

PDBlocal: A web-based tool for local inspection of biological macromolecular 3D structures

Pan Wang^{*,†}, Guangxiao Yang^{*} and Guangyuan He^{*,‡}
**The Genetic Engineering International Cooperation Base of
Chinese Ministry of Science and Technology*

*The Key Laboratory of Molecular Biophysics of
Chinese Ministry of Education
College of Life Science and Technology
Huazhong University of Science and Technology
Wuhan 430074, P. R. China*

*†College of Electronics and Information Engineering
South-Central University for Nationalities
Wuhan 430074, P. R. China
‡hegy@hust.edu.cn*

Received 6 July 2017

Accepted 20 July 2017

Published 12 October 2017

Functional research on biological macromolecules must focus on specific local regions. PDBlocal is a web-based tool developed to overcome the limitations of traditional molecular visualization tools for three-dimensional (3D) inspection of local regions. PDBlocal provides an intuitive and easy-to-manipulate web page interface and some new useful functions. It can keep local regions flashing, display sequence text that is dynamically consistent with the 3D structure in local appearance under multiple local manipulations, use two scenes to help users inspect the same local region with different statuses, list all historical manipulation statuses with a tree structure, allow users to annotate regions of interest, and save all historical statuses and other data to a web server for future research. PDBlocal has met expectations and shown satisfactory performance for both expert and novice users. This tool is available at <http://labsystem.scuec.edu.cn/pdblocal/>.

Keywords: Biological macromolecule; 3D visualization; molecular local structure; web; Jmol.

1. Introduction

Computerized three-dimensional (3D) graphical visualization of biological macromolecular structures

is of fundamental importance in structural bioinformatics.¹ Experienced biologists might discover important biological information by intuitively

[‡]Corresponding author.

This is an Open Access article published by World Scientific Publishing Company. It is distributed under the terms of the Creative Commons Attribution 4.0 (CC-BY) License. Further distribution of this work is permitted, provided the original work is properly cited.

inspecting these 3D graphs. In particular, the inspection of these macromolecular 3D structures must usually be focussed on specific regions, such as knot regions,² disordered protein regions,³ metal sites,⁴ ligand binding sites,⁵ domains,⁶ and channels,⁷ because these regions may involve important biomolecular functions.

On the one hand, Protein Data Bank (PDB)⁸ currently functions as an authoritative, centralized archive of experimentally determined biological macromolecular structures for 3D structural visualization. It is jointly maintained by wwPDB,⁹ whose members include RCSB PDB and BMRB in the USA, PDBe in the EU, and PDBj in Japan. This archive is updated weekly and includes 127,823 entries as of 18 March 2017. Users can freely and easily find structural data related to their research fields through simple or complex queries on its website.

On the other hand, many 3D visualization tools have been developed for researchers. Improving of these tools mainly involves two aspects:

(1) First, as biological macromolecules often contain thousands or tens of thousands of atoms, the design of traditional visualization tools has centered around the highest interaction speed of the whole molecule for a long time.^{10,11}

(2) At the same time, these visualization tools can be classified as standalone (Rasmol,¹² Swiss-PdbViewer,¹³ Cn3D,¹⁴ Jmol,¹⁵ VMD,¹⁶ UCSF Chimera,¹⁷ Coot,¹⁸ AstexViewer,¹⁹ and so on) or web-based (JSmol,²⁰ NGL Viewer,²¹ iview,²² molmil,²³ Aquaria,²⁴ iCn3D (<http://www.ncbi.nlm.nih.gov/Structure/icn3d/icn3d.html>), and so on). Most of these tools provide additional functions. Examples are as follows:

- A. Though having a similar interface to Rasmol, Jmol is based on Java and can be easily integrated into a web page as a Java Applet.
- B. AstexViewer is another Java-based program that aims to assist with structure-based drug design.
- C. In the 3D scene of Jmol or iCn3D, distances between atoms can be measured by clicking these atoms, and each atom can be labeled.
- D. Structural studies in Aquaria can be based on precalculated all-against-all comparisons of Swiss-Prot²⁵ and PDB sequences.
- E. Currently, tools with a haptic interface, such as NDKmol (<http://webglmol.osdn.jp/>), have become very attractive.

However, an investigation among 30 researchers in biological macromolecular structure-related fields from two Chinese universities (Huazhong University of Science and Technology and South-Central University for Nationalities) indicated that although current mainstream molecular 3D visualization tools have high 3D interaction speeds and many useful additional functions, both standalone and web-based tools have shown limitations or inconveniences in the 3D inspection of local regions; these limitations or inconveniences are listed below:

(1) Traditional tools have already provided some 3D interaction means of inspecting local regions (Fig. 1). In practice, these means are useful and necessary to some extent, but they also have shortcomings, which are listed below:

- A. In traditional tools, the selected region can be translated out of the main body of a molecule by mouse interaction to reduce the visual distraction from other molecular regions. This interaction has the following shortcomings:
 - a. The relative position between the selected region and its neighbors in 3D space cannot be observed.
 - b. The molecular chain is cut off; thus, after many selections and mouse interactions in different regions, the chain could become indecipherable.
 - c. Only one local region can be manipulated by mouse interaction at a time, which is inconvenient in comparative studies of multiple regions.
- B. Users can choose to hide unselected regions to reduce visual distractions in the selected region. Under this circumstance, the relative position between the selected region and its neighbors in 3D space cannot be observed.
- C. The selected regions can be highlighted in different colors than other regions. However, the molecule could become indecipherable when too many colors have been used for different regions.
- D. The selected region can be haloed. However, the halo itself can create additional visual distractions.
- E. Apart from the selected region, other regions can be changed into Trace mode (a smooth curve interpolated between alpha carbons) to reduce visual distractions. Because Trace mode

