The Coot User Manual

Paul Emsley
paul dot emsley at bioch dot ox dot ac dot uk

Table of Contents

1 Introduction 1
- 1.1 Citing Coot and Friends 1
- 1.2 What is Coot? 1
- 1.3 What Coot is Not 1
- 1.4 Hardware Requirements 2
  - 1.4.1 Mouse 2
- 1.5 Environment Variables 2
- 1.6 Command Line Arguments 3
- 1.7 Web Page 4
- 1.8 Crash 4

2 Mousing and Keyboarding 5
- 2.1 Next Residue 5
- 2.2 Keyboard Contouring 5
- 2.3 Mouse Z Translation and Clipping 5
- 2.4 Keyboard Translation 6
- 2.5 Keyboard Zoom and Clip 6
- 2.6 Scrollwheel 6
- 2.7 Selecting Atoms 6
- 2.8 Virtual Trackball 6
- 2.9 More on Zooming 7

3 General Features 8
- 3.1 Version number 8
- 3.2 Antialiasing 8
- 3.3 Molecule Number 8
- 3.4 Display Issues 9
  - 3.4.1 Stereo 9
  - 3.4.2 Pick Cursor 9
  - 3.4.3 Origin Marker 9
- 3.5 Screenshot 9
- 3.6 Raster3D output 10
- 3.7 Display Manager 10
- 3.8 The Modelling Toolbar 11
- 3.9 The file selector 11
  - 3.9.1 File-name Filtering 11
  - 3.9.2 Filename Sorting 11
  - 3.9.3 Save Coordinates Directory 11
- 3.10 Scripting 11
  - 3.10.1 Python 12
    - 3.10.1.1 Python Commands 12
  - 3.10.2 Scheme 12

3.10.3 Coot State 12
3.10.4 Key Binding 13
3.10.5 User-Defined Functions 13
3.11 Backups and Undo 13
3.11.1 Redo 14
3.11.2 Restoring from Backup 14
3.12 View Matrix 14
3.13 Space Group and Symmetry 15
3.14 Recentring View 15
3.15 Views 15
3.16 Clipping Manipulation 16
3.17 Background colour 16
3.18 Unit Cell 16
3.19 Rotation Centre Pointer 16
3.20 Orientation Axes 16
3.21 Pointer Distances 16
3.22 Crosshairs 16
3.23 3D Annotations 17
3.24 Frame Rate 17
3.25 Program Output 17

4 Coordinate-Related Features 18

4.1 Reading coordinates 18
4.1.1 A Note on Space Groups Names 18
4.1.2 Read multiple coordinate files 18
4.1.3 SHELX .ins/.res files 18
4.2 Atom Info 19
4.3 Atom Labeling 19
4.4 Atom Colouring 19
4.5 Bond Parameters 20
4.5.1 Bond Thickness 20
4.5.2 Display Hydrogens 20
4.5.3 NCS Ghosts Coordinates 20
4.5.4 NCS Maps 21
4.5.5 Using Strict NCS 21
4.6 Download coordinates 21
4.7 Get Coordinates and Map from EDS 21
4.8 Save Coordinates 22
4.9 Setting the Space Group 22
4.10 Anisotropic Atoms 22
4.11 Symmetry 22
4.11.1 Missing symmetry 23
4.12 Sequence View 23
4.13 Print Sequence 23
4.14 Environment Distances 23
4.15 Distances and Angles 23
4.16 Zero Occupancy Marker 23
4.17 Atomic Dots 24

4.18 Ball and Stick Representation 24
4.19 Mean, Median Temperature Factors 24
4.20 Secondary Structure Matching (SSM) 24
4.21 Least-Squares Fitting 25
4.22 Ligand Overlaying 25
4.23 Writing PDB files 26

5 Modelling and Building 27

5.1 Regularization and Real Space Refinement 27
  5.1.1 Dictionary 28
  5.1.2 Sphere Refinement 28
  5.1.3 Refining Specific Residues 29
  5.1.4 Refining Carbohydrates 29
  5.1.5 Planar Peptide Restraints 29
  5.1.6 The UNK residue type 30
  5.1.7 Moving Zero Occupancy Atoms 30
5.2 Changing the Map for Building/Refinement 30
5.3 Rotate/Translate Zone 30
5.4 Rigid Body Refinement 30
5.5 Simplex Refinement 31
5.6 Post-manipulation-hook 31
5.7 Baton Building 31
  5.7.1 Undo 32
  5.7.2 Missing Skeleton 32
  5.7.3 Building Backwards 32
5.8 Reversing Direction of Fragment 33
5.9 C\$alpha$ -> Mainchain 33
5.10 Backbone Torsion Angles 33
5.11 Docking Sidechains 33
5.12 Rotamers 34
  5.12.1 Auto Fit Rotamer 34
    5.12.1.1 Backrub Rotamers 35
  5.12.2 De-clashing residues 35
5.13 Editing chi Angles 35
5.14 Torsion General 36
  5.14.1 Ligand Torsion angles 36
5.15 Pep-flip 36
5.16 Add Alternate Conformation 36
5.17 Mutation 37
  5.17.1 Mutating DNA/RNA 37
  5.17.2 Multiple mutations 37
  5.17.3 Mutating to a Non-Standard Residue 38
  5.17.4 Mutate and Autofit 38
  5.17.5 Renumbering 38
5.18 Importing Lignds/Monomers 38
5.19 Ligand from SMILES strings 38
5.20 Find Ligands 39
  5.20.1 Flexible Ligands 39

5.20.2 Adding Ligands to Model . . . 39
5.21 Flip Ligand . . . 40
5.22 Find Waters . . . 40
5.22.1 Refinement Failure . . . 40
5.22.2 Blobs . . . 41
5.23 Add Terminal Residue . . . 41
5.24 Add OXT Atom to Residue . . . 41
5.25 Add Atom at Pointer . . . 42
5.26 Place Helix . . . 42
5.27 Building Ideal DNA and RNA . . . 42
5.28 Merge Molecules . . . 42
5.29 Temperature Factor for New Atoms . . . 42
5.30 Applying NCS Edits . . . 43
5.31 Running Refmac . . . 43
5.32 Running SHELXL . . . 44
5.33 Clear Pending Picks . . . 44
5.34 Delete . . . 44
5.35 Sequence Assignment . . . 45
5.36 Building Links and Loops . . . 45
5.37 Fill Partial Residues . . . 45
5.38 Changing Chain IDs . . . 45
5.39 Setting Occupancies . . . 46
5.40 Fix Nomenclature Errors . . . 46
5.41 Rotamer Fix Whole Protein . . . 46
5.42 Refine All Waters . . . 46
5.43 Moving Molecules/Ligands . . . 46
5.44 Modifying the Labels on the Model/Fit/Refine dialog . . . 47

6 Map-Related Features . . . 48
6.1 Maps in General . . . 48
6.1.1 Map Reading Bug . . . 48
6.2 Create a Map . . . 48
6.2.1 Auto-read MTZ file . . . 48
6.2.2 Reading CIF data . . . 48
6.2.3 Reading PHS data . . . 49
6.3 Map Contouring . . . 49
6.4 Map Extent . . . 50
6.5 Map Contour “Scrolling” Limits . . . 50
6.6 Map Line Width . . . 50
6.7 “Dynamic” Map colouring . . . 50
6.8 Difference Map Colouring . . . 51
6.9 Make a Difference Map . . . 51
6.10 Make an Averaged Map . . . 51
6.11 Map Sampling . . . 51
6.12 Dragged Map . . . 51
6.13 Dynamic Map Sampling and Display Size . . . 51
6.14 Skeletonization . . . 52
6.15 Map Sharpening . . . 52

6.16 Pattersons . . . 52
6.17 Masks . . . 52
6.17.1 Example . . . 53
6.18 Trimming . . . 53
6.19 Map Transformation . . . 53
6.20 Export Map . . . 54

7 Validation . . . 55
7.1 Ramachandran Plots . . . 55
7.2 Geometry Analysis . . . 55
7.3 Chiral Volumes . . . 55
7.3.1 Fixing Chiral Volume Errors . . . 56
7.4 Blobs: a.k.a. Unmodelled density . . . 56
7.5 Difference Map Peaks . . . 56
7.6 Check Waters by Difference Map . . . 56
7.7 Molprobity Tools Interface . . . 57
7.8 GLN and ASN B-factor Outliers . . . 57
7.9 Validation Graphs . . . 58
7.9.1 Residue Density Fit . . . 58
7.9.2 Rotamer Analysis . . . 58
7.9.3 Temperature Factor Variance . . . 58
7.9.4 Peptide Omega Angle Distortion . . . 58

8 Representation . . . 59
8.1 Surfaces . . . 59

9 Hints and Usage Tips . . . 60
9.1 Documentation . . . 60
9.2 Low Resolution . . . 60
9.3 Coot Droppings . . . 60
9.4 Clearing Backups . . . 61
9.5 Getting out of “Translate” Mode . . . 61
9.6 Getting out of “Continuous Rotation” Mode . . . 61
9.7 Getting out of “Label Atom Only” Mode . . . 61
9.8 Button Labels . . . 61
9.9 Picking . . . 61
9.10 Resizing View . . . 62
9.11 Scroll-wheel . . . 62
9.12 Slow Computer Configuration . . . 62

10 Other Programs . . . 63
10.1 findligand . . . 63

11 Scripting Functions . . . 64

12 More Scripting Functions . . . 65

13 Scheme Scripting Functions **66**

Concept Index **67**

Function Index **70**

1 Introduction

This document is the Coot User Manual, giving an overview of the interactive features. Other documentation includes the Coot Reference Manual and the Coot Tutorial. These documents should be distributed with the source code.

1.1 Citing Coot and Friends

If have found this software to be useful, you are requested (if appropriate) to cite:

> "Features and Development of Coot" P Emsley, B Lohkamp, W Scott, and K Cowtan Acta Cryst. (2010). D66, 486-501 *Acta Crystallographica Section D-Biological Crystallography* **66**: 486-501

The reference for the REFMAC5 Dictionary is:

> REFMAC5 dictionary: "Organization of Prior Chemical Knowledge and Guidelines for its Use" Vagin AA, Steiner RA, Lebedev AA, Potterton L, McNicholas S Long F, Murshudov GN *Acta Crystallographica Section D-Biological Crystallography* **60**: 2184-2195 Part 12 Sp. Iss. 1 DEC 2004"

If using "SSM Superposition", please cite:

> "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions" Krissinel E, Henrick K *Acta Crystallographica Section D-Biological Crystallography* **60**: 2256-2268 Part 12 Sp. Iss. 1 DEC 2004

The reference for the the Electron Density Server is:

> GJ Kleywegt, MR Harris, JY Zou, TC Taylor, A Wählby, TA Jones (2004), "The Uppsala Electron-Density Server", *Acta Crystallographica Section D-Biological Crystallography* **60**, 2240-2249.
>
> Please also cite the primary literature for the received structures.

