomic Regulation in Barcelona, for his guidance on the usage of T-Coffee.

FUNDING

VIPERdb is being developed as a training/service and dissemination component of the National Institutes of Health Research Resource: Multiscale Modeling Tools for Structural Biology (MMTSB), which is fully funded by the National Center for Research Resources of the National Institutes of Health (RR12255). Funding for open access charge: NIH, NCRR, RR12255.

Conflict of interest statement. None declared.

REFERENCES

1. Shepherd,C.M., Borelli,I.A., Lander,G., Natarajan,P., Siddavanahalli,V., Bajaj,C., Johnson,J.E., Brooks,C.L. III and Reddy,V.S. (2006) VIPERdb: a relational database for structural virology. *Nucleic Acids Res.*, **34**, D386–D389.
2. Reddy,V.S., Natarajan,P., Okerberg,B., Li,K., Damodaran,K.V., Morton,R.T., Brooks,C.L. III and Johnson,J.E. (2001) Virus Particle Explorer (VIPER), a website for virus capsid structures and their computational analyses. *J. Virol.*, **75**, 11943–11947.
3. Natarajan,P., Lander,G.C., Shepherd,C.M., Reddy,V.S., Brooks,C.L. III and Johnson,J.E. (2005) Exploring icosahedral virus structures with VIPER. *Nat. Rev. Microbiol.*, **3**, 809–817.
4. Shepherd,C.M. and Reddy,V.S. (2005) Extent of protein-protein interactions and quasi-equivalence in viral capsids. *Proteins*, **58**, 472–477.
5. Damodaran,K.V., Reddy,V.S., Johnson,J.E. and Brooks,C.L. III. (2002) A general method to quantify quasi-equivalence in icosahedral viruses. *J. Mol. Biol.*, **324**, 723–737.
6. Carrillo-Tripp,M., Brooks,C.L. III and Reddy,V.S. (2008) A novel method to map and compare protein-protein interactions in spherical viral capsids. *Proteins*, **73**, 644–655.
7. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
8. Brooks,B., Bruccoleri,D., Olafson,D., States,D., Swaminathan,S. and Karplus,M. (1983) CHARMM: a program for macromolecular energy, minimization and dynamics calculations. *J. Comput. Chem.*, **4**, 187–217.
9. O'Sullivan,O., Suhre,K., Abergel,C., Higgins,D.G. and Notredame,C. (2004) 3DCoffee: combining protein sequences and structures within multiple sequence alignments. *J. Mol. Biol.*, **340**, 385–395.
10. Gille,C. and Frommel,C. (2001) STRAP: editor for STRuctural Alignments of Proteins. *Bioinformatics*, **17**, 377–378.
11. Bajaj,C., Djeu,P., Thane,A. and Siddavanahalli,V. (2004) In *Proceedings of the Annual IEEE Visualization Conference*. IEEE Computer Society Press, Austin, TX, pp. 243–250.
12. Kraulis,P. (1991) MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. *J. Appl. Crystallogr.*, **24**, 946–950.

13. Pettersen,E.F., Goddard,T.D., Huang,C.C., Couch,G.S., Greenblatt,D.M., Meng,E.C. and Ferrin,T.E. (2004) UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.*, **25**, 1605–1612.
14. Taylor,W.R., Flores,T.P. and Orengo,C.A. (1994) Multiple protein structure alignment. *Protein Sci.*, **3**, 1858–1870.
15. Konagurthu,A.S., Whisstock,J.C., Stuckey,P.J. and Lesk,A.M. (2006) MUSTANG: a multiple structural alignment algorithm. *Proteins*, **64**, 559–574.
16. Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, **33**, 2302–2309.
17. Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.
18. Frishman,D. and Argos,P. (1995) Knowledge-based protein secondary structure assignment. *Proteins*, **23**, 566–579.
19. Sippl,M.J., Suhrer,S.J., Gruber,M. and Wiederstein,M. (2008) A discrete view on fold space. *Bioinformatics*, **24**, 870–871.
20. Sippl,M.J. and Wiederstein,M. (2008) A note on difficult structure alignment problems. *Bioinformatics*, **24**, 426–427.