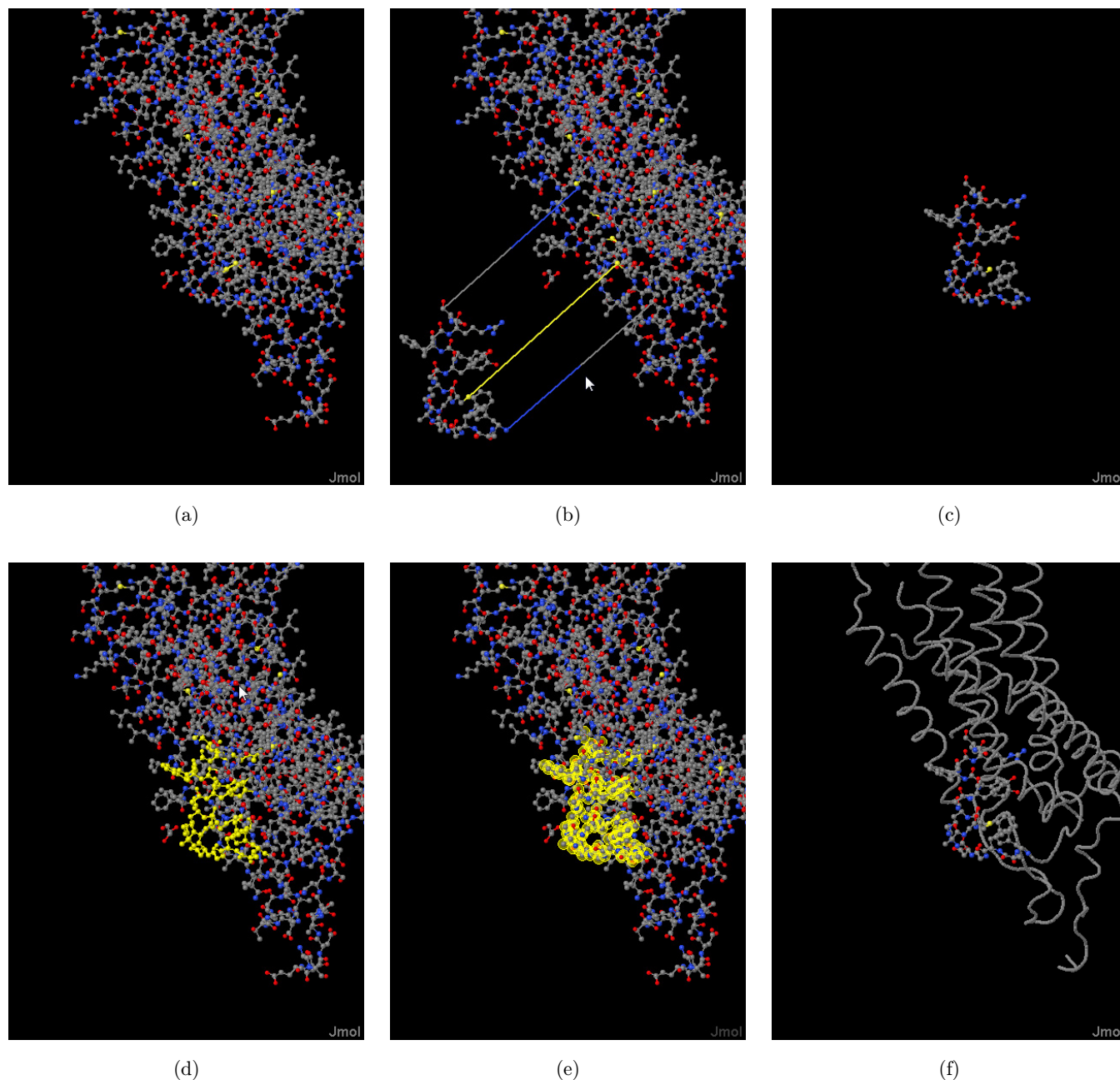


Fig. 1. Examples of 3D interaction means of inspecting local regions provided by traditional tools. (a) Original status of a molecular 3D structure (PDB ID: 5T04) in a 3D scene. (b) The selected region can be translated out of the main body of the molecule by mouse interaction. (c) The unselected regions can be hidden. (d) The selected region can be highlighted in a different color. (e) The selected region can be haloed. (f) Apart from the selected region, other regions can be changed into Trace mode.

can only show the skeleton of the main molecular chain, the relative position between atoms of the selected region and its neighbors in 3D space cannot be observed.

(2) According to the common view of biology, which is that sequence determines structure and structure determines function, 3D structure research usually requires the corresponding sequence as a reference.

In traditional tools, the reference features (Fig. 2) have weaknesses:

- A. For various reasons, such as inclusion of the monomer (nucleotide or amino acid) numbers, the reference sequence display may have fewer monomers per row or might rely on extensive use of the scrollbar, which does not align with the sequence-reading habit of most users.