1.2 What is Coot?

Coot is a molecular graphics application. Its primary focus is crystallographic macromolecular model-building and manipulation rather than representation *i.e.* more like Frodo than Rasmol. Having said that, Coot can work with small molecule (SHELXL) and electron microscopy data, be used for homology modelling, make passably pretty pictures and display NMR structures.

Coot is Free Software. You can give it away. If you don't like the way it behaves, you can fix it yourself.

1.3 What Coot is Not

Coot is not:

- CCP4's official Molecular Graphics program[1]

---

[1] The official CCP4 graphics program (which contains parts of Coot (and Coot contains parts of CCP4MG)), CCP4MG is under the direct control of Liz Potterton and Stuart McNicholas.

- a program to do refinement[2]
- a protein crystallographic suite[3].

1.4 Hardware Requirements

The code is designed to be portable to any Unix-like operating system. Coot certainly runs on SGI IRIX64, RedHat Linux of various sorts, SuSe Linux[4] and MacOS X (10.2). The sgi Coot binaries should also work on IRIX.

If you want to port to some other operating system, you are welcome[5]. Note that your task will be eased by using GNU GCC to compile the programs components.

1.4.1 Mouse

Coot works best with a 3-button mouse and works better if it has a scroll-wheel too (see Chapter 2 for more details)[6].

1.5 Environment Variables

Coot responds to several environment variables that modify its behaviour.

- `COOT_STANDARD_RESIDUES` The filename of the pdb file containing the standard amino acid residues in "standard conformation"[7]
- `COOT_SCHEME_DIR` The directory containing standard (part of the distribution) scheme files
- `COOT_SCHEME_EXTRAS_DIR` A ':'-separated list of directories containing bespoke scheme files. This variable is not set by default. If you set it, Coot will test each ':'-separated string that it points to a directory, and if it does, Coot will load all the `.scm` files in that directory.
- `COOT_PYTHON_EXTRAS_DIR` A ':'-separated list of directories containing bespoke python files. This variable is not set by default. If you set it, Coot will test each ':'-separated string that it points to a directory, and if it does, Coot will load all the `.py` files in that directory.
- `COOT_REF_STRUCTS` The directory containing a set of high resolution pdb files used as reference structures to build backbone atoms from Cα positions
- `COOT_REF_SEC_STRUCTS` The directory containing a set of high-quality structures to be used as templates for fitting beta strands. If this is not set, then the directory COOT_REF_SEC_STRUCTS will be used to find the reference pdb files.
- `COOT_REFMAC_LIB_DIR` Refmac's CIF directory containing the monomers and link descriptions. In the future this may simply be the same directory in which refmac looks to find the library dictionary.

---

[2] although it does have a local refinement algorithm it is no substitute for refmac (a wrapper for refmac is available).

[3] that's the job of the CCP4 Program Suite.

[4] so far only 8.2 verified.

[5] it's Free Software after all and I could give you a hand.

[6] I can get by with a one button Macintosh - but it's not ideal.

[7] as it is known in Clipper.

- `COOT_SBASE_DIR` The directory to find the SBASE dictionary (often comes with CCP4).
- `COOT_RESOURCES_DIR` The directory that contains the splash screen image and the GTK+ application resources.
- `COOT_BACKUP_DIR` The directory to which backup are written (if it exists as a directory). If it is not, then backups are written to the current directory (the directory in which coot was started).

And of course extension language environment variables are used too:

- `PYTHONPATH` (for python modules)
- `GUILE_LOAD_PATH` (for guile modules)

Normally, these environment variables will be set correctly in the coot shell script.

1.6 Command Line Arguments

Rather that using the GUI to read in information, you can use the following command line arguments:

- `--c` *`cmd`* to run a command *`cmd`* on start up
- `--script` *`filename`* to run a script on start up (but see Section Section 3.10 [Scripting], page 11)
- `--no-state-script` don't run the `0-coot.state.scm` script on start up. Don't save a state script on exit either.
- `--pdb` *`filename`* for pdb/coordinates file
- `--coords` *`filename`* for SHELX .ins/.res and CIF files
- `--data` *`filename`* for mtz, phs or mmCIF data file
- `--auto` *`filename`* for auto-reading mtz files (mtz file has the default labels FWT, PHWT)
- `--map` *`filename`* for a map (currently CCP4-format only)
- `--dictionary` *`filename`* read in a cif monomer dictionary
- `--help` print command line options
- `--stereo` start up in hardware stereo mode
- `--version` print the version of coot and exit
- `--code` *`accession-code`* on starting Coot, get the pdb file and mtz file (if it exists) from the EDS
- `--no-guano` don't leave "Coot droppings" i.e. don't write state and history files on exit
- `--side-by-side` start in side-by-side stereo mode
- `--update-self` command-line mode to update the coot to the latest pre-release on the server
- `--python` an argument with no parameters - used to tell Coot that the -c arguments should be process as python (rather than as scheme).
- `--small-screen` start with smaller icons and font to fit on small screen displays
- `--zalman-stereo` start in Zalman stereo mode

So, for example, one might use:

- `coot --pdb post-refinement.pdb --auto refmac-2.mtz --dictionary lig.cif`

1.7 Web Page

Coot has a web page:

- `http://www.biop.ox.ac.uk/coot`

There you can read more about the CCP4 molecular graphics project in general and other projects which are important for Coot[8].

1.8 Crash

Coot might crash on you - it shouldn't.

Whenever Coot manipulates the model, it saves a backup pdb file. There are backup files in the directory `coot-backup`[9]. You can recover the session (until the last edit) by reading in the pdb file that you started with last time and then use `File -> Recover Session...`.

I would like to know about coot crashing[10] so that I can fix it as soon as possible. If you want your problem fixed, this involves some work on your part sadly.

First please make sure that you are using the most recent version of coot. I will often need to know as much as possible about what you did to cause the bug. If you can reproduce the bug and send me the files that are needed to cause it, I can almost certainly fix it[11] - especially if you use the debugger (gdb) and send a backtrace too[12]. Note that you may have to source the contents of `bin/coot` so that the libraries are can be found when the executable dynamically links.

---

[8] coot has several influences and dependencies, but these will not be discussed here in the User Manual.

[9] COOT_BACKUP_DIR is used in preference if set

[10] The map-reading problem (documented in Section Section 6.1 [Maps in General], page 48) is already known.

[11] now there's a hostage to fortune.

[12] to do so, please send me the output of the following: `$ gdb 'which coot-real' *corefile*` and then at the `(gdb)` prompt type: `where`, where *`corefile`* is the core dump file, '`core`' or '`core.4536`' or some such.

2 Mousing and Keyboarding

How do we move around and select things?

`Left-mouse Drag`
Rotate view

`Ctrl Left-Mouse Drag`
Translates view

`Shift Left-Mouse`
Label Atom

`Right-Mouse Drag`
Zoom in and out

`Ctrl Shift Right-Mouse Drag`
Rotate View around Screen Z axis

`Middle-mouse`
Centre on atom

`Scroll-wheel Forward`
Increase map contour level

`Scroll-wheel Backward`
Decrease map contour level

See also Chapter Chapter 9 [Hints and Usage Tips], page 60 for more help.

2.1 Next Residue

`‘‘Space’’`
Next Residue

`‘‘Shift’’ ‘‘Space’’`
Previous Residue

See also “Recentring View” (Section Section 3.14 [Recentring View], page 15).

2.2 Keyboard Contouring

Use `+` or `-` on the keyboard if you don’t have a scroll-wheel.

2.3 Mouse Z Translation and Clipping

Here we can change the clipping and Translate in Screen Z

`Ctrl Right-Mouse Drag Up/Down`
changes the slab (clipping planes)

`Ctrl Right-Mouse Drag Left/Right`
translates the view in screen Z

2.4 Keyboard Translation

`Keypad 3`	Push View (+Z translation)
`Keypad .`	Pull View (-Z translation)

2.5 Keyboard Zoom and Clip

`N`	Zoom out
`M`	Zoom in
`D`	Slim clip
`F`	Fatten clip

2.6 Scrollwheel

When there is no map, using the scroll-wheel has no effect. If there is exactly one map displayed, the scroll-wheel will change the contour level of that map. If there are two or more maps, the map for which the contour level is changed can be set using either `HID -> Scrollwheel -> Attach scroll-wheel to which map?` and selecting a map number or clicking the "Scroll" radio button for the map in the Display Manager.

You can turn off the map contour level changing by the scroll wheel using:

```
(set-scroll-by-wheel-mouse 0)
```

(the default is 1 [on]).

2.7 Selecting Atoms

Several Coot functions require the selecting of atoms to specify a residue range (for example: Regularize, Refine (Section Section 5.1 [Regularization and Real Space Refinement], page 27) or Rigid Body Fit Zone (Section Section 5.4 [Rigid Body Refinement], page 30)). Select atoms with the Left-mouse. See also Picking (Section Section 9.9 [sec_picking], page 61).

Use the scripting function `(quanta-buttons)` to make the mouse functions more like other molecular graphics programs to which you may be more accustomed[1].

2.8 Virtual Trackball

You may not completely like the way the molecule is moved by the mouse movement[2]. To change this, try: `HID -> Virtual Trackball -> Flat`. To do this from the scripting interface: `(vt-surface 1)`[3].

If you *do* want `screen-z rotation` screen-z rotation, you can either use Shift Right-Mouse Drag or set the Virtual Trackball to Spherical Surface mode and move the mouse along the bottom edge of the screen.

---

[1] See also Section 2.9 [more on zooming], page 7

[2] Mouse movement in “Spherical Surface” mode generates a component of (often undesirable) screen z-rotation, particularly noticeable when the mouse is at the edge of the screen.

[3] `(vt-surface 0)` to turn it back to “Spherical” mode.

2.9 More on Zooming

The function `(quanta-like-zoom)` adds the ability to zoom the view using just Shift `+` Mouse movement[4].

There is also a Zoom slider (`Draw -> Zoom`) for those without a right-mouse button.

---

[4] this is off by default because I find it annoying.

3 General Features

The map-fitting and model-building tools can be accessed by using `Calculate -> Model/Fit/Refine...`. Many functions have tooltips[1] describing the particular features and are documented in Chapter Chapter 5 [Modelling and Building], page 27.

