XmMol User's Manual

XmMol ©, P. Tuffery†, 1993-2001.
Version 3.10

†: U436 INSERM, Equipe de Bioinformatique Génomique et moléculaire, Université Paris 7, Tour 53, 1^{er} étage, 2 place Jussieu, 75251, Paris cedex 05.

Reference:
"XmMol: An X11 and Motif program for macromolecular visualization and modelling.", P. Tuffery, J. Mol. Graph., 1995, 13, p67-72.

http://duc.urbb.jussieu.fr/XmMol.html

June 19, 2001

Contents

Introduction

What is XmMol?

XmMol is a desktop tool for (macro)molecular visualisation, that tries to be as multi-user platform independent as possible, and as open as possible to user specific requests via customisation/extension facilities. Despite its main focus is visualisation, XmMol also support in its core some modelling tools. XmMol was developed in C, and is based upon X11 and Motif. Thus, XmMol can also work through networks, or/and more generally on any display supporting the X11 protocol.

All graphics are performed using the drawing capabilities of the X11 library. To get interactivity, the standard molecule display is, currently, wire-frame. However, in order to keep some possibility of generating cartoons including space filling models, some interface with programs such as MolScript (Kraulis, 1991), Raster3D (Bacon & Anderson, 1988 ; Merritt & Murphy, 1994) or QTree (Martin, 1993) have been implemented.

A main particularity of XmMol is its ability to easily be interfaced with external applications. This can be made at two levels:

(i) a simple macro parsing of commands sent to the shell allows to output and/or (re-)input coordinate files. This can for example be used in conjunction with BABEL (Walters & Stahl, 1992) to make XmMol load most of the file format supported by converting them toward PDB and loading them.

(ii) a dialog library allows external programs to communicate with XmMol using a protocol based upon software interrupts (POSIX). Having linked a program with the communication library, this program can be run as a "delegate" task of XmMol: you ask XmMol to run it, which triggers XmMol to fork it as a delegate task able of dialoging with XmMol.

This level gives access to most of the parameters of each file loaded, but also to most of the parameters of the viewing and drawing processes. Typical applications can be to use XmMol as a visualisation tool for minimizers or molecular dynamics, but also through this process, you can develop super-imposition modules or write your own external renderer (supporting ball and sticks for example). Finally, the use of shared memory is also supported for some fields.

Some history about XmMol

XmMol development started in 1993 since the author was somewhat upset not to have a correct and quick macromolecular visualisation tool running on HP and providing a display on its X terminal. Moreover, having spent some time in different labs, and thus on different Unix systems, using different graphics packages, the author's wish was also to get some platform independent tool.

Version 1 (1993) resulted in a molecular visualisation tool for PDB files, and XmMol (mac) files, based upon X11, but running on all the Unix platforms the author could test.

Version 2 (1994) came with facilities implemented to take profit from external programs, either using the possibility to send Unix shell commands from XmMol, or using a protocol allowing a direct command exchange with processes external to XmMol. For the latter procedure, the choice was to define a communication protocol based on POSIX (again this should be supported by most machines at this time), and to make the dialog possible between two processes, one being a delegate task, forked from XmMol. In theory, several delegates could be used at the same time.

Version 3 (1996) is mainly a generalisation of the drawing and colouring schemes supported. Each file and each fragment of each file can be associated with its own molecular type, rendering and colouring styles. Thus it is possible to display one file/fragment as a ribbon coloured according to its secondary structure, another one as wireframe coloured according to the user interactive choice, another as a trace coloured by residue types, etc. At the same time, the interface of XmMol with programs such as MolScript, Raster3D (and also in a restrained way with QTree) make it easy to generate static cartoons including various space filling representations.

Version 3.1 (1997) supports a generalized interface towards external delegates. Also incorporates enhanced interface.

What XmMol cannot do

- Since XmMol requires multiprocessing, it has not been implemented on PC running MS/DOS or early versions of Windows. Neither was it implemented on Apple's machines. Even if things are moving, such implementations are not currently considered.

- XmMol current version does not support sophisticated molecular representations. In a general way, up to version 3.xx, it does not support currently any built-in surface rendering.

- XmMol has only a few modelling tools implemented. It has no built-in or distributed geometric builder for proteins of nucleic acids, and no tools to generate mutations or such modelling operations.

- XmMol does not incorporate any built-in module allowing energy evaluation.

As a summary, there are many other things that XmMol's core will not do. However, it is possible to implement many of them as external delegate tasks. If you are not a developer or if you are just looking for a "ready to use" molecular visualisation program, it is right to think to XmMol as a desktop tool that allows an easy glance at several of your structures together, and allows to get or prepare some views for your iconography.

Registering XmMol

XmMol is freely available for licensed non profit institutions (for any commercial use, please contact the author). The form can be loaded at (anonymous ftp):

duc.urbb.jussieu.fr

On receipt of the license, instructions on how to down-load run-times of the software for a series of platforms are sent: HP 720/735, SGI Crimson (system 4), IBM RS 6000 (AIX), Dec Alpha, SUN, Linux PCs. Since XmMol has many possibilities to be interfaced with external applications, the sources will not, in general, be distributed. However, for specific platforms, or for specific needs, you can contact the author:

tuffery@urbb.jussieu.fr

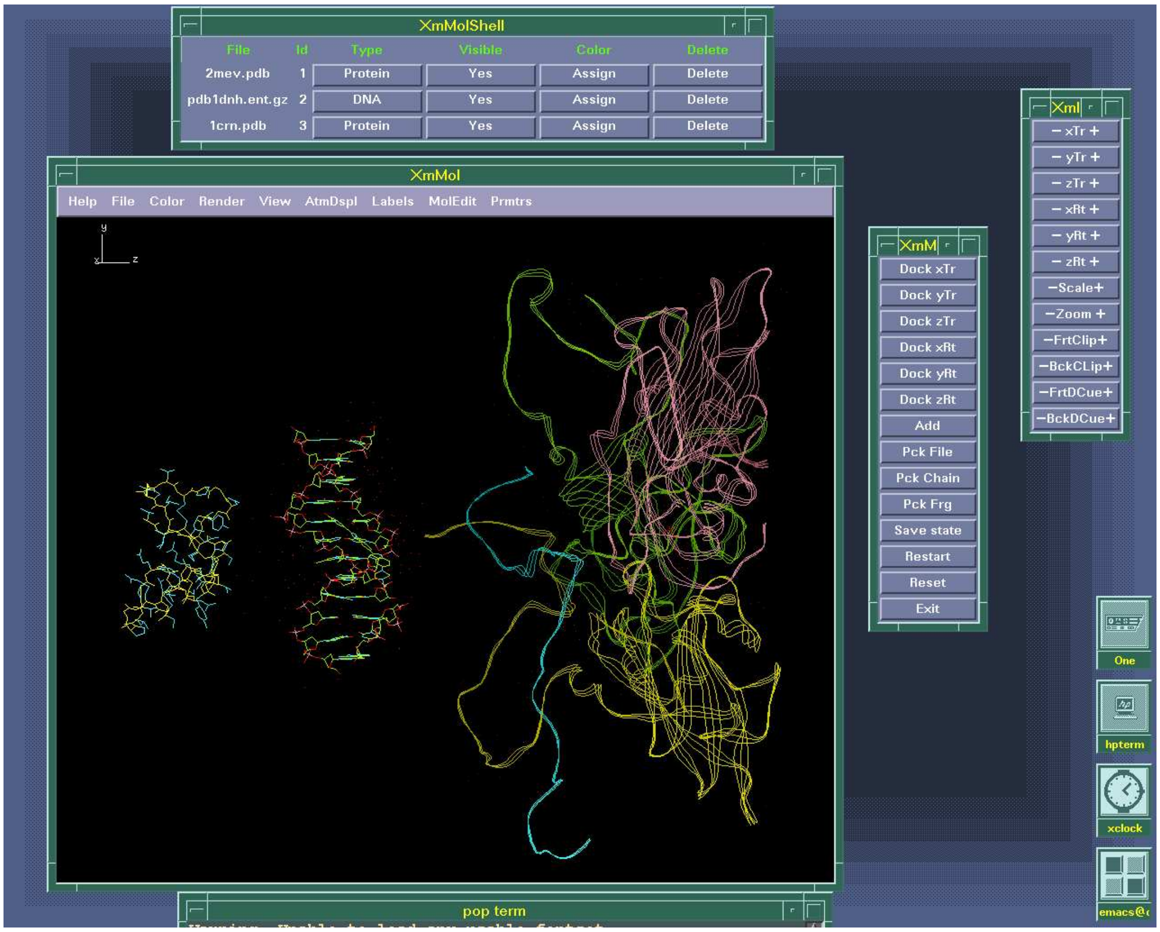
Disclaimer

XmMol is not a freeware, not a shareware. It is only distributed on request, and only to non-profit organisations that have registered. It remains the property of the author. No rights for distribution, modification, inclusion in other programs without prior consent of the author.

The program is distributed as is, WITHOUT ANY WARRANTY OF ANY KIND. Should the software prove defective, the users have to assume the whole responsibility for any necessary servicing, repair or correction.

1 A sample introductory session

This example relates how to view together the Picornavirus coat protein (pdb2mev.ent.gz), the crambin (1crn.pdb) and a DNA (pdb1dnh.ent), starting from the PDB files.



- Start XmMol and load the files by invoking:

  > XmMol -ipdb pdb2mev.ent.gz 1crn.pdb pdb1dnh.ent

  Note that XmMol will automatically gunzip the file pdb2mev.ent.gz. Also, you just need one argument -ipdb (all the names until the next -*option* are considered as PDB input files).

  Alternatively, you can start XmMol alone, and make three call to the **File:Open:PDB** menu to load the files (you can use the keyboard shortcut <Alt p>).

- Toggle the **Visible** items of pdb1dnh.ent and 1crn.pdb to **No**.

  Then use the item **Render:RenderType:Ribbon** of the main window menu bar to get a spline representation of pdb2mev.ent. Alternatively, you can use the keyboard shortcut <Alt r>.

Use the **Color:ByChains** item of the main window menu bar to get different colors assigned to each chain of pdb2mev.ent. Alternatively, you can use the keyboard shortcut <Ctrl c>

- Toggle the **Visible** item of pdb2mev.ent to **No** and that of 1crn.pdb to **Yes**.

  Use the **Color:BB/Res** to get a coloration different for backbone and side chains, or simply press the key combination <Ctrl b>.

- Toggle the **Visible** items of pdb2mev.ent and pdb1dnh.ent to **Yes**.

  Move the cursor of the mouse within the graphic window. Press the middle button and move the mouse to rotate the files. Relax the button.