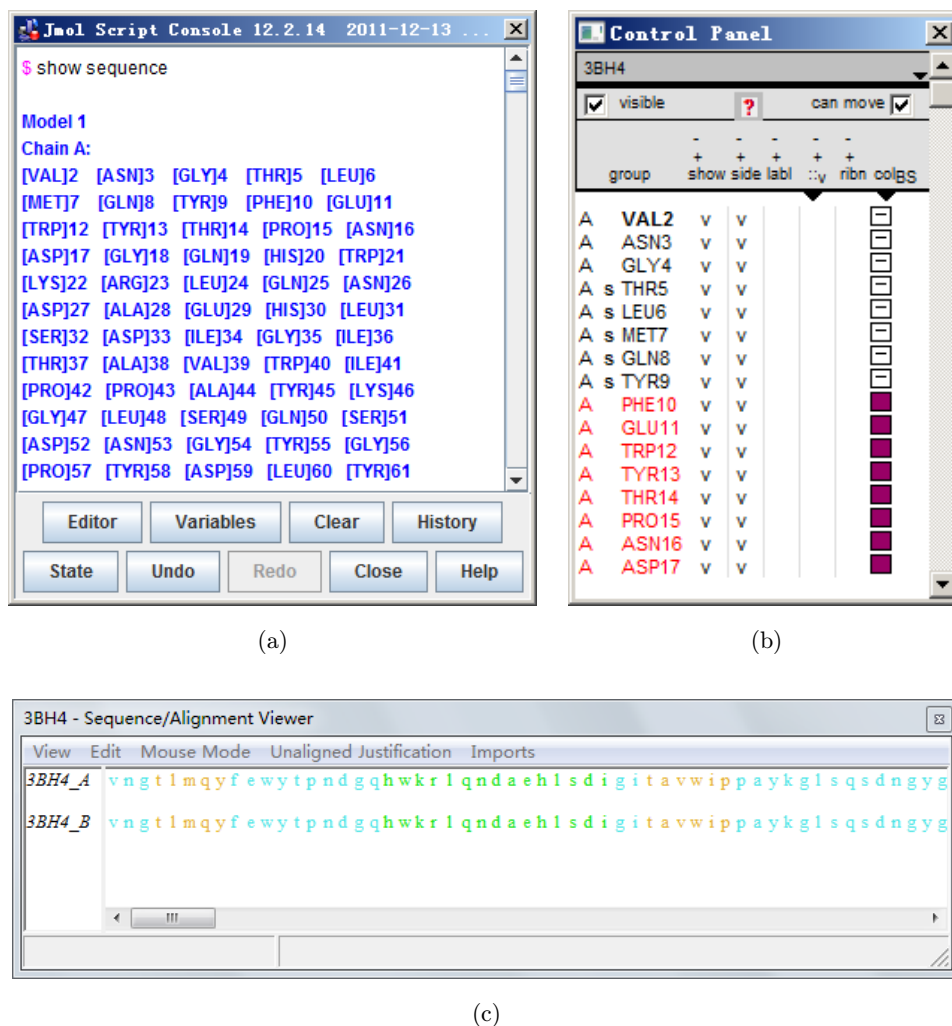


Fig. 2. Sequence reference features in three traditional molecular 3D visualization tools. (a) Sequence information display in the console window of Jmol 12.2.14. (b) Control panel of Swiss-PdbViewer 4.1.0. (c) Sequence/Alignment Viewer of Cn3D 4.3.1.

B. In some tools, such as Jmol, the reference sequence text does not dynamically change its status with the local interaction means listed in Fig. 1 (Fig. 2(a)). In other tools, such as Swiss-PdbViewer, the dynamic reference is not intuitive because the changed objects are not the sequence text but rather additional marks that are listed as elements of a matrix and represent different interaction means for different monomers (Fig. 2(b)).

C. It is inconvenient to compare multiple local regions, as they may be from one or multiple PDB files and are far away from each other in sequence text.

(3) In practice, switching and comparing historical manipulation statuses using Undo and Redo in traditional tools have imperfections. In these tools,

if new manipulation steps occur after an Undo, the undone statuses are lost and cannot be redone for comparisons with other statuses. Meanwhile, it is inconvenient for users to compare different historical statuses in only one 3D scene by Undo or Redo.

(4) In traditional standalone tools, the final 3D status can be saved as a text file for future research. However, it is inconvenient to save all historical manipulation statuses, or to annotate key statuses of interest, especially when studying complex macromolecules. None of the current web-based tools allows users to save manipulation statuses or annotations to the servers.

PDBlocal, a tool based on Jmol that uses web technologies to extend its functions for interactive inspections of biological macromolecular local 3D

structures, was developed to compensate for these limitations and inconveniences.

2. Methods

2.1. Running environment

PDBlocal is a web-based tool that adopts Browser/Server (B/S) architecture and selects JavaScript and PHP as the browser-side and server-side programming languages, respectively.

In addition to cross-platform-ability, user convenience, and potential as a collaborative platform,²⁶ the web-based running environment of PDBlocal has two other important advantages over standalone ones:

(1) It can easily save the history of a case study to the server and allows users to safely pause and continue their work on different computers at any time.

(2) On the one hand, Jmol is based on Java and can be easily integrated into web pages as a Java Applet, unlike most other widely used tools; on the other hand, Jmol provides a web-based JavaScript Library (Jmol.js), which can serve as a good Application Programming Interface (API) for extending functions with JavaScript programs. Therefore, because of its Jmol- and web-based environment, PDBlocal can inherit the powerful original functions of Jmol and focus on the extension of functions for interactive inspection of biological macromolecular local 3D structures through web technologies.

There are two obvious limitations of a Java Applet:

- (1) It is not supported in several common browsers, such as Firefox 52+ and Chrome 45+, because of security concerns.
- (2) Java Runtime Environment (JRE) must be preinstalled by users, and in some cases, additional configurations are needed.

First, investigation has shown that Java Applet can be perfectly supported by Internet Explorer and Safari, which are, respectively, the default browsers on Windows and Mac OS X, the two operating systems of most personal computers.

Secondly, the Jmol development team also currently provides JSmol, a JavaScript-based tool that provides almost the same API as Jmol but can be run on devices without JRE. However, Jmol

typically gives faster performance and smoother rotation than JSmol by a factor of 6–10, especially in cases rendering structures of 16,000+ atoms with surface creation.²⁰ As PDBlocal is designed to be able to load multiple macromolecular structures for comparisons of local regions, Jmol is more suitable for the application of PDBlocal than JSmol.

2.2. Main user interface

The main user interface of PDBlocal is composed of five functional areas (Fig. 3):

- (1) The “3D Scene” area is the place where molecular 3D structures can be displayed and be rotated, zoomed, or translated by mouse interaction.
- (2) The “Local Selection” area enables the selection of local regions of the loaded molecules.
- (3) The “Sequence Text” area dynamically displays molecular sequences in text mode.
- (4) The “Annotation” area enables annotation in a case study.
- (5) The “Main Control” area provides most manipulation functions, such as changing the color of local regions and switching to a historical status.