`F5:`	posts the Model/Fit/Refine dialog
`F6:`	posts the Go To Atom Window
`F7:`	posts the Display Control Window

3.1 Version number

The version number of Coot can be found at the top of the "About" window (`Help -> About`).

This will return the version of coot:

```
$ coot --version
```

There is also a script function to return the version of coot:

```
(coot-version)
```

3.2 Antialiasing

The built-in antialiasing (for what it's worth) can be enabled using:

```
(set-do-anti-aliasing 1)
```

The default is `0` (off).

This can also be activated using `Edit Preferences -> Others -> Antialiasing -> Yes`.

If you have an nVidia graphics card, external antialiasing can be actived setting the environment variable `__GL_FSAA_MODE`. For me a setting of `5` works nicely and gives a better image than using Coot's built-in antialiasing.

Also for nVidia graphics card users, there is the application nvidia-settings:

`Antialiasing Setting -> Override Application Settings` and slide the slider to the right. On restarting Coot, it should be in antialias mode[2].

3.3 Molecule Number

Coot is based on the concept of molecules. Maps and coordinates are different representations of molecules. The access to the molecule is *via* the *molecule number*. It is often important therefore to know the molecule number of a particular molecule.

The Molecule Number of a molecule can be found by clicking on an atom of that molecule (if it has coordinates of course). The first number in brackets in the resulting text in the status bar and console is the Molecule Number. The Molecule Number can also be found in Display Control window (Section Section 3.7 [Display Manager], page 10). It is also displayed on the left-hand side of the molecule name in the option menus of the "Save Coordinates" and "Go To Atom" windows.

---

[1] Put your mouse over a widget for a couple of seconds, if that widget has a tooltip, it will pop-up in a yellow box (or a grey box for some reason if you are using Macintosh).

[2] that works for me, at least.

3.4 Display Issues

The “graphics” window is drawn using OpenGL. It is considerably smoother (i.e. more frames/sec) when using a 3D accelerated X server.

The view is orthographic (*i.e.* the back is the same size as the front). The default clipping is about right for viewing coordinate data, but is often a little too “thick” for viewing electron density. It is easily changed (see Section Section 3.16 [Clipping Manipulation], page 16).

Depth-cueing is linear and fixed on.

The graphics window can be resized, but it has a minimum size of 400x400 pixels.

3.4.1 Stereo

Hardware Stereo is an option for Coot (`Draw -> Stereo... -> Hardware Stereo -> OK`), side-by-side stereo is not an option.

The angle between the stereo pairs (the stereo separation) can be changed to suit your personal tastes using:

```
(set-hardware-stereo-angle-factor angle-factor)
```

where *`angle-factor`* would typically be between 1.0 and 2.0

3.4.2 Pick Cursor

When asked to pick a residue or atom, the cursor changes from the normal arrow shape to a "pick" cursor. Sometimes it is difficult to see the default pick cursor, so you can change it using the function

```
(set-pick-cursor-index i)
```

where *`i`* is an integer less than 256. The cursors can be viewed using an external X program:

```
xfd -fn cursor
```

3.4.3 Origin Marker

A yellow box called the “origin marker” marks the origin. It can be removed using:

```
(set-show-origin-marker 0)
```

Its state can be queried like this:

```
(show-origin-marker-state)
```

which returns an number (0 if it is not displayed, 1 if it is).

3.5 Screenshot

A simple screenshot (image dump) can be made using `Draw -> Screenshot -> Simple...`. Note that in side by side stereo mode you only get the left-hand image.

3.6 Raster3D output

Output suitable for use by Raster3D's "render" can be generated using the scripting function

```
(raster3d file-name)
```

where *`file-name`* is such as `"test.r3d"`[3].

There is a keyboard key to generate this file, run "render" and display the image: Function key F8.

You can also use the function

```
(render-image)
```

which will create a file '`coot.r3d`', from which "render" produces '`coot.png`'. This png file is displayed using ImageMagick's display program (by default). Use something like:

```
(set! coot-png-display-program "gqview")
```

to change that to different display program ("gqview" in this case).

```
(set! coot-png-display-program "open")
```

would use Preview (by default) on Macintosh.

To change the widths of the bonds and density "lines" use (for example):

```
(set-raster3d-bond-thickness 0.1)
```

and

```
(set-raster3d-density-thickness 0.01)
```

Similarly for bones:

```
(set-raster3d-bone-thickness 0.05)
```

To turn off the representations of the atoms (spheres):

```
(set-renderer-show-atoms 0)
```

3.7 Display Manager

This is also known as "Map and molecule (coordinates) display control". Here you can select which maps and molecules you can see and how they are drawn[4]. The "Display" and "Active" are toggle buttons, either depressed (active) or undepressed (inactive). The "Display" buttons control whether a molecule (or map) is drawn and the "Active" button controls if the molecule is clickable[5] (*i.e.* if the molecule's atoms can be labeled).

The "Scroll" radio buttons sets which map is has its contour level changed by scrolling the mouse scroll wheel.

By default, the path names of the files are not displayed in the Display Manager. To turn them on:

```
(set-show-paths-in-display-manager 1)
```

If you pull across the horizontal scrollbar in a Molecule view, you will see the "Render as" menu. You can use this to change between normal "Bonds (Colour by Atom)","Bonds (Colour by Chain)" and "Cα" representation There is also available "No Waters" and "Cα + ligands" representations.

---

[3] Povray support is only semi-working, there is a problem with the orientation of the image.

[4] to a limited extent.

[5] the substantial majority of the time you will want your the buttons to be both either depressed or undepressed, rarely one but not the other.

3.8 The Modelling Toolbar

You might not want to have the right-hand-side vertical toolbar that contains icons for some modelling operations[6] displayed:

```
(hide-modelling-toolbar)
```

to bring it back again:

```
(show-modelling-toolbar)
```

3.9 The file selector

3.9.1 File-name Filtering

The "Filter" button in the fileselection filters the filenames according to extension. For coordinates files the extensions are ".pdb" ".brk" ".mmcif" and others. For data: ".mtz", ".hkl", ".phs", ".cif" and for (CCP4) maps ".ext", ".msk" and ".map". If you want to add to the extensions, the following functions are available:

- `(add-coordinates-glob-extension` *`extension`*`)`
- `(add-data-glob-extension` *`extension`*`)`
- `(add-map-glob-extension` *`extension`*`)`
- `(add-dictionary-glob-extension` *`extension`*`)`

where *`extension`* is something like: `".mycif"`.

If you want the fileselection to be filtered without having to use the `"Filter"` button, use the scripting function

```
(set-filter-fileselection-filenames 1)
```

3.9.2 Filename Sorting

If you like your files initially sorted by date (rather than lexicographically, which is the default) use:

```
(set-sticky-sort-by-date)
```

3.9.3 Save Coordinates Directory

Some people prefer that the fileselection for saving coordinates starts in the original directory (rather than the directory from which they last imported coordinates). This option is for them:

```
(set-save-coordinates-in-original-directory 1)
```

3.10 Scripting

There is an compile-time option of adding a script interpreter. Currently the options are python and guile. It seems possible that in future you will be able to use both in the same executable. The binary distribution of Coot are linked with guile, others with python.

Hundreds of commands are made available for use in scripting by using SWIG, some of which are documented here. Other functions documented less well, but descriptions for them can be found at the end of this manual.

---

[6] British modelling, of course

Commands described throughout this manual (such as `(vt-surface 1)`) can be evaluated directly by Coot by using the “Scripting Window” (`Calculate -> Scripting...`). Note that you type the commands in the upper entry widget and the command gets echoed (in red) and the return value and any output is displayed in the text widget lower (green). The typed command should be terminated with a carriage return[7]. Files[8] can be evaluated (executed) using `Calculate -> Run Script...`.

Note that in scheme (the usual scripting language of Coot), the parentheses are important.

To execute a script file from the command line use the `--script` *`filename`* arguments (except when also using the command line argument `--no-graphics`, in which case you should use `-s` *`filename`*).

After you have used the scripting window, you may have noticed that you can no longer kill Coot by using Ctrl-C in the console. To recover this ability:

```
(exit)
```

in the scripting window.

3.10.1 Python

Coot has an (optional) embedded python interpreter. Thus the full power of python is available to you. Coot will look for an initialization script (`$HOME/.coot.py`) and will execute it if found. This file should contain python commands that set your personal preferences.

3.10.1.1 Python Commands

The scripting functions described in this manual are formatted suitable for use with guile, *i.e.*:

```
(function arg1 arg2...)
```

If you are using Python instead: the format needs to be changed to:

```
function(arg1,arg2...)
```

Note that dashes in guile function names become underscores for python, so that (for example) `(raster-screen-shot)` becomes `raster_screen_shot()`.

3.10.2 Scheme

The scheme interpreter is made available by embedding guile. The initialization script used by this interpreter is `$HOME/.coot`. This file should contain scheme commands that set your personal preferences.

3.10.3 Coot State

The “state” of Coot is saved on Exit and written to a file called `0-coot.state.scm` (scheme) `0-coot.state.py` (python). This state file contains information about the screen centre, the clipping, colour map rotation size, the symmetry radius, and other molecule related parameters such as filename, column labels, coordinate filename *etc.*.

---

[7] which causes the evaluation of the command.

[8] such as the Coot state file (Section Section 3.10.3 [Coot State], page 12).

Use `Calculate -> Run Script...` to use this file to re-create the loaded maps and models that you had when you finished using Coot[9] last time. A state file can be saved at any time using `(save-state)` which saves to file `0-coot.state.scm` or `(save-state-filename "thing.scm")` which saves to file `thing.scm`.

When Coot starts it can optionally run the commands in `0-coot.state.scm`.

Use `(set-run-state-file-status i)` to change the behaviour: `i` is `0` to never run this state file at startup, `i` is `1` to get a dialog option (this is the default) and `i` is `2` to run the commands without question.

3.10.4 Key Binding

"Power users" of Coot might like to write their own functions and bind that function to a keyboard key. How do they do that?

By using the add-key-binding function:

`(add-key-binding` *`function-name key function`*`)`

where *`key`* is a quoted string (note that upper case and lower case keys are distinguished - activate get upper case key binding you need to chord the shift key[10]).

for example:

`(add-key-binding "Refine Active Residue with Auto-accept" "x" refine-active-residue)`

Have a look at the key bindings section on the Coot wiki for several more examples.

3.10.5 User-Defined Functions

"Power users" of Coot might also like to write their own functions that occur after picking an atom (or a number of atoms)

`(user-defined-click` *`n_clicks udfunc`*`)`

define a function *`func`* which runs after the user has made *`n_clicked`* atom picks. *`func`* is called with a list of atom specifiers - the first member of which is the molecule number.

