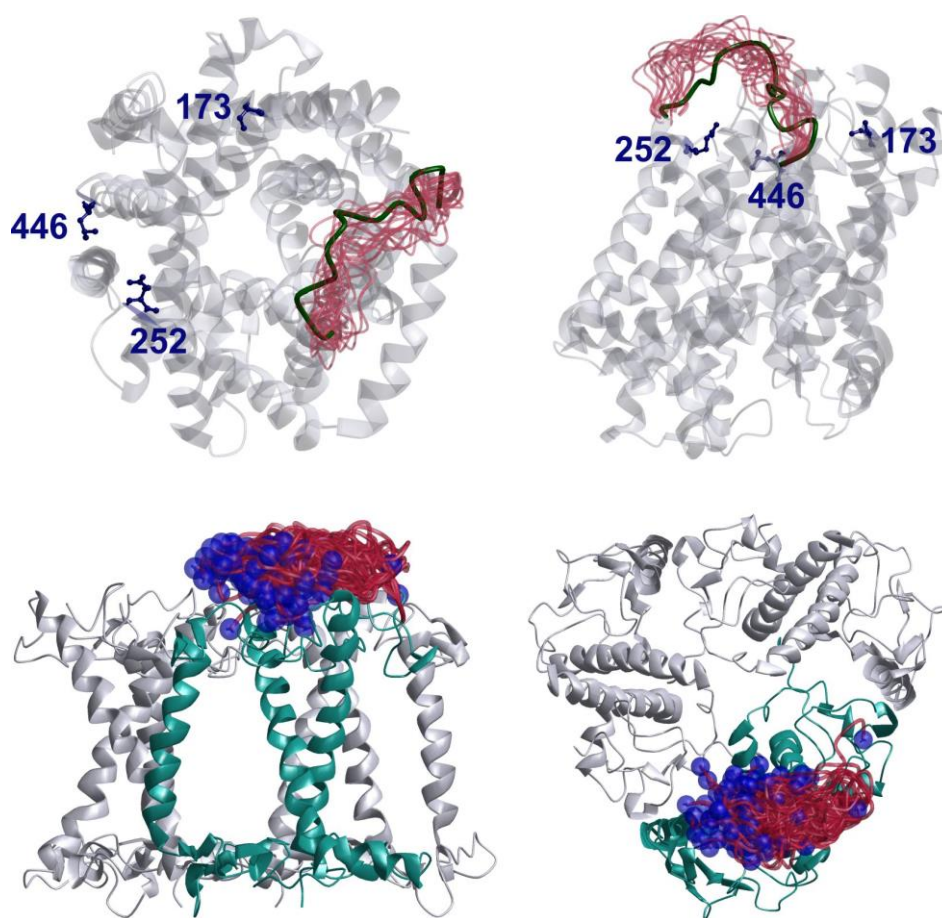


# MMM Version 2015.1

## Multiscale modeling of macromolecular systems

*Yevhen Polyhach and Gunnar Jeschke, with design support by Enrica Bordignon, ETH Zurich; Stefan Stoll, University of Washington*

An open-source program for visualization, inspection, and improvement of models of proteins and protein assemblies based on restraints from multiple experimental techniques.



### *Front cover*

The picture shows modeling of disordered domains of proteins based on long-range distance distribution restraints (LRDDR). The upper two panels depict the outcome of a test, where the well-defined loop of residues 71-87 (green coil) was remodelled by an ensemble of conformations based on 24 LRDDR to spn-labelled beacon residues 173, 252, and 446. In this case, the restraints were created in silico. The lower two panels depict modelling of the N-terminal residues 3-13 of major plant light harvesting complex LHCII from restraints internal to the modelled loop, beacon restraints, homotrimer restraints, and bilayer immersion depth restraints. See [changes in version 2015.1](#) for a short description of these new functions.

The corresponding restraint files can be found in the [demo](#) directory of MMM.

## **Preface**

This manual is an automatic compilation of MMM help files and is provided for overview. The PDF version shows links in blue color, but these links are not active. Please use the on-line help if you want to browse via the links. Links to other help files were converted into cross references to subsections or subsubsections (e.g., *Section 3.2* or *Section 1.2.12*). Subsections and subsubsections correspond to individual help pages. Sections (e.g., 1 General) group help pages. Links to different paragraphs in the same help file do not result in cross references in this manual.

# Contents

<b>1 General</b>	<b>6</b>
1.1 Overview	6
1.2 Getting started	7
1.2.1 Demo model of LHCII	9
1.3 Change history	13
1.4 Model concept	19
1.5 Object concept	21
1.6 Addressing	22
1.6.1 Substructure addressing	26
1.7 Visualization	28
1.7.1 Density cubes	30
1.8 Selection	30
1.9 Annotation	31
1.10 References	34
1.10.1 Search PubMed	43
1.10.2 Define citation format	48
1.11 Find by keyword	50
1.12 Information in PDB files	52
<b>2 User interface</b>	<b>57</b>
2.1 Main window	57
2.1.1 Menu/File	57
2.1.2 Menu/Edit	57

<b>2.1.3 Menu/Build</b>	<b>65</b>
<b>2.1.4 Menu/Display</b>	<b>68</b>
<b>2.1.5 Menu/Analysis</b>	<b>70</b>
<b>2.1.6 Menu/Dynamics</b>	<b>73</b>
<b>2.1.7 Menu/EPR</b>	<b>74</b>
<b>2.1.8 Menu/Biochemistry</b>	<b>76</b>
<b>2.1.9 Menu/Predict</b>	<b>76</b>
<b>2.1.10 Menu/Help</b>	<b>78</b>
<b>2.1.11 Toolbar</b>	<b>79</b>
<b>2.1.12 View control</b>	<b>84</b>
<b>2.1.13 Depth cueing</b>	<b>85</b>
<b>2.1.14 Command line</b>	<b>87</b>
<b>2.1.15 Model panel</b>	<b>88</b>
<b>2.1.16 Model window</b>	<b>90</b>
<b>2.1.17 Message board</b>	<b>92</b>
<b>2.2 Commands</b>	<b>93</b>
<b>2.3 Hierarchy window</b>	<b>112</b>
<b>2.4 Color selection</b>	<b>115</b>
<b>2.5 Transparency selection</b>	<b>116</b>
<b>2.6 Graphics mode</b>	<b>116</b>
<b>2.7 Report editor</b>	<b>117</b>
<b>3 Methods</b>	<b>118</b>
<b>3.1 Spin labeling and site scans</b>	<b>118</b>
<b>3.2 DEER simulation and fit</b>	<b>126</b>
<b>3.3 Localization of spin labels</b>	<b>133</b>

<b>3.4 Networks of sites</b>	<b>138</b>
<b>3.5 Docking of protomers in a complex</b>	<b>144</b>
<b>3.6 Domain ensemble</b>	<b>147</b>
<b>3.7 Coarse graining</b>	<b>158</b>
3.7.1 Elastic network models	160
3.7.2 Gaussian network models	162
3.7.3 Anisotropic network models	164
3.7.4 Fit from template	167
<b>3.8 Fit from template/Modeller</b>	<b>175</b>
<b>3.9 Scripting</b>	<b>182</b>
<b>3.10 Superimposing structures</b>	<b>185</b>
<b>3.11 Binding sites</b>	<b>187</b>
<b>3.12 Assign transmembrane helices &amp; strands</b>	<b>189</b>
<b>3.13 Build lipid bilayer</b>	<b>191</b>
<b>3.14 P-31 Mims ENDOR simulation</b>	<b>194</b>
<b>4 Protocols</b>	<b>198</b>
<b>4.1 Protocol for spin labeling and DEER</b>	<b>198</b>
4.1.1 Loading the protein	198
4.1.2 Site scan protocol	201
4.1.3 DEER simulation protocol	209
<b>5 Conditions and Acknowledgment</b>	<b>211</b>
<b>5.1 License</b>	<b>211</b>
<b>5.2 Third-party software</b>	<b>212</b>
<b>5.3 Credits</b>	<b>216</b>

# 1 General

## 1.1 Overview

### What is MMM?

MMM is an open-source Matlab program for [multiscale modeling](#) (Section 1.4) of macromolecular systems, in particular proteins and protein complexes. [Modeling in MMM](#) (Section 1.4) is based on experimental restraints, canonical bond lengths, bond angles and conformations of secondary structure elements, as well as structural homology.

Models consist of a hierarchical assembly of [objects](#) (Section 1.5) (structure elements) so that different parts of the structure can be represented with different resolution.

MMM can also be used to inspect [PDB files](#) (Section 1.12) for more than just the structural information. MMM communicates to you mainly via the [message board](#) (Section 2.1.17) of the main window. Even if you work with another window, you should keep an eye on the message board. Occasionally MMM wants to be sure that you notice a warning or error message and will then pop up a message window.

### Basic functionality

MMM also provides [protein visualization](#) (Section 1.7) with (future) emphasis on structural disorder or uncertainty and systematic coarse graining. There is a number of possibilities to [select objects](#) (Section 1.8) for visualization and modeling.

Models in MMM can be [annotated](#) (Section 1.9) on the level of structure elements (objects), such as peptide chains, residues, or atoms and [references](#) with full-text access information can be stored with the model and linked to structure elements. Structures can be [analyzed](#) in terms of the structural context of objects, distances, angles, and dihedral angles.

### Special functionality

Experimental data for several techniques, mainly electron paramagnetic resonance (EPR) techniques, can be predicted for a modelled structure. For that, [spin labels](#) (Section 3.1) can be attached to the protein using a rotamer library approach and favorable attachment sites can be predicted by scanning the whole protein. MMM also provides convenient access to servers for [prediction](#) (Section 2.1.9) of secondary, tertiary, and quaternary structure, disorder and structural transitions.

## Fitting with distance restraints


Conformational transitions can be fitted from spin label (or  $C^\alpha$ - $C^\alpha$ ) distance restraints and a PDB structure for the initial state by using an elastic network model. See [Fit from template](#), (Section 3.7.4) which can be accessed from the **Build** menu.

Peptide chains can be modelled from distance restraints and optional additional restraints on secondary structure or immersion depth of a residue in a lipid bilayer using the [domain ensemble](#) (Section 3.6) builder. This feature is particularly suitable for intrinsically disordered domains or intrinsically disordered proteins.

## User interface

The [user interface](#) of MMM consists of a main control window and optional windows for display of the three-dimensional structure and of the hierarchy of structural objects. Further windows open for specific tasks, such as spin labeling site scans or comparison of predicted with experimental distance distributions. Part of the functionality of MMM is available via a [command interface](#) (Section 2.2) and a [script interpreter](#). (Section 3.9)

## Help

General help can be accessed via the menu and context-sensitive help for the current window via a help button .

# 1.2 Getting started

## Installation

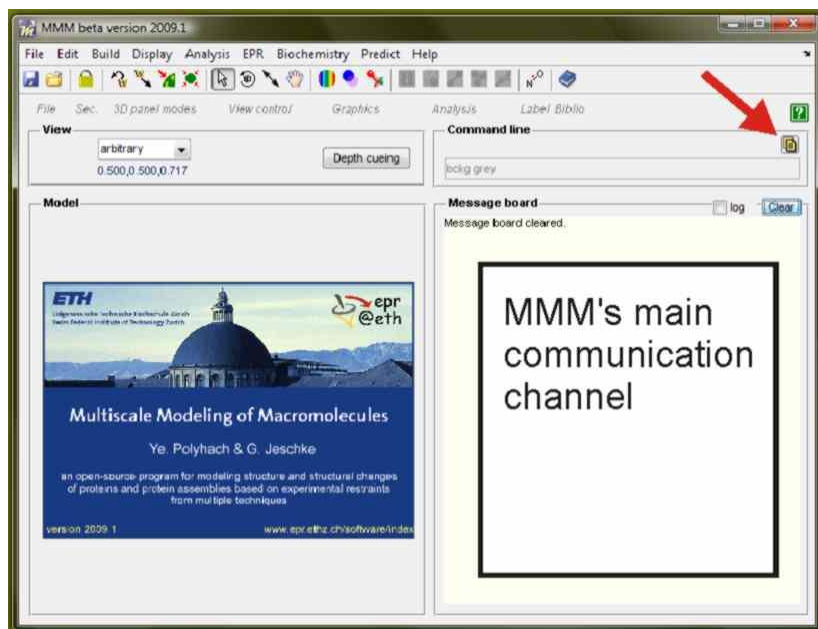
MMM comes as a ZIP file with everything you need, except for optional [third-party software](#) (Section 5.2) that you need to download and install yourself for licensing reasons. Unpack the ZIP file into a directory of your choice. Add the directory to the Matlab path, for instance by using the **File/Set path...** menu item in the Matlab main window. Save the Matlab path by clicking the **Save** button if you want to keep it for future Matlab sessions. If you click the **Apply** button the path is set only for the current session.

You can now start MMM either via the Matlab Start button (left bottom corner of the main window) by selecting item **Toolboxes/MMM/Start** or by typing **MMM** in the Matlab command window.

For better performance, you may want to change the preferred PDB server (faster downloading of PDB files) and the SFX server (easier fulltext access for references without digital object identifier). Go to the [File/Preferences](#) (Section 2.1.1) menu item. Change the SFX server address in the edit field to the one of your home institution. If you do not have one, you can try the one of TU Berlin given above the edit field. This SFX server first creates the DOI the next step brings you to the journal homepage,

which will recognize whether your institution has a subscription to this journal. Nevertheless, it is strongly recommended that you find the address of your own institution's SFX server (Berlin's taxpayers may not want to sponsor your research).

## Demo script



You can load a demo script `cover_picture_2009.mmm` via the [script](#) (Section 3.9) load button marked with a red arrow in the screenshot above. It is recommended to close and restart MMM before using this script, so that results really look as described in the following. This script demonstrates several features of MMM, including [PDB download](#), (Section 2.2), structure display, selection of residues and cofactors, coordinate frame transform to make a symmetry axis the new z axis, display of a solvent accessible surface (if Michel Sanner's [MSMS](#) (Section 5.2) program is also installed), spin labeling, and spin label display. The script will run without MSMS being installed, just the solvent accessible surface will not be computed and displayed.

If you are not connected to the internet, you need to use the script `cover_picture_2009_no_internet.mmm`.

Running the script takes about ten minutes on a not too old laptop computer, it is usually faster with a desktop computer. Update of the 3D model display may be a bit slower than update of the Message board, hence, display update may lag behind the explanation. The script demonstrates capabilities of MMM and working with [commands](#) (Section 2.2) and [scripts](#). (Section 3.9). The same functionality is accessible via the graphical [user interface](#).

After running the script, the cover picture is in the clipboard.



## Demo PDB file and initialization script

The folder `LHCII` contains the PDB file `2BHW`. You can load this via the [File/New/From PDB.../local file](#) (*Section 2.1.1*) menu item. An initialization script `2BHW.mmm` exists in the same folder. This script shows how to define secondary structure when loading a PDB file and how to control initial model display. You may also decide not to use this script (MMM asks). You will then end up with standard display and secondary structure assignment from the PDB file. You can obtain a better automatic secondary structure assignment by installing the [DSSP](#) (*Section 5.2*) program of Kabsch and Sander.

The model freshly created from the PDB file will already contain a few [references](#) (*Section 1.10*) and will already have a few automatic [annotations](#) (*Section 1.9*) that can be inspected most easily with the keyword finder accessible via the [Edit/Find by keyword](#) (*Section 2.1.2*) menu item.

Demo model

MMM comes with a demo model of major plant light harvesting complex LHCII, which you find in the subfolder LHCII as file `LHCII_demo_model.mat`. This model can be loaded via the [File/Open...](#) (*Section 2.1.1*) menu item. It demonstrates concepts such as superposition of two structures, synonyms for structure names, annotation of a model, using a bibliography, and automatic literature search. See [description of the LHCII model](#) (*Section 1.2.1*) for a more detailed explanation.

1.2.1 Demo model of LHCII

## Concepts

This model of major plant light harvesting complex LHCII demonstrates how information on a biostructural problem can be organized in MMM. LHCII is a complex of a 25 kDa membrane protein with eight chlorophyll *a*, six chlorophyll *b* and four carotenoid molecules that forms trimers and crystallizes with some lipid molecules in well defined positions. Two highly resolved crystal structures are available, with PDB identifiers 1RTW (spinach)<sup>1</sup> and 2BHW (pea).<sup>2</sup> Functions of LHCII involve collection of photons from sunlight and transfer of the excitation energy towards photosynthetic reaction centers,<sup>3</sup> non-photochemical quenching of excitation energy under excessive light irradiation,<sup>4</sup> regulation of the balance between reaction rates at photosynthetic reaction centers I and II,<sup>5</sup> and probably promotion of stacking of the thylakoid membrane in chloroplasts (grana formation).<sup>6</sup>

The demo model uses the following concepts:

- [Superposition of two structures](#)
- [Synonyms for structure names](#)
- [Annotation](#)
- [Bibliography \(literature references\)](#)
- [Automatic periodic reference list update](#)
- [Spin labeling sitescan](#)

## Superposition of two structures

The two structures are very similar.<sup>3</sup> To visualize this, the PDB file 1RWT was edited to retain only one trimer (chains B, F, and G). The edited file is available in the LHCII subfolder as [1RWT\\_BFG.pdb](#). The original PDB file could also have been used, but this would have made the model unnecessarily large.

Structure 2BHW was [transformed into its symmetry frame](#), (*Section 4.1.1*), where the C3 axis of the trimer is the z axis. The viewing direction is along the C3 axis, which is coincident with the membrane normal. Structure 2BHW was displayed as a ribbon model with colorscheme chain, i.e. the three chains A, B, and C are colored on a red via green to blue scale. The edited structure 1RWT was [superimposed](#) (*Section 2.1.2*) onto the structure of 2BHW by minimizing the root mean square deviation of the backbone atom coordinates of residues 55-86 of chain B from the corresponding atom coordinates of chain B of structure 2BHW. Chains B, F, and G of structure 1RWT were displayed as a grey ribbon model.

The superposition is almost perfect, despite the fact that the two sequences differ in 16 amino acids. You can see these differences by [selecting](#) (*Section 1.8*) chains B of both models in the hierarchy window and using the [Analysis/Sequence alignment](#) (*Section 2.1.5*) menu item

The frame transformations and the editing of 1RWT were annotated at structure level. This is good practice- if you change anything with respect to the original structure, you should annotate it. Some changes, such as spin labeling, are automatically annotated by MMM.

## Synonyms for structure names

PDB identifiers are somewhat cryptic. The two structures are from proteins of two different plants, spinach and pea. Hence, the [synonym](#) (*Section 2.2*) "spinach" was assigned to 1RWT and the synonym "pea" to 2BHW. Thus, [objects](#) (*Section 1.5*) in the two structures can be [addressed](#) (*Section 1.6*) with structure name 1RWT or spinach resp. 2BHW or pea. In its own address display, MMM uses the last defined synonym, in this case the plant names.

Although this is not used in the LHCII model, chains can also be assigned synonyms. This is useful for complexes of several different proteins, where chains are better addressed by their protein name than by a one-letter chain identifier (which might even differ between different structures of the same complex).

## Annotation

Information on structure/function relationships is in terms of interactions between parts of the structure (objects in MMM) with other parts of the structure or with substrate molecules or in terms of the consequence of a structural change on properties of the protein. Such information should be stored with the structural model and it should be assigned to the part of the structure that is concerned.


Hence, MMM allows for [annotating](#) (Section 1.9) whole structures, chains, residues, atoms, and atom locations with free-text comments. Furthermore, keywords can be assigned to such objects, so that annotations for a certain concept can be found easily. Finally, objects can also be assigned literature [references](#). Hence, the annotation of an object can also inform the user of the model which papers refer to this particular object, i.e., to a residue with known functional relevance.

The LHCII demo model is heavily annotated, mainly based on Ref. 3. An overview is obtained most easily via the [Edit/Find by keyword](#) (Section 2.1.2) menu item, which opens the [Keyword finder](#). (Section 1.11)


The keyword list is alphabetically ordered. As an example, click on the keyword `conserved residues`. The listbox on the right now displays a list of all residues that were annotated as reserved. The first one, `[pea] (A) {1}139; GLU` is selected. Click the button `Show annotation` and answer the query about discarding the old selection with `Yes`. The annotation window for this residue opens. To find the annotation page about residue conservation you have to scan through the pages with `Page` control in the upper left corner of the window. You find the annotation on p. 4. You can close the annotation window with the `OK` button.

MMM also makes some automatic annotations when [reading the PDB file](#) (Section 1.12) or when a command leads to changes in the model. Slide the keyword list down and select `metal centers`. In the `Referring objects` list on the right, select `[pea] (A) {1}601.MG; CLA`. Click `Show annotations`. For this automatic annotation, the annotation window automatically opens at the corresponding page. You can see here which atoms in the structure coordinate the magnesium atom of CLA 601.

## Bibliography (literature references)

MMM includes a basic [reference manager](#), (Section 1.10) which is accessible via the  toolbutton. The LHCII demo model contains 51 references on structure and function of LHCII and on structures of closely related light harvesting complexes.

You can inspect a reference by clicking on the `Abstract` button in the reference window. The abstract appears in the web browser, if the reference is known to PubMed (see `Reference number 5` as an example). When the abstract is found on PubMed, you are asked whether you want to use the PubMed record for update of the bibliographic information. This is particularly useful for references that were "in press" or available only on-line when they were first imported into the model (update then provides the full information, once available). If the reference is unknown to PubMed, no web browser window opens, but a message is displayed in the `Message board` of the main window (see `Reference number 51` as an example).

You may have fulltext access via the  fulltext button, depending on entitlements of your institution. If the bibliographic information includes a digital object identifier (DOI), a web browser is opened directly at the journal homepage for this paper (see `Reference number 5` as an example). If not, MMM tries to localize the reference via SFX (see `Reference number 50` as an example). The default setting of MMM

specifies the SFX server of ETH Zurich. You may want to use the SFX server of the library of your home institution instead. Only this server will guide you to fulltext access for papers that are not available free of charge, but are subscribed by your institution.

## Automatic periodic reference list update

The reference manager of MMM has a [search function for PubMed](#). (Section 1.10.1) By using the **Autosearch** button in the search window, you can store search terms and the date when this particular search was performed. Whenever the model is opened later, MMM checks how old the last search is and searches again if more than seven days have passed (only new references from that time period are added or updated).

The LHCII demo model contains two autosearches for reviews on LHCII, one with the search term "LHCII" and another one with "LHC-II". You can inspect these searches via the **Autosearches** listbox in the search window. Above the listbox you find a **Deactivate** button for deactivating this search without deleting it, and a **Delete** button for deleting it. If a search is deactivated, an **Activate** button is available for reactivating it. Only the active autosearches are performed when loading the model.

## References

- [1] Liu, Z., H. Yan, K. Wang, T. Kuang, and J. Zhang *et al.*. 2004. Crystal structure of spinach major light-harvesting complex at 2.72 Å resolution. *Nature*. 428:287-92. [2] Standfuss, J., A. C. Terwisscha van Scheltinga, M. Lamborghini, and W. Kühlbrandt. 2005. Mechanisms of photoprotection and nonphotochemical quenching in pea light-harvesting complex at 2.5 Å resolution. *EMBO J.* 24:919-28.
- [3] Barros, T. and W. Kühlbrandt. 2009. Crystallisation, structure and function of plant light-harvesting Complex II. *Biochim. Biophys. Acta*. 1787:753-72.
- [4] van Grondelle, R. and V. I. Novoderezhkin. 2006. Energy transfer in photosynthesis: experimental insights and quantitative models. *Phys. Chem. Chem. Phys.* 8:793-807.
- [5] Allen, J.F. and J. Forsberg. 2001. Molecular recognition in thylakoid structure and function. *Trends. Plant. Sci.* 6:317-26.
- [6] Dekker, J.P. and E. J. Boekema. 2004. Supramolecular organization of thylakoid membrane proteins in green plants. *Biochim. Biophys. Acta*. 1706:12-39.

## Spin labeling sitescan


A [spin labeling site scan](#) (Section 3.1) was performed on chain A of structure 2BHW. An [overview of the results](#) is available in HTML format.

## 1.3 Change history

### Changes of version 2015.1 with respect to 2013.2

This version is compatible with the new graphics engine of Matlab R2014. As a consequence, rotating and scaling of models is much faster, whereas generation of model graphics is somewhat slower (if Matlab 2014 or later is used).

#### Changed functionality

- The default rotamer libraries for MTSL have been replaced by new versions based on Monte Carlo scans of torsion angle space and the UFF force field. These new libraries are parametrized for a weak S $\delta$ -C $\alpha$  interaction and perform slightly better as judged by tests against a set of 80 distance distributions for doubly spin-labelled T4 lysozyme. The old libraries are still available by direct selection if the lock toolbutton  is deactivated before rotamer computation.
- Chains with more than 1'000 residues can now be displayed and worked with in the hierarchy window.
- The previous limit of 20'000 atoms per chain for reading PDB files is lifted. MMM now first scans the PDB file for the highest atom number and then allocates sufficient memory.
- The [rotamers command](#) (Section 2.2) now defaults to ambient temperature (298 K) rather than cryogenic temperature (175 K), as systematic comparison with experimental data has shown that ambient-temperature computations agree better with experiment.

#### New functionality

- With the [domain ensemble](#) (Section 3.6) window, accessible via the [Build menu](#) (Section 2.1.3) it is now possible to model peptide chain segments that are not resolved in a crystal structure. The Monte Carlo modeller supports N-terminal, C-terminal and inserted loops and label-to-label distance distribution restraints, bilayer insertion depth restraints, and secondary structure restraints. Special handling is provided for distance distribution restraints stemming from homooligomers in which all protomers are singly labeled at the same site.
- Additional spin labels are available: IA-Proxyl attached to 4-thiouracil (IAP-4TU), Maleimido-Proxyl (MA-PROXYL), brominated MTSL (Br-MTSL), M8DOTA-Gd, MTS attached to 4-thiouracil (MTS-4TU), label R5 attached to thiophosphate of a nucleotide (R5-TP), spin-labeled vitamin B12 (CNC-NO), Maleimido-Gd-DOTA (DOTA-Gd), Maleimido-Gd-DTPA (DTPA-Gd), and the Hubbell-Fleissner unnatural amino acid label K1 (HF-K1).
- The [DEER window](#) (Section 3.2) now allows for computing distance distributions averaged over an ensemble, for instance for NMR structures. The feature can be activated or deactivated with a checkbox above the label selection listbox. For non-ensemble structures, the feature has no effect. The feature is not applicable for multi-spin effects (more than two labels selected). In that case, a warning is displayed in the main window and the checkbox is automatically deactivated.
- The [magic fit feature in the Edit menu](#) (Section 2.1.2) can now be used to superimpose individual models in an ensemble structure onto a target structure. See also the description of the [magic fit window](#). (Section 3.10)

- The [libcomp command](#) (Section 2.2) allows for systematic comparison of spin label position predictions by two different rotamer libraries. It can be used for comparing alternative libraries for the same label or for comparing different label types.
- The [A plot command](#) (Section 2.2) allows for plotting lines between two atoms specified by their MMM [addresses](#). (Section 1.6) This is useful for visualizing constrained distances in a model.

## Bug fixes

- The check for agreement between sequence definition in PDB files (SEQRES) and actual sequence did not work for DNA and RNA. DNA and RNA sequences from PDB files with SEQRES records are now properly displayed in the Hierarchy window.

## Changes of version 2013.2 with respect to 2013

### New functionality

- [Grid search based rigid-body docking](#) has been added to the [Predict/quaternary menu](#). (Section 2.1.9) Given at least 4, but preferably more, distance constraints between protomers in a homooligomer or at least 6, but preferably more distance constraints between the two protomers in a heterodimer, MMM can perform an exhaustive search of parameter space to determine relative orientation and translation of the protomers. The grid search result can be automatically refined by non-linear least squares minimization of the distance constraint deviation. On the occasion of adding this functionality to the Predict menu, links in this menu were updated, outdated links removed, and a new prediction site for conserved domains and a link to the ExpASY site added.
- Given at least three, but better at least five distance distributions between reference sites and a point, this point can be [localized](#). (Section 3.3) The distributions can be provided as mean distance/standard deviation or as DeerAnalysis output files. Visualoutput indicates uncertainty of the localization. A PDB file can be written that characterizes uncertainty by an ensemble of models for the localized point.
- Given at least 3M-6 pairwise distance distributions in a set of M points, a [network model](#) (Section 3.4) can be created for the set of points. If at least three points are reference sites in a structure, the network model is fitted to the structure. The distributions can be provided as mean distance/standard deviation or as DeerAnalysis output files. Visual output indicates uncertainty of the network points. A PDB file can be written that characterizes uncertainty by an ensemble of models for the network. This is a slightly extended implementation of a distance matrix geometry approach suggested in [Gaffney et al., Biophys. J. 2012, 103, 2134-2144](#), which should be cited when using this feature.

### Bug fixes

- A flag for separate coloring of N-O midpoint spheres (`show adr label`) for the  $\chi_3$  p/m rotamers was inadvertently switched on in the first 2013 distribution. This led to an error when displaying N-O midpoint spheres for IA-Proxyl labels. The flag is now switched off. It can be switched on (line 23 of `plot_label.m` in the `\private` subdirectory) to have separate coloring for MTSL. For IA-Proxyl, this flag is now ignored. Thanks to Enrica Bordignon for reporting.
- When rotamers were actually attached to the structure and a PDB file was written out, superposition of the backbone atoms of the labeled residue was slightly off. This



led to a small change in C $\alpha$ -C $\beta$  distance and a slight tilt of the sidegroup. Predictions of distance distributions or other properties of the rotamer distribution were not affected. The problem has been fixed. Thanks to Gregor Hagelueken for reporting.

## Changes of version 2013 with respect to 2011.2

### New functionality

- Estimate of the quality of models for structural transitions was added in menu [Analysis/Model comparison](#). (Section 2.1.5) You need an elastic network model and a homology model for the same structural transition to use this.
- Spin labeling and spin labeling site scans now allow for using adapted rotamer libraries. To protect inexperienced users, this feature is locked (similar to secondary structure editing). See [Labeling conditions](#).
- Rotamer computations can now be tested systematically and automatically against sets of experimental mean distances measured on known protein structures. See [libtest command](#). (Section 2.2)
- The PDB reader now tolerates CYANA output PDB files that contain pseudo-atoms and XPLOR files that contain dummy atoms.
- [Interfacing to the Tinker suite](#) (Section 5.2) of molecular force field and molecular dynamics programs has been prepared (not yet used in standard MMM functionality).
- Rotamer libraries can now define their own optimized 'forgive factor'. The new adapted libraries do so.

### Bug fixes

- Setting the x, y, or z axis to the viewing direction (Build menu) sometimes deformed the structure. This was fixed.
- Missing Latex font issue in the reference window (on some computer systems) was solved. The reference window now supports the new file extension `.ciw` of the ISI data base bibliography output (Web of Science).

## Changes of version 2011.2 with respect to 2011.1

### New functionality

Fitting of conformational changes with Modeller and of constraint-supported homology modeling in general is implemented in the `Build` menu as [Fit from template/Modeller](#). (Section 3.8)

The algorithm for elastic network model-based fitting of conformational changes is improved and for both features, help files were extended.

## Changes of version 2011.1 with respect to 2011

### New functionality

Fitting of conformational changes with an elastic network model approach is implemented in the `Build` menu as [Fit from template/ENM](#). (Section 3.7.4)

## Changes of version 2011 with respect to 2010

### Bug fixes

- Some issues with PDB reading and saving (incompatibilities with some formats and output errors in certain contexts) were fixed.
- Reference import from SciFinder was corrected to obtain a consistent output format for references retrieved by different search engines.
- Double-dash for page ranges was missing in Latex bibitem, this was corrected.

### New functionality, speed-up and better handling

- Rotamer computation and site analysis are faster by more than a factor of ten.
- The HTML output format for site scans was improved (more compact, easier to read).
- Elastic network models can be derived and analyzed, see [Dynamics menu](#). (Section 2.1.6)
- Protein motion can be visualized using the [motion command](#). (Section 2.2)
- Modeller from the Sali lab was interfaced for loop reconstruction, see [Third-party software](#). (Section 5.2) Homology modelling will follow later.
- [MUSCLE from Robert C. Edgar](#) (Section 5.2) was interfaced for sequence alignment and is bundled with MMM, see [Analysis menu](#). (Section 2.1.5)
- Structure superposition of different proteins can now be based on sequence alignment, see [Structure superposition window](#). (Section 3.10)
- A new tool button (red St Andrew's cross) allows to cancel all current selections.
- Reference formatting with author list abbreviation by *et al.* now allows for a different maximum number of authors without *et al.* and to be listed if *et al.* is used. See `def_reference_formats.m` definition `reference_formats(19)` for an example of how to do that.

## Changes of version 2010 with respect to 2009

### Bug fixes

- version 2009 did not bond nucleotides in RNA and DNA, did not recognize DNA/RNA sequences, and had no ribbon graphics for DNA/RNA
- when selecting a residue in version 2009, atom graphics was not highlighted yellow
- subsequent coordinate transformations (center, symmetry) sometimes lead to wrong center coordinates in version 2009
- in some PDB files, sequence information (SEQRES) does not match coordinate information, MMM 2009 corrected for this only for single-chain proteins, now it works also for protein oligomers
- spin labeling site scans of selected residues did not properly distinguish between intrachain and interchain pairs
- spin labeling failed when a structure contained elements whose atomic number was larger than the one of the last defined element (iron)
- some official PDB files have wrong formatting of the metal coordination remark (example: 1K4C), these ones could not be read by MMM 2009, now they can be read, but metal coordination information is not repaired
- on attachment of labels, internal sequence representation was not updated, i.e. the native residue type was still displayed in the hierarchy window
- A new bug appeared in Matlab 2010a- the internal web browser now fails with many web pages. This was fixed by making the default browser of the operating system



also MMM's default browser. As some older Matlab versions have some problems with transmitting the URL to some browsers, you can still select use of the Matlab browser under [Menu/File/Preferences](#). Don't forget to save your preferences.

- MMM can now read PDB files with negative residue numbers and residue number 0. Astonishingly, this is not excluded by PDB format specification and such files exist on the PDB server. MMM uses renumbering of residues in this case.
- [colorscheme sequence](#) is now properly based on the number of residues in the peptide or nucleotide chain, excluding cofactors and water

## New functionality

- Spin label rotamer computation automatically disregards water molecules in the structure.
- DEER formfactor simulations take into account multi-spin effects if more than two spin labels are selected, see description of [DEER window](#). (*Section 3.2*)
- To provide complete and precise control of model view direction, the camera up vector is now displayed in the view panel and can be set by the [camup command](#) (*Section 2.2*) and by the [view command](#) (*Section 2.2*) with six arguments.
- The model view, including zoom, can now be stored with the **M** button in the [View panel](#), and can be retrieved later with the [memory](#) subitem of the [View panel](#) popupmenu.
- When saving a model in MMM format, the current view, including zoom, and the detachment state of the 3D display are stored. When a model saved in MMM 2010 or later is reloaded, these properties are restored.
- The 3D plot can be zoomed in/out in any state of view control with the mouse scroll wheel.
- A new [colorscheme](#) (*Section 2.2*) has been added that helps to visualize variability in NMR structure ensembles or structure ensembles generated by elastic network model based fitting from a template.
- A new [colorscheme difference](#) (*Section 2.2*) has been added that visualizes residue-by-residue differences between two structures (accessible only via command line). Lipid bilayers can be inserted into a model and can be fitted to  $\alpha$ -helical bundles or  $\beta$ -barrels ([Build/Bilayer](#) (*Section 2.1.3*) menu item).
- P-31 Mims ENDOR spectra can be predicted for structures that contain phosphorous nuclei, in particular DNA and RNA, and for structures that contain a lipid bilayer.
- Solvent accessibilities can be computed for residues and cofactors if [MSMS](#) (*Section 5.2*) is installed ([Analysis/Accessibility](#) (*Section 2.1.5*) menu item). This feature is experimental.
- Helices and strands can be assigned as transmembrane (TM) and the structure can be transformed into a frame where the mean axis of an  $\alpha$ -helical bundle or  $\beta$ -barrel is the z axis ([Edit/Assign TM helices and strands](#) (*Section 2.1.2*) menu item).
- If DSSP information is present, secondary structure can be reassigned according to this information. Such information is created when reading a PDB file, if [DSSP](#) (*Section 5.2*) is installed. Differences to secondary structure information in original experimental PDB files is minor, but it may help in modelling.
- Structures can be transformed into the current viewing frame, so that atom coordinates coincide with current appearance of the model ([Build/Transform to viewing frame](#) (*Section 2.1.3*) menu item).
- Structure superposition with the [Edit/Superimpose structures...](#) (*Section 2.1.2*) menu item can now check residue correspondence. The feature also works (when disabling this check) if in both structures the same number of objects of the same type (for instance 40 residues each) are selected. This objects are superimposed in the sequence of selection.

- A very basic editor for display of and minor changes in plain text files was added ([Edit/Reports](#) (Section 2.1.2) menu item).
- Sidechains can be repacked (or generated), if the third-party software [SCWRL4](#) (Section 5.2) is installed. Use the [Build/Grow/repack sidechains](#) (Section 2.1.3) menu item for instance to test how sensitive predicted spin label rotamers are to sidechain positions.
- Structures can be copied, so that changed structures can be easily compared to the original structure ([Build/Copy structure](#) (Section 2.1.3) menu item).
- Non-native residues can be replaced by their native equivalents ([Build/Replace non-standard residues](#) (Section 2.1.3) menu item), for instance selenocysteine and selenomethionine by cysteine and methionine, respectively. Use this if you want to compute spin label rotamers at such sites. This feature also allows to remove spin labels (residue is replaced by cysteine, orientation of C<sup>β</sup> and S corresponds to leading rotamer).
- Commands [repack](#), (Section 2.2) , [scopy](#), (Section 2.2) , and [replace](#) (Section 2.2) provide script access to these new functions.
- Crystal contacts can be analyzed ([Edit/Reports](#) (Section 2.1.5) menu item) via repacking sidechains by SCWRL4 in the absence and presence of the crystal environment and visualizing residue-by-residue differences with `colorscheme difference`.
- The `Analysis` menu also provides access to crystal contact and contact of structural units analyses via the OCA server at Weizmann institute.
- The hierarchy window has two new buttons in the `Structure` field for direct access to PDBwiki and electron density visualization for this structure. The PDBwiki page contains many further links. The electron density server also contains very useful other links.
- More pseudo-PDB formats (PDB output files of other software that do not conform to the specified PDB format) can be read by MMM.
- Masses of structures, chains, and chain models are now reported without counting water molecules. The command `mass` has an option `-water` to count water.
- Determination of (pseudo-)symmetry axes and symmetry transform now generally disregards water molecules. MMM checks whether third-party software (MSMS, DSSP, SCWRL4) is accessible on the Matlab path. If not, menu items are blocked that depend on such software.
- Unit cell information for crystal structures (CRYST1, ORIGXn, SCALEn records) is now read from and written to PDB files. Be aware that this information may not be meaningful in PDB output files if you have changed the structure.
- Sessions are now logged by default (you can switch this off with the checkbox `log` at the top of the `Message board` panel).
- Logging of the session is now done by default (you can switch it off with the checkbox `log` at the top of the `Message board` panel).
- The log file is easily accessible by the `Show log` button at the top of the `Message board` panel. This provides full access to history of the Message board and of the command line.
- Bug reports can now be created for sending to us ([Help/Create bug report](#) (Section 2.1.10) menu item).

## 1.4 Model concept

### Introduction

An MMM model is a collection of related structures. The individual structures may correspond to different states of the same protein, they may be different components of a protein complex, or they may be related by structure homology of the whole biomacromolecule or of domains. These structures are defined with respect to a common coordinate frame, the model frame.

A structure consists of one or more biomacromolecules, cofactors, and auxiliary objects, such as a lipid bilayer, a micelle, a surface, or an AFM tip. A biomacromolecular complex can thus be defined either on model level, with the individual components as different structures or on structure level. Interconversion between these two modes is possible: several structures within a model can be merged to a single structure and a single structure can be subdivided into several structures.

A structure is a hierarchical collection of objects. For instance, it may consist of several protein molecules (chains), which in turn consist of residues, which in turn consist of atoms, for which in turn several locations may be defined. This [object concept](#) (Section 1.5) allows for a simple [addressing of substructures](#). (Section 1.6)

The concept of MMM is not restricted to biomacromolecules. Supramolecular assemblies of small molecules (analogous to cofactors) and synthetical macromolecules can be treated in the same way. This requires definition of the components and monomers in a similar way as cofactors and amino acid residues are defined in the protein data base (PDB).

### Multiscale models

Function of biomacromolecules is based on structural dynamics on different length and time scales. Furthermore, experimental techniques for the characterization of structure and dynamics of macromolecules provide information on different length and time scales with different resolution. For these reasons, structure and structural dynamics are defined and- should be analyzed- on different length and time scales. It follows that knowledge of a static atomistic structure is neither a necessary condition for understanding aspects of protein function nor a sufficient condition for completely understanding it. Systematic approaches for converting a detailed (e.g., atomistic) model to a model with lower resolution are [coarse graining](#) (Section 3.7) approaches.

Modeling of biomacromolecular complexes should thus strive to define structures and state transitions between alternative structures on the appropriate length scales for understanding function. This length scale may be different for different parts of the structure and for different aspects of function. For instance, function of the active centre of a metalloprotein can only be fully understood at atomistic resolution, with additional information on electronic structure being required. Access of the substrate

to the active site and release of the product may depend on larger scale movements of protein domains that are better specified in a [coarse grained](#) (*Section 3.7*) or cartoon model.

Hence, MMM allows for specifying different parts of the same structure with different resolution and on different levels of coarse graining. In fact, even the same part of the structure can be defined on different levels of coarse graining to facilitate visualization of different aspects of function.

Such multiscale modeling is also convenient for combining experimental data from different techniques into the same structural model, as these data tend to apply to different length scales and tend to have different spatial and time resolution. That way results of different techniques can be quantitatively tested for consistency and can be used as complementary restraints on structure and structural transitions.

## **Keeping model and experimental restraints separate**

Any structural model is based on primary experimental data and on restraints on distances and angles obtained from these data. Except in trivial cases, these restraints are never sufficient to specify an atomistic structure. Further assumptions are made. Some of them are well justified, such as canonical values for bond lengths and bond angles. Other assumptions, based for instance of energy minimization with a molecular force field, may be less secure. Some assumptions are made intuitively- and thus subjectively- by a human. They may not be formulated in a precise way, sometimes they may even be subconscious.

A structural model thus contains more information than the experimental data. The additional information is not precisely known and part of it may not be correct. It follows that a structural model is never complete, it may change when further information becomes accessible. Clearly, the updated model should be consistent with all available information. Such incremental improvement of models is possible only if the experimental restraints are stored together with the model. If the conversion of primary experimental data to restraints is non-trivial (also requires assumptions), the primary data need to be stored.

## **Creating models in MMM**

Currently, MMM is mainly intended as a toolbox for creating and testing structural models that are (mainly) based on distance constraints between spin labels. The following modeling features are implemented:

- generation of a model for an unknown structure in state B of a protein, if the structure in state A is known and distance constraints for state B have been determined [by anisotropic elastic network models](#) (*Section 3.7.4*)
- generation of a model for an unknown structure of a protein from a template structure by [constraint-supported homology modeling](#). (*Section 3.8*) The template can be another protein structure with sufficient sequence identity or similarity or a structure of the same protein in a different state.

- addition of a [coarse-grained lipid bilayer model](#) (*Section 3.13*) to a membrane protein structure with the possibility to fit thickness and orientation of the bilayer (based on membrane propensities for residues in  $\alpha$ -helices and  $\beta$ -strands).

## Ownership of a model

The concept of an incrementally improving model that takes into account experimental information from many different techniques is fundamentally at odds with the concept of a model "owned" by a certain number of authors that cannot be extended after publication. While MMM allows for defining and storing models that belong to a single group of authors, it encourages sharing of models and contributions by several groups, i.e., a continuous extension of the list of model authors.

Due to current social rituals in the science community, some extent of secrecy about primary information and current understanding of a biological system may be necessary. Hence, MMM allows to define annotation texts, restraints, and primary experimental data as private, accessible to a group, or public and to save models for exchange with only the appropriate extent of information.

## 1.5 Object concept

### What are objects in MMM?

Objects are parts of a structure, such as macromolecule chains, cofactor molecules, amino acid residues, or atoms. Thus, there is a hierarchy of objects: A chain consists of residues which in turn consist of atoms. Several objects on the same hierarchy level can be combined to an object on a higher hierarchy level. For instance, several residues can make up a secondary structure element, such as a helix. On a larger scale, several residues can make up a domain.

### Why is an object concept useful?

A hierarchy of structure objects allows to define structure and structural state transitions with the resolution and detail that is appropriate for understanding of function and is justified by the underlying experimental data (see also [Model concept](#)) (*Section 1.4*). Depending on available experimental data and flexibility of different parts of the structure, it may be appropriate to specify one domain of a peptide chain with fully atomistic detail (as a collection of atoms) and another domain, for instance a terminal loop, at lower resolution as a section of space accessible to this domain. A hierarchy of objects allows to add more detail when more information becomes available.

### What information is stored with an object?

An object always contains chemical structure information, for instance the sequence of residues for a peptide chain or the structural topology for a cofactor. If information

on a lower hierarchy level is available, the object also stores the addresses of all its child objects.

An object is related to the spatial structure by point coordinates, angles, and possibly a local frame. For a simple spherical object, such as an atom location, only the midpoint coordinates are specified. Objects at higher hierarchical levels may have more coordinates, for instance a residue can be specified by the C<sup>α</sup> coordinates, two backbone dihedrals, and an appropriate number of sidechain dihedrals. A cylinder model of a helix is specified by two endpoint coordinates. This coarse-graining allows for a reduction in the number of free parameters of the model to match the number of available experimental and (safe) internal restraints.

For visualization, an object is related to one or more graphics objects. The type and addresses of these graphics objects are also stored with the structure object.

Objects are related to function. For instance, a certain atom or residue of a peptide chain may be involved in substrate binding or a residue may be a phosphorylation site involved in regulation. Such information, which is often known from biochemical experiments, is stored in the form of clear text annotations. Such [annotations](#) (Section 1.9) can be associated with objects on any hierarchy level. Objects can also be assigned a keyword, such as "galactose binding site" or "trimerization" or "cancerogeneous mutation" and references which mention this particular object.

## What can be done with an object?

Objects can be [selected](#) (Section 1.8) and then be interrogated for information stored with them. They can be modified, as happens for mutation or spin-labeling of a residue. Furthermore, objects can be [visualized](#). (Section 1.7) They can be transformed in space, i.e., be rotated or translated with respect to the other objects. Distances, angles, and dihedral angles between objects can be displayed.

## How are objects identified?

Each object in MMM has a [systematic address](#). (Section 1.6) . This address is displayed when information on an object is output. It can be used for object selection via the [command line](#) (Section 2.1.14) or in [scripts](#), (Section 3.9) and also in Matlab source code. For instance, [2BHW](B)123.CA refers to the C<sup>α</sup> atom of residue 123 in chain B of structure 2BHW, while [2BHW](C) addresses the whole chain C in the same structure.

# 1.6 Addressing

## General address format

*[structure](chain){coordinate set}residue.atom:location*



Parts of the address may be missing, as specified in [Defaults](#).. A slightly modified format is used for [addressing secondary structure elements and domains](#). (Section 1.6.1)

- *structure*- structure identifier, either the number of the structure or the PDB identifier or a user-defined [synonym](#)
- *chain*- chain identifier, character (A-Z) or a user-defined [synonym](#)
- *coordinate set*- number of the coordinate set within an ensemble of structures (e.g. NMR structures)
- *residue*- residue number
- *atom*- atom identifier, usually up to two characters, with atom identifiers for common residues and cofactors being defined in PDB template files
- *location*- location identifier, up to two characters, both letters, only required if an atom has [alternate locations](#)

### Examples:

`[2BHW] (B) 123.CA`

addresses the C<sup>&alpha;</sup> atom of residue 123 of chain B in the structure with PDB identifier 2BHW (major plant light harvesting complex IIb)

`[2CUU] 131.N1:B`

addresses location B of the nitroxide N atom (*atom identifier*: N1) of residue 131 (R1A, a cysteine labeled by MTSL) in the structure with PDB identifier 2CUU (spin-labeled T4 lysozyme)

### Special addresses (shortcuts)

- The asterisk `*` addresses all currently selected objects.
- The exclamation mark `!` addresses the current structure, shown in the title of the main window.
- The `#` character addresses the current chain, shown in the title of the main window.

### Special addresses (surfaces, densities, motion arrows)

Surface objects, such as [densities](#) (Section 2.1.4) or [solvent accessible surfaces](#) (Section 2.1.3) and also sets of motion arrows have addresses that start with a dollar sign `$`, followed by a type identifier, and a tag that is unique for the given type (format: `$type:tag`), where type is `density` for density objects, `SAS` for solvent accessible surfaces, and `motion` for sets of motion arrows.

*Example:* `$SAS:SAS_1` addresses the solvent accessible surface with tag "SAS\_1".

Surface and motion arrow addresses cannot be used in all places where addresses of structure objects can be used. They are supported only by the [show](#), (Section 2.2), [hide](#), (Section 2.2), [color](#), (Section 2.2), [transparency](#), (Section 2.2), and [delete](#) (Section 2.2) commands.

## Defaults

- *structure*- defaults to the current structure, which is displayed in the title bar of the [main window](#) (Section 2.1)
- *chain*- defaults to the current chain, which is displayed in the title bar of the [main window](#) (Section 2.1)
- *coordinate set*- often there is only a single coordinate set and this part of the address can be left out, otherwise *all coordinate sets* are addressed simultaneously if this part is missing
- *residue*- should always be specified, use : (colon) to address all residues in a chain
- *atom*- if atom identifier is missing, the whole residue is addressed
- *location*- often there is only a single location and this part can be left out; otherwise *all alternative locations* are addressed if this part is missing

### Examples:

123

addresses residue 123 of the current chain in the current structure

(C) : .CA addresses all C<sup>&alpha</sup> atoms in chain C of the current structure

## Addressing several objects at once (list addressing)

Several structures, chains, coordinate sets, residues, atoms, or locations can be addressed at once by replacing identifiers with comma-separated lists.

### Examples:

[2BHW] (A,C) 96.CB

addresses the C<sup>&beta</sup> atoms of residues 96 in both chains A and C of structure 2BHW (but not in chain B of this structure)

[2BHW] (A,C) 96,123.CA,CB

addresses the C<sup>&alpha</sup> and C<sup>&beta</sup> atoms of residues 96 and 123 in chains A and C of structure 2BHW

Such list addressing accepts that certain combinations may not exist. For instance, if residue 123 were a glycine, its C<sup>&alpha</sup> atom would be addressed and no error would be returned although there is no C<sup>&beta</sup> atom. A continuous range of residues can be addressed by using a hyphen.

### Example:

[2BHW] (A) 96-123.CA

addresses the C<sup>&alpha</sup> atoms of residues 96-123 in chain A of structure 2BHW



Note that addressing a large number of objects via a combination of list tags can lead to noticeable computation times for address resolution.

## Addressing residues and atoms by type

Residues and cofactors can be addressed by their PDB residue types, enclosed in double quotation marks. This can be combined with list addressing by a comma-separated list. MMM accepts common nomenclature "Xaa" for any natural amino acid residue (thus excluding modified amino acids and cofactors), "Asx" to abbreviate "Asn", "Asp" and "Glx" to abbreviate "Gln", "Glu".

### Examples:

```
[2CUU] "R1A".N1,O1
```

addresses the nitroxide group nitrogen and oxygen atoms of all residues R1A (cysteine labeled by MTSL) in the structure with PDB identifier 2CUU.

```
[2BHW] "CHL", "CLA"
```

addresses all chlorophyll a and b molecules in all chains of the structure with PDB identifier [2BHW].

Atoms can be addressed by element.

### Examples

```
."Fe"
```

addresses all iron atoms in the current chain of the current structure.

## Working with alternate locations

Alternate locations are often specified for only a few atoms in a structure, chain, or residue. To address all atoms (using the single location of those atoms that do not have alternate locations), use the address format given above.

### Example:

```
[2CUU] "R1A".B
```

addresses all residues with type R1A in structure 2CUU, selecting location B for all those atoms which do have alternate locations.

If you want to inspect only atoms with alternate locations, use an exclamation mark ! to enforce location addressing.

### Example:

```
[2CUU] "R1A", "ASN".:B!
```

addresses only atoms with alternate locations, and of those only location B, for residues of types R1A and ASN in structure 2CUU

## Synonyms for structure and chain identifiers

For convenience, MMM allows to define synonyms for structure and chain identifiers. For instance, after defining the synonym LHCIIb for structure 2BHW using a [command](#), (Section 2.2) `[LHCIIb] (A) 123.CA` addresses the C<sup>&alpha;</sup> atom of residue 123 of chain A in structure 2BHW.

Usually, such synonyms will be the established names of the proteins or protein complexes, however, the user is completely free in the choice of synonym.

## 1.6.1 Substructure addressing

### Preface

While [general addressing](#) (Section 1.6) is fully defined by the chemical structure of a macromolecule or macromolecular complex, addressing of substructures (secondary structure elements and domains) depends on definitions of these substructures. Such definitions are usually subjective. Hence, general addressing is robust: it always returns the same objects in the same structure. In contrast, assignment of substructures may change, i.e., different scientists may have different opinions about which residues belong to a certain helix and even whether this is a continuous helix or a helix-loop-helix domain. Hence, substructure addressing is not robust.

As far as possible analysis and manipulation of a model should thus be performed by general addressing. However, multiscale modeling in general and certain types of coarse graining in particular require definitions of substructures. It is strongly recommended, although not mandatory, to collect all substructure definitions in an [initialization script](#) and to renounce interactive reassignments of substructures. To protect the user from inadvertent redefinition of substructures, assignments of secondary structure and domains are locked and the lock needs to be explicitly removed before redefinition.

### Secondary structure addressing

Internally MMM treats secondary structure elements as a special type of a domain. A domain is a set of residues. Different from other domains, secondary structure elements are defined by default when reading a PDB file and allow for internal addressing of residues. Address format:

*[structure](chain){coordinate set}<type.tag>relative residue number.atom:location*

Parts of the address may be missing, as explained for [general addressing](#).