Unlike the original Jmol, user manipulations in PDBlocal (other than the rotation, zoom, or translation of 3D objects by mouse interaction in virtual scenes) are controlled through a variety of familiar and easy-to-use web page controls, such as buttons and drop-down menus, instead of the text commands and top (or pop-up) menus used by Jmol.

2.3. Functionality

PDBlocal provides the following functions to facilitate research on local regions:

(1) PDBlocal supports loading multiple structural files for comparative study based on Jmol. Moreover, the 3D structures and sequence texts of the molecules are displayed in the “3D scene” and “Sequence Text” areas, respectively, which can reference each other.

(2) In the “Local Selection” area, users can select molecular standard local regions (nonstandard groups will be discussed below) in several ways, such as setting numbers of the starting and ending amino acids. For easier reference research between

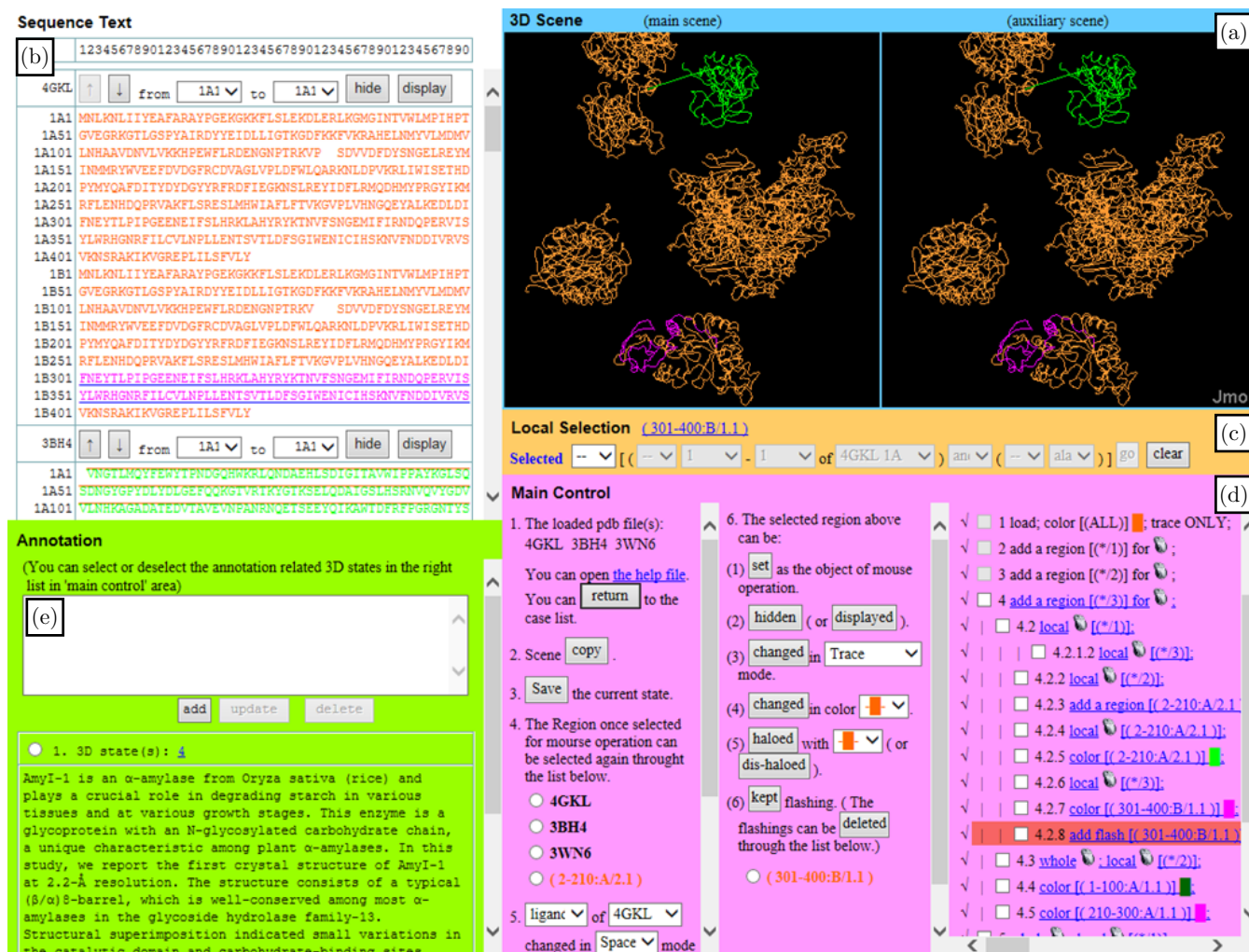


Fig. 3. Main user interface of PDBlocal. (a) "3D scene" area. (b) "Sequence Text" area. (c) "Local Selection" area. (d) "Main Control" area. (e) "Annotation" area.

3D structures and sequences, the basic objects of the local selection are monomers and not atoms. In the "Sequence Text" area, the text of the currently selected region will be synchronously underlined. Before further manipulation, the selected region can be increased or decreased, and finally, the region can be either continuous or discrete.

(3) In the "Main Control" area, users can perform several types of manipulations in which the selected local region in the 3D scene can be (A) set as the object of mouse interaction (including translation and rotation); (B) hidden; (C) changed in mode (including Spacefill, Ball & Stick, Sticks and Trace); (D) changed in color; (E) haloed; or (F) flashed. The first five manipulations provide the traditional means of 3D local inspection (Fig. 1). Keeping the local region flashing is a new useful

function based on the fact that human eyes are more sensitive to dynamic objects than to static ones. In its assistance with local inspection, the flashed or nonflashed regions in the structures do not need to be translated out of the main body, hidden, changed in color or mode, or haloed, and multiple regions can be simultaneously flashed; thus, the shortcomings of the local 3D interaction means of inspection in traditional tools (described in Sec. 1) can be mostly overcome. Flashed regions are saved as options in a list, and flashing can be stopped by removing the options.