  Press the Ctrl and Shift keys and the middle button and move the cursor up and down to adjust the scaling.

- Use the <ALT c> keyboard shortcut or the **AtmDspl:QckSelect:Hide Pick Chain** menu item and pick one chain of the pdb2mev.ent file. Then use <ALT r> to make the others chains rendered as trace. Use <ALT a> to display all the chains of the file. Alternatively, use the Frg RndrAs item in the file summary window, and select the type of display you want for the different fragments of the pdb2mev.ent file.

  You will note that the files 1crn.pdb and pdb1dnh.ent are somewhat overlapping. To distinguish between the two files, use the **MolEdit:Dock** item of the menu bar. A Dock window will appear.

- Press the **Pck File** button of the dock panel, then click one atom of the pdb1dnh.ent file.

  Move the mouse pointer within the **Dock yTr** button of the dock panel. Click the right button and hold it: the pdb2mev.ent starts moving relatively to the two other files. The magnitude of each elementary translation is controlled by the position of the mouse pointer relatively to the centre of the **Dock yTr** button. You can go outside the button to get wider moves, or go any side. The move keeps on until you relax the button.

  Act in a similar manner to get the appropriate rotations and translations according to the X and Z axes. If you need to modify the view of the three files all together, you can just act the same way by using the buttons of the general 3D control panel of XmMol.

  Move the mouse pointer into the graphic window, press the middle button and move the mouse. Note that the three files rotate.

  Toggle the Caps-Lock key. Again, move the mouse pointer into the graphic window, press the middle button and move the mouse. Note that only the file pdb1dnh.ent is now affected.

  Then press the **Save state** button, and the **Exit** button.

- If you wish, use the left mouse button to pick some atoms. Play with the **Labels:Labels:short** menu item to modify the atom label format.

- Toggle the picking to distance measurement by selecting the **Labels:Pick:Distances** menu item, or alternatively press <Ctrl d>. Then pick two atoms consecutively. They can belong to different files. The distance between the two atoms will appear.

- Use the **File:Snapshot:PostScript** item of the menu bar. Toggle the **PostScript printer** to **Color** if you have a PostScript color printer. Choose a name for the postscript file, and press OK. You now have a PostScript file of your view.

- To exit XmMol, use the **File:Quit** menu or press <Alt q>.

2 XmMol basics

2.1 Running XmMol

Once installed, you can call XmMol by simply invoking:
XmMol
or
XmMol <command line options>

If you wish to run XmMol in background, you man first have to configure your tty to allow the generation of SIGTTOU signals when background jobs attempt output. This can be done for example using the **stty -tostop** shell command.

2.1.1 command line options

The main basic command line options are:

-ipdb <File name(s)> : to input PDB files (as many as you want, but must be preceded by at least one -ipdb.

-ixmol <File name(s)> (also equivalent to -imac) : to load coordinate file in the mac format (mac is the internal file format of XmMol). For example:

XmMol -ixmol toto.mac tata.mac -ixmol tutu.xmol -ipdb RNA1.pdb DNA2.pdb.gz

REMARK: compressed, packed or gzipped files will be automatically uncompressed. This requires that you can access the "zcat", "pcat" or "gzcat" commands, and have rights to write in /tmp.

-icfg <File name> : to input a configuration file. On entry, if this option is not specified, XmMol with try to load a default configuration file located as $XmMOLDOCPATH/XmMol.cfg. If it is not found, XmMol will use its internal default values. Note that there is no necessity to input a configuration file to get XmMol work.

-icmd <File name> : to input a command file. Such files can be used to configure XmMol menubar and to install items corresponding to external commands.

-dlgt <File name> : to invoke a delegate as soon as XmMol starts.

-bw : to run black and white mode.

-noPixmap : to run without double buffer.

-dCueRange <range> : to specify the contrast accessible by the depth cueing. range specifies the intensity difference for front and back colors. Default is 40000. You can go up to 64000.

-gcmap : to use XInstallColormap instead of XSetWindowColormap. Avoid color change when not in the XmMol windows. OBSOLETE since 2.0.

-wcmap : to use XSetWindowColormap instead of XInstallColormap. Can be interesting on SGI since they support more than one Cmap together). The default for XmMol since 2.0.

-dfltcolors : to force XmMol NOT to alloc its own colormap but to use the default colormap. Note that XmMol will use the colors as are in the colormap. Be prepared to some surprises. This option can however be usefull in some cases where you have a low level of memory in your X station, or if you cannot allocate any colormap (should not be the case if the huge majority of configurations).

-fullcmap : to force XmMol to use the maximum number of colors contained within a colormap. By default, XmMol uses only free pixels of the standard colormap. This way, you will have in a typical case 256 colors. But you will also have color blinking when moving the pointer trough windows that do not belong to XmMol. This option can be usefull if you are running XmMol on a X terminal, and simultaneously running many others applications.

REMARK 1: the command line is passed to the XtInitialize call of XmMol. If you are an expert of X11, this can be a way to setup the font used, the window border colors and so on.

REMARK 2: XmMol might hang or core dump when invoked on some displays if the X11 resources have not been installed correctly. Please refer to the installation procedure. For some PC X servers, the only visuals supported are DirectColor or TrueColor, which may trigger XmMol to black and white.

2.2 XmMol user interface: The different windows

Three basic windows are used:

- The main window : it supports the graphic display and the menu bar. When the mouse cursor is within the graphic window, the middle button can control interactive 3D translation, rotation and scaling. The mouse left button can control atom picking to display labels or to get some informations.

- The 3D control panel window : it allows the control for interactive 3D transformations, such as translations, rotations, scaling, depth cueing and clip plane positioning. When the mouse cursor is within theses windows, the mouse left button controls interactive 3D transformations corresponding to the different items.

- The File summary window : for each file input, it displays the file name, its type, and allows some file level assignments. The different field are:

File : The filename (last part of the disk name and not the file header).

Id : A number that is the internal id of the file. Might be used to send Shell commands (see XmMol.doc1, or the Help, Topics, Menu doc.

Type : The type of the file. Might be a combination of Protein, DNA, RNA, Unknwn. This push button allows you to redefine interactively the type of the fragments more than one residue long of the file.

Frg RndrAs : This push button allows you to redefine interactively the rendering mode of each fragment more than one residue long. The rendering mode associated with each fragment might be either wireframe, trace or ribbon.

Visible : To set the whole file visible or not. Note that when not visible, this file does not support most of the operations, such as atom display selection, labelling or so.

Color : A push button that allows interactive edition of the file color fields.

Delete : a shortcut to close the file.

- Other temporary windows appear for specialized operations, notably for atom display and color selection, file I/O, color editing, docking.

When windows are overlapped by windows external to XmMol, you can raise all the windows opened by XmMol with the **File:Raise All** menu or the shortcut <Ctrl r> once the mouse pointer is within the main window.

2.3 Mouse bindings

First note that the buttons can be re-assigned using the configuration file. If you prefer for example to click the left button instead of the middle button to perform geometric transformations, refer to the section corresponding to configuration file. Here, the buttons functions are described as they are affected in XmMol default configuration. Button 1 is the left button, button 2 is the middle button and button 3 the right button.

button1 : this button has several functionalities depending on its location.

- selection operations involving the user interface : menu item selection, push buttons in the different windows, etc...
- atom picking within the graphic area : By default, picking one atom of a file will toggle the display of its name. But atom picking can be also be used for distance or angle measurement. Refer to the measure section.
- 3D transformation control when used within the 3D interface window : When the mouse cursor is within one of the 3D transformation item, clicking left will start the corresponding 3D transformation. Moving the mouse while keeping the button pushed will modify the speed and direction of the transformation (according to the left / right position relative to the centre of the window).

button 2 : its main use is for controlling 3D transformations within the graphic area. Its function depends on key combinations in which it is used.

button 2 : controls mouse driven rotation. Once pushed, the transformation starts. You can then move the mouse to change the magnitude and direction of the transformation. Button release stops it. Vertical displacement controls rotation around the X eye reference axis. Horizontal displacement controls rotation around the Y eye reference axis.

Ctrl + Button 2 : Since version 3.0, holding Ctrl + Button 2 will allow mouse driven translations.

Shift + Ctrl + Button2 : Since version 3.0, holding Shift + Ctrl + Button2 controls scaling. Up/Down moves will affect the scaling.

Caps-Lock + button 2 : When Docking, the caps-lock key toggles the button2 control either to the view (default), or to the displacement of the docked objects. (WARNING: do not forget the caps lock since it does interfere with the keyboard shortcuts!). Caps-lock + button 2 will trigger a control of 3D rotation restrained to the docked objects.

Caps-Lock + Ctrl + button 2 : will trigger a control of 3D translation restrained to the docked objects.

button 3 : This button is only used within Forme to set a target position and start deformation of the backbone (if you have this part of the package).

2.4 Keyboard shortcuts

A lot of keyboard shortcut are installed to access the different menu items. They consist generally in combining two keys: **Alt + letter** or **Ctrl + letter** Have a look on the right side of the menus to know the different shortcuts. PAY ATTENTION TO THE FACT THAT THESE CORRESPOND TO LOWER CASE LETTERS. SINCE UNIX MAKES A DIFFERENCE BETWEEN UPPER AND LOWER CASE, THE SHORTCUTS ARE INEFFECTIVE WHEN THE CAPS-LOCK KEY IS TURNED ON (SUCH AS CAN BE THE CASE WHEN DOCKING) !

The shortcuts are:

- Global level utilities
  - <Ctrl r> : raise all windows
  - <Alt d> : run a delegate
  - <Alt q> : quit XmMol
  - <Alt f> : start Forme
  - <Ctrl p> : Postscript snapshot
- File input/output
  - <Alt x> : input XmMol format file
  - <Alt p> : input PDB format file
  - <Alt s> : save to file
  - <Ctrl r> : raise all windows
- Rendering types

- <Alt r> : toggle toward ribbons
- <Alt t> : toggle toward trace
- <Alt r> : toggle toward wire-frame
- <Alt z> : toggle stereo

• Display patterns

- <Alt b> : toggle view only backbone
- <Alt h> : toggle view hydrogens
- <Alt a> : set all the fragments active
- <Alt m> : set all the atoms visible
- <Alt k> : keyboard residue selection
- <Alt c> : make the fragments of the chain containing the next atom picked non active
- <Alt g> : make the fragment containing the next atom picked non active
- <Ctrl s> : call atom display selector

• Colouring schemes

- <Ctrl h> : toggle toward atom type colouring
- <Ctrl b> : toggle toward backbone versus side-chains colouring
- <Ctrl t> : toggle toward residue type colouring
- <Ctrl c> : toggle toward chain colouring
- <Ctrl g> : toggle toward fragment colouring
- <Ctrl f> : toggle toward file colouring
- <Ctrl u> : call atom color selector

• Labels and measures

- <Ctrl l> : toggle toward one label per residue
- <Alt l> : toggle for long labels
- <Ctrl j> : flush labels
- <Ctrl a> : toggle toward picking for atom label
- <Ctrl e> : toggle toward picking for residue label
- <Ctrl d> : toggle toward picking for inter atom distance
- <Ctrl v> : toggle toward picking for valence angle measure
- <Ctrl x> : toggle toward picking for dihedral angle measure
- <Ctrl z> : toggle rule visible

2.5 Basic concepts for file management

The way the files are managed within XmMol has some consequences on the way some rendering patterns will be obtained. Mainly, remember that the file type is not innocent concerning the rendering you will obtain.