- *type*- type of the secondary structure element, **H** for helix, **E** for strand (part of a sheet), **L** for loop

- *tag*- identifier for the secondary structure element. For default definitions this is a number, any tag can be defined by the user

### Remarks:

Currently MMM does not distinguish between different types of helices and strands. However, all types defined in the dictionary of secondary structure of proteins (DSSP, W. Kabsch, C. Sander, **1983**, *Biopolymers*, 22: 2577-2637) are reserved.

If an atomistic structure exists, there is often an established nomenclature of helices, strands/sheets and loops, given in the paper that explains this structure. Information on this nomenclature and proper assignment of secondary structure is usually missing in the PDB file. It is strongly recommended to implement these established definitions in an [initialization script](#) that is read by MMM whenever this PDB structure is loaded.

Residue numbers are relative to the first residue in the secondary structure element. Be aware that this may cause problems if the length of a secondary structure element is redefined.

### Examples:

```
[2BHW] (A) <H.2>3.CA
```

addresses the C<sup>&alpha;</sup> atom of the third residue in the second helix of chain A of structure 2BHW. This works immediately after reading the PDB file.

```
[2BHW] (A) <H.B>3.CA
```

the same as above, but assuming now that a nomenclature of helices was established where the second helix is termed helix B. This works only if an initialization script was read or if helix B was defined by a command.

## Domain addressing

A domain is defined as a set of residues that are functionally related or can move as a whole with respect to other parts of a macromolecule or macromolecular complex. The definition is done by a [command](#) (Section 2.2) or a series of commands in an [initialization script](#). Such commands can also be performed during interactive work with MMM, but this is strongly discouraged. Domains do not have a substructure, i.e., they can only be addressed as a whole.

### Address format:

```
[structure]|domain|
```

where *domain* is the identifier (tag) of a domain that was defined previously.

### Example:

[1ZCD]|translocation|

addresses the substrate translocation domain in the structure with PDB identifier 1ZCD.

## 1.7 Visualization


### What can MMM visualize?

MMM visualizes proteins and cofactors in common display formats, such as ribbon models, stick models, space-filling models or wire models. In addition, [density cubes](#) (Section 1.7.1) and bilayers can be visualized. For [spin labels](#) (Section 3.1) a special visualization mode displays only the midpoints of N-O bonds as spheres with the sphere radius encoding population of spin label conformations (rotamers). Possible [localizations of spin labels](#) (Section 3.3) with respect to a reference structure or with respect to each other ([networks](#)) (Section 3.4) can also be visualized. During development of MMM, further visualization modes will be added.

### Default visualization

By default, MMM displays a ribbon model of a protein with color according to secondary structure and does not display cofactors after loading a PDB file. When opening a model that was stored in MMM format, the previous appearance of the model is restored.

### Interactive visualization

Interactive visualization requires that objects are selected by one of the standard [selection methods](#). (Section 1.8) . The display mode can then be selected via the  [Graphics mode toolbutton](#) (Section 2.1.11) or the [Display/3D display mode](#) (Section 2.1.4) menu item.

### Visualization via the command line

Objects can be [addressed](#) (Section 1.6) and visualized at once via the [show command](#). (Section 2.2)

**Visualizing binding sites** Binding sites can be selected and visualized using the [sites window](#). (Section 3.11)

### Colors and color schemes

MMM allows for assigning colors to each individual object or to several selected objects at once by color name, from a palette, or by direct input of RGB (red/green/blue) color codes.


Color assignment to [selected objects](#) (Section 1.8) is possible via the  [Color toolbar](#) (Section 2.1.11) or the [Display/Color](#) (Section 2.1.4) menu item.

In the same way color schemes can be assigned, such as color by chain, by sequence, by B factor, or by charge.

Without previous selection, color can be assigned by the the [color command](#) (Section 2.2) or a color scheme by the the [colorscheme command](#). (Section 2.2)


## Transparency

For each individual object or several selected objects a transparency between 0 (fully transparent, invisible) and 1 (fully opaque) can be selected. Transparency values smaller than zero correspond to default transparency, which is opaque, unless alternate conformations are specified. In the latter case the transparency value corresponds to the population of the conformation.

Transparency assignment to [selected objects](#) (Section 1.8) is possible via the  [Transparency toolbar](#) (Section 2.1.11) or the [Display/Transparency](#) (Section 2.1.4) menu item.

Without previous selection, transparency can be assigned by the the [transparency command](#). (Section 2.2)

## View selection

MMM can automatically zoom in on selected objects. Use the [zoom in](#) (Section 2.2) command or the  [View selected toolbar](#) (Section 2.1.11) or the [Display/View selection](#) (Section 2.1.4) menu item.

## Depth cueing

A better 3D impression can be obtained with [emulated depth cueing](#). (Section 2.1.13) Note that this does not mix well with object transparency.

## Switching off atom graphics

To speed up the user interface and interactive rotation, zoom, and pan of 3D graphics, atom graphics can be temporarily [reduced or switched off](#). (Section 2.1.12)

## 1.7.1 Density cubes

### What are density cubes?

Density cubes visualize a spatial distribution of an object, for instance of the N-O midpoints of a spin label. They can be derived from experiment, MD simulations, or computations within MMM. Density cubes can be visualized at different intensity levels, whereas absolute levels are normalised to maximum density and relative levels to the total integral of the density. For instance, one can visualize the space where the N-O midpoint of a spin label is with a probability of 90%.

### Loading density cubes

See menu [File/Load density](#). (Section 2.1.1)

### Displaying density cubes

See menu [File/Display density](#). (Section 2.1.4)

## 1.8 Selection

### What is selection good for?

Interactive manipulation and [visualization](#) (Section 1.7) of a structural model requires that functions are applied to only part of the structure. This part of the structure is defined by the selected [objects](#), (Section 1.5), such as selected chains, residues, or atoms. For instance, you may want to change color of only a few residues involved in substrate binding of an enzyme. You can explicitly select the objects as explained below or implicitly select them by an [address](#) (Section 1.6) as a [command](#) (Section 2.2) argument, in this case the first argument of the `color` command.

At any time no object can be explicitly selected, a single object can be selected, or a collection of objects at the same or at different hierarchy levels can be selected.

### Storing a selection

You might want to store complex selections for later use. This can be done by defining the selection as a domain, using either the [domain \\* domain\\_name](#) (Section 2.2) command or the [Edit/Domain...](#) (Section 2.1.2) menu item.

### Interactive selection

#### Hierarchy window

The [hierarchy window](#) (Section 2.3) provides an overview of all structures in the model from structure model, via chains, different coordinate sets for chains in NMR models, residues

and cofactors, atoms, down to atom locations. On each level objects can be picked and selected by mouse clicks. This is the best way for systematic selection if you do not know the structure yet, as it combines inspection and selection.

## Model window or model panel

In the 3D [model window](#) (Section 2.1.16) or [model panel](#) (Section 2.1.15) of the main window, objects can be [selected by clicking](#). (Section 2.1.16) This is most convenient if you want to know what is close to an object.

## Selection by command

Objects can also be selected via the [select command](#). (Section 2.2) . The address format of MMM is rather powerful and allows to build complex selections quite fast. This is the most convenient selection mode if you already know your structure and if you are familiar with MMM's [address system](#). (Section 1.6)


## Selection by keyword

If the model is annotated, objects may be assigned keywords. You can then easily display and select all or part of the objects associated with a certain keyword via the [Edit/Find by keyword](#) (Section 2.1.2) menu item. See also description of the [find by keyword window](#). (Section 1.11)

# 1.9 Annotation

## What can you do with the annotation window?

An MMM model contains more information than just structure data. MMM can be used like a laboratory notebook for exploring structure/function relationships. Observations made during work with the model and information retrieved from papers can be stored where they belong- with the chains, residues, or atoms concerned. This information can be retrieved by selecting this object or by [keyword search](#). (Section 2.1.2) . [References](#) (Section 1.10) can be stored together with their fulltext links.

This window is accessible only if a single object is selected (you can see that in the [message board](#) (Section 2.1.17) and by availability of the corresponding [tool button](#) (Section 2.1.11) ).

## Pages

A default annotation with two pages exists for any object. User annotations start at page 3, you can use as many pages as you wish. The current **Page** is displayed in the left upper corner of the window, it is not possible to directly input the page number. Instead, navigate with the < and > buttons.

## Default pages

Page 1 gives the [MMM address](#) (*Section 1.6*) of the selected object with a detailed description of what it means. The type of residue (three-character PDB code, which is the three-letter code for amino acids), the mass and the mean coordinates are also displayed. If the object is a whole structure, header information from the PDB file is also displayed and the number of atoms and residues is listed.

If [DSSP](#) (*Section 5.2*) secondary structure information is available, the first page of an residue lists the DSSP secondary structure assignment, the predicted water accessibility (ignoring water access blocking by cofactors), and the backbone dihedrals.

Page 2 contains a list of objects that are close to the selected objects (structural context). The distance to these objects (between mean coordinates) is also shown. A default context radius is used to generate this list, which is 3.5 Å for atoms, 8 Å for residues, and 40 Å for chains. Note that context with a non-default radius can be displayed in the message board via the corresponding [tool button](#) (*Section 2.1.11*)



If [DSSP](#) (*Section 5.2*) secondary structure information is available, the second page of an residue also lists hydrogen bonding partners and hydrogen bond energies.

Additional automatic annotation pages may exist, if there was special information in the PDB file or if a task performed in MMM has annotated the object.

## Privacy

Non-default annotations (pages 3 and higher) can be assigned privacy levels. That way you can share your model file without sharing all the (possibly insecure or still confidential) information you have stored with it. All information tagged **Public** is always stored, all information tagged **group** is stored only if you save your model for private use or exchange within your group, and all information tagged **user** is only stored when you save the model for private use.

Note that MMM saves by default in private mode, so that information is not unintentionally lost.

[Keywords](#) do not have individual privacy modes. It is the policy of MMM to save all keywords that are referred to by any object with a public annotation. If the model is saved for sharing with the group, additionally all keywords are saved that are referred to by any object with an annotation visible to the group. The same policy applies to references. In particular, references and keywords that are not used by any object are not stored when a file is saved for sharing with the group or public, while they are stored if the project is saved in the normal way **Save** and **Save as...** items of the file menu.



## Keywords

The **Keyword list** displays all keywords that are currently defined *in your model* in alphabetical order. Note that all keywords starting with capital letters are found before all keywords starting with small letters. You can define a new keyword using the **New** button for the **Keywords** field. Defining a new keyword automatically adds it to the **Keywords** field of the current object.

To add or remove a keyword from the **Keywords** field *of the current object*, select it in the **Keyword list** and click the + or - button.

Keywords are assigned to the object, not to an individual annotation page. If you want to remember to which annotation the keyword belongs, mention it in the **Annotation text** edit field.

## Automatic annotation

The following annotations are automatically generated by MMM:

- binding sites from SITE and REMARK 800 records in PDB files
- metal coordination information from REMARK 620 in PDB files
- missing atoms information from REMARK 470 in PDB files
- information on spin-labeled residues by the spin labeling module

Automatic annotations start with an asterisk \* in the first line immediately followed by a keyword. The first line should not be edited, as otherwise MMM cannot amend the annotation if further annotations of the same type are made (they will then end up on extra pages).

## References

The **Reference list** displays all [references](#) (*Section 1.10*) that are currently defined *in your model* in alphabetical order of their short names. Note that all short names starting with capital letters are listed before all short names starting with small letters. You can add new references to the model by using the **New** button for the **References** field. Adding references automatically adds the current citation in the [Reference window](#) (*Section 1.10*) to the **References** field of the current object.


To add or remove a citation from the **References** field *of the current object*, select it in the **Reference list** and click the + or - button.

References are assigned to the object, not to an individual annotation page. If you want to remember to which annotation the reference belongs, mention it in the **Annotation text** edit field.

## 1.10 References

### What can you do with the reference window?

The reference window of MMM is a basic, self-contained reference manager with PubMed search function, web access to Abstracts and full texts, a range of import and export options and the possibility of weakly automatic updates of bibliographies by PubMed searches. You could actually generate MMM projects (models) that contain only a bibliography and use MMM purely as a reference manager. You can also use it in conjunction with EndNote or BibTeX.

Furthermore, MMM models with actual protein structures may have references to papers, books, and webpages associated with [annotations](#) (Section 1.9) of an [object](#), such as a structure, a peptide chain, a residue, or even an atom. For that reason the reference window can be either accessed via the **New** and **Edit** buttons in the **References** part of the [annotation window](#) (Section 1.9) or via the [Edit/References](#) (Section 2.1.2) menu item or the  [References](#) (Section 2.1.11) toolbar button. If the reference window is accessed from the annotation window, it is modal, i.e., no other MMM (or Matlab) window can be accessed before closing it again. To manage references or look at abstracts and full text, better access the window via the toolbar button or menu.

The reference window allows to input references by [PubMed search](#), by import from [EndNote](#), [BibTeX](#), [ISI Web of Knowledge](#), [SciFinder](#), [Google Scholar](#) or by [direct manual input](#). Existing references can be edited.

References can be [exported in BibTex format](#) and in a format that can be read by [EndNote](#). They can also be [saved or copied to the clipboard](#) in citation formats for several journals.

The **Search** button also allows to [search for a reference within the existing bibliography](#). (Section 1.10.1)

Structures imported from the Protein Data Bank (PDB) often contain already one or more references from the JRNL and REMARK 1 records. Bibliographic information is incomplete in the PDB, hence these references are automatically "repaired" via PubMed if the computer is online during PDB reading and if the PDB file lists the PubMed ID of the reference.

### ### Warning ###

The reference window is automatically closed if a new model is loaded or graphics is redisplayed. You may lose edited and imported references. Please complete your work in the reference window before you do other things.

## Adding a bibliography to a model

An MMM model contains at least one structure, possibly several structures. You may have a literature list for the proteins whose structures these are or you may assemble it via search engines.

### Direct search

MMM can directly [search PubMed](#) (Section 1.10.1) and import the results. By setting up [automatic searches](#) (Section 1.10.1) the references of a model can be always kept up to date. For this you have to specify the concepts that are most relevant for this model or authors who publish often on similar structures.

### Import

MMM supports several [Import/Export formats](#), which can be selected via the corresponding popup menu at the bottom left of the reference window. These are:

- [MEDLINE](#), a tagged text format that can be exported by PubMed
- [SciFinder tagged format](#), a text export format of CAS SciFinder
- [ISI CGI](#), the export format of ISI Web of Knowledge intended for EndNote, RefMan, and ProCite
- [EndNote text export](#), a tagged clear text format that can be exported from EndNote
- [BibTeX](#), a LaTeX bibliography format that can be exported by Google Scholar
- [MMM](#), the own bibliography format of MMM, which preserves all information when transferring references between MMM projects

After selecting the format, clicking on the [Import](#) button opens a file dialog for selection of the bibliography files. Several bibliography files, even in different formats can be imported after each other. With MEDLINE (PubMed) import, MMM checks whether the new references are unique, i.e., whether a reference with the same PubMed ID already exists. In that case, it is updated rather than added, the old short name is kept and an existing DOI is kept, if PubMed does not provide a DOI. Note that this does not work if the same reference exists, but was first imported without PubMed ID (or was directly input).

### Direct input

You can also directly type your reference into the provided mask. For that, you first select the [Reference type](#) in the corresponding popup menus (top right). Defined reference types are:

- Article
- Book
- Book section
- Webpage
- Conference
- Dissertation
- Preprint

- Patent

Depending on reference type, certain fields of the input mask may change their title. It is not required to fill all fields of the mask. To input the next reference, use the **>** button of the **Reference number** control. The current reference is temporarily stored.

## Editing short names

References are most easily referred to by mnemonic short names, such as *Einstein:1905* for a paper written by Albert Einstein in 1905. If a bibliography is imported in BibTeX format, such short names are usually already defined (for instance, Google™ Scholar export to BibTeX generates them automatically). For [PubMed searches](#) (Section 1.10.1) and MEDLINE import the mnemonic short names are appended with an underscore and the unique PubMed identifier (PMID) to make sure that the short names are unique (example: *Puskin:1975\_162831*). You can remove the PMID and underscore if you wish. MMM will check whether the short name is still unique. Otherwise, append characters, such as in *Puskin:1975b*.

For import other than from BibTeX and MEDLINE you can replace the default short names of MMM (reference numbers in square brackets) by your own mnemonic names. Edit the **Short name** field for that. MMM will check whether a given short name already exists. In that case a message box pops up and the old short name is retained.

## Input an URL

A Uniform Resource Locator (URL) can be input for any reference type by changing the type to Webpage, editing the **URL** field, and changing back. The URL will then still be displayed instead of editors. Note however that for edited books and sections of edited books this makes the **Editors** field inaccessible.

## Finishing bibliography input

After finishing your input, click the **OK** button to store all new references with the model. Clicking the **Cancel** button will cancel *all edits* done after the reference window was opened (there is a query whether you really want to do that). The reference window will close and the reference list will be updated in the [annotation window](#). (Section 1.9)

## Recommendations on search engines

### PubMed

This is for free, easy to work with, and usually sufficient for work on protein structures (see also [Search PubMed](#)) (Section 1.10.1). However, it does cover less literature than CAS SciFinder, ISI Web of Knowledge, or Google Scholar. As of October 27th 2009, PubMed provided more Digital Object Identifiers than any other search engine. This is the preferred way for [fulltext access](#). Due to the unique

PubMed identifiers, MMM can also check whether a reference does already exist. Unless you are searching for references that are only remotely related to a biological system or want to include dissertations, PubMed is the engine of choice.

## **CAS SciFinder**

If you have access to this expensive search machine, it will usually give the most complete answers, as the database is based on a well thought keyword system and keyword assignment is not left to authors and journals. SciFinder searches may turn out some references twice (once with proper publication year and once with the publication year in the publication date field). Use the remove duplicates link above the reference list to avoid this. Unlike all the other search engines, SciFinder also lists dissertations. It also lists a few books, but these ones with lousy bibliographic information (no publisher, no ISBN).

## **ISI Web of Knowledge**

This is not for free either, but it is a choice that may provide more references than PubMed. You will miss references if the keywords are not in the title or abstract or journal keyword field. ISI Web of Knowledge provides Digital Object Identifiers (DOIs) for newer literature.

## **Google Scholar**

This is also for free and has good coverage of literature. It is not so convenient to work with, in particular because each reference has to be saved individually and references are always sorted by number of citations. Unlike ISI Web of Knowledge and PubMed it also finds books.

## **Editing references**

### **Browse references**

You can browse references by using the < and <>> buttons of the `Reference number` field. To go to the last reference, input a very large number into the `Reference number` edit field. Clicking the > button again at the last reference inserts a new reference with empty fields.

### **Accessing a reference by its short name**

Use the ? button above the `Short name` field.

### **Deleting a reference**

Use the `Delete` button near the `Reference number` field. Note that references that are linked to annotations (objects) cannot be deleted (MMM checks this).

## Editing bibliographic information

Note that MMM offers to automatically update bibliographic information when you access the [abstract](#) via PubMed. Hence, edited information may be overwritten. Exceptions are the short name, which is never updated on reimport and the DOI, which is updated only if the PubMed record really provides one (an existing DOI is not erased). If you want to edit other fields and keep your own information, you should not accept the update.

For journal articles, it is more advisable to rely on the automatic updates of bibliographic information. For books, book sections, dissertations, websites, and conference proceedings bibliographic information in common data bases is usually lousy, and you *do* want to edit it (abstracts are rarely accessible from PubMed anyway for these reference types).

## Abstract display and updating of bibliographic information


Although MMM does not store abstracts, it can download them from the internet if the PubMed identifier (PMID) is supplied with the reference (default for MEDLINE import or literature search from within MMM) or if enough information is available to find the reference in PubMed. If no PMID is available, MMM uses nmaes of all authors with at most one initial, the journal name, and the year in the initial search. The abstract of the first (newest) matching reference is displayed. If the initial search turns out several matches, MMM searches for the reference with matching volume and first page. A warning is shown that the displayed abstract is not the abstract of the matching reference. In this case, accept the matching reference for update and click the [Abstract](#) button once again to see the matching abstract.

The abstract is displayed in a browser window, author address and keywords are also displayed, if they are available. The publication language is shown and, for reviews, the number of references is given. Note that abstract display requires internet access. If the connection is broken, an error message is displayed in the message board of the main window.


If the reference (or a likely candidate) could be localized, MMM asks whether this reference should be accepted for update of the bibliographical information. Accept it unless you have very good reasons not to.

If the reference cannot be localized, the journal may just not be covered by PubMed (at least not the volume you are looking for) or it may be wrong (this happens more often than you would think). Depending on the information you have, you may try to localize the reference with the [PubMed search](#) (Section 1.10.1) of MMM or another literature database.

## Fulltext access

References imported from ISI Web of Knowledge or PubMed may already have a Digital Object Identifier that you can see in the [DOI](#) field. By clicking the globe button  the corresponding web page is loaded into a Matlab browser. Note that the

reference window is modal if it was accessed from the annotation window. That means that the Matlab browser can be accessed only after these two windows have been closed (sorry, we have reasons for that). However, you can access the full text without closing the reference window, when you opened it via the toolbutton or **Edit** menu.

References imported from CAS SciFinder or Webpages may have a Uniform Resource Locator (URL) associated with them. This will be displayed in an URL field. By clicking the globe button  the corresponding web page is loaded into a Matlab browser. See above for accessibility of the browser window.

If the DOI is not available, MMM tries to locate the fulltext via SFX. Please check before, whether PubMed can retrieve the DOI (use the **Abstract** button), as even the two-step process Abstract/DOI is faster and often more convenient than SFX. For SFX access, the library of your institution needs to have an SFX server and MMM needs to know its internet address. You can find this address when you access SFX from another literature data base or you can ask your librarian. This address needs to be input in the file `initialize_MMM.m` which is found in the MMM main directory, e.g. `web_adr.SFX='sfx.ethz.ch:9003/sfx_locator';`. A later version of MMM will have a preferences window to make this more convenient.

Occasionally SFX cannot locate the full text, although the bibliographical information is correct and your institution does have fulltext access to this journal and time period. The SFX server will then simply claim that there is no full text available. In such cases you may want to access the homepage of the journal and click through its archives (the hard way). Journal homepage also often help if bibliographic information is incomplete or incorrect (many have their own search engine).

Note also that fulltext access for paid journals will work only if you are in the web domain of an institution with a subscription. Free journals can be accessed from everywhere, even via SFX with the ETH Zurich server (courtesy of the IDS Swiss German Library Network).

## Exporting references for MMM

### PubMed

Perform a search and select the desired references by activating the corresponding checkboxes (left of the reference). Now click the **Send to:** link near the right top of the **Results:** panel and activate the **File** radiobutton in the **Choose Destination** panel. A new panel appears, where you select in the **Format** popup menu **MEDLINE** and where you can select how to sort the references. Finally click the **Create File** button and save the file where you want.

(tested on Oct 27th 2009 using [the preview of new PubMed](#))



## EndNote

Open the corresponding EndNote library and select the references that you want to export to MMM, for instance, by clicking on them while holding the `Ctrl` key. Go to menu `File/Export...` and select in the `Save as type:` popup menu `Text file (*.txt)`. Select in the `Output format:` popup menu `Show All Fields`. Make sure that the `Export Selected References` checkbox is activated, unless you want to export the whole library. Edit the `File name:`, navigate to the desired directory and click on the `Save` button.

(tested with EndNote X2)

## ISI Web of Knowledge

Perform a search in the desired database, usually you should select `All databases`. Select the references that you want to export to MMM by activating the corresponding checkboxes (left from the reference). If all your references are on the same page, click the `Save to EndNote, RefMan, ProCite` button above the reference list. A dialog window appears that allows to choose directory and file name.

If you want to select references on several pages, click the `Add to Marked List` button above the reference list *before you change page* (otherwise your selections are lost). Select throughout all pages and finally click the `Marked List` link near the top of the page. On the next page, click the `Save to EndNote, RefMan, ProCite` button. A dialog window appears that allows to choose directory and file name.

(tested on Oct 27th 2009)

## SciFinder (CAS)

Perform a search and select the desired references by activating the corresponding checkboxes (left of the reference). Beware of the `Select All` link, which almost inevitably select some papers twice. Now click on the `Export` link near the right top of the `References` panel. A window appears, where you make sure that under `Export:` the `Only selected answers` radio button is activated. Input the `File Name:` and select from the `File type` popup menu `Tagged format (*.txt)`. Now click the `Export` button and save the file where you want.

(tested on Oct 27th 2009)

## Google Scholar

Setting BibTeX output format

This step is required only once on a given computer and browser. It requires that your browser accepts cookies. If you delete cookies later you have to repeat this step.



Go to [Google Scholar](#) and click the [Scholar Preferences](#) link right from the [Search](#) button. In the [Bibliography manager](#) field (near the bottom of the page) activate the [Show links to import citations into](#) radiobutton and select from the popupmenu on the right [BibTeX](#). Now click the [Save Preferences](#) button (top right of the page).

## Exporting references

Perform a search. If your preferences are set correctly (see above), each citation has a pale [Import into BibTeX](#) linke beneath. To export this reference, right-click the link and select [Save link as...](#) from the context menu that appears.

References can be exported only one by one, issue information is missing, and no fulltext access information (DOI) is exported. The issue information can be obtained by exporting from Google Scholar to EndNote and then [exporting from EndNote](#).

(tested on Oct 27th 2009)

## Exporting references from MMM to other software

MMM can export in ISI .cgi format, for which [EndNote](#) has an import filter, in [BibTeX](#) format and in a MEDLINE-like format. The MEDLINE-like format is supported because it preserves PubMed identifiers and can be used to exchange reference lists between MMM models. Export in SciFinder tagged format or EndNote text export format is not possible.

## EndNote

Select [ISI CGI](#) in the [Import/Export format](#) popup menu and click the [Export](#) button. A file dialog appears for selecting directory and file name. The Message board in the main window reports that the file was saved and gives the file name. The following information is stored:

PT (document type)

- [J](#) for articles and reviews
- [B](#) for books
- [S](#) for book sections
- [W](#) for webpages (non-standard)
- [C](#) for conferences (non-standard)
- [D](#) for dissertations (non-standard)
- [P](#) for patents

Apart from that the title (TI), author list (AU), document type (DT, Article or Review), journal (JI), volume (VL), issue (IS), first (BP) and last (EP) page and the publication year (PY) and, if available, the DOI (DI) are stored. For patents, the patent number (PN) is also stored.

To import the references into EndNote, use the [File/Import...](#) menu item and select [ISI-CE](#) as the import option with [No translation](#). For dissertations, the university information

will be found in the Notes field of the EndNote record. For patents, the country has to be inferred from the first two characters of the patent number.

(tested with EndNote X2)

## BibTeX

BibTeX export stores the short name and supports document types `@article`, `@book`, `@inbook` (book section), `@conference`, and `@phdthesis` (dissertation). The `school` field of the `@phdthesis` record is taken from the `Corporate Source` field in SciFinder records or has to be provided in the `Publisher` field of MMM for other input modes.

Webpages, preprints, and patents are stored as `@misc` records. Generally, `author`, `title`, `journal`, `year`, `volume`, `issue`, and `pages` fields are written. In addition, `publisher`, `location`, `book title`, `chapter`, `editors`, and `URL` fields are provided, if such information is present in the original records.

Note that results may depend on the formatting of imported references. Formatting was optimized for records obtained from PubMed. In particular, references imported from PDB files will give poor results if no PubMed ID is given in the PDB file. MMM stores BibTeX references with initials first, surnames last without commas, but accepts the version with surname first, comma, initials on import. Mixtures of full first names and initials with full stops in the same name will confuse the import filter.

For patents, you have to define your own output formatting. The patent country (code, such as `WO` for world, `EP` for European patent, `JP` for Japan) is provided in the field `country` and the number in the field `number` of the `@misc` style.

(tested with TeXnic Center and the citation style of Bioinformatics as well as `apaplus`).

## Copying and saving in citation (text) formats

The bottom panel `Save or copy to clipboard in citation format` can be used to print a nicely formatted list of the bibliography or to copy individual citations to a text processing program. The panel contains a listbox for selecting the journal (citation format), a popupmenu for selecting the save or copy format, checkboxes for selecting whether the number or short name should be included, and pushbuttons to `Copy` to the clipboard or to save the current reference or all references in the specified citation format. The bottom line in the panel shows a preview of the citation format for a fake test reference.

### Selecting the journal (citation format)

Citation formats of a few journals are preprogrammed. They are given in a listbox in alphabetical order of the abbreviated journal name. You can either scroll through the whole list or preselect by the first letter or first two letters. To preselect, first click some journal name in the list. Now type the first letter of the wanted journal on the

keyboard. The listbox will jump to the first journal that starts with this letter (if there is one). Now you can either scroll down or type the second letter while holding down the shift button. The listbox will jump to the first journal whose abbreviated name starts with the two letters you typed (if there is one).

*Example:*

Click anywhere in the listbox and type `j <Shift> p`. Now `J Phys Chem` will be selected.

If your journal (or a matching format) cannot be found and if you dare, you can [define a citation format](#) (Section 1.10.2) yourself. Otherwise [ask us](#) for implementing the required format.

## Selecting output format

MMM can generate the citation in `LaTeX`, including italic and bold formatting of parts of the citation, as a `bibitem` for `LaTeX`, which will include the short name as a label for the reference, and as plain text (italic and bold formatting are lost). These three formats are available both with the clipboard and for saving the citation(s) to a file. The Rich Text Format `RTF` is available only for saving, as Matlab's clipboard function can export only strings. Files saved in `RTF` can be imported by Microsoft Word, OpenOffice, and indeed most text processor programs. Italic and bold formatting are preserved. The `HTML` format is used in web pages. Italic and bold formatting are preserved.

## Including reference numbers or short names

The citation can be saved plain, without any identifier (unselect both checkboxes), with the reference number in brackets, or with the short name in brackets. The last two modes are useful if you want a printout of the whole bibliography of a model.

## 1.10.1 Search PubMed

### Two functions of this window

The search window can be used for [PubMed](#) literature search and for [searching a reference in an existing bibliography](#). Neither of these searches is case sensitive, i.e., it does not matter whether words are written in lower case or upper case. A search for `j magn reson` will also find `J Magn Reson` and `J MAGN RESON`.

### What is PubMed?

PubMed is a freely accessible, powerful database search engine sponsored by the National Institute of Health (NIH) of the USA and maintained by the [National Center for Biotechnology Information \(NCBI\)](#). For work on biomacromolecular models, the literature database of PubMed, MEDLINE, should be generally sufficient to retrieve relevant references. See the [journal list](#) for coverage. To find dissertations and book

sections, you may want to use [CAS SciFinder](#), (Section 1.10) , if you have access to it and to find references in more obscure journals or see how often papers are cited, you may want to use [ISI Web of Science](#). (Section 1.10) The latter two search engines are not for free, but your institution may have a license.

PubMed is distinguished by an easy and flexible internet access, by unique identifiers for references, and by good support of digital object identifiers (DOIs) for fulltext access. Furthermore, PubMed maintains a systematic classification (keyword) system for references. Although searches for combinations of several concepts may be easier and more successful with CAS SciFinder, PubMed is often the search engine of choice.

## What can be searched?

### Concepts

Concepts may include the names of proteins, their functions, chemical modifications, and general concepts, such as "structure" or "crystallization" or "NMR". Different concepts are separated by a semi-colon. Hence, the query `LHCII structure` will yield less results than the query `LHCII; structure`. In the former case, "LHCII structure" is a single concept and the two words must appear in this sequence in the record. In the latter case, the two words can separately occur anywhere in the record.

Concepts can be searched in `all fields` of a record, which is the default, or only in `title & abstract`, only in the `keywords` or only in `major topics`. Consider the last two options if your original search returns too many references.

*Example:*

```
light-harvesting; state transition; phosphorylation;
```

### Authors

Authors can be searched by giving only their surname, the surname with up to two initials (space between surname and initials, no space between two initials) or the surname and the first name. The option with surname and first name is discouraged, as it works only for newer records and does not work with all journals.

Names of several authors are separated by semi-colons. Commas and full stops are ignored.

The search can be restricted to the `first author` or `last author`. The latter version may help to restrict results to the main field of interest of senior authors who are involved in many collaborations.

*Example:* `Perutz MF; Kendrew J`

## Years

Restricts the search to a single year or a period of a few years. Use this if you search for the older references on a topic where very many papers were published recently. PubMed will generally return the newest references first and because of the restriction in the **Maximum number** of records you may not be able to access the older references otherwise. If you input in the format **1989-** the input is automatically amended with the current year for the end of the range, e.g. **1989-2009**. If you input in the format **-1989**, the input is automatically amended to a period that should include the earliest references in PubMed, **1900-1989**.

In [automatic searches](#), the **Years** field is ignored and only new references added to PubMed between the previous search and the automatic search are considered.

*Example:*

in Concepts: **LHCII**

in Years: **1945-1985**

## Reviews

**Activate the **reviews only** checkbox to search for only reviews. Maximum number**

You are unlikely to closely examine more than 50 references, indeed you may find that your patience wears out earlier. Use the **Maximum number** popup menu to restrict yourself to 10 or 20 results.

MMM will report in the Message board of the main window how many references were really found in the search. If the maximum number was indeed found, you have received only the most recent papers (and the earlier ones tend to be more important!). Try to narrow down your search.

## Journal

This is useful mainly if you want to import a specific reference that you already know. Use either the abbreviated form or full name of the journal. Full stops for the abbreviated form are not required (and are ignored, if present).

*Examples:*

combine Concept **protein** with option **reviews only** with:

**J Chem Biol**

or

**Angewandte Chemie**

## Keyword list

If the model contains at least one structure that was downloaded from the protein data bank (PDB), keywords were imported. These keywords are displayed in the **Keyword list**. Since they describe the context of a given structure, they are a good starting point for literature search.

You can select only one keyword as a **Concept** by clicking on the **!** button above the list or add a keyword to the concepts by clicking on the **+** button.

## Running the search

Click the **Search** button. MMM will not query PubMed. If the internet connection is broken, an error message is displayed both in the Message board of the main window and as a message box and control returns to the search window. Otherwise the search window closes and the new references, if any, are added to the reference window. The Message board of the main window reports how many references were found and how many of those did already exist in the reference list.

For existing references, bibliographic information is updated, except for the short name. This fixes references of papers that were in press when you first searched them.

## Autosearch and saving searches

The **Autosearch** feature provides for automatic updates of the reference list or can be used to store search terms and modes for later use.

Define your [search terms and modes](#) and click on the **Autosearch** button. A question dialog appears that asks for a (short) name for the search. If this name already exists as name of another search or if you provided an empty name, a message box appears, the search is not stored and not performed and control returns to the search window.

If a valid search name was provided, the search is stored and performed for the first time. The search window closes and control returns to the reference window, if PubMed can be accessed. If not, the internet connection is probably broken and a message box appears. Note that the search is still stored and will be automatically performed when the model is loaded for the next time (and the internet is accessible).

If PubMed could be accessed, the access date is stored with the search. This search will now be performed whenever the model is opened in MMM and the previous search results are older than 7 days. Only records that were added to PubMed between the previous search and the new search will be downloaded (plus a safety margin of one day). That way, old references that you deleted are not downloaded again, although a deleted reference may occasionally appear a second time due to the safety margin (but not a third time).

If you do not want such automatic updates, but you still want to save the search for later manual searching, open the search window again by clicking the **Search** button in the [reference window](#), (*Section 1.10*) pick the search from the **Autosearches** popup menu and **Deactivate** it with the corresponding button. The search remains stored and can always be reactivated with the **Activate** button. Note that many active automatic searches may slow down opening of the model considerably.

To delete the search terms and modes, use the **Delete** button. Note that changes in searches are stored immediately. Even if you use the **Cancel** button to close the search window, these changes remain.

## Overwhelming number of references

The number of results can be restricted with the **Maximum number** popup menu. Generally, MMM does not allow to retrieve more than 100 references with a single search, as its reference manager is not equipped for automatic selection (search in search results).

If you are back in the [reference window](#) (*Section 1.10*) and still overwhelmed with the number of references, close the reference window with the **Cancel** button, reopen it, reopen **Search** and try a more focused search (more search terms, or search only in **keywords** or even **major topics** or restriction to **only reviews** or to the last few **years**). If you need search in search results, use the official interface of **PubMed**, available through a link near the top of the search window (right click the [PubMed](#) link). In this case you have to [save and reimport](#) (*Section 1.10*) the references.

## Searching a reference in the existing bibliography

The search mask can as well be used to find a reference in the existing bibliography (all references accessible in the [reference window](#)) (*Section 1.10*). If the search does not find a matching reference, control remains with the search window and a message is displayed in the Message board of the main window. If the search finds exactly one matching reference, the search window closes automatically and the found reference is displayed in the [reference window](#). (*Section 1.10*)

If the search returns more than one matching reference, another window opens for selection of one of these references by the user. After selection and a click on the **OK** button of the selection window, both the selection and search window are closed automatically and the selected reference is displayed in the [reference window](#). (*Section 1.10*) If the **Cancel** button is clicked, the selection window closes and control returns to the search window. Click **Cancel** in the search window, if you want to return to the reference window.

For bibliography searches, the concepts field is converted to a field **Title words**. Independent phrases or words for search in the title field are separated by semi-colon. As an example **high field** will only find references whose title contains the phrase "high field" exactly as written, "high-field" will not be matched. In contrast



`high; field` will also find references that contain the phrase "high-field" in the title (but also "High-spin compounds measured at low fields").

The concepts modes (`Where to search?`) are ignored in bibliography searches; words are searched only in the title (MMM does not store the abstract or keywords).

The `Maximum number` selection is also ignored. A bibliography search will always return *all matching references* and offer the user the choice in an alphabetically ordered list.

All other search fields and the modes `first author`, `last author`, and `reviews only` can be used.

## Closing the search window

The search window closes automatically after a successful `Search` or `Autosearch`. This will not happen if the internet connection is broken (to avoid that you loose your search terms and modes). In that case and if you can't fix the connection, save the search by clicking the `Autosearch` button and then close the window with the `Cancel` button.

### 1.10.2 Define citation format

## How MMM creates citations

A citation in MMM can contain all or part of the following information:

- an `author list`: `A`
- the journal name in `abbreviated form`: `J`
- the publication year: `Y`
- the volume of the journal: `V`
- the issue: `I`
- the start page: `P` and end page: `E`
- the title of the paper: `T`

No special formats for books and book sections are supported.

The items of this list can be in any order and any of the items, as well as separators between them, can be formatted **bold** or *italic*, but not both. The order of items and the separators between them are defined by a `format string`. Formats and their assignment to journals are defined in the file `def_reference_format.m`, which is found in the `/definitions` subdirectory of MMM.

## Defining order of items and separators

The format string is stored as field `reference_formats(i).order`, where `i` is the index of the format. For example:

A, T, J V (Y) P-E.

defines a citation that starts with the author list, which is followed by a comma, the title, a comma, the year, the volume (separated by only a space from the year and enclosed in parentheses) and the start and end page separated by a hyphen. The citation ends with a full stop.

Font formatting is defined by a string `reference_formats(i).font` of exactly the same length. In this string, `p` stands for plain, `i` for *italic*, and `b` for **bold**.

## The author list

### Order of authors and separators

This is defined by a string `reference_formats(i).authors`, with identifiers `F` (first author), `I` (any author other than first or last), and `L` (last author). For example:

`F, I, and L`

defines a list, where authors are separated by commas and there is an "and" between the second-to-last and last author, with a comma before that "and". If there are only two authors, the comma before the "and" will be skipped.

Order of surname and initialsIn addition, the sequence of surname `S` and initials `I` needs to be defined, separately for the first author (`reference_formats(i).first_author`), last author (`reference_formats(i).last_author`), and any other author (`reference_formats(i).any_author`). For example:

`reference_formats(i).first_author='S, I.'` means that for the first author, the surname is followed by a comma, a space, the initial(s) and each initial is followed by a full stop.

### Maximum number of authors

Some journals abbreviate long author lists by *et al.*. The maximum number of names to be written out is given in `reference_formats(i).max_authors`. If this field is empty, always all authors are listed. Formatting of *et al.* cannot be influenced, it is always *italic* and ends with a full stop.

## How MMM formats abbreviated journal titles

The internal reference format of MMM is based on PubMed abbreviated journal titles and no correspondence list between full and abbreviated titles is kept. This works reasonably well, although there are exceptions (for instance, PubMed lists *Chem. Eur. J.* as *Chemistry*). Most journals use (superfluous) punctuation (full stops) with abbreviated journal names, while PubMed does not. Hence, MMM must decide (heuristically) where to put a full stop, if this is required. Whether or not the journal uses punctuation, is defined by

`reference_formats(i).journal_delimiter` which can be `'.'` (full stop punctuation) or `''` (no punctuation).

MMM's heuristic is based on the length of components of the journal title and assumes that abbreviated words do not end on vowels or a closing parenthesis. This does not always get it right, but usually it does.

## Assigning formats to journals

Almost all permutations in order, choices of separators, etc. are used by some journal. Nevertheless, there are more journals than formats. MMM does not mind redefinition of the same format, but avoid it, if you can. For instance, journals of the same publisher often share the same (or at least a similar format).

**### Warning ###** Publishers and their journal teams are not only idiosyncratic with respect to their citation formats, they are also fickle. Formats do change from time to time.

To assign an existing format to a journal, you need to define, e.g.

```
journals(k).name='J Irreprod Res';
```

`journals(k).format=i`; where *i* is the index of the format and *k* the index of the journal. Make sure that *k* is not yet used for another journal. If you define a new format, make sure that *i* is not yet used for an existing format. **Limitations**

Although MMM's citation format is reasonably flexible and the citation compiler avoids a few pitfalls, the citations will not always be precisely formatted. Occasionally there may be surplus or missing full stops. If a journal lists up to four author names, but only the first author and *et al.* for more than four authors, MMM will list the first four authors.

If you are very ambitious, you can edit by hand. Otherwise leave it, the journals will correct small glitches themselves, if there are only a few.

## Why is this so complicated?

Because nobody cares to standardize this. If you don't dare to define an additional format, but you need one, [ask us](#) for implementing it.

## 1.11 Find by keyword

### The keyword concept of MMM

Structure objects, such as residues, chains, or atoms can be annotated with information on function, biochemical experiments, observations in MD simulations etc. To easily find objects that are associated with a certain concept, such as

substrate transport or pH-dependent structural changes, keywords can be assigned to objects in the [annotation window](#). (Section 1.9)

To demonstrate the annotation and keyword concept the MMM distribution contains an annotated and referenced model for major plant light harvesting complex II (`LHCII_MMM.mat`).

The keyword finder window contains a list of keywords (left), a list of objects(right) and buttons for [selecting objects](#),, for [displaying the annotation](#) that contains the keyword, and [closing the window](#).

## Selection by keyword

By default objects are added to the already existing selection, if any. If you want to start with a clean slate, first click the `Clear` button.

If you click on an item in the `Keywords` list, the `Referring objects` list is filled with all objects that have an annotation with this keyword. Only a single keyword can be picked at once. You can then pick one or several objects from the `Referring objects` list (hold down the `Shift` or `Ctrl` keys during clicking for picking multiple objects). By clicking the `Select` button, you add all picked objects to the selection. Alternatively, you can use the `Select all` button to add all objects from the `Referring objects` list to the selection.

## Displaying annotation

To display the [annotation](#) (Section 1.9) of a picked object, click the `Show annotation` button. If several objects are picked in the `Referring objects` list, the annotation will be shown for the first picked item. Note that this and only this object will be selected after the annotation window is opened. If a selection of objects exists when you click the button, a question dialog opens and you are given the opportunity to cancel your request and keep the old selection.

If the keyword corresponds to an automatic annotation made by MMM (e.g., when reading the PDB file or in spin labeling), the annotation window opens at the corresponding automatic annotation page. Otherwise the annotation window opens at page 1 and you may have to click through the pages to find the remark that corresponds to the keyword.

Note also that you can return to the keyword finder only after you have closed the [annotation window](#). (Section 1.9)

## Closing the window

Click the `OK` button. All existing selections will be kept.

## 1.12 Information in PDB files

### What is the PDB format?

"The Protein Data Bank (PDB) format provides a standard representation for macromolecular structure data derived from X-ray diffraction and NMR studies. This representation was created in the 1970's and a large amount of software using it has been written." (cited from the [Protein Data Bank homepage](#)). As is often the case with official information on the PDB format, this statement is incomplete. The PDB allows for experimental techniques other than x-ray and NMR and the files contain more than only structure data.

Actually, the format has developed quite a bit since the 1970s. For some remarks on strength and weaknesses of the format, see [An assessment of the PDB format](#).. See [PDB format homepage](#) for a current format description (213 pages for Version 3.20). The following is just a condensed overview of the most important information.

### Information content

The PDB format is based on the notion that there is a unique structure of a protein, so that the coordinate of each atom can be exactly specified in a three-dimensional Cartesian coordinate frame. Although this is not strictly true, it is a reasonable approximation for those states of those proteins that are accessible to x-ray structure determination (less so for solution NMR). Accordingly, most of the lines in a PDB file specify atom coordinates and their uncertainties (B factors or anisotropic temperature factors). Atoms are assigned to residues and cofactors and residues and cofactors are assigned to chains. The assignment of cofactors to chains is sometimes arbitrary, as they may not be bound to a single peptide chain. In x-ray structures atoms or residues may be missing, although they were present in the protein under investigation. This means that the corresponding electron density could not be found or assigned, which is often an indication that there is *no unique position* of these atoms or residues. Furthermore, there may be alternate locations for the same atom, if the density could not be fitted by a single position.

Atom coordinate information is read and processed by most protein visualization programs, although some of them ignore the uncertainties (B factors, anisotropic temperature factors) and many ignore alternate atom locations. For NMR structures, usually several "models" (conformations) are given to provide an estimate of combined flexibility and uncertainty. Some protein visualization programs read only one of those models.

There is considerably more information in PDB files, which is ignored by most programs that process them. This information includes:

- the title of the protein and the protein class
- the primary citation (journal article) and, possibly, further literature references
- the biological or chemical source of the protein, e.g. the organism
- keywords

- information on the experiment that provided the structure data
- the number of alternative models for the structure (mainly NMR structures)
- author information
- information on revisions of the PDB file
- for diffraction experiments, the resolution
- metal coordination by residues and cofactors
- binding sites (usually for cofactors)
- information on the UniProt databank entry of the amino acid sequence(s)
- the sequence of the protein studied (not always consistent with coordinate entries)
- sequence modifications, e.g. mutations
- list of helices, quite often nonsensical and justly ignored by most visualization programs
- list of sheets
- list of bonds, except between amino acid atoms

MMM reads and processes all this information, except for the biological or chemical source and information on revisions. The list of helices is repaired, when found to be nonsensical. It is advisable to [redefine helices](#), (*Section 2.2*) using the information (and helix nomenclature) that is often provided in the primary citation. For sheets, the current version of MMM processes only the assignment of residues to strands, but ignores the register information how the strands form sheets (although this information is reliable). This may be mended in a future version of MMM.

In fact, PDB files may contain even more information than listed above. Most of this information is about experimental details that are not needed for modeling or structure inspection in MMM. Some other information might potentially be useful, but is missing in a huge majority of PDB files. If more PDB files with such information appear, MMM's PDB reader will be updated to use it. Note also that even the information listed above is not present in all PDB files. The PDB format defines some information as mandatory and other information as optional (and some optional information as mandatory for certain experiment types).


## How MMM stores and displays information from PDB files

MMM creates a hierarchical model of the structure that contains all residues and cofactors of which at least some atom coordinates are known and all atom coordinates, including alternate locations with their declared populations. Coordinate sets for all models of an NMR structure are stored and are individually accessible. If you are a programmer and are interested in details of the internal data structure, consult the comment section (first lines) of file `rd_pdb.m`, which can be found in the main directory of MMM.

Furthermore, MMM maintains sequence information that also lists missing residues, as far as this can be reconstructed from the sequence information in the PDB file. This is very often- but not always- the case.


MMM stores secondary structure assignments and a list of helices and strands for all chains. Unlike many other visualization programs, MMM does not try to second guess this information by analysis of backbone geometry. We advise to stick with the secondary structure assignment by the authors of the PDB entry, which,

unfortunately, is not found in the PDB file but usually in the primary paper describing the structure. Note that sometimes even the secondary structure information in the PDB file may be reliable.


Author, title, protein class, and keyword information can be inspected via the [annotation page](#) (Section 1.9) of the structure (select the whole structure and click the  [annotation toolbutton](#) (Section 2.1.11) ). Keywords are also listed when you do a [PubMed literature search](#). (Section 1.10.1)

A list of binding sites is maintained and can be accessed via the [Analysis/Sites](#) (Section 2.1.5) menu subitem. Furthermore, the residues and cofactors involved in the binding sites are [annotated](#) (Section 1.9) and can be easily found via the [keyword finder](#), (Section 1.11) , which is accessible from the [Edit menu](#). (Section 2.1.2) . The [Analysis/Sites](#) menu subitem permits convenient [binding site visualization](#), (Section 3.11) but note that the software used by the PDB annotators for binding site declaration is quite generous in granting a residue binding site status. A coordinative or hydrogen bond does not seem to be a requirement, and probably not even van-der-Waals contact.

A list of metal binding sites is maintained, but is not accessible via a dedicated window. Rather, metal binding atoms and the metal atoms itself are [annotated](#) (Section 1.9) and can be found via the [keyword finder](#), (Section 1.11) , which is accessible from the [Edit menu](#). (Section 2.1.2) The metal binding site information appears to be reliable, if provided. Note that it may be missing, even if there are metal binding sites. Mutations with respect to the database sequence (MODRES records in the PDB file) are annotated at residue level and can be found with the keyword [mutations](#).

The primary journal reference and any further references are stored in the [reference list](#) (Section 1.10) of the model. They are assigned short names that start with [PDB\\_](#) and the PDB identifier, the primary reference is marked. You can change these short names if you wish. The references are accessible via the  [references toolbutton](#). (Section 2.1.11) Reference information in the PDB file is incomplete and formatted in an ugly way. If possible, MMM reconstructs the citation by PubMed access.

Resolution, which is specified only for diffraction based structures, is shown in the title of the main window.

The UniProt sequence database information is stored for each chain. In the hierarchy window, an UniProt button  allows access to the corresponding webpage, which is a rich source of further information on this protein. This webpage may also have information on related structures.

## **An assessment of the PDB format**

The PDB format was defined in the 1970s at first exclusively for x-ray structures of proteins, at a time, when structural biology was in its infancy and computers worked with punch cards limited to 80 characters per line. Computer memory was so



expensive that people cared about saving one bit per character and thought it was unnecessary luxury to distinguish between upper and lower case letters.

The PDB format has since evolved and is still able to accommodate the most important information about atomistically resolved new structures, with some limitations. After more than three decades of evolution, entropy of the PDB format is still surprisingly low, which indicates that the original idea was a stroke of genius. We still think that one should not stick so closely to an idea for such a long time, even if it was a good one.

From the aspect of current knowledge about protein structure and the state of the art of informatics, the PDB format may appear outdated. It will still survive, since there are so many programs that process it. Modernization of the format with [duplication of the data bank](#) was attempted, but the new format provides only part of the information. The Protein Data Bank has an internal data base that is more modern, as is obvious from their web page display. MMM provides direct access to the web page for a given structure via the [PDB PDB button hierarchy window](#). (Section 2.3)

Generally, the current format does not pose big problems. There are three notable exceptions:

- HELIX records
- insertion codes
- the SEQRES record

The HELIX records appear to be generated by incorrect software, most likely wrong post-processing of DSSP information. In a substantial fraction of PDB files they are obviously wrong, i.e. consecutive helices in the same chain overlap or a continuous helix is split. They are also typically longer by one residue each on both sites than you would expect. Information from these records should be used only with utmost care. Currently MMM reads the records and corrects internal inconsistencies. If [DSSP](#) is installed, MMM reassigns secondary structure based on DSSP (with a little bit generosity to "repair" helix breaks that most structural biologists would not consider as helix breaks). The user is free to redefine secondary structure assignment (and then keep it stable when the model is reloaded in MMM format). By writing an [initialization script](#), (Section 3.9) a stable individual secondary structure definition can also be achieved when reloading the same PDB file as a local file.

The insertion codes were invented to match residue numbering to some perceived reference sequence of a related protein. Although the originators probably had good intentions and some reasons, we consider this idea as flawed. First, there is no generally accepted reference sequence for a group of proteins. Second, the PDB format does not even provide a means for listing the reference sequence used by the authors (and its origin). As a result, the residue numbering with insertion codes is arbitrary and it comes at the expense of having residue numbers that are no longer unique.

Current PDB format definitions (Version 3.20) do not advertise insertion codes anymore. However, they do not yet seem to be outlawed for new PDB entries, as they should be in our opinion.

MMM does read the insertion codes and uses them for constructing the residue address. This guarantees that in structures with insertion-code residues every residue and atom can still be addressed. What cannot be solved is the mismatch between residue numbering in the PDB structure and in the true sequence of the chain. There are two possible workarounds. First, one could automatically renumber residues to match the true sequence. This would introduce a mismatch with numbering in the original PDB file. Second, one could keep residue numbers from the original PDB file and suppress residues with insertion codes in sequence display.

The second choice is closer to the intentions of the authors of the original PDB file and- in our opinion- creates less confusion (some confusion is unavoidable). Hence, MMM works this way. However, a future version might provide the option of residue renumbering (secondary literature on such structures often does this).

The SEQRES record can only be fully interpreted together with the DBREF record (no problem) *and* a query to the data base that contains the original sequence (this is a problem).

If the DBREF record points to an UNIPROT sequence and the computer is connected to the internet when reading the PDB file, MMM solves this problem by downloading the sequence in FASTA format.

Furthermore, PDB files are not necessarily consistent about the use of MODRES and SEQADV records in reporting on engineered mutations. The current version of MMM analyzes the MODRES record, which is the proper place for this information according to the official format definition. SEQADV is supposed to be only for unclarified differences between database and true sequence. See, e.g. PDB file [1QJP](#) for a case that does not conform to official format.

## Duplication of the data bank and PDBML format

Finally, an XML format (PDBML) has been developed and the data bank duplicated. The main design decisions in this development were simplicity of translation from PDB format into this format (and back) and preservation of the semantics (J. Westbrook, N. Ito, H. Nakamura, K. Henrick, H. M. Berman, *Bioinformatics* **2005** 21, 988–992. doi:10.1093/bioinformatics/bti082). This format leads to much larger files and appears to list only atom coordinates. The advantage is that there is an abundance of software for reading, processing, and writing XML files. To our opinion, this advantage is far outweighed by the disadvantage of information loss.