(4) PDBlocal takes the following approach to overcome shortcomings b and c of the mouse interactions described in Sec. 1. First, local regions for mouse interaction in the 3D scene are also listed as options in the "Main Control" area. If multiple

structural files are loaded in a case study, there are also options to include each whole structural object from one file (represented by the filename), besides the local regions manually set by the user. Different local regions for mouse interactions can be quickly switched by clicking the options. Furthermore, for all options except filenames, the selected regions that they represent cannot overlap each other, and they can only be removed indirectly through the Undo function (discussed subsequently).

(5) Ligands, ions, and other nonstandard groups of 3D structures in the 3D scene are designed to be selected, hidden, and changed in mode or color in the “Main Control” area to balance user demand and simplify the operation of the graphical interface. Even though the whole structural object (from one file) and its nonstandard groups can be operated by mouse or flashed, the two types of interactions cannot be performed for each nonstandard group alone.

(6) In the “Sequence Text” area, several measures have been adopted to improve the visualization of the reference sequence texts:

- A. The sequence text contains monomers, but not their numbers, and each monomer is represented by a single letter so that more monomers can be displayed per row.
- B. The sequence text can be viewed as a matrix and each row contains 50 monospaced characters (including monomer letters or placeholders). The numbers of first characters of rows are listed on the left of the matrix, and each of these numbers is made up of three parts, which are the related numbers of model, chain, and monomer from original structural files. Column numbers are listed as a row above the matrix and are kept visible when the matrix is scrolled with a vertical scrollbar. These measures can help users find the positions of local regions.
- C. Users can hide parts of the sequence text, which are replaced by short links. When the mouse cursor hovers over these links, the text pops up. When multiple files are loaded, their display order can be changed as needed. These measures make it easier to compare nonadjacent local regions.
- D. For further assistance with 3D inspection, the text display of local regions can be dynamically

changed with special local manipulations, as listed below:

- a. The currently selected molecular local region will be underlined in the sequence text (e.g., DSLAPEDGSHR).
- b. When it is manually set as the object of mouse interaction in the 3D scene, the local region will be overlined in the sequence text (e.g., DSLAPEDGSHR), and the overline will be set to a different color each time.
- c. The local regions in the sequence text and 3D scene will be the same color.
- d. The regions that are hidden in the 3D scene will be replaced with ellipsis points in sequence text (e.g., ...).
- e. The regions that are haloed in the 3D scene will be struck through in the sequence text (e.g., ~~DSLAPEDGSHR~~). The middle lines and halos will be the same color.
- f. When flashing is set in the 3D scene, a flashing block is added over the local region in the sequence text, and the block will be a different color each time.
- g. When the mode of a local region changes, the background color of the region in the sequence text will change (e.g., **DSLAPEDGSHR**) and different background colors will correspond to different modes.

(7) The “3D Scene” area includes two 3D scenes: one main and one auxiliary. The above-mentioned local manipulations act on the former. The current 3D status of the main scene can be duplicated in the auxiliary scene. Then, users can perform different interactions in the two scenes and can simultaneously inspect the same local region with different statuses, such as viewing angles, colors, and modes (Fig. 4). Furthermore, in the auxiliary scene, the other original functions of Jmol, such as changing the color of oxygen atoms, can be performed by issuing commands in the console opened through a pop-up menu (Fig. 4), which is shielded in the main scene to avoid influencing some special functions of PDBlocal.

(8) In PDBlocal, if new manipulations occur after an Undo, the undone statuses can still be saved, which is different from the settings in the original Jmol. The status involves the “3D Scene”, “Sequence Text” and “Main Control” areas. All the statuses are represented by nodes in a tree structure, so that their entries are especially displayed in

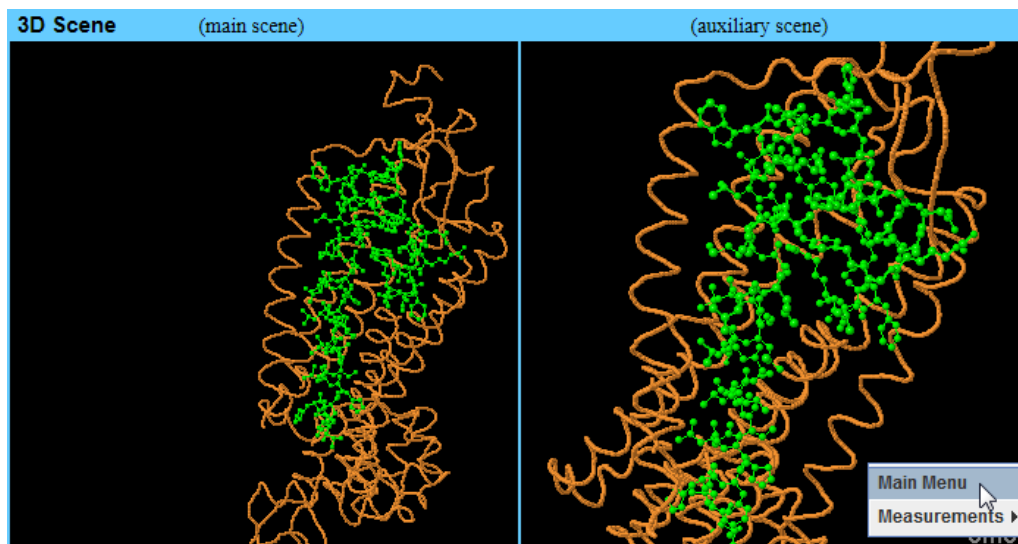


Fig. 4. A demonstration of the two 3D scenes in PDBlocal.

the form of a tree structure for easy identification (the right column of Fig. 3(d)). Commands from the manipulation steps are included in the entries, which can help users review the manipulation process. Users can switch to any status directly and can conveniently compare different 3D statuses with the assistance of the auxiliary scene. In addition, the useless status (along with its following statuses) can be removed.