2.5.1 What happens when a PDB file is loaded

1 Atom identification (Based on the atom name) and atom type assignment. This can trigger some mistakes. Some of them are trapped, such as CA (Calcium) versus CA (alpha carbon). Other problems may occur for exotic groups.

2 Bond detection. It is based upon minimal and maximal inter-atom distances, for each atomic type. The values of these matrices can be modified in the configuration file. Note that you can prevent this computation by toggling the **MolEdit:Connectivities:ScratchBonder** menu item off.

3 Fragments and chains are detected.

4 Molecular types are assigned for each fragment. In the current version, it can be either protein, ADN, ARN or unknown. The type detection is based upon the residue names. XmMol will try the names of different residues. Note that since version 3.1, files containing mixed molecules (for example DNA and protein) are correctly managed. Fragment types have implications concerning the spline computation, the detection of hetero group, the detection of atoms that are considered as backbone or side chain, and finally the assignment of secondary structure (for protein).

5 Splines are computed. Note that you can prevent this computation toggling the **Prmtrs:FileDisplay** menu item off.

7 Secondary structure is assigned for proteins.

8 The display of atoms is set according to the current quick selection mask.

9 The color of atoms is set according to the current colouring mask. The file colors are set.

10 The view parameters are reset to ensure all the files are visible. This should soon be under the control of a **Prmtrs:FileDisplay:ViewAutoSet** menu item.

11 The rendering is performed.

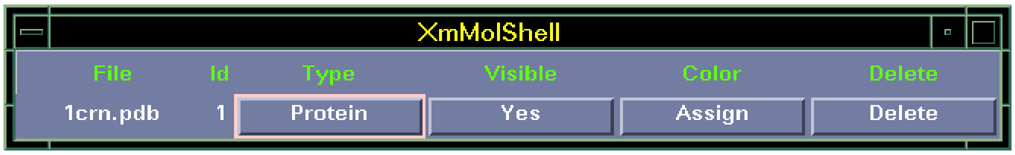
2.5.2 Chains versus fragments

XmMol uses two concepts for ranges of residues that belong to one file.

Chains correspond to the residues that are explicitly labelled as different chains within the input file.

Fragments (**Segments** in the old notation that might remain used in some text) correspond to the series of residues that are found to be bonded to each other. Thus, a chain that has missing residues will be divided in several fragments. Also, in general, Hetero groups that are not bonded to residues form fragments.

2.5.3 The file/fragment types and their implications



Note again that the file and fragment types assigned by XmMol have several implications. They condition the hetero group detection, the backbone/side chain atom assignment, the putative secondary structure assignment and the spline computation (different parameters for different atoms). Thus, if XmMol failed, you are left the opportunity to select the file type by hand by pushing the type button corresponding to the file.

2.5.4 The file colors

Once loaded, each file is assigned a series of colors that will be used for different color schemes such as *color by file*, *color by chains*, *color by fragments*. For each of these, XmMol manages indexes so that the different files cover the whole range of colors supported by the session of XmMol. However, this can lead to non satisfactory color assignments.

Pushing the Color assign button of the file summary window opens a file color selection box that allows the interactive re-definition of file, chains and fragments colors. To reassign one color, pick the item that you wan to modify, then pick one of the possible color.

Also note that every RGB color can be interactively modified (see the corresponding section), and that the colors used by XmMol can be defined within the configuration file.

2.5.5 The file activation mask: local versus global behaviour

Once loaded, each file is first displayed using the current rendering mode (selected as an item of the **Render:RenderType** menu. The atom colors are defined using the quick colouring pattern corresponding to the **Color** menu, and the atom display is set accordingly to the display pattern of the **AtmDspl:QckSelect** menu. Of course, you may want that each file do not have the same rendering, atom display and colouring patterns. To achieve this, XmMol applies, by default, the modifications of the patterns to only the file that are currently set as **Visible** in the file summary window. To get some files displayed as ribbons coloured by chains while others will be rendered as wire-frame coloured by atom and others rendered as trace coloured by residue type, you just have to mask the different files, and apply the desired rendering, atom display and color patterns to the visible files, and then to switch until you get the desired display. Refer to *A sample introductory session* for a practical example. Also note that this *local* way of applying the patterns can be triggered off by toggling the **Prmtrs:FileDisplay:DplMode** to global. In this case, all the files, visible or not will be affected in the same manner.

2.5.6 Active versus inactive fragments

Since version 3.1, the basic entities used for rendering are fragments. To get one fragment rendered as wireframe and another displayed as ribbon, XmMol uses a system of **activation** mask for each fragment, that is an extension of the file activation mask. Using the item **AtmDspl:QckSelect:HidePickFrg** or its shortcut ¡ALT g¿ will make the fragment containing the next atom picked inactive. (You can also obtain the same quick unactivation with the **FileBox Frg:RndrAs:NoDisplay** facility.) It is no longer displayed, and any change in the type of rendering or any quick display mask or colour mask will not affect it untill it has been reactivated by the **AtmDspl:QckSelect:Show All Frg** facility. Note that this is different from making all the atoms of the fragment hidden: simply, the fragment is no longer considered. As for file activation, this behaviour is under the control of the **Prmtrs:FileDisplay:DplMode** global/local item.

2.6 On line help

Some help can be accessed during a session in the **Help** menu. Three different levels will briefly explain the basics, the menus or the XmMol commands supported. Note that to access these, you need to setup the XmMOLDOCPATH environment variable to a directory where the files XmMol.doc0, XmMol.doc1, XmMol.doc2 and XmMol.doc3 are located.

3 Controlling the view

3.1 Basics concerning the 3D transformation

XmMol performs its viewing transformations in homogeneous coordinates. Thus, each step of the viewing transformation can be expressed as a 4x4 matrix, that is refreshed when necessary. The viewing transformation is decomposed as follows (from right to left):

$Crd_{screen} = PP * CG2E * Tr * St * Rt * Sc * E2CG * W2E * Dck * Crd_{world}$

where the original world coordinates Crd_{world} are transformed into the final screen coordinates Crd_{screen}, and :

Dck : is a putative matrix used for docking (this transformation is performed in the real space).

W2E : is a matrix tranforming the coordinates from the *world* (real space) to the *eye* reference (viewing space). It depends of the location of the eye relative to the scene that is looked at.

E2CG : is a translation matrix to move the origin to the current transformation center. It depends on the location of the transformation center (the point invariant in space).

Sc : is a scaling matrix. It is build from the concatenation of the consecutive elementary scaling transformations.

Rt : is a rotation matrix. It is build from the concatenation of the consecutive elementary rotations around X, Y and Z.

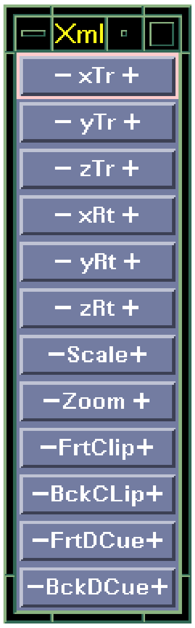
St : is a putative supplementary rotation matrix used for stereo.

Tr : is a translation matrix. It is build from the concatenation of the consecutive elementary translations around X, Y and Z.

CG2E : is the reverse of E2CG.

PP : is the plane projection matrix. It depends on the zoom parameters.

3.2 Basic 3D control



Most of the view parameters are accessible through the 3D panel opened besides XmMol main window, in addition to the fact that the mouse middle button can also control some transformations (see the Mouse button section).

Pick one item using the mouse left button, and hold it down. The magnitude and direction of the move are defined depending on the position of the mouse pointer relatively to the centre of the item (horizontally) (not limited to the item edges). To stop, just relax the mouse button. The magnitude of the elementary transformation used can be accessed by the **Prmtrs:TransfosSpeed** menu.

The reference used for the view is defined as follows: X horizontal, Y vertical and Z pointing toward the user. Apart from the translations (Tr) and rotations (Rt), the user can control the scaling, Zoom, clipping and Depth Cueing from this panel. Note the difference between scaling and zooming is as follows: scaling simply multiplies the coordinates in the eye reference by some factor. Zooming correspond to a displacement of the projection plane along Z. Thus, modifying the zoom affects the perspective transformation.

The clipping is here implemented in a simple way. Two buttons control the displacement of back and front planes along the Z axis.

The depth Cueing is defined as the variation of intensity of the colors between two planes perpendicular to the Z axis. Only the location of the planes can be controlled interactively in XmMol: the variation of intensity of the colors between the front and the back planes is defined when XmMol allocates its pixels, and can only be affected when the program starts by the command line option -dCueRange, that allows to define the difference of the red, green and blue values between the front and back planes (by default, it is set to 40000, white being defined as 64000 for both red, green and blue). Thus, narrowing the distance between the two planes (front and back) will increase the contrast between the front and the back of the molecules displayed.

The 3D transformations matrices can be reset by using the **restart View** menu item.

3.3 Centring the view

By default, the view is centred on the geometric centre of the files displayed. It is possible to assign it to any other atom in the menu View.

The **reCenter** item will compute the new centre depending on the atoms that are visible. The **centerOnPick** item will place the next atom picked (mouse left button) as the new centre. The **transfoOnPick** item will make the next atom picked (mouse left button) as the transformation center (might be placed anywhere within the view). Using the **reset View** leads to re-defining the centre as the geometric centre. Using the **restart View** will in addition also initialize the 3D transformation matrices.

3.4 Turning off Depth Cueing

This facility is left for machines that have low computational rates. Turning off the color depth cueing (using the **Render:DCue** menu, will save some time (but loose some quality).

3.5 Turning off image buffering

This facility is left for X displays that have low memory spaces. Turning off the use of a pixmap (using the **Render:PixmapUse** menu, will save some memory. This will lead to some blinking of the images.