## 2 User interface

### 2.1 Main window

#### What can you do in the main window?

All functionality of MMM, except for selection of objects (see: [Selection](#)) (Section 1.8) and some controls of the 3D model view (see: [Model window](#)) (Section 2.1.16) are accessible from the main window. You can load and save models in MMM format, load and save structures in PDB format, select the view on the 3D model, change color and transparency of selected objects, obtain information on a selected object, measure distances, angles, and dihedral angles, modify the model and predict structures or spectra.

The main window is also the main information source on the current state of the model (selections, ongoing computations) and on warnings or errors.

#### How can you do it?

The main window consists of six parts that help you with control and provide information. Most functions are accessible via the [menu](#), the most frequently used functions also via a [toolbar](#). (Section 2.1.11)

Information on the current view is provided in the [view panel](#), (Section 2.1.12), which also allows to set standard views and to toggle emulated depth cueing. (Section 2.1.13). Unless the 3D model is detached, it is displayed in the [model panel](#). (Section 2.1.15)

Some of the functionality is also accessible via the [command line](#), (Section 2.1.14) which simplifies complicated selections and allows for changes of color, transparency etc. without explicit selection of objects. Furthermore, command scripts can be run. The [message board](#) (Section 2.1.17) displays information on changes in the model, ongoing computations, warnings, and errors.

#### 2.1.1 Menu/File

##### Open...

Opens a model, which may consist of several related structures, in binary MMM format (a Matlab file). The model contains information on structures, the 3D view, graphics modes for the objects, coloring, transparency, object selections, and auxiliary information, such as results of spin labeling site scans.

After opening a model you can proceed with your work where you stopped before saving the model. The state of the MMM user interface (opened windows, content of message board etc.) is not saved or loaded with a model.

## Open without display.

Opens a model, but does not display the graphics and does not perform [reference autosearches](#). (Section 1.10.1) . This is convenient if you only want to inspect [keywords](#) (Section 1.11) and [annotations](#) (Section 1.9) or want to look up or export references, as the model is loaded much faster.

You are advised not to save the model again after opening it in this mode (results are not well tested). This mode is for inspection only.

## Save as...

Saves a model, which may consist of several related structures, in binary MMM format (a Matlab file). See item [Open...](#) for what is and what is not saved. The program suggests a filename based on date and time.

When trying to overwrite an existing file we sometimes encountered an error "No permission.", which seems to be a Matlab problem. It disappears when the save operation is repeated. MMM catches this error and pops up a message box that informs that the save process has to be repeated. Sorry.

## Share

Saves a model for sharing with other members of the own group (or collaborators) or for public sharing. The only difference to saving the model is that private [annotations](#) (Section 1.9) and [references](#) (Section 1.10) are not stored and that group-restricted annotations and references are not stored in public mode.

## New...

Creates a new model from an atomistic structure (PDB format) or a sequence (yet to be implemented). The current model is discarded after an inquiry- save it before! MMM imports atomic coordinates (including alternate locations and alternate models), B factors, anisotropic temperature factors, sequence information, secondary structure information (except for beta-sheet registry), the header, the title, information on sequence modification, and on atom connection of non-standard residues/cofactors. See [Information in PDB files](#) (Section 1.12) for a full description of how MMM processes PDB files. PDB file format v 3.2 is assumed, a few deviating formats can also be loaded (e.g., MolProbity proton definitions).

## Local file

A locally stored PDB file can be selected and loaded.

You are strongly advised to use original PDB files that were originally downloaded from the Protein Data Base rather than PDB files processed and saved before by another program. This is because processing and visualization programs

gegenerally do not read (and save) all information from the original PDB files and MMM is likely to use information that was ignored by other programs.

### **Direct PDB access**

This requires an internet connection and knowledge of the four-character PDB identifier. An input dialog for the PDB identifier appears and if a valid identifier is provided, the PDB file is loaded. Otherwise, an error or warning message appears in the Message Board.

### **Add...**

Adds an atomistic structure (PDB format) or a sequence (yet to be implemented) to an existing model. See "New..." for what information MMM reads from a PDB file.

### **Local file**

A locally stored PDB file can be selected and loaded.

You are strongly advised to use original PDB files that were originally downloaded from the Protein Data Base rather than PDB files processed and saved before by another program. This is because processing and visualization programs geenerally do not read (and save) all information from the original PDB files and MMM is likely to use information that was ignored by other programs.

### **Direct PDB access**

This requires an internet connection and knowledge of the four-character PDB identifier. An input dialog for the PDB identifier appears and if a valid identifier is provided, the PDB file is loaded. Otherwise, an error or warning message appears in the Message Board.

\

**Save PDB as...** Saves the current atomistic structure as a PDB file in format v3.2. The title of the main window tells you what structure this is and the structure popupmenu of the [hierarchy window](#) (*Section 2.3*) can be used to select it. Note that this does not save the whole model if it consists of several structures or contains coarse-grained domains. Note also that information on visualization and object selections are lost and that only the 26 leading rotamers can be saved for spin labels. Use "Save as..." for saving the complete model.

Limitations: Only up to 26 alternate locations are saved for an atom. As spin label rotamers are stored as alternate locations, this may lead to loss of information (only the leading 26 rotamers with highest populations are saved to the PDB file). Use "Save as..." to maintain all information on rotamers.

Note also that MMM, as virtually any other program that processes PDB structures reads only part of the information contained in an "official" PDB file. Hence, you should always keep a copy of the original file.

## **Export visualization...**

Export the 3D visualization in the detached [model window](#) (*Section 2.1.16*) to a graphics file in one of the formats specified below. This item is active only if the model window is detached (use the [detach toolbutton](#) (*Section 2.1.11*) if required).

### **bmp**

24-bit bitmap (true color). Fastest export and reasonable size. Resolution is the same as in the model window. To obtain best results, maximize the model window before and [zoom](#) (*Section 2.1.11*) the model so that the parts of interest fill the whole figure window.

### **emf**

Windows extended metafile. Slower export and almost six times larger than bmp. The same format as used with the [copy tool](#) (*Section 2.1.11*) and the Windows clipboard. Use this with MS Office applications.

### **eps**

Encapsulated postscript with 300 dpi resolution and a 72 dpi TIFF preview included. Particularly useful for LaTeX users. Very slow export and very large file.

### **jpeg**

The well known JPEG format. Smallest file size and relatively fast export, but supposedly quality is slightly worse than with tiff or bmp. You need a good printer to see the difference, though.

pdfAdobe's portable document format with 600 dpi resolution. Slow export, much smaller files than eps, but unlike with eps the picture may be clipped at maximum screen size.

### **png**

Portable network graphics 24-bit bitmap format. Usually the best compromise between size, exporting speed, and graphics quality.

### **tiff**

A 300 dpi TIFF file. Preserves full information at about the same size as bmp with reasonably fast export speed. Is accepted by almost any graphics and office program, except for LaTeX.

## Load density

Loads a density cube file for visualization in the model. Currently two binary formats are supported, the MMM native format and a format exported by a LabView program written by Christian Altenbach.

The native MMM format is a Matlab variable file with variables  $x$ ,  $y$ , and  $z$  specifying the axes and variable cube of array size  $[nx, ny, nz]$  containing the densities. Variables  $x$ ,  $y$ , and  $z$  are vectors of lengths  $nx$ ,  $ny$ , and  $nz$  and specify the coordinates of the voxels.

Altenbach's format is a double precision big endian binary file with the first nine numbers specifying the number of data points and coordinates of the cube in the order  $nx$ ,  $x0$ ,  $dx$ ,  $ny$ ,  $y0$ ,  $dy$ ,  $nz$ ,  $z0$ ,  $dz$ . Here  $x0$ ,  $y0$ , and  $z0$  are the starting coordinates of the  $x$ ,  $y$ , and  $z$  axis and  $dx$ ,  $dy$ , and  $dz$  are the respective increments. The following  $nx \cdot ny \cdot nz$  values are the densities.

Each density cube is assigned a tag for later reference. The default tag is `density_n`, where  $n$  is a running index. The tag can be edited. Note that density cubes are defined on model level and refer to the (global) model frame. The density cube is assigned to the coordinate frame of the current structure. It will transform with this structure, if you superimpose the structure onto another structure or if you make a symmetry axis,  $\alpha$ -helix bundle mean axis or  $\beta$ -barrel mean axis the new  $z$  axis. Such transformations are rather slow (voxels must be interpolated).

See the [Display menu help](#) (Section 2.1.4) for information on how to display density cubes.

## Preferences

Allows to set the preferred PDB server (Europe, USA, or Japan) and make the SFX (library) server of your home institution known to MMM for fulltext access of references. If your computing environment allows for establishing a processor pool (for instance, your PC has a multi-core processor) and the Parallel processing toolbox of Matlab is available, MMM can make use of parallel computing for some time-consuming tasks. To allow for comfortable working with your computer while MMM is running, you may want to restrict the number of processor cores that are claimed by MMM. The default value is 6, which proved convenient on a Windows 7 PC with 8 CPU cores. This should be adjusted if your CPU has more or less than eight cores.

You can also specify whether the Matlab browser or the system browser is your preferred browser for viewing help files and internet homepages. The default is `mixed`, which displays help files in the Matlab browser and internet homepages in the system browser.

[Anisotropic network model \(ANM\)](#) (Section 3.7.3) parametrization can be controlled in the corresponding panel. You can choose how force constants between network



nodes ( $C^\alpha$  positions) are computed. Keep with the defaults, unless you understand the issue and have a particular reason. Activation of the checkbox [imANM](#) makes only sense if you work with membrane proteins and if a [lipid bilayer](#) ([Section 3.13](#)) was added, so that the protein is in a standard orientation. In that case the implicit membrane (ANM) is used, as described in Lezon and Bahar, *Biophys. J.* 2012, 102, 1331-1340.

Finally, you can specify the name of the current version of the Modeller executable, if you want to work with [Modeller](#) ([Section 5.2](#)) for loop reconstruction and constraint-supported homology modeling.

## Close

Closes MMM. You are asked whether you want to save the current model.

## 2.1.2 Menu/Edit

### Undo

Undo last command, if possible. Although MMM has a multi-level undo for certain commands, most changes in a model cannot be undone. Please save your model periodically or before you apply strong changes.

### Redo

Redo last undone command.

### Annotation

Annotate currently selected object (see [annotation window](#)) ([Section 1.9](#)). This menu item is enabled only if a single object is selected.

### Find by keyword

Find objects that are annotated with a certain keyword. See also [keyword finder](#). ([Section 1.11](#)) This function also allows for selecting all objects associated with a certain keyword and inspecting annotations of objects. For example, you can find residues with missing atoms, residues involved in metal binding, or residues, for which spin label rotamers were computed.

### Center

Centers the current structure. All atomistic and coarse grained coordinates are changed, so that the mean of all atom coordinates is zero. This function takes time as the graphics has to be recomputed (the current structure may shift with respect to other structures).

## Symmetry frame

Determine a symmetry axis or pseudo-symmetry axis from selected objects. For a C(n) rotation axis this requires that n objects are selected and that each of these objects has the same number of coordinates. Example: Three equivalent chains in a homotrimer define a C(3) axis. The structure is rotated so that the symmetry axis becomes the z axis. This requires recomputation of graphics and thus takes some time. Water molecules are generally disregarded in this computation.

If the selected n objects do not conform to C(n) symmetry, an approximate pseudo-symmetry axis is determined that minimizes the root mean square deviation of the mean coordinates of equivalent points from the axis. Note also that in some cases apparently equivalent chains may have different numbers of cofactors. In this case MMM states that the objects do not define a symmetry axis. Workaround: Define domains that contain only the amino acid residues and determine symmetry axis from them.

## Assign TM helices and strands

Allows for assignment of helices and strands as transmembrane (TM) helices and strands, for computation of the mean axis of a helix bundle or  $\beta$ -barrel (approximate membrane normal), and for a coordinate transformation that makes this axis the z axis of the structure frame. This is a necessary step before [building a bilayer](#), (Section 2.1.3), unless the membrane normal was derived by [symmetry](#).

A [special window](#) (Section 3.12) opens to aid the assignment and to visualize the approximate membrane normal.

## Superimpose structures

Superimpose a "moving" structure onto a template structure by transforming the coordinate frame of the former structure (rotation and translation). The superposition minimizes r.m.s.d. of atom coordinates of the two structures or of selected parts of the two structures. Coordinates of the template structure do not change. Coordinates of the moving structure change. A dialog opens that allows for selection of the template and moving structure from popupmenus that contain all structures in the model. The fit mode can be selected by three radiobuttons and a checkbox. To superimpose cofactors, use the **all atom** mode. Superposition of peptide chains usually works better when selecting only backbone atoms or even only the C $^{\alpha}$  atoms.

It is assumed that the same chain identifiers (uppercase letters) correspond to equivalent chains in both structures and that, within each chain, residues with the same numbers correspond to each other. If the **selected only** checkbox is deactivated (default), all common residues in all chains of both structures are used for r.m.s.d. minimization. If one or both structures have several coordinate sets (models) for a chain the first coordinate sets (**{1}** in MMM [addressing](#)) (Section 1.6) are used for minimization.

The chains and residues to be used in r.m.s.d. minimization can be freely selected. Selection is always on residue level. Cofactors can be selected by their residue numbers. Note that cofactors may be ignored if only backbone or C<sup>α</sup> atoms are fitted.

Selections in the two structures may differ, but only residues that are selected in both structures are considered in fitting.

The **Save** button allows for saving the affine transformation matrix (a 4×4 matrix specifying rotation and translation) for an active coordinate transformation of the moving structure to a text file. Use the **Cancel** button if you only want to save this matrix without actually changing coordinates of the moving structure. Use the **OK** button to actually transform the moving structure.

Do not forget to save the MMM model (project) or a PDB file of the transformed structure.

## Domain

Define domain from selected objects or add selected objects to a domain. All selected objects must belong to the same structure to do this.

## References

Search, import, edit, inspect, and manage literature (and webpage) references. Abstracts and fulltext can be downloaded if the link information is available.

## Reports

A very basic editor for plain text files. Can be used for inspecting result files from DEER simulations or accessibility predictions and is automatically opened after accessibility computations. Text can be changed and saved. Not suitable for large files and not convenient for heavy editing.

**Fit from template** Structure modeling based on sparse distance constraints and coarse-grained [elastic network models](#). (Section 3.7.1) . The elastic network model is derived from a template structure, which can be either a structure of the same system in a different state or a homology model. The template structure must be the current structure in MMM (select in the [hierarchy window](#)) (Section 2.3) ). There is a [dedicated window](#) (Section 3.7.4) for importing constraints and performing the fit. Note that this window can take minutes to an hour to open (depending on size of the structure), as setup and normal mode analysis of an [anisotropic network model](#) (Section 3.7.3) is involved.

## 2.1.3 Menu/Build

### Copy structure

Copies basic information of a structure (including all coordinates) and creates a new structure from it. The new structure is made the current structure. The user is asked to supply a structure tag. Use this together with [colorscheme](#) (Section 2.2) [difference](#) to visualize structural changes during model building.

### Replace non-standard residues

Replaces non-standard residues such as selenocysteine or selenomethionine as well as spin-labels by their native amino acid equivalents (cysteine or methionine). A window appears where the user can select residue types to be replaced. Only the first location is kept (leading rotamer in case of spin labels). This will hide replaced residues in the 3D plot (if they were displayed before). The user needs to redisplay them if required.

### Repair sidechains

Repairs missing sidechains using the external freeware program [SCWRL4](#). (Section 5.2) The license for SCWRL4 must be obtained by the user and SCWRL4 must have been installed by the user and must be on the Matlab path. At least the backbone atoms must exist. The residues are no longer marked as incomplete in the sequence display of the hierarchy window, but the annotation with keyword "missing atoms" for the residues is kept.

### Grow/repack sidechains

Adds or repacks sidechains using the external freeware program [SCWRL4](#). (Section 5.2) The license for SCWRL4 must be obtained by the user and SCWRL4 must have been installed by the user and must be on the Matlab path.

### Reassign secondary structure

This requires that [DSSP](#) (Section 5.2) was installed at the time when the original PDB file was read in or a model was created. Furthermore, secondary structure assignment is protected. You first need to unprotect it by click the [lock toolbutton](#) (Section 2.1.11) or issuing the [unlock command](#). (Section 2.2) Helices and strands in the current structure are reassigned based on DSSP information. Don't forget to lock secondary structure afterwards.

### Transform to viewing frame

When the structure is rotated in the 3D model window or panel, atom coordinates are unchanged. Only the viewing direction is changed. In some cases it is useful to actually transform the structure to a coordinate frame that coincides with the viewing

frame. By using the subitems *x along viewing vector*, *y along viewing vector*, and *z along viewing vector* you can transform the structure so that any of the Cartesian frame axes is along the viewing direction. For *x* and *y*, the *z* axis is chosen as the camera up vector, i.e. as the upright direction on your screen. For *z* as the viewing direction, the *y* axis is chosen as the camera up vector. In a model with several structures, this transformation is applied to all structures.

## Bilayer

Definition of a lipid bilayer (membrane) with the membrane normal being the current *z* axis. Use the [Symmetry axis](#) (Section 2.1.2) or [Assign TM helices and strands](#) (Section 2.1.2) subitem of the *Edit* menu for conversion to such a frame.

[Bilayer definition](#) (Section 3.13) involves manual specification or automatic fitting of the central plane of the bilayer and of its width. Fitting is based on lipid propensities of residues.


Note that structures with bilayers require that the bilayer normal is the *z* axis of the coordinate frame. Any subsequent coordinate transformation of the structure will make the bilayer invalid.

## Solvent accessible surface

This feature requires installation of M. F. Sanner's MSMS program (see [third-party software](#)) (Section 5.2). A probe radius (effective solvent molecule radius) of 1.5 Å is assumed.

If no objects are [selected](#), (Section 1.8) the solvent accessible surface is computed for the current structure. Resolution adapts to the number of atoms (usually 1 vertex/ Å<sup>2</sup> for large structures, but can be up to 3 vertices/ Å<sup>2</sup> for smaller structures). In structures with several coordinate sets, only the first set is used (sorry).

Partial surfaces can also be computed for a chain (by default first coordinate set), for a selected coordinate set of a chain, and for a residue. Resolution adapts to the number of atoms. For these computations a *single* object has to be selected. Strictly speaking, these surfaces are not fully solvent accessible, as parts of them may be blocked by other residues, chains, or cofactors. They are, however, of interest for visualizations and are therefore provided. A window appears for input of a tag for the computed surface. This tag is used later to access the surface and the underlying data. A default tag is suggested, but you can choose freely, as long as the new tag is not yet used for another surface.

The solvent accessible surface (SAS) is displayed immediately after computation as a semi-transparent aquamarine cloud. it can be hidden (and redisplayed without computation) via the  [graphics mode](#) (Section 2.1.11) toolbutton. Note that some object must be selected to use this button.

The SAS is also assigned an address `$SAS:tag`, where *tag* is the tag provide above. The surface is known by this address to the [show](#), (Section 2.2) , [hide](#), (Section 2.2) , [color](#), (Section 2.2) , and [transparency](#) (Section 2.2) commands, which affect only the visualization, but not the underlying data. By using the [delete](#) (Section 2.2) command, both the surface and the underlying data can be deleted.

*Examples:*

```
color $SAS:SAS_1 aliceblue % changes color of the surface with tag SAS_1 to a  
very pale blue
```

```
hide $SAS:SAS_1 % hides the surface with tag SAS_1
```

```
show $SAS:SAS_1 % redisplay the surface with tag SAS_1 after being hidden, this is  
fast, no recomputation required
```

```
delete $SAS:SAS_1 % deletes the surface with tag SAS_1 and the underlying data
```

Note that MMM may have to recompute the surface for other tasks, if you delete it instead of only hiding it

## Fit from template

Fit a structural model from distance constraints and a template structure by using either an elastic network model and a variation of the Zheng/Brooks algorithm (W. J. Zheng, B. R. Brooks, Biophys. J. 2006, 90, 4327-4336.) or [Modeller](#). (Section 5.2)

This opens either the [Fit from template/ENM window](#) (Section 3.7.4) or the [Fit from template/Modeller window](#). (Section 3.8)

Another menu subitem allows for refinement of Modeller-generated structure ensembles.

## Domain ensemble

With the [domain ensemble](#) (Section 3.6) window peptide chain segments can be modelled that are not resolved in a crystal structure. The Monte Carlo modeller supports N-terminal, C-terminal and inserted loops and allows for label-to-label distance distribution restraints, bilayer insertion depth restraints, and secondary structure restraints. Special handling is provided for distance distribution restraintsstemming from homooligomers in which all protomers are singly labeled at the same site.

## 2.1.4 Menu/Display

### View selection

Zoom camera to the selected objects. To further enhance a detail view, you may use emulated depth cueing (*Section 2.1.13*) (button in the [view panel](#)) (*Section 2.1.12*) .

### View whole model

Sets camera properties so that the whole model is seen.

### Hierarchy

Open the [hierarchy window](#) (*Section 2.3*) for object selection, editing of secondary structure, and sequence display.

### Color

Assign color (by name or from palette) or color scheme to selected objects (see [color selection window](#)) (*Section 2.4*) .

### Transparency

Assign transparency to selected objects (see [transparency selection window](#)) (*Section 2.5*) .

### 3D display mode

Assign a 3D display mode (e.g. wire, stick, space-filling, ribbon, string) to selected objects (see [graphics mode window](#)) (*Section 2.6*).

### Density

Control display of density cubes. A dialog appears that allows to select the density cube from a list of all previously loaded data sets (see [File menu help](#) (*Section 2.1.1*) for how to load density cubes).

The contour level is relative to the range of the density data and can vary between 0 and 1. It can be changed by direct input of the number or with a slider; the two controls are coupled. By default data are normalized to the density integral. At a level of 0.8 then an isosurface is drawn that includes 80% of all density. In this mode, larger levels correspond to larger density clouds.

If the `normalize to integral` checkbox is deactivated, data are normalized to maximum density. At a level of 0.8 then only voxels are shown that have at least



80% of the maximum density. In this mode, larger levels correspond to smaller density clouds.

In both modes, a level  $\geq 0.999$  suppresses display. This can also be used to hide a previously displayed density cube.

Transparency of the density cloud can be specified between 0 (invisible) and 1 (opaque). Opaque looks most impressive if nothing inside the cloud needs to be seen. Otherwise the default value of 0.5 (semi-transparent) is usually a good compromise.

The color of the density cloud can be specified via a palette by using the `Color`.

Density display resets the camera view so that all displayed objects are seen.

The density surface is also assigned an address `$density:tag`, where *tag* is the tag provided above. The surface is known by this address to the [show](#), (Section 2.2) , [hide](#), (Section 2.2) , [color](#), (Section 2.2) , and [transparency](#) (Section 2.2) commands, which affect only the visualization, but not the underlying data. By using the [delete](#) (Section 2.2) command, both the surface and the underlying data are deleted. If the surface is to be redisplayed later, the density data have to be [reloaded](#). (Section 2.1.1)

*Examples:*

```
color $density:spin aliceblue % changes color of the density cloud with tag "spin" to a very pale blue
```

```
hide $density:spin % hides the surface with tag "spin"
```

```
show $density:spin % redisplay the surface with tag "spin" after being hidden, this is fast, no recomputation required
```

```
delete $density:spin % deletes the surface with tag "spin" and the underlying data
```

The last command frees the memory and storage space of the density cube, which may be substantial.

## Depth cueing

Switch on emulated depth cueing to obtain a "more 3D" impression (see depth cueing description) (Section 2.1.13) .

## Localization

Displays possible locations of a spin label with respect to a reference structure as an isoprobability surface. The localization can also be presented as an ensemble of pseudo-atoms at possible spin label localizations and can be saved into a PDB file in that way. See [Spin label localization](#) (Section 3.3) for detailed explanation.

## 2.1.5 Menu/Analysis

### Context

Explore the context of a selected object (other objects on the same hierarchy level that are in the vicinity). This menu item is only enabled if a single object is selected. The context radius to explore can be input, a default value is suggested that depends on the type of object (chain, residue, atom). The information is displayed in the message board.

Note that context information with the default radius is also accessible via the annotation button (white "i" on green background) or the "Annotation" item in the [edit menu](#). (*Section 2.1.2*)

### Distance

Displays the distance between two selected objects in the message board. For objects larger than atoms, the distance between mean coordinates is displayed. The item is only enabled if two objects are selected.

### Angle

Displays the angle between three selected objects (at the second object) in the message board. For objects larger than atoms, the angle between mean coordinates is displayed. The item is only enabled if three objects are selected.

### Dihedral

Displays the dihedral (torsion) angle for four selected objects in the message board. For objects larger than atoms, the dihedral angle between mean coordinates is displayed. The item is only enabled if four objects are selected.

### Sequence alignment

This uses [MUSCLE](#) (*Section 5.2*) AND requires that two chains are selected. For example, load PDB structures 1OMP and 2HQ0 and select chain A of each structure in the [Hierarchy window](#) (*Section 2.3*) or by the command

```
select [1OMP,2HQ0] (A)
```

Now use the `Analysis/Sequence alignment` menu item to see the alignment in a report editor in Clustal format. The result may differ from a ClustalW result, since the algorithm is different. Currently, only pairwise alignment is supported.

## Ramachandran plot

Compare backbone dihedrals  $\phi$  and  $\psi$  with expectations from residue-specific Ramachandran plots (not yet implemented).

## Contacts and clashes

Analyze the model for contacts and clashes. The following subitems exist.

### *Crystal contacts (SCWRL4)*

Based on the assumption that significant crystal contacts tend to change packing preferences of side chains. Two copies of the current structure are created. In the first copy, side chains are repacked with SCWRL4 without considering crystal environment. In the second copy, they are repacked considering this environment (note that SCWRL4 does not consider cofactors, non-native amino acids and nucleic acids in neighboring unit cells). A ribbon model of the first copy is shown and [colorscheme](#) (Section 2.2) *difference* is applied to visualize residue-by-residue differences between the first and second copy. Parts of the structure where sidechain rotamers are unaffected by the crystal environment are blue, residues with sidechain r.m.s.d. larger than 3 Å between rotamers with and without crystal environment are shown red. For sidechain r.m.s.d. between 0 and 3 Å, a rainbow color grading with 20 steps is applied. Residues with r.m.s.d. larger than 0.2 Å are reported in the Message board. Use the *Show log* button in the main window to inspect this list (or save a copy). Significant packing effects are those where several nearby residues repack.

This feature requires that [SCWRL4 from Dunbrack's lab](#) (Section 5.2) is installed.

Do not apply any coordinate transformations (*Center, Symmetry, Transform to viewing frame*, definition of transmembrane domains or lipid bilayers) before trying to analyze crystal contacts. MMM does not transform the unit cell definition, hence the results would be wrong. To avoid this, unit cell information is removed on coordinate transformations.

### *Crystal contacts (CryCo server)*

If the current structure is a native PDB structure (with valid PDB identifier), this calls the CryCo server at Weizmann Institute with this PDB identifier. The homepage of this server is self-explaining and has its own visualization capabilities. CryCo is based on distance analysis.

### *Contacts of structural units*

If the current structure is a native PDB structure (with valid PDB identifier), this calls the CSU server at Weizmann Institute with this PDB identifier. The homepage of this server is self-explaining and has its own visualization capabilities.

Note that unlike CryCo, CSU can also be used with modified structural models. Save your model as a PDB file, using the [File/Save PDB as..](#) (*Section 2.1.1*) menu item and upload it on the CSU server.

## Accessibility

Analyze the current structure for accessibility of residues and cofactors to solvents. This requires that [MSMS by Michel Sanner](#) (*Section 5.2*) is installed.

If any structures, chains, chain models (coordinate set) or residues/cofactors are selected, the analysis is performed for all selected residues/cofactors and all residues/cofactors in selected objects. If nothing is selected, the analysis is performed for the current chain in the current structure. A query dialog window opens that allows to specify the effective radius for the solvent molecule (radius of the probe sphere). The default value is 1.5 Å for water.

The solvent accessible surface (SAS) is given as a polyhedron with triangular faces. For each residue or cofactor, the total surface accessibility in Å<sup>2</sup> is computed by summing the area of all triangular faces that are in contact with the residue. If some atoms have alternate locations, only the first location of the atom is considered. For spin-labeled residues in particular, this means that the total surface accessibility of the leading rotamer (the one with the highest population) is computed. To compute spin label accessibilities that refer to the N-O group and the whole rotamer ensemble, use the [EPR/Accessibility](#) (*Section 2.1.7*) menu item instead.

For native amino acid residues, the relative accessibility is also computed as the ratio between the accessible surface and total surface of the amino acid (total surface area values are from D. Whitford, Proteins. Structure and Function, John Wiley & Sons, Chichester (2005), Table 2.2 and refer to the residue X in a G-X-G tripeptide). For non-native amino acids, such as spin-labeled residues, or cofactors the relative accessibility is specified as -1.

Results are written to a plain text (ASCII) file `accessibility.dat` in the temporary directory. This file is automatically opened in the [report editor](#). (*Section 2.7*) . It starts with three comment lines (first character is %) that explain the content. The following lines contain two numerical values, the total accessibility in Å<sup>2</sup> and the relative accessibility, followed by a % character and the residue address in [MMM address format](#). (*Section 1.6*)

If you want to store the results for later use, please save them to another directory. The temporary directory is regularly cleaned.

## Sites

Inspect, edit or newly define binding sites via the [sites window](#). (*Section 3.11*)

## Model comparison

For a structural transition starting from an initial structure, a figure of merit for two alternative models is computed by comparing the two models and comparing each of them to the initial structure. A figure of merit of 1 indicates perfect agreement, a figure of merit larger of 1/3 reasonable agreement. Please also compare models visually, using the [motion command](#). (*Section 2.2*)

## 2.1.6 Menu/Dynamics

### General remarks

Features for coarse-grained structural dynamics are currently under development. They are not included in public release of MMM 2010. The corresponding menu items are thus disabled.

### Gaussian network model

Computes a residue-level Gaussian network model with nodes at coordinates of the C $^{\alpha}$  atoms. Dependence of force constants on C $^{\alpha}$ -C $^{\alpha}$  distance is optimized for obtaining good agreement of computed and experimental B factors and for providing a well resolved covariance matrix.

On 32-bit Matlab, the structure must not have more than 3600 residues. Network computation may take minutes for large protein complexes. Computation time scales with the cube of the number of residues.

See [GNM window](#) (*Section 3.7.2*) for further information.

### Anisotropic network model

Computes a residue-level anisotropic network model with nodes at coordinates of the C $^{\alpha}$  atoms. Dependence of force constants on C $^{\alpha}$ -C $^{\alpha}$  distance is optimized for obtaining a good description of large-scale domain movements.

On 32-bit Matlab, the structure must not have more than 1200 residues. Network computation may take tens of minutes for large protein complexes. Computation time scales with the cube of the number of residues.

See [ANM window](#) (*Section 3.7.3*) for further information.

## 2.1.7 Menu/EPR

### Site scan

#### selected chains or current chain

Scans selected chains for spin labeling sites that are favourable for accessibility or distance measurements. See also [spin labeling](#). (*Section 3.1*) Note that a site scan can take hours to complete. The results are stored with the model and a report is stored as an HTML file and opened in a Matlab browser. The structure remains unchanged, i.e., labels are not actually attached.

#### selected residues

Computes rotamers of a spin label for all currently selected residues. See also [spin labeling](#). (*Section 3.1*) Note that a site scan takes 1-2 minutes per selected residue. The results are stored with the model and a report is stored as an HTML file and opened in a Matlab browser. The structure remains unchanged, i.e., labels are not actually attached.

### Label

Spin-label selected residues. All rotamers with sufficiently low energy will be attached as alternate locations. Computation of the rotamers takes a minute or several minutes per residue. Label type and experimental conditions are specified in the [labeling window](#). (*Section 3.1*) If rotamers were previously computed in a site scan, labeling takes only seconds rather than minutes.

### DEER

Predict distance distributions between labelled sites and DEER data. Predictions can be superimposed with experimental data. For this to make sense at least two residues must have been spin-labelled in the same structure. See [DEER window](#) (*Section 3.2*) for further explanation.

### Accessibility

Predicts accessibility measures for water, oxygen, or NiEDDA for a spin-labelled site. Previous computation of spin label rotamers by a [spin-labeling site scan](#) (*Section 3.1*) is required. Unless accessibility is computed for only a single site, *labels should not be attached*. Otherwise neighboring spin labels may block access to the spin label under consideration, which would be observed in experiments on a singly labelled sample.

All spin-label analysis features are at beta-version level of development in MMM 2010. A forthcoming paper will describe implementation and performance. Changes

in the underlying subroutines (and thus in results) may happen in the next version of MMM.

### **water**

This predicts a deuterium three-pulse ESEEM accessibility parameter, as defined in:

A. Volkov, C. Dockter, T. Bund, H. Paulsen, G. Jeschke, Biophys. J., 96, 1124-1141 (2009)

Two files are created in the temporary directory `MMM\tmp`, a detailed report `label_accessibility.dat` with information on accessibility of individual rotamers of all labels and a summary report `avg_label_accessibility.dat` on average accessibility of each label. The summary report is displayed in the [report editor](#). (Section 2.7) The underlying approach will be described in a forthcoming paper.

### **oxygen**

This predicts an oxygen accessibility parameter, with parametrization based on computations for a spin-labeled WALP-23 peptide and experimental data from:

R. D. Nielsen et al. JACS 2005, 127, 6430-6442.

Two files are created in the temporary directory `MMM\tmp`, a detailed report `label_accessibility.dat` with information on accessibility of individual rotamers of all labels and a summary report `avg_label_accessibility.dat` on average accessibility of each label. The summary report is displayed in the [report editor](#). (Section 2.7) The underlying approach will be described in a forthcoming paper.

### **NiEDDA**

This predicts a NiEDDA accessibility parameter, with parametrization based on computations for T4 lysozyme mutants, corresponding to experimental data from:

C. Altenbach et al. Biophys. J. 2005, 89, 2103-2112.

Two files are created in the temporary directory `MMM\tmp`, a detailed report `label_accessibility.dat` with information on accessibility of individual rotamers of all labels and a summary report `avg_label_accessibility.dat` on average accessibility of each label. The summary report is displayed in the [report editor](#). (Section 2.7) The underlying approach will be described in a forthcoming paper.

## **P-31 Mims ENDOR**

Predicts  $^{31}\text{P}$  Mims ENDOR spectra for a spin label attached to a protein near or within a lipid bilayer or for structures with phosphorous atoms (e.g. protein-DNA and protein-RNA complexes or spin-labeled DNA/RNA. See description of the [ENDOR window](#). (Section 3.14)



The lipid bilayer model is based on DPPC parameters in:

N. Kucerka et al., Biophys. J. 2008, 95, 2356-2367.

## 2.1.8 Menu/Biochemistry

### Mutation

Mutation requires that a single residue is [selected](#). (Section 1.8) A window appears that allows to mutate this residue to any other natural amino acid. If you choose the original type of amino acid, the sidechain is repacked. The function requires that you have the third-party software [SCWRL4](#) (Section 5.2) installed and that all backbone atoms of the selected residue are specified in the structure.

## 2.1.9 Menu/Predict

### CD spectrum

Opens a Matlab browser displaying a Web server that allows for CD spectrum prediction from a PDB file. Note that only standard amino acids, not cofactors, are considered in this computation. Save your structure as a PDB file and upload this PDB file to the server. Results will be sent by e-mail. The server is courtesy of The Hirst group, University of Nottingham.

### Secondary

Offers access to several Web servers for prediction of secondary structure from amino acid sequence. These servers are:

- PHYRE, courtesy Structural Bioinformatics Group, Imperial College, London
- PSIPRED, courtesy Bioinformatics Group, University College London
- SAM, courtesy Bioinformatics, UC Santa Cruz
- PORTER, courtesy University College, Dublin
- SABLE, courtesy University of Cincinnati

### Tertiary

Offers access to several Web servers and a program for prediction of tertiary structure from amino acid sequence. These servers are:

- HHpred, courtesy Max Planck Institute for Developmental Biology, Tübingen
- PHYRE, courtesy Structural Bioinformatics Group, Imperial College, London
- SAM, courtesy Bioinformatics, UC Santa Cruz

## Quaternary

Offers access to [MMMDock](#), (*Section 3.5*) a constraint-based rigid-body docking module of MMM, and to the HADDOCK Web server. An EPR-constraint based HADDOCK version is under development by Morgan Bye (Norwich) and will be linked when publicly available. The server is:

- HADDOCK, courtesy Bijvoet Center for Biomolecular Research, Utrecht University (requires registration)

## TM topology

Offers access to several Web servers for prediction of transmembrane topology from amino acid sequence. Unlike tertiary structure prediction this works quite well even if no structure for a similar integral membrane protein is known. The servers are:

- MEMSAT3 (via PSIPRED), courtesy Bioinformatics Group, University College London
- TMHMM, courtesy Center for Biological Sequence Analysis, Technical University of Denmark
- PHILIUS, courtesy Yeast Resource Center, University of Washington

## Disorder

Offers access to a Web server for prediction of disordered domains of a protein from the sequence:

- Spritz, courtesy University College Dublin and University of Padua

## Hinges

Offers access to Web servers for prediction of hinges that allow for relative movement of protein domains. A PDB file must be uploaded. The servers are:

- StoneHinge, courtesy Yale University
- HingeMaster, courtesy Yale University

## Domains

Offers access to a Web server for prediction of conserved domains of a protein from the sequence:

- NCBI conserved domains, courtesy NCBI

## Toolboxes

Offers access to a Web page that lists a lot of protein prediction and analysis software (no longer updated, but still very useful):

- ExPASy, courtesy of Swiss Institute for Bioinformatics

## 2.1.10 Menu/Help

### Overview

Opens the help index file in a Matlab browser.

### About MMM

Displays information on version and authors of MMM.

### Credits

Acknowledges open-source software used in development of MMM, authors of modules, and advice.

### License

The general open-source license conditions that apply to MMM and all open-source software from other authors that was included into MMM. By using MMM you agree to the terms of this license.

### Third-party software

Explains how to install third-party software that is not bundled with MMM (for licensing reasons), but is used for some functionality (because it is so good). Generally we consider only third-party software that is free for academic use and sufficiently well documented.

### Create bug report

Helps to report a bug in MMM to the program authors. The bug report contains information on the version of MMM, the Matlab version, the computer (Windows, 64-bit Windows, Linux, Mac) and on all output to the Matlab command window since the start of the session, which includes Matlab error messages. Unless you have actively switched off session logging, the report also contains all output to the message board and all issued commands (input into the command line and commands automatically generated by MMM subroutines).

The bug report is opened in MMM's report editor (usually as read-only) and an e-mail to the MMM authors is prepared by opening your system's mailtool with their e-mail address. Nothing is sent automatically.

Please specify a subject line for the mail (we appreciate MMM bug: *task*, where *task* describes what you wanted to do). Please give a bit additional information and copy the error report from the report editor window into the e-mail (or attach the log file).

Note that a log file may contain sensitive information about the project you are working on. If you don't want MMM developers to see certain things, please delete them manually from the log text before you send the mail!

If you switch off logging right at the beginning of the session, only version information and error information is contained in the error report. In many, but not all cases, this will be sufficient to find the bug. Please include at least the final section of a bug report:

```
BUG> follows the output into the Matlab command window:
```

and

```
BUG>--- end of Matlab command window output ---
```

## 2.1.11 Toolbar

### What can you do with the toolbar?

The toolbar provides quick access to the most frequently used functions of MMM. Follows a description of all buttons.



#### Save

Saves the current model in a *temporary* directory with a generic file name based on date and time (backup save). The model is saved in MMM's binary format and includes all display and selection information, except for the state of the user interface (open windows).

Note that the file is not saved with the file name and in the directory that were given in the previous use of the [File/Save as..](#) (Section 2.1.1) menu item, although this is the standard behavior of most programs. MMM does not do that to avoid that you unintentionally overwrite an old file. The reasoning is that you may want to keep copies of different versions of a model, for instance, with different coarse graining or different types of labels attached.

Note also that MMM checks every 30 days whether there are old files (older than 30 days) in its temporary directory and offers the user to delete them. Hence, you are strongly advised to copy temporary files that you actually want to keep to another directory immediately after an MMM session. This is particularly important if several users share the same installation of MMM.



#### Open

Opens an MMM model, asking for the file name. The file dialog starts in the current directory. Change to the /tmp directory to select the last saved file for recovery.

## **Lock (toggle button)**

Locks/unlocks secondary structure definitions. By default, secondary structure definitions are locked except during execution of an initialization script immediately after loading a PDB file.

The reason for locking secondary structure assignments is maintaining consistency in coarse graining. When locked, the user cannot inadvertently redefine secondary structure by clicking or executing secondary structure definition commands.

## **Detach**

Detaches the 3D display as a separate [model window](#) (*Section 2.1.16*) that can be resized and maximized. Use this if you want to copy the model to the clipboard or if you want to see details. Zooming the model and detail views also require a detached model window. The 3D graphics can be reattached by toolbar button, display submenu item or simply by closing the model window.

## **Attach**

Reattaches the 3D display to the main window. If the 3D graphics is too large, press the "View all" button.

## **View selected**

Zooms the camera so that the selected objects completely fill the model window. This works properly only with the detached model window (graphics is not clipped to the model panel).

## **View all**

Zooms the camera so that all displayed graphics is seen.

## **Selection mode (toggle button)**

When selection mode is on (button pressed down), left clicking on a graphic object selects the corresponding structure object and undoes previous selections.

Left clicking, while the shift key is pressed, selects the clicked object in addition to the previous selection. This is also achieved by clicking with the middle mouse button.

Left clicking on a selected object, when the Ctrl key is pressed, unselects this object and does not change the remaining selections. This is also achieved by clicking with the right mouse button.

A (left) double click on an object removes all selections.

The current selections are listed in the message board. Note that only up to about 20 selected objects fit the message board (the last 20 are displayed).

The Selection mode button is released when clicking the Rotate, Zoom, or Pan buttons. Note that the Selection button has to be activated if the 3D graphics is attached and the [command line](#) (Section 2.1.14) is to be used.

### **Rotate (toggle button)**

When Rotate is on (button pressed down), mouse clicks and movements in the [model panel](#) (Section 2.1.15) or [model window](#) (Section 2.1.16) allow for rotating the object.

Click left and draw for rotating the camera around the object, release to update the view indicator in the view control panel. (Section 2.1.12)

Click left while pressing the shift key (or click with the middle button) and draw to rotate the vertical axis of the camera around the view direction.

The Rotate button is released when clicking the Selection mode, Zoom, or Pan buttons.

In the Rotate mode, atom graphics may be partially or fully suppressed to avoid extreme slowdown of the graphical user interface. Atom graphics is automatically restored to the previous state when you return to the Selection mode or during copying or exporting of the visualization.

### **Zoom (toggle button)**

When Zoom is on (button pressed down), mouse clicks and movements in the [model panel](#) (Section 2.1.15) or [model window](#) (Section 2.1.16) allow for zooming the object.

Click left and draw down to enlarge the graphics (zoom in), click and draw up to shrink it (zoom out).

Click left while pressing the shift key (or click with the middle button) to pan (move the graphics midpoint by drawing) while in Zoom mode..

The Zoom button is released when clicking the Selection mode, Rotate, or Pan buttons.

In the Zoom mode, atom graphics may be partially or fully suppressed to avoid extreme slowdown of the graphical user interface. Atom graphics is automatically restored to the previous state when you return to the Selection mode or during copying or exporting of the visualization.



## **Pan (toggle button)**

When Pan is on (button pressed down), mouse clicks and movements in the [model panel](#) (Section 2.1.15) or [model window](#) (Section 2.1.16) move the midpoint of the graphics in the viewing plane.

The Pan button is released when clicking the Selection mode, Rotate, or Zoom buttons.

In the Pan mode, atom graphics may be partially or fully suppressed to avoid extreme slowdown of the graphical user interface. Atom graphics is automatically restored to the previous state when you return to the Selection mode or during copying or exporting of the visualization.



## **Color selection**

Assign a uniform color or a color scheme to the currently selected objects. A [color selection window](#) (Section 2.4) opens.



## **Transparency selection**

Assign a transparency to the currently selected objects. A [transparency selection window](#) (Section 2.5) opens.



## **Graphics mode**

Assign graphics modes to the currently selected objects. A [graphics mode window](#) (Section 2.6) opens.



## **Copy to clipboard**

Copies the current 3D visualization in the detached [model window](#) (Section 2.1.16) to the clipboard. This toolbutton is active only when the model window is detached. If necessary, use the [detach toolbutton](#) before.



## **Annotation**

Is active only if exactly one object is selected and allows for editing the annotations (notes, keywords, and references) for this object. An [annotation window](#) (Section 1.9) opens.



## **Context**

Is active only if exactly one object is selected. Searches for other objects on the same hierarchy level that are close to the selected object.



A dialog opens for specifying the context radius. Default values are suggested, 3.5 Å for atoms and atom locations, 8 Å for residues, and 40 Å for chains.

Results are displayed in the message board.



## Distance

Is active only if exactly two objects are selected. Displays the distance between atoms (and atom locations) or the distance between mean coordinates of other objects (residues, cofactors, chains).

Results are displayed in the message board.



## Angle

Is active only if exactly three objects are selected. Displays the angle between objects at the second selected object. For objects larger than atoms, mean coordinates are used.

Results are displayed in the message board.



## Dihedral

Is active only if exactly four objects are selected. Displays the dihedral angle (torsion angle) defined by these objects in the sequence of selection (see selection list in message board). For objects larger than atoms, mean coordinates are used.

Results are displayed in the message board.



## Spin labeling

Attach spin label to the selected residues. Selected objects other than residues are ignored. Use Menu/EPR/Site scan to do this for a whole chain.

A window for specifying [labeling conditions](#) opens.

After specifying conditions and clicking on OK, computation progress is displayed in the message board. A status bar window shows progress and approximated computation time left for each single residue. The computation for a residue can be interrupted by trying to close the status bar window.

Note that these computations can take long (several minutes per residue). Consider running a site scan (see [submenu EPR](#)) (*Section 2.1.7*) over lunch break or over night. With a completed site scan, (in silico) spin labeling takes only seconds and you know beforehand which sites are favorable.

## References

Search, import, edit, inspect, and manage literature (and webpage) references. Abstracts and fulltext can be downloaded if the link information is available.

## 2.1.12 View control

### What can you do with view control?

View control informs you on the current camera view direction and orientation, allows you to select axes of the Cartesian frame as viewing axes, and can be used to toggle atom graphics display as well as emulated [depth cueing](#). (Section 2.1.13)

### Cartesian frame display

A trihedron with red (x axis), green (y axis), and blue (z axis) vertices shows how the camera sees the 3D model. An object rotation with mouse click and drag, the tripod updates only after mouse button release.

### Memorizing a view

You have maybe found the perfect view for your problem, but just in case it is not yet perfect, you want to play a bit more. How do you memorize what you had already found?

Click the **M** button in the **View** panel. This memorizes *all* camera settings, including zoom. You can later retrieve the last memorized view by selecting item **memory** from the popup menu to the right of the **M** button.

### Special views

A popup menu right from the Cartesian frame display allows to select viewing directions along the x, -x, y, -y z, and -z axes. If none of these directions is selected, this popup menu displays the term **arbitrary**. Selecting one of the special views resets camera zoom so that all displayed graphics is seen.

### Viewing vector

Below the popup menu, the current viewing direction is specified as a vector in dark blue writing. This allows to write down the current viewing direction and to reproduce it exactly at a later time by using the [view command](#). (Section 2.2) . This is followed by a slash and the camera up vector in dark green writing. To obtain exactly the same appearance, viewing direction and camera up vector must coincide. The camera up vector can be set by the [camup command](#) (Section 2.2) or, together with the viewing vector, by the [view command](#) (Section 2.2) with six arguments.

## Depth cueing

This button activates [Depth cueing](#), (*Section 2.1.13*) which has a dedicated control window.

## Atom graphics

Beyond a certain number of graphics objects Matlab graphics becomes very slow. This slows down operations in the graphical user interface and can make it completely impractical to rotate, zoom, and pan the 3D model interactively.

This situation is most likely encountered when you display large conformational ensembles of spin labels (many rotamers) with atom graphics, i.e., as stick, wire, ball & stick, or space-filling representations.

As a workaround, click the `Atom graphics on` button. It changes to `Atom graphics low` with yellow lettering. The graphical representation for each displayed atom is replaced by a single dot and graphics speeds up. If the speed up does not suffice, click the button again. It changes to `Atom graphics off` with red lettering. All atom graphics is now invisible, but not deleted. Graphics speeds up once again.

Once you are finished with setting up your graphics, click the button until the lettering becomes green again and reads `Atom graphics on`. Allow some time between clicks for display update.

Note that you can generate atom graphics with the [show](#) (*Section 2.2*) command or the [graphics mode](#) (*Section 2.6*) window even when the `Atom graphics` button is switched off. The graphics just is not displayed until you click the button. For a reason unknown to us it is sometimes necessary to click twice through all the states before the graphics is really displayed.

## 2.1.13 Depth cueing


### Depth cueing concept


Depth cueing helps to see in a two-dimensional graphics of the model which objects are situated in the foreground and which ones in the background. There are several ways to achieve that. Objects can blend into the background color with increasing depth and they can be made more transparent with increasing depth. In Matlab graphics, a combination of both these modes together with a uniform white background works best.

Modern graphics cards allow for more sophisticated and faster hardware-supported depth cueing. Currently, Matlab graphics does not support this feature. Hence, depth cueing in MMM is emulated and a bit slow.

## Depth cue control

### Preparation

Depth cueing is most useful for detail views of a model although it can also be used for viewing the whole model. To use it for the whole model, unselect all objects, for instance with an `unselect *` command in the command line or double click on an object in the 3D graphics display. Then click the  `View all` toolbutton. If necessary, adjust the view as described in the [model panel help page](#). (Section 2.1.15)

To use it for a detail view, [select](#) (Section 1.8) the objects that you want to see, for instance in the [hierarchy window](#), (Section 2.3), by clicking on the structure in the [model window](#) (Section 2.1.16) or by the [select command](#). (Section 2.2) Then click the  `View selected` toolbutton.

### Switching on

Depth cueing is toggled (switched on/off) by clicking on the `Depth cueing` button in the `View` panel of the main window or switched on via the `Depth cueing` item of the `Display` menu.

A `Depth cueing control` window opens, the background color of the [model panel](#) (Section 2.1.15) or [model window](#) (Section 2.1.16) is switched to white, selection highlights are switched off, and depth cueing is enabled. Objects in the front that could obscure the view are cut away.

### Adjusting

Depth cueing is controlled by two planes that are perpendicular to the viewing direction  $d$ . Your eyes are at a negative coordinate along  $d$ , the background is at a positive coordinate. The `Front plane` is by default set at  $d=0$ , corresponding to the front edge of the selected objects or the whole model. Front edge computation is approximate, small adjustments may be necessary.

You can adjust the `Front plane` by the corresponding slider or by directly setting a value in the edit field. Note that depth cueing does not automatically update, you have to click the `Update` button.

All objects closer to you than the front plane are cut away. The front plane also corresponds to fully opaque objects with full color saturation. If the front plane is too close to your eye (large negative values), everything will look very pale.

The `Back plane` is by default set at  $d=1.5$ , corresponding to 1.5 times the extension of the selected objects (or whole model) behind the front plane. This makes sure that the whole selection can still be seen rather clearly. Shifting the backplane to the back (larger values of  $d$ ) makes more objects visible in the background and the color

saturation gradient is decreased. You can shift the `back plane` with the corresponding slider or by direct input in the edit field.

## Switching off




Click the code>Cancel button of the `Depth cueing control` window or, alternatively, click once more on the `Depth cueing` button in the `View` panel of the main window. The window closes, depth cueing is deactivated, and selection highlights- if any- are restored. The previous background color is also restored.

## 2.1.14 Command line

### What can you do with the command line?

Part of the functionality of MMM is available via [commands](#). (Section 2.2) . In particular, selection of objects and change of the graphics mode, color, and transparency of objects are often more conveniently done via commands than via clicks. This applies even more to definitions of domains and secondary structure elements. Commands can be collected in [scripts](#). (Section 3.9)

### Behaviour of the command line

The command line can only be selected, when the main window cursor is in selection mode (button  is activated, buttons  and  are deactivated) or if the [model panel](#) (Section 2.1.15) is detached as a [model window](#). (Section 2.1.16)

If not selected the command line has the grey background of the main window. Any commands that are performed from a script or by menu and tool button functions are echoed with dark grey letters and are added to the command history list.

When selected, the command line becomes white. Click twice to see the cursor. A right click clears the command line (but not the history list). Commands can be directly written into the line. Previously issued commands can be selected by the cursor up (`&uarr`) and down (`&darr`) keys. To actually execute a command, press the Return (`&crarr`) key.

The command `help` displays the [list of available commands](#) (Section 2.2) in the [message board](#). (Section 2.1.17) Commands can be abbreviated, i.e. only the first letters need to be given if they are unique. For instance, `attach` can be abbreviated by `att`, while `col` is not unique (`color` or `colorscheme`). If an abbreviated command is not unique, no action is taken and a warning is displayed in the message board.

### Script button

Allows to run a command [script](#). (Section 3.9) By default, scripts are expected in the `/scripts` subdirectory of the MMM directory and to have the extension `.mmm`, although files with other extensions can be loaded.

Commands in the script are echoed in the command line and added to the command history list. Comments are echoed in the [message board](#). (Section 2.1.17)

## 2.1.15 Model panel

### What can you do with the model panel?

The model panel displays the 3D graphics of the model as a small, fixed-size picture within the main window of MMM. The model can be rotated, zoomed, and panned (shifted) by mouse click and drag in the panel. The view can also be changed using the [view control panel](#). (Section 2.1.12)


Structure [objects](#), (Section 1.5), such as residues or atoms, can be selected by mouse click and the [background color](#) can be selected among black, grey, and white by using a [context menu](#) (click with the right mouse button).


The same functionality is available with a larger and scalable picture by detaching the model panel as a [model window](#). (Section 2.1.16)

### Getting the best view on the model


#### You do not yet know what you are looking for


To explore the model in general, you may want to turn and zoom it. Actually the model remains fixed in space and the camera is moved around to simulate such rotation and zoom.


To rotate the model, activate the  rotate toolbutton. Now click left and draw for rotating the camera around the object, release to update the view indicator in the view control panel. (Section 2.1.12) Click left while pressing the shift key (or click with the middle button) and draw to rotate the vertical axis of the camera around the view direction.