3.11 Backups and Undo

By default, each time a modification is made to a model, the old coordinates are written out[11]. The backups are kept in a backup directory and are tagged with the date and the history number (lower numbers are more ancient[12]). The "Undo" function discards the current molecule and loads itself from the most recent backup coordinates. Thus you do not have to remember to "Save Changes" - coot will do it for you[13].

If you have made changes to more than one molecule, Coot will pop-up a dialog box in which you should set the "Undo Molecule" *i.e.* the molecule to which the Undo operations will apply. Further Undo operations will continue to apply to this molecule until there are none left. If another Undo is requested Coot checks to see if there are other molecules that can be undone, if there is exactly one, then that molecule becomes the “Undo Molecule”, if there are more than one, then another Undo selection dialog will be displayed.

---

[9] in that particular directory.

[10] funny that

[11] this might be initially surprising since this could chew up a lot of disk space. However, disk space is cheap compared to losing you molecule.

[12] The coordinates are written in pdb format - that's OK, isn't it?.

[13] unless you tell it not to, of course - use (*e.g.*) `(turn-off-backup 0)` to turn off the backup (for molecule 0 in this case).

You can set the undo molecule using the scripting function:

`(set-undo-molecule` *`imol`*`)`

If for reasons of strange system[14] requirements you want to remove the path components of the backup file name you can do so using:

`(set-unpathed-backup-file-names 1)`

3.11.1 Redo

The “undone” modifications can be re-done using this button. This is not available immediately after a modification[15].

3.11.2 Restoring from Backup

There may be certain circumstances[16] in which you wish to restore from a backup but can’t get it by the “Undo” mechanism described above. In that case, start coot as normal and then open the (typically most recent) coordinates file in the directory `coot-backup` (or the directory pointed to the environment variable `COOT_BACKUP_DIR` if it was set) . This file should contain your most recent edits. In such a case, it is sensible for neatness purposes to immediately save the coordinates (probably to the current directory) so that you are not modifying a file in the backup directory.

See also Section Section 1.8 [Crash], page 4.

3.12 View Matrix

It is sometimes useful to use this to orient the view and export this orientation to other programs. The orientation matrix of the view can be displayed (in the console) using:

`(view-matrix)`

Also, the internal representation of the view can be returned and set using:

`(view-quaternion)` to return a 4-element list

`(set-view-quaternion i j k l)` which sets the view quaternion.

So the usage of these functions would be something like:

```
(let ((v (view-quaternion)))
   ;; manipulate v here, maybe
   (apply set-view-quaternion v))
```

---

[14] or system manager.

[15] It works like the “Forwards” buttons in a web browser - which is not available immediately after viewing a new page.

[16] for example, if coot crashes.

3.13 Space Group and Symmetry

Occasionally you may want to know the space group of a particular molecule. Interactively (for maps) you can see it using the Map Properties button in the Molecule Display Control dialog.

There is a scripting interface function that returns the space group for a given molecule[17]:

`(show-spacegroup imol)`

You can force a space group onto a molecule using the following:

`(set-space-group imol space-group)`

where *space-group* is one of the standard CCP4 space group names (*e.g.* `"P 21 21 21"`).

To show the symmetry operators of a particular molecule use: `(get-symmetry imol)` which will return a list of strings.

3.14 Recentring View

- Use Control + left-mouse to drag around the view
- or
- middle-mouse over an atom. In this case, you will often see “slide-recentring”, the graphics smoothly changes between the current centre and the newly selected centre.
- or
- Use `Draw -> Go To Atom...` to select an atom using the keyboard. Note that you can subsequently use “Space” in the “graphics” window (OpenGL canvas) to recentre on the next Cα.
- or
- To centre on an arbitrary position (x,y,z), use the scripting function `(set-rotation-centre x y z)`.
- or
- Use the keyboard: [Ctrl G] then key in a residue number and (optionally) a chainid and press Return

If you don’t want smooth recentring (sliding) `Edit -> Preferences -> Smooth Recentring -> Off`. You can also use this dialog to speed it up a bit (by decreasing the number of steps instead of turning it off).

3.15 Views

Coot has a views interface (you might call them ”scenes“) that define a particular orientation, zoom and view centre. Coot and linearly interpolate between the views. The animation play back speed can be set with the ”Views Play Speed“ menu item - default is a speed of 10.

The views interface can be found under the Extensions menu item.

---

[17] if no space group has been assigned it returns `‘‘No spacegroup for this molecule’’`

3.16 Clipping Manipulation

The clipping planes (a.k.a. “slab”) can be adjusted using `Edit -> Clipping` and adjusting the slider. There is only one parameter to change and it affects both the front and the back clipping planes[18]. The clipping can also be changed using keyboard “D” and “F”.

It can also be changed with Ctrl + Right-mouse drag up and down. Likewise the screen-Z can be changed with Ctrl + Right-mouse left and right[19].

One can “push” and “pull” the view in the screen-Z direction using keypad 3 and keypad “.” (see Section Section 2.4 [Keyboard Z Translation], page 6).

3.17 Background colour

The background colour can be set either using a GUI dialog (`Edit$ -> Background Colour`) or the function `(set-background-colour 0.00 0.00 0.00)`, where the arguments are 3 numbers between 0.0 and 1.0, which respectively represent the red, green and blue components of the background colour. The default is (0.0, 0.0, 0.0) (black).

3.18 Unit Cell

If coordinates have symmetry available then unit cells can be drawn for molecules (`Draw -> Cell & Symmetry -> Show Unit Cell?`).

3.19 Rotation Centre Pointer

There is a pink pointer at the centre of the screen that marks the rotation centre. The size of the pointer can be changed using `Edit -> Pink Pointer Size...` or using scripting commands: `(set-rotation-centre-size 0.3)`.

3.20 Orientation Axes

The green axes showing the orientation of the molecule are displayed by default. To remove them use the scripting function;

```
(set-draw-axes 0)
```

3.21 Pointer Distances

The Rotation Centre Pointer is sometimes called simply “Pointer”. One can find distances to the pointer from any active set of atoms using “Pointer Distances” (under Measures). If you move the Pointer (*e.g.* by centering on an atom) and want to update the distances to it, you have to toggle off and on the “Show Pointer Distances” on the Pointer Distances dialog.

3.22 Crosshairs

Crosshairs can be drawn at the centre of the screen, using either the C key[20] in graphics window or `Draw -> Crosshairs...`. The ticks are at 1.54Å, 2.7Å and 3.8Å.

---

[18] I find a clipping level of about 3.5 to 4 comfortable for viewing electron density maps - it is a little “thinner” than the default startup thickness.

[19] Inspired by PyMol? Yep. . . sure was!

[20] and C again to toggle them off.

3.23 3D Annotations

Positions in 3D space can be annotated with 3D text. The mechanism to do this can be found under Extensions -> Representations -> 3D Annotations. 3D Annotations can be saved to and loaded from a file.

3.24 Frame Rate

Sometimes, you might as yourself "how fast is the computer?"[21]. Using `Calculate -> Frames/Sec` you can see how fast the molecule is rotating, giving an indication of graphics performance. It is often better to use a map that is more realistic and stop the picture whizzing round. The output is written to the status bar and the console, you need to give it a few seconds to "settle down". It is best not to have other widgets overlaying the GL canvas as you do this.

The contouring elapsed time[22] gives an indication of CPU performance.

3.25 Program Output

Due to its "in development" nature (at the moment), Coot produces a lot of "console"[23] output - much of it debugging or "informational". This will go away in due course. You are advised to run Coot so that you can see the console and the graphics window at the same time, since feedback from atom clicking (for example) is often written there rather than displayed in the graphics window.

- Output that starts "ERROR..." is a programming problem (and ideally, you should never see it).
- Output that starts "WARNING..." means that something probably unintended happened due to the unexpected nature of your input or file(s).
- Output that starts "DEBUG..." has (obviously enough) been added to aid debugging. Most of them should have been cleaned up before release, but as Coot is constantly being developed, a few may slip through. Just ignore them.

---

[21] compared to some other one.

[22] prompted by changing the contour level.

[23] *i.e.* the terminal in which you started Coot.

4 Coordinate-Related Features

4.1 Reading coordinates

The format of coordinates that can be read by coot is either PDB or mmCIF. To read coordinates, choose `File -> Read Coordinates` from the menu-bar. Immediately after the coordinates have been read, the view is (by default) recentred to the centre of this new molecule and the molecule is displayed. The recentring of the view after the coordinates have been read can be turned off by unclicking the "Recentre?" radio-button.

To disable the recentring of the view on reading a coordinates file via scripting, use: `(set-recentre-on-read-pdb 0)`. However, when reading a coordinates file from a script it is just as good (if not better) to use `(handle-read-draw-molecule-with-recentre` ***`filename`*** `0)` - the additional `0` means “don’t recentre”. And that affects just the reading of ***`filename`*** and not subsequent files.

Note that as of version 0.6.2 Coot can read MDL mol/mol2 files (the atom names are not unique (of course), but at least you can see the coordinates).

4.1.1 A Note on Space Groups Names

Coot uses the space group on the “CRYST1” line of the pdb file. The space group should be one of the xHM symbols listed (for example) in the CCP4 dictionary file ‘`syminfo.lib`’. So, for example, `"R 3 2 :H"` should be used in preference to `"H32"`.

4.1.2 Read multiple coordinate files

The reading multiple files using the GUI is not available (at the moment). However the following scripting functions are available:

`(read-pdb-all)`

which reads all the “*.pdb” files in the current directory

`(multi-read-pdb` ***`glob-pattern dir`***`)`

which reads all the files matching ***`glob-pattern`*** in directory ***`dir`***. Typical usage of this might be:

`(multi-read-pdb "a*.pdb" ".")`

Alternatively you can specify the files to be opened on the command line when you start coot (see Section Section 1.6 [Command Line Arguments], page 3).

4.1.3 SHELX .ins/.res files

SHELX ".res" (and ".ins" of course) files can be read into Coot, either using the GUI `File -> Open Coordinates...` or by the scripting function:

`(read-shelx-ins-file` ***`file-name`***`)`

where ***`file-name`*** is quoted, such as `"thox.ins"`.

Although Coot should be able to read any SHELX ".res" file, it may currently have trouble displaying the bonds for centro-symmetric structures.

ShelxL atoms with negative `PART` numbers are given alternative configuration identifiers in lower case.