3.6 Stereo views

Stereo (**Render:Stereo** or < Alt z >) will lead to the display of 2 views rotated by a few degrees on each half of the graphic window. The rotation value (degrees) as well as the distance separating the two views (pixels) can be modified in **Prmtrs:Stereo** menu.

4 Selecting the atoms displayed

The control of the display of atoms is possible through two mechanisms.

- The quick selection offers the possibility to get frequent patterns of atom display, such as backbone only, no hydrogens, ...
- The selection panel offers an exhaustive control of the display in various ways down to the display of one atom.

4.1 Quick selection

It is accessible through the different items of **AtmDspl:QckSelect** menu. This selection mechanism supports several features that distinguish it from the more general selection panel

- The atom display selection is by default only applied to the files currently **Visible**. This behaviour can be switched off via the **Prmtrs:FileDisplay:DplMode** menu item. Toggling it to DplMode:Glb (global) will trigger the selection to all the files loaded, without any consideration of its current visibility.
- The pattern currently active is automatically applied to the files loaded, (except the residue range mask). For example, having selected backbone only before loading a file will result in the display of the backbone of the file.
- There is no feedback between the items of the quick selection process and the corresponding items of the general selection panel. This can lead to unexpected results when you previously used the general selection panel.
- The keyboard residue range selector allows to select the residues that will be displayed. All others will be hidden. WARNING: the residue numbering is that of the input file (might not start at 1 from the first residue of the file).
- Some of the items have keyboard shortcuts.

4.2 Using the atom display selection panel

4.2.1 How does it work ?

When entering this mode, the display of the files is moved to local display mode. All the atoms are displayed (unless you ask not by toggling the **"Display shown only"** button. Two colors (or dashed lines for Black and White) are employed to differentiate atoms that will remain visible and those that will be hidden. The default is to color displayed atoms red (plain), hidden blue (dashed).

This panel can be used as follows:

1 Select something to apply a display mask on. This corresponds to both **"select global level"**, **"Select local level"** items.

2 Select more precisely what kind of atoms you want to consider by selecting one of the **"sub region ..."** items.

3 Apply a display mask by pushing one of the **"Perform"** items (**"Hide"**, **"Show"**, **"Hide H"** or , **"Show H"**).

4 Some options are available. In particular, you can modify the logical mask that combines **"Global"** and **"local"** levels to AND or to OR (see below).

5 To finish and come back to the normal display mode, push the **"Finish"** button on the top of the panel.

4.2.2 Combining the "Global" and "Local" selection levels

By default the **"Global"** and **"Local"** selection levels are merged in a unique level. Thus, selecting an item in either the global or local level will cancel any other mask of both global and local levels. You can however decide to use the local level as a sub-selection process of the global level. To do so, first toggle the corresponding **Options** button to **"Global AND Local"**. Now, you must select one item in each of the levels before applying a display mask. For example, if you want to hide only the side chain of one residue type in the selection you enter on the keyboard, first press **"KbdResRng"** of the global level, enter you keyboard selection, then press the **"Pick Res Type"** or press the button corresponding to the residue name that you want in the local level. Now press the **"SC"** button in the **"Sub region ..."**. Finally perform **"Hide"**.

4.2.3 Items description

The description is given from top to bottom of the panel.

Finish : To exit the selection process and get back to the previous display mode.

Select global level : A first level of selection

All : As it says.

AtmShwn : Only the atoms currently displayed are considered (by default, this is displayed in red (plain)).

AtmHdn : Only the atoms currently hidden are considered (by default, this is displayed in blue (dashed)).

KbdRsRng : A keyboard **residue** selection. Its syntax allows to select different non contiguous residue ranges in different files. The files are selected using their File Id's (displayed in the file Summary window) followed by a ":". Then ranges of residues can be specified in the form X-Y, where X is the residue number starting the region and Y the residue number ending the region. X and Y must be separated by "-". The numbers correspond to the file numbering, NOT to the rank order of the residues from the beginning of the file.

Alternatively, you can also specify a single residue by its number. You can enter as many regions or residues you want, separated by ",".

If the file has several chains, the numbers are completed by the chain Id. For example residue 23 of chain C must be named as C23. If you mention 23 and there are residues A23,B23, C23 and D23 in the file, the first occurence of a residue 23 in the file will be considered.

Finally, the wild card "$" can be used for files or for residues.

PckMol : Allows the selection of one file by picking one of its atoms.

PckChn : Allows the selection of one chain of one file by picking one of its atoms.

PckFrg : Allows the selection of one fragment of one file by picking one of its atoms.

PckRsRng : Allows the selection of a residue range of one file by picking two atoms belonging to the starting and ending residues.

PckRes : Allows the selection of one residue of one file by picking one of its atoms.

PckAtm : Allows the selection of one atom by picking it.

Select local level : A second level of selection

Sphere around pick : Allows the selection of all the atoms located at a distance less than a threshold to the next picked atom. The default is usually set to 8. Å. You can modify it by using the "Set sphere radius" option.

Pick Res Type : Allows the selection of all the residue of type corresponding that of the next picked atom.

explicit residue type list : XmMol maintains a list of all the residue types that have been detected. This provides an easy shortcut to the previous item.

Options : Some options to setup various behaviours

Global OR local : This toggle allows to consider the global and local levels as one single level (OR, the default) or as two embedded levels (AND, the alternate state). See upper.

Set sphere radius : To setup the radius of the sphere used when pushing the "sphere around pick" item . Note that the threshold must be set before performing the selection.

Display shown only : A way to trigger the display of only the atoms that will remain displayed on exit. This can of some help when trying to select among large files.

Sub region ... : The more thin level of selection

All : As it says

BB : The backbone

SC : The side chains (or bases for nucleic acids)

Het : The hetero groups

Atm Type : A keyboard input list of atom names separated by ",". No wild card for generic atom names, such as N* for example. $ means all atom names.

Perform : **Hide** : It applies the hide mask to the selection currently defined.

Show : It applies the show mask to the selection currently defined.

Hide H : It applies the hide mask to the hydrogens of the selection currently defined.

Show H : It applies the show mask to the hydrogens of the selection currently defined.

4.2.4 Note on mouse usage

During the selection process, the results of mouse picks can be used for different goals, such as defining residue ranges, or selecting a fragment, a chain, ... When using such facilities, the picks are parsed to obtain the desired information and do not result in atom or residue information. For example, if you want to hide series of residues, you can select the **PckRsRng** item, then pick two residues, apply **Hide**, then repick two other residues, re-use **Hide**, etc. The user may however want to identify atoms or residues during this process. To obtain this, simply select an item such as **All** that does not require any parsing of mouse pick results.

5 Selecting the atoms color

The control of the color of atoms is possible through two mechanisms very similar to those of the atom display selection.

- A series of pre-defined patterns offer the possibility to quickly get frequent colouring patterns, such as backbone versus side chains, colouring by files, ...
- The color selection panel offers an exhaustive control of the color of each atoms in various ways.

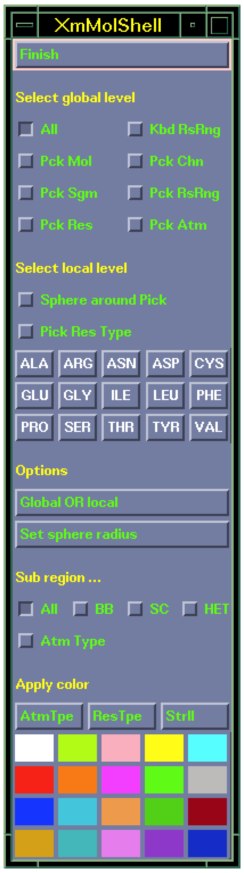
5.1 Quick colouring patterns

Some frequently invoked colouring patterns correspond to the different items of the **Color** menu. Note that:

- The patterns are by default only applied to the files currently **Visible**. This behaviour can be switched off via the **Prmtrs:FileDisplay:DplMode** menu item. Toggling it to DplMode:Glb (global) will trigger the pattern application to all the files loaded, without any consideration of its current visibility.
- The pattern currently selected is automatically applied to the files loaded. For example, having selected backbone/side chain colouring will result in the display of the input file using different colors for backbone and side chains. of the file.
- There is no feedback between the items of the quick selection process and the corresponding items of the general color selection panel.
- Most of the items have keyboard shortcuts.

5.2 Using the atom color selection panel

5.2.1 How does it work ?



This panel is accessible trough the **Color:Edit:Users atom colors** menu item. It allows the user to assign each atom the desired color. Its global organisation is very similar to that of the atom display selection panel. In fact, the two panel share most of their resources, which explains that you cannot access both of them simultaneously. This is a choice since the atom display selection panel uses a particular colouring.

This panel is organised to be used in the same way as the atom display selection panel (For more information about the global versus local selection, please refer to the corresponding section of the **Atom display selection**):

- 1 Select something to apply a color mask on. This corresponds to both **"select global level"**, **"Select local level"** items.
- 2 Select more precisely what kind of atoms you want to consider by selecting one of the **"sub region ..."** items.
- 3 Apply a color mask by pushing one of the **"Perform"** items (**"Hide"**, **"Show"**, **"Hide H"** or , **"Show H"**).
- 4 Some options are available. In particular, you can modify the logical mask that combines **"Global"** and **"local"** levels to AND or to OR (see below).

- 5 To finish and come back to the normal display mode, push the **"Finish"** button on the top of the panel.

5.2.2 Items description

For more information about most of the items, please refer to the corresponding section of the **Atom display selection**. We describe here only the specific items of "Apply color".

AtmTpe : Apply to each atom of the selection, a colouring patterns where each atom is coloured according to its generic class. The different classes are: H (hydrogens), C (carbons), N (nitrogen), O (oxygens), S (sulfur), P (phosphors), and others. The class of each atom is determined according to the first letter of its name. XmMol internal defaults are to color H in white, C in green, O in red, P in pink, S in yellow, N in blue and others in orange. Note that these colors can be modified either interactively, or by using a configuration file.

ResTpe : Apply to each residue type a different color. The colors depend on the order of apparition of the residue types and on the number of colors used by XmMol.

StrII : Each residue is coloured according to its secondary structure. The defaults are red for α helices, blue for β strands, green for coil, yellow for turns. Note that these colors can be modified either interactively, or by using a configuration file.

explicit colors : each colors currently loaded by XmMol can be applied to the current selection. Just pick the color to apply it. In back and white, different dashes are proposed.

5.3 Modifying the pre-defined colors

When using XmMol on a Pseudo-Color display (the most frequent case), you can access most of the definitions relative to the colors. Two main levels are to be distinguished:

- The RGB values defining each primary color such as red, green, ...
- The primary color used for different colouring pattern such as backbone/side chains, secondary structure, ... and the primary colors used for each file specific colors.