To zoom the model activate the  zoom toolbutton. Now Click left and draw down to enlarge the graphics (zoom in), click and draw up to shrink it (zoom out). Click left while pressing the shift key (or click with the middle button) to pan (move the graphics midpoint by drawing) while in Zoom mode.

Model graphics is not clipped to the model panel and may "invade" other parts of the main window when you zoom in. If this disturbs you, detach the panel using the


 detach toolbutton.

If you want to shift the center of the 3D model activate the  pan toolbutton. Now mouse click and drag moves the midpoint of the graphics in the viewing plane.

When you are done, do not forget to reactivate the  selection mode. Otherwise you cannot select objects by click and the [command line](#) is inaccessible.

For very complex graphics (many graphics objects), a Matlab graphical user interface becomes very slow. MMM encounters this problem when a lot of atom graphics is displayed, in particular for spin label rotamers. In this case atom graphics is automatically reduced or switched off in the rotate, zoom, and pan mode. The [View control panel](#) (Section 2.1.12) informs about this. Atom graphics is automatically restored when the Selection mode is activated or during copying or exporting the visualization.

## You already know what you are looking for

It is most convenient to first select the objects by [mouse click](#) or via the [select command](#) (Section 2.2) or in the [hierarchy window](#). (Section 2.3) . Now use the  view selected toolbutton. The camera is directed at the selected objects and, zoomed so that these objects approximately fill the 3D panel. Disturbing objects in the front are cut away. This can be combined with [depth cueing](#). (Section 2.1.13)

If you want to see all graphics again, use the  view all toolbutton.

## Selecting objects

To select a single object click with the left mouse button on it. This unselects all previously selected objects. Selected objects are displayed yellow.

To add an object to an existing selection, press **Shift** while clicking or click with the middle mouse button.

To remove an object from the selection while other selected objects remain selected, press **Ctrl** while clicking or click with the right mouse button.

To unselect all objects, double click on an object.

Clicking on ribbon graphics selects residues, clicking on atom graphics (wire, stick, ball&wire, space-filling plots) selects an atom, or, if there are alternate locations, a location. Clicking on a string graphic selects a whole chain. Clicking on label graphics, selects the corresponding spin label rotamer. Selecting label rotamers can be tricky if there are many of them, as the N-O midpoints are shown as transparent spheres and one often inadvertently clicks on a pale sphere in front of the intended opaque sphere.

If you need more sophisticated selection capabilities, use the [select command](#) (Section 2.2) or the [hierarchy window](#). (Section 2.3)



## Changing background color

A right mouse click in the panel, but outside any graphics object, opens the context menu. Navigate to the [background](#) line and select the desired color.

## Context menu

A right mouse click in the panel, but outside any graphics object, opens the context menu. You can change between the selection, rotation, zoom, and pan modes (alternative to tool buttons), change the color or transparency of the currently selected objects, detach the model panel as a separate [model window](#) (Section 2.1.16) and change the [background color](#).


## 2.1.16 Model window

### What can you do with the model window?

The model window displays the 3D graphics of the model as a scalable picture in a separate window. The model can be rotated, zoomed, and panned (shifted) by mouse click and drag in the panel. The view can also be changed using the [view control panel](#). (Section 2.1.12)

Structure [objects](#), (Section 1.5), such as residues or atoms, can be selected by mouse click and the [background color](#) can be selected among black, grey, and white by using a [context menu](#) (click with the right mouse button).


The 3D picture can be copied to the Windows clipboard (MS Windows only) or printed.

Most of this functionality is also available with a small, fixed-sized picture in the [model panel](#) of the [main window](#). (Section 2.1) The model window can be attached to the main window with the  attach toolbutton.


### Getting the best view on the model


#### You do not yet know what you are looking for


To explore the model in general, you may want to turn and zoom it. Actually the model remains fixed in space and the camera is moved around to simulate such rotation and zoom.

To rotate the model, activate the  rotate toolbutton. Now click left and draw for rotating the camera around the object, release to update the view indicator in the view control panel. (Section 2.1.12) Click left while pressing the shift key (or click with the middle button) and draw to rotate the vertical axis of the camera around the

view direction. The rotation mode can also be activated by pressing the 'r' key while the model window is the active window.


To zoom the model activate the  zoom toolbutton. Now Click left and draw down to enlarge the graphics (zoom in), click and draw up to shrink it (zoom out). Click left while pressing the shift key (or click with the middle button) to pan (move the graphics midpoint by drawing) while in Zoom mode. The zoom mode can also be activated by pressing the 'z' key while the model window is the active window.

If you want to shift the center of the 3D model activate the  pan toolbutton. Now mouse click and drag moves the midpoint of the graphics in the viewing plane. The rotation mode can also be activated by pressing the 'p' key while the model window is the active window.

When you are done, do not forget to reactivate the  selection mode. Otherwise you cannot select objects by click.

For very complex graphics (many graphics objects), a Matlab graphical user interface becomes very slow. MMM encounters this problem when a lot of atom graphics is displayed, in particular for spin label rotamers. In this case atom graphics is automatically reduced or switched off in the rotate, zoom, and pan mode. The [View control panel](#) (Section 2.1.12) informs about this. Atom graphics is automatically restored when the Selection mode is activated or during copying or exporting the visualization.

## You already know what you are looking for

It is most convenient to first select the objects by [mouse click](#) or via the [select command](#) (Section 2.2) or in the [hierarchy window](#). (Section 2.3) . Now use the  view selected toolbutton. The camera is directed at the selected objects and, zoomed so that these objects approximately fill the 3D panel. Disturbing objects in the front are cut away. This can be combined with [depth cueing](#). (Section 2.1.13)

If you want to see all graphics again, use the  view all toolbutton.

## Selecting objects

To select a single object click with the left mouse button on it. This unselects all previously selected objects. Selected objects are displayed yellow.

To add an object to an existing selection, press **Shift** while clicking or click with the middle mouse button.

To remove an object from the selection while other selected objects remain selected, press **Ctrl** while clicking or click with the right mouse button.

To unselect all objects, double click on an object.

Clicking on ribbon graphics selects residues, clicking on atom graphics (wire, stick, ball&wire, space-filling plots) selects an atom, or, if there are alternate locations, a location. Clicking on a string graphic selects a whole chain. Clicking on label graphics, selects the corresponding spin label rotamer. Selecting label rotamers can be tricky if there are many of them, as the N-O midpoints are shown as transparent spheres and one often inadvertently clicks on a pale sphere in front of the intended opaque sphere.

If you need more sophisticated selection capabilities, use the [select command](#) (Section 2.2) or the [hierarchy window](#). (Section 2.3)

## Changing background color

A right mouse click in the panel, but outside any graphics object, opens the context menu. Navigate to the `background` line and select the desired color.

## Context menu

A right mouse click in the panel, but outside any graphics object, opens the context menu. You can change between the selection, rotation, zoom, and pan modes (alternative to tool buttons), change the color or transparency of the currently selected objects, copy or print the graphics, detach the model panel as a separate [model window](#) and change the [background color](#). Copying to the clipboard works only under MS Windows. Note that for complex graphics, activation of the context menus may take some time.

The copy format is determined in Matlab general preferences, accessible via the `File/Preferences...` menu item of the Matlab main window. In the left list, go to the `Figure Copy Template` item and the `Copy Options` subitem. For MS Windows we recommend `Preserve information (metafile if possible)` and in all cases we recommend `Force white background`. We also recommend to activate the `Match figure screen size` checkbox.

## 2.1.17 Message board

### What is the message board for?

The message board is one central place where MMM informs you about the state of the program, about progress of computations, about query results, such as distances, angles, and dihedral angles, and about warnings and errors. The last 22 lines of message text are displayed. If you select objects, the [addresses](#) (Section 1.6) of the last 22 selected objects are shown.

A permanent record of all messages during a session can be written to a log file.

## Message logfile

When the log file checkbox is activated, MMM creates a file stamped with the current date and time and the extension `.log` in its temporary directory. From now on, until deactivation of the checkbox or leaving MMM, all output to the message board is also written to the log file.

Note that MMM checks every 30 days whether there are old files (older than 30 days) in its temporary directory and offers the user to delete them. Hence, you are strongly advised to copy temporary files that you actually want to keep to another directory immediately after an MMM session. This is particularly important if several users share the same installation of MMM.

## Clearing the message board

The message board is designed for single-line messages and MMM usually sends only single-line messages to it. However, no programmer is perfect, multi-line messages may occasionally arise and lead to overflow. Click the `Clear` button to get a clean message board again. This will not affect the log file, if log is activated.

## 2.2 Commands

In the following, commands are listed alphabetically.

### `attach`

Attaches the 3D [model window](#) (Section 2.1.16) as the [model panel](#) (Section 2.1.15) to the main window. If the 3D model is already displayed in the main window, the command has no effect. The command can be undone by `detach`.

### `bckg`

Changes the background color of the [model window](#) (Section 2.1.16) or [model panel](#). (Section 2.1.15). Possible arguments are `grey` (default), `white` (enforced with [depth cueing](#) (Section 2.1.13) or for copying to the clipboard) and `black`.

*Example:*

```
bckg white
```

### `camup`

Sets a defined camera up vector (rotation about viewing direction).

If called with one argument, the camera up direction can be `x|-x|y|-y|z|-z`, whereas `+x|+y|+z` are accepted for `x|y|z`. If called with three arguments, the three numbers define the camera up vector. Note that `+z` or `0 0 1` is the default camera up

vector. The current camera up vector is displayed in the **View** panel (dark green numbers). This command resets the camera zoom so that all displayed graphics is seen.

*Examples:*

```
camup -z % topple the model upside down w.r.t. normal viewing direction
```

```
camup 0 0.707 0.707 % turn the camera up vector to an arbitrary, but well defined direction
```

## color

Changes the color of [objects](#), (Section 1.5) , such as chains, residues, atoms, or atom locations. The objects can be selected via their systematic [address](#) (Section 1.6) or the color of all currently selected objects (address **\***) can be changed. The color argument can be an [SVG named color](#) or an RGB triple (red, green, and blue fraction between 0 and 1).

*Examples:*

```
color (B)123-138 limegreen
```

```
color "His" 0 1 1
```

```
color * blue
```

Color changes can be taken back by issuing an [undo](#) command (multiple undo possible).

## colorscheme

Applies a colorscheme to [objects](#), (Section 1.5) , such as chains, residues, atoms, or atom locations. The objects can be selected via their systematic [address](#) (Section 1.6) or the colorscheme of all currently selected objects (address **\***) can be changed. The second argument can be one of the following color schemes:

Color scheme **secondary** assigns standard metal colors to secondary structure elements in ribbon plots (helices: copper, sheets: steelblue, loops: gold).

Color scheme **chain** assigns rainbow colors from blue to red to the chains starting from chain A to the last chain.

Color scheme **sequence** assigns rainbow colors from blue to red within a chain starting from the N terminus to the C terminus.

Color scheme **Bfactor** assigns rainbow colors from blue to red according to the B factor, with the blue end corresponding to a coordinate r.m.s.d. of about 0.45 Å (B

factor 16) and the red end to about 1.5 Å (B factor 169). Color grade is proportional to the r.m.s.d., i.e. to the square root of the B factor.

Color scheme `Bfactor_tight` assigns rainbow colors from blue to red according to the B factor, with the blue end corresponding to the minimum B factor and the red end to the maximum B factor in the structure. Color grade is proportional to the r.m.s.d., i.e. to the square root of the B factor.

Color scheme `charge` assigns rainbow colors from blue to red according to charge. Dark blue: +2, blue +1, grey 0, red: -1, dark red: -2.

Color scheme `hydropathy` assigns rainbow colors from blue to red according to hydropathy, with the blue end corresponding to the most hydrophilic and the red end to the most hydrophobic residues.

Color scheme `helix_propensity` assigns rainbow colors from blue to red according to helix propensity, with the blue end corresponding to residues that tend to form an alpha helix and the red end to helix-breaking residues (Pro). Color grade is proportional to the square root of helix propensity.

Color scheme `ensemble` assigns rainbow colors from blue to red according to per residue  $C^{\alpha}$  r.m.s.d. of individual structures from the mean structure of an ensemble (PDB structures with several models, such as NMR structures, or ensembles generated by elastic network model based fitting from a template). The blue end corresponds to the minimum r.m.s.d. and the red end to the maximum r.m.s.d. of any residue in the structure. Color grade is proportional to the r.m.s.d. Note that *relative* variability within the structure is visualized by color, absolute variability has to be visualized by superimposing all structures. This colorscheme applies only at residue level to ribbon, coil,  $C^{\alpha}$  stick and  $C^{\alpha}$  wire models.

*Examples:*

```
colorscheme (:) Bfactor
```

Colorscheme changes cannot be undone.

## copy

Copies the current visualization in the detached [model window](#) (Section 2.1.16) to the clipboard. If the 3D model is attached, use the command sequence [detach](#), [copy](#), [attach](#).

The copy format is determined in Matlab general preferences, accessible via the `File/Preferences...` menu item of the Matlab main window. In the left list, go to the `Figure Copy Template` item and the `Copy Options` subitem. For MS Windows we recommend `Preserve information (metafile if possible)` and in all cases we recommend `Force white background`. We also recommend to activate the `Match figure screen size` checkbox.

## delete

Deletes an object together with the underlying data. Use the [hide](#) command to delete (or hide) only the graphical representation of the object.

Currently, this is supported only for surfaces. Future versions will allow to delete structures, coordinate sets, chains, residues, atoms, or atom locations.

*Example:*

```
delete $SAS:SAS_1 % deletes the solvent accessible surface (Section 2.1.3) with tag SAS_1
```

## detach

Detaches the 3D [model panel](#) (Section 2.1.15) as a separate [model window](#). (Section 2.1.16) . If the 3D model is already detached, the command has no effect. The command can be undone by the [attach](#) command. Loading a new PDB structure or model automatically reattaches the model window.

## domain

Defines a set of residues or whole secondary structure elements (helices, sheets, loops) as a domain. If the domain already exists, the command can be used with the + modifier to add residues or secondary structure elements to a domain. A domain is associated with a structure, not a chain, hence it can contain residues and cofactors from different chains. Individual atoms or whole chains cannot be added to a domain.

*Syntax:*

```
domain (+) address domain_name
```

Any other command can then use |domain\_name| as a valid address.

*Examples:*

```
domain 12-40,83-97 translocation % defines residues 12-40 and 83-97 in the current structure as a domain with name "translocation"
```

```
domain [NhaA]<H.IVp> substrate-binding % defines helix IVp in structure NhaA as substrate binding domain
```

```
domain + [NhaA]<H.IVc> substrate-binding % adds helix IVc in structure NhaA to the substrate binding domain
```

```
color [NhaA]|substrate-binding| crimson % changes color of the substrate binding domain ins structure NhaA
```



The command cannot be undone and does not need to be undone. You can always redefine a domain by using `domain` without the `+` mode.

## download

Download structure from PDB web server. The structure is not displayed, but display is initialized if it was not before. Internet access is required. A valid PDB identifier (case insensitive) must be provided.

*Usage:*

```
download pdb_identifier
```

## dssp

Reassigns secondary structure according to DSSP information, if present. Such information is created when reading a PDB file, if [DSSP](#) (Section 5.2) is installed. This requires application of `unlock` before and `lock` after, as secondary structure information is protected.

*Usage:*

```
dssp address
```

where *address* may pertain to a whole structure, to a chain, one model of a chain or a range of residues.

## echo

Displays the text argument in the Message Board of the Main Window. This is mainly intended for [scripts](#) (Section 3.9) to inform the script user of performed actions.

## helix

Defines a stretch of residues as a helix and assigns a name to this helix. The first argument specifies the structure and chain, the second argument the name, the third and fourth argument the range of residues. As an optional argument, the fifth argument specifies a special helix type. Currently, the only special type is TM (transmembrane).

This command is mainly intended for initialization scripts (see [Scripting](#)) (Section 3.9). MMM does not check for consistency of secondary structure definitions and does not update the display after redefinition. By default, secondary structure definitions are locked (see commands [lock](#) and [unlock](#)) and an ERROR message is displayed when the `helix` command is issued.

The hierarchy window is automatically closed, as current display becomes invalid.

*Example:*

`unlock` % unlocks secondary structure definition, unnecessary in initialization scripts

`undefine` % deletes previous secondary structure definitions

`helix (A) H1 10 30` % in chain A assign residues 10-30 as helix H1

`helix (A) TM1 45 66 TM` % in chain A assign residues 45-66 as transmembrane helix TM1

The command cannot be undone. Use [undefine](#) to undefine all helices.

## help

Without an argument, `help` displays the list of available commands in the [message board](#). (Section 2.1.17) . With a valid command as an argument, the command help page is opened at the corresponding line. With any other argument, an error message is displayed in the [message board](#). (Section 2.1.17)

## hide

Hides (deletes) the graphics of [objects](#), (Section 1.5) , such as chains, residues, atoms, or atom locations. The objects can be selected via their systematic [address](#) (Section 1.6) or all currently selected objects (address `*`) can be hidden.

*Example:*

`hide [2BHW] (B) 123-138` % hides residues 123-138 in chain B of structure 2BHW

The `hide` command does not hide spin label graphics (spheres for midpoint of the N-O bond and molecular frame). Use `show (C) 147 label_hide` to hide label graphics for residue 147 in chain C (sorry for the nonsensical syntax, see [show](#)).

## label

Attaches precomputed rotamers for a spin label at a given site (or sites). No undo is available. In many cases you might rather want to use the [Site scan](#) (Section 2.1.7) and [Attach precomputed labels](#) (Section 2.1.7) items of the `EPR` menu. Mainly intended for scripts.

*Usage:*

`label address type temperature`

where address must be provided and must address a residue or several residues. The other arguments are optional. `type` specifies the label and can be `MTSL` or `IA-PROXYL` with allowed synonyms `MTSSL` and `R1A` for `MTSL` and `IA1` and `IAP` for `IA-`

PROXYL. This argument defaults to `MTSL`. *temperature* can be `cryo` (175 K) or `ambient` (298 K) or any number specifying a temperature in Kelvin. It defaults to `cryo`. If you want to specify the temperature, you must specify the label.

Labeled residues are annotated with the keyword "spin label attached".

Please use command [rotamers](#) with the same arguments for computing the rotamers, unless this was done before in a [Site scan](#). (Section 2.1.7)

*Example:*

```
label (A)123 IA-PROXYL ambient % label residue 123 in chain A by iodoacetamido-  
PROXYL to simulate room temperature experiments
```

## libcomp

Systematic comparison of spin label position predictions by two different rotamer libraries, which can be used for comparing alternative libraries for the same label or for comparing different label types.

*Usage:*

```
libcomp address library1 library2 >output_file
```

where *address* is a mandatory argument and specifies the structure, chain, chain model, or residues for which the libraries are to be compared. Mandatory arguments *library1* and *library2* specify the libraries to be compared. The libraries must be present on the current Matlab path, usually they should be stored in MMM subdirectory `\rotamer_libraries`.

A text output file can be specified by the optional argument `>output_file`. If this argument is missing, output is directed to the message board and stored in the log file.

Arguments can also be specified in an input file. In this case, usage is:

```
libcomp <input_file> >output_file
```

The input file must contain a line

```
#LIBCOMP library1 library2
```

followed by one or more lines of MMM addresses for residues. It must finish with a line

```
#END
```

## libtest

Systematic and automatic test of rotamer computations against a set of experimental mean distances for known structures. No undo is available. It is strongly recommended to use this feature after an MMM restart and to close MMM after using it.

### Usage:

```
libtest test_data_file output_file library
```

where *test\_data\_file* is a mandatory argument and specifies the file with the test data set including extension (see below). The other arguments are optional.

*output\_file* specifies the file name for the result file. It defaults to the name of *test\_data\_file* appended by *\_prc* and the same extension. Unless a full pathname is provided, the file is written to the current Matlab directory. *library* defines the rotamer library file (including extension *.mat*). This argument defaults to the library specified in *test\_data\_file*. If you want to specify the library as an argument, you must specify the output file name.

Statistics of the test are written to a file based on file name *output\_file* with extension *.log*.

Note that all required PDB files for structures referred to in the test data file must be available on the Matlab path. The structures required for test data provided with MMM are stored in MMM subdirectory *tests*.

### Example:

```
libtest testdata_298K.dat test_xray_298K.dat R1A_298K_xray.mat % this may  
take half an hour or so
```

Currently provided test data sets are for MTSL only and are named *testdata\_298K.dat*, *outliers.dat*, and *rotamer\_test.dat*. The format is given in comment lines of these text (ASCII) files. They are found in subdirectory *rotamer\_libraries*. The corresponding PDB files are found in subdirectory *tests*.

Currently available rotamer libraries are of the form *R1A\_T\_type.mat*, where *T* is either *175K* or *298K* and *type* is *090619* (original MMM libraries for MTSL), *culled* (all rotamers with respect to chi1 and chi2 removed that were never observed in crystal structures), *xray* (chi1, chi2 probabilities as in the set of all reliable x-ray structures), *flat* (equal probabilities for all rotamers), *sezer12* (chi1, chi2 probabilities as in the MD computations of D. Sezer, J. H. Freed, B. Roux 2008 *J. Phys. Chem. B* 122:5755-5767, Table 10), or *warsh* (chi1, chi2 probabilities as in D. T. Warshaviak, L. Serbulea, K. N. Houk, W. L. Hubbell 2011 *J. Phys. Chem. B* 115:397-405). For more information, see G. Jeschke 2013 *Progr. Nucl. Magn. Reson.* submitted.

## lock

Locks the secondary structure definitions (this is the default state of MMM). The lock can be released by the command [unlock](#).

## loop

Defines a stretch of residues as a loop and assigns a name to this loop. The first argument specifies the structure and chain, the second argument the name, the third and fourth argument the range of residues.

This command is mainly intended for initialization scripts (see [Scripting](#)) (Section 3.9). MMM does not check for consistency of secondary structure definitions and does not update the display after redefinition. By default, secondary structure definitions are locked (see commands [lock](#) and [unlock](#)) and an ERROR message is displayed when the `loop` command is issued.

The hierarchy window is automatically closed, as current display becomes invalid.

*Example:*

`unlock` % unlocks secondary structure definition, unnecessary in initialization scripts

`undefine` % deletes previous secondary structure definitions

`loop (A) L1-2 31 35` % in chain A assign residues 10-30 as loop L1-2

The command cannot be undone. Use [undefine](#) to undefine all loops.

## mass

Displays the mass of the addressed object in the [message board](#). (Section 2.1.17)  
For structures, chains, and chain models water molecules are not counted into the mass. Use option `-water` to include water.

*Examples:*

`mass (A)` % displays mass of chain A

`mass (A) -water` % displays mass of chain A, including water molecules.

## motion


For a pair of structures that are related by a structural transition, display a pale green coil model of the first structure and crimson conical arrows that point to the second structure.

### Usage:

```
motion adr1 adr2
```

where *adr1* is the address of the moving structure and *adr2* the address of the target.

Note that the structures are actually superimposed and that this cannot be undone. Note also that the whole structure is superimposed and moved, even if the addresses are addresses of chains.

The conical arrows are associated with the moving structure. To remove them from the display, the moving structure must be hidden. Alternatively, any object can be selected and the  [Graphics mode toolbar](#) (Section 2.1.11) be clicked to access the [Graphics window](#). (Section 2.6) In this window the motion arrows can be selected from a popup menu and deactivated by unchecking the corresponding checkbox.

## ortho

Changes the projection mode for the 3D model to orthographic (the default). Can be undone by command [persp](#).

## pdbload

Load structure in PDB format from local disk. The structure is not displayed, but display is initialized if it was not before. The file must be on the current Matlab path, which is automatically the case, if it is stored in an MMM subdirectory (the subdirectory must have existed before starting MMM).

### Usage:

```
pdbload filename
```

We advise to provide a filename *with* extension. Otherwise MMM tries to find a file with extension `.pdb` and, if this cannot be found, a file with extension `.ent`. We also advise against using the "single-chain" files with extensions `.pdb1` etc. that can be found on the PDB Web server as "Biological Assembly 1" etc.

## persp

Changes the projection mode for the 3D model to perspective. Can be undone by command [ortho](#).

## plot

Plots a line connecting two atoms that are specified by addresses *adr1* and *adr2*.

*Usage:*

```
plot adr1 adr2 [[width] color]
```

The linewidth and color arguments are optional. Only SVG color names are accepted for the color argument. Width defaults to 6 (suitable for copying a full-screen figure), and color defaults to darkgreen.

*Examples:*

```
plot 123.CA 325.CA
```

## redo

Redoes the last undone command. This may or may not work (as in most programs). It should work certainly for simple things like color and transparency changes, it works approximately for moderately complex things as selections and graphics display and it does not work for complex things like secondary structure definitions.

## repack

Repacks (or creates) sidechains of native amino acids in a structure. Hetero amino acids will have missing sidechains and even backbone atoms may be lost. Consider applying the `replace` command before to substitute them by corresponding native amino acids. This requires that [SCWRL4](#) (Section 5.2) from the Dunbrack lab is installed and on the Matlab path.

*Usage:*

```
repack address
```

```
repack address crystal
```

where *address* is an [MMM object address](#) (Section 1.6) of a structure. The current version does nothing if lower-level objects are addressed. The optional argument `crystal` selects the `-#` option of SCWRL4, which calculates side-chain conformations within the crystal environment. This can be used to detect packing effects. The option requires that the original PDB file has a CRYST1 record.

## replace

Replaces non-standard residues such as selenocysteine or selenomethionine as well as spin-labels by their native amino acid equivalents (cysteine or methionine). Only the first location is kept (leading rotamer in case of spin labels). The command will hide replaced residues in the 3D plot (if they were displayed before). The user needs to redisplay them if required.



### Usage:

```
replace address hettype
```

where *address* is an [MMM object address](#) (Section 1.6) of a structure. When lower-level objects are addressed, only addressed residues are replaced. The address must correspond to residues in only one structure. The argument *hettype* specifies which residue type(s) are to be replaced. Residue types are three-character identifiers specified by PDB and the non-standard identifier **IA1** for iodoacetamido-PROXYL. In a list of residue type, the individual types need to be separated by colons (:). Only a colon in place of *hettype* leads to replacement of all non-standard residues defined in MMM.

### Examples:

```
replace [3K2R] V1A % replaces the two imidazole-type spin labels in T4 lysozyme structure 3K2R by cysteins
```

```
replace [2X79] MSE:R1A % replaces in structure 2X79 all selenomethionines by methionines and removes all MTS-spin labels attached before (replaced by cysteine)
```

```
replace [2X79]20-100 : % replace all non-standard residues known to MMM among residues 20 to 100 of structure 2X79
```

## report

Writes information on spin-labels and distance distributions to a report file. If the file exists, the information is appended. Useful mainly in command scripts.

### Usage:

```
report filename information [...]
```

where *filename* is the name of the report file (plain text) that will be written to the *current* Matlab directory (unless a pathname is provided) and *information* specifies the type of information to be reported. Additional arguments depend on the type of information requested:

```
report filename label address extent
```

where *address* is an [MMM residue address](#) (Section 1.6) and *extent* is an optional argument that is either **brief** (default) or **full**. In **brief** mode, only mean position and r.m.s.d. are reported.

```
report filename distance address1 address2 extent
```

where *address1* and *address2* are [MMM residue addresses](#) (Section 1.6) and *extent* is an optional argument that is either **brief** (default) or **full**. In **brief** mode, only mean distance and r.m.s.d. are reported, in full mode the whole distribution is written out.

### Examples:

`report LHCII_labels.txt label [2BHW]123` % writes mean position and r.m.s.d of the label at residue 123 of structure [2BHW] to the file LHCII\_labels.txt. The label must have been computed with `rotamers`, but not necessarily attached. The labels must have been computed with `rotamers`, but not necessarily attached.

## rotamers

Compute rotamers for a spin label at a given site (or sites). In many cases you might rather want to use the [Site scan](#) (Section 2.1.7) items of the `EPR` menu, which allows for more input and output options. Mainly intended for scripts.

### Usage:

```
rotamers address type temperature
```

where `address` must be provided and must address a residue or several residues. The other arguments are optional. `type` specifies the label and can be `MTSL` or `IA-PROXYL` with allowed synonyms `MTSSL` and `R1A` for `MTSL` and `IA1` and `IAP` for `IA-PROXYL`. This argument defaults to `MTSL`. `temperature` can be `cryo` (175 K) or `ambient` (298 K) or any number specifying a temperature in Kelvin. It defaults to `ambient`. If you want to specify the temperature, you must specify the label.

Basic rotamer statistics information (number of significantly populated rotamers, partition function, r.m.s.d. of N-O midpoint coordinates) is stored in the residue annotations with the keyword "rotamers computed". The [Site scan](#) (Section 2.1.7) menu item allows for more complete analysis.

The spin labels are only computed, not immediately attached. Please use the command [label](#) with the same arguments to actually attach them.

### Example:

```
rotamers (A)123 IA-PROXYL ambient % label residue 123 in chain A by  
iodoacetamido-PROXYL to simulate room temperature experiments
```

## SAS

Creates and displays the solvent accessible surface for a single object, which must be a structure, chain, coordinate set of a chain, or residue. This requires that [MSMS](#) (Section 5.2) by Michel Sanner is installed.

### Usage:

```
SAS address tag (radius)
```

where *address* is an [MMM object address](#), (Section 1.6) *tag* is the tag by which the surface can be accessed later, and *radius* is an optional probe radius in units of Å. The probe radius defaults to 1.5 Å, which is Sanner's suggested value for water.

The surface is originally displayed with aquamarine color and 50% transparency.

*Usage:*

```
SAS ! native % water accessible surface with tag "native" for the current structure (!)
```

```
color $SAS:native cornflowerblue % change color to cornflowerblue
```

Note that a surface address always begins with \$, followed by the surface type (here SAS).

## scopy

Makes a copy of a structure with a new structure tag. The copied structure becomes the current structure.

*Usage:*

```
scopy address tag
```

where *address* is an [MMM object address](#) (Section 1.6) of a structure and *tag* is the tag by which the new structure can be accessed later.

## select

Selects [objects](#), (Section 1.5), such as structures, chains, domains, secondary structure elements, residues, atoms, or atom locations via their systematic [address](#). (Section 1.6)

*Examples:*

```
select [2BHW] (B) 123,145,217 % select residues 123, 145, and 217 in chain B of structure 2BHW
```

```
select "His" % select all histidine residues in the current structure and chain
```

```
select (A) % select the helix with name H3 in chain A of the current structure
```

Selections of already selected objects are ignored, a warning is issued in the [message board](#). (Section 2.1.17). Selections can be taken back by issuing an [unselect](#) command to the same objects. Note that undo of a selection may have unintended consequences after a double selection- the previously doubly selected object will also be unselected.

The command

```
select *
```

seems to be nothing, since the asterisk `*` stands for all objects that are currently selected. However, it is useful for displaying the current selection in the message board.

## sheet

Defines a stretch of residues as a strand (part of a sheet) and assigns a name to this strand. The first argument specifies the structure and chain, the second argument the name, the third and fourth argument the range of residues. As an optional argument, the fifth argument specifies a special strand type. Currently, the only special type is TM (transmembrane).

This command is mainly intended for initialization scripts (see [Scripting](#)) (Section 3.9). MMM does not check for consistency of secondary structure definitions and does not update the display after redefinition. By default, secondary structure definitions are locked (see commands [lock](#) and [unlock](#)) and an ERROR message is displayed when the `sheet` command is issued.

The hierarchy window is automatically closed, as current display becomes invalid.

*Example:*

```
unlock % unlocks secondary structure definition, unnecessary in initialization scripts
```

```
undefine % deletes previous secondary structure definitions
```

```
sheet (A) SI 36 49 % in chain A assign residues 36-49 as strand SI
```

```
sheet (A) TMS1 58 76 TM % in chain A assign residues 58-76 as a transmembrane strand TMS1
```

The command cannot be undone. Use [undefine](#) to undefine all sheets.

## show

Displays graphics for [objects](#), (Section 1.5), such as chains, residues, atoms, or atom locations. The objects can be selected via their systematic [address](#) (Section 1.6) or graphics can be shown for all currently selected objects (address `*`) at once. The second argument can be one of the following graphics modes:

Graphics mode `wire` applies to atoms and locations and displays a wire plot with lines for the bonds and dots for the atoms.

Graphics mode `stick` applies to atoms and locations and displays a stick plot with cylinders for bonds and spheres with the same radius for atoms.

Graphics mode `ball&stick` applies to atoms and locations and displays sticks for the bonds and slightly larger spheres for atoms.

Graphics mode `space-filling` applies to atoms and locations and displays a space-filling model (spheres with the van-der-Waals radius).

Graphics mode `CaWire` applies to chains and residues and displays a wire model of the C<sup>α</sup> trace. Cofactors are shown in `wire` mode. For DNA and RNA chains, the wire model connects the midpoints of the C3'-C4' bond of the ribose or deoxyribose.

Graphics mode `CaStick` applies to chains and residues and displays a stick model of the C<sup>α</sup> trace. Cofactors are shown in `stick` mode. For DNA and RNA chains, the stick model connects the midpoints of the C3'-C4' bond of the ribose or deoxyribose.

Graphics mode `ribbon` applies to chains and residues and displays a ribbon model of amino acid residues. Cofactors are not shown. For DNA or RNA, the backbones are shown as ribbons, the connection between each midpoint of the C3'-C4' bond of the ribose/deoxyribose and the atom of the base bound to the sugar is shown as a tube, and the plane of the base as a filled polygon. Nucleotides are color-coded: G is lightgreen, C is lightcoral, T/U is gold, and A is mediumpurple.

Graphics mode `coil` applies to chains and residues and displays a coil model of the backbone. Individual residues can be selected by clicking in coil mode. Cofactors are not shown. Coil mode also applies to DNA/RNA.

Graphics mode `string` looks similar as `coil`, but creates a single graphics object per chain. This allows for selecting chains by a single click, but does not allow for selecting residues by clicking into the graphics. Cofactors are not shown. String mode also applies to DNA/RNA.

Graphics mode `water` displays water molecules, which are suppressed in all other graphics modes.

The following graphics modes exist only if spin labels are defined and apply to residues:

Graphics mode `label` shows the midpoint of the N-O bonds of all label rotamers as small spheres. Relative sphere volume corresponds to relative label population, the total sphere volume is selected so that both very narrow and very broad rotamer distributions can be well seen. If a single rotamer makes up at least 99.5% of the total population, its sphere radius corresponds to the half of the sum of the van-der-Waals-radii of nitrogen and oxygen plus the N-O bond length (full extension of the N-O group). In all other cases spheres are smaller.

Graphics mode `label_frame` shows the midpoint of the N-O bonds of all label rotamers as spheres and the molecular frame with a red x axis, green y axis, and a blue z axis. This gives an impression on orientational distribution.

Graphics mode `label_hide` hides label graphics without affecting other graphics of this residue.

*Examples:*

`show (A)123-136 ball&stick %` displays residues 123-136 in ball & stick mode

`show (A) coil %` displays a coil model of chain A

`show (A) "Gly" space-filling %` displays all glycine residues of chain A in space-filling mode

Graphics are first displayed in a default color scheme, atoms and locations by element color ribbons by the secondary structure color scheme and other graphics by their uniform color. Use command [color](#) to change color afterwards. If there are alternative locations for an atom, transparency is assigned by population. Use command [transparency](#) to change transparency.

## symmetry

Checks whether selected objects define a symmetry or pseudosymmetry axis. If so, transforms the structure to a frame where this axis is the z axis and where the center of the structure is on this axis.

The number of selected objects determines the multiplicity of the axis. All selected objects must belong to the same structure, this structure is transformed. All selected objects must have the same number of atoms (sorry).

This command cannot be undone and it hides the transformed structure. To preserve the display properties of the structure, please use the [Edit/Symmetry](#) (Section 2.1.2) menu item, which cares about automatic redisplay. The command version is mainly intended for scripts.

## synonym

Assigns a synonym to a structure or chain name. The first argument is a structure or chain [address](#), (Section 1.6) the second argument the synonym for the structure or chain name.

*Example:*

`synonym [2BHW] LHCII`

`show [LHCII] coil %` displays a coil model of structure 2BHW

## transparency

Assigns a transparency value to [objects](#), (*Section 1.5*), such as chains, residues, atoms, or atom locations. The objects can be selected via their systematic [address](#) (*Section 1.6*) or transparency can be assigned to all currently selected objects (address *\**) at once. The second argument is the opaqueness between 0 (fully transparent) and 1 (fully opaque). A negative value is interpreted as request to assign transparency according to population of alternate locations (the default mode).

*Examples:*

`transparency (C)147 0.5 %` makes residue 147 in chain C semi-transparent

`transparency (A)"Asn" -1 %` assigns default transparency (by population of alternate locations) to all asparagine residues in chain (A)

## undo

Undoes the last issued command. This may or may not work (as in most programs). It should work certainly for simple things like color and transparency changes, it works approximately for moderately complex things as selections and graphics display and it does not work for complex things like secondary structure definitions.

## undefine

This command is mainly intended for initialization scripts (see [Scripting](#)) (*Section 3.9*). Undefines secondary structure assignments (helix, strand/sheet, and loop definitions).


Without argument, the command operates on the current chain, as indicated in the title of the main window. If a structure address is given as argument, the command operates on all chains of this structure. If a chain address is given, it operates on this chain.

The `undefine` command is usually blocked (error is displayed in the [message board](#)) (*Section 2.1.17*), since secondary structure definitions are locked by default. Use the [unlock](#) command before, if you must do that outside an initialization script, and do not forget to [lock](#) again. Only the list of helices, strands, and loops is deleted, secondary structure assignment of residues is kept. The hierarchy window is automatically closed, as current display becomes invalid.

## unlock

Unlocks the secondary structure definitions, which are locked by default in MMM. The lock can be imposed again by the command [lock](#). Use `unlock` only if you understand the consequences of modifying secondary structure definitions in the



current state of the model. Make a backup copy of the model (  save button) before you do that. The same function unlocks the protection against using alternative rotamer libraries or alternative parameters in elastic network modeling of structural transitions.

## unselect

Unselects [objects](#), (Section 1.5) , such as structures, chains, domains, secondary structure elements, residues, atoms, or atom locations via their systematic [address](#), (Section 1.6) if they were previously selected.

*Example:*

```
unselect [2BHW] (B)145 % unselects residue 145 in chain B of structure 2BHW
```

If the addressed objects were not selected, the command has no effect. No warning is issued.

## view

Sets a defined viewing direction.

If called with one argument, the viewing direction can be  $x|-x|y|-y|z|-z$ , whereas  $+x|+y|+z$  are accepted for  $x|y|z$ . If called with three arguments, the three numbers define a viewing vector. If called with six arguments, the six numbers define a viewing vector and a camera up vector. Note that the current viewing vector (dark blue numbers) and camera up vector (dark green numbers) are displayed in the [View](#) panel. This command resets the camera zoom so that all displayed graphics is seen.

*Examples:*

```
view -z % look at the model from the -z direction
```

```
view 0.5 0.5 0.707 % view at the model from an arbitrary, but well defined direction
```

```
view 0.5 0.5 0.707 0 0 -1 % view at the model from an arbitrary, but well defined direction and upside down compared to usual camera up vector
```

## zoom

Zooms in to selected objects or zooms out to display the whole model.

*Examples:*

```
zoom in % define the camera view so that the select objects fill the model window
```

```
zoom out % define the camera view so that all displayed graphics is shown
```

The `zoom in` command may produce funny results if the 3D model is shown in the [model panel](#) (Section 2.1.15) (attached state), as graphics is not clipped to the panel.

## 2.3 Hierarchy window

### What can you do in the hierarchy window?

The hierarchy window is mainly used for obtaining an overview of a structure and for selecting objects. The structures within an MMM model (project), the chains within each structure and their sequences, the presence of several coordinates sets (models) for a chain, the presence and nature of cofactors, the atom identifiers for each residue or cofactor, and the presence of alternate locations for atoms can all be easily recognized. Furthermore, secondary structure definitions are displayed with the sequence and secondary structure assignments of residues can be changed, although such changes are discouraged (see below).

### Inspection of the structure

The hierarchy window contains popup menus for structures, chains, coordinate sets, and cofactors and listboxes for atoms and atom locations. When you preselect a new structure or new chain in the corresponding popup menus the [sequence display](#) this structure or chain are made current. The current structure and chain are also displayed in the title of the main window. All popup menus on lower levels are updated to reflect this choice, while the listboxes are not updated. Preselection in the popup menu is for inspection, it does yet not add the structure or chain to the list of selected objects (see [Selection](#)).

To display the atoms of a residue or cofactor, this residue or cofactor must be added to the selected objects (see [Selection](#)). You can always remove it again.

Pushbuttons near the `Structure` popup menu provide access to the PDB homepage for this structure, the PDBwiki entry, and the electron density server (EDS) page. This works only for structures with a valid PDB identifier. Electron density information may not exist for older crystal structures and does not exist for NMR structures.

A pushbutton next to the `Chain` popup menu provides access to the UniProt sequence homepage for this chain, if a valid database reference entry was given in the PDB file on loading. Note that the sequence used in structure determination may be modified.

### Sequence inspection

The sequence is displayed in single-letter code format with vertical lines at residues 0, 10, 20, etc. and 50 residues per line. Non-standard residues are displayed as a question mark. When you click on the question mark, the PDB residue type is identified in the `Information` line on the bottom and in the title of the `Atoms` listbox.

Models created from properly formatted PDB files with MMM 2010 or later contain information on the type of sequence (peptide, DNA, RNA, unknown polymer). This information is shown in the title of the sequence panel. Panel background color is used to alert the user. For DNA, nucleotides are represented by upper-case letters, for RNA by lower-case letters.

If secondary structure of a peptide sequence is defined, it is displayed schematically above the single-letter codes. Thin green sawtooth lines correspond to loops, thick orange wavy lines to helices, and thick blueish straight lines to strands (sheets).

Residues that exist in the original sequence, but are not defined in the atomic or coarse-grained structure, are shown as grey letters and do not respond to clicking. Residues that are defined in the actual structure, but for which atoms are missing (usually side group atoms) are shown with bold dark green single-letter codes. When clicking on these single-letter codes, you can see in the **Atoms** listbox which atoms are defined in the structure.

Information on hydropathy, helix propensity, charge for peptide chains can be displayed as background color by activating the corresponding radiobuttons. For hydropathy and helix propensity the color code is a blue (low) to yellow (high) scale. In other words, hydrophobic residues are yellow and helix breakers (Gly, much more Pro) are blue. For charge, a single positive charge is coded light blue, a single negative charge light red. Double charges (not yet implemented) correspond to darker color shades. There is no effect of these radiobuttons on DNA or RNA sequence display.

DSSP secondary structure assignment (if available, see [third-party software](#)) (*Section 5.2*) can be visualized by color coding of the residue letters. In this color coding  $\alpha$ -helices (DSSP code H) are copper (like the wavy helix line), unassigned residues limegreen (like the loop line), and extended strand residues (DSSP code E) steelblue (like the strand bar). Other types are B (isolated  $\beta$ -bridge, lightskyblue), G ( $3_{10}$ -helix, sandybrown), I ( $\pi$ -helix orangered), T (turn, darkgreen), and S (bend, forestgreen). These colors roughly match the corresponding simplified secondary structure assignment, which knows only helix, strand (sheet), and loop. Slight differences with even automatic secondary structure assignment are possible, as MMM repairs small gaps or extends secondary structure elements if the backbone dihedrals of neighboring residues are very close to canonical values for the secondary structure.

## Selection of objects

Structures, chains, coordinate sets, cofactors, atoms, and locations can be selected by the corresponding ! and + buttons and unselected by the - buttons. The ! unselects all previously selected objects and makes the object preselected in the popup menu or listbox the only selected object. The + button adds this object to the existing selection and the - button removes it from the existing selection.

All cofactors can be added at once by the ++ button and removed at once by the -- button.

Residues are selected by clicking in the sequence display. Single residues are selected by clicking on their single-letter code. The left mouse button selects exclusively the clicked residue, corresponding to a ! button, the middle mouse button adds the residue to the selection, corresponding to the + button, and the right mouse button unselects the residue, corresponding to the - button. Instead of the middle mouse button you can also use the left button while pressing the `Shift` key, instead of the right mouse button, the left mouse button while pressing the `Ctrl` key.

The current residue is highlighted in bright red color, while other selected residues are highlighted in a slightly darker shade of red.

All residues in a secondary structure element can be selected, added to the selection, or removed from the selection by clicking on the corresponding graphic element in the sequence display. Green sawtooth lines denote loops, orange wavy lines helices, and blueish straight lines strands (sheets). The last residue of the last selected secondary structure element becomes the current residue.

Double clicking in the sequence panel, but outside the sequence, unselects everything.

For the last selected residue or cofactor, all atoms that are specified in the structure are listed in the `Atoms` listbox. The title of this listbox is updated to display the current residue or cofactor. Likewise, if alternate locations exist for the last selected atom, they are displayed in the `Locations` listbox and the title of this listbox is updated.

Residue selection in the hierarchy window is slightly inconvenient when you want to select by number of residue and very inconvenient when you want to select ranges of residues that are not whole secondary structure elements or residues by amino acid type. In such cases you better select by [command](#). (*Section 2.2*)

## Limitations

Some PDB structures contain several residues with the same number, distinguished by "insertion codes". Somebody meant well when introducing this, but in retrospect it is a very bad idea that creates a lot of confusion. To avoid mismatch between residue numbers in the sequence display and in the PDB file, MMM *excludes* all residues with an insertion code from the sequence. Such residues cannot be selected by clicking in the sequence display, but they are accessible by clicking on 3D graphics or via the [select command](#). (*Section 2.2*) Use `select [1N8S] (A) 30B` to select residue 30 with insertion code B in chain A of structure 1N8S.\

It is also possible to have residue number zeros or even negative residue numbers in PDB files. In that case, MMM rennumbers residues starting at 1.

If a chain contains one or more residues with an insertion code or residues with numbers smaller than one, the displayed sequence is built from scratch, i.e., missing residues at the N terminus are represented by question marks (identity unknown) and missing residues at the C terminus are not shown at all.

The SEQRES records of PDB files do not always match the residue numbering in the ATOM records. MMM checks for such mismatches and if they are found, the displayed sequence is also built from scratch. Again, structure 1N8S (now chain C, which is actually the C-terminal part of chain A with a large number of residues missing in between the end of chain A and the beginning of chain C) is a good example.

The type of helix of DNA/RNA sequences is not recognized.

## 2.4 Color selection

### What can you do with the color selection window?

This window allows to assign a single [color](#) or a [color scheme](#) to all currently selected objects. If you do not want to reassign color, use the [Cancel](#) button. You can assign color without first selecting objects by using the [color command](#) (Section 2.2) in the [command line](#). (Section 2.1.14)

### Choosing a color

MMM supports SVG color names (an internet browser standard) and freely selectable RGB colors. With SVG color names it is easier to obtain reproducible graphics and consistent look. Freely selectable RGB colors allow for finer shading.

Colors can be selected from the SVG colors listbox or from a palette that appears on clicking of the [Palette](#) button. The title of the SVG listbox shows the currently selected color on white background. Likewise the label of the [Palette](#) button shows the color currently selected in the palette.

For assigning a single color to all selected objects there are two modes. In the [color name \(SVG\)](#) mode colors selected from the palette are approximated by the SVG color that is closest in RGB space. Please check the result by comparing the colors of the [Palette](#) button label and the [SVG colors](#) listbox title. If you are unsatisfied with the approximation, change the [Mode](#) to [palette](#).

### Color schemes

Predefined color schemes assign colors to objects according to some property of these objects, such as secondary structure, position in the sequence, or charge. They can be selected by the [Mode](#) radiobuttons.

For a description of the available color schemes see the [colorscheme command](#). (Section 2.2)

## 2.5 Transparency selection

### What can you do with the transparency selection window?

This window allows to assign a [transparency value](#) or transparency choice by population (occupancy) to all currently selected objects. If you do not want to reassign transparency, use the `Cancel` button. You can assign transparency without first selecting objects by using the [transparency command](#) (Section 2.2) in the [command line](#). (Section 2.1.14)

### Transparency values

If the `Mode` radiobutton `by value` is activated, all selected objects are assigned the same transparency, which is given by the `Value` in the edit field. This value is between 0 (fully transparent, objects become invisible, but are not deleted) and 1 (fully opaque, default value). The edit field does not accept values outside this range.

If the `Mode` radiobutton `by population` is activated, each object is assigned an individual transparency based on population. This value is 1 (opaque), unless there are alternate locations for an atom or the observed electron density was too weak to assume full occupation at the specified coordinates.

### Conflict with depth cueing

Note that the use of transparency in visualization does not mix well with MMM's [emulated depth cueing](#). (Section 2.1.13) In depth cueing transparency is used to fade objects in the background. If you activate depth cueing, a transparent object may appear too far in the background.

## 2.6 Graphics mode

### What can you do with the graphics mode window?

This window allows to determine the appearance of the currently selected objects. If the selection contains [objects](#) (Section 1.5) on different hierarchy levels, individual graphics modes can be specified for these levels. This window also allows to display objects that are hidden by default, such as water molecules or electron spin positions of spin labels (`Auxiliary graphics`).

If you do not want to reassign graphics modes, use the `Cancel` button. You can assign graphics modes without first selecting objects by using the [show command](#) (Section 2.2) in the [command line](#). (Section 2.1.14) . The description of this command also explains the [graphics modes](#). (Section 2.2)

Specification of `None` as the graphics mode for a hierarchy level hides the selected objects on this level. This can also be achieved by the [hide command](#). (Section 2.2)

## Surface display control

If surfaces, e.g. density surfaces or solvent accessible surfaces (SAS) are defined, this window allows to control their display. The `Surfaces` popup menu then contains all available surfaces in a tag format `type:tag`, where `type` defines the type of surface, e.g. `density` or `SAS` and `tag` is the tag given by the user when importing or generating the underlying data (importing a density cube or computing the SAS).

When an item in this popup menu is selected, the state of the `active` checkbox right from the popup menu indicates the current display status. Activate this checkbox to display or deactivate it to hide the surface. Surfaces are redisplayed (fast) without computation.

If no surfaces are defined, the popup menu and checkbox are disabled.

## 2.7 Report editor

### What is the report editor?

The report editor is mainly intended for displaying short text files created by MMM in a temporary folder to report results of computations. It allows to insert comments and to save the files. It can also be used as a small notepad. Furthermore it can be used to inspect short plain text (ASCII) files generated by other programs. For long files, such as PDB files, you might rather want to use the Matlab editor or another standalone editor.

### Limitations

The report editor expects files with less than 80 characters/line. Other text files are displayed, but cannot be saved.

### Load

Any plain text (ASCII) file can be loaded via the `Load` button.

### Save as

Text files with less than 80 characters per line can be saved via the `Save as` button. Reports generated by MMM in the temporary folder can thus be saved to another folder if you want to keep them. Files older than 30 days are automatically cleared from the temporary folder.

### Closing the editor



Clicking on the **OK** button closes the editor. Unless the editor is in read-only mode, there is a query whether you want to save changes (even if you did not make any changes).

## 3 Methods

### 3.1 Spin labeling and site scans

#### Spin labeling concept

##### General concept

MMM describes spin labels by a set of alternative conformations, rotamers, which can be attached without serious clashes with atoms of other residues or cofactors. The individual rotamers are assigned Boltzmann populations corresponding to an estimate of the sum of their internal energy and interaction energy with the protein. All simulations of experiments on spin labels are then based on the population weighted average over the ensemble of rotamers. Since version 2010 clashes with water are automatically disregarded.

For further information, please read the [publications](#) (Section 5.3) on this approach. Full-text links:


[Rotamer library and attachment](#)

[Site scans](#)

Spin labels can also be [localized](#) (Section 3.3) with respect to a reference structure or with respect to each other ([networks](#)). (Section 3.4)

##### Rotamer libraries

The conformations allowed for an isolated spin label and their relative energies are stored in rotamer libraries. Currently such libraries exist for methanethiosulfonate spin label (MTSL) and iodoacetamido-PROXYL (IA-PROXYL) for cryogenic temperature and ambient temperature. As the cryogenic temperature, 175 K was selected, close enough to the glass transition temperatures of water/glycerol mixtures as a solvent for soluble proteins and lipid bilayers. Temperature scaling of the Boltzmann distribution is possible, but with the system not being ergodic, is only an approximation and should be restricted to small temperature intervals.

Some further libraries, based on the same rotamers but with different relative probabilities of these rotamers, have been created for tests. These libraries can be used after unlocking advanced features with the  Lock/Unlock (toggle button)).


## Interaction with the protein

Interaction with atoms of the protein is estimated as dominated by packing effects (only pairwise Lenard-Jones potentials for label and protein atoms) and uses a distance cutoff to speed up computation. The protein backbone and conformations of native side groups are currently considered as static.

### Why a site scan?


MMM has a two-step process of attaching labels. First spin label rotamers are computed for a set of selected residues or selected chains and after that they are attached to the model structure.

The reason is that rotamers computed in site scans do not interfere with each other, but they do interfere with labels that are already attached. A one-step labeling procedure would thus require to attach all desired labels at once.


In a [site scan](#) rotamer distributions are computed and stored and a HTML report of favourable labeling sites is prepared, but the structure remains unchanged. If you later select residues that were already computed in a site scan and click the labeling toolbutton  the rotamers are attached at once (in seconds). If you have prepared and stored a model with a complete site scan, you can always reload it and attach the desired labels. Furthermore, you can attach rotamers from a site scan consecutively without any interference (the distributions were all computed for the virgin structure).

The main results from the site scan (number of significantly populated rotamers, partition function, r.m.s.d. of the N-O midpoint coordinate) are stored as residue annotations with the keyword `rotamers computed`. Hence, you can find all residues for which rotamers were computed via the [keyword finder](#) (Section 1.11) and can obtain these statistics via the `Show annotation` button.

## Attach precomputed labels

Select the residues for labeling by any [selection method](#), (Section 1.8), for instance in the [hierarchy window](#), (Section 2.3), by clicking on the structure in the [model window](#) (Section 2.1.16) or by the [select command](#). (Section 2.2) Now click the labeling toolbutton . A window `Set labeling conditions` opens. Pick the `Label` from the corresponding listbox and select the temperature with the `cryogenic` or `ambient` radiobutton. Direct temperature input in the corresponding edit field is also possible, but should be used with care and only if you understand the concept of rotamer computations well.