To write a SHELX ".ins" file:

`(write-shelx-ins-file imol file-name)`

where *imol* is the number of the molecule you wish to export.

This will be a rudimentary file if the coordinates were initially from a "PDB" file, but will contain substantial SHELX commands if the coordinates were initially generated from a SHELX `ins` file.

4.2 Atom Info

Information about about a particular atom is displayed in the text console when you click using middle-mouse. Information for all the atoms in a residue is available using `Info -> Residue Info...`.

The temperature factors and occupancy of the atoms in a residue can be set by using `Edit -> Residue Info...`.

4.3 Atom Labeling

Use Shift + left-mouse to label atom. Do the same to toggle off the label. The font size is changeable using `Edit -> Font Size...`. The newly centred atom is labelled by default. To turn this off use:

`(set-label-on-recentre-flag 0)`

Some people prefer to have atom labels that are shorter, without the slashes and residue name:

`(set-brief-atom-labels 1)`

To change the atom label colour, use:

`(set-font-colour 0.9 0.9 0.9)`

4.4 Atom Colouring

The atom colouring system in coot is unsophisticated. Typically, atoms are coloured by element: carbons are yellow, oxygens red, nitrogens blue, hydrogens white and everything else green (see Section Section 3.7 [Display Manager], page 10 for colour by chain). However, it is useful to be able to distinguish different molecules by colour, so by default coot rotates the colour map of the atoms (*i.e.* changes the H value in the HSV[1] colour system). The amount of the rotation depends on the molecule number and a user-settable parameter:

- `(set-colour-map-rotation-on-read-pdb 30)`.

The default value is 31°.

Also one is able to select only the Carbon atoms to change colour in this manner: `(set-colour-map-rotation-on-read-pdb-c-only-flag 1)`.

The colour map rotation can be set individually for each molecule by using the GUI: `Edit -> Bond Colours...`.

---

[1] Hue Saturation Value (Intensity).

4.5 Bond Parameters

The various bond parameters can be set using the GUI dialog `Draw -> Bond Parameters` or *via* scripting functions.

The represention style of the molecule that has the active residue (if any) can be changed using the scroll wheel with Ctrl and Shift.

4.5.1 Bond Thickness

The thickness (width) of bonds of individual molecules can be changed. This can be done via the `Bond Parameters` dialog or the scripting interface:

`(set-bond-thickness` ***`thickness imol`***`)`

where `imol` is the molecule number.

The default thickness is 3 pixels. The bond thickness also applies to the symmetry atoms of the molecule. The default bond thickness for new molecules can be set using:

`(set-default-bond-thickness` ***`thick`***`)`

where ***`thick`*** is an integer.

There is no means to change the bond thickness of a residue selection within a molecule.

4.5.2 Display Hydrogens

Initially, hydrogens are displayed. They can be undisplayed using

`(set-draw-hydrogens` ***`mol-no`*** `0)`[2]

where ***`mol-no`*** is the molecule number.

There is a GUI to control this too, under "Edit -> Bond Parameters".

4.5.3 NCS Ghosts Coordinates

It is occasionally useful when analysing non-crystallographically related molecules to have "images" of the other related molecules appear matched onto the current coordinates. It is important to understand that these ghosts are for displaying differences of NCS-related molecules by structure superposition, not displaying neighbouring NCS related molecules. As you read in coordinates in Coot, they are checked for NCS relationships and clicking on "Edit -> Bond Parameters -> Show NCS Ghosts" -> "Yes" -> "Apply" will create "ghost" copies of them over the reference chain[3].

Sometimes SSM does not provide a good (or even useful) matrix. In that case, we can specify the residue range ourselves and let the LSQ algorithm provide the matrix. A gui dialog for this operation can be found under `Extensions -> NCS -> NCS Ghosts by Residue Range...`.

The scripting function is used like this:

`(manual-ncs-ghosts` *`imol resno-start resno-end ncs-chain-ids`*`)`

Typical usage: `(manual-ncs-ghosts 0 1 10 (list "A" "B" "C"))`

note that in ***`ncs-chain-ids`***, the NCS master/reference chain-id goes first.

---

[2] they can be redisplayed using `(set-draw-hydrogens` *`mol-no`* `1)`.

[3] the reference chain is, by default, the first chain of that type in the coordinates file. The reference (master) chain can be changed using the NCS Ghosts Control dialog.

4.5.4 NCS Maps

Coot can use the relative transformations of the NCS-related molecules in a coordinates molecule to transform maps. Use `Calculate -> NCS Maps...` to do this (note the NCS maps only make sense in the region of the reference chain (see above).

Note also that the internal representation of the map is not transformed. If you try to export a NCS overlay map you will get an untransformed map. A transformed map only makes sense around a given point (and when using transformed maps in Coot, this reference point is changed on the fly, thus allowing map transformations on the fly). [This applies to NCS overlap maps, NCS averaged maps are transformed].

This will also create an NCS averaged map[4].

4.5.5 Using Strict NCS

Coot can use a set of strict NCS matrices to specify NCS which means that NCS-related molecules can appear like convention symmetry-related molecules.

`(add-strict-ncs-matrix` *`imol ncs-chain-id ncs-target-chain-id m11 m12 m13 m21 m22 m23 m31 m32 m33 t1 t2 t3`*`)`

where *`ncs-chain-id`* might be "B", "C" "D" (etc.) and *`ncs-target-chain-id`* is "A", i.e. the B, C, D molecules are NCS copies of the A chain.

for icosahedral symmetry the translation components *`t1`*, *`t2`*, *`t3`* will be 0.

You need to turn on symmetry for molecule *`imol`* and set the displayed symmetry object type to "Display Near Chains".

4.6 Download coordinates

Coot provides the possibility to download coordinates from an OCA[5]. (*e.g.* EBI) server[6] (`File -> Get PDB Using Code...`). A pop-up entry box is displayed into which you can type a PDB accession code. Coot will then connect to the web server and transfer the file. Coot blocks as it does this (which is not ideal) but on a semi-decent internet connection, it's not too bad. The downloaded coordinates are saved into a directory called '`coot-download`'.

It is also possible to download mmCIF data and generate a map. This currently requires a properly formatted database structure factors mmCIF file[7].

4.7 Get Coordinates and Map from EDS

Using this function we have the ability to download coordinates and view the map from structures in the Electron Density Server (EDS) at Uppsala University. This is a much more robust and faster way to see maps from deposited structures. This function can be found under the File menu item.

This feature was added with the assistance of Gerard Kleywegt. If you use the EDS, please cite GJ Kleywegt, MR Harris, JY Zou, TC Taylor, A Wählby & TA Jones (2004), "*The Uppsala Electron-Density Server*", *Acta Cryst.* **D**60, 2240-2249.

---

[4] that also only makes sense in the region of the reference chain.

[5] OCA is "goose" in Spanish (and Italian)

[6] the default is the Weizmann Institute - which for reasons I won't go into here is currently much faster than the EBI server.

[7] which (currently) only a fraction are.

4.8 Save Coordinates

On selecting from the menus `File -> Save Coordinates...` you are first presented with a list of molecules which have coordinates. As well as the molecule number, there is the molecule name - very frequently the name of the file that was read in to generate the coordinates in coot initially. However, this is only a *molecule* name and should not be confused with the filename to which the coordinates are saved. The coordinates *filename* can be selected using the `Select Filename...` button.

If your filename ends in `.cif`, `.mmcif` or `.mmCIF` then an mmCIF file will be written (not a "PDB" file).

4.9 Setting the Space Group

If for some reason, the pdb file that you read does not have a space group, or has the wrong space group, then you can set it using the following function:

`(set-space-group imol symbol)`

e.g.:

```
(set-space-group 0 "P 41 21 2")
```

4.10 Anisotropic Atoms

By default anisotropic atom information is not represented[8]. To turn them on, use `Draw -> Anisotropic Atoms -> Show Anisotropic Atoms? -> Yes`, or the command: `(set-show-aniso 1)`.

You cannot currently display thermal ellipsoids[9] for isotropic atoms.

4.11 Symmetry

Coordinates symmetry is "dynamic". Symmetry atoms can be labeled[10].

Every time you recentre, the symmetry coordinates are updated. The information shown contains the atom information and the symmetry operation number and translations needed to generate the atom in that position. To show the symmetry operator as a string (rather than a (1-based) index into the list of symmetry operators (as is the default)) Use `Draw -> Show Symmetry> -> Expanded Symmetry Atom Labels`. Coot generates symmetry related atoms by moving the current set close to the origin by a translation, performing the symmetry expansion around the origin and moving the the symmetry coordinates back by applying the inverse of the origin translation. The origin translation is also displayed in curly braces, e.g. "1 -1 0".

By default symmetry atoms are not displayed.

If you want coot to display symmetry coordinates without having to use the gui, add to your '`~/.coot`' the following:

```
(set-show-symmetry-master 1)
```

---

[8] using thermal ellipsoids

[9] in the case of isotropic atoms, ellipsoids are spherical, of course.

[10] symmetry labels are in pale blue/purple and also provide the symmetry operator number and the translations along the a, b and c axes.

The symmetry can be represented as Cαs. This along with representation of the molecule as Cαs (Section Section 3.7 [Display Manager], page 10) allow the production of a packing diagram.

4.11.1 Missing symmetry

Sometimes (rarely) coot misses symmetry-related molecules that should be displayed. In that case you need to expand the shift search (the default is 1):

```
(set-symmetry-shift-search-size 2)
```

This is a hack, until the symmetry search algorithm is improved.

4.12 Sequence View

The protein is represented by one letter codes and coloured according to secondary structure. These one letter codes are active - if you click on them, they will change the centre of the graphics window - in much the same way as clicking on a residue in the Ramachandran plot.

4.13 Print Sequence

The single letter code (of the *imol*th molecule) is written out to the console in FASTA format. Use can use this to cut and paste into other applications:

```
(print-sequence imol)
```

4.14 Environment Distances

Environment distances are turned on using `Info -> Environment Distances...`. Contacts to other residues are shown and to symmetry-related atoms if symmetry is being displayed. The contacts are coloured by atom type[11].

4.15 Distances and Angles

The distance between atoms can be found using `Info -> Distance`[12]. The result is displayed graphically, and written to the console.

4.16 Zero Occupancy Marker

Atoms of zero occupancy are marked with a grey spot. To turn off these markers, use:

```
(set-draw-zero-occ-markers 0)
```

Use an argument of 1 to turn them on.

---

[11] contacts involving two hydrogens or at least one carbon atom are yellow, denoting 'bump'. Hydrogen contacts display cut-off are based on the user-defined maximum distance, but shortened by 0.5Å per hydrogen atom.