Primary colors correspond to the colors not shaded. XmMol will internally define colors derived from these for the depth cueing. The number of colors derived for each primary color is 8. This can be internally modified by XmMol if the space left in the colormap is low.

5.3.1 Naming your own default colors

On a color display, 30 colors are defined to be the default of XmMol:

White, GreenYellow, pink, Yellow, Cyan, Red, Orange, magenta, Green, Grey76, Blue, Turquoise, SandyBrown, YellowGreen, Brown,Gold, Aquamarine, Orchid, Violet, MediumBlue, BlueViolet, Coral, LightGrey, CornFlowerBlue, PaleGreen, GoldenRod, Orange, DimGrey, Old-Lace, SlateBlue.

Note that it does not mean that all these colors will be used by XmMol: XmMol will, depending on the space left in the colormap, use the maximum of them, starting from the first, i.e. white. Also note that black is not a color, even if used: you do not need to define it.

All these colors can be user defined within a configuration file. The names that you can select are those that are defined in the rgb.txt database of your X11 server.

Once modified, you can get back to XmMol initialisation default colors by using the **Color:Edit:DfltRG** menu item.

5.3.2 Interactively modifying the RGB values of the primary colors

The Red, Green and Blue values can be modified interactively by a panel accessible by using the **Color:Edit:EditRGBvalues** menu item. To modify one color, pick it and then move the sliders corresponding to the R, G and B values. This re-definition is interactive: the files displayed are affected by the modifications of the colors.

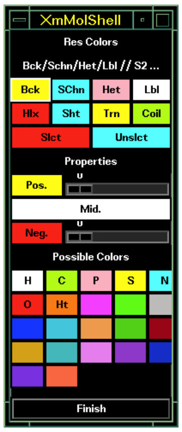
Once modified, you can reset the colors to their initial values by using the **Color:Edit:ResetRGBvalues**

5.3.3 Reassigning the colors used for the quick colouring patterns

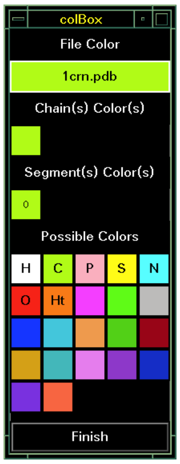
The colors that are employed when applying a backbone/side chain pattern (yellow and blue by default), or secondary structure pattern (helix red, sheet blue by default), as well residue type colors can be interactively reassigned by using the panel accessible through the **Color:Edit:EditStdColors** menu item.

Simply pick the color you want to reassign, and then choose amongst the Possible colors by picking it. Note that you can also modify interactively the RGB values of these colors while modifying the selection. Thus, you can modify these colors in any way at your convenience.

Once modified, the colors assigned at the initialization of XmMol can be restored with the **Color:Edit:ResetStdColors** menu item.



5.3.4 Reassigning the colors use for each file, chains and fragments



These colors can be accessed for each file by pushing the Color assign button of the file summary box. Pick the color that you want to reassign and then select one of the possible colors by picking it. For example, load the 1crn.pdb file, push the Color assign button of the file and select a coloration by chain (< Ctrl c >). Then pick the square corresponding to the chain color in the panel and click red for example. The 1crn file is now displayed in red when you select the coloration by chain.

Once modified, the colors assigned by default by XmMol can be restored with the **Color:Edit:ResetFileColors** menu item.

5.3.5 Black & White versus Color modes

Even when running XmMol on a color display, you are left the opportunity to use a black & white display. Note that in this mode, XmMol colouring schemes are superseded by dashing modes. Since XmMol can output close to WYSIWYG PostScript files, this can be used as a help for preparing black & white illustrations. The modes can be simply toggled by using the **Color:BW** and **Color:Color** menu items.

6 Displaying files in various ways

6.1 The different drawing modes

The current version of XmMol supports only line drawing (for historical reasons, this was chosen since it allows a reasonable interactivity on most X servers). However, it is possible to implement external renderers for supplying space filling rendering. Also, it is possible to access static space filling representations (cartoons) by using the interfaces with MolScript, Raster3D or QTree.

Three display modes are supported:

Wireframe : This is the default mode. Lines are drawn for each bond. In this mode, you can also trigger the double bonds to be represented by two lines with the **Render:RenderType:Lines** menu (Warning: this is a quick and dirty implmentation).

Trace : A simplified representation. Only the lines between the α carbons are drawn.

Ribbon : Smoothed representation of the main chain. Note that the number of ribbons as well as other parameters of spline computations are accessible through the configuration file.

The different modes can be accessed trough the Render:RenderType menu or via the keyboard shortcuts associated.

6.2 Combining Ribbons, wireframe and trace renderings

When a file is loaded, its rendering is set to the current rendering mode. You can change the rendering mode at any time. However, this will, by default, only affect the files that are currently set as "visible". Thus, the procedure to get different files displayed using different rendering modes is simply to make "non visible" the files that you do not want to affect by your next drawing mode selection and them to get them "visible" again.

Since version 3.1, it is possible to make fragments of a same file be rendered differently. The more general way to achieve this is to make "non visible" all the files or fragments that you do not want to assign a new rendering type, then to select the new rendering type and then make everything visible again. To achieve this, a simpler way is to use the Frg RndrAs facility of the file summary box. Pushing the button corresponding to one file will manage a panel that allow the direct selection of the type of rendering for fragments more than 1 residue. From this panel, you can also make each fragment non active by selecting the "noDisplay" item. Smaller fragments can only be assigned the default global rendering type (this to avoid huge panels describing each water molecule).

As well, you can load the file several times to get for example a ribbon representation of the backbone associated with a wireframe representation of the side chains.

Finally, it is possible to force all the files or fragments (even if not "visible") to adopt the same rendering procedure by toggling the **Prmtrs:FileDisplay:DplMode** item.

6.3 The drawing parameters

Since in the current version, we use X11 line drawing algorithm, these simply correspond to an access to some of the X11 line drawing parameters. They might be of weak interest, except if you wish to get larger lines for a picture for example. The parameters can be modified by using the **Render:LineStyle** menu items.

6.3.1 The line width

By default, X11 line drawing most efficient algorithm corresponds to a line width 0. This was chosen as the XmMol default. However, many servers can as well be efficient for larger line widths (for example SGI's). Note that XmMol does not manage the decrease of line width together with depth cueing. So, all the line are drawn with the same width.

6.3.2 The line style

An access to X11 line dashing instead of plain lines.

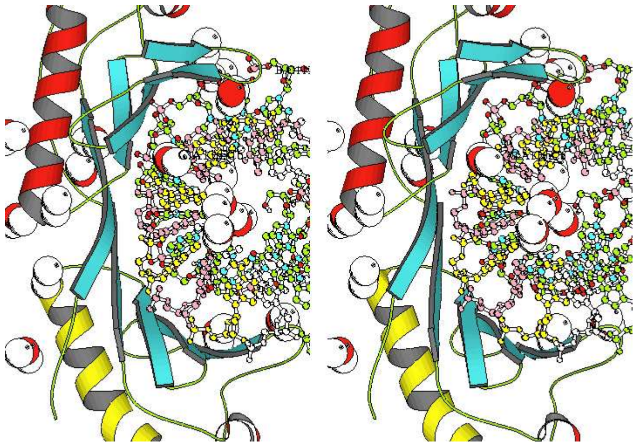
7 Cartoons

The **File:Snapshot** menu gives access to some facilities to generate figures in some different formats and representations. This is also a way to get some surface rendering views of the molecules. In all the cases, XmMol tries ot preserve "at best" the correspondance between its display and the images that will be generated. The different interfaces allow to generate representations such as wireframe representations, ribbons, ball and stick, worms, or CPK. Most of the programs will in turn generate images in formats that can be edited by programs such as ImageMagick's display or such as xv, to add any title or legend.

7.1 PostScript snapshots

Black and white or color postscript output of the views can be obtained in the **File:Snapshot:Postscript** menu. The file generated corresponds strictly to the current display, except for the labels for which the font type and size must be assigned as an option. Specifying a white background automoatically triggers the inversion of the white color.

7.2 Generating input for MolScript



XmMol can generate input files for MolScript (Kraulis, 1991). XmMol versions prior to 3.1 made it possible to prepare molscript input for either ribbon (the default), ball and stick or line drawing. Since version 3.1, it is possible to prepare input that merges different MolScript representations. The option that allowed in version previous to 3.1 to generate MolScript input files adapted to the PDB files non locally generated by XmMol has been suppressed. In the same time, the interface has evolved to get closer to a WYSIWYG standard. Most of this is achieved by saving coordinate files that contain only the atoms displayed by XmMol, in their viewing coordinates. Note however that a strict correspondence is not possible for several reasons: the viewing transformations of XmMol and MolScript are not strictly identical: In particular, keep in mind that XmMol uses perspective projection while MolScript uses parallel projection. Also, some differences observed on the edges of the views are partly due to the fact that the image output by MolScript and the XmMol graphic window may have a horizontal/vertical size ratio different. Finally, the level of the display details that can be accessed by XmMol or MolScript differ: XmMol' various colour and display masks to various atom subsets might have no exact correspondence in MolScript.

Concerning the correspondences used for the different drawing schemes (ribbon, wireframe), XmMol prepares MolScript input according to the following rules:

(i) All **protein** fragments displayed as ribbon in XmMol are displayed using MolScript helix, turn, coil and strand features. For each secondary structure element, the colors correspond only when all the residues of the block have the same color in XmMol. When not, XmMol sets for the element its default secondary structure color (as set when using the Color by secondary structure mask) as MolScript color.

(ii) Fragments represented in XmMol as their trace as well as nucleic acid fragments represented as ribbons are not considered and thus are will not be drawn by MolScript.

(iii) Fragments displayed as wireframe in XmMol can be displayed by MolScript using either ball-and-stick, CPK or lines . For wireframe and CPK, the colors defined in the molscript input file are, by default, set to the standard atom colors of XmMol (the default is green for C, yellow for S, red for O, etc...). However, if all the visible atoms of one fragment share the same color in XmMol, this default is superseded by that color. Finally, it is possible to force MolScript to make each atom's color as in XmMol by using the AtmColor switch in the options. However, this will make XmMol generate large MolScript input files that may require to enlarge the dimension parameters of MolScript (refer to the molscript.dim file, particularly MAXGST).

Simple line drawing was preserved but offers a poor interest compared to the other XmMol cartoon possibilities. The default is not to use color for them (using black and white according to the background), except if all the visible atoms of one fragment share the same color in XmMol.