(9) In the “Annotation” area, 3D statuses (represented by links) of local regions of interest or whole structures in the scene can be annotated with text.

(10) From the beginning of a case study, the structure files, historical manipulation statuses, and annotations can be saved to the server in real time for future research. When a user enters a former case, the complete history of the case will be restored, and the case study can be continued.

2.4. Solutions to key problems

2.4.1. Interface integration

All contents of the five functional areas in the main user interface cannot be displayed simultaneously on a full screen with mainstream resolutions, and refreshing the whole web page could impair the user’s inspection or interaction. Therefore, three web technologies have been introduced:

(1) The web page has been divided into several frames. Then, some content, such as 3D scenes,

can be visible whereas others can be displayed with scrollbars. Furthermore, refreshing a frame will not influence other frames.

- (2) Asynchronous JavaScript and XML (AJAX) have been used to speed up page refresh when the browser interacts with the server.
- (3) With Cascading Style Sheets (CSS), some information pops up when the mouse cursor hovers over special links.

2.4.2. Dynamic reference of sequence text

The dynamic reference of sequence text in PDBlocal has been introduced in Sec. 2.3. According to the HyperText Markup Language (HTML) rule, however, it could hardly be realized by changing text styles (including font color, background color, underline, middle line, overline, hiding and flashing) of regions directly in sequence text because these regions may overlap each other. Therefore, the HTML technique of DIVision (DIV) was flexibly applied:

- (1) As a whole, different text styles are involved in different DIV blocks and are simultaneously presented by completely overlapping these blocks to more easily change text styles.
- (2) The two styles of font color and hiding directly affect the sequence text, which is involved in a single DIV block.
- (3) Other styles have their respective DIV blocks in which monomer letters are replaced by spaces to prevent double vision caused by overlap.

- (4) The DIV blocks are arranged in an appropriate order to prevent them from obscuring one another. For example, the DIV block of background color style should be placed at the bottom.
- (5) For the style of flashing, the DIV block, in which related local regions are set with different background colors, rotates between being hidden and displayed.

2.4.3. Tree-structured display of entries of historical manipulation statuses

Similar to the dynamic reference of the sequence text, the tree-structured display of entries of historical manipulation statuses is also difficult to realize using regular HTML elements. Hence, the entries are specially numbered and listed as follows:

(1) The entry number is formatted as $a_1.a_2.a_3, \dots, a_n$ (a_1, \dots, a_n are integers; $a_n \geq 1$ when $n = 1$; $a_1, \dots, a_{n-1} \geq 1, a_n \geq 2$ when $n > 1$). For example, “3”, “5.2” and “2.1.2” are correct numbers, but “0” and “2.1” are incorrect. Meanwhile, number relationships between each entry and its direct successors in the tree structure are designed as shown in Table 1. For example, the entries numbered “2.6”, “2.5.2”, and “2.5.1.2” are direct successors of the entry numbered “2.5”. If a_n is equal to 1 when $n > 1$ in the entry number format just described, it could be difficult to number more than two direct successors of each entry according to the user’s reading habit (for example, the direct successors of the number “2.5” could be only “2.6” and “2.5.1” in this situation).

(2) The list of entries is designed by a table of contents, which is a common form of tree structure. This list is based on the following rules:

A. One row contains one entry.

Table 1. Number relationships between each entry and its direct successors in the tree structure.

Entry	Number
E_1	$a_1.a_2.a_3. \dots a_n$
The first direct successor of E_1	$a_1.a_2.a_3. \dots (a_n + 1)$
The second direct successor of E_1	$a_1.a_2.a_3. \dots a_n.2$
The third direct successor of E_1	$a_1.a_2.a_3. \dots a_n.1.2$
The fourth direct successor of E_1	$a_1.a_2.a_3. \dots a_n.1.1.2$
...	...

- B. Successors are all listed below their precursors.
- C. Among the direct successors of an entry, a former one lies below a latter one, and the last one is next to the entry.
- D. In a row, vertical bars may be placed on the right of each entry to further improve visualization, and the number of vertical bars is equal to the number of the periods in the entry number (e.g., “| 2.2” and “| | 3.2.6”).

An example based on the number and list designs above is shown in the right column of the “Main Control” area in Fig. 3.

2.4.4. Web-based real-time saving of historical statuses after each manipulation step

The historical manipulation statuses of PDBlocal in comparison to those of traditional tools have more entries (listed as a tree structure), involve more contents (displayed in the “3D Scene”, “Sequence Text”, and “Main Control” areas), and are saved to the server in real time through the Internet after each manipulation step. Therefore, web-based saving of historical manipulation statuses could be greatly influenced by the running conditions of the client, server, and network. The following methods have been adopted:

(1) AJAX is applied to save status data to the server asynchronously in the background without hindering the behavior and display of the existing page. Meanwhile, when a status is saved completely, a mark “√” will emerge before its entry so that user can be alerted upon exit if all the statuses have been saved.

(2) The saved data of a status do not involve the whole status. For example, only the 3D transforming status of a local region for mouse interaction or the whole 3D object in a scene will be saved after a mouse interaction, and neither will be saved after other manipulations. This method can greatly reduce data transfer on the network. On the other hand, when the user switches to a status, an opposite scanning of its precursors is needed to restore the status. However, the restoration can be swift because it runs only on the client side.

(3) If a certain 3D object successively undergoes multiple mouse interactions, only the last status will be saved.

3. Results and Discussion

This section summarizes the practical advantages of the new methods and techniques in PDBlocal compared with other tools through a case study and some feedback from users.