MMM now checks for each selected residue if rotamer computations for that label and that temperature were already done. If so, the label is attached, if not, a warning message is shown in the [message board](#) (Section 2.1.17) of the [main window](#). (Section 2.1)

Labels will not be displayed automatically. Use the  graphics toolbutton for display.

Label attachment leads to an automatic residue annotation with the keyword `spin label attached`. Hence, you can find all labeled residues via the [keyword finder](#) (Section 1.11) and can obtain the corresponding rotamer statistics via the `Show annotation` button.

## Spin labeling site scan

### Is your protein structure suitable for labeling?

A protein structure downloaded from PDB may need some modification before labeling to get the most realistic results. First, mutations to non-native amino acids, such as selenocysteine (CSE) or selenomethionine (MSE), may have been used to solve the phase problem in x-ray crystallography. The protein that is used in EPR experiments most likely has the native amino acids instead. You can fix this with the [Build/Replace non-standard residues](#) (Section 2.1.3) menu item. This replaces selenium by sulfur in MSE and CSE and reduces the Se-C bond lengths to the typical values of S-C bond lengths. The residues are renamed to MET or CYS. The same function can also replace existing spin labels, if you use a crystal structure that has already spin labels attached. This is required as MMM labels only native amino acids. Spin labels are replaced by cysteine.

Second, some amino acids may have missing sidechains. Although it is proper in crystallography not to specify sidechains if the corresponding electron density is missing, just ignoring these sidechains is not the best choice for subsequent spin label rotamer computations. The problem can be identified in sequence display in the [hierarchy window](#), (Section 2.3), where residues with missing atoms are displayed by green letters, or by the [Edit/Find by keyword](#) (Section 2.1.2) menu item (keyword: missing atoms). You can (try to) repair all sidechains with the [Build/Repair sidechains](#) (Section 2.1.3) menu item. This requires that you have the third-party software [SCWRL4](#) (Section 5.2) installed, which we strongly recommend. All residues will be repaired for which at least all backbone atoms exist. The function reports in the Message Board of the main window whether all residues could be repaired. If not, you should check if incomplete residues are spatially close to your spin labels.

Third, your protein may have been mutated to make it amenable to spin labeling. Typically, this involves removal of all or most native cysteines. You should perform the same mutations on the structural model in MMM, using the [Biochemistry/Mutation](#) (Section 2.1.8) menu item. This also requires that you have the third-party software [SCWRL4](#) (Section 5.2) installed and that all backbone atoms of the mutated residues are specified in the structure.

### Preparation

A site scan pertains to selected chains or coordinate sets of chains (in NMR structures with several chain models) or to selected residues. If nothing is selected,

the current chain (displayed in the title of the main window) is labelled. To scan a chain or several chains, select them by any [selection method](#), (Section 1.8) , for instance in the [hierarchy window](#), (Section 2.3) , by clicking on the structure in the [model window](#) (Section 2.1.16) or by the [select command](#). (Section 2.2) Now select the `EPR/ Site scan/ selected chains or current chain` menu item. A `Site scan setup` window opens that allows you to determine the labeling pattern. If there are alternative chain models (coordinate sets) for a selected chain, you are previously asked to determine which one should be labeled.

To scan selected residues, select them by any [selection method](#), (Section 1.8) , for instance in the [hierarchy window](#), (Section 2.3) , by clicking on the structure in the [model window](#) (Section 2.1.16) or by the [select command](#). (Section 2.2) Now select the `EPR/ Site scan/ selected residues` menu item. A `Site scan setup` window opens that allows you to exclude certain residue types from the scan (this is handy for domain scans, if you are reluctant to label charged residues).

### Site scan setup

Here you can select `Residue types` that you want to label or want to exclude from labeling. Each of the 20 native amino acids can be selected or unselected by checkbox. There are also two predefined labeling patterns. The `conservative` pattern selects native Cys, Ile, Leu, Met, Ser, Thr, and Val residues, i.e., those residue types for which mutation to Cys is least likely to affect structure and function. The pattern `all` selects all amino acids, including non-native amino acids. For chain scans, `conservative` is the default choice, for selected residue scans `all` is the default choice.

If your system has a symmetry axis or pseudo-symmetry axis and you have applied the `Symmetry` command in the `Edit` menu before, the `z axis is a unique axis`. The same applies if you have defined a bilayer with the `z axis` being the bilayer normal. In these cases you may want to activate the `z axis is a unique axis` checkbox, as spatial distribution of the label in `z` direction is of interest.

After a site scan, rotamer distributions of label pairs can be used for `Distance analysis`. Such analysis can suggest label pairs that are suitable for CW EPR or DEER distance measurements. You can select by radiobuttons whether you want distance analysis to be performed within the same chain (`intrachain all`) for residue pairs in different chains (`interchain all`) only for residues with the same number in different chains (`interchain equivalent`, useful for homooligomers) or if you want to create homooligomer distance distributions by symmetry.

With the latter option you need to scan only one of the equivalent chains in a homooligomer. This requires that the `z axis` is the symmetry axis and that you know how many protein molecules make up the homooligomer (`Multiplicity`).

If you want to analyze rotamer statistics with your own program or need populations of all rotamers in the library with original rotamer numbers, you can request a clear text file of rotamer statistics output by activating the `Save statistics` checkbox. If

you scan several chains at once, separate statistics files are written for individual chains. The file name(s) will be derived from the name of the HTML output file (see below) and statistics files will only be written if the HTML file is also written. For the format of this file, see [Clear text output](#).

You can also save the coordinates of all rotamers for all labeled residues in individual (pseudo-)PDB files. This is not needed in application work and can create a large number of very long files. If you must do this, activate the `Save PDB rotamers` checkbox. There is a warning about the storage requirements. See [Pseudo PDB format for rotamers](#) for the format description.

Click the `OK` button to proceed or the `Cancel` button to cancel the computation.

## Setting labelling conditions

For each chain you can set labeling conditions separately. This allows to scan with different labels for different chains. A window `Set labeling conditions` opens (see also [Labeling conditions window](#)). Pick the `Label` from the corresponding listbox and select the temperature with the `cryogenic` or `ambient` radiobutton. Direct temperature input in the corresponding edit field is also possible, but should be used with care and only if you understand the concept of rotamer computations well.

## HTML output of analysis results

After the labeling conditions for all chains are set, a file dialog opens that allows to select a file name for the HTML output file with results of the site scan analysis (spatial restrictions of spin labels at the selected sites, distance analysis). Decide on a file name or click `Cancel` to perform a site scan without HTML output.

## Progress of the computation

Progress is reported in the [message board](#) (*Section 2.1.17*) of the [main window](#) (*Section 2.1*) and, for individual residues, in a status bar. The message board tells how many residues are already completed and also reports on the number of significantly populated rotamers and the uncertainty of the NO bond midpoint for each individual residue (labeling site).

## Stopping computation


If necessary, you can stop the computation by closing the status bar that reports on progress for individual residues (click the cross in the top right corner). A question dialog appears that will let you `Stop` or `Resume` the computation. Note that the remainder of the site scan will be skipped if you decide to `Stop`. Analysis of the site scan will also be skipped and no HTML output stored.

## Displaying the HTML file

After the site scan is completed, you are asked whether to open the HTML result file in a web browser. If you decide not to, the file will still be stored for later use.

## Attaching labels

After a site scan, rotamer distributions are stored, but labels are not attached to the structure. It is strongly recommended to save the mode ([File/Save as...](#) menu item) before actually attaching any labels.

Now first unselect the chains and then select those residues where you want to attach labels. Click the labeling toolbutton . The labels will be attached in only seconds per label.

## How reliable are the predictions?

There are several possible sources of error:

- limited precision of the crystal structure or NMR structure of the protein
- differences in sidechain conformation between the crystalline state and the state in solution
- the assumption of a rigid backbone and only one rigid conformation of neighboring sidechains in the current algorithm for computing rotamer populations
- neglect of interactions other than repulsion and dispersion forces (Lennard-Jones potential) in computing the energy of spin-label/protein interaction

An estimate of the errors caused by the first two sources can be obtained by repeating the spin labeling procedure with repacked sidechains. For this, make a copy of the structure (before spin labelling), using the [Build/Copy structure](#) (*Section 2.1.3*) menu item. Repack sidechains in this copy by using the [Build/Repack sidechains](#) (*Section 2.1.3*) menu item (this requires that you have the third-party software [SCWRL4](#) (*Section 5.2*) installed). Do the same *in silico* spin labeling as for the original structure and compare the results (HTML output or distance distributions). In combination with a full site scan, this can help to identify sites, where spin label conformation is very sensitive to conformation of neighboring sidechains. Such sites are better avoided.

Further reliability tests become possible, if you have experimental distance distributions and want to understand the deviation between prediction and experiment. For this, you can use the [Any rotamers?](#) (*Section 3.2*) and [Range analysis](#) (*Section 3.2*) features in the DEER window.

## Finding spin labels in a structure

Residues that are spin labeled or for which rotamers were computed are [automatically annotated](#) (*Section 1.9*) and tagged with the keyword `spin labels`. Hence, a list of all spin labeled residues in a structure can be obtained with the [keyword finder](#) (*Section 1.11*) window, which is accessible via the [Edit/Find by keyword](#) (*Section 2.1.2*) menu item.

## Clear text output

If only one set of residues is scanned with the same rotamer library or only a single chain is scanned, one statistics file is written. It is stored in the same directory as the

HTML file and the name is derived from the name of the HTML file without extension by appending `_1.dat`. If several chains are scanned, separate statistics files are written for the individual chains. They are consecutively numbered (`_1.dat`, `_2.dat`, ...).

The format of the clear text output file is explained by an example:

1<sup>st</sup> line of file:

```
%% Rotamer site scan with library IA1_175K_090814
```

lists the library file used during the whole site scan.

The following lines are separated into records, one record per successfully scanned residue. For a library that contains  $n$  possible rotamers, each record has  $n+3$  lines. The record lines are:

1<sup>st</sup> line of record:

```
% [2BHW] (A){1}106 labeled at calibration temperature 175.0 (label IA1)
```

gives the residue address, the calibration temperature of the library and the label type as clear text.

2<sup>nd</sup> line of record:

```
108 37 175.0 0.50 2
```

gives the total number of rotamers (column 1), the number of significant rotamers (column 2), the actual labeling temperature (column 3), the "forgive factor" (column 4), and the label type as a number column (5, 1 corresponds to MTSL, 2 to IA-PROXYL).

3<sup>rd</sup> line of record:

```
56.292 45.484 61.913 145.834 1.38553
```

The first three columns give the C $\alpha$  coordinate of the residue (in Å), column 4 the net external potential and column 5 the partition function.

The following  $n$  lines in the record (here 108 lines) provide information on the individual rotamers. All of them have the following format:

```
49.863 41.213 59.022 0.014 0.01741
```

The first three columns give the midpoint coordinate of the N-O bond (in Å), column 4 the relative population due to only the external potential and column 5 the total population (total populations are normalized, they add up to unity).

This file can be read by Matlab with a single load command.



## Pseudo PDB format for rotamers

In the PDB mindset rotamers correspond to alternate locations of sidechain atoms. Alternate locations are tagged by a single upper-case letter, hence only up to 26 rotamers can be stored in a PDB file that conforms to the official format specification. In fact, if MMM saves a PDB file with attached rotamers, the 26 leading rotamers (with the largest populations) are saved, all others are skipped. The MMM model has information on all rotamers, but in rare cases you might want to have a PDB file with all rotamers in the same coordinate frame as the protein structure. This may be useful for methodological work on rotamers with other visualization programs that understand PDB format.

For this purpose, a pseudo PDB format is defined in which the number of a rotamer in the original library is stored as the residue number. Hence, the set of rotamers corresponds to a chain A with all residues sharing the same positions of the N, CA, and C atoms. Only atom records (HETATM) and bond records (CONNECT) are specified. The HEADER line (example):

```
HEADER SPIN LABELING WITH LIBRARY IA1_175K_090814
```

specifies the rotamer library and the TITLE line

```
TITLE [2BHW] (A){1}160 LABELED WITH IA1 AT 175 K
```

the MMM address of the labeled residue, the label type (here IA1 = iodoacetamido PROXYL) and the labeling temperature. A REMARK 4 is written to specify that this is a non-standard format:

```
REMARK 4 MMM ROTAMERS (PSEUDO-PDB FORMAT) RESIDUE NUMBERS ARE ROTAMER IDS
```

In the atom (HETATM) records, populations are stored as occupancies. Occupancies have only two significant digits. Hence, round-off errors may be significant and the populations may not add up to unity. The PDB file should thus be used only in conjunction with the [statistics file](#) (five significant digits for populations). Therefore, MMM forces that a corresponding statistics file is written when the individual PDB files are written.

A fake B factor is specified that corresponds to 1 Å r.m.s.d. of all atom positions.

The format can be read again by MMM (use the `File/Add.../From PDB.../local file` menu item) and is recognized as a special format. The structure tag is constructed based on the address of the labeled residue and the keyword `rotamers`. Backbone bond generation and DSSP secondary structure assignment (if installed) are suppressed. When displaying low population rotamers (population  $\leq 0.005$ ), you have to change transparency, otherwise they are invisible.

As the format is non-standard, there is some risk that other visualization programs do not process it properly. We have tested Swiss-PdbViewer 4.0.1 and Accelrys DS Visualizer Accelrys DS Visualizer v2.0.1.7347. Both these programs appear to work properly (within their own limits) with this format.

## 3.2 DEER simulation and fit

### DEER simulations and fits in MMM

The DEER window is intended for the following tasks:

- simulation of spin label distance distributions, including distance distributions between spin labels and native paramagnetic centers or between paramagnetic centers
- prediction of minimum and optimum values for the (maximum) dipolar evolution time in DEER/PELDOR or DQ-EPR
- estimation of multi-spin effects on dipolar evolution functions and form factors for systems with more than two labels
- direct comparison of predicted and experimental data
- tests on the likely reason for differences between predicted and experimental data (see [Any rotamers?](#) and [range analysis](#) functions)
- comparison of MMM (rotamer library) predictions of distance distributions and simulated form factors or primary DEER data with those from PRONOX or MtsslWizard
- determining mean spin label coordinates by selecting the label in the **Labels & selected atoms** listbox. The Cartesian coordinates are displayed below this listbox.

This window can be accessed via the menu item [EPR/DEER](#), (*Section 2.1.7*), but only if the current structure has at least two paramagnetic centers. Paramagnetic centers can be spin labels or selected atoms. See the help page on [spin labeling](#) (*Section 3.1*) to find out how structures are spin labeled. See the help page on [selection](#) (*Section 1.8*) to find out how particular atoms are selected.

If you work with systems with more than two labels, it is strongly suggested to [load experimental data](#) before you define the spin system as this avoids unnecessary long recomputations. For spin pairs, the sequence of inputs does not matter.

### Defining the spin system

The labeled sites can be selected from the **Labels & selected atoms** list box, where they are listed in MMM's [address format](#). (*Section 1.6*) The first tag in square brackets is the structure tag (often a PDB identifier), the second tag in parentheses the chain identifier, the number in curly brackets a number of the coordinate set (usually always 1, except for NMR structures) and the last number without any brackets the residue number. For selected atoms, the address is followed by a semi-colon and the residue name.

Note that paramagnetic centers other than spin labels are defined by the location(s) of a single atom. Centers of spin density that are defined by several atoms are not supported in this version. If the selected atom has several alternate locations, all locations will contribute to the distance distribution.

Pick the site (spin-labeled residue or atom) by clicking on the corresponding line in the list box. You can then select *only this site* with the **!** button above the list, you

can add this site to an existing spin system with the + button, and remove it from the spin system with the - button. Double selections of the same site or deselections of a site that is not in the spin system are ignored; a message window is displayed.

The ++ button selects all labeled sites. Be careful if you have many labeled sites, as this can lead to rather long computations.

The currently defined *Spin system* is displayed in the corresponding panel to the left. If the system contains at least two labels, the distance distribution and form factor are automatically computed and displayed.

## Loading experimental data

Experimental data can be loaded directly in Bruker's Xepr (EleXsys) format via the *Load Xepr* button. In this case, no experimental distance distribution is available and only the primary data, not a form factor can be fitted.

If you have experimental data in other formats, or want to preprocess them, or want to compare also an experimental distance distribution, use DeerAnalysis. MMM can import data from DeerAnalysis version 2006 or later. Such data are loaded via the *Load DeerAnalysis* button.

## Fit options

If experimental data are available, MMM tries to fit the primary data by using the simulated distance distribution for two spins or simulated rotamer distributions for more than two spins together with a homogeneous background function. This may allow to detect errors in background correction made in model-free pre-processing.

The *background model* can be selected from the corresponding popup menu. The options are *3D* (soluble proteins or membrane proteins in detergent micelles), *2D* (membrane proteins in large unilamellar vesicles or bilayers), *fractal* (may sometimes better fit membrane proteins), or *none* (may be a good option if concentration was very low). In the fractal mode, the dimension is a fit parameter.

You may also decide whether or not the modulation depth is fitted (activate or deactivate the corresponding checkbox). Usually it is appropriate to treat modulation depth as a fit parameter and this is also the default behavior. When loading data from DeerAnalysis, the modulation depth is initialized according to the background correction that was performed in DeerAnalysis. Otherwise, you can directly input modulation depth in the corresponding edit field.

## Spin label pairs

In many cases the system of interest contains just two spin labels. After [selecting the labels](#) the predicted *Distance distribution* is automatically displayed as a red plot. If experimental data were [loaded](#) in DeerAnalysis format, the experimental distribution is displayed as a black line. The C<sup>α</sup>-C<sup>α</sup> distance is plotted as a vertical

dotted green line. The distance range is automatically selected so that it contains all plotted distributions and the C<sup>α</sup>-C<sup>α</sup> distance. Contributions at distances below 1 nm are suppressed in the distance distribution and distances below 1.5 nm in the simulation.

MMM determines the mean value  $\langle r \rangle$  and standard deviation  $sr$  of the predicted distance distribution, excluding distances below 1 nm. Furthermore, the program makes two predictions for the maximum dipolar evolution time (interpulse delay  $\tau_2$  in four-pulse DEER). The minimum dipolar evolution time *Minimum tmax* should suffice to obtain a reliable experimental value for the mean distance. The optimum dipolar evolution time *Optimum tmax* should provide a reliable value for the width, and in most cases, also a reliable shape of asymmetric distributions. Both values are given in microseconds. All values are displayed left of the plot.

The form factor (dipolar evolution function after background correction) is also automatically computed in a time range between zero and the optimum dipolar evolution time. If the optimum dipolar evolution time exceeds 8  $\mu s$ , the simulated data are restricted to the range between 0 and 8  $\mu s$ .

If experimental data are available, the form factor is computed in the time range of the experimental data, but not before zero time. If the experimental data were loaded in Xepr (EleXsys) format, the *Zero time* can be manually adjusted in the corresponding edit field. For data loaded in DeerAnalysis format, no further zero time correction is possible.

The *DEER trace* plot can be either the form factor (if no experimental data are available or the *Form factor* checkbox is activated) or the primary data. If primary data were loaded in Xepr format, form factor display is not possible, the corresponding checkbox is disabled.

In either case the predicted data are displayed as a red trace and the experimental data- if any are available- as a black trace. A normalized root mean square deviation between experimental and simulated data is displayed in the *r.m.s.d.* field left of the plot. The modulation depth is automatically fitted unless the *Fit modulation depth* checkbox in the *Fit options* panel is deactivated (do this only if you have a good reason). The fit value is displayed in the edit field right of the checkbox.

If primary experimental data are displayed, the background contribution is also automatically fitted, according to the *Background model* selected in the corresponding popup menu of the *Fit options* panel. Use model *none* if you do not want a background fit. The fit value *Background decay* constant is also displayed in the *Fit options* panel.

## More than two spin labels

If the spin system contains more than two labels most functionality is the same as for [spin pairs](#). The predicted mean distance, standard deviation, minimum and optimum dipolar evolution time now correspond to the total distance distribution, which is the

sum of all pair distributions. This distribution is displayed as a red line in the *Distance distribution* plot.

The individual distributions and their corresponding C<sup>α</sup>-C<sup>α</sup> distance are displayed in shades between green (first pair) and blue (last pair). These traces are clickable. When clicked, they are displayed as bold lines and the corresponding spin pair is shown in the *Sel. distr.* field above the plot.

The form factor and primary data are now simulated considering multi-spin effects (combination frequencies). This takes much longer than a pair computation. Up to expected computation times of about 2 minutes on a current PC, a systematic computation for combinations of all rotamers of all labels is performed. If the expected computation time is longer, a Monte Carlo simulation is performed for 100000 arbitrary orientations and selections of rotamer combinations. In all multi-spin simulations, distances shorter than 1.5 nm are *not* suppressed.

The importance of multi-spin correlations depends on modulation depth (G. Jeschke, M. Sajid, M. Schulte, A. Godt, *Phys. Chem. Chem. Phys.* **2009**, 11, 6580-6591.). If experimental data were imported from DeerAnalysis, the modulation depth determined by background fitting in DeerAnalysis is assumed as total experimental modulation depth. Otherwise a default value of 0.4 is assumed. You can input the total modulation depth in the edit field right of the *Fit modulation depth* checkbox, independently of the presence of experimental data. A new input leads to re-computation of multi-spin effects. Note that the input value changes due to automatic fitting of the modulation depth (if selected), yet the input value is still used for the simulation. If you want the fitted value to be used in the simulation, you have to explicitly input the same value that is displayed in the edit field after the fit (sorry, there are reasons for that).

If no experimental data are present, the form factor simulated without multi-spin correlations is also displayed as a magenta dotted trace.

## Copying and printing plots

The plots can be detached as separate Matlab figures via the corresponding *Detach* buttons. Use the figure menu to copy or plot (or edit) them.

## Saving simulated data

The *Save* button in the left bottom corner of the window can be used to save the simulated data in text (ASCII) format, similar to the output format of DeerAnalysis. In any case the three files *\*\_res.txt*, *\*\_distr.dat* and *\*\_fit.dat* are saved, where \* stands for the basis file name that can be given in a file dialog window. The file *\*\_res.txt* contains information on simulation and fit parameters, the file *\*\_distr.dat* the distance distribution and the file *\*\_fit.dat* the form factor. The contents of file *\*\_res.txt* are self-explaining.

In file `*_distr.dat` the first column is the distance axis. If experimental data were loaded from DeerAnalysis, the second column is the experimental distance distribution and the third column the predicted one. Otherwise, the second column is the predicted distance distribution. The distance range extends from 0 to 15 nm.

Likewise, in file `*_fit.dat` the first column is the time axis. If any experimental data were loaded, the second column is the experimental form factor and the third column the predicted one. Otherwise, the second column is the predicted form factor.

A fourth file `*_bckg.dat` exists if and only if experimental data were loaded. This file contains in the first column the time axis, in the second column the primary experimental data (zero time corrected), in the third column the fitted background function, and in the fourth column the predicted primary data.

## **"Any rotamers?"- Is the assumed backbone model wrong?**

There exist four major causes for deviations between experimental distance distributions and those predicted *via* rotamer computations:

- precision of the rotamer library approach is limited
- sidechain conformations in the underlying model (for instance in the crystal structure used to compute rotamers) is not the same as in the EPR sample (crystal vs. solution, detergent vs. lipid environment)
- the experimental distribution contains artefacts due to multispin contributions, aggregation, or imperfect background correction
- the structure of the protein or nucleic acid backbone differs between the underlying model and the EPR measurement

It may be of methodological interest to study the first two causes and it is certainly important to understand the third one. However, the fourth one is of most interest in a biological context.

The `Any rotamers?` button helps to check, whether the deviation could be explained by one of the first two causes or both. The algorithm is based on the idea that the complete rotamer library (e.g. 210 rotamers for MTSSL) samples the full conformational space available to the label. If no combination of rotamers at the two labeling sites can explain a measured distances, this distance must have arisen from either experimental imperfections or a difference in backbone structure between model and reality. In particular, if the major fraction of a distance distribution cannot be explained by any combination of rotamers and singly labelled samples exclude strong aggregation or background problems, the protein structure in the EPR tube differs from the structural model.

The `Any rotamers?` button is accessible only after an experimental distance distribution was loaded. When you click on it, all pairs of rotamers are used to fit the experimental distance distribution, regardless of predicted rotamer populations. This neglects any clashes with neighboring sidechains or other parts of the protein and any conformational preferences of the labels. The best-fit distribution is displayed as a thick red dotted line. If this distribution covers the major part of the experimental



distribution, minor differences are most likely only background or aggregation problems, perhaps also a minor contribution from other backbone conformations. If a major part of the experimental distribution is not covered, there is no way for the underlying structural model to be in agreement with experiment.

Note that the argument cannot be turned around. If "Any rotamers?" can fit the experimental distribution, there might still be significant differences between backbone structure in the model and in reality. Better understanding can be gained with the `Range analysis` feature described below.

## "Range analysis"- Which rotamers are over- or underrepresented?

Suppose there is a difference between predicted and experimental distance distribution that can be mended by "Any rotamers?" (see above). Then the `Range analysis` feature may help to understand the cause.

Range analysis provides statistics on  $\chi_1/\chi_2$  rotamers in a text file and visualization of rotamers that significantly contribute to the distance distribution in a certain range. To use visualization, you need to display the cloud of spin label rotamers before you open the DEER window. You should also color this cloud for contrast with black, for instance by setting residue color to red.

Select a range by direct input into the edit fields for the left (blue) and right (magenta) cursor, or use the `<` and `>` buttons for these cursors. Now click the `Range analysis` button. The algorithm finds all rotamer pairs that contribute significantly to the distance distribution in this range. From the list of pairs, the contributing rotamers at the first and second site will be determined. Rotamer statistics for the nine possible  $\chi_1/\chi_2$  rotamers [(m,m); (m,t); (m,p); (t,m); (t,t); (t,p); (p,m); (p,t); (p,p)] is computed both for the full distribution and for the selected range. Comparison in a table shows whether a given  $\chi_1/\chi_2$  rotamer contributes more or less in this range compared to the full distribution. The rotamers contributing significantly in the selected range are colored black.

If the whole clouds are colored black and the tables show no significant differences in  $\chi_1/\chi_2$  distributions, rotamers in the selected range are not a specific subgroup of all rotamers. However, sometimes the situation is clear cut and a certain branch of the cloud (half-cloud) is black, the other one still of the original color. This phenomenon may be seen for one or both sides. It is then usually correlated to a preference for certain  $\chi_1/\chi_2$  rotamers.

Suppose now that you have analyzed a range which occurs in the predicted distribution, but not in the experimental one. In this case you know that the black branch of rotamers is relatively suppressed with respect to the red branch. Depending on your system, this may have one of the following reasons:

- nearby sidechains have different conformations in the EPR tube than in the underlying model
- your protein interacts with another protein and the suppressed branch is near the binding interface



- preferential interactions favour the red branch, but are not accounted for in the current rotamer computation algorithm

Note that the inverse problem of computing contributing rotamers from a known distance distribution is not unique. In general, if several pairs contribute at the same distance, their relative population is unknown. Therefore, the algorithm is only approximate, the result still depends somewhat on the rotamer populations from the original computation. For this reason it is good practice to look at least at two distance distributions for a residue where you suspect suppression (or enhancement) of a branch of the rotamer cloud. If analysis of both distributions suggests similar branch suppression, the effect is probably real.

## Comparison mit PRONOX

For pair distributions, comparison with PRONOX (M. M. Hatmal, Y. Y. Li, Hegde BG, Hegde PB, C. C. Jao, R. Langen, I. S. Haworth, Biopolymers 97 (2012) 35-44) has been implemented. PRONOX can be run on a web server that can in turn be accessed with the **P** pushbutton below the panel **Labels & selected atoms**. PRONOX expects absolute residue numbers in the PDB file as input. The absolute number for the currently picked residue is given at the bottom of the **Labels & selected atoms** panel. Note that you must specifically tell PRONOX when you wish to label a glycine (results are wrong when you forget this). You may be better off using the low clash mode (0.4 instead of 0.75 in column vdW for both sites) by default (see G. Jeschke, Progr. Nucl. Magn. Reson. Spectr. 2013, submitted).

After running PRONOX, you end up at the result page. Save the distance file by right-clicking on the view **distances** link. By default this file has the extension **.add**. You may want to inspect the results before by left-clicking on the same link and scrolling down (this will show you whether the computation was successful).

Now you can load the **.add** file into MMM using the **Load PRONOX** pushbutton. Depending on the number of conformations this can take quite long. If an MMM distance distribution is already displayed at this time, the PRONOX distance distribution will be shown as a dark blue dotted line. If experimental data are loaded these will also be shown as solid black lines and the PRONOX DEER trace or form factor as a dark blue dotted line.

The PRONOX data are not displayed unless also an MMM distance distribution (red line) is displayed. They remain stored, however, for future display until the DEER window is closed. PRONOX data can be overwritten by loading a new data set.

## Comparison mit MtsslWizard

For pair distributions, comparison with MtsslWizard (G. Hagelueken, R. Ward, J. H. Naismith, O. Schiemann, Appl. Magn. Reson. 42 (2012) 377-391) has been implemented. MtsslWizard is a plugin for PyMOL, see the corresponding [PyMOL Wiki page](#). You need to install PyMOL and the plugin to work with it. Save the results as a **.txt** file, as is default in MtsslWizard.

Now you can load the `.txt` file into MMM using the `Load MtsslWizard` pushbutton. This can take a few seconds. If an MMM distance distribution is already displayed at this time, the MtsslWizard distance distribution will be shown as a dark green dotted line. If experimental data are loaded these will be shown as solid black lines and the MtsslWizard DEER trace or form factor as a dark green dotted line.

The MtsslWizard data are not displayed unless also an MMM distance distribution (red line) is displayed. They remain stored, however, for future display until the DEER window is closed. MtsslWizard data can be overwritten by loading a new data set.

## 3.3 Localization of spin labels

### What is localization?

#### General concept

Localization determines possible coordinates for a single point in space from distances to a set of known reference points. This is also known as trilateration or multilateration and, in the first papers in the field of pulse dipolar EPR spectroscopy, has been termed triangulation. MMM uses the general term localization because the implementation goes beyond multilateration by taking into account the uncertainty of the known distances.

Typical application scenarios are the localization of spin-labeled cofactors or substrates with respect to a protein, see [Gaffney et al., Biophys. J. 2012, 103, 2134-2144](#) or the localization of a spin-labeled residue in a domain that is not resolved in a crystal structure or NMR structure. If you wish to localize *several* unknown points, you can still use localization by running it for each individual point and combining the results from the output PDB files. However, if you have a sufficient number of distances also between the unknown points, you should consider [network determination](#), (Section 3.4) which is also required if the number of reference points is insufficient.

#### Limitations

At least three distances to distinct reference points are required. In this case localization will result in an ambiguous result, with two sets of coordinates that are mirror images with respect to the reference point triangle. This ambiguity is resolved by a fourth reference point unless the symmetry is accidentally maintained by the fourth point. In practice, localization with four reference points still results in substantial uncertainty if the distances are uncertain. This is always the case with spin labels, since predictions of spin label conformational distribution are uncertain. Hence, whenever possible, at least five reference points should be used.

The point to be localized can also be a paramagnetic center that is not a spin label. However, the current implementation can directly compute reference coordinates from a given structure only for spin labels. Other reference points have to be

specified with their full coordinates in the [constraint file](#).. Currently it is not possible to mix automatic coordinate computation for spin labels with direct specification of other reference points. As a workaround you may determine mean spin label reference coordinates with the [DEER window](#) (Section 3.2) and write these coordinates in the constraint file.

## Algorithm

The multilateration problem with more than three reference points can be linearized, however, such linearization is numerically unstable with respect to errors in the given distances (W. S. Murphy Jr., Master thesis, Colorado School of Mines, 2007). We use linearization to compute a first guess for the unknown coordinates and avoid the instability by non-linear least squares minimization starting from this guess. With three reference points (trilateration) the two possible point coordinates are computed from analytical formulas.

The coordinates obtained by multilateration is only used to predict a region (or two regions for trilateration) in space where the label can be located. We define this region as a cube with 6 nm side length whose center is the multilaterated point. This cube is sampled by a grid with uniform spacing  $\Delta x = \Delta y = \Delta z = 0.4 \text{ \AA}$ . We now compute the probability for the spin label to be located at a given grid point. From the  $n$  experimental distances to the reference points and the corresponding distances of the grid point to the reference points we calculate probabilities  $p_i$  ( $i = 1 \dots n$ ) that these distances are consistent with experiment. This is done either by assuming a Gaussian distribution of the individual distances, with mean values and standard deviations given in the [constraint file](#), or by using a numerical distribution that must be specified also in the constraint file. Assuming that the distance distributions are uncorrelated, the total probability for the label to be located at this grid point is the product of the  $n$  individual probabilities.

After computing this total probability at all grid points, we know the probability density distribution in the cube at 0.4 Å resolution. We now find a level  $I_{1/2}$  of the normalized probability density so that an probability density isosurface at this level includes 50% of the total probability. This level is determined at 1% resolution of the maximum probability density. For an unambiguous localization, the isosurface is displayed as a red semitransparent surface object. For an ambiguous localization, one of the solutions is displayed by a red and the other one by a green isosurface. Note that inclusion of 50% of the total probability is also usual practice in displaying thermal ellipsoids in crystal structures of small molecules.

Two important quality measures are reported in the **Message board**. The highest location probability is a measure for the maximum probability that all distance constraints are simultaneously fulfilled at any point in space. A perfect solution has a value of 1. Values smaller than 1 are to be expected even if the distance distributions are known exactly, since they will be correlated to some extent and this correlation was neglected in computing probability density. However, values smaller than 0.25 usually indicate problems with the experimental constraints. The second quality measure is the maximum relative probability at density cube border. A good solution has 0.00%, but values of a few percent are acceptable. Values larger than 10%

indicate that the experimental constraints are insufficient for localization. If the constraints are very poor, it may even happen that the 50% isosurface extends beyond the 6 nm cube. This is easily recognized by the displayed surface not being closed. In such cases, additional experimental constraints are required.

For interactive analysis of the result in MMM or other PDB viewers the spatial probability distribution can be sampled by an ensemble of point coordinates. For that, the volume enclosed by the 50% isosurface is successively filled by spheres, starting at the point of maximum probability density. The next spheres are also located in the point of maximum probability density, but under the constraint that they do not intersect any of the spheres that were put before. The spheres are displayed as grey balls with a transparency corresponding to relative probability density at their center. Number and diameter of the spheres can be specified in the [constraint file](#). If the specification is missing only a single sphere with a diameter 2 Å is created at the point of maximum probability density.

The ensemble of spheres that sample the 50% probability density volume can be written out as a PDB file (see [constraint file](#)). The spheres are specified as sodium atoms, each sphere corresponding to a separate model. Relative probability densities at the sphere centers are stored as occupations. This file can be re-read by MMM and is also compatible with Accelrys Discovery, Chimera, PyMol, and Swiss-PdbViewer 4.1.0.

## Constraint file

A constraint file serves for both specifying the distance constraints between the reference points and the point to be localized and for controlling output. A typical constraint file looks like this:

```
% Localization test restraints for SBL1 with respect to lipxygenase
structure 1YGE

% Restraints from B. J. Gaffney, M.D. Bradshaw, J. Freed, P. Borbat,
Biophys. J. 2012, 103, 2134-2144

# PDB 1YGE % this file needs to be loaded into MMM before selecting
localization

# ENSEMBLE 20 3

# OUTPUT Gaffney_SBL1

# LOCATE SBL1

270 2.30 0.45

480 4.30 0.45

569 3.80 0.45

619 4.70 0.45
```

```
782 3.75 0.45
```

```
# END
```

This file is provided as `localization_test.dat` in MMM subdirectory `\test\multilateration`.

Lines starting with a percent character, %, are comment lines. Lines starting with a hash characters, #, are command lines. Other lines provide additional arguments to the previous command. The optional `# PDB` command specifies the reference structure. The reference structure must have been loaded before into MMM and must be the active structure. If there is a mismatch between the tag of the active structure and the identifier specified in the `# PDB` command, a dialog window pops up for confirmation. Apart from activating this security question, the `# PDB` command has no function.

The optional `# ENSEMBLE` command specifies how many spheres shall be used for sampling the 50% probability volume (default 1) and which diameter they shall have (default 2 Å).

The optional `# OUTPUT` command specifies whether a PDB file with the localization information is written. If the command is present without an argument, a PDB file with name `locations_tag.pdb` is written, where `tag` is the tag specified with the `# LOCATE` command. A user-defined filename can be specified as an argument of the `# OUTPUT` command, as in the example above. If the filename argument does not have an extension `.pdb`, this extension is automatically appended. Unless a full path name is given, the PDB file is written to the directory from which the constraint file was loaded.

The mandatory `# LOCATE` command signifies that the following lines until the next command are constraint lines. Its first argument is the tag for the localized point (`SBL1` in the example) that is also used for tagging the 50% isosurface, as a PDB identifier in the output PDB file, and in an automatically generated filename for the output PDB file. If this tag is missing, it defaults to `loc`, although it is not recommended to omit the tag. The second, optional argument can be used to specify the isosurface level (probability included in the volume, defaults to 0.5). As a third, optional argument, the type of labels at the reference points can be specified (currently either `MTSL` or `IA-Proxyl`, with `MTSL` being the default. As a fourth, optional argument, the temperature of the rotamer library can be specified. It defaults to `298 K`.

The `# LOCATE` command is followed by as many constraint lines as there are reference points. Each constraint line has at least three arguments. The first argument is the address of the spin-labeled residue, either just the residue number or an MMM address, such as `(B)23` for residue 23 in chain B of the active structure. The second argument is the experimental mean distance (in nanometers) between the reference label and the paramagnetic center to be localized. The third argument is the standard deviation of this distance, also in nanometers. Distances and

standard deviations can alternatively be specified in Ångstroems. For that the `# LOCATE` command has to be preceded by the command `# UNITS A` or `# UNITS ANGSTROEM`.

Constraint lines can specify the filename of a distance distribution file in DeerAnalysis output format. In that case probability densities are taken from the experimental distribution rather than computed from the mean distance and standard deviation. Unless a full pathname is provided, MMM expects the distance distribution file in the same directory from which the constraint file was loaded.

If necessary, coordinates for reference points can be directly specified with the `# REFERENCE` command, as in the following example:

```
# REFERENCE

D50 -28.0 13.5 18.2

D166 -16.4 -3.2 -10.9

D217 -24.8 -17.8 20.0

D220 -14.6 -25.8 3.4

# LOCATE SANO

D50 3.40 0.46


D166 3.13 0.31

D217 3.76 0.12

D220 3.94 0.24

# END
```

The lines after the `# REFERENCE` command consist of the reference site tag and the x, y, and z coordinates. Please note that coordinates are specified in Angstroems, irrespective of the units for distance restraints. If a `# REFERENCE` command exists, *all* reference sites must be specified by coordinates.

If you want to mix spin labeled reference sites and other reference sites, you need to determine mean spin label coordinates manually and write them into the constraint file. To do so, [select](#) (Section 1.8) the residues to be spin labelled and perform a [spin labeling site scan](#), (Section 3.1), attach the labels by clicking the labeling toolbutton . Then go to the [DEER window](#) (Section 3.2) and select the labeled sites, one by one, in the `Labels & selected atoms` listbox. The coordinates in Angstroem appear below the listbox.

It is good practice to finish the constraint file by an `# END` command.



## 3.4 Networks of sites

### What is a network in MMM?

#### General concept

A network in MMM is a set of  $M$  sites with some or all of the  $M(M-1)/2$  site-pair distances being known. If all distances are known the geometry of the network polyhedron can be fully determined, except that the mirror image is also a possible solution. If at least 4 of the sites are reference points in a known structure, this ambiguity can also be resolved. In fact, the set of  $M$  3D coordinates has only  $3M-6$  degrees of freedom, as rotation and translation of the whole network do not affect internal geometry or node-to-node distances. Hence, the problem can still be solved when  $3M-6$  distances are known. For instance, a network with 6 nodes has 15 distinct node-to-node distances, whereas 12 distances are sufficient to determine geometry.

If at least 3 of the sites are reference points in a known structure, the network can be fitted to the known structure with mirror image ambiguity. With 4 reference points, the mirror image ambiguity vanishes.

Typical application scenarios are the localization of spin-labeled cofactors or substrates with respect to a protein, see [Gaffney et al., Biophys. J. 2012, 103, 2134-2144](#) or the localization of several spin-labeled residues in a domain that is not resolved in a crystal structure or NMR structure. The feature may also be useful if the structure is completely unknown. In that case the network together with the known number of amino acid residues between sites and possibly secondary structure restraints from accessibility measurements or a web server may help to develop hypotheses on the structure. If you do not have enough distances between the reference points, but at least three distances between each of the unknown points and some reference points, you should use [localization](#) (Section 3.3) instead.

#### Limitations

At least three distances to distinct reference points are required. In this case localization will result in an ambiguous result, with two sets of coordinates that are mirror images with respect to the reference point triangle. This ambiguity is resolved by a fourth reference point unless the symmetry is accidentally maintained by the fourth point. In practice, localization with four reference points still results in substantial uncertainty if the distances are uncertain. This is always the case with spin labels, since predictions of spin label conformational distribution are uncertain. Hence, whenever possible, at least five reference points should be used.

The network nodes do not need to be spin labels. However, the current implementation can directly compute reference coordinates from a given structure only for spin labels. Other reference points have to be specified with their full coordinates in the [constraint file](#). Currently it is not possible to mix automatic coordinate computation for spin labels with direct specification of other reference



points. As a workaround you may determine mean spin label reference coordinates with the [DEER window](#) (Section 3.2) and write these coordinates in the constraint file.

## Algorithm

If all the  $M(M-1)/2$  site-pair distances are known, solution of the problem by distance matrix geometry is straightforward. The  $M \times M$  distance matrix is converted to a metric matrix (`metric_matrix.m`), the metric matrix is diagonalized, and if the first three eigenvalues are all positive, the network point coordinates are given by the first three eigenvectors scaled by the square root of the corresponding eigenvalue (`dmat2coord.m`, see also G. M. Crippen, T. F. Havel, Distance Geometry and Molecular Conformation, Research Studies Press Ltd. Taunton, 1988.). If the input distances are inconsistent with a 3D structure the matrix cannot be embedded (zero or negative values among the first three eigenvalues). In that case the program returns with an error message in the **Message board**.

Since the distances are fraught with errors, there is no guarantee that the node coordinates indeed fulfill all distance constraints. The coordinates are iteratively refined (`bound_refiner.m`) until they are within lower and upper bounds specified in the [constraint file](#). For mean distance/standard deviation pairs ( $r, \sigma$ ), the bounds are taken as  $r - 2\sigma$  and  $r + 2\sigma$ . Lower bounds below 3 Å and upper bounds above 100 Å are set to these general bounds. The general bounds can be redefined in the [constraint file](#). If iterative refinement fails, the program returns with an error message in the **Message board**.

If not all site-pair distances are known, the missing distances have to be estimated from lower and upper bounds (restraints). Such bounds can be specified for individual distances in the [constraint file](#). If constraints ( $r, \sigma$ ) or restraints (lower and upper bound) have been specified for at least  $3M-6$  site pairs, a Monte Carlo solution is attempted. For the missing distances random guesses are made in the interval between the general lower and general upper bound (default 3 to 100 Å). It is possible to specify only a tighter lower or only a tighter upper bound for individual distances in the [constraint file](#). Bounds are refined by triangle bound smoothing (G. M. Crippen, T. F. Havel, Distance Geometry and Molecular Conformation, Research Studies Press, Taunton, 1988, p. 252 f.).

In Monte Carlo mode, the program attempts to compute 1000 sets of consistent node coordinates (models). After a computation time of 1 hour the program is stopped, even if less than 1000 sets were found. In this case the problem is not sufficiently constrained. The program returns with an error message in the **Message board** if it did not find any solution, otherwise it reports the number of trials and the number of failed embeddings. The preliminary node coordinates are computed as the mean coordinates of all models.

The coordinates obtained in the previous step are used to obtain estimates of uncertainty for each network node. For each node a cube with 6 nm side length with the node as its center is searched for locations that are consistent with the mean coordinates of the other nodes and the known distances. This cube is sampled by a

grid with uniform spacing  $\Delta x = \Delta y = \Delta z = 0.4 \text{ \AA}$ . The probability for the spin label to be located at a given grid point is computed. From the  $n$  experimental distances to the reference points and the corresponding distances of the grid point to the reference points the program calculates probabilities  $p_i$  ( $i = 1 \dots n$ ) that these distances are consistent with experiment. This is done either by assuming a Gaussian distribution of the individual distances, with mean values and standard deviations given in the [constraint file](#), or by using a numerical distribution that must be specified also in the constraint file. Assuming that the distance distributions are uncorrelated, the total probability for the node to be located at this grid point is the product of the  $n$  individual probabilities. If an upper bound/lower bound restraint is violated at the grid point, the probability is set to zero.

After computing this total probability at all grid points, we know a probability density distribution in the cube at  $0.4 \text{ \AA}$  resolution. We now find a level  $I_{1/2}$  of the normalized probability density so that an probability density isosurface at this level includes 50% of the total probability. This level is determined at 1% resolution of the maximum probability density. The isosurface is displayed as a red semitransparent surface object. Note that inclusion of 50% of the total probability is also usual practice in displaying thermal ellipsoids in crystal structures of small molecules. Note, however that in the case at hand the isosurface is only a rough approximation since in computing this surface, no uncertainty was assigned to the coordinates of all other nodes. Nevertheless, visualization of these surfaces provides a qualitative picture of how well or how poorly the problem is constrained. The isosurfaces are clickable. When clicked, the site tag, the center of gravity of the density cloud, and the point of highest probability are reported in the **Message board**.

Important quality measures are reported during the computation in the **Message board**. The highest location probability is a measure for the maximum probability that all distance constraints are simultaneously fulfilled at any point in space. A perfect solution has a value of 1. Values smaller than 1 are to be expected even if the distance distributions are known exactly, since they will be correlated to some extent and this correlation was neglected in computing probability density. However, values smaller than 0.25 usually indicate problems with the experimental constraints. The second quality measure is the maximum relative probability at density cube border. A good solution has 0.00%, but values of a few percent are acceptable. Values larger than 10% indicate that the experimental constraints are insufficient for localization. If the constraints are very poor, it may even happen that the 50% isosurface extends beyond the 6 nm cube. This is easily recognized by the displayed surface not being closed. In such cases, additional experimental constraints are required.

For interactive analysis of the result in MMM or other PDB viewers the spatial probability distribution for each node can be sampled by an ensemble of point coordinates. For that, the volume enclosed by the 50% isosurface is successively filled by spheres, starting at the point of maximum probability density. The next spheres are also located in the point of maximum probability density, but under the constraint that they do not intersect any of the spheres that were put before. The spheres are displayed as grey balls with a transparency corresponding to relative probability density at their center. Number and diameter of the spheres can be

specified in the [constraint file](#). If the specification is missing only a single sphere with a diameter 2 Å is created at the point of maximum probability density.

The ensembles of spheres that sample the 50% probability density volume of the nodes can be written out as a PDB file (see [constraint file](#)). The spheres are specified as sodium atoms, each set of  $M$  spheres for the  $M$  nodes corresponding to a separate model. Relative probability densities at the sphere centers are stored as occupations. This file can be re-read by MMM and is also compatible with Accelrys Discovery, Chimera, PyMol, and Swiss-PdbViewer 4.1.0. Inspection of the set of models is particularly convenient with Accelrys DS Visualizer.

## Constraint file

A constraint file serves for both specifying the distance constraints between the reference points and the point to be localized and for controlling output. A typical constraint file looks like this:

```
% Network fit and superposition for localizing LOPTC in liopxygenase
structure 1YGE

% Restraints from B. J. Gaffney et al., Biophys. J. 2012, 103, 2134-2144

# PDB 1YGE % if this file is loaded into MMM before selecting network,
% the network is fitted to the five reference points

# ENSEMBLE 20 2

# OUTPUT Gaffney_network

# NETWORK SBL1 0.5 MTSL 298 constraints

270 480 3.25 0.30

270 569 3.70 0.30

270 619 4.70 0.30

270 782 3.80 0.30

480 569 3.75 0.30 % comment out this line for Monte Carlo test

480 619 4.65 0.30

480 782 5.25 0.30

569 619 4.40 0.30

569 782 5.20 0.30

619 782 2.20 0.30
```

```

LOP 270 2.30 0.30

LOP 480 4.30 0.30

LOP 569 3.80 0.30

LOP 619 4.70 0.30

LOP 782 3.75 0.30

# END

```

This file is provided as `network_Gaffney.dat` in MMM subdirectory `\test\multilateration`.

Lines starting with a percent character, %, are comment lines. Lines starting with a hash characters, #, are command lines. Other lines provide additional arguments to the previous command. The optional `# PDB` command specifies the reference structure. A reference structure can be loaded into MMM and must then be the active structure. If there is a mismatch between the tag of the active structure and the identifier specified in the `# PDB` command, a dialog window pops up for confirmation. Apart from activating this security question, the `# PDB` command has no function. Network computation can be run without reference structure.

The optional `# ENSEMBLE` command specifies how many spheres shall be used for sampling the 50% probability volumes (default 1) and which diameter they shall have (default 2 Å).

The optional `# OUTPUT` command specifies whether a PDB file with an ensemble of possible network coordinates is written. If the command is present without an argument, a PDB file with name `network_tag.pdb` is written, where `tag` is the tag specified with the `# NETWORK` command. A user-defined filename can be specified as an argument of the `# OUTPUT` command, as in the example above. If the filename argument does not have an extension `.pdb`, this extension is automatically appended. Unless a full path name is given, the PDB file is written to the directory from which the constraint file was loaded.

The mandatory `# NETWORK` command signifies that the following lines until the next command are constraint lines. Its first argument is the tag used in place of a PDB identifier in the output PDB file (`SBL1` in the example) and in an automatically generated filename for this file. If this tag is missing, it defaults to `DGM1`, although it is not recommended to omit the tag. The second, optional argument can be used to specify the isosurface level (probability included in the volume, defaults to 0.5). As a third, optional argument, the type of labels at the reference points can be specified (currently either `MTSL` or `IA-Proxyl`, with `MTSL` being the default. As a fourth, optional argument, the temperature of the rotamer library can be specified. It defaults to 298 K. The fifth, optional argument specifies the extent of visualization (default is `constraints`). With `full` both spheres at the nodes and the constraints are visualized. Constraints between two reference sites are shown as dark blue lines

and other constraints as red lines. With `nodes` only the node spheres are shown. With `constraints` only constraints between nodes are shown. With `none` both node and constraint display is suppressed. The probability isosurfaces and the spheres sampling them are always displayed.

The `# NETWORK` command is followed by as many constraint lines as there are reference points. Each constraint line has at least three arguments. The first argument is the address of the spin-labeled residue, either just the residue number or an MMM address, such as `(B)23` for residue 23 in chain B of the active structure. For distance constraints the second argument is the experimental mean distance (in nanometers) between the reference label and the paramagnetic center to be localized. The third argument is the standard deviation of this distance, also in nanometers. Distances and standard deviations can alternatively be specified in Ångströms. For that the `# NETWORK` command has to be preceded by the command `# UNITS A` or `# UNITS ÅNGSTROEM`.

To input restraints (lower or upper bounds) the second argument (lower bound) or third bound (upper bound) must be given as negative numbers. If only one of the two arguments is negative, the positive one is ignored and replaced by the default bound.

Constraint lines can specify the filename of a distance distribution file in DeerAnalysis output format. In that case probability densities are taken from the experimental distribution rather than computed from the mean distance and standard deviation. Unless a full pathname is provided, MMM expects the distance distribution file in the same directory from which the constraint file was loaded.

The optional commands `# LOWER` and `# UPPER` allow for adjusting the default lower and upper distance bounds that are applied to node pairs for which no constraint or restraint was specified. The defaults are 3 and 100 Å, respectively. Adjusting these bounds according to the expected network size and minimum distance between nodes can strongly improve model quality in cases where some node pairs are otherwise unconstrained. Note that these bounds must be consistent with all node pair distances in the network, including those ones that are specified by individual constraints or bounds.

If necessary, coordinates for reference points can be directly specified with the `# REFERENCE` command, as in the following example:

```
# REFERENCE  
  
S270 17.13 59.97 2.23  
  
S480 -7.19 45.70 2.54  
  
S569 12.41 47.50 33.61  
  
S619 25.57 16.19 10.35  
  
S782 41.14 29.12 1.84
```


```
# NETWORK SBL1 0.5 MTSL 298 full

S270 S480 3.25 0.30

...

# END
```

The lines after the `# REFERENCE` command consist of the reference site tag and the x, y, and z coordinates. Please note that coordinates are specified in Angstroms, irrespective of the units for distance restraints. If a `# REFERENCE` command exists, *all* reference sites must be specified by coordinates. Note also that the arguments `MTSL 298` of the `# NETWORK` command are ignored in this example.

If you want to mix spin labeled reference sites and other reference sites, you need to determine mean spin label coordinates manually and write them into the constraint file. To do so, [select](#) (Section 1.8) the residues to be spin labelled and perform a [spin labeling site scan](#), (Section 3.1), attach the labels by clicking the labeling toolbutton . Then go to the [DEER window](#) (Section 3.2) and select the labeled sites, one by one, in the `Labels & selected atoms` listbox. The coordinates in Angstrom appear below the listbox.

It is good practice to finish the constraint file by an `# END` command.

## 3.5 Docking of protomers in a complex

### Docking concept

#### General concept

MMM performs a constraint driven rigid-body docking of (two) protomers in a complex. No further adjustments of the docked molecules as for example relaxation of the complex interface by molecular dynamics is performed. In this way, a user can see how well the monomers can be brought together using solely experimental information, and at the same time possibly judge how good the available pool of experimental constraints is. Two docking algorithms are offered: a grid search and a non-linear fitting.

#### Grid search vs fitting

Both approaches pursue the same goal of putting two rigid partners of a complex together satisfying given constraints in the best possible way. In the fitting approach, the whole space of possible mutual arrangements of the complex partners is not sampled. The algorithm advances through space governed by a certain optimization criterion aiming to locate the best solution fast (i.e. a global minimum on the error surface). Because rigid body docking is a multidimensional task (6 free parameters in a general heterodimer case - 3 Euler angles and a 3-component translation vector in a Cartesian space), practically any existing fitting algorithm is likely to end up in a



local minimum of the error surface, especially if the surface is rather rough. Grid search can help with that problem. With this approach, the space of mutual orientations and translations of the complex partners is discretized into grids, and every grid is visited during a grid search run. With that, chances to miss the global minimum are reduced, especially if the chosen grid is sufficiently fine. In contrast to fitting, however, a grid search can be a very time consuming task (can last hours or days). One can combine the two approaches so that the grid search is performed first aiming to get close to the global minimum, and then the fitting is initiated from the best grid search model.