[12] Use `Angle` for an angle, of course.

4.17 Atomic Dots

You can draw dots round arbitrary atom selections

`(dots` ***`imol atom-selection dot-density radius`***`)` The function returns a handle.

e.g. put a sphere of dots around all atoms of the 0th molecule (it might be a set of heavy atom coordinates) at the default dot density and radius:

```
(dots 0 "/1" "heavy-atom-sites" 1 1)
```

You can't change the colour of the dots.

There is no internal mechanism to change the radius according to atom type. With some cleverness you might be able to call this function several times and change the radius according to the atom selection.

There is a function to clear up the dots for a particular molecule ***`imol`*** and dots set identifier ***`dots-handle`***

`(clear-dots` ***`imol dots-handle`***`)`

There is a function to return how many dots sets there are for a particular molecule ***`imol`***:

`(n-dots-set` ***`imol`***`)`

4.18 Ball and Stick Representation

Fragments of the molecule can be rendered as a "ball and stick" molecule:

`(make-ball-and-stick` ***`imol atom-selection bond-thickness sphere-size draw-spheres-flag`***`)`

e.g. `(make-ball-and-stick 0 "/1/A/10-20" 0.3 0.4 1)`

The ball-and-stick representation can be cleared using:

`(clear-ball-and-stick` ***`imol`***`)`

4.19 Mean, Median Temperature Factors

Coot can be used to calculate the mean (average) and median temperatures factors:

`(average-temperature-factor` ***`imol`***`)`

`(median-temperature-factor` ***`imol`***`)`

-1 is returned if there was a problem[13].

4.20 Secondary Structure Matching (SSM)

The excellent SSM alogrithm[14] of Eugene Krissinel is available in Coot. The GUI interface is straight-forward and can be found under `Calculate -> SSM Superpose`. You can specify the specific chains that you wish to match using the `"Use Specific Chain"` check-button.

There is a scripting level function which gives even finer control:

`(superpose-with-atom-selection` ***`imol1 imol2 mmdb-atom-selection-string-1 mmdb-atom-selection-string-2 move-copy-flag`*** `)` the *`move-copy-flag`* should be 1 if you want to apply the transformation to a copy of *`imol2`* (rather than *`imol2`* itself). Otherwise, *`move-copy-flag`* should be 0.

---

[13] *e.g.* this molecule was a map or a closed molecule.

[14] the same one as in the CCP4 program SUPERPOSE

mmdb atom selection strings (Coordinate-IDs) are explained in detail in the mmdb manual.

Briefly, the string should be formed in this manner:

`/mdl/chn/seq(res).ic/atm[elm]:aloc`

e.g. `"/1/A/12-130/CA"`

4.21 Least-Squares Fitting

There is a simple GUI for this `Calculate -> LSQ Superpose...`

The scripting interface to LSQ fitting is as follows:

`(simple-lsq-match` *`ref-start-resno ref-end-resno ref-chain-id imol-ref mov-start-resno mov-end-resno mov-chain-id imol-mov match-type`*`)`

where:

- *`ref-start-resno`* is the starting residue number of the reference molecule
- *`ref-end-resno`* is the last residue number of the reference molecule
- *`mov-start-resno`* is the starting residue number of the moving molecule
- *`mov-end-resno`* is the last residue number of the moving molecule
- *`match-type`* is one of `'CA`, `'main`, or `'all`.

e.g.: `(simple-lsq-match 940 950 "A" 0 940 950 "A" 1 'main)`

More sophisticated (match molecule number 1 chain “B” on to molecule number 0 chain “A”):

```
(define match1 (list 840 850 "A" 440 450 "B" 'all))
(define match2 (list 940 950 "A" 540 550 "B" 'main))
(clear-lsq-matches)
(set-match-element match1)
(set-match-element match2)
(lsq-match 0 1) ; match molecule number 1 onto molecule number 0.
```

4.22 Ligand Overlaying

The scripting function

`(overlap-ligands` *`imol-ligand imol-ref chain-id-ref resno-ref`*`)`

returns a rotation+translation operator which can be applied to other molecules (and maps). Here, *`imol-ligand`* is the molecule number of the ligand (which is presumed to be a a molecule on its own - Coot simply takes the first residue that it finds). *`imol-ref chain-id-ref resno-ref`* collectively describe the target position for the moving *`imol-ligand`* molecule.

The convenience function

`(overlay-my-ligands` *`imol-mov chain-id-mov resno-mov imol-ref chain-id-ref resno-ref`*`)` wraps `overlap-ligands`.

The GUI for the function can be found under

`Extensions -> Modelling -> Supperpose Ligands...`

4.23 Writing PDB files

As well as the GUI option `File -> Save Coordinates...` there is a scripting options available:

`(write-pdb-file` *`imol pdb-file-name`*`)`

which writes the *`imol`*th coordinates molecule to *`filename`*.

To write a specific residue range:

`(write-residue-range-to-pdb-file` *`imol chain-id start-resno endresno pdb-file-name`*`)`

5 Modelling and Building

The functions described in this chapter manipulate, extend or build molecules and can be found under `Calculate -> Model/Fit/Refine...`. When activated, the dialog "stays on top" of the main graphics window[1]. Some people think that this is not always desirable, so this behaviour can be undone using:

```
(set-model-fit-refine-dialog-stays-on-top 0)
```

5.1 Regularization and Real Space Refinement

Coot will read the geometry restraints for refmac and use them in fragment (zone) idealization - this is called “Regularization”. The geometrical restraints are, by default, bonds, angles, planes and non-bonded contacts. You can additionally use torsion restraints by `Calculate -> Model/Fit/Refine... -> Refine/Regularize Control -> Use Torsion Restraints`. Truth to tell, this has not been successful in my hands (sadly).

“RS (Real Space) Refinement” (after Diamond, 1971[2]) in Coot is the use of the map in addition to geometry terms to improve the positions of the atoms. Select “Regularize” from the “Model/Fit/Refine” dialog and click on 2 atoms to define the zone (you can of course click on the same atom twice if you only want to regularize one residue). Coot then regularizes the residue range. At the end Coot, displays the intermediate atoms in white and also displays a dialog, in which you can accept or reject this regularization. In the console are displayed the χ^2 values of the various geometrical restraints for the zone before and after the regularization. Usually the χ^2 values are considerably decreased - structure idealization such as this should drive the χ^2 values toward zero.

The use of “Refinement” is similar - with the addition of using a map. The map used to refine the structure is set by using the “Refine/Regularize Control” dialog. If you have read/created only one map into Coot, then that map will be used (there is no need to set it explicitly).

Use, for example, `(set-matrix 20.0)`

to change the weight of the map gradients to geometric gradients. The higher the number the more weight that is given to the map terms[3]. The default is 60.0. This will be needed for maps generated from data not on (or close to) the absolute scale or maps that have been scaled (for example so that the sigma level has been scaled to 1.0).

For both “Regularize Zone” and “Refine Zone” one is able to use a single click to refine a residue range. Pressing A on the keyboard while selecting an atom in a residue will automatically create a residue range with that residue in the middle. By default the zone is extended one residue either size of the central residue. This can be changed to 2 either side using `(set-refine-auto-range-step 2)`.

Intermediate (white) atoms can be moved around with the mouse (click and drag with left-mouse, by default). Refinement will proceed from the new atom positions when the mouse button is released. It is possible to create incorrect atom nomenclature and/or chiral volumes in this manner - so some care must be taken. Press the A key as you left-mouse click to move atoms more "locally" (rather than a linear shear) and CTRL key as you left-mouse click to move just one atom.

---

[1] given a half-decent window manager

[2] Diamond, R. (1971). A Real-Space Refinement Procedure for Proteins. *Acta Crystallographica* **A**27, 436-452.

[3] but the resulting χ^2 values are higher.

In more up to date versions, Coot will display colour patches (something like a traffic light system) representing the chi squared values of each of types of geometric feature refined. Typically "5 greens" is the thing to aim for, the colour changes occurring at chi squared values 2, 5 and 8 (8 being the most red).

To prevent the unintentional refinement of a large number of residues, there is a "heuristic fencepost" of 20 residues. A selection of than 20 residues will not be regularized or refined. The limit can be changed using the scripting function: *e.g.* `(set-refine-max-residues 30)`.

5.1.1 Dictionary

The geometry description for residues, monomers and links used by Coot are in the standard mmCIF format. Because this format alows multiple comp_ids (residue types) to be described within a cif loop, it is hard to tell when a dictionary entry needs to be overwritten when reading a new file. Therefore Coot makes this extra constraint: that the "chem_comp" loop should appear first in the comp list data item - if this is the case, then Coot can overwrite an old restraint table for a particular comp_id/residue-type when a new one is read.

By default, the geometry dictionary entries for only the standard residues are read in at the start[4]. It may be that your particular ligand is not amongst these. To interactively add a dictionary entry use `File -> Import CIF Dictionary`. Alternatively, you can use the function:

`(read-cif-dictionary` *`filename`*`)`

and add this to your `.coot` file (this may be the preferred method if you want to read the file on more than one occasion).

Note: the dictionary also provides the description of the ligand's torsions.

5.1.2 Sphere Refinement

Sphere refinement selects residues within a certain distance of the residue at the centre of the screen and includes them for real space refinement. In this way, one can select residues that are not in a linear range. This technique is useful for refining disulfide bonds and glycosidic linkages.

To enable sphere refinement, Right-mouse in the vertical toolbutton menu, Manage buttons -> [Tick] Sphere Refine -> Apply. You will need a python-enabled Coot to do this.

The following adds a key binding (Shift-R) that refines resides that are within 3.5Å of the residue at the centre of the screen:

```
(define *sphere-refine-radius* 3.5)

(add-key-binding "Refine residues in a sphere" "R"
    (lambda ()
     (using-active-atom
```

---

[4] And a few extras, such as phosphate

```
    (let* ((rc-spec (list aa-chain-id aa-res-no aa-ins-code))
           (ls (residues-near-residue aa-imol rc-spec *sphere-refine-radius*)))
      (refine-residues aa-imol (cons rc-spec ls))))))
```

5.1.3 Refining Specific Residues

You can specify the residues that you want to refine without using a linear or sphere selection usine `refine-residues`. For example:

```
(refine-residues 0 ’(("L" 501 "") ("L" 503 "")))
```

will refine residues A501 and A503 (and residue A502 (if it exists) will be an anchoring residue - used in optimizing the link geometry of the atoms in A501 and A503).