G protein-coupled receptors (GPCRs) are known for their major roles in signal transduction in cells. GPCRs are characterized by an extracellular N-terminus, followed by seven transmembrane α -helices (TM-1 to TM-7) connected by three intracellular (IL-1, IL-2, and IL-3) and three extracellular (EL-1, EL-2, and EL-3) loops, and finally an intracellular C-terminus. Therefore, a GPCR can be divided into at least 15 local regions. These local regions can be conserved or variable, and their inspection or comparison can be helpful in many studies. In a case study for the comparison of the GPCRs from two PDB files numbered 5T04 and 5G53, PDBlocal shows distinctive advantages compared with other standalone or web-based tools, such as Rasmol, Jmol, and iCn3D, in several aspects:

(1) *Interactive inspection of biological macromolecular local 3D structures*

iCn3D and other tools usually choose Ribbon²⁷ or Trace as the default display mode of 3D models to balance the inspection of both local regions and whole models in a 3D scene. As studies on GPCRs usually need to focus on the specificities of local regions, the Ball & Stick mode is often used to display the 3D statuses of residues or atoms. These 3D statuses, however, can hardly be recognized because there are too many balls and sticks. In PDBlocal, keeping local regions flashing can easily solve this problem, as they are similar to flashing outdoor LED advertising sign, which can usually be recognized at night no matter how complicated the lightened background is.

(2) *Reference of the sequence text*

Unlike Jmol and most other tools, iCn3D supports a simple dynamic reference between sequence and structure, which involves just one manipulation (a change in color) of the local regions. In PDBlocal, however, the local regions can be synchronously changed in sequence and structure with more local manipulations, such as hiding, changing the mode, haloing, flashing, and setting the object of mouse interaction; moreover, any nonadjacent local regions

in the sequence text can be centrally compared, even though each sequence text of the two PDB files has more than 900 monospaced characters (including monomer letters or placeholders).

(3) *Inspection of a local region with different 3D statuses*

In the inspection of the TM-7 region in one GPCR, the comparison of TM-7 regions in two GPCRs, or studies on the 3D relationship between a GPCR and its ligand, the 3D status needs to be copied from one scene to another to synchronously inspect the same local region with different statuses, such as viewing angles, colors, and modes. Copying can involve the process of saving the original 3D scene, re-opening the same tool, and restoring the saved 3D scene in Rasmol, Jmol, and iCn3D, but it can be easily performed from the main scene to the auxiliary scene by clicking a button in PDBlocal.

(4) *Switching and comparing historical manipulation statuses*

In PDBlocal, all the historical manipulation statuses are organized in a tree structure; after an Undo, a new manipulation will generate a new branch of the tree and the undone statuses can still be saved. This design can help users directly switch to any historical status and compare different historical statuses with the assistance of the auxiliary 3D scene. Jmol, iCn3D, and other tools can only use Undo/Redo or Save/Restore to switch or compare historical statuses. iCn3D lacks some special functions for studying local regions, such as translating local regions by mouse, so it usually has a relatively uncomplicated 3D scene and does not need to organize historical statuses through a tree structure.

(5) *Saving a case study*

In Rasmol, Jmol, and iCn3D, the current 3D status can be saved or exported as the image for future research. However, these functions are not enough for this case study of GPCRs, which may be very complex with too many 3D objects of local regions and a long history of manipulations. In PDBlocal, 3D statuses of interest can be annotated with texts through B/S technologies, and all the historical manipulation statuses and annotations can be automatically saved to the server, which can allow users to pause and continue their research more safely and conveniently compared with other tools.

(6) Inheriting and extending traditional functions

Unlike iCn3D, another newly developed tool, PDBlocal, was developed on API of the traditional tool, Jmol. Therefore, PDBlocal not only has its specific functions but also has inherited the powerful original functions of Jmol. For example, in this case study on GPCRs, novice users can easily select all TM-7s in the GPCRs through web page controls and manually superimpose them by mouse in the 3D scene for comparison, and experts can superimpose these local regions more accurately through Jmol commands. As another example, in the “Local Selection” area of PDBlocal, users can select only standard residues, but any atom combination can also be selected through Jmol commands.

PDBlocal has been provided to 30 researchers (mentioned in Sec. 1) to further demonstrate the practicability of its new methods and techniques. These researchers, both expert and novice users, consider PDBlocal to have good performance at avoiding the limitations or inconveniences of other tools, which have been listed in Sec. 1.

The application result also shows one limitation of PDBlocal, which is that JRE must be preinstalled and configurations may be needed. Section 2.1 has already discussed the merits and demerits of JRE in PDBlocal. PDBlocal has provided detailed help files and videos on its homepage to alleviate the problems caused by the limitation of JRE for new users. In later applications, all users with basic computer skills can easily use this tool under the guidance of these help files or videos.

The feedback from some researchers indicates that with the rapid development of bioinformatics, many practically improved methods have been introduced for researching local regions of biological macromolecules, such as target-template alignment²⁸ and detection of ligand binding sites²⁹; however, a careful and intuitive inspection of local 3D structures plays an important role.

Moreover, this tool can be easily integrated with other web-based bioinformatics tools for molecular structure research and will be extended to include new functions as needed.

Acknowledgments

This work was supported by the International Science and Technology Cooperation Key Projects of the Chinese Ministry of Science and Technology

(2009DFB30340), National Major Projects of China (2016ZX08010004-004), National Natural Sciences Foundation of China (81373379), Natural Science Foundation of Hubei Province of China (2012FFB07402), and Fundamental Research Funds for the Central Universities in China (South-Central University for Nationalities, CZY12004). We also thank Drs Szabolcs Makai and Javeed Choudary for critically reading this paper.