Note that for Euler rotations, a passive definition of rotation matrices is adopted.

## Docking constraints

The ultimate goal of MMMDock is to support any kind of experimental constraints which would help in docking.

In the current release (v. 2013.2) only EPR distance constraints in a form of a mean distance and a standard deviation are supported. Constraint support will be extended already in the next MMM release.

## Working with MMMDock

### Constraints file

Constraints are organized in an ASCII file with \*.dat extension. Examples of the constraints files can be found in the ...\\tests\\docking folder of your MMM installation. All strings starting with % are treated as comments and ignored while strings starting with # are keywords with values. Such keyword lines may be followed by additional argument lines. The order in which the keywords are given is irrelevant. A string #PDB pdb1 pdb2 tells the programme that there are two monomers to be docked with pdb id's pdb1 and pdb2 (heterodimer case). Only one pdb\_id given would correspond to a homodimer case. A string #DEER MTSL 175 means that the type of constraints is DEER, a spin label used is MTS while the rotamer library to be used by MMM during docking should correspond to a frozen solution (indicated by 175). An order in which two pdb id's are given defines which structure is fixed (the first one given) during docking and which is being moved (the second one). Constraints table follows. Record with a constraint may look like: (A) 65 (A) 135 4.63 0.22 specifying distance constraint between site 65 (chain A, structure pdb1) and site 135 (chain A, structure pdb2). IMPORTANT: the first column in the constraints table corresponds to the pdb\_id which is given first in the # PDB line. If no keyword for units is given, nanometers are assumed. The constraint file ends with the line # END (after that everything is skipped).

### MMMDock window

Structures of monomers to be docked have to be loaded into MMM prior to opening the MMMDock window with the [Predict menu/quaternary/MMMDock](#) (Section 2.1.9) menu item. For a homodimer case, one structure is necessary while for a heterodimer - two. In the freshly loaded window, press Load restraints list and



choose desired list file. The program checks if the sites specified do exist in the structures and if sites were already labelled previously (for other purposes). If a site exists and is not labelled, it will be labelled upon loading the constraints list. When constraints are successfully loaded, more functionality becomes accessible. For a homodimer case, there is a choice of treating it as a homodimer (4 degrees of freedom, faster search), or as a heterodimer (6 degrees of freedom, longer search). For a heterodimer case there is no choice, and this option is deactivated. By default, the fitting is accessible only after the constraints were loaded. Switching between the fitting and the grid search approaches is done by choosing a corresponding radiobutton on the *Docking method* panel.

To start fitting, give initial values for the Euler rotation and the translation vector in the *start values* panel and press the "Run docking" button. A popup window will show a real-time performance of the fitting algorithm. When finished, the bestfit values are shown in the *best fit values* panel. If *Save docking statistics* and *Save transformed pdb's* checkboxes were checked (default behaviour), the transformed protomers will be saved as PDB files and the statistics of the docking session will also be saved (both under default names containing a reference to a real time moment). In the statistics file, the information about initial template, docking constraints and the docking result is given.

For docking, the user is expected to provide start, end and the number of values for the Euler rotation angles and the translation vector. The grid size is automatically updated and shown in the *Grid size* window. If the specified grid is very large, not only the computation time is long, but also the size of the internal grid variable can become too large to keep it in the memory. For such cases, a *Stack mode* is initiated. The stack is the number of grid models which are kept in the memory during the grid search. Only good model may enter the stack. With that, every grid point is visited and evaluated, but only a number of models limited by the stack size variable is returned. Grids with less than 1'000'000 points are small so that the *Stack mode* is disabled.

By checking *Automatically fit from the best grid point* the user initiates automatic fitting from the best found grid point at the end of grid searching. Docking is initiated by hitting the *Run docking* button. Important steps and states of the docking session are indicated in the MMM Message board.

When the user quits MMMDock by clicking the *Finish/exit* button, the program checks if docking results were saved. One may perform numerous docking attempts without closing the MMMDock window, and if *Save docking statistics* and *Save transformed pdb's* are activated, saving occurs on every run. But if no saving was selected, then hitting the *Finish/exit* button allows for saving the results of the last docking run.

## Examples and tests

Load structure *1ZCDA.pdb* (only chain A of the NhaA crystal structure) from the */test* subdirectory and select the *Predict/quaternary/MMMDock* menu item. Using the *Load restraint list* button, load the constraint file *NhaA\_docking\_constraints\_MMMtest.dat*. Switch *Docking method* to *grid search* by activating the appropriate radiobutton. Use a 5-degree Euler angle grid by editing *Alpha elem* to 73 and *Beta elem* to 37. Use a 2.5 Å translation grid by editing *x elem* and *y elem*

to 61 each. Activate `Automatically fit from the best grid point` and click `Run docking`. Now is your chance to have a coffee and a nice, long chat with your colleagues (or even lunch, depending on your computer). After the grid search has completed, the model of the docked complex is displayed.

You might want to try the same with an artificial heterodimer case, using structure files `2LZM_fragment1.pdb` and `2LZM_fragment2.pdb` and constraint file

`T4L_docking_constraints_MMMtest.dat`. Despite having more free parameters, this case is better constrained (higher constraint quality for this soluble protein, larger constraint/degrees of freedom ratio), so that the grid can be a bit more rough (10 degree, 37 Alpha and Gamma steps, 19 beta steps, 5 Å). In addition, the fragments are smaller and you may want to limit the translation range to -40... 40 Å (17 translation steps in each dimension). Computation time allows for an afternoon in the experimental lab, or, if your computer is slow, a good night of sleep. The result reproduces the T4 lysozyme structure 2LZM (except for the missing Gly-113) with 2.86 Å backbone r.m.s.d.

## 3.6 Domain ensemble

### Introduction

This builder is intended for disordered or partially disordered domains that may be N-terminal, C-terminal, or may connect two resolved residues (closed loops) in an existing protein structure. Alternatively, it can be used to model intrinsically disordered proteins (free standing mode). It requires that SCWRL4 is installed and present on the Matlab path.

Possible restraints include

- distances between one spin label in the disordered domain and one label (the beacon) in the resolved part of the structure
- distances between two labels in the disordered domain
- the distance between a labeled residue and the equivalent residue in the neighboring protomer of an oligomer
- residue depth in a lipid bilayer
- secondary structure, also specified by propensities ( $\alpha$ -helix,  $\beta$ -strand, PPII helix)
- specification of cis peptide links

In the absence of any restraints, the ensemble conforms to residue-specific Ramachandran statistics for protein domains without secondary structure (default) or such statistics derived from the entire PDB (option that can be selected by the user). Residue-specific Ramachandran statistics was derived by digitizing the plots from: Sven Hovmöller, Tuping Zhou, Tomas Ohlson (2002) *Conformations of amino acids in proteins*. Acta Cryst. D **58**, 768-776. Definition of backbone dihedral angles for helices and strands is also based on the analysis described in this paper.

Random cis links are generated according to the statistics of the [Jena library of Biological Macromolecules](#), i.e. with a probability of 5% from pre-proline residues and of 0.03% otherwise.

The ensemble can be visualized during the computation and can be saved from the MMM File menu as a PDB file with several models. Fulfillment of distance and depth restraints by the ensemble can be visualized.

Modeling requires a restraint file (the format is described below) and, optionally, a structure in which the disordered domain is anchored.

The template structure (if any) must be loaded and must be the current structure in MMM before the window is opened. Otherwise the program assumes the free standing mode.

## Restraint file

Currently, the following restraint keywords are defined for domain ensemble modeling:

- **# PDB** (single-line), argument: PDB identifier of template structure, the chain can be given after a colon (:)
- **# DOMAIN** (single-line), arguments: number of the first and last residue to be modelled
- **# NANCHOR** (single-line), argument: MMM address of the residue in the template structure that precedes the first residue of the domain to be modelled
- **# CANCHOR** (single-line), argument: MMM address of the residue in the template structure that follows the last residue of the domain to be modelled
- **# SEQUENCE** (single-line), argument: sequence (in single-letter amino acid code) of the domain to be modelled
- **# ENSEMBLE** (single-line), arguments: number of conformations to be modelled and ensemble probability, which defaults to 0.5 (see below)
- **# DEER** (block), arguments: spin label and temperature (in K) for rotamer simulation, starts block for EPR (DEER and CW EPR) restraints
- **# OLIGOMER** (block), arguments: multiplicity n, spin label and temperature (in K) for rotamer simulation, starts block for EPR (DEER and CW EPR) restraints
- **# DEPTH** (block), arguments: spin label and temperature (in K) for rotamer simulation, starts block for depth restraints, spin label can be replaced by CA for C $\alpha$
- **# HELICES** (block), no arguments, starts specification of secondary structure restraints for ( $\alpha$ )-helices.
- **# STRANDS** (block), no arguments, starts specification of secondary structure restraints for  $\beta$ -strands.
- **# CISPEPTIDE** (single-line), arguments: residue numbers, separated by spaces, of residues that have cis peptide links to the next residue
- **# APROP** (block), no arguments, starts specification of  $\alpha$ -helix propensity restraints
- **# BPROP** (block), no arguments, starts specification of  $\beta$ -strand propensity restraints
- **# PPROP** (block), no arguments, starts specification of PPII-helix propensity restraints
- **# END** (single-line), no arguments, stops reading of the file

Among these keywords, **# DOMAIN** and **# SEQUENCE** are mandatory and, currently, and at least one anchor ( **# NANCHOR** for a C-terminal loop or **# CANCHOR** for an N-terminal loop)) is required if beacon restraints are used. If both **# NANCHOR** and **# CANCHOR** are provided, a closed loop is modelled. Without anchors the loop is free standing and cannot be related by restraints to another structure.

## Remarks

# **PDB**: This is used for a check whether the current structure in MMM, which can be picked in the [hierarchy window](#), (Section 2.3) is the requested template structure. If the current structure in MMM has a different PDB identifier than specified in the constraints file, the user can either use the current structure or load the specified structure (if it is available on the FTP server of the PDB).

# **DEER**: The arguments of this keyword supersede any spin labeling with a different label or at a different temperature that you may already have performed on the template structure (rotamers are automatically recomputed). If rotamer computation ([site scan](#)) (Section 3.1) was already performed for the specified sites (see below) with matching spin label and temperature, rotamer computation is skipped. It is not necessary to compute labels before, since version 2011 rotamer computation is fast. In any case you should not actually attach the labels before fitting.

*DEER restraint specification:* DEER restraints are specified by [MMM addresses](#) (Section 1.6) of the two residues involved, followed by the mean distance (in nm) and the standard deviation (in nm). Alternatively a lower and upper bound for the distance (in nm) can be provided by using negative numbers.

The program determines from the given residue addresses whether one residue is in the template structure (beacon restraint) or both residues are in the domain to be modelled (internal restraint). Restraints between two residues in the template structure are ignored and a warning is displayed in the Message Board of the MMM main window.

It is good practice to comment each restraint (where does it come from, functional assay, etc.). Constraints from CW EPR are currently specified as DEER constraints.

# **OLIGOMER**: You can specify modal distances (the nearest neighbor distance) between equivalent residues in different protomers of the same homooligomer. The distance is interpreted as the side length of a polygon with  $n$  vertices, corresponding to a homooligomer with  $C_n$  symmetry. The coordinate frame  $z$  axis is the  $C_n$  axis. If this feature is used for anchored loops, you must transform the template structure to this symmetry frame before you open the **Domain ensemble** window. The transformation can be achieved with the **Symmetry frame** feature in the [Edit menu](#). (Section 2.1.2)

Oligomer restraints must refer to residues inside the loop to be modelled.

*Oligomer restraint specification:* Oligomer restraints are specified by [MMM address](#) (Section 1.6) of the residue in the modeled domain (residue number is sufficient), followed by the mean distance to the equivalent residue in the first neighbor protomer (in nm) and the standard deviation (in nm). Alternatively a lower and upper bound for the distance (in nm) can be provided by using negative numbers.

# DEPTH:

The use of depth restraints requires that the z axis is perpendicular to the lipid bilayer *and* that the center of the bilayer corresponds to  $z = 0$ . For homooligomeric membrane proteins z axis alignment is best done with the `Symmetry frame` feature in the Edit menu. For other membrane proteins, it is best to first assign transmembrane helices and strands using the `Assign TM helices or strands` item in the Edit menu. In both cases the origin of the z axis can be defined with the bilayer with the `Bilayer` feature in the `Build` menu. This requires that the third-party software MSMS is installed for computing the solvent-accessible surface.

*Depth restraint specification:* Depth restraints are specified by [MMM address](#) (Section 1.6) of the residue in the modeled domain (residue number is sufficient), followed by the absolute value of the distance (in nm) from the bilayer center and the standard deviation (in nm). Alternatively a lower and upper bound for the distance (in nm) can be provided by using negative numbers.

Length of the sequence (keyword # `SEQUENCE`) and range of residues to be modelled (keyword # `DOMAIN`) must agree.

# `APROP`: The following lines up to the next keyword have two arguments each, the address of a residue with non-zero  $\alpha$ -helix propensity and the propensity (a number between 0 and 1). Note that a definition by the keyword # `HELIX` will override propensities of all residues within the helix (sets them to unity). Likewise, only one type of propensity (either  $\alpha$ -helical or  $\beta$ -strand or PPII) should be specified for a given residue, otherwise program behaviour is undefined. The program does not check whether the user has specified contradictory secondary structure definitions.

# `BPROP`: The following lines up to the next keyword have two arguments each, the address of a residue with non-zero  $\beta$ -strand propensity and the propensity (a number between 0 and 1). Note that a definition by the keyword # `STRAND` will override propensities of all residues within the strand (sets them to unity). Likewise, only one type of propensity (either  $\alpha$ -helical or  $\beta$ -strand or PPII) should be specified for a given residue, otherwise program behaviour is undefined. The program does not check whether the user has specified contradictory secondary structure definitions.

# `PPROP`: The following lines up to the next keyword have two arguments each, the address of a residue with non-zero PPII-helix propensity and the propensity (a number between 0 and 1). Only one type of propensity (either  $\alpha$ -helical or  $\beta$ -strand or PPII) should be specified for a given residue, otherwise program behaviour is undefined. The program does not check whether the user has specified contradictory secondary structure definitions.

## Monitoring distances

Sometimes restraints cannot be derived unambiguously from experimental distance distributions. Yet these distributions contain information that can be used for validating models. If the distribution contains multiple peaks, for instance from

oligomerization, a first modeling run may even reveal which one has to be assigned to the intra-protomer distance. This peak could then be converted to a restraint and the restraint be used in a second run. To facilitate such an approach, distances and depths can be monitored, i.e. their distribution over the whole ensemble can be displayed after modeling. For this, the potential restraint (DEER, oligomer, or depth) has to be specified in the restraint file, but with a mean distance and standard deviation of zero (see example below).

## Example restraint files

**Active form of pro-apoptotic protein Bax in liposomes**, see: S. Bleicken, G. Jeschke, C. Stegmüller, R. Salvador-Gallego, A.J. García-Sáez, E. Bordignon, (2014) *Structural Model of Active Bax at the Membrane* Mol. Cell **56**, 496-505 (2014)

```
% Restraints for Bax active form residues 123-149

# PDB BAXC

# DOMAIN 123 149

# NANCHOR [BAXC] (A)122

# SEQUENCE KALSTKVPELIRTIMGWTLDFLRERLL

# ENSEMBLE 20 0.5

# HELICES

130-146

# DEER MTSL 298

(A)126 (A)55 3.83 0.38 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)126 (A)62 3.13 0.39 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)126 (A)72 2.28 0.40 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)126 (A)87 2.67 0.31 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)126 (A)101 3.71 0.62 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)126 (A)149 3.86 0.39 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)149 (A)55 4.60 0.86 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)149 (A)62 5.43 0.48 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)149 (A)72 5.25 0.58 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

(A)149 (A)87 4.33 0.37 % Bleicken et al. Mol. Cell 56, 496-505 (2014)
```

```

(A)149 (A)101 4.42 1.01 % Bleicken et al. Mol. Cell 56, 496-505 (2014)

# OLIGOMER 2 MTSL 298

% only monitoring

126 0 0 % distribution exists, hard to assign

149 0 0 % distribution exists, slightly ambiguous

# END

```

**N-terminal domain of major plant light harvesting complex LHCII**, see: N. Fehr, C. Dietz, Y. Polyhach, T. von Hagens, G. Jeschke, H. Paulsen, *submitted*

```

% Restraints for N-terminal domain residues 3-13 of LHCII, Fehr et al.

# PDB 2BHW:A

# DOMAIN 3 13

# CANCHOR [2BHW] (A)14

# SEQUENCE SATTKKVASSG

# ENSEMBLE 20 0.5

# DEER IA-PROXYL 298

3 34 2.66 1.60 % N. Fehr, heterogeneous trimer
3 59 1.50 1.66 % N. Fehr, heterogeneous trimer
7 34 2.67 1.62 % N. Fehr, heterogeneous trimer
7 59 1.92 1.69 % N. Fehr, heterogeneous trimer
11 34 2.45 1.56 % N. Fehr, heterogeneous trimer
11 59 1.50 1.73 % N. Fehr, heterogeneous trimer

# OLIGOMER 3 IA-PROXYL 298

3 3.65 1.28 % A. Müller, C. Dockter, singly labeled trimer
4 3.72 1.21 % A. Müller, C. Dockter, singly labeled trimer
7 4.00 1.18 % A. Müller, C. Dockter, singly labeled trimer
9 4.31 1.34 % A. Müller, C. Dockter, singly labeled trimer
10 4.64 1.46 % A. Müller, C. Dockter, singly labeled trimer

```



```

11 4.57 1.17 % A. Müller, C. Dockter, singly labeled trimer
12 4.96 1.36 % A. Müller, C. Dockter, singly labeled trimer

# DEPTH CA

3 -1.87 -3.5 % A. Volkov, C. Dockter, deuterium ESEEM
4 -1.75 -3.5 % A. Volkov, C. Dockter, deuterium ESEEM
7 -1.79 -3.5 % A. Volkov, C. Dockter, deuterium ESEEM
9 -1.75 -3.5 % A. Volkov, C. Dockter, deuterium ESEEM
10 -1.89 -3.5 % A. Volkov, C. Dockter, deuterium ESEEM
11 -1.73 -3.5 % A. Volkov, C. Dockter, deuterium ESEEM
12 -2.06 -3.5 % A. Volkov, C. Dockter, deuterium ESEEM

# END

```

Note that in the second example, depth restraints are defined as lower and upper bounds for the C $\alpha$  z coordinate. These restraints are based on a correlation between the deuterium ESEEM water accessibility parameter  $\Pi(\text{D}_2\text{O})$  and this coordinate established on spin-labeled residues in the resolved part of the LHCII structure.

## Options and runtime information

Apart from specifying restraints, the user has some further control on how the ensemble is computed. This concerns the following options:

### *Ramachandran statistics*

By default, the program uses digitized versions of the residue-specific Ramachandran plots from Hovmöller *et al.*, 2002 corresponding to only protein sections without secondary structure. The user can select the statistics from the same paper that includes sections with secondary structure, but note that spontaneous formation of helices and strands in a Monte Carlo simulation is very unlikely. If secondary structure is suspected, the user needs to provide restraints for it, which can come, e.g., from spin-labeling EPR site scans or from NMR residual dipolar couplings or chemical shifts.

### *Ensemble probability*

Restraints can be defined as lower and upper bounds. In this case the restraint is fulfilled in all accepted structures and contributes a probability density of 1. In most cases it is better to define the restraints by a Gaussian distribution with mean value  $\langle r \rangle$  and standard deviation  $\sigma_r$ . In this case the restraint contributes a probability density  $p_i = \exp\{-[(r - \langle r \rangle)/\sigma_r]^2\}$ , where  $r$  is the distance between the predicted mean

distance between spin label positions for the current conformation. The  $p_i$  for all Gaussian restraints are multiplied,  $P = \prod_i p_i$ . A simulated conformation is rejected as soon as  $P$  falls below a threshold value  $P_{\text{thr}}$ . This threshold value roughly ensures that the accepted models span the region in space where 50% of all conformations are located that are consistent with the restraints. The 50% level is the same that is used for visualizing thermal ellipsoids in crystal structures. It corresponds to a value of 0.5 for the ensemble probability.

Lower values than 0.5 produce a narrower ensemble, larger values a broader ensemble. If all restraints are consistent, a probability of 0.5 is a good choice. In the restraints are uncertain or ambiguous or distance distributions have been misinterpreted, a choice of 0.5 may not produce any accepted conformations or only very few of them. In order to track down the problematic restraints, it may be useful to produce a large number of conformations that have a lower probability for fulfilling all restraints. Usually, 0.75 is a good choice. Note that a choice of 1.00 would produce an ensemble that is "blind" to all Gaussian restraints, whereas a choice of 0 makes it practically impossible that any computed conformation will be accepted. Hence, the control parameter is limited to the range between 0.05 and 0.95.

#### *Sidegroup clash threshold*

After a backbone model has been decorated with sidegroups by SCWRL4, it is tested for clashes between heavy atoms within the simulated domain as well as between the simulated domain and the template structure. In most cases the default sidegroup clash threshold of 2.00 Å is fine for this test. However, if the domain is tightly packed against the protein it is unlikely that a Monte Carlo algorithm will find the properly packed structure. This is also because the algorithm does not recompute sidegroup packing by SCWRL4 after attachment of the domain to the template, but for the isolated domain before attachment.

As the algorithm is not intended for finding well-defined structures, but rather conformation ensembles for disordered domains, this should not pose a big problem. However, if a set of restraints that are considered to be reliable does not produce any models because of sidegroup clashes, it may be advisable to lower the threshold to 1.3 Å or even 1.0 Å, corresponding to the forgive factors that we typically use for the similar problem of attaching rotamers from a coarse library to a protein structure. A future version may provide the option to repack side groups on the complete model after domain attachment if clashes are observed.

#### *Ensemble size*

The ensemble size specified by the restraint file can be changed *after* the restraint file had been loaded. Please note that in the default `attach models` mode, for each conformation the template structure is copied. During testing, we have not encountered memory problems for ensembles of up to 500 models. However, the limit depends on the size of the template structure. Usually it is not necessary to generate more than 100 or 200 conformations.

If the domain is disordered indeed and the set of restraints is consistent, ensemble generation is reasonably fast. If the structure is rather well-defined (many narrow restraints) or restraints are inconsistent, it may take very long to generate the requested number of models. Computation time can be limited in the **Ensemble size** panel. Please note that the default value of 2 hours may have to be increased if more than 20 models are requested or if a closed loop is computed, in particular, if the last or second-to last residue is a proline.

Alternatively, the maximum number of Monte Carlo trials can be used to limit total computational effort. The default value of 50'000'000 has been sufficient in all cases that we tried and corresponds to very long computation times with a single processor core even for short domains.

### *Post processing*

When the program generates acceptable backbone coordinates, they are saved as a PDB file, the backbone is decorated with side groups by SCWRL4, which generates another PDB file, the model including side groups is tested for internal clashes and clashes with the template, and if it survives this test, it is attached to the template, displayed as coil graphics (crimson) and the C-terminal C $\alpha$  atoms is displayed as a space-filling model (blue). For N-terminal loops, the N-terminal C $\alpha$  atom is displayed instead and for free standing loops, the N-terminal C $\alpha$  atom is displayed as a dark green sphere and the C-terminal C $\alpha$  atom as a blue sphere.

Displayed models have their weight in the ensemble encoded by transparency. The PDB files of the individual conformations are saved in the current Matlab directory. These default settings provide the user with the full information, prevent that files are later deleted during garbage collection in the MMM temporary directory (after 30 days), and generate a structure that can be saved as a PDB file for the whole ensemble from the **File** menu in the MMM main window. The log file is generally saved in the directory from where the restraint file was coming and its name is generated from the name of the restraint file and the date and time when the run was started.

If the individual conformations are automatically attached to the template and the structure is later saved from the main window, the individual PDB files are not anymore required. When the **save to current directory** checkbox is deactivated, they are saved in the MMM temporary directory and disposed of automatically after 30 days. Note also that the individual PDB files are overwritten if another run for the same domain and the same template is started.

Automatic display of coil models and automatic display of the C-terminal C $\alpha$  atom can be suppressed by deactivating the corresponding checkboxes.

Attachment of the individual conformations to the template can also be deactivated. This has the advantage that the template structure remains unchanged, thus allowing for further runs without reloading the template. For the combination of very large ensembles with very large template structures, memory problems that may arise from copying the template for each individual model can be avoided. Note,

however, that without attachment distance and depth distributions are not computed for the whole ensemble, except for free standing loops.

For free standing loops, attachment is not possible. If the `attach models` checkbox is activated in such a computation, the conformations are collected in a single structure that can later be saved from the main window as a PDF file of the ensemble model.

## Runtime information

During the run, some diagnostic information is updated after each block of 10'000 Monte Carlo trials. Thus, the user always has an overview what fraction of the maximum time and number of Monte Carlo trials have been spent, how many loop models (valid conformations) have been found, and how many valid backbone models have been found.

Furthermore, it is reported what percentage of Monte Carlo trials was unsuccessful due to restraint violations. If the ensemble is well restrained, this number should be close to 100%. In fact, in some cases the probability that a trial fulfills all restraints is less than  $10^{-4}$  (less than one per block), which leads to the display of 100% restraint violations.

The generation of a backbone model may also fail due to internal clashes of backbone atoms in the domain or clashes of backbone atoms with the template. The percentage of Monte Carlo trials where this happened are also displayed.

Among the valid backbone models, some will clash internally and some will clash with the template after side group attachment. The percentages given for these rejections are relative to the number of valid backbone models found, not to the number of Monte Carlo trials.

For closed loops, the algorithm may sometimes fail to bring the last backbone atom into the convergence radius of the C-terminal anchor residue or it may be impossible to find a set of allowed Ramachandran values for the last residue that preserves restraint fulfillment and is consistent with allowed Ramachandran values of previous residues in the domain. The latter problem arises especially in cases where the last or second-to-last residue of a closed loop is a proline. The percentage of such cases is reported relative to the total number of Monte Carlo trials.

Finally, a plot reports how the rejection probability of models was distributed over the residues. A high value for a particular residue indicates that the restraints specified for this residue were particularly hard to fulfill. In cases where the restraints are nearly uniformly distributed along the chain, as in the LHCII example above, a very high probability for a single residue may indicate that this residue is overrestrained, possibly because standard deviations or bounds are too narrow, or is inconsistently restrained, because one or more restraints are simply wrong. There may also be cases where almost all rejections occur at a single residue for good reasons. The Bax example above is such a case. Only residues 126 and 149 are restrained and as the domain is C-terminal, residue 126 is encountered first. Because this residue is well restrained (without being overrestrained), most Monte Carlo trials fail already

there. The rejection probability at residue 149 is lower simply because only for a small fraction of all Monte Carlo trials residue 149 is ever reached.

For closed loops, the center residue tends to have an enhanced rejection probability. This is a feature of the algorithm, which tests the backbone for internal clashes and clashes with the template before the second half of the domain is generated. This enhances computational efficiency, since generation of the second half of the loop is computationally much more expensive for closed loops that have to "home in" on the C-terminal anchor residue.

## Inspecting restraint fulfillment

It is good practice to check how well the individual restraints are fulfilled by the ensemble. If there were no implicit restraints from bond lengths, bond angles, allowed Ramachandran angles, or the necessity of closing the loop and if the simplified prediction of the spin label coordinate from the backbone coordinates of the residue would be perfect, distance distributions computed from the ensemble should match the Gaussian distributions provided as restraints. Usually, this is observed only for part of the restraints.

One reason is that during backbone generation, side group coordinates are not available, which makes prediction of the spin label coordinate less certain. After the whole domain has been modeled and the side groups have been attached, the distance distribution is recomputed with the rotamer library approach. The distance distribution for the ensemble is the probability-weighted sum of such distance distributions for the individual conformations. If the neighboring side groups impose a systematic shift on the spin label position of a restrained residue, the restraints for this residue may not exactly be matched.

While this is a deficiency of the algorithm, in practice the more likely cause for deviations are uncertainties in the experimental distance distribution or in deriving a Gaussian restraint from this distribution. Hence, the simulated distributions for the ensemble are best compared to the raw distance distributions as they can be obtained, for instance, with [DeerAnalysis](#). For that purpose, DeerAnalysis result files can be imported by using the [Load](#) button above the plot.

## Loading restraints

The [Restraints](#) button allows for loading a restraint file. Please watch the [Message Board](#) of the MMM main window, as any problems will be reported there. After loading is completed you can use the [Show log](#) button in the main window to inspect all previous output to the [Message Board](#). If the restraint file has correct syntax, provides all mandatory information, and is formally consistent, the [Run](#) button is activated. The [Restraints](#) button is then deactivated. If you want to simulate with other restraints, you need to close and reopen the window.

## Running the computation

The `Run` button becomes active only after loading restraints. When it is clicked the computation starts. After each block of 10'000 Monte Carlo trials the `Runtime information` (see above) is updated.

The computation stops when the requested number of models (conformations) has been generated, the maximum allotted time has been passed, or the maximum number of Monte Carlo trials has been spent (whichever condition is fulfilled first). After a successful run, the `Rejection distribution along the loop` plot changes to a `Restraint fulfillment` plot.

Note that the template structure has been "spent" after a run, unless you had deactivated the `attach models` checkbox, which is discouraged. If you wish to create another ensemble from the same template, you need to close the window, reload the template structure in the main window, and start over.

## Closing the window

By clicking the `close` button the window is closed. All information is lost, except for saved single conformations, the structure with the attached domains, and the log file of ensemble generation.

## 3.7 Coarse graining

### What is coarse graining?

The structure of a biomacromolecule is fully specified by the coordinates of all atoms (we neglect partial charges in this discussion). However, such a representation may not always be appropriate. First, the available experimental data may not allow to specify all atom positions. For instance many x-ray structures of proteins are arguably overspecified compared to the information that is really inherent in the electron density cloud (or diffraction pattern) at the given resolution. Second, biomacromolecules under physiological conditions are inherently dynamic and in an ensemble of molecules of the same primary structure, atoms will be at slightly different positions in individual molecules. For instance it will rarely if ever happen that the sidegroup of a given residue at the surface of a protein has the same conformation in all molecules of the sample. Third, when considering long-time dynamics or large-scale structural transitions, it may not be feasible to perform simulations at full atomic detail and it may not be necessary either, as on such time scales and during such changes details of atomistic structure vary and may not always be of interest (exception: binding sites).

In all these cases it is more appropriate to consider models with lower resolution. However, given the wealth of information inherent in the atomistically defined structures of the protein data bank (PDB) and the sophisticated parametrization of atomistic force fields for molecular dynamics computations, it is often possible and



most appropriate to *start from* an atomistic model. Such a model is coarse grained by combining several atoms into a pseudo-particle. For instance, each amino acid residue can be represented by two pseudo-particles, one for the backbone and one for the sidegroup. On the next level of coarse graining it could be represented by a single pseudo-particle, positioned at the coordinate of the C<sup>&alpha;</sup> atom. Further on, several residues could be combined into one collectively moving entity.

This combination of several particles into a single pseudo-particle reduces the number of degrees of freedom of the model and the number of pairwise interactions. Hence, the model can be specified from fewer experimental constraints and simulation of model dynamics becomes much faster. Already when going from an atomistic model to one pseudo-particle per residue the savings are huge.

The downside of coarse graining is that the model becomes less precise, not only in terms of resolution, but also in terms of interactions between the (pseudo-)particles. The true pairwise interaction potentials between the pseudo-particles are complicated, anisotropic, and may even depend on structural context. In practice, simplified potentials are used that strike a compromise between ease of use and precision. At a given level of sophistication, ultimate precision is obtained by parametrizing the potentials on the basis of atomistic force fields for the very system under study. In other words, coarse grained models are either relatively precise or relatively universal.

The molecular dynamics community strives for ultimate precision in coarse graining at the cost of universality. MMM follows biophysical approaches that value universality more than precision, for the reason that it is concerned with deriving structures and structural transition pathways from scarce experimental constraints. Scarcity of constraints already imposes a limit on precision. Furthermore, experience with long-range distance measurements by EPR spectroscopy has shown that in a number of cases the width of distance distributions can be explained only when assuming flexibility of the backbone of the biomacromolecule. In such cases the nature of the problem itself poses a limit to the precision of the model.

## Coarse graining approaches in MMM

The current version of MMM provides isotropic and anisotropic [Elastic network models](#) (Section 3.7.1) and simple [lipid bilayer](#) (Section 3.13) models. Generally, proteins are coarse grained to a single pseudo-particle per residue positioned at the C<sup>&alpha;</sup> coordinate of the atomistic model. Bilayers are infinitely extended and homogeneous parallel to the bilayer plane and feature variation of (residue-specific) interaction potentials and of chemical composition along the bilayer normal, for instance distribution profiles of water, oxygen, and phosphorous nuclei of phospholipid headgroups.

All interaction potentials are universal, with the exception of interaction of residues with a lipid bilayer that depends on secondary structure of the residue. Even in that case, the interactions can be computed from the atomistic structure without specific parametrization.



## Fine graining

MMM allows for reconvertng a coarse-grained model to an atomistic model in the following way. A local frame is defined by the  $C^{\alpha}$  coordinates of residues  $i-2$ ,  $i-1$ ,  $i$ ,  $i+1$ , and  $i+2$  (at least three of these residues are needed). The translation and rotation of this local frame between the original atomistic *template structure* and the current coarse-grained structure is determined by least square superposition of the  $C^{\alpha}$  coordinates of the template structure onto the coordinates of the coarse-grained model. Now all atoms of residue  $i$  are subject to this translation and rotation.

The user should realize that this involves approximations. For instance, if the  $C^{\alpha}$  network has deformed, preferred sidechain conformations may have changed. To account for this, sidechains can be repacked using [SCWRL4](#), (Section 5.2) but note that this again involves approximations.

Such fine graining may be necessary to perform certain tests on the model, to compare it to certain types of experimental data, or to update other approximate information, for instance on spin label conformational distribution. The user should be aware that the full specification of all atom coordinates in a fine-grained model is to some extent an illusion, in fact, to a larger extent than it was already an illusion in experimental template structure used at the start for coarse graining. Starting from a homology model may be an exception. In this case an experimentally informed fine-grained  $C^{\alpha}$  model may well be more precise than the template structure.

## 3.7.1 Elastic network models

### Introduction

An elastic network model (ENM) represents each particle of a system as a point-like node and interactions between particles by a spring with a certain force constant. For visualization you can think of small metal balls connected by springs. In the context of MMM the particles are actually pseudo-particles, corresponding to amino acid residues. The location of the network nodes (balls) is at the  $C^{\alpha}$  coordinate of the residue.

There exist several different ways of assigning the force constants. A very popular choice is to use a uniform force constant up to a certain cutoff radius, which is typically 7.3 Å for residue-based [Gaussian network models](#) (Section 3.7.2) and about 13 Å for residue-based [Anisotropic network model](#) (Section 3.7.3) models (defined in global variables `ENM_param.rc` and `ENM_param.rc_ANM` that are set in `initialize MMM.m`). Although MMM has setup subroutines `setup_GNM.m` and `setup_ANM.m` for such cutoff-based ENMs and will use them for perturbation response scanning (C. Atilgan, Z. N. Gerek, S. B. Okzan, A. R. Atilgan, Biophys. J. 2010, 99, 933-943), all functions currently supplied through the graphical user interface use a polynomial dependence of force constants on pair distance  $r$ . This reflects recent discussions that indicate better performance and physical foundation of this choice (L. Yang, G. Song, R.-L. Jernigan, Proc. Natl. Acad. Sci. USA 2009, 106, 12347–12352; K. Hinsen, Proc. Natl. Acad. Sci. USA 2009, 106, E128). In particular, an  $r^2$

dependence is used for the isotropic Gaussian network models, as this provides best prediction of crystallographic B factors and highly resolved [covariance matrices](#). (Section 3.7.2) For anisotropic network models an  $r^{-6}$  dependence following (K. Hinsen et al. Chem Phys. 2000, 261, 25-37.) is used, whereas force constants of first and second neighbors in the peptide sequence are amplified by a factor of 10000 to stabilize peptide bond geometry (this is justified by the narrow distribution of first and second neighbor distances in known protein structures).

Vibrational dynamics of an ENM with  $N$  nodes, corresponding to a peptide chain with  $N$  residues, is easily computed by setting up the Hessian matrix from the force constants and solving the eigenproblem of this matrix. The first six eigenvalues are zero and the corresponding eigenvectors describe global rotation and translation of the whole network. The remaining  $3N-6$  eigenvalues are all positive and are squared vibration frequencies of the *normal modes*. The eigenvector of each normal mode describes the amplitude and- for anisotropic network models- the direction of relative movement of the nodes with this particular vibrational frequency.

The relevance of such normal mode analysis derives from the fact that slow modes, i.e. the modes with the lowest vibrational frequencies, contribute most to large amplitude vibrations. They correspond to deformations for which the network is "soft". It stands to reason and has been tested for many pairs of known protein structures that large-amplitude structural changes of proteins proceed mainly along a few slow modes.

For this reason, slow normal modes can be used to visualize potential large-amplitude dynamics. They can also be used as a basis for [fitting structural transitions](#) (Section 3.7.4) from a small number of distance constraints (W. Zheng, B. R. Brooks, Biophys. J. 2006, 90, 4327-4336).

see also: M. M. Tirion, Phys. Rev. Lett. 1996, 77, 1905; I. Bahar, T.R. Lezon, A. Bakan, I.H. Shrivastava, Chem. Rev. 2010, 110, 1463-1497.

## Limitations

ENMs are coarse-grained models. If they were originally derived from an atomistic template structure, they can be fine-grained to atomistic models even after deformation, but you should be aware that such a fine-grained model is only a rough approximation for true atom positions.

Further limitations, even on the level of  $C^\alpha$  traces arise from the approximations inherent in modeling- the rather crude assignment of universal force constants and the assumption of harmonic motion close to each point where a network model is set up and analyzed.

If an ENM is used for fitting of structural transitions, you cannot expect that the final structure has a  $C^\alpha$  r.m.s.d. better than 1 Å, even if there are plenty of constraints with very small errors in the constraints. Typically you should expect errors of 2-3 Å. This limitation arises mostly from the restriction to a small number of slow modes. If all  $3N-6$  modes could be used, perfect agreement would be achieved in the absence of

errors in the constraints. However, this would require that  $3N-6$  linearly independent constraints are known, which is unrealistic.

In practice, errors in the distance constraints limit precision, so that it does not make sense to use a large number of normal modes for fitting. Typically, a single chain can be fitted with about 10 modes (W. Zheng, B. R. Brooks, *Biophys. J.* 2006, 90, 4327-4336).

A limitation on network size (residue number) arises from the need to set up and diagonalize the Hessian matrix. With a contemporary PC or laptop computer, systems with up to 2000 residues are accessible.

## 3.7.2 Gaussian network models

### Introduction

A Gaussian network model (GNM) is a special [Elastic network model](#) (*Section 3.7.1*) that considers only the amplitude of residue fluctuations, but not their direction. For instance, a GNM can predict the B factors (temperature factors) in crystal structures, while a more involved [Anisotropic network model](#) (*Section 3.7.3*) would be required to predict anisotropic temperature factors. With respect to an ANM a GNM has the advantages that normal mode analysis is 27 times faster and that interpretation is easier, as fluctuation of each residue and correlation of each residue pair are characterized by a single number, instead of a  $3 \times 3$  matrix.

In MMM, GNMs are used to aid domain assignment, diagnose ENM performance via comparison of experimental and predicted B factors, and to recognize dynamic correlations between residues via the [covariance matrix](#).

Once computed, the GNM is stored with the structure and is saved, when the whole MMM model is saved. Likewise, the  $C^\alpha$  network (coarse-grained structure) is stored and saved with the model. Hence, on repeated activation the GNM window opens much faster than on first activation.

### GNM computation in MMM

An GNM for the current structure is computed and an inspection window is opened via the GNM menu item in the [Dynamics menu](#). (*Section 2.1.6*). Note that computation can take between a few seconds and a few minutes, depending on size of the structure. Computation time scales with the third power of the number of amino acid residues. Computation is always for the current structure, which can be picked in the [hierarchy window](#). (*Section 2.3*)

MMM uses force constants for residue pairs that scale with the inverse square of the  $C^\alpha$ - $C^\alpha$  distance between the two residues. This provides better agreement of predicted and experimental B factors than other ENM-based approaches (L. Yang, G. Song, R.-L. Jernigan, *Proc. Natl. Acad. Sci. USA* 2009, 106, 12347–12352) and leads to a well resolved [covariance matrix](#).

## Limitations


Elastic network models have some tolerance against missing residues in the core of a protein, however, for quantitative work you may want to have a continuous peptide chain. If the template structure is incomplete (missing stretches of residues), consider (homology) modeling or other reconstruction, for instance with the Modeller software. Although such modelling is worse than having reliable experimental data, it is certainly better than ignoring the problem.

Computation of a GNM can fail for more than 6000 residues in the structure due to memory limitations with a typical desktop or laptop computer at the time of writing. Use a computer with larger memory available to Matlab to avoid this problem.

## Covariance matrix

In a GNM, each residue has only one isotropic coordinate  $q_i$ . For a pair of residues  $i$  and  $j$  we now consider correlated coordinate changes, i.e., we ask the question whether the coordinates  $q_i$  and  $q_j$  move strongly and in the same direction in a given normal mode. We sum over all normal modes, weighing them with the inverse corresponding eigenvalue to account for different vibration amplitudes. The result  $\langle \Delta q_i \Delta q_j \rangle$  is the matrix element  $C_{ij}$  of the covariance matrix  $\mathbf{C}$ . This covariance matrix can be computed from the eigenvectors and eigenvalues of the Hessian. The diagonal elements are proportional to predicted B factors.

Large positive elements of the covariance matrix correspond to strongly correlated motion of residues  $i$  and  $j$ , the residues belong to the same domain or connect domains. Negative values indicate anticorrelated motion, the residues belong to different domains. Small absolute values indicate that motion of the residues is (nearly) uncorrelated.

A two-dimensional plot of the covariance matrix provides a fast overview of domains in a protein and allows to pinpoint residue-residue contacts that are crucial for transmitting mechanical perturbations between remote parts of the structure. For this reason, such a plot is shown in the upper left corner of the GNM window. In select mode, correlation peaks (black off-diagonal points in the plot) can be identified (addresses of the correlated residues are displayed in the text line above the plot) and the corresponding residues are selected (and highlighted in the 3D model). In the inspect mode the residues are identified, but not selected. Use the  [View selected](#) (*Section 2.1.11*) tool button in the main window to zoom into the contact in the 3D model window. In zoom mode, a detail of the covariance matrix can be visualized by zooming into the plot (left click into the plot and hold left mouse button down, drag mouse to create a detail rectangle, release left mouse button when the rectangle has the proper extension).

For multi-chain protein complexes, diagonal blocks of the covariance matrix that correspond to individual chains can be marked with the toolbar button. Existing selections can be cancelled with the toolbar button.

The covariance matrix plot is also linked to the auxiliary plot window (right upper corner of the GNM window), if the latter plot is used for display of individual normal modes. In this case, range selections made in the normal mode are also shown as dashed green lines in the covariance matrix.

## Mode distribution

The auxiliary plot (upper right corner of the GNM window) can display different types of information. Immediately after the window is opened, the distribution of vibration frequencies is displayed (normalized vibration frequency versus mode number).

## B factors

B factor prediction can be displayed in the auxiliary plot by selecting the corresponding item from the popup menu. If the current structure is a crystal structure and B factors are defined, experimental B factors are shown as a black line. Predicted B factors are shown as a red line and are scaled and shifted by a constant offset for optimum superposition with the experimental B factor. The constant offset should be zero according to theory and is often quite small, but in some cases the experimental B factors obviously do have such an offset.

## Individual normal modes

Residue fluctuations in individual normal modes can be visualized in the auxiliary plot by selecting the **Single mode** item of the popup menu. The black line is a plot of the normalized eigenvector. Residues with large positive or negative amplitudes move strongly in this mode, residues with zero amplitude do not move. Residue pairs with the same sign of the amplitude move in a correlated way in this mode, while residues with different signs move in an anticorrelated way. Often the first mode identifies the domains, while the next slow modes identify subdomains. Note that mode numbers in GNM can not be directly compared to mode numbers in ANM, which have by a factor three more modes.

In the single-mode display of the auxiliary plot, ranges of residues can be marked by left-clicking (lower limit of range) or right-clicking (upper limit) into the plot. If desired, the range can be automatically optimized to a range of residues with the same sign of amplitude by clicking the **Optimize range** button. All residues in the range can be selected for domain selection or definition by clicking the **Select range** button. Alternatively, the residues can be marked by the same color in the 3D model window by clicking the **Color** button for domain visualization.

## 3.7.3 Anisotropic network models

### Introduction

An anisotropic network model (ANM) is a special [Elastic network model](#) (Section 3.7.1) that considers both the amplitude and direction of residue fluctuations. Hence,

the ANM can be used for visualizing the motion of the C $\alpha$  trace in individual normal modes. This provides an impression of likely large-scale motions of the protein and a dynamic visualization of partitioning of the protein into domains.

In MMM, ANMs are used for such visualization and for [fitting structures](#) (Section 3.7.4) from an existing template structure. The ANM window is used for inspecting the ANM and for normal mode visualization. This window also allows to save the coarse-grained model (C $\alpha$  trace) into an individual Matlab variable.

Once computed, the ANM is stored with the structure and is saved, when the whole MMM model is saved. Likewise, the C $\alpha$  network (coarse-grained structure) is stored and saved with the model. Hence, on repeated activation the ANM window opens much faster than on first activation.

## ANM computation in MMM

An ANM for the current structure is computed and an inspection window is opened via the ANM menu item in the [Dynamics menu](#). (Section 2.1.6) . Note that computation can take between a few seconds and about an hour, depending on size of the structure. Computation time scales with the third power of the number of amino acid residues and is about 27 times longer than for a [Gaussian network model](#). (Section 3.7.2) . Computation is always for the current structure, which can be picked in the [hierarchy window](#). (Section 2.3)

## ANM types in MMM

The type of ANM can be selected in the [Preferences window](#), (Section 2.1.1) accessible via the [File menu](#). In all cases network nodes are at the positions of C $\alpha$  atoms, only computation of the force constants varies. The *native* ANM of MMM uses force constants that scale with the inverse sixth power of the C $\alpha$ -C $\alpha$  distance between network nodes and for direct neighbor and next neighbor residue pairs force constants are amplified by a factor of 10000 (Jeschke parametrization). This is somewhat unphysical, but stabilizes peptide bond geometry during fitting and avoids the relaxation step in fitting with the Zheng/Brooks algorithm (W. Zheng, B. R. Brooks, Biophys. J. 2006, 90, 4327-4336).

The *default* ANM parametrization of MMM is ed-ENM by Orellana et al., J. Chem. Theor. Comput., 2010, 6, 2910-2923. Further choices include a parametrization very similar to Hinsen's (Hinsen et al., Chem. Phys. 261 (2000) 25) and uniform force constants with cutoff at 10 or 13 Å. Finally, ed-ENM-p is offered as an implementation of ed-ENM without cutoff distance. If in doubt, keep with the default.

## Limitations

Elastic network models have some tolerance against missing residues in the core of a protein, however, for quantitative work you may want to have a continuous peptide chain. If the template structure is incomplete (missing stretches of residues), consider (homology) modeling or other reconstruction, for instance with the Modeller



software. Although such modelling is worse than having reliable experimental data, it is certainly better than ignoring the problem.

Computation of an ANM can fail for more than 2000 residues in the structure due to memory limitations with a typical desktop or laptop computer at the time of writing. Use a computer with larger memory available to Matlab to avoid this problem.

## C<sup>α</sup> wireframe model

The left half of the ANM window displays the C<sup>α</sup> wireframe model for normal mode visualization. By default, the wireframe is colored according to [colorscheme chain](#), (Section 2.2) i.e. different chains in a protein complex are distinguished by color. Alternatively, you can use a neutral grey color for the whole model.

The wireframe model display is linked to the full 3D model display of MMM. When you rotate the model in the 3D model window, the wireframe will rotate the same way. That way you can easily compare, side-by-side, normal mode motions to any type of static visualization that is possible within MMM. This is why normal mode visualization was implemented in MMM. For producing pretty cartoon animated GIF normal mode visualizations (via MolScript) you might consider the [ElNémo server](#) instead. That said, normal mode visualizations can also be [saved as AVI movies](#) from MMM's ANM window, with limited visualization quality. The ANM window provides more flexibility and interactivity in visualizations than web servers. Hence, you may want to inspect potential protein dynamic behavior by MMM and make animations for presentations with a web server.

## Saving the coarse grained model

The coarse-grained model can be saved into a Matlab file that contains only this variable by clicking on the **Save** button. This is a feature for experimenting with the program and is not needed by the normal user. This button may vanish in future versions. The ANM and coarse-grained model are automatically saved when the whole MMM model is saved from the main window.

## Animation

Animation is accessible via the **Animation control** panel. Select the mode to be visualized. Default behavior is that the vibration is shown with moderate amplitude, scaled by the inverse square vibrational frequency, seen from the same viewing perspective as the static model in the 3D model window. First the visualization is computed frame by frame, then it is played back by default for five full oscillations. This behavior gives you an impression on relative vibration amplitudes in the different modes. It may not be ideal for visualizing motion in higher (faster) modes, as these have very low scaled oscillation amplitudes.

Amplitude of the oscillation can be changed by the same factor for all modes using the **Amplitude** slider. Relative amplitudes can be changed by selecting the **thermal**



(default) or `unscaled` radio buttons. The latter choice provides better visualization for fast modes.

The oscillation can be superimposed with rotation of the model about the x axis (along viewing direction, y (horizontal axis in screen plane), or z axis (vertical axis in screen plane). A full rotation (360°) is superimposed with the number of oscillation cycles requested. Note that computation of animations with superimposed rotation takes much longer, as with rotation frames for all oscillation cycles must be computed, while without rotation only frames for a single oscillation are computed.

You can choose between grey and black background with the `black bckg.` checkbox. Interactive animation is started by clicking on the `Animate` button. Note that the animation is not stored. It is recomputed when you click this button repeatedly, even if parameters are unchanged.

### Movie of an animation in AVI format

You can produce and store a movie of an animation in AVI format by clicking the `Save AVI` button. The animation should be set up before (and preferably tested on screen) as [described above](#). Quality of Matlab-generated AVI movies is rather low. The next version of MMM will allow for saving a trajectory as a PDB file with several models, for visualization in UCSF Chimera, or concatenated PDB files, for visualization in VMD. Both programs can create high quality movies.

## 3.7.4 Fit from template

### Introduction

This fit procedure can be used to test for consistency between a structural model and (EPR) distance constraints, to refine a model by distance constraints, and to characterize large-scale structural changes of a protein or protein complex, if the structure in one state and distance constraints for another state are known. In addition, the fit procedure can be used for testing how well a pair of known structures is connected by motion along only a few normal modes of an [elastic network model](#). (*Section 3.7.1*)

The  $C^\alpha$  trace r.m.s.d. between the "true" structure and the fitted model can be as small as 1.5 Å in ideal cases, but typically r.m.s.d. between 2.5 and 4 Å should be expected. A full atomistic model of the final structure is reconstructed, but note that this is based on rigid-body transformation of sidechains from the template structure with respect to the backbone.

We emphasize that the concept of a single "true" structure of a protein or protein complex is largely fiction. Nevertheless *some parts* of a protein are fixed at atomic resolution. The models obtained with this coarse-grained fit procedure should not be discussed in terms of such atomistic detail. Only speculations are possible on this level. The models can be discussed in terms of large-scale differences between structures or large-scale movements exceeding coordinate changes by about 2 Å.

## Specification of constraints and fit parameters

Fits are controlled by an ASCII text input file that is opened via the `Load constraints` button. Comment lines or comments in command lines are introduced by the percent character `%`. Keyword lines start with the hash character `#`. There must be a space between the hash character and the keyword. Two types of keywords exist, single-line keywords (all parameters are specified on the same line as the keyword) and block keywords (some parameters follow in additional lines, this is used for constraint specification). Blocks end when the next keyword is encountered. The last block must be followed by an `# END` keyword line. Keywords are not case sensitive. Keywords and arguments are separated by spaces.

Currently, the following keywords are defined:

- `# PDB` (single-line), argument: PDB identifier of template structure
- `# basis` (single-line), argument: number of slow [ANM](#) (*Section 3.7.3*) modes to be used as a basis
- `# DEER` (block), arguments: spin label and temperature (in K) for rotamer simulation
- `# direct` (block), no arguments
- `# ensemble` (single line), arguments: size of an ensemble of fit structures to be generated, (optional) estimated uncertainty of constraints, (optional) specifier `all` for using all constraints for each structure (see Remarks below)
- `# target` (single line), argument: PDB identifier of the target structure, used only for test computations
- `# TIF` (single line), argument: threshold for active space extension, to be changed only by experts
- `# MMAX` (single line), argument: iteration cycle where active space extension is complete, to be changed only by experts
- `# END` (single-line), no arguments

### Remarks

Future versions may have additional keywords. If you are a programmer and want to implement keywords of your own, please contact us before. The constraint file is processed by function `rd_restraints.m`.

`# PDB`: Currently this is only used for a check. The current structure in MMM, which can be picked in the [hierarchy window](#), (*Section 2.3*) is used as the template structure. If the current structure in MMM has a different PDB identifier than specified in the constraints file, a message window pops up and the user is asked whether he/she wants to continue. The template structure is *not* automatically loaded.

`# basis`: Currently the number is used as the total number of slow modes both in single-chain fits and multi-chain protein complex fits. This behavior may change once we have more experience. Probably, the argument will specify the number of slow modes *per chain* in the future. In any case, the number may be slightly larger than the number of constraints in exploratory work, but should not be much larger. It is good practice to use slightly *more* constraints than basis modes if you want to discuss the model in a publication.

# **DEER**: The arguments of this keyword supersede any spin labeling with a different label or at a different temperature that you may already have performed on the template structure (rotamers are automatically recomputed). If rotamer computation ([site scan](#)) (*Section 3.1*) was already performed for the specified sites (see below) with matching spin label and temperature, rotamer computation is skipped. It is not necessary to compute labels before, in version 2011 rotamer computation is fast. In any case you should not actually attach the labels before fitting.