5.1.4 Refining Carbohydrates

Refining carbohydrates monomers should be as straightforward as refining a protein residue. Coot will look in the dictionary for the 3-letter code for the particular residue type, if it does not find it, Coot will try to search for dictionary files using “-b-D” or “-a-L” extensions.

When refining a group of carbohydrates, the situation needs a bit more explanation. For each residue pair with tandem residue numbers specified in the refinement range selection, Coot checks if these residue types are are furanose or pyranose in the dictionary, and if the are both one or the other, then it tries to see if there are any of the 11 link types (BETA1-4, BETA2-3, ALPHA1-2 and so on) specified in the dictionary. It does this by a distance check of the potentially bonding atoms. If the distance is less than 3.0Å, then a glycosidic bond is made and used in the refinement.

Bonds between protein and carbohydrate and branched carbohydrates can be refined using “Sphere Refinement”.

Instead of using a sphere to make a residue selection, you can specify the residues directly using `refine-residues`, for example:

```
(refine-residues 0 ’(("L" 501 "") ("L" 503 "")))
```

LINK and LINKR cards are not yet used to determine the geometry of the restraints.

5.1.5 Planar Peptide Restraints

By default, Coot uses a 5 atom (CA-1, C-1, O-1, N-2, CA-2) planar peptide restraints. These restraints should help in low resolution fitting (the main-chains becomes less distorted), reduce accidental cis-peptides and may help “clean up” Ramachandran plots.

```
(add-planar-peptide-restraints)
```

And similarly they can be removed:

```
(remove-planar-peptide-restraints)
```

There is also a GUI to add and remove these restraints in `Extensions -> Refine... -> Peptide Restraints...`

5.1.6 The UNK residue type

The `UNK` residue type is a special residue type to Coot. It has been added for use with Buccaneer. Don't give you ligand (or anything else) the 3-letter-code `UNK` or confusion will result[5].

5.1.7 Moving Zero Occupancy Atoms

By default, atoms with zero occupancy are moved when refining and regularizing. This can sometimes be inconvenient. To turn of the movement of atoms with zero occupancy when refining and regularizing:

```
(set-refinement-move-atoms-with-zero-occupancy 0)
```

5.2 Changing the Map for Building/Refinement

You can change the map that is used for the fitting and refinement tools using the `Select Map...` button on the Model/Fit/Refine dialog.

5.3 Rotate/Translate Zone

“Rotate/Translate Zone” from the “Model/Fit/Refine” menu allows manual movement of a zone. After pressing the “Rotate/Translate Zone” button, select two atoms in the graphics canvas to define a residue range[6], the second atom that you click will be the local rotation centre for the zone. The atoms selected in the moving fragment have the same alternate conformation code as the first atom you click. To actuate a transformation, click and drag horizontally across the relevant button in the newly-created “Rotation & Translation” dialog. The axis system of the rotations and translations are the screen coordinates. Alternatively[7], you can click using left-mouse on an atom in the fragment and drag the fragment around. Use Control Left-mouse to move just one atom, rather than the whole fragment. If you click Control Left-mouse whilst *not* over an atom then you can rotate the fragment using mouse drag. Click “OK” (or press Return) when the transformation is complete.

To change the rotation point to the centre of the intermediate atoms (rather than the second clicked atom), use the setting:

```
(set-rotate-translate-zone-rotates-about-zone-centre 1)
```

5.4 Rigid Body Refinement

“Rigid Body Fit Zone” from the “Model/Fit/Refine” dialog provides rigid body refinement. The selection is zone-based[8]. So to refine just one residue, click on one atom twice.

Sometimes no results are displayed after Rigid Body Fit Zone. This is because the final model positions had too many final atom positions in negative density. If you want to over-rule the default fraction of atoms in the zone that have an acceptable fit (0.75), to be (say) 0.25:

```
(set-rigid-body-fit-acceptable-fit-fraction 0.25)
```

---

[5] unless you are using Buccaneer, of course

[6] if you want to move only one residue, then click the same atom twice.

[7] like Refinement and Regularization

[8] like Regularization and Refinement.

5.5 Simplex Refinement

Rigid body refinement via Nelder-Mead Simplex minimization is available in Coot. Simplex refinement has a larger radius of convergence and thus is useful in a position where simple rigid body refinement finds the wrong minimum. However the Simplex algorithm is much slower. Simplex refinement for a residue range *`start-resno`* to *`end-resno`* (inclusive) in chain *`chain-id`* can be accessed as follows:

`(fit-residue-range-to-map-by-simplex` *`start-resno end-resno alt-loc chain-id imol imol-for-map`*`)`

There is currently no GUI interface to Simplex refinement.

5.6 Post-manipulation-hook

If you wanted automatically run a function after a model has been manipulated then you can do so using by creating a function that takes 2 arguments, such as:

`(post-manipulation-hook` *`imol manipulation-mode`*`)`

`manipulation-mode` is one of *`(DELETED)`*, *`(MUTATED)`* or *`(MOVINGATOMS)`*.

And of course *`imol`* is the model number of the maniplated molecule.

(It would of course be far more useful if this function was also passed a list of residues - that is something for the future).

5.7 Baton Building

Baton build is most useful if a skeleton is already calculated and displayed (see Section Section 6.14 [Skeletonization], page 52). When three or more atoms have been built in a chain, Coot will use a prior probability distribution for the next position based on the position of the previous three. The analysis is similar to that of Oldfield & Hubbard (1994)[9], however it is based on a more recent and considerably larger database.

Little crosses are drawn representing directions in which is is possible that the chain goes, and a baton is drawn from the current point to one of these new positions. If you don't like this particular direction[10], use `Try Another`. The list of directions is scored according to the above criterion and sorted so that the most likely is at the top of the list and displayed first as the baton direction.

When starting baton building, be sure to be about 3.8Å from the position of the first-placed Cα, this is because the next Cα is placed at the end of the baton, the baton root being at the centre of the screen. So, when trying to baton-build a chain starting at residue 1, centre the screen at about the position of residue 2.

It seems like a good idea to increase the map sampling to 2 or even 2.5 (before reading in your mtz file) [a grid sampling of about 0.5Å seems reasonable] when trying to baton-build a low resolution map. You can set the map sampling using `Edit -> Map Parameters -> Map Sampling`.

---

[9] T. J. Oldfield & R. E. Hubbard (1994). "Analysis of Cα Geometry in Protein Structures" *Proteins-Structure Function and Genetics* **18(4)** 324 – 337.

[10] which is quite likely at first since coot has no knowledge of where the chain has been and cannot score according to geometric criteria.

Occasionally, every point is not where you want to position the next atom. In that case you can either shorten or lengthen the baton, or position it yourself using the mouse. Use “b” on the keyboard to swap to baton mode for the mouse[11].

Baton-built atoms are placed into a molecule called “Baton Atom” and it is often sensible to save the coordinates of this molecule before quitting coot.

If you try to trace a high resolution map (1.5Å or better) you will need to increase the skeleton search depth from the default (10), for example:

```
(set-max-skeleton-search-depth 20)
```

Alternatively, you could generate a new map using data to a more moderate resolution (2Å), the map may be easier to interpret at that resolution anyhow[12].

The guide positions are updated every time the “Accept” button is clicked. The molecule name for these atoms is “Baton Build Guide Points” and is is not usually necessary to keep them.

5.7.1 Undo

There is also an “Undo” button for baton-building. Pressing this will delete the most recently placed Cα and the guide points will be recalculated for the previous position. The number of “Undo”s is unlimited. Note that you should use the “Undo” button in the Baton Build dialog, not the one in the “Model/Fit/Refine” dialog (Section Section 3.11 [Backups and Undo], page 13).

5.7.2 Missing Skeleton

Sometimes (especially at loops) you can see the direction in which the chain should go, but there is no skeleton (see Section Section 6.14 [Skeletonization], page 52) is displayed (and consequently no guide points) in that direction. In that case, “Undo” the previous atom and decrease the skeletonization level (`Edit -> Skeleton Parameters -> Skeletonization Level`). Accept the atom (in the same place as last time) and now when the new guide points are displayed, there should be an option to build in a new direction.

5.7.3 Building Backwards

The following scenario is not uncommon: you find a nice stretch of density and start baton building in it. After a while you come to a point where you stop (dismissing the baton build dialog). You want to go back to where you started and build the other way. How do you do that?

- Use the command:

  ```
  (set-baton-build-params start-resno chain-id "backwards")
  ```

  where ***start-resno*** would typically be 0[13] and ***chain-id*** would be `""` (default).
- Recentre the graphics window on the first atom of the just-build fragment
- Select “Ca Baton Mode” and select a baton direction that goes in the “opposite” direction to what is typically residue 2. This is slightly awkward because the initial

---

[11] “b” again toggles the mode off.

[12] high-resolution map interpretation is planned.

[13] *i.e.* one less than the starting residue in the forward direction (defaults to 1).

baton atoms build in the "opposite" direction are not dependent on the first few atoms of the previously build fragment.

5.8 Reversing Direction of Fragment

After you've build a fragment, sometimes you might want to change the direction of that fragment (this function changes an already existing fragment, as opposed to Backwards Building which sets up Baton Building to place new points in reverse order).

The fragment is defined as a contiguous set of residues numbers. So that you should be sure that other partial fragments which have the same chain id and that are not connected to this fragment have residue numbers that are not contiguous with the fragment you are trying to reverse.

5.9 C\alpha -> Mainchain

Mainchain can be generated using a set of Cαs as guide-points (such as those from Baton-building) along the line of Esnouf[14] or Jones and coworkers[15]. Briefly, 6-residue fragments of are generated from a list of high-quality[16] structures. The Cα atoms of these fragments are matched against overlapping sets of the guide-point Cαs. The resulting matches are merged to provide positions for the mainchain (and Cβ) atoms. This procedure works well for helices and strands, but less well[17] for less common structural features.

This function is also available from the scripting interface:

`(db-mainchain imol chain-id resno-start resno-end direction)`

where direction is either `"backwards"` or `"forwards"`.

Recall that the `chain-id` needs to be quoted, *i.e.* use `"A"` not `A`. Note that `chain-id` is `""` when the Cαs have been built with Baton Mode in Coot.

5.10 Backbone Torsion Angles

It is possible to edit the backbone ϕ and ψ angles indirectly using an option in the Model/Fit/Refine's dialog: "Edit Backbone Torsions..". When clicked and an atom of a peptide is selected, this produces a new dialog that offers "Rotate Peptide" which changes this residues ψ and "Rotate Carbonyl" which changes ϕ. Click and drag across the button[18] to rotate the moving atoms in the graphics window. You should know, of course, that making these modifications alter the ϕ/ψ angles of more than one residue.