References

1. J. L. Moreland, A. Gramada, O. V. Buzko, Q. Zhang, P. E. Bourne, “The Molecular Biology Toolkit (MBT): A modular platform for developing molecular visualization applications,” *BMC Bioinform.* **6**, 21 (2005).
2. W. R. Taylor, “Protein knots and fold complexity: Some new twists,” *Comput. Biol. Chem.* **31**, 151–162 (2007).
3. H. Huang, A. Sarai, “Analysis of the relationships between evolvability, thermodynamics, and the functions of intrinsically disordered proteins/regions,” *Comput. Biol. Chem.* **41**, 51–57 (2012).
4. C. Andreini, I. Bertini, G. Cavallaro, R. J. Najmanovich, J. M. Thornton, “Structural analysis of metal sites in proteins: Non-heme iron sites as a case study,” *J. Mol. Biol.* **388**, 356–380 (2009).
5. M. Hernandez, D. Ghersi, R. Sanchez, “SITEHOUND-web: A server for ligand binding site identification in protein structures,” *Nucleic Acids Res.* **37**, W413–W416 (2009).
6. C.-H. Tai, V. Sam, J.-F. Gibrat, J. Garnier, P. J. Munson, B. Lee, “Protein domain assignment from the recurrence of locally similar structures,” *Proteins* **79**, 853–866 (2011).
7. T. B. Masood, S. Sandhya, N. Chandra, V. Natarajan, “CHEXVIS: A tool for molecular channel extraction and visualization,” *BMC Bioinform.* **16**, 119 (2015).
8. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne, “The Protein Data Bank,” *Nucleic Acids Res.* **28**, 235–242 (2000).
9. H. Berman, K. Henrick, H. Nakamura, “Announcing the worldwide Protein Data Bank,” *Nat. Struct. Biol.* **10**, 980 (2003).
10. T. Can, Y. Wang, Y. F. Wang, J. Su, “FPV: Fast Protein Visualization Using Java 3D™,” *Bioinformatics* **19**, 913–922 (2003).
11. C. Bajaj, P. Djeu, V. Siddavanahalli, A. Thane, TexMol: Interactive visual exploration of large flexible multi-component molecular complexes,

- Proc. Ann. IEEE Visualization Conf.*, Los Alamitos: IEEE Computer Society, Austin TX USA, pp. 243–250 (2004).
12. H. J. Bernstein, “Recent changes to RasMol, recombining the variants,” *Trends Biochem. Sci.* **25**, 453–455 (2000).
 13. N. Guex, M. C. Peitsch, “SWISS-MODEL and the Swiss-PdbViewer: An environment for comparative protein modeling,” *Electrophoresis* **18**, 2714–2723 (1997).
 14. Y. Wang, L. Y. Geer, C. Chappey, J. A. Kans, S. H. Bryant, “Cn3D: Sequence and structure views for Entrez,” *Trends Biochem. Sci.* **25**, 300–302 (2000).
 15. A. Herráez, “Biomolecules in the computer: Jmol to the rescue,” *Biochem. Mol. Biol. Educ.* **34**, 255–261 (2006).
 16. W. Humphrey, A. Dalke, K. Schulten, “VMD: Visual molecular dynamics,” *J. Mol. Graph.* **14**, 33–38 (1996).
 17. E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, T. E. Ferrin, “UCSF Chimera — A visualization system for exploratory research and analysis,” *J. Comput. Chem.* **25**, 1605–1612 (2004).
 18. P. Emsley, B. Lohkamp, W. G. Scott, K. Cowtan, “Features and development of Coot,” *Acta Crystallogr. D Biol. Crystallogr.* **66**, 486–501 (2010).
 19. M. J. Hartshorn, “AstexViewer™: A visualisation aid for structure-based drug design,” *J. Comput. Aided Mol. Des.* **16**, 871–881 (2002).
 20. R. M. Hanson, J. Prilusky, Z. Renjian, T. Nakane, J. L. Sussman, “JSmol and the next-generation web-based representation of 3D molecular structure as applied to Proteopedia,” *Israel J. Chem.* **53**, 207–216 (2013).
 21. A. S. Rose, P. W. Hildebrand, “NGL Viewer: A web application for molecular visualization,” *Nucleic Acids Res.* **43**, W576–W579 (2015).
 22. H. Li, K.-S. Leung, T. Nakane, M.-H. Wong, “iview: An interactive WebGL visualizer for protein–ligand complex,” *BMC Bioinform.* **15**, 56 (2014).
 23. G.-J. Bekker, H. Nakamura, A. R. Kinjo, “Molmil: A molecular viewer for the PDB and beyond,” *J. Cheminform.* **8**, 42 (2016).
 24. S. I. O’Donoghue, K. S. Sabir, M. Kalemanov, C. Stolte, B. Wellmann, V. Ho, M. Roos, N. Perdigão, F. A. Buske, J. Heinrich, B. Rost, A. Schafferhans, “Aquaria: Simplifying discovery and insight from protein structures,” *Nat. Methods* **12**, 98–99 (2015).
 25. E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, P. Bansal, A. J. Bridge, S. Poux, L. Bougueleret, I. Xenarios, “UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: How to use the entry view,” *Methods Mol. Biol.* **1374**, 23–54 (2016).
 26. F. Mwalongo, M. Krone, G. Reina, T. Ertl, “State-of-the-art report in web-based visualization,” *Comput. Graph. Forum* **35**, 553–575 (2016).
 27. M. Carson, C. E. Bugg, “Algorithm for ribbon models of proteins,” *J. Mol. Graph.* **4**, 121–122 (1986).
 28. A. Barbato, P. Benkert, T. Schwede, A. Tramontano, J. Kosinski, “Improving your target-template alignment with MODalign,” *Bioinformatics* **28**, 1038–1039 (2012).
 29. C. H. Ngan, D. R. Hall, B. Zerbe, L. E. Grove, D. Kozakov, S. Vajda, “FTSite: High accuracy detection of ligand binding sites on unbound protein structures,” *Bioinformatics* **28**, 286–287 (2012).