*DEER constraint specification*: DEER constraints are specified by [MMM addresses](#) (*Section 1.6*) of the two residues involved, followed by the mean distance (in nm) and the full width of the distance range (in nm). The width of the distance range is only used for relative weighting of constraints, hence a rough estimate is sufficient. In single-chain structures, the residue number suffices as an address. In multi-chain protein complexes the chain identifier in parentheses must precede the residue number (without intervening space). The structure must not be specified in the residue address. It is good practice to comment each constraint (where does it come from, functional assay, etc.). Constraints from CW EPR are currently specified as DEER constraints.

# **direct**: This keyword allows to specify direct constraints between C $\alpha$  atoms of residues, for instance to fix part of the structure during the fit or to use information from an incomplete crystal structure.

*direct C $\alpha$  constraint specification*: Direct constraints are specified by [MMM addresses](#) (*Section 1.6*) of the two residues involved, followed by the distance (in nm) and the full width of a distance range (in nm). The width of the distance range specifies how strongly this distance should be restrained. This can be used to weigh direct constraints with respect to each other or with respect to experimental constraints. In single-chain structures, the residue number suffices as an address. In multi-chain protein complexes the chain identifier in parentheses must precede the residue number (without intervening space). The structure must not be specified in the residue address. It is good practice to comment each direct constraint (reason why it is provided).

# **ensemble**: A structural model makes sense only together with some idea about its uncertainty. In template-based fitting uncertainty derives from sparsity of distance constraints, errors in distance constraints, approximative character of coarse-grained [elastic network modeling](#), (*Section 3.7.1*) and with spin-label constraints, approximations involved in rotamer computations. Although the problem is too complex for mathematically strict error analysis, some estimate of uncertainty can be obtained by generating an ensemble that reflects sparsity of constraints and uncertainty in experimental distances. Note that uncertainty in experimental distances also includes uncertainties in rotamer simulation. An ensemble will not reflect, however, errors in the structure caused by a basis of slow modes that is too small to cover all relevant degrees of freedom of the protein backbone.

*Alternative ways of generating ensembles*: The ensemble can reflect only sparsity of restraints, only uncertainties in experimental distances, or both. In all cases the first, obligatory argument specifies the total number of structures in the ensemble. This is

also the number of individual fits that have to be performed. As a single fit may take several hours for large structures, we advise to select this parameter with care and to generate an ensemble only after a single structure considering all constraints and without random distance errors was computed. Unless a third argument `all` is provided, the ensemble is generated by systematic neglect of individual constraints. Single constraints are omitted in the order of increasing deviation between experimental distance and distance in the template structure. Pairs and triples of constraints are omitted in the order of increasing sum square deviation between experimental distances and distance in the template structure. No more than three constraints can be omitted in any individual structural model. At least two DEER constraints must remain, i.e. ensemble generation in this way is not possible for fitting with only direct constraints. If any of these conditions cannot be fulfilled, the program creates the largest ensemble possible with fulfilling all conditions (this number is then smaller than the requested ensemble size). Typically, one would have at least 10 DEER constraints and would use an ensemble size matching the number of constraints (each constraint is neglected in one structure, no structure has more than one neglected constraint). If a second argument to the `# ensemble` keyword is given, it is interpreted as a uniform uncertainty of all DEER distance constraints in units of nm. The uncertainty is taken as a standard deviation by adding random numbers with Gaussian distribution to all specified mean experimental distances. This is done in addition to neglecting individual constraints (or constraint pairs or triples), unless a third argument to the `#ensemble` keyword is given. This argument can only be `all`, any other argument in this place is ignored in the current version. If it is `all`, then all constraints are used in any structure in the ensemble. The fits are then just started from distance constraints with different added random errors.

In test mode, a file `pdbid_ensemble.log` is written in the current directory that lists the actually used constraints for each structure and diagnostic information on the fit. Here `pdbid` is the PDB identifier of the target structure, as specified in the constraint file.

`# target`: This is used for testing algorithms and for parametrization. For that a model of a target structure is created from a template input structure and a list of distance constraints for the target. In such tests, the target structure must be known and loaded into MMM. The name of the structure, usually the PDB identifier, must be provided as an argument. If this is done, MMM will track the C<sup>α</sup> r.m.s.d. between current model and the known target structure during fitting. If a single model is requested (and parallel computing was not initialized before), the r.m.s.d. changes are displayed in auxiliary Matlab windows. If an ensemble model is computed and the parallel computing toolbox is available and MMM preferences request more than a single CPU core, the auxiliary figure will not appear. This also happens for a single model, if parallel computing was used before in the same Matlab instance. In that case, fit diagnostics can be visualized after completion of the fit by using the `D` button on the top left of the `Restraint matching` plot.

`# TIF`: Active space extension was optimized on a set of 18 structure pairs of proteins with between 100 and 1000 residues and different types of large-scale motion. The maximum number of normal modes in active space is specified by a

threshold value for the normalized inverse frequency of the normal modes. The default value of 0.075, specified as `ENM_param.tif` in the subroutine `initialize_MMM.m` means that the maximum active space includes all modes whose inverse frequency is at least 0.075 time the inverse frequency of the slowest mode (or whose frequency is at most  $13 \frac{1}{3}$  as large as the one of the slowest mode). By specifying `# TIF 0.05` active space would be more extended, by specifying `# TIF 0.10` it would be less extended. Don't specify this parameter unless you have read and understood the paper on optimization of the algorithm.

`# MMAX`: Active space extension was optimized on a set of 18 structure pairs of proteins with between 100 and 1000 residues and different types of large-scale motion. By default the maximum active space extension is attained after 30 iteration steps, as specified in `ENM_param.mmax` in the subroutine `initialize_MMM.m`. By specifying `# MMAX 20` active space becomes faster, by specifying `# MMAX 40` it becomes slower. Don't specify this parameter unless you have read and understood the paper on optimization of the algorithm.

`# END`: Make it your practice to always conclude a constraint file with the `# END` keyword. The current version may work if you forget this, but we do not guarantee that this will also be the case in future versions of MMM.

## Example constraint file

```
% Experimental DEER restraints for ATP-EDTA form of MalFGK2 (Male and
maltose present)

# PDB 2R6G

# basis 10

# DEER MTSL 175

(F)92 (F)205 4.05 2.12 % Grote et al., 2009, JBC, 284, 17521-68

(F)205 (F)252 1.48 0.76 % Grote et al., 2009, JBC, 284, 17521-68

(F)205 (F)239 2.38 0.49 % Grote et al., 2009, JBC, 284, 17521-68

(F)239 (F)252 1.98 0.48 % Grote et al., 2009, JBC, 284, 17521-68

...

# direct

(F)406 (G)195 3.84 1.00 % coupling helices

(F)399 (G)190 2.33 1.00 % coupling helices

(F)421 (G)210 1.81 1.00 % "gate" helices

# END
```

## Specification of the elastic network model force constants

The elastic network model force constants depend on distance between C $^{\alpha}$  atoms. The form of this dependence and a cutoff distance are different for different forms of the elastic network model. The default model of MMM is ed-ENM, which was fitted to reproduce the protein flexibility observed in MD simulations (Orellana *et al.* 2010 *J. Chem. Theor. Comput.* 6:2910-2923). Other possible choices are discussed in G. Jeschke 2012 *Z. Phys. Chem.* 226:1395-1414 ([link](#)). The type of ENM can be selected via the menu item [File/Preferences](#) (Section 2.1.1) before opening the fit window.

## Limitations

Computation of an [ANM](#), (Section 3.7.3) which is a required step in this fitting procedure, can fail for more than about 2000 residues in the structure due to memory limitations with a typical desktop or laptop computer at the time of writing. Use a computer with larger memory available to Matlab to avoid this problem.

## C $^{\alpha}$ wireframe model

The left half of the ANM window contains the [State transition animation](#) panel for displaying a C $^{\alpha}$  wireframe model for normal mode visualization. On opening of the window, the wireframe is colored according to [colorscheme chain](#), (Section 2.2), i.e. different chains in a protein complex are distinguished by color. After a [constraint file](#) is loaded using the [Load constraints](#) button, the wireframe changes to neutral grey color and experimental distance constraints are visualized as color-coded connections between sites. Sites (currently only spin labels) are visualized as blue dots, connected by thin blue tether lines to the C $^{\alpha}$  node of the corresponding residue in the wireframe model. The color code is green for experimental distances that are within the experimental (error) range around the distance in the current model. Directly after loading constraints, before fitting, the current model is the template structure. The color code is yellow for distances outside the error range but within twice this range and red for distances outside twice the range. An overview of [Constraint matching](#) information is given in the auxiliary plot in the right half of the window. Both plots can be copied into separate Matlab figures for copying or printing by clicking on the respective [c](#) buttons.

The wireframe model display is linked to the full 3D model display of MMM. When you rotate the model in the 3D model window, the wireframe will rotate the same way (a copy in a separate Matlab figure window does not move along). The wireframe model is also used for animation of structural changes during the fit. This mode is available via the animation control panel after the fit was performed.

## Fitting

Fitting is initiated by clicking the fit button. This button is enabled only after a [constraintsitting is initiated by clicking the fit button. This button is enabled only after a constraints file](#) has been loaded. Fit progress is displayed in auxiliary Figure 1 (a



normal Matlab figure) in terms of constraint r.m.s.d. in Å. The plot is updated after each iteration. If an ensemble model is computed and the parallel computing toolbox is available and MMM preferences request more than a single CPU core, the auxiliary figure will not appear. This also happens for a single model, if parallel computing was used before in the same Matlab instance. In that case, fit diagnostics can be visualized after completion of the fit by using the **D** button on the top left of the *Restraint matching* plot.


Note also that the progress window appears only after initialization of the fit, which requires normal mode analysis of and [anisotropic network model](#) (Section 3.7.3) and can take minutes to an hour. Each iteration takes the same time, hence you really have to be patient for large structures.

Fitting normally stops when the constraint r.m.s.d. falls below 1.9 Å (if label-to-label constraints are present) or 0.2 Å (if only direct C<sup>α</sup>-C<sup>α</sup> distance constraints are present). For test and development purposes, fitting can be extended by activating the *overfit* checkbox at the bottom right of the window.

If the *overfit* checkbox is activated, fits will continue until the constraint r.m.s.d. is lower than 0.2 Å (also for label-label constraints) or constraint r.m.s.d. converges (it reduces by less than 1% of the largest reduction in any iteration step, after such saturation, 10 more iteration cycles are performed). Fitting also invariably stops if the maximum number of iteration cycles (100, specified in *ENM\_param.cycles* in subroutine *initialize\_MMM.m*) is reached or if the maximum computation time (12 hours for a single model) is exceeded. If you run into a maximum computation time problem, you can change variable *maxtime* in routine *fit\_by\_ANM\_transition.m* in the *\private* subdirectory of MMM (specified around line 153).

Note that overfitting has no effect on the model structure that is returned. This is always the structure that corresponds to the appropriate constraint r.m.s.d. threshold.

## Fit options

Except for the overfitting feature, which will not do any damage, only cost time, choice of fit options is restricted to prevent inexperienced users from falling into traps. Hence, the mode of mode recomputation during iterative fitting and the choice of fit algorithm can only be changed if the lock in the main window has been released (click the  tool button) *before* opening the *Fit from template/by Elastic network model* window.

A popup menu below the *Fit* button allows for changing the treatment of modes and application of local constraints (expert mode). The default is mode recomputation in each iteration step and not using local constraints for direct and next neighbor residues (*recompute/no local constraints*). Mode reorientation (*reorient*) is faster, but may compromise result quality. Unless you understand the issue, keep with the defaults.



If you wish to work with the original Zheng/Brooks algorithm rather than with thermal steps, you have to use the `recompute` mode (with local constraints) for the normal modes and you have to use the [ENM parametrization](#) by Jeschke that additionally rigidifies local geometry. MMM will make these selections for you, if you deactivate the checkbox `thermal`.

The `use force field` option requires that [SCWRL4](#) (Section 5.2) and [Tinker](#) (Section 5.2) are on the Matlab path. It has the advantage that in each step a full-atom model is created and locally optimized. However, this does not appear to improve fit quality significantly and it slows down fitting very strongly. Use of this option is currently discouraged. The option is currently incompatible with parallel ensemble computing. If you want to use it, change the number of requested CPU cores in [File/preferences](#) (Section 2.1.1) to 1.

## Parallel computing support

For ensemble computations, parallel computing is offered if the Parallel Computing toolbox of Matlab is available. You can restrict the number of processor cores used by MMM in the [Preferences](#) (Section 2.1.1) window, accessible via the `File` menu, to allow for convenient working with your computer while MMM is performing time-consuming tasks. If you do not have the Parallel Computing toolbox or only a single processor core, the ensemble is computed sequentially.

## Param.

The `Param.` button can be used to parametrize active mode extension on tests sets of structure pairs. This is a developer feature.

## Saving the coarse-grained fit result

After fitting, the final coarse-grained model ( $C^{\alpha}$ ) coordinates can be saved by clicking on the `Save` button. This feature is used only by programmers and may be removed or replaced in future versions. Fit results and displacement vectors for each fit step are automatically stored. The fit result is assigned a structure tag `trx`, where  $x$  is the next free structure number ( $x=2$ , if only the template structure was loaded and this is the first fit). When the whole model is saved from the [file menu](#) (Section 2.1.1) of the main window, the fit result is saved with the model. To generate a PDB file of the fit model, use [Save as PDB](#) (Section 2.1.1) in the file menu of the main window, after making `trx` (usually `tr2`) the current structure in the [hierarchy window](#). (Section 2.3)

## Animation

The animation shows how the fit proceeded from the template structure to the final structure. For a state transition of the protein, this should be a reasonable approximation of the transition pathway. Best results are obtained when the checkboxes `restraints` and `initial & final` are unchecked. Default behavior is

that wireframe is animated for 5 transitions forth and back between the template and final structure, seen from the same viewing perspective as the static model in the 3D model window. First the visualization is computed frame by frame, then it is played back by default for five full oscillations. The number of such cycles (oscillations) can be changed.

The oscillation can be superimposed with rotation of the model about the x axis (along viewing direction, z axis (vertical axis in screen plane)). A full rotation (360°) is superimposed with the number of oscillation cycles requested. Note that computation of animations with superimposed rotation takes much longer, as with rotation frames for all oscillation cycles must be computed, while without rotation only frames for a single oscillation are computed.

You can choose between grey and black background with the `black bckg.` checkbox. Interactive animation is started by clicking on the `Animate` button. Note that the animation is not stored. It is recomputed when you click this button repeatedly, even if parameters are unchanged.

## Movie of an animation in AVI format

You can produce and store a movie of an animation in AVI format by clicking the `Save AVI` button. The animation should be set up before (and preferably tested on screen) as [described above](#).

Quality of Matlab-generated AVI movies is rather low. The next version of MMM will allow for saving a trajectory as a PDB file with several models, for visualization in UCSF Chimera, or concatenated PDB files, for visualization in VMD. Both programs can create high quality movies.

## Closing the window

By clicking the `Close` button below the wireframe model plot the window is closed. On reopening it will not appear in the same state and animation of a previous fit will no longer be possible (in the current version). Instead, the window reopens with the template structure displayed for a new fit.

# 3.8 Fit from template/Modeller

## Introduction

This feature requires that [Modeller](#) is installed in an MMM subdirectory, or a writable directory known in the Matlab path, and that the name of the executable, for instance `mod9.10`, is properly specified in the preferences window, which is accessible from the [File menu](#). (*Section 5.2*) . (*Section 2.1.1*)

This fit procedure can be used to create a homology model based on a template structure of a related protein, a sequence alignment, and a set of distance

constraints and, optionally, secondary structure constraints. In simple cases, where sequence identity is moderate to high, sequence alignment can be automated by using the alignment program [MUSCLE](#), (Section 5.2) which is bundled with MMM. If the template and target sequence are the same (a model of unknown state B structure of a protein is to be computed from the known structure of state A and constraints for state B), no alignment file is required.

We emphasize that the concept of a single "true" structure of a protein or protein complex is largely fiction. Nevertheless *some parts* of a protein are fixed at atomic resolution. The models obtained with the Modeller fit procedure should not be discussed in terms of such atomistic detail, unless sequence identity is very high at least near the details to be discussed. Otherwise, only speculations are possible on this level.

The current MMM version allows for homology modelling of *only single-chain proteins*. It is still in a development stage. In particular, modelling supported by experimental constraints may still change. The help file does not describe all features. We suggest that you wait for our publication on the approach before using the feature for serious work or that you contact us.

The template structure must be loaded and must be the current structure in MMM before the window is opened.

## Specification of sequence alignment

Skip this step if you model a structural transition between states of the same protein.

Homology modeling is based on an alignment between the sequence of the template protein and the sequence of the protein to be modeled (target protein). MMM can read alignment files in FASTA format (default extensions `.afa`, `.fa`, `.fasta`, `.ebi`, `.txt`), Clustal format (default extensions `.clw`, `.aln`, `.clustalw`), and in Modeller's native PIR format (default extensions `.ali`, `.pir`). The native format of MMM is FASTA and this is assumed for any file extension that is not assigned to another format. Note, however, that the file dialog window by default lists only files in the native Modeller format PIT (`*.pir`, `*.ali`). With the `Files of type:` popup menu in this window, you can select that FASTA files, Clustal files, or any file type is listed.

The alignment is loaded via the `Alignment` button. MMM checks whether the template sequence exists in the alignment file and, if not, adds this sequence and aligns it to the other sequences by using [MUSCLE](#). (Section 5.2) This feature also allows working from only the template structure and a FASTA file of the target sequence, without explicitly specifying the alignment. However, we discourage this automatic alignment mode in cases where sequence identity is lower than 50%.

Sequence identity is displayed in the **Message board** of the main MMM window after the alignment or target sequence was loaded. Furthermore, the `Alignment` button is color-coded after loading an alignment. Red color corresponds to 20% sequence identity or less ("midnight zone", unreliable without additional constraints),

green color to 40% sequence identity or more (rather reliable homology models). In between ("twilight zone"), color is linearly interpolated on an RGB scale, with yellow corresponding to 30% sequence identity. This is really a traffic light code. There are cases where driving over a red traffic light will work out for you, but be aware that it is dangerous.

## Specification of constraints and fit parameters

Fits are controlled by an ASCII text input file that is opened via the [Constraints](#) button. After the constraints are loaded, they are visualized in the [wireframe model](#) using a color code that indicates whether they are matched (green, within the standard deviation of the distance specified in the constraint file), poorly matched (yellow, within twice the standard deviation), or not matched (red, deviating by more than twice the standard deviation). The plot [Restraint matching](#) (top right) visualizes the same information ordered by constraint number and a [report editor](#) ([Section 2.7](#)) opens that provides this information in text format, so that you can save it.

Comment lines or comments in command lines are introduced by the percent character %. Keyword lines start with the hash character #. There must be a space between the hash character and the keyword. Two types of keywords exist, single-line keywords (all parameters are specified on the same line as the keyword) and block keywords (some parameters follow in additional lines, this is used for constraint specification). Blocks end when the next keyword is encountered. The last block must be followed by an `# END` keyword line. Keywords are not case sensitive. Keywords and arguments are separated by spaces.

Currently, the following keywords are defined:

- `# PDB` (single-line), argument: PDB identifier of template structure, the chain can be given after a colon (:)
- `# DEER` (block), arguments: spin label and temperature (in K) for rotamer simulation, starts block for EPR (DEER and CW EPR) constraints
- `# DIRECT` (block), no arguments, starts specification block for C<sup>α</sup>-C<sup>α</sup> constraints.
- `# HELICES` (block), no arguments, starts specification of secondary structure constraints for (α-)helices.
- `# STRANDS` (block), no arguments, starts specification of secondary structure constraints for β-strands.
- `# SHEETS` (block), no arguments, starts specification of secondary structure constraints for the alignment of β-strands to sheets.
- `# ALIGN` (single-line), argument: file name of alignment file, we recommend to use the [Alignment](#) button instead
- `# CHAINS` (single-line), arguments: identifiers of the chains, this allows to name the chain in the target structure other than `A`
- `# REALIGN` (single-line), argument: threshold for realigning the sequences, not yet implemented
- `# END` (single-line), no arguments, stops reading of the file

## Remarks

Future versions may have additional keywords. If you are a programmer and want to implement keywords of your own, please contact us before. The constraint file is processed by function `rd_restraints.m`. Note also that some other keywords are already in use for [elastic network model fitting](#). (Section 3.7.4)

# **PDB**: This is used for a check whether the current structure in MMM, which can be picked in the [hierarchy window](#), (Section 2.3) is the requested template structure. If the current structure in MMM has a different PDB identifier than specified in the constraints file, the user can either use the current structure or load the specified structure (if it is available on the FTP server of the PDB).

# **DEER**: The arguments of this keyword supersede any spin labeling with a different label or at a different temperature that you may already have performed on the template structure (rotamers are automatically recomputed). If rotamer computation ([site scan](#)) (Section 3.1) was already performed for the specified sites (see below) with matching spin label and temperature, rotamer computation is skipped. It is not necessary to compute labels before, since version 2011 rotamer computation is fast. In any case you should not actually attach the labels before fitting.

*DEER constraint specification:* DEER constraints are specified by [MMM addresses](#) (Section 1.6) of the two residues involved, followed by the mean distance (in nm) and the full width of the distance range (in nm). The width of the distance range is not currently used in homology modeling. In single-chain structures, the residue number suffices as an address. In multi-chain protein complexes the chain identifier in parentheses must precede the residue number (without intervening space). The structure must not be specified in the residue address.

It is good practice to comment each constraint (where does it come from, functional assay, etc.). Constraints from CW EPR are currently specified as DEER constraints.

# **direct**: This keyword allows to specify direct constraints between  $C^\alpha$  atoms of residues, for instance to fix part of the structure during the fit or to use information from an incomplete crystal structure.

*direct  $C^\alpha$  constraint specification:* Direct constraints are specified by [MMM addresses](#) (Section 1.6) of the two residues involved, followed by the distance (in nm) and the full width of a distance range (in nm). In single-chain structures, the residue number suffices as an address. In multi-chain protein complexes the chain identifier in parentheses must precede the residue number (without intervening space). The structure must not be specified in the residue address. It is good practice to comment each direct constraint (reason why it is provided).

# **ensemble**: Such a keyword exists, but is not used for homology modeling. Please specify the number of computed models and selected ensemble in the **Modeller control** panel of the fit window..

# END: Make it your practice to always conclude a constraint file with the # END keyword. The current version may work if you forget this, but we give no guarantee that this will also be the case in future versions of MMM.

### Example constraint file:

```
% Experimental DEER restraints for PutP

# PDB 2XQ2:A

# CHAINS A

# DEER MTSL 175

41 62 3.23 0.49

41 446 4.74 0.64

...

# HELICES

8-24

44-55

...

# END
```

## Creating and importing models

The `Run options` subpanel of the `Modeller control` panel allows for specifying how Modeller is called. For very long jobs (large protein or large ensemble size) you may wish to run Modeller as a background job by activating the `background run` checkbox. Note, however, that you will lose some information, the visualization of constraint matching in the wireframe model and `Constraint matching` plot are not available. The advantage of a background run is that you can do other work with MMM while the Modeller job runs and that you can even close MMM and Matlab (and thus free a Matlab network license for other users). Once you restart MMM it will check for running and completed jobs and allow for loading results of completed jobs. You can also get an overview of background jobs and interrogate the current state of the computation via the `Jobs` menu.

In a background run, a console window (DOS window) may open that will state `'import site' failed; use -v for traceback`. This warning message is a minor bug of Modeller, which does not affect your results. Do not close that window (you can minimize it)! Job control of a background job also reports the runtime of a background job and allows for displaying the complete (Modeller) logfile and for killing (deleting) a background job.



The **Models** edit field allows for specifying the size of the ensemble to be computed by Modeller. This may be larger than the ensemble of models you wish to import later (see below).

You can specify structure identifiers, which are not necessarily PDB identifiers, for the template and target structures. These identifiers are used for Modeller output and input files that are stored in the `\tmp` subdirectory of MMM, and the target identifier also specifies the name of the structure after importing results.

The **repacking** checkbox requests an alternative spin label rotamer computation that we implemented for test purposes. It will strongly slow down rotamer computation and does not have any significant advantage. You may want to use it occasionally to test how sensitive your results are with respect to specifics of spin label rotamer modeling. It requires that [SCWRL4](#) (Section 5.2) is installed as third-party software.

MMM will always automatically remove spin labels after importing models and will repack sidechains after this, if [SCWRL4](#) (Section 5.2) is installed as third-party software. We strongly recommend doing this, SCWRL4 is free for academic use.

Information in the **Import models** subpanel should ideally also be specified before running Modeller, in particular the **Target ID**. The edit field **Min. GA341** specifies a threshold, below which models should be rejected. The authors of Modeller suggest that models with GA341 score below 0.70 are unreliable. The default value of MMM is a bit more conservative (0.75), but you can change it. It can be decreased down to 0.50 with the **-** button and down to 0.30 by direct input. Using such low scores is supported for development purposes and for early stages of modeling, where the alignment might still be poor. We strongly advise against publishing models with GA341 below 0.70. If you really must do so, at least take pains to explain to the readers of your paper that this is a rough, uncertain model.

The **Ensemble** edit field allows you to specify how large the imported ensemble should be. This number cannot be larger than the number of computed models, hence, the two edit fields are coupled. It can be smaller if you wish to select potentially more reliable models from a large computation.

In any case, the imported ensemble will exclude all models below the GA341 threshold and may thus be smaller than requested, in particular, if sequence identity of the alignment was poor. If there exist more computed models above the GA341 threshold than are requested in the imported ensemble, the ensemble is created from the models with the highest DOPE score (an independent energy function, which is not the Modeller objective energy function). Such selection follows guidelines issued by the authors of the Modeller package.

The Modeller computation is started with the **Run** button. If you run it in the foreground, the message board will report Modeller log output after the computation is finished. This report will end with a warning, which results from a minor Modeller bug that does not affect your results. The cursor will also change from watch to arrow when a foreground Modeller run is completed. At that point you need to import results with the **Import** button.



After import, the constraint matching of the first loaded homology model is visualized in the [wireframe model](#) using a color code that indicates whether they are matched (green, within the standard deviation of the distance specified in the constraint file), poorly matched (yellow, within twice the standard deviation), or not matched (red, deviating by more than twice the standard deviation). The plot [Restraint matching](#) (top right) visualizes the same information ordered by constraint number and a [report editor](#) ([Section 2.7](#)) opens that provides this information in text format, so that you can save it.

For a background job, a window [Modeller job completed](#). will pop up once the job is finished. You can import results immediately or after closing and reopening MMM. The latter choice is advisable if you are currently working on another project and you do not want to add the imported ensemble to the model in MMM. Note that the imported structure is not automatically displayed (could take very long for a large ensemble) and that you do not receive reports on constraint matching for a background job.

If you had closed MMM or even Matlab after starting a background job, MMM will check for completion when it is opened the next time. If the job is already completed, you will be offered to import the results. If the job has not yet been completed, MMM will periodically check for completion as long as it is running.

If your computer crashed while you were running a background job or if a background job crashed on its own account, you need to delete it from the list of running background jobs using the [Jobs/Job control](#) menu item.

## **C<sup>α</sup> wireframe model**

The left half of the ANM window contains the [State transition animation](#) panel for displaying a C<sup>α</sup> wireframe model for normal mode visualization. On opening of the window, the wireframe is colored according to [colorscheme chain](#), ([Section 2.2](#)) , i.e. different chains in a protein complex are distinguished by color. After a [constraint file](#) is loaded using the [Constraints](#) button, the wireframe changes to neutral grey color and experimental distance constraints are visualized as color-coded connections between sites. Sites (currently only spin labels) are visualized as blue dots, connected by thin blue tether lines to the C<sup>α</sup> node of the corresponding residue in the wireframe model. The color code is green for experimental distances that are within the experimental (error) range around the distance in the current model. Directly after loading constraints, before fitting, the current model is the template structure. The color code is yellow for distances outside the error range but within twice this range and red for distances outside twice the range. An overview of [Constraint matching](#) information is given in the auxiliary plot in the right half of the window. Both plots can be copied into separate Matlab figures for copying or printing by clicking on the respective [c](#) buttons.

## Closing the window

By clicking the **Close** button below the wireframe model plot the window is closed. On reopening it will not appear in the same state and animation of a previous fit will no longer be possible (in the current version). Instead, the window reopens with the template structure displayed for a new fit.

## 3.9 Scripting

### What is scripting?

Some tasks in MMM are best performed using the basic [command language](#). (Section 2.2) . Although commands can be input one by one *via* the [command line](#), (Section 2.1.14) it may sometimes be more convenient and reliable to write commented command scripts.

In particular, definition of secondary structure elements and of synonyms for the structure name and for particular chains is most conveniently performed directly after reading a PDB file using an [initialization script](#).

### Syntax

The same [command syntax](#) (Section 2.2) applies as for the [command line](#). (Section 2.1.14) In addition, comments can be added after a command or as separate lines. A comment is introduced by the `%` character. In pure comment lines the `%` character must be the first character, except for spaces.

Nonexisting commands or commands with wrong arguments are simply ignored and the remaining script is further executed. The corresponding error messages are shown in the [message board](#). (Section 2.1.17)

## Initialization scripts

### Introduction

Initialization scripts are useful when a structure is read from a PDB file. For instance, secondary structure information in PDB files is often nonsensical and therefore is not used by most protein visualization programs. MMM uses the assignments of secondary structure types in a PDB file (but not the lists of helices and sheets, which are often not self-consistent), but it is still better to redefine secondary structure elements, using the nomenclature and residue ranges given in an accompanying paper.

Furthermore, the common names of a protein complex or of its constituents (peptide chains) cannot be reliably extracted from PDB files. These names can be defined as synonyms. Otherwise, the structure can be addressed only by the PDB identifier and the chains by uppercase characters.

Finally, default display behavior (ribbon plot for peptide chains, no display of cofactors) may not be intended. If an initialization script is performed, MMM suppresses default display. If the script does not contain show commands, nothing is displayed, although the structure is still loaded.

### How to write an initialization script?

An initialization script may contain any [command](#) (Section 2.2) that could be executed from the [command line](#), (Section 2.1.14) except for the `undo` and `redo` commands. It must have the same file name as the PDB file (except for the extension), must have the extension `.mmm` and must be stored in the same directory as the PDB file.

### How is the initialization script executed?

After reading a PDB file, MMM checks if a corresponding initialization script exists. If so, the user is asked whether it should be run. If the user answers 'No', secondary structure definitions from the PDB file are locked and default graphics is displayed (ribbon plot without cofactors). If the user answers 'Yes', secondary structure definitions are unlocked and the script is executed. Default graphics display is skipped. The script defines which graphics, if any, is to be displayed.

### Why to write an initialization script?

If you want to redefine secondary structure, do it in an initialization script. Every other way is bound to create a mess. And you want to redefine it, as the secondary structure information in a PDB file often is a mess, too.

### An example script

This is an initialization script for PDB file 1ZCD.

```
synonym [*] NhaA % assigns the synonym 'NhaA' to the current structure

undefine % undefines any existing secondary structure elements (defined
during PDB read)

% the following lines define transmembrane helices and the strands
mentioned in the accompanying paper

loop [NhaA] 11 9 11

helix [NhaA] I 12 30

loop [NhaA] 12 31 43

sheet [NhaA] a 44 50

loop [NhaA] 13 51 52
```

sheet [NhaA] b 53 58

helix [NhaA] II 59 85

loop [NhaA] 14 86 94

helix [NhaA] II 95 116

loop [NhaA] 15 117 120

helix [NhaA] IVp 121 131

loop [NhaA] lIV 132 133

helix [NhaA] IVc 134 143

loop [NhaA] 16 144 149

helix [NhaA] V 150 175

loop [NhaA] 17 176 181

helix [NhaA] VI 182 200

loop [NhaA] 18 201 204

helix [NhaA] VII 205 218

loop [NhaA] 19 219 222

helix [NhaA] VIII 223 236

loop [NhaA] 110 237 246

helix [NhaA] IX 247 271

loop [NhaA] 111 272 289

helix [NhaA] X 290 311

loop [NhaA] 112 312 326

helix [NhaA] XIc 327 336

loop [NhaA] lIX 337 339

helix [NhaA] XIp 340 350

loop [NhaA] 113 351 356

helix [NhaA] XII 357 382

loop [NhaA] 114 383 384

```
show [NhaA] ribbon % display a ribbon plot

% and show the residues in the strands also as stick plots

show [NhaA] (A)<E.a> stick

show [NhaA] (A)<E.b> stick

detach % detaches the model panel into a separate model window
```

## Limitations

The [undo](#) (Section 2.2) and [redo](#) (Section 2.2) commands cannot be executed in scripts.

In scripts, commands that cannot be undone are performed without previous query.

## 3.10 Superimposing structures

### Superimposing structures or parts thereof

This function allows to superimpose two structures or parts of two structures by minimization of the root mean square deviation of atom positions (A. D. McLachlan, *J. Mol. Biol.* **1979**, 128, 49-79). One of the structures serves as the template and is unchanged, the other structure is the moving structure and is rotated and translated to achieve the best fit. By selecting always the same template structure with different moving structures you can superimpose multiple structures. See also explanation of the corresponding [menu item](#). (Section 2.1.2)

Note that one of the structures has to be redrawn, and as MMM is a bit slow with graphics, this may take some time. Fitting starts only when you click the **OK** button. To find out only the r.m.s.d. without actually transforming a structure, [save the transformation matrix](#).

Superposition of complete structures works only if the chain identifiers match. Otherwise, select corresponding residues, use the **only selected** option, and deactivate the **check residue correspondence** checkbox.

### Superimposing complete structures

#### Selection of template and moving structures

Two popup menus allow for selection of the **Template structure** and **Moving structure**. Note that atoms are superimposed by matching chain identifier, matching residue number, and matching atom tag.

## Atoms to be fitted

The `Fit mode` group of radiobuttons allows to select which atom types are to be fitted. For identical sequences you might want to superimpose `all atoms`, but most of the time you may want to superimpose only `backbone atoms`. In rare cases you may want to restrict superposition to the  $C^\alpha$  atoms.

## Superimposing parts of structures

To superimpose only parts of a structure, select the corresponding residues both in the intended template and intended moving structure. If different sets of residues are selected, only matching residues are considered (chain identifier, residue number, and insertion code are checked). You can also select cofactors, if they have the same chain identifiers and residue numbers in both structures.

If chain identifiers or numbers of residues to be superimposed do not match, superposition can still be done after careful residue selection. In both structures to be superimposed you have to select corresponding residues in exactly the same sequence. The number of selected residues should be the same in both structures (surplus residues in the moving structure are ignored, surplus residues in the target structure create an error). Only residues, not atoms, or whole chains can be selected for this mode. You need to deactivate the `check residue correspondence` checkbox before performing the transformation.

In the `Fit mode` panel, activate the `only selected` checkbox.

## Sequence alignment based superposition

For unlike proteins it may be tedious to manually generate selections of corresponding residues. MMM can try to align the sequences of the two proteins and superimpose either the aligned residues or the whole structure. As an example you can load PDB structures 1OMP and 2HQ0 and superimpose them by checking the `aligned residues` checkbox.

## Display motion arrows

If the two structures are of the same protein in different states, you may display 3D (conical) arrows that depict the structural transformation. For that it is best to display only the moving structure as a coil. See also the [motion command](#) (*Section 2.2*) for how to do that in a single step.

## Ensemble mode

The ensemble mode is useful if the moving structure is an ensemble of different conformations, whereas the template structure is a single model. In that case it is usually appropriate to separately fit each individual conformation in the ensemble to the template. This is achieved by activating the checkbox. If the checkbox is

deactivated, the selected conformation (if any) or the first conformation is fitted to the template and all other conformations are translated and rotated in the same way.

## Saving a transformation matrix

Instead of actually moving the structure, you can just save the 4×4 affine transformation matrix that achieves the superposition. Click on the **Save** button and select the filename (a selfexplanatory name is suggested). Then close the window by clicking the **Cancel** button so that actual transformation and redrawing are avoided. The r.m.s.d. of both structures is still shown in the Message board.

## 3.11 Binding sites

### What are sites in MMM?

In MMM, the term "sites" is used in two ways, for *binding sites*, as in this page, and for *labeling sites* (see [spin labeling site scans](#)).

Information on binding sites may be [imported from a PDB file](#) (Section 2.1.1) or may be [supplied by the user](#).

In general, binding sites are defined as a set of residues involved in binding of a cofactor. In contrast, metal coordination is defined by a set of coordinating atoms. If such information exists, it can be found via the **metal coordination** keyword with the [keyword finder](#). (Section 1.11)

The sites window can be accessed via the [Analysis/Sites](#) (Section 2.1.5) menu item.

### Selection of residues in a site

By default objects are added to the already existing selection, if any. If you want to start with a clean slate, first click the **Clear** button.

If you click on an item in the **Sites** list, the **Residues and cofactors** list is filled with all objects that have an annotation with this keyword. Only a single keyword can be picked at once. You can then pick one or several objects from the **Residues and cofactors** list (hold down the **Shift** or **Ctrl** keys during clicking for picking multiple objects). By clicking the **Select** button, you add all picked objects to the selection. Alternatively, you can use the **Select all** button to add all objects from the **Residues and cofactors** list to the selection.

### Displaying annotation

To display the [annotation](#) (Section 1.9) of a picked residue or cofactor, click the **Show annotation** button. If several objects are picked in the **Residues and cofactors** list, the annotation will be shown for the first picked item. Note that this and only this object will be selected after the annotation window is opened. If a selection of objects






exists when you click the button, a question dialog opens and you are given the opportunity to cancel your request and keep the old selection.

The annotation window opens at the automatically created page for binding sites.

Note also that you can return to the sites window only after you have closed the [annotation window](#). (Section 1.9)

## Visualizing a binding site

This procedure will cancel all existing [selections](#). (Section 1.8) . After opening the sites window via the [Analysis/Sites](#) (Section 2.1.5) menu item, select the binding site that you wish to visualize from the `Sites` listbox. Now first click the `Clear` button, then click the `Select all` button.

Via the  [Graphics mode toolbar](#) (Section 2.1.11) select the desired graphics mode; "stick" will usually work best. Detach the 3D model with the  [Detach toolbar](#) (Section 2.1.11) if you did not do this before. Now click the  [View selected toolbar](#) (Section 2.1.11) to zoom in on the binding site. [Adjust the view](#) (Section 2.1.16) if necessary. Now click on the `Depth cueing` button in the main window and [adjust depth cueing](#) (Section 2.1.13) if necessary.

## Defining a new binding site

To define a new binding site, [select](#) (Section 1.8) the residues and cofactors involved in the binding before you access the sites window. After opening the sites window (use the [Analysis/Sites](#) (Section 2.1.5) menu item), the preliminary definition of the new site is found as the last item of the `Sites` list. Pick this site by clicking on it and check that the `Residues and cofactors` listed in the right listbox are indeed the ones that belong to the site (if not, close the window by `Cancel` and correct the selection). Edit the site `Description` in the corresponding edit field and select an `Evidence type` from the popup menu. Now click the `New` button.

The site is now defined, but not yet stored and the `New` button is disabled. The residues involved in the site have been automatically annotated. To complement these annotations with additional information from your experiments (or from literature), pick these residues one by one in the `Residues and cofactors` listbox, then click the `Show annotation` button, go to the page with `*Binding sites` annotations and make your comments.

The new site will only be stored, once you close the window with the `OK` button. If you do not want to store it, use the `Cancel` button.

## Editing a site description

A binding site is identified by a three-character tag (starting with "AC1") and a short free text description. Tags are assigned automatically and can not be edited. Editing the descriptions of those sites that were imported from a PDB file is discouraged, but possible. If you want to add additional information, better use the residue [annotations](#) (Section 1.9) accessible via the `Show annotations` window.

If you wish to edit the description of a user-defined site, pick this site in the `Sites` listbox and correct the text in the `Description` edit line. The changed description will be displayed in the list immediately, but will be stored in the model only once you leave the sites window with the `OK` button. You can cancel changes by closing the window with the `Cancel` button.

## Selecting evidence type

For binding sites imported from a PDB file, the evidence type is `atomistic structure` and should not be changed, although this is possible. For [new binding sites](#) defined by the user, the default mode `conjecture` should be changed to something more meaningful, such as `mutation/assay`, `other biochemistry`, `homology`, or `spectroscopy`. Further explanations can be given in residue [annotations](#). (Section 1.9)

## Closing the window

Click the `OK` button. All existing selections will be kept. A [newly defined site](#) will be stored and edited site descriptions will be updated. If the `Description` or `Evidence type` of a new site were edited but the `New` button was not clicked, the new site is not fully defined. In this case there is a query whether you want to close the window without defining the new site.

Alternatively, click the `Cancel` button. All existing selections will be kept. A [newly defined site](#) will not be stored and edited site descriptions will not be updated. There is a query whether you really want to discard edits.

# 3.12 Assign transmembrane helices & strands

## Why should you assign helices and strands as "transmembrane"?

Accessibility computations for membrane proteins require a lipid bilayer model. Such a model is also useful in visualizations as the function of a membrane protein can only be understood in relation to the membrane. The most simple lipid bilayer model has three parameters: the direction of the membrane normal (unit vector in  $z$  direction), the  $z$  coordinate of the bilayer center, and the bilayer thickness. For homooligomers, the membrane normal is along the [symmetry axis](#). (Section 2.1.2) In other cases, the membrane normal can be approximated from the average direction of transmembrane helices or strands.

In ribbon plots, declared transmembrane helices and strands are shown with slightly stronger colors. This may also help to get a better overview of the structure of a membrane protein. If you do not like these color differences, you can apply the [colorscheme secondary](#) (Section 2.2) command. This results in the same color for all helices and strands.

## Alternatives

Transmembrane helices and strands can also be assigned directly with the [helix](#) (Section 2.2) and [sheet](#) (Section 2.2) commands. In fact, the assignment is best done in an [initialization script](#). (Section 3.9) Even if you do that, the assignment window is required for determination of the membrane normal.

For building a bilayer you just need to orient the structure so that the z axis is the approximate membrane normal. This can also be done by visual inspection. You rotate the structure so that the  $\beta$ -barrel or  $\alpha$ -helix bundle appears to have the correct orientation. Either you try to look along the membrane normal, then your viewing direction should become the z axis of the structure. Or you look sideways on the protein (perpendicular to the membrane normal), then the viewing directions should become the x axis. Transform the structure to the membrane frame by using the [Build/Transform to viewing frame/x along viewing vector](#) (Section 2.1.3) menu item or the [Build/Transform to viewing frame/z along viewing vector](#). (Section 2.1.3)

In complicated cases (large bundles with some helices sticking out of the membrane), assign TM helices with the helix command in a script, use the [Assign TM helices and strands](#) (Section 2.1.2) menu item and the **OK** button to find the approximate normal, set the view along x, correct the orientation of the protein as good as you can (if you can) by rotating the structure, select the [Build/Transform to viewing frame/x along viewing vector](#) (Section 2.1.3) menu item and finally fit the membrane normal in the [bilayer builder](#). (Section 3.13)

## Picking and unpicking transmembrane helices & strands

The TM assignment window always relates to the current structure. The left listbox shows all helices and strands in all chains of this structure, using MMM's [secondary structure address format](#). (Section 1.6.1) After a semi-colon (address delimiter), the range of residues is shown for easier comparison with the original publication on the structure.

If TM helices or strands are already declared then those are listed in the right listbox. Otherwise the right listbox is empty.

Helices or strands can be picked by clicking in the listbox (use the **Ctrl** or **Shift** keys for multiple selection) and assigned as "transmembrane" by the **>** button. Double assignments of the same element are ignored. Assignments can be deleted by picking the helix or strand in the right listbox and using the **<** button. All TM helices and strands can be unassigned at once with the **<<** button.

## Displaying the membrane normal

The membrane normal is computed and displayed with the `Show membrane normal` button. For this, direction vectors are computed for all selected TM helices and strands (thin red lines). These direction vectors are defined by minimum mean square deviation of all C $\alpha$  atoms from the line. The (approximate) membrane normal is the mean of all vectors, it is shown as a thick red line. Depending on the viewing angle, it may be easy or difficult to spot.

If necessary, the vectors and normals can be hidden with the `Hide membrane normal` button. Otherwise they disappear automatically when the window is closed.

## Transforming into the bilayer frame

For [building the bilayer](#) (Section 3.13) the structure has to be transformed to the bilayer frame. In this frame, the bilayer normal is the z axis and the origin is at the center of the mean direction vector of the  $\alpha$ -helical bundle or  $\beta$ -barrel. The origin thus approximately coincides with the center of the bilayer and the center of the bundle or barrel.

Transformation and redisplay are effected by the `OK` button. This also closes the window and hides the direction vectors and membrane normal.

## Staying in the original frame

The assignment window can also be used for just assigning TM helices and strands for visualization with a different color without transformation of the coordinate frame. For this, use the `Cancel` button. The window is closed, all current assignments from the right listbox are kept, and the direction vectors and membrane normal are hidden.

If you choose the `Cancel` button, subsequent building of a bilayer will not work, unless you have previously set the proper z axis via [symmetry](#). (Section 2.1.2) . You can always return to this window via the [Assign TM helices and strands](#) (Section 2.1.2) menu item and click `OK` to perform the transformation.

## 3.13 Build lipid bilayer

### What is a lipid bilayer in MMM and what it is needed for?

A lipid bilayer in MMM is a two-dimensional object associated with a structure. A model can thus contain several bilayers, one for each structure. The position of the bilayer with respect to the protein is defined by a *membrane normal*, which is associated with the z axis of the coordinate frame, and by the central plane position on the normal ( $z=0$ ).

The bilayer is homogeneous along its lateral extension, i.e. parallel to the xy plane. Along the membrane normal, a water profile, and oxygen profile, and- for phospholipids- a  $^{31}\text{P}$  density profile are defined. Together with a solvent accessible surface, these profiles allow for computation of water and oxygen accessibility of residues or spin labels and for the prediction of  $^{31}\text{P}$  Mims ENDOR spectra of spin labels.

Note that for structures with bilayers the coordinate frame cannot be subsequently transformed without destroying the bilayer.

The distribution of lipid functional groups and water are modelled according to N. Kucerka *et al.*, Biophys. J. **2008**, 95, 2356-2367. The oxygen profile was parametrized from data given in R.D. Nielsen *et al.* J. Am. Chem. Soc. **2005**, 127, 6430-6442.

## Before you can build a lipid bilayer

Bilayer building requires that [MSMS by Michel Sanner](#) (Section 5.2) is installed and known on the Matlab path.

The parameters of the bilayer, direction of the membrane normal, position of the bilayer centre, and thickness need to be adjusted to the structure under consideration. This can be done by visual inspection or by fitting. Fitting is based on minimization of the free energy of membrane insertion, which in turn is computed from *lipid propensities* of the residues that are in contact with the bilayer.

Fitting requires reasonable starting values for the membrane normal direction and centre coordinate. The whole range of reasonable values for the bilayer thickness is automatically scanned.

If the structure is an homoligomer, the membrane normal can be exactly predicted, as it coincides with the symmetry axis. In that case, use the [Edit/Symmetry frame](#) (Section 2.1.2) menu item to transform the structure to a frame where the symmetry axis is the z axis. The centre of the bilayer is initialized as the geometrical center of the structure. This principle also works for heteroligomers that have at least two copies of the same polypeptide chain in symmetry-related positions. If the system has pseudo-symmetry, as for instance the bacterial photosynthetic reaction centre, this pseudo-symmetry can be used to obtain a good approximation of the membrane normal.

If the membrane normal cannot be inferred from symmetry, it can be guessed from the mean direction of transmembrane helices ( $\alpha$ -helical bundle) or strands ( $\beta$ -barrel). This technique works better for  $\beta$ -barrels than for  $\alpha$ -helical bundles. In either case it provides a reasonable first approximation that can be improved by fitting or visual inspection. Use the [Edit/Assign TM helices and strands](#) (Section 2.1.2) menu item to define the membrane normal this way. The centre of the bilayer is initialized as the geometrical centre of the bundle or barrel.

In some cases a look along the x direction will tell you that the approximate membrane normal is just that- approximate. You can try to improve it by rotating the structure. This works best by rotating just the camera-up vector (press the middle mouse button or the left mouse button plus the **shift** key and drag the mouse). After you are satisfied, select the [Build/Transform to viewing frame/x along viewing vector](#) (Section 2.1.3) menu item. Now choose a view along y and repeat the procedure (select the [Build/Transform to viewing frame/y along viewing vector](#) (Section 2.1.3) after you are satisfied).

You are now ready to build a bilayer ([Build/Bilayer](#) (Section 2.1.3) menu item).

## Manual adjustment

The midpoint of the bilayer is initialized to  $z=0$  and the thickness to 35 Å. For manual adjustment by visual inspection, it is advisable to select a view along the x or y direction. You might also want to keep an eye at the **Interaction energy**. To obtain realistic (relative) energies, you need to select the **alpha-helical bundle** for bundles and the **beta-barrel** radiobutton for barrels, so that the appropriate lipid propensities are used ( $\alpha$ -helical bundle [Adamian\_2005]: Adamian, L., V. Nanda, W. F. DeGrado, and J. Liang. 2005. Empirical lipid propensities of amino acid residues in multispan alpha helical membrane proteins. *Proteins*. 59:496-509;  $\beta$ -barrel [Hayat\_2007]: Hayat, S., Y. Park, and V. Helms. 2007. A Statistical Approach to Deriving and Analyzing a Propensity Scale for Predicting Exposed Transmembrane Beta Barrel Residues from Protein Sequence. *Proc.. NIC. Workshop*. 36:137-140.). The interaction energy should be negative. If not, either the protein is not an integral membrane protein or the initial values for the bilayer centre/ membrane normal are poor.

Manual adjustment of the bilayer thickness for a  $\beta$ -barrel involves matching the outer planes to the upper and lower rim of the barrel by using the **Midpoint** and **Thickness** sliders. If the barrel does not have a well-defined upper or lower rim, you better use the **Fit** button. For an  $\alpha$ -helical bundle, you may also be able to identify upper and lower rims (most helices end at about the same z coordinate).

The free energy may increase during manual adjustment, as lipid propensities of amino acid residues are approximate values and contributions of residues in loops are not well accounted for. However, the free energy should not increase a lot and it should not be positive after the adjustment.

## Fitting the bilayer

If you are insecure about manually adjusting the bilayer midpoint and thickness, you can fit them by minimizing the interaction energy predicted from lipid propensities of solvent-accessible residues. To do that, make sure that the proper propensities (**alpha-helical bundle** or **beta-barrel** are selected and that the **fit membrane normal** checkbox is unchecked. Click the **Fit** button. To avoid being trapped in a local minimum, MMM performs a grid search for z from -10...10 Å and for the



thickness from 20..50 Å. This is followed by simplex minimization of the free energy near the global minimum found in the grid search.

If you are also insecure about the membrane normal direction, check the `fit membrane normal` checkbox before fitting. Two consecutive grid searches and a final simplex minimization are performed in this case. The energy surface is very rough in this four-parameter fit. To converge to the global minimum you may have to repeat the whole process: Click `OK`, select the `Build/Bilayer` menu item once again and fit once again with the `fit membrane normal` checkbox checked.

Lipid propensities are approximate quantities, hence the free energy of membrane insertion is an approximate quantity. Keep this in mind when making conclusions based on the orientation, position, and thickness of a fitted bilayer.

## Display control

The bilayer is visualized by three semi-transparent planes with black edges. The outer boundaries are pink, the centre plane is a light green. Checkboxes control which of these planes are shown. The lateral extension of the planes can be adjusted with the `<>` (extension) and `><` (compression) buttons. The visualization mode that is active when clicking the `OK` button is kept after closing the window. Clicking the `Cancel` button closes the window and deletes the visualization.

## Finishing bilayer building

You can accept the bilayer by clicking the `OK` button. If the membrane normal was fitted or the midpoint displaced, the structure is transformed into a frame where the membrane normal is the z axis and the central plane is at  $z=0$  and the model is redisplayed. Otherwise the coordinate frame is not changed. In either case the window is closed and the current visualization state of the bilayer is kept. All parameters of the bilayer are stored with the model.

The `Cancel` button serves for closing the window without any change to the model. All parameters of the bilayer are lost.

# 3.14 P-31 Mims ENDOR simulation

## P-31 Mims ENDOR simulations in MMM

The P-31 Mims ENDOR window is intended for the following tasks:

- simulation of P-31 Mims ENDOR spectra in spin-labeled structures with phosphorous-containing biomacromolecules, such as DNA, RNA, and phosphorylated proteins
- simulation of P-31 Mims ENDOR spectra in spin-labeled structures of peptides or membrane proteins in a lipid bilayer
- estimation of multi-spin effects on dipolar evolution functions and form factors for systems with more than two labels



- direct comparison of predicted and experimental data

This window can be accessed via the menu item [EPR/P-31 Mims ENDOR](#), (Section 2.1.7) but only if the current structure has at least one paramagnetic center. Paramagnetic centers can be spin labels or selected atoms. See the help page on [spin labeling](#) (Section 3.1) to find out how structures are spin labeled. See the help page on [selection](#) (Section 1.8) to find out how particular atoms are selected.

Furthermore, the structure with the paramagnetic center must have phosphorous atoms in it or a bilayer must be defined in this structure (see [how to build a bilayer](#)) (Section 3.13). The ENDOR window will open even if this is not the case, but all distance distributions and simulated spectra will be flat lines.

## Defining the spin system

The sites of the electron spin is selected in the `Labels & selected atoms` list box, where all spin labels and selected atoms are listed in MMM's [address format](#). (Section 1.6) The first tag in square brackets is the structure tag (often a PDB identifier), the second tag in parentheses the chain identifier, the number in curly brackets a number of the coordinate set (usually always 1, except for NMR structures) and the last number without any brackets the residue number. For selected atoms, the address is followed by a semi-colon and the residue name. The selected paramagnetic center is displayed in the `Spin system` panel and its position in current Cartesian coordinates of the structure is shown below the listbox. If a bilayer is defined, the bilayer central plane is the xy plane (last coordinate zero). Thus, the last coordinate is the distance from the bilayer central plane. For spin labels, the coordinates given are mean coordinates of the N-O bond midpoints of all rotamers.

Note that paramagnetic centers other than spin labels are defined by the location(s) of a single atom. Centers of spin density that are defined by several atoms are not supported in this version. If the selected atom has several alternate locations, all locations will contribute to the distance distribution.