Finally, a special mention was preserved for water atoms since it might be desirable to represent them using CPK even when ball and stick or lines are used for the rest of the display, and can be set using the option *draw CPK solvent*.

Two levels of selection of the rendering for each wireframe fragment exist:

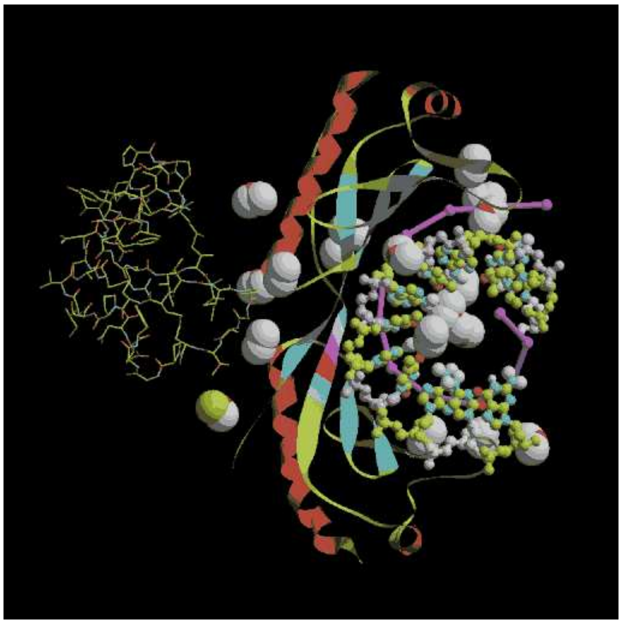
- A global level that can be set by using the **Ball and Stick**, **CPK** and **Lines** buttons. It allows to define the default rendering modes that will be employed for fragments displayed in wireframe. Selecting one item also triggers the initialisation of the corresponding mode for all the fragments currently displayed as wireframe.

- A level local to each fragment currently displayed as wireframe can be accessed by the **Wireframe fragment detail** button. It allows a different selection for all the fragments of size more than one residue (this to avoid huge fragment number when for example numerous water molecules are present). Note that since widgets are only set for some fragments, it is not allowed to modify the rendering type of one fragment when this **Wireframe fragment detail** is displayed.

Extra options concern (i) the label size and (ii) the possibility of using MolScript coupled with Raster3D (see below) to produce higher quality images. This will however not preserve the labels, or the stereo graphics. Note that for the labels, it is easy to re-introduce them using a tool like ImageMagick's display, or xv.

Depending on the program accessible by your PATH environment variable, XmMol can run molscript and display its output (display only flag), or simply write the molscript input, and the associated PDB files. The display requires an access to molscript and gs (ghostscript) by default. This can be redefined in the XmMol configuration file.

7.3 Generating input for Raster3D



XmMol can generate input for Raster3D (Bacon & Anderson, 1988 ; Merritt & Murphy, 1994). In the present version, only ball and stick or hard sphere input can be generated. The options correspond to the Raster3D parameters described in the Raster3D documentation.

Since version 3.1, the interface has much evolved to get closer to a WYSIWYG standard. Notabely, XmMol ribbons are now represented in Raster3D either using ribbon or worms representation. The quality of these representations depend on the number of ribbons and the number of control points used by XmMol. XmMol wireframe or trace can be represented as ball and stick or CPK using raster3D. The relative sizes of the spheres and cylinders can be set in the options as a multiplicative factor to the atom Van der Waals radii. It is possible to assign each fragment of length more than 1 residue a Raster3D rendering mode (other fragments are assigned the default rendering mode). The correspondence between the views generated by XmMol and Raster3D including geometry as well as the colors is preserved. However no stereo views can presently be generated and labels are not preserved. Most of the Raster3D rendering parameters can be accessed by XmMol using the option panel.

Two levels of selection of the rendering for each wireframe fragment exist, similar to those existing for MolScript:

- A global level corresponds to the rendering possibilities proposed in the **R3D default rendering** section. It allows to define the default rendering modes that will be employed for fragments displayed in wireframe, trace or as ribbons in XmMol. Selecting one item also triggers the initialisation of the corresponding rendering type for all the fragments currently displayed in the mode associated with the item.

- A level local to each fragment currently displayed as wireframe can be accessed by the **Fragment detail** button. It allows a different selection for all the fragments of size more than one residue. Note that since widgets are only set for some fragments, it is not allowed to modify the rendering type of one fragment when this **Wireframe fragment detail** is displayed.

Depending the program accessible by your PATH environment variable, XmMol can run the Raster3D render program and display its output (display only flag), or simply write the Raster3D input file. The display requires an access to render (Raster3D) and display (ImageMagick) by default. This can be redefined in the XmMol configuration file.

7.4 Generating input for QTree

XmMol can generate input for QTree (Martin, 1993). In the present version, ball and stick, worms or CPK input can be generated. The options correspond to the parameters described in the QTree documentation. However, one option was added to allow the image centring correspondance between XmMol and QTree. A description of the differences in the qtree file command.c to install this option is given in the QTreeDiffs.c included in the distribution.

Note that the present version is not fully interfaced with QTree, and that you may encounter problems with the scaling of the images generated compared to their actual XmMol scaling.

Depending the program accessible by your PATH environment variable, XmMol can run the QTree programs and display its output (display only flag), or simply write the input file.

8 Loading and saving files in different formats

8.1 Supported file formats

XmMol can presently read its own internal format and the PDB format. For other formats, you can use converters toward PDB or XmMol's format. University of Arizona BABEL's supports the conversion of most formats toward PDB (Walters & Stahl, 1992). The XmMolBabel delegate program is a simple interface with this program.

8.2 Note about the PDB files

XmMol checks the type of the file according to its first line. For PDB files, there are many possible keywords. Most of them are taken into account. If however, your file begins with an exotic keyword, XmMol will answer that the file was not recognised as a PDB file. To solve this problem, simply add a dummy HEADER first line for example.

The CONECT fields are systematically be taken into account, and no check on their validity is performed: the bond detection only considers pairs of atoms that were not linked together by such CONECT fields.

Finally, the fields of the ATOM lines located after the Z coordinate are ignored (version 3.1).

8.3 XmMol internal file formats

XmMol's internal format is close to the *mac* format used by programs such as Jumna, PCurves, Polypep-Shuffle-Smd, that was developped by the group of R. Lavery at Institut de Biologie Physico-Chimique, France.

The files have the following format:

1 header line: This line may be anything (less than 80 characters) but MUST begin with a "#" character (for compatibility reasons with old format).

1 summary line: 2 integer that are the number of atoms and the number of residues. (mac format is striclty 2I5)

coordinate lines: 1 line per atom. The format is, using FORTRAN conventions:

A4,3(10.5F),2(I3),8F,A1,A3,C,I4,I4,I5

that correspond to:

Name of atom (4 char), X, Y, Z (10.5), atomNumber (I3), atom class (I3), atom charge (8.4), a blank, residue name (3 char), chain identifier (1 char), number of residue (I4), is whitin a ring (I4)(not used whithin XmMol), number of atom (I5)

for example:

```
HA    21.09977 -13.49282  -9.22249  1 12   .0711 GLU    2   0   19
```

connectivity lines: For each atom, a list of the covalently bonded atoms is given. Each atom has at maximum 6 ligands. The format is (FORTRAN convention): I5), 6I5, A, I5),6I5 the number of the atom followed by ')', the number of its ligands, 0 means empty entry. Two such entries per line, separated by a blank. The number of the atom always start at 1 and make a series up to the number of atom of the file, even if the atom numbers of each coordinate line are not these (in this current version). Negative values indicate double bonding.

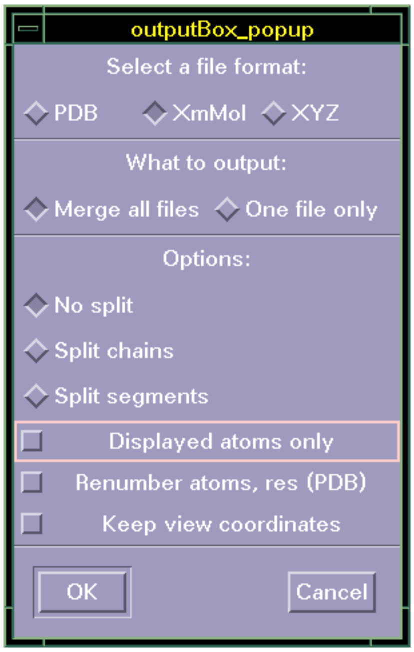
example:

```
1)  2  -8  9  0  0  0    2)  1  3  5  6  0  0
```

the first atom of the file has 3 neighbours, the second, 8th (double bond) and ninth atoms of the file.

WARNING: for files including more than 9999 atoms, the I5 format of the connectivity matrix is automatically superseeded with a I6 format.

8.4 File output



Saving the coordinates you have visualized by XmMol can be made in different ways. Using the **File:SaveAs** menu will lead to a panel such as below.

File output can be made as XmMol, PDB or XYZ formats. The XYZ format might not correspond to other XYZ formats: it just consists in the series of the atom coordinates.

Considering you might have several files loaded, an opportunity is left to concatenate them all in one file (Merge all files), but you might also want just to save the coordinate of one file. Since files might consist in several chains or fragments, it is possible to choose whether all the chains or fragments will be saved within a same file or to split them in one file per chain or fragment. In such a case, the names will be derived from the name you will enter, for example toto, will result in the creation of the files totofrg1.pdb to totofrg3.pdb if there are 3 fragments. Chain identifier will be use instead if you split the file by chains. Another way of saving only part of the atoms is to setup the **Displayed atoms only** flag.

The **Renumber atoms, res (PDB)** flag will trigger the re-numbering of all the atoms and residues from 1, but makes only sense for the PDB format. This is automatically performed for XmMol format.

Finally, it is also possible to keep the coordinates of the view.

9 Identifying atoms and residues

Items concerning the identification of residues and atoms are located in the **Labels** menu. Two different type of questions are adressed: (i) systematic labelling to locate a residue or atom. This is achieved by some of the **Labels** menu items. (ii) local labelling to get information on a residue, atom that you have located. This is achieved by picking the atoms or residues.

The systematic display of one label per residue for all the residues of the files can be switched by the **Labels:ResLabels** flag.

The display of the name of the residue or atom of one atom can be set by picking one atom once the **Label:Pick** flag is set either to Atm or Res Labels. Re-picking one atom for which a label is already displayed will remove it.

9.1 Short and long label formats

Two formats of the labels can be toggled for the residue and atom labels. The long format has the format: C.RN:A, where C is the chain id (if any), R is the residue name(3 letters), N the residue numberand A the atom name.

The short format includes only the residue number.