The subset of phosphorous nuclei is selected by radiobuttons in the `P-31 nuclei` subpanel of the `Spin system` panel. Selection `molecule` considers only "structural" phosphorous, even if a bilayer model is present. Selection `bilayer` considers only phosphorous from an MMM bilayer model. Selection `both` considers all phosphorous nuclei. This is the default.

## The lipid bilayer model

The current lipid bilayer is the DPPC model of Kucerka et al. (N. Kucerka et al., Biophys. J. 2008, 95, 2356-2367) with scaling of the bilayer thickness, assuming a thickness of 35 Å for the DPPC model. The model is defined as a 3D grid of phosphorous locations with spacing of 0.5 Å. All positions within 25 Å are considered. Fractional P-31 populations at these positions are computed from the z-profile (along the bilayer normal) of phosphocholine groups (tag PCN) in the DPPC model. Fractional population is zero for positions within the protein slabs. Protein

slabs are computed on a 2D  $xy$  grid with 0.2 Å spacing by computing potential collisions with protein atoms at a certain  $x_i, y_i$  coordinate. Lennard-Jones interaction energies (only repulsive part) are computed separately for the top and bottom monolayer by summing over all protein atoms in the  $z$ -range of the monolayer, which extends from  $z=0$  to  $z \pm d/2$ , where  $d$  is the bilayer thickness. The lipid population at this  $x_i, y_i$  coordinate is computed as a Boltzmann population at 293 K. Populations below 0.05 are assumed to be zero. Holes in the upper and lower monolayer protein slabs are filled, as lipid molecules are not expected to insert into such holes (sometimes they do, but they may then be resolved in the crystal structure).

Only positions with  $z$ -profile population larger than 1% are considered in the final model. Typically, these are between 10'000 and 100'000 positions. The populations are scaled so that the integral over a slab of monolayer unblocked by protein with an area equal to the area-per-lipid for DPPC is unity.

## Loading experimental data

Experimental data can be loaded directly in Bruker's Xepr (EleXsys) format via the [Load Xepr](#) button. A Mims ENDOR spectrum with a frequency axis in MHz is expected. The static field read from the `.DSC` file is used to compute an initial estimate of the P-31 Zeeman frequency. Only the real part of the spectral intensities is considered. It is assumed that zero amplitude corresponds to zero stimulated echo and stimulated echo amplitude for an off-resonant r.f. pulse is positive. Hence, ENDOR peaks correspond to a decrease in positive spectral intensity. If in doubt, [ask us](#) about a suitable PulseSPEL program.

Data in a generic ASCII format can be read via the [Load ASCII](#) button. The first column must define a frequency axis in MHz, the second column real spectral intensities.

## Bckg. polynomial

Experimental ENDOR baselines (off-resonant intensities) may not be completely flat. MMM corrects for this on loading data, by assuming that the first and last 10% of data points are baseline (you should measure spectra with at least that much baseline). The first 2.5% of data points are neglected, as baseline may be spoiled here due to thermal effects. Baseline correction is done by fitting a polynomial function to the baseline ranges, extrapolating this function over the whole frequency axis, and dividing the spectrum by this function. This also normalizes baseline intensity to unity. The default baseline polynomial order is 0 (flat line). Polynomial order can be increased up to 5, but note that values above 2 indicate that something is wrong with the spectrum.

Use the `<` and `>` buttons or direct input into the edit field to adapt polynomial order. Display of the experimental spectrum changes automatically.

## Distance distribution

The distance distribution is automatically recomputed when you change selection of the paramagnetic center or of the contributing P-31 nuclei. Computation may take long if a bilayer is present and the distance distribution for the selected paramagnetic center was not computed before.

The distance distribution due to structural phosphorous is displayed in blue, the one due to phosphorous from the lipid bilayer in green, and the sum distribution in red. Often you may see only the red line, as one of the contributions may be totally missing or be insignificant.

Display range can be adapted and the plot detached in a separate Matlab window for copying, printing, and further inspection using the corresponding user interface elements in the `Distance distribution` panel.

## Mims ENDOR spectrum

Conversion from the distance distribution to the Mims ENDOR spectrum depends on the first interpulse delay  $\tau$  (tau) of the stimulated echo sequence (blind spot behavior) and on the length of the radiofrequency pulse (power broadening). MMM does not read these values from the `.DSC` file. You need to input them directly into the corresponding edit fields, if simulation needs to match experiment. Sensible default values of  $\tau=600$  ns and  $t_{rf}=10$   $\mu$ s are given.

Matching experiment also requires that the center frequency (P-31 Zeeman frequency) of the experimental spectrum is known. When loading Xepr data, this frequency is computed from the magnetic field. Note however that the magnetic field is measured by a Hall probe with limited precision and calibration may be surprisingly poor, as the Hall probe in pulse EPR spectrometers is located far from the magnet center. In many cases, you are better off by checking the `Fit center freq.` checkbox. This is not the default, as it may fail spectacularly if the experimental and simulated distance distributions do not match. The center frequency can also be input manually. The experimental spectrum as displayed in the plot should be as symmetric as possible with respect to zero frequency offset ( $f-f_0$ ).

The maximum ENDOR effect (strongest decrease of stimulated echo intensity in percent) is displayed as it provides a fast estimate of P-31 proximity to the paramagnetic center. Furthermore the full width at 1.5% signal reduction is displayed if the maximum signal reduction is at least 1.5%.

## Copying and printing plots

Both plots can be detached as separate Matlab figures via the corresponding `Detach` buttons. Use the figure menu to copy or plot (or edit) them.

## Saving simulated data

The **Save** button in the left bottom corner of the window can be used to save the simulated data in text (ASCII) format, similar to the output format of DeerAnalysis. In any case the three files `*_res.txt`, `*_distr.dat` and `*_spc.dat` are written, where `*` stands for the basis file name that can be given in a file dialog window. The file `*_res.txt` contains information on simulation and fit parameters, the file `*_distr.dat` the distance distribution and the file `*_fit.dat` the form factor. The contents of file `*_res.txt` are self-explaining.

In file `*_distr.dat` the first column is the distance axis in units of Angstroem. The second column is the predicted distance distribution. The distance range extends from 0 to 25 Å.

In file `*_spc.dat` the first column is the frequency offset axis in kHz. The second column is the simulated spectrum.

A third file `*_fit.dat` exists if and only if experimental data were loaded. This file contains in the first column the frequency offset axis in kHz, in the second column the normalized, baseline corrected and center-frequency shifted experimental data, and in the third column the simulated spectrum interpolated to the frequency axis increment of the experimental spectrum.

# 4 Protocols

## 4.1 Protocol for spin labeling and DEER

### A three-step tutorial

This protocol serves at the same time as worked example that you can reproduce from files provided together with MMM.

For *in silico* spin labeling, visualization, and spin labeling you take the following three steps:

- [Loading the protein](#) (Section 4.1.1)
- [Site scan](#) (Section 4.1.2)
- [DEER simulation](#) (Section 4.1.3)

### 4.1.1 Loading the protein

#### How to load the protein

The PDB file of the protein under investigation (in this example the major light harvesting complex LHCII, PDB code 2BHW) can be loaded with:

File/new/from PDB/local file (if the original PDB is saved in your own folder, in this case the LHCII subfolder of MMM. Beware that several protein visualization programs do not save the complete PDB information, so prefer the original PDB)

Or

File/new/from PDB/direct PDB access (providing the PDB identifier, in this case 2BHW, the original PDB is loaded from the PDB website)

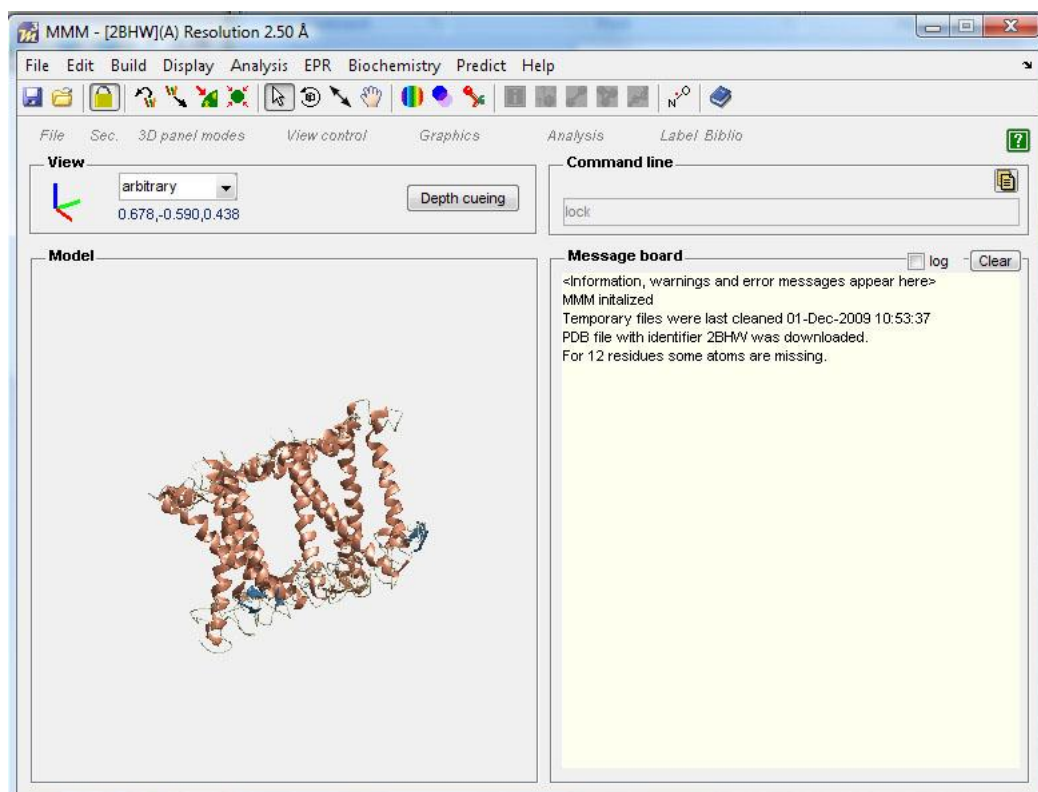
The file is uploaded and visualized in the model window in ribbon representation. If you use the local file, answer the question about running the initialization script with No. The message board shows information, warnings and error messages.

In this case:

MMM initialized

Temporary files were last cleaned 01-Dec-2009 (this is the last time MMM cleaned the temp files. When the log on the message board is activated, the temp files are automatically created and stored for 30 days. After that a message will appear asking if you want to delete them.)

For 12 residues some atoms are missing (important information for the rotamers calculation)



You can save the model at any stage of your analysis, with:

File/Save as

By default a Matlab variable (.mat) file will be saved named: *MMM\_year-month-day\_h-min-s.m*, e.g. *MMM\_2009-12-14\_09-30-43.m*). However, you can choose any file name you wish. Next time you will open MMM you can load the saved model, using:

File/open/ and selecting *MMM\_2009-12-14\_09-30-43*

## Change symmetry frame

For easier display of the protein, in some cases it is useful to change the coordinate system of the PDB file. For example you can choose to have the z axis of the coordinate system perpendicular to the membrane plane (for membrane proteins) or parallel to a special rotation axis. In the case of 2BHW, a homo-trimer, you can change the coordinate system based on the molecular symmetry.

For that, select residues or cofactors which define the symmetry of the molecule. In this case the carotenoids LUX501 in chains A, B, C will define the C3 symmetry axis.

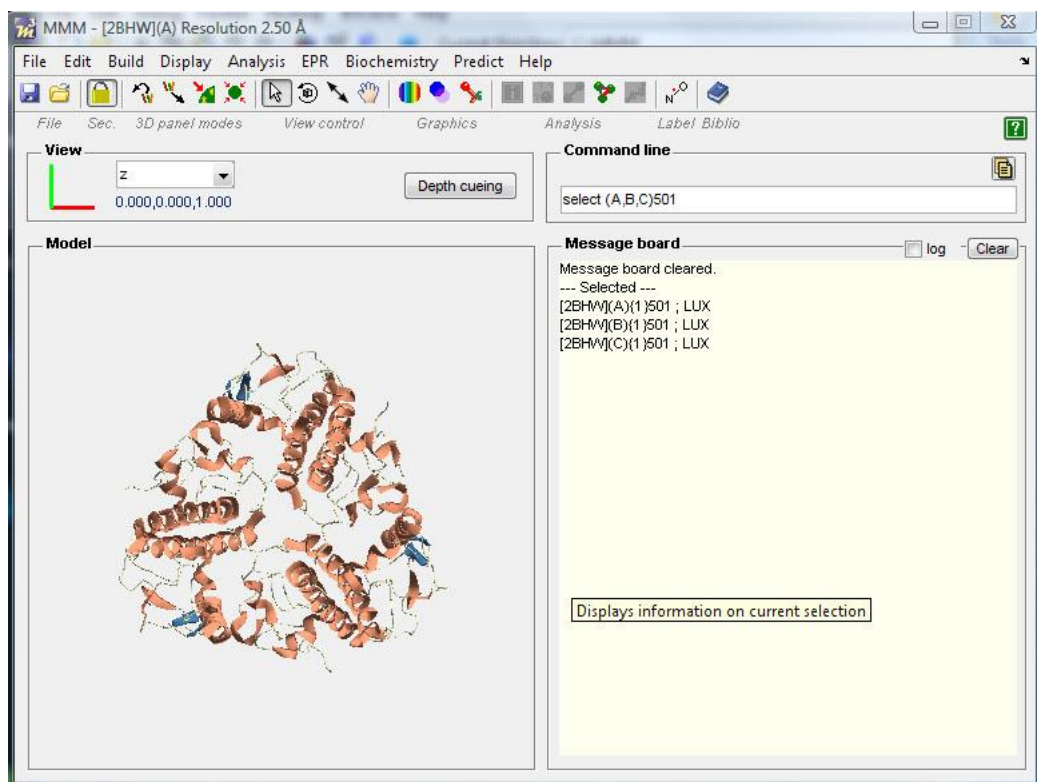
To select them write the command line:

```
select (A,B,C)501
```

The three carotenoids are selected and can be used to build the new symmetry frame by:

Edit/Symmetry Frame

The protein will be redisplayed with the z axis parallel to the C3 symmetry axis. The view is set along the z axis.



The view can be easily changed by changing the axes in the **View** panel.

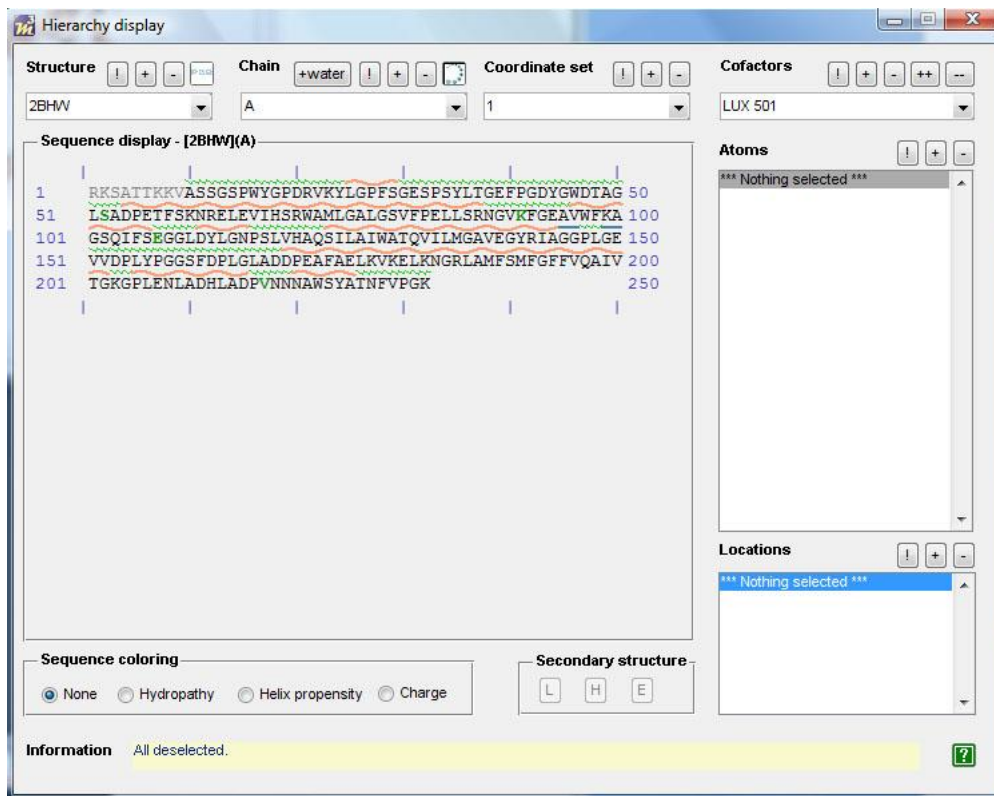
You may now continue with the [Site scan](#). (Section 4.1.2)

## 4.1.2 Site scan protocol

### Hierarchy window, residues selection

MMM provides a Hierarchy display (**Display/Hierarchy**) which displays the primary sequence (with missing residues or atoms marked), the chains (A, B, C), and the cofactors (in this case carotenoids, chlorophylls, and lipids) as defined in the PDB file. Selection of residues or chains can also be performed here.





The sequence color code means:

- **Black** amino acids: all atoms are present
- **Grey** aa: the backbone is missing
- **Green** aa: atoms of the side chain are missing
- **Red** aa: selected aa, last selection is a bit lighter than older ones

In the primary sequence shown in the hierarchy window all residues can be clicked.

Left click on residue S106 selects *only* this residue (the previous selection LUX 501 (A,B,C) will be discarded). This corresponds to the command line:

```
select [2BHW] (A) {:}106
```

Middle click (or Shift + left click) on residue S160 adds this residue to the existing selection. This corresponds to the command line:

```
select [2BHW] (A) {:}106,160
```

or simply (because only one PDB identifier and one model is present)

```
select (A)106,160
```

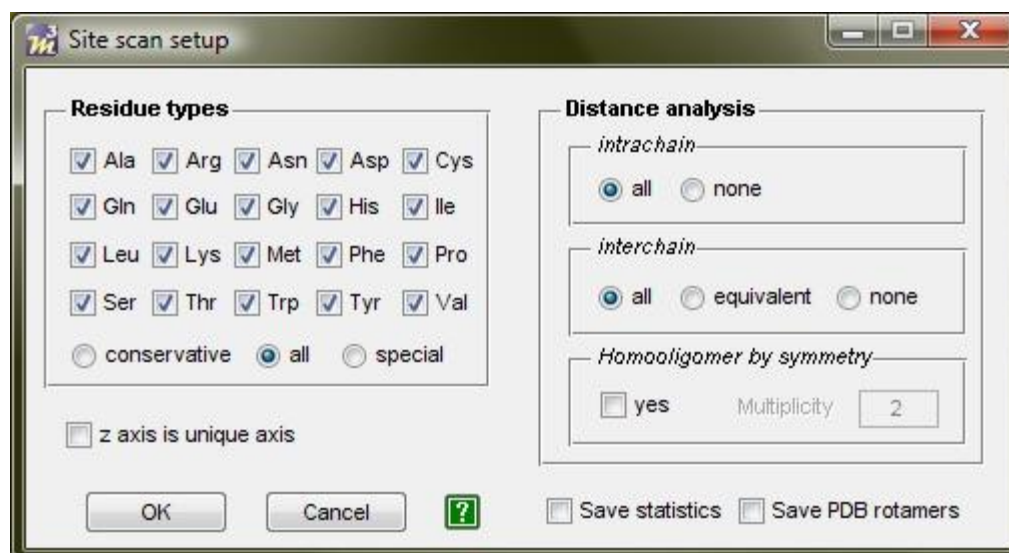
For more information on the hierarchy window, use the  button.

## Site scan: rotamer analysis on 106 and 160

To spin label *in silico* positions 106 and 160 (already selected) and simulate the interspin distances, [spin labels](#) (Section 3.1) can be attached to the protein using a rotamer library approach.

To compute the rotamers go to:

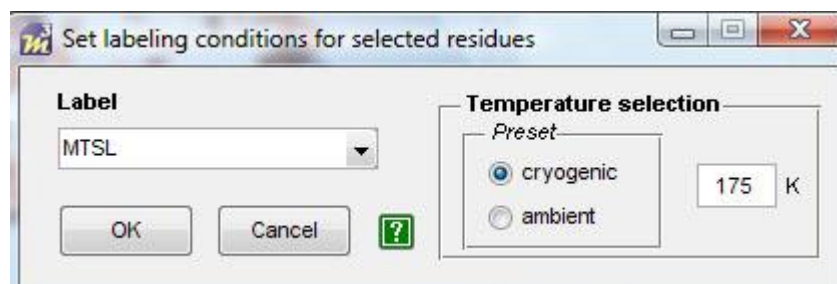
EPR/Site scan/selected residues



This window allows choosing residue types, intra- and/or inter-chain distances. Ignore the additional saving options for methodological work (or look them up in the [complete spin labeling description](#)) (Section 3.1). In the case at hand two sites (106, 160) are already selected, thus just press **OK**.

**Remark:** If you want to calculate favorable attachment sites by scanning the whole protein, select the three chains A, B, C. In the site scan you can remove side chains which you do not want to label (e.g. Pro, Gly...) and decide if you need all intra-chain distances in the further analysis.

Now choose the spin label (MTSL or IA-PROXYL) and the temperature defining the rotamers populations (cryogenic or ambient).

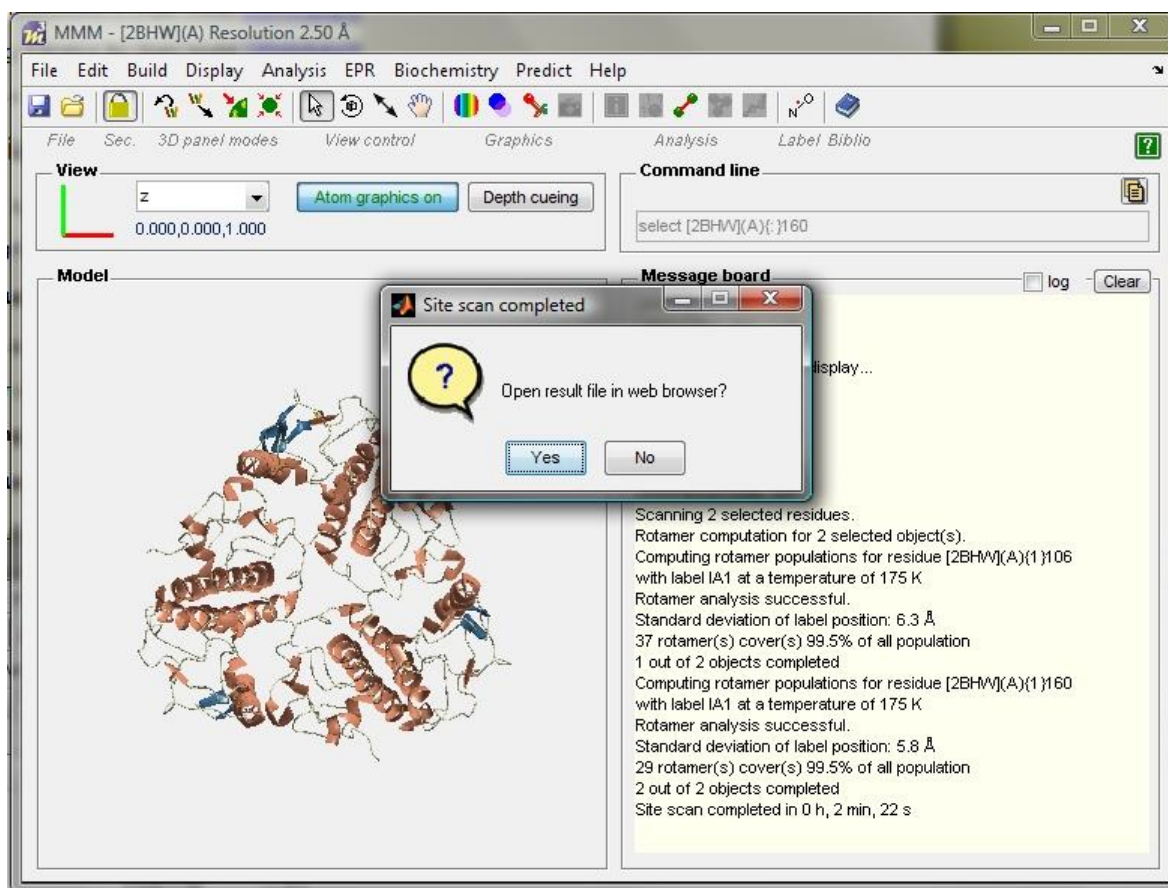


Choose **IA-PROXYL** at **175 K**. Then press **OK**.

You can choose to save the Site analysis in HTML format (recommended). The default file name (you can change it) is:

MMM-site\_scan\_2009-12-14\_10-47-19.html

Now the calculation is running and you can see the computation details in the message board. When the site scan is completed a message will appear asking if you want to open the result file in web browser (For scanning the entire protein it is recommended to run the program overnight).



The result file is opened in the Matlab browser and looks like this:

MMM spin labeling site scan

File Edit View Go Debug Desktop Window Help

Location: file:///C:/Users/jeschke/Documents/MMM/documentation/MMM\_site\_scan\_2009-1

## MMM spin labeling site scan

For missing residues no rotamer had sufficiently low energy.

### Overview

Spatial restrictions in chain model [\[2BHW\]\(A\){1}](#)

Labeling statistics for chain model [\[2BHW\]\(A\){1}](#)

Distance analysis within chain model [\[2BHW\]\(A\){1}](#); [DEER](#); [CW](#); [long \(DEER\)](#); [too short](#); [too long](#); [too tight](#)

### Spatial restrictions for sites in chain model [\[2BHW\]\(A\){1}](#)

(sorted by ascending spatial distribution of NO group)

(very tight positions, where labeling may fail, are displayed at the end)

**Residue [\[2BHW\]\(A\){1}160 \(Ser\)](#) labeled by IA1 in loop**

NO position r.m.s.d. 0.58 nm

Number of rotamers: 29 with partition function 0.58722

**Warning:** This residue is involved in a binding site

**Residue [\[2BHW\]\(A\){1}106 \(Ser\)](#) labeled by IA1 in loop**

NO position r.m.s.d. 0.63 nm

Number of rotamers: 37 with partition function 1.38553

[To overview](#)

### Labeling statistics for sites in chain model [\[2BHW\]\(A\){1}](#)

In 2 residues there are 0 tight sites where labeling might fail.

### Loop sites

Mean r.m.s.d. 0.60 nm

Minimum r.m.s.d. 0.58 nm

Maximum r.m.s.d. 0.63 nm [To overview](#)

### Intrachain distances in chain model [\[2BHW\]\(A\){1}](#)

(sorted in categories by ascending relative distribution width)

**Most favorable for DEER measurements (1.8 nm <= < 4.5 nm)**

Distance between residues [\[2BHW\]\(A\){1}106 \(Ser\)](#) and [\[2BHW\]\(A\){1}160 \(Ser\)](#) is 4.15 nm

standard deviation is 0.52 nm and relative width 12.6 %

[To overview](#)

Done

The output of the rotamer calculation gives important information on the r.m.s.d. of the NO positions attached at one site and the number of rotamers (correlated to the accessible space, thus the mobility of the spin label), the partition function (numbers <0.05 denote sites difficult to label due to steric hindrance).

At this point save the model (recommended). The model (MAT file) will be saved with the symmetry frame given, and the results of the rotamer calculation. No spin labels have yet been attached to the model, so it can be used as starting model again to calculate the rotamers on other sites.

When you reload the model residues, for which rotamers were computed in a previous site scan, can be found via the `Edit/Find by keyword` menu item. The keyword for this is "rotamers computed". Via the `Show annotation` button in the keyword finder you can also access the statistics information (number of rotamers, partition function, NO position r.m.s.d.) for a residue without searching in the HTML file.

In the following, we assume a view in the lipid bilayer plane, perpendicular to the C3 symmetry axis. Select this by setting `x` in the `View` panel popup menu.

Now attach the pre-computed rotamers to the sites using the menu item:

`EPR/Attach pre-computed rotamers`

or the button .

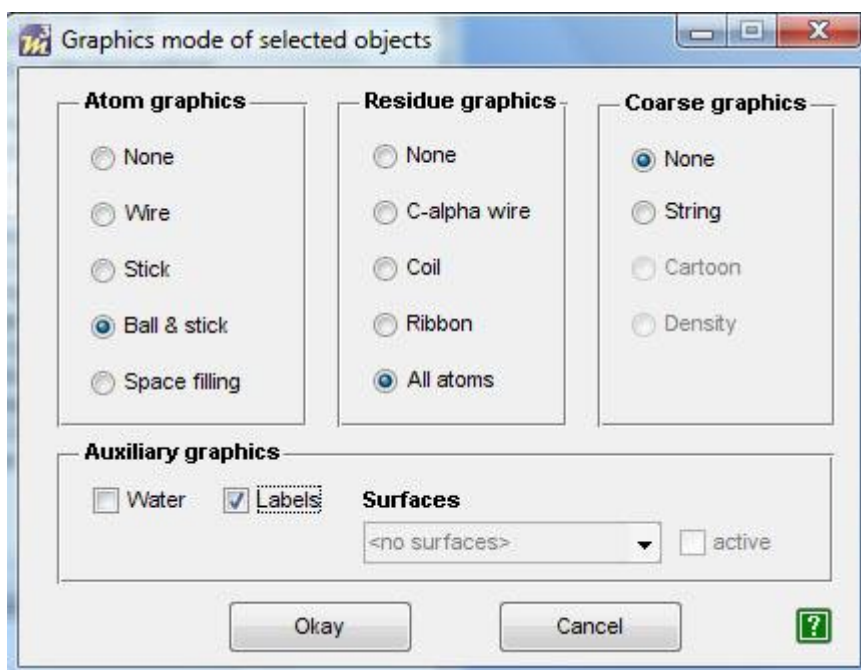
Choose the rotamers you computed (`IA-PROXYL`, `cryogenic`).

At this point the original Ser residues at positions 106 and 160 have been replaced by the spin labeled side chains computed (n rotamers). This attachment cannot be undone! Thus, the model from now on will contain the attached rotamers, and it is not recommended to use it to perform additional calculations on other sites (the attached rotamers will be taken into account in the new calculation).

To show the attached rotamers on the model use the menu item:

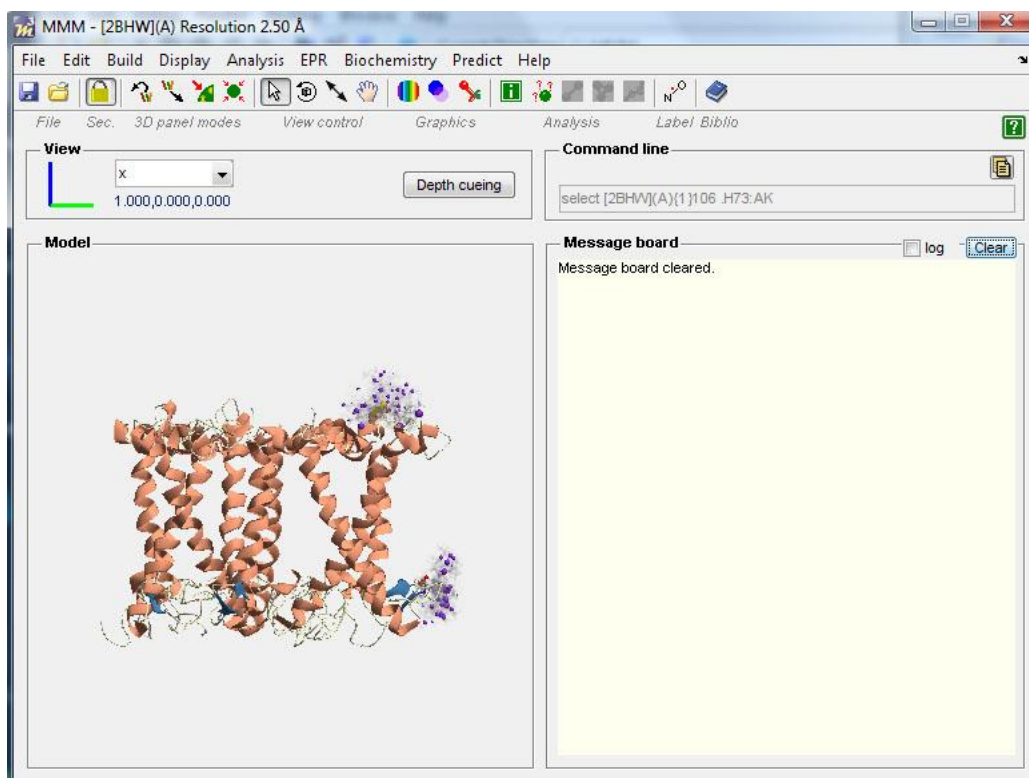
`Display/3D display mode`






Choose Atom graphics/Ball & stick, Auxiliary graphics/Labels, Okay

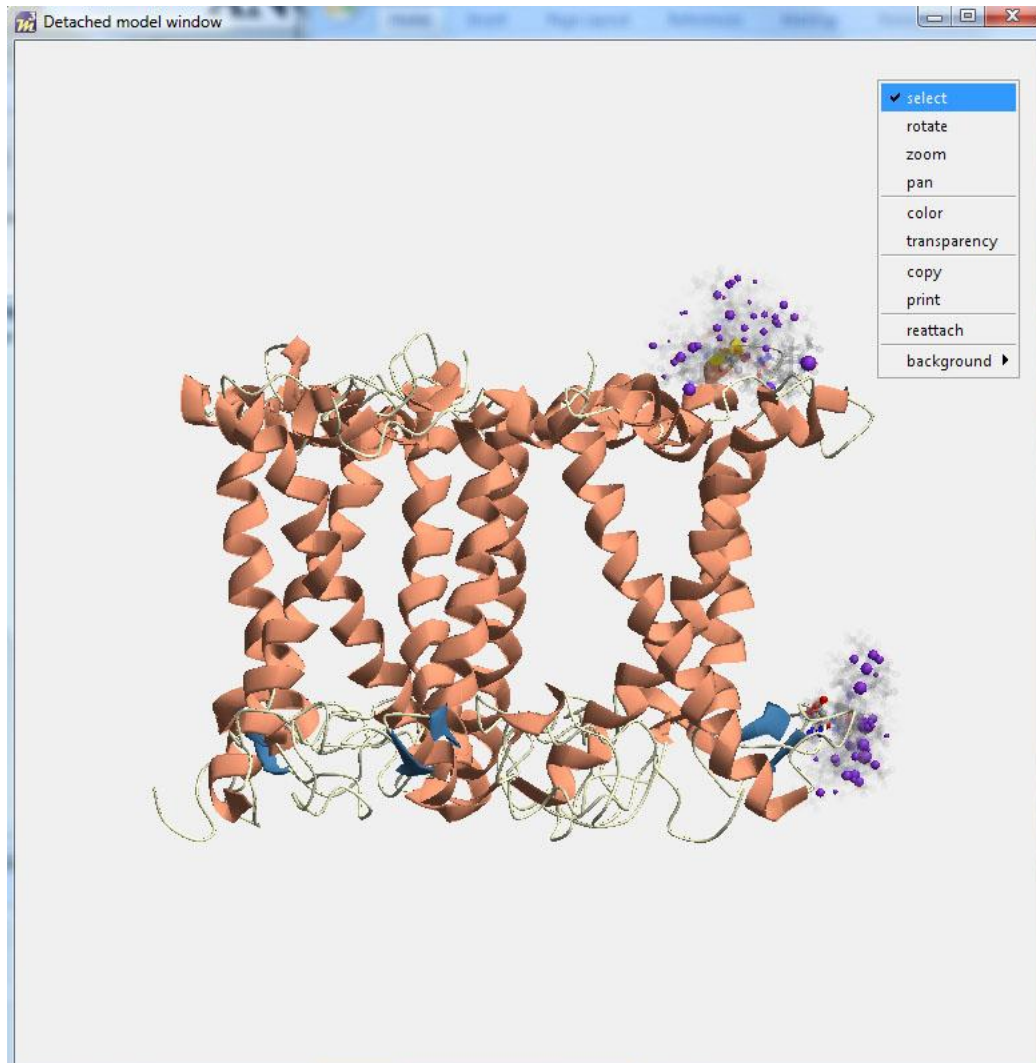
This will take a while, but eventually you will see a display like this:




The two spin labeled sites are shown. The violet clouds denote the position of the NO group, with size according to population. The rotamers are displayed in ball and stick representation.

The yellow color on the side chains will disappear if you give the command `unselect` \*.

To copy the figure, detach the model with the  toolbutton or right click on the model and choose detach.



On the detached window, right-click to see the context menu, choose copy (this takes a while, the model window will shortly flash white at the moment of copying) and paste the resulting figure (in the clipboard in extended metafile format) where you need. You can also use the camera toolbutton  in the main window for copying. If you want to save the picture as a file in one of many graphics format, you can use the menu item [File/Export visualization](#) (Section 2.1.1) of the main window.

To increase resolution, you can enlarge the window and zoom the protein region under investigation with the tools provided.



You may now want to switch atom graphics off to get faster response from the graphical user interface. To do so, click twice on the **Atom graphics on** button in the **View** panel of the main window. The button will change to red **Atom graphics off**.

You may save the model with the spin labels attached, but it is recommended not to overwrite the model that was saved directly after the site scan. When you reload a model with attached labels, you can find the labeled residues via the **Edit/Find by keyword** menu item. The keyword for this is "spin label attached".

You may now continue with a [DEER simulation](#). (*Section 4.1.3*)

## 4.1.3 DEER simulation protocol

### Simulate DEER traces based on the computed rotamers

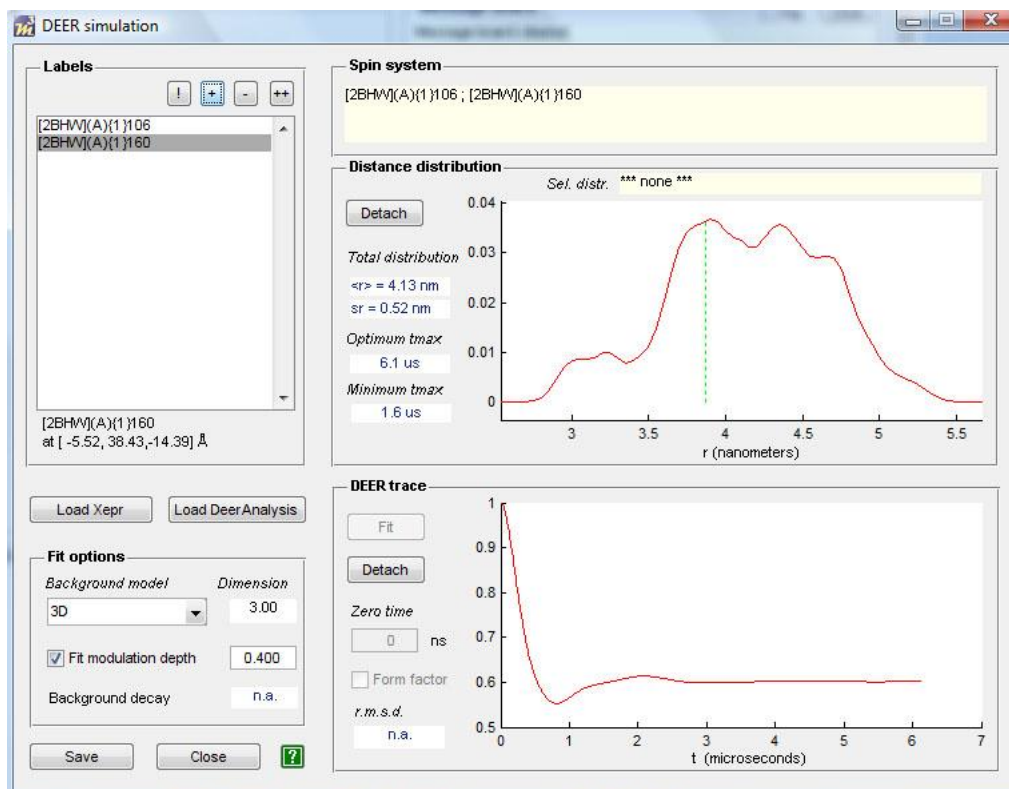
MMM allows simulating a DEER trace and the distance distribution based on the computed rotamers. To simulate the DEER trace use the menu item:

**EPR/DEER**

A window will appear where you can select which computed side chains to use for the DEER simulation (in this case there are only two computed sites: 106 and 160).

Select the site 106, press **+** to add it in the spin system, then select the second site, press **+**.

Once you selected the first pair of spins, the DEER trace and the distance distribution will appear.



To copy the figures, **Detach** them and copy/paste the Matlab figures by using the menu of the detached figures.

## Compare simulated and experimental DEER traces

To compare the computed DEER trace with the experimental one, choose

**Load Xepr** (to load the original DEER trace in Xepr format) And choose the **Fit options** for the background.

Or

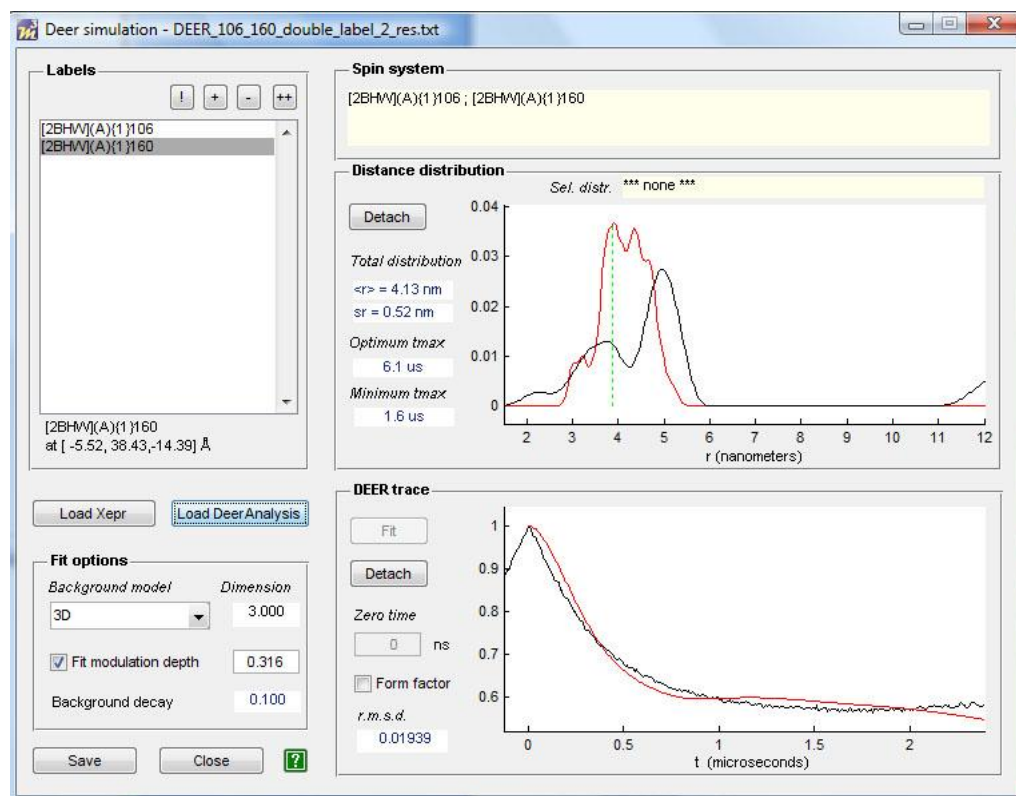
**Load Deer Analysis** (to load the processed DEER data that were saved with DeerAnalysis)

The latter option is recommended in the case the background correction was difficult to perform, and the analysis has been optimized. However, if you have doubts about the background correction performed in DeerAnalysis, you should also try whether the model-based approach of MMM can find a more convincing fit.

You find the experimental data and DeerAnalysis result for this label pair in the LHCII subdirectory of MMM.

The comparison will appear in the same window. To copy the comparison plots, detach the figures as before. With the **Save** button you can save the underlying data

as ASCII files and process them in other programs (e.g. Origin or, if you must, Excel).



Comparison of the computed and experimental traces gives a qualitatively good agreement in this case. Beware that close to position 106 there is a residue with missing atoms (green amino acid in the Hierarchy window). The possible influence of missing atoms on the calculation must be investigated.

## 5 Conditions and Acknowledgment

### 5.1 License

#### License text

Copyright © 2009-2015, Yevhen Polyhach, Stefan Stoll & Gunnar Jeschke

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Contributed software

See the [Contributed software section](#) (Section 5.3) of the credits page for the copyright holders of contributed programs.

Also see [third-party software](#) (Section 5.2) for a list of programs that can be installed to enhance MMM's functionality, *after* you have obtained a license from the original authors (free for academic use).

## 5.2 Third-party software

### Why does MMM depend on software that is not included in the distribution?

For many problems in bioinformatics, good software exists that is free for academic use. In many cases the underlying algorithms are not trivial, so that recoding would be time-consuming and would result in modules that are much less tested (and could be much less trusted) than established software.

Existing programs are of three types:

- programs that are accessible only via web server
- programs that can be downloaded, but are accessible only via their own user interface
- programs with command line access and clearly defined input and output formats

The policy of MMM about these types of programs is as follows. Communication with web servers is via saving input in MMM, uploading it with a browser, downloading the results, which are sometimes provided via e-mail, and importing them into MMM.

Some web servers allow for direct queries, specification of a well defined output format and provide a web page in this format. In this case MMM connects directly. There are also web services with clearly defined interface. The current version does not yet access such services, but future versions will communicate directly with them.

For downloadable programs with their own user interface and no command-line version communication is via saving files from MMM (usually in PDB format), reimporting them into the third-party program, processing them, exporting them from this program, and reimporting into MMM.

For command-line programs with clearly defined interface, MMM provides direct access via menu items or dedicated windows. This functionality works only when the corresponding program is installed. Generally, such programs are not bundled with MMM. The user is responsible for obtaining the license from the original authors, downloading and installing the program. Third-party programs are bundled with MMM only when the original authors explicitly permit redistribution. In these cases, they are listed in the [Credits](#). (Section 5.3)

## General installation procedure for third-party software

In the list below, you find web access links for this software ([please report broken links](#)). Check whether you need a license (sometimes you can simply download), apply for the license, download the software. Now there are two possibilities. In the first one, install the software in the `third_party` subdirectory of MMM. You can have nested subdirectories in this subdirectory. In principle, third-party software packages can also have their own subdirectory under the MMM main directory. After a restart, MMM will automatically find the software (as long as the authors did not change the calling syntax ([please report such cases](#))).

As a second possibility, you can install the program wherever you wish (in fact, you might already use the program and it *is* installed somewhere else). In this case, Matlab needs to know the path to the program (see [Viewing and Setting the Search Path](#) in Matlab help). In some cases, for instance for Modeller, MMM must be able to write in a subdirectory of the program directory. Under Windows, such programs cannot be installed in `C:/Programs` (or similar), they must be installed in your Documents folder.

## DSSP (Secondary structure assignment)

This is the standard program for unambiguous definition of secondary structure. It can be downloaded from a [DSSP Homepage](#) at Radboud University Nijmegen where the group of [Gerd Vriend](#) maintains it. We use it in MMM to reassign secondary structure in PDB files and to obtain initial estimates on accessibility of residues. The proper citations are: Joosten RP, Te Beek TAH, Krieger E, Hekkelman ML, Hooft RWW, Schneider R, Sander C, Vriend G, NAR 2010; [DOI access](#) and Kabsch W, Sander C, Biopolymers. 1983 22 2577-2637. A MS Windows version of DSSP is

bundled with MMM under the Boost licence (see `/third_party` subdirectory for the license file).

## **Modeller (homology modeling without or with experimental restraints)**

This is an extensive, well tested homology modeling package from the [Sali lab](#). Actually, in this case the tail MMM is wagging the dog Modeller. However, Andrej Sali encourages programming of user interfaces for Modeller. The proper citation:

N. Eswar, M. A. Marti-Renom, B. Webb, M. S. Madhusudhan, D. Eramian, M. Shen, U. Pieper, A. Sali. Comparative Protein Structure Modeling With MODELLER. Current Protocols in Bioinformatics, John Wiley & Sons, Inc., Supplement 15, 5.6.1-5.6.30, 2008. has the short names `Eswar:2008_modeller`.

We strongly recommend to install Modeller for MMM use in a subdirectory of MMM. In any case it cannot be installed in a Windows program directory, since MMM must be able to write a batch file in the same directory where the executable resides.

Different Modeller versions have different names of the executable. This name can be specified in the preferences window, for instance `mod9v10` for version 9.10 published in September 2011.

MMM can use Modeller to repair gaps in PDB structures (mostly missing loops). We plan to provide an interface for homology modeling in the next version.

## **MSMS (Solvent accessible surface)**

This neat program by [Michel Sanner](#) can be downloaded from the [Molecular Surfaces Computation Homepage](#). We use it in MMM to visualize solvent-accessible surfaces and to compute water and oxygen accessibility parameters for spin labels. The proper citation:

Sanner, M.F., A. J. Olson, and J. C. Spehner. 1996. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*. 38:305-20.

has the short name `Sanner:1996_msms`.

## **MUSCLE (Sequence alignment)**

This compact, fast, and easy-to-interface program is downloadable from [Robert C. Edgar](#)'s homepage and can compute pairwise as well as multiple sequence alignments. The proper citation:

Edgar, Robert C. (2004), MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research* 32(5), 1792-97.

has the short name `Edgar:2004_muscle`.

Starting from MMM 2011, MUSCLE is bundled with MMM with permission from Robert C. Edgar.

## SCWRL4 (Sidechain generation and repacking)

Software for generation or repacking of sidechains by use of a rotamer library. This can be downloaded from the [Dunbrack lab webpage](#).

G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack, Jr. 2009. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins* 77:778-795.

has the short name `Krivov:2009_scwrl4`.

## Tinker (molecular force field computations)

Tinker is a modular package for molecular force field and molecular dynamics computations that can be obtained from the [Ponder lab](#). Tinker interfacing is prepared and tested, but not yet used in standard MMM functionality (there is an advanced feature in elastic network model based fitting of structural transitions). If you want to access Tinker in your own programming, see for example routines

```
put_Tinker_coor.m, read_Tinker_xyz.mm, get_Tinker_energy.m,  
get_Tinker_minimize.m.
```

The following Tinker binaries and force field parameter files should be on the Matlab path *in the same directory*:

pdxyz.exe, minimize.exe, analyze.exe, testgrad.exe, xyzpdb.exe, oplua.prm, oplaa.prm, amber99.prm, charmm19.prm

Citation:

J.W. Ponder, F.M. Richards 1987. *J. Comput. Chem.* 8:1016-1024.

## How to cite third-party software

Whenever you use functionality of MMM that does depend on third-party software, you should cite not only MMM but also this software. MMM makes this easy for you by automatically adding the appropriate reference to the bibliography of the model (if the original authors provided one or we could find one). If you are an original author who is unhappy with the selected citation, [please contact us](#). The `short name` for the appropriate citation is given in the description of the particular module above.

It is the responsibility of an MMM user that the citation appears not only in the model bibliography, but also in papers, if the use of this program was substantial in solving a problem or preparing a figure.



## 5.3 Credits



### **Rotamer libraries and spin labeling**

Yevhen Polyhach, ETH Zurich

### **Speed up, site scan output, user interface advice**

Stefan Stoll, University of Washington

### **Broader compatibility of PDB writer**

Morgan Bye, Henry Wellcome Unit for Biological EPR, University of East Anglia, Norwich. U.K.

### **User interface, graphics, architecture of the program and data structure**

Gunnar Jeschke, ETH Zurich

### **Consultant on design and functionality, protocols**

Enrica Bordignon, ETH Zurich

### **Helpful discussions and suggestions**

Christoph Dockter, Gregor Hagelueken, Daniel Hilger, Heinrich Jung, Harald Paulsen, Kay Diederichs, Tona von Hagens

### **Example DEER data for LHCII**

Aleksei Volkov

### **Contributed software**

*obtained via [Matlab Central](#) unless links are given*

### **General machinery for 3D rotations and Euler angles**

SpinCalc.m by *John Fuller*, 2009

## Rotation, zoom, and pan of the 3D model

view3D.m by *Torsten Vogel*, 1999

## 3D tube (basis of ribbon models)

tubeplot.m by *Janus H. Wesenberg*, 2004

## Fast 3D sphere plot

BuildSphere.m by *Luigi Giaccari*, 2009

## Periodic table of elements

pertable.m by *Gerhard Nieuwenhuijs*, 2009

## Status bar for computation progress

statusbar.m by *Marcel Leutenegger*, 2007

## Coordinate transform of density cubes

affine.m by *Jimmy Shen*, 2009 (renamed affine\_cube.m)

## Sequence alignment

[muscle.exe](#) by *Robert C. Edgar*, 2010. Please cite:

Edgar, Robert C. (2004), MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research* 32(5), 1792-97.

## Data

### Residue-specific Ramachandran plots

digitized from PDF of: Sven Hovmöller, Tuping Zhou, Tomas Ohlson (2002) *Conformations of amino acids in proteins*. *Acta Cryst. D* **58**, 768-776.

## References

The following reference is recommended when you use MMM for predictions of spin label conformations and distance distributions:

- Y. Polyhach, E. Bordignon, G. Jeschke, *Phys. Chem. Chem. Phys.*, 2010, . DOI: 10.1039/c0cp01865a [Full text](#)

The following additional reference is recommended when you use MMM for site scans, to decide which sites to label:

- Y. Polyhach, G. Jeschke, *Spectrosc. Int. J.*, 2010, **24**, 651-659. [Full text](#)

These references are automatically added to the model when you perform rotamer computations or a site scan for whole chains.

The following additional references are recommended when you use MMM for modeling of structural transitions with elastic networks:

- G. Jeschke, *J. Chem. Theor. Comput.*, 2012, **8**, 3854-3863. [Full text](#)
- G. Jeschke, *Z. Phys. Chem.*, 2012, **226**, 1359-1414. [Full text](#)

If you use the network localization feature, please cite the original work that proposed a very similar approach:

- B. Gaffney, M.D. Bradshaw, S.D. Frausto, F. Wu, J.H. Freed, P. Borbat *Biophys. J.*, 2013, **103**2134-2144. [Full text](#)

## Financial support

The open-source project MMM is supported by Swiss National Science Foundation, grants SNF 200021\_121579, 200020\_132255, 200020\_14441, and 20020\_157034.