10 Displaying measures

Some measures can be displayed on the screen by picking atoms (mouse left buttons). The **Label:Pick** flag controls the type of the measure performed. Once displayed, the measures can be either hidden or cleared all together (**Label:Measures:clear** or **Label:Measures:Shown/Hidden**. The display of a single measure can be cleared by re-picking the atoms defining it. This supposes that the **Label:Pick** is set to the type of the measure to be cleared.

10.1 Measuring inter-atomic distances

Inter-atomic distances can be measured by picking consecutively two atoms (mouse left button), once the **Label:Pick** flag has been set to distances. Two types of distances can be set: a real distance (the default) or a distance in projection on the screen plane. The toggle of the type corresponds to the **Label:Dist** flag.

Re-picking the two atoms defining the measure will clear the display of the measure.

10.2 Measuring valence angles

Valence angles are defined by 3 atoms consecutively picked. The values are displayed in degrees. Re-picking the 3 atoms will clear the display of the value.

10.3 Measuring dihedral angles

Dihedral angles are defined by 4 atoms consecutively picked, similarly to what is described for valence angles.

10.4 More angles

As a default, XmMol will check that the atoms defining angles are covalently bonded together. This control can be removed by setting the **Label:Angle** flag to non-bonded, thus giving access to the measure of more general angles defined by three or four atoms.

10.5 Displaying a rule

The **Label:Rule** menu toggles the display of a rule. It is graduated in angstrom. This rule is compatible with the stereo mode.

10.6 Displaying hydrogen bonds

Hydrogen bonds constitute a special topic in XmMol: it will only detect them within files that specify the coordinates of the hydrogens, but it has no built in method to compute the location of the hydrogens. Thus, the **Label:HBonds** menu will be inactive on most of the PDB files. However, it is simple to interface XmMol with any method that detects hydrogen bonds. An example of a delegate task that will compute and make XmMol display the hydrogen bonds of a molecule is now distributed in XmMolHbPlus.c. This process however require that you have installed the hbplus (McDonald & Thornton, 1994) program.

11 Some modelling tools

11.1 Docking

XmMol offers some manual docking possibilities that are accessible trough the **MolEdit:Dock** menu.

To understand how docking can be used, remember that XmMol just modifies the relative position of the molecules by inserting a transformation matrix somewhere in the cascade of the view transformation. Thus, the coordinates are NOT modified, even if the view is, untill you request it.

Once the docking panel is displayed, you first have to define what type of structure you want to select to be displayed relatively to the others. Thus, either pick the "PckFile", *"Pck Chain" or Pck Frg" button. Then pick one atom of the corresponding structure. You can perform this cycle several times to add several files or fragments to the moving set. You can also remove elements of the moving set by switching the "Add" flag to "Remove".

Once your set is defined, you can proceed to the relative displacements. Two possibilities of controlling the moves exist: using the "Dock xTr" to "Dock zRt" buttons in cunjunction with the mouse left button, or using the mouse middle button in conjunction with the "Caps Lock" key.

The "Dock xTr" to "Dock zRt" buttons can be used in the same way that the 3D panel buttons: Pick one of them using the mouse left button, and hold it down. The magnitude and direction of the move are defined depending on the position of the mouse pointer relatively to the center of the button (horizontally). To stop, just relax the mouse button.

Using the mouse middle button is by default used to control the translation or rotation applied to all the files. It is possible to use it for the docking by toggling the "Caps lock" key. Once it is set, mouse middle button alone will modify the rotation of the docking set, mouse middle button holding the "Ctrl" key will affect the translation.

To make the coordinate transformation permanent, it is necessary to push the "Save state" button. Once this is done, there is no way to undo the transformation else than re-doing the inverse docking manually.

The reset button flushes the docking transformation matrix and initialize it to the identity matrix.

The restart button performs as the reset button, but also flushes the information concerning the moving set.

11.2 Fitting

XmMol has no built in superimposition tool. However, a simple exemple of such a tool is provided as a delegate (XmMolSuperImp or XmMolShmSuperImp) using the SuperPos packages (Sippl & Stegbuchner, 1991). These delegates provide a basic way to superimpose fragments, chains or molecules. A more sophisticated delegate can easily be derived from them.

11.3 Forme: interactive backbone deformation

11.3.1 Forme basic concepts

Forme (P. Tuffery, J. Mol. Graph., 1991, 9, 175) is a tool that allows to interactively model the geometry of the backbone of a protein, single strand RNA or single strand DNA. Geometrical transformations are made by series of rotations around degrees of freedom (DF) that are the dihedral of the backbone of the molecule selected by the user. Forme will manage so that the successive rotations allow one special "Anchor" point (an atom of the backbone), on which a "force" is applied, to reach a target position (user defined via the mouse). Currently, the anchor point must be outside the region that encountains the DFs, and only stretching operators have been implemented. Thus, 3 regions of the backbone can be distinguished:

- a first region R1 that is BEFORE the first DF. The coordinates of this region will not be affected by the transformations.
- a region R2 that contains the DFs around which the rotations will be selected. Thus, this region will have its geometry changed. Note that the region can be fragmented as several sets of DFs, for example if you want to apply a move on two loops but preserve the geometry of a helix between the two loops.
- a region R3 that is after the last DF and starts at the "anchor". This region will keep its geometry unchanged but will be moved due to the changes of the geometry of R2. Constraints can be applied so that R3 moves but keep its overall orientation relative to R1 unchanged (invariant stretch) or not (simple stretch).

Then, a "force" can be applied on the anchor, and the rotations are organised so that the move of the anchor answers the "force". On this implementation of Forme, the "force" is defined by a vector going from the anchor to the mouse pointer, in a plane parallel to the screen plane and containing the anchor.

11.3.2 Applying a transformation

To enter Forme, use the **MolEdit:Forme** menu. The menu bar is now changed and the specific menus appear. Most of menu topics are identical to that of the main menu bar. A small difference is located within the color menu, where "Forme color" is defined. The **WrkRegion** menu contains the items needed to define the anchor and the set of degrees of freedom. The **Defo** menu allows to select the type of deformation to apply to the region defined. Note that undoing is simply considered as a deformation operator, and is therefore to be used in conjunction with the mouse right button.

To apply a transformation, you must:

(i) define a deformation region, (see the wrk Region menu)

(ii) select a point for applying the "force" (see the wrk Region menu)

(iii) select a deformation operator, (see the Defo Menu)

(iv) apply a "force".

- To define a region, use the AddZone item. The click (left button) on the structure the atoms of the backbone that surround the region that supports the deformation. To fragment the zone according to your convenience, you can remove degrees of freedom using the DelZone item and performing the same selection procedure. The allowed degrees of freedom are coloured differently (A control to ensure your picks were successful). The Reset Zone allows to clear all the DFs and start over the definition of R2.

- To select the point the "force" will be applied on, use the SetAnchor item of the Wrk Region menu, then pick the desired atom on the structure. It must be marked with a box (A control to ensure your pick was successful).

- To select a deformation operator, choose one in the DefoControl menu. The currently selected transformation operator is labelled on the menu bar. The default is simple stretch. Simple stretch (The default): will just ensure that the anchor point will fit the target location. Invariant Stretch: will ensure both that anchor point will fit the target location and that R3 will not rotate relatively to R1 (i.e. is only displaced by translation (not by any rotation)). Undo: allows to undo 500 last moves.

- To proceed, you now just have to click on the mouse RIGHT button. The current location of the mouse defines a target point that is in the same plane (parallel to the screen) that the anchor. You can interactively move it during the transformation (that will end when you release the mouse right button). The amplitude of the elementary rotations are correlated to the distance between the anchor and the target points. So, do not hesitate to move the mouse far from the anchor point to obtain large moves. The move will end at the super-imposition of the anchor and the target point (mouse pointer), or when you stop holding the mouse right button.

  Forme only works on copies of the coordinates. When you agree a transformation, you can make the coordinates saved using the "Save State" item of the file menu. the "Reset state" item brings the coordinates directly back to the last "SaveState" (starting coordinates by default).

  On entry of the Forme menu, a new color mode "Forme color" is accessible, and displays only the backbones and the current DFs. After having performed your selections of DFs and anchor, you can toggle this mode to any other colouring mode.

- The Quit forme item of the file menu will lead back to the standard XmMol menu bar. Again, you will be asked if you want to keep your changes or not.

WARNING: Remember that on exit, all your deformation is lost except if you make it explicitly permanent by using the Save state item or answering Yes. Using the "Reset state" item brings the structure back to the previously saved state (unlike the undo operator). When entering Forme, the current coordinates are automatically pushed as the current save state.

DOUBLE WARNING: Do not forget that (i) Forme is only a geometrical tool (will not cry if crossing bonds or overlapping atoms) (ii) the side chains just follow the move (no corrections within Forme) and (iii) as we use a finite number of degrees of freedom, all the target locations may not be reached by all the possible sets of degrees of freedom (as a principle, the smallest the number of degrees of freedom, the most limited the reachable target locations, and for moves with invariance of the tail region, the less accurate the control on the move).

TRIPLE WARNING: Obviously, some experience is required to appreciate which deformation region to select to obtain the move wanted. Further versions of Forme may be enhanced relatively to that. Moving the mouse pointer during deformation is often a good idea to get out local conformational traps.

12 Dialog with external applications

12.1 Using macros in shell commands for I/O operations

The **File:ShellCommand** allows XmMol to execute shell commands. It is coupled to a simple macro parser that allows some dialog with external programs, via possibilities of coordinate input/output. This will hang XmMol untill the command is ended, unless you send it in the background using the shell & facility.

The command can be any single line that would be accepted by a shell. Some special motifs are filtered to allow easier file handling. They are on the form:
<$,#><i,o><P,M,X>Id(filename)
where

- **$** Using $, the I/O operation is performed and the filename string is passed to the shell command line.
- **#** Using #, the 'filename' part of the command is not passed to the shell, i.e. the output or input is performed, but this part of the command does not appear in the shell command. This is a shortcut for creating or modifying disk files easily.
- **i** The file is input by XmMol.
- **o** The file is output by XmMol.
- **P** The file is in the PDB format.
- **M** The file is in the XmMol mac format.
- **X** The file is in the XmMol XYZ format (only the coordinates are output).
- **Id** corresponds to the current Id of the file that is displayed in the file Box. A 'n' value implies, for input that a new Id will be created.
- **filename** is the name of the file implied. This is the name that will effectively be written or read, and then used by the shell command line if you are using the $ character.

Remember that first output (if any) is performed, second the shell command is performed, third the input (if any) is performed.

For example:

$oP1(toto.brk) : will write the file loaded that has Id 1 as toto.brk, in the PDB format, and then execute the shell command "toto.brk", which means nothing since toto.brk is not a shell script or an executable file.

#iPn(toto.brk) : will read the file toto.brk, in the PDB format. The command "toto.brk" is NOT passed to the shell. This is the correct way to input toto.brk. $iPn(toto.brk) would mean nothing since toto.brk is expected to be a PDB file and cannot correspond to a shell command.

#oP1(toto.brk) : will write the file currently loaded with id 1 as toto.brk, in the PDB format.

#iP1(toto1.brk) : will read the file toto1.brk, in the PDB format and replace the coordinates of the file toto.brk, currently loaded with the Id 1. This will not work if the toto.brk and toto1.brk do not correspond to the same molecule.

#iP1(toto1.brk) #iP1(toto2.brk) : will read the file toto2.brk, in the PDB format and replace the coordinates of the file toto.brk. The #iP1(toto1.brk) is superseded with #iP1(toto2.brk).

cat $oP1(toto.brk) : will first write toto.brk, then cat it.

#oP1(toto.brk) cat toto.brk : the same in a different style. Only "cat toto.brk" is sent to the shell, once toto.brk has been written.

cat $oP1(toto.brk) | grep CA > $iPn(tutu.brk) : will write toto.brk, cat it and send it to the grep command. The result will be written as tutu.brk and loaded by XmMol as a file in the PDB format.

babel -imol mycrds.mol -op $iPn(mycrds.pdb) : will make babel convert the file mycrds.mol that is in a SYBIL file format to a PDB file mycrds.pdb that will be input.

#oP1(toto.brk) babel -ip toto.brk -omol toto.mol : will write the coordinate set of the file input with Id 1 into the file toto.brk and then make babel convert it to a SYBIL format.

babel -ip $oP1(toto.brk) -omol toto.mol : Another way of doing the same.

12.2 Interfacing external programs as delegate tasks

12.2.1 The XmMol command language

The commands supported by XmMol give access to most of the data fields of the files input, to the parameters controlling the view, the display, the colors, the display of the atoms. The command also provide some way of using the X11/Motif interface to open box and query parameter values. Some examples of tasks that have been interfaced with XmMol are provided within the distribution. Concerning other possibilities, note that it is really simple to make

XmMol visualise the steps of a minimisation or of a molecular dynamics. It is also possible to interface external graphic programs that perform surface rendering.

For a detailed description of the commands, please refer to the XmMol.doc2 file (you can call it while running XmMol in the Help menu), or to the XmMol delegate task reference manual.

12.2.2 Sending commands directly to XmMol

The commands were mainly defined to be used by external delegates. Thus, not all the commands can be run from command files (for example all the commands that request data transmission with the eternal delegate process).

Keyboard input of a command However, some commands can be send directly to XmMol by typing them in the box opened by the **File:XmMolCommand** menu. For example you can load the 1crn.pdb file by entering the command:

XmIOLoadFile 1crn.pdb DcPDB

A main interest of this feature remains the possibility to interactively customize the menu bar (refer to the corresponding section).

Loading a command file Command files consist of series of orders that XmMol will interpret, just as if you were entering series of commands by using the **File:XmMolCommand** menu.

The purpose of using command files can be considered as twice:

- It is a way for recording a series of order to get a complex work pre-loaded.
- It is a way of customizing XmMol. Commands give access to most of the parameters of the viewing. They can also be used for configuring XmMol menu bar (refer to the corresponding section).

A sample command file is XmMol.cmd

12.2.3 Programming delegate tasks

XmMol is able to dialog with external programs using a synchronous protocol based upon software interrupts. Such programs, that are written independently of XmMol are a way to provide XmMol with "delegate" tasks. The possibilities offered by this feature allow to customise XmMol to take advantage of most of the molecular modelling or visualisation methods that are not built in it. Using such a protocol, you can very easily interface any modelling, molecular dynamics or even graphic method that you would like to use. More generally, this gives access to the user favourite methods without requiring to rewrite and include them within XmMol's core.

The way to proceed is as follows:

- interface the program so that it can dialog with XmMol. This consists simply in linking the XmMolDelegateLib.o object file that describes the communication protocol, and declaring within the program that it can dialog with XmMol with one call at the beginning of the program: XmMolConnect(), and one et the end: XmMolDisconnect(). Having achieved this, the program is able to perform requests to XmMol and to get, send or share data from, to or with it. A detailed description on how to write or interface external programs with XmMol can be found in XmMol.doc2, as well as a list of the commands available.

- to take advantage of the method implemented in your program as a feature of XmMol, start it with the **File:XmMolDelegate:New** menu (enter its name in the box). Since the dialog is based on interrupts, the **File:XmMolDelegate** menu contains different items that allow to suspend, re-activate or kill the delegates.

One advantage of the protocol is that the two processes only dialog when needed, i.e. XmMol does not scan periodically a file to check if some delegate needs some exchange of data or information. In the case of a minimiser for example, once started, the dialog protocol is designed so that you can still work in XmMol: load other files, modify the atom display or colors, etc. Also, for some fields of each file input in XmMol (such as the coordinates), sharing memory with the delegate tasks is possible: this makes the interface much faster.

Also, interfacing a program with XmMol does not mean that it is now dependent of XmMol: you can run it alone, all the calls to XmMol remain silent and do not affect the program if it is not run from XmMol. For example, you can use a minimiser interfaced with XmMol independently of it, and use it from XmMol when you want to visualise the different steps of a minimisation.

Finally, as described in the previous section, once a delegate task is written, you can make it a “part” of XmMol by defining a menu item that call it.

For a detailed description of how writing delegates, please refer to the XmMol Command Language For Delegate Tasks Reference Manual.

13 Customizing XmMol

13.1 X11 resources

Most of the X11 resources of XmMol can be assigned within a .Xdefaults file or using the xrdb -add .XmMolrc command. Most of the classes managed are described within the .XmMolrc file of the distribution. For each class, you can specify at least the background and foreground colors, the font used. Refer to your X11 or Motif user guide for a list of resources available.

A good idea to see all the menuBar items:

XmMol*FontList: -*-swiss 742-bold-r-normal-*-140-*-p-100-*

You might need even smaller fonts if your screen is less than 17 feet.

A sample configuration is given in the inline Help. It seems however better when the resources are configured for all the users by the system administrator.

13.2 Re-assigning some parameter default values in a configuration file

Some parameters can be defined on entry of XmMol by using a "configuration file". These parameters concern the color used for the drawing, some postscript parametrisation (B&W/Color, font name & size, ...), the spline parameters (number of ribbons, dot weights), the bonding matrix (min and max distances), the parameters used for the HBond detection, mouse buttons assignment, the postscript, AVS and MTV viewers accessible for displaying the cartoons.

All these parameters are described in the XmMol.cfg file that comes with the distribution.

Configuration file can be loaded at any time using the **File:Open:Cfg** menu.

When you start XmMol, it will automatically search for a $XmMOLDOCPATH/XmMol.cfg file, if you have not specified any other configuration file using the -icfg command line argument. Since a special item (DFLTCMDFILE) of the configuration files is to specify a default command file, this is also a way to access a default command file used for example to implement permanent items in the menu bar.

13.3 New menus/items in the menu bar

The XmAddUserMenu allows to add a new sub menu to every menu of the menu bar. For example:

> *XmAddUserMenu File MyIOs* would create and add an empty sub menu "MyIOs" to the menu File of the menuBar.

You can also add a new menu on the menu bar by not specifying the father menu Name. For example:

> *XmAddUserMenu MyTasks* would create and add an empty menu "MyTasks" on the menu bar.

The XmAddUserCmd command allows to add items to the main menu bar of XmMol. When selected, the items will execute either shell commands (DcCMD type) or start delegate tasks (DcDLG). For example

XmAddUserCmd DcDLG MolEdit Fit %XmMolShmSuperImp% adds to the menu “MolEdit” the item “fit” that will start the delegate XmMolShmSuperImp.

*XmAddUserCmd DcCMD File:Open ls %ls -al *.pdb%* adds to the sub menu “Open” of the “File” menu the item “ls” that will execute the shell command “ls -al *.pdb”.

The menus are specified by their names. Sub menus must be specified in the form: MenuName:SubMenuName:SubSubMenuName. The command itself MUST be enclosed within “%”. This symbol allows the command to be composed of several words.

Note that reciprocal functions exist to remove the menu items inserted (see the XmMol Delegate Tasks documentation).

14 References

- Bacon, D. J. & Anderson, W.F. (1988) "A fast Algorithm for Rendering Space-Filling Molecule Pictures". (abstract of the Seventh Annual Meeting of the Molecular Graphics Society), J. Molec. Graphics 6, 219-220.

- Kraulis, P. (1991) "MOLSCRIPT: a program to produce both detailed and schematic plots of proteins". J. Appl. Cryst 6, 219-220.
  mail to per.kraulis@sto.pharmacia.se

- Martin, A. C. R. (1993-1995) "QTree V2.1 Copyright (C)", SciTech Software, 23 Stag Leys, Ashtead, Surrey KT21 2TD, UK.
  http://www.biochem.ucl.ac.uk/ martin ftp://ftp.biochem.ucl.ac.uk/pub/martin (martin@biochem.uc

- McDonald, I.K. & Thornton J.M. (1994), "Satisfying Hydrogen Bonding Potential in Proteins", J. of Mol. Biol. 238, 777-793.
  http://www.biochem.ucl.ac.uk/bsm/biocomp/index.html

- Merritt, E. A. & Murphy, M. E. P. (1994) "Raster3D Version 2.0 - A Program for Photorealistic Molecular Graphics", Acta. Cryst. D50, 869-873.
  ftp://ftp.bmsc.washington.edu/pub/raster3d

- Sippl, M., Stegbuchner, H. (1991) "Superposition of three-dimensional objects: A fast and numerically stable algorithm for the calculation of the matrix of optimal rotation", Computers and Chemistry, 15, 73-78.
  ftp://gundi.came.sbg.ac.at/pub/SuperPos

- Tuffery, P., Vaney, M. C., Mornon, J. P. and Hazout, S. (1991) "Forme: An interactive package for protein backbone deformation", J. Molec. Graphics 9, 175-181.

- Tuffery, P. (1995) "XmMol: An X11 and Motif program for macromolecular visualization and modelling.", J. Molec. Graphics 13, 67-72.
  ftp://duc.urbb.jussieu.fr

- Walters, P. & Stahl, M. (1992-1994), "Babel version 1.1 Copyright (C)", Dolata Research Group, Department of Chemistry, University of Arizona, Tucson, AZ 85721, USA.
  ftp://joplin.biosci.arizona.edu/pub/Babel