5.11 Docking Sidechains

Docking sidechains means adding sidechains to a model or fragment that has currently only poly-Ala, where the sequence assignment is unknown. The algorithm is basically the same as in Cowtan's Buccaneer, but with some corners cut to make things (more or less) interactive. The algorithm uses the shape of the density around the C-beta position to estimate the probability of each sidechain type at that position.

---

[14] R. M. Esnouf "Polyalanine Reconstruction from Cα Positions Using the Program *CALPHA* Can Aid Initial Phasing of Data by Molecular Replacement Procedures" *Acta Cryst.* , D**53**, 666-672 (1997).

[15] T.A. Jones & S. Thirup "Using known substructures in protein model building and crystallography" *EMBO J.* **5**, 819–822 (1986).

[16] and high resolution

[17] *i.e.* there are severely misplaced atoms

[18] as for Rotate/Translate Zone (Section Section 5.3 [Rotate/Translate Zone], page 30).

The function is accessed via the `Extensions -> Dock Sequence` menu item. First, a sequence should be assigned from a PIR file to a particular chain-id and model number. Secondly `Extensions -> Dock Sequence -> Dock Sequence on this fragment....` Choose the model to build on and then `Dock Sequence!` If all goes well, the model will be updated with mutated residues and undergo rotamer seach for each of the new residues. If the sequence alignment is not sufficiently clear, then you will get a dialog suggesting that you extend or improve the fragment.

5.12 Rotamers

The rotamers are generated[19] from the backbone independent sidechain library of the Richardsons group[20].

The m, t and p stand for "minus (-60)", "trans (180)" and "plus (+60)". There is one letter per χ angle.

Use keyboard . and , to cycle round the rotamers.

5.12.1 Auto Fit Rotamer

"Auto Fit Rotamer" will try to fit the rotamer to the electron density. Each rotamer is generated, rigid body refined and scored according to the fit to the map. Fitting the second conformation of a dual conformation in this way will often fail - the algorithm will pick the best fit to the density - ignoring the position of the other atoms.

The algorithm doesn't know if the other atoms in the structure are in sensible positions. If they are, then it is sensible not to put this residue too close to them, if they are not then there should be no restriction from the other atoms as to the position of this residue - the default is "are sensible", which means that the algorithm is prevented from finding solutions that are too close to the atoms of other residues. `(set-rotamer-check-clashes 0)` will stop this.

There is a scripting interface to auto-fitting rotamers:

`(auto-fit-best-rotamer` ***`resno alt-loc ins-code chain-id imol-coords imol-map clash-flag lowest-rotamer-probability`*** `)`

where:

`resno` is the residue number

`alt-loc` is the alternate/alternative location symbol (*e.g.* `"A"` or `"B"`, but most often `""`)

`ins-code` is the insertion code (usually `""`)

`imol-coords` is the molecule number of the coordinates molecule

`imol-map` is the molecule number of the map to which you wish to fit the side chains

---

[19] since version 0.4

[20] SC Lovell, JM Word, JS Richardson and DC Richardson (2000) "The Penultimate Rotamer Library" Proteins: Structure Function and Genetics 40: 389-408. You can get the paper from `http://kinemage.biochem.duke.edu/databases/rotamer.php`

`clash-flag` should the positions of other residues be included in the scoring of the rotamers (*i.e.* clashing with other other atoms gets marked as bad/unlikely)

`lowest-rotamer-probability`: some rotamers of some side chains are so unlikely that they shouldn't be considered - typically 0.01 (1%).

You can change the auto-fit rotamer fitting algorithms using

`(set-rotamer-search-mode mode)`

where *`mode`* is one of `(ROTAMERSEARCHAUTOMATIC)`, `(ROTAMERSEARCHLOWRES)` (*i.e.* "Backrub Rotamers" (*vide infra*)) or `(ROTAMERSEARCHHIGHRES)` (the conventional/high-resolution method using rigid-body fitting).

By default, the auto-fit rotamer method is `(ROTAMERSEARCHAUTOMATIC)`.

5.12.1.1 Backrub Rotamers

By default, Auto Fit Rotamer will switch to "Backrub Rotamer"[21] mode when fitting against a map of worse than 2.7Å. This search mode moves the some atoms of the mainchain of the neighbouring residues. After rotation of the central residue and neighbouring atoms around the "backrub vector", the individual peptides are back-rotated (along the peptide axis) so that the carbonyl oxygen are placed as near as possible to their original position. The Ramachandran plot is not used in this fitting algorithm.

5.12.2 De-clashing residues

Sometimes you don't have a map[22] but nevertheless there are clashing residues [23] (for example after mutation of a residue range) and you need to rotate side-chains to a non-clashing rotamer. There is a scripting interface:

`(de-clash imol chain-id start-resno end-resno)`

`start-resno` is the residue number of the first residue you wish to de-clash

`end-resno` is the residue number of the last residue you wish to de-clash

`imol` is the molecule number of the coordinates molecule

This interface will not change residues with insertion codes or alternate conformation. The *`lowest-rotamer-probability`* is set to 0.01.

5.13 Editing chi Angles

Instead of using Rotamers, one can instead change the χ angles (often called "torsions") "by hand" (using "Edit Chi Angles" from the "Model/Fit/Refine" dialog). To edit a residue's χ_1 press "1": to edit χ_2, "2": χ_3 "3" and χ_4 "4". Use left-mouse click and drag to change the χ value. Use keyboard "0"[24] to go back to ordinary view mode at any time during the editing. Alternatively, one can use the "View Rotation Mode" or use the CTRL key when moving the mouse in the graphics window. Use the Accept/Reject dialog when you have finished editing the χ angles.

---

[21] "The Backrub Motion: How Protein Backbone Shrugs When a Sidechain Dances" *Structure*, Volume **14**, Issue 2, Pages 265-274 I. Davis, W. Bryan Arendall III, D. Richardson, J. Richardson

[22] for example, in preparation of a model for molecular replacement

[23] atoms of residues that are too close to each other

[24] that's "zero".

For non-standard residues, the clicked atom defines the base of the atom tree, which defines the “head” of the molecule (it’s the “tail” (twigs/leaves) that wags). To emphasise, then: it matters on which atom you click!

By default torsions for hydrogen atoms are turned off. To turn them on:

```
(set-find-hydrogen-torsions 1)
```

To edit the rotatable bonds of a ligand using this tool, you will need to have read in the mmCIF dictionary beforehand.

5.14 Torsion General

You need to click on the torsion-general button, then click 4 atoms that describe the torsion - the first atom will be the base (non moving) part of the atom tree, on clicking the 4th atom a dialog will pop up with a "Reverse" button. Move this dialog out of the way and then left mouse click and drag in the main window will rotate the "top" part of the residue round the clicked atoms 2 and 3. When you are happy, click "Accept".

If you are torsion generaling a residue that has an alt conf, then the atoms of residue that are moved are those that have the same alt conf as the 4th clicked atom (or have an blank alt conf).

5.14.1 Ligand Torsion angles

For ligands, you will need to read the mmCIF file that contains a description of the ligand’s geometry (see Section Section 5.1 [Regularization and Real Space Refinement], page 27). By default, torsions that move hydrogens are not included. Only 9 torsion angles are available from the keyboard torsion angle selection.

5.15 Pep-flip

Coot uses the same pepflip scheme as is used in `O` (*i.e.* the C, N and O atoms are rotated 180^o round a line joining the Cα atoms of the residues involved in the peptide). Flip the peptide again to return the atoms to their previous position.

5.16 Add Alternate Conformation

The allows the addition alternate (dual, triple *etc.*) conformations to the picked residue. By default, this provides a choice of rotamer (Section Section 5.12 [Rotamers], page 34). If there are not the correct main chain atoms a rotamer choice cannot be provided, and Coot falls back to providing intermediate atoms.

The default occupancy for new atoms is 0.5. This can be changed by using use slider on the rotamer selection window or by using the scripting function:

```
(set-add-alt-conf-new-atoms-occupancy 0.4)
```

The remaining occupancy of the atoms (after the new occupancy has been added) is split amongst the atoms that existed in the residue before the split. It is important therefore that the residues atoms have sane occupancies before adding an alternative conformation.

The default Split Type is to split the whole residue. If you want the default to be to split a residue after (and including) the CA, then add to your ‘`.coot`’ file:

```
(set-add-alt-conf-split-type-number 0)
```

5.17 Mutation

Mutations are available on a 1-by-1 basis using the graphics. After selecting “Mutate…” from the “Model/Fit/Refine” dialog, click on an atom in the graphics. A “Residue Type” window will now appear. Select the new residue type you wish and the residue in the graphics is updated to the new residue type[25]. The initial position of the new rotamer is the *a priori* most likely rotamer. Note that in interactive mode, such as this, a residue type match[26] will not stop the mutation action occurring.

5.17.1 Mutating DNA/RNA

Mutation of DNA or RNA can be performed using “Simple Mutate” from the Model/Fit/Refine dialog. Residues need to be named "Ad", "Gr", "Ur" etc.

5.17.2 Multiple mutations

This dialog can be found under `Calculate -> Mutate Residue Range`. A residue range can be assigned a sequence and optionally fitted to the map. This is useful converting a poly-ALA model to the correct sequence[27].

Multiple mutations are also supported *via* the scripting interface. Unlike the single residue mutation function, a residue type match *will* prevent a modification of the residue[28]. Two functions are provided: To mutate a whole chain, use `(mutate-chain` ***`imol chain-id sequence`***`)` where:

`chain-id` is the chain identifier of the chain that you wish to mutate (*e.g.* `"A"`) and

`imol` is molecule number.

`sequence` is a list of single-letter residue codes, such as `"GYRESDF"` (this should be a straight string with no additional spaces or carriage returns).

Note that the number of residues in the sequence chain and those in the chain of the protein must match exactly (*i.e.* the whole of the chain is mutated (except residues that have a matching residue type).)

To mutate a residue range, use

- `(mutate-residue-range` ***`imol chain-id start-res-no stop-res-no sequence`***`)`

where

`start-res-no` is the starting residue for mutation

`stop-res-no` is the last residue for mutation, *i.e.* using values of 2 and 3 for ***`start-res-no`*** and ***`stop-res-no`*** respectively will mutate 2 residues.

Again, the length of the sequence must correspond to the residue range length. Note also that this is a protein sequence - not nucleic acid.

For mutation of nucleic acids, use:

`(mutate-nucleotide-range` ***`imol chain-id resno-start resno-end sequence`***`)`

---

[25] Note that selecting a residue type that matches the residue in the graphics will also result in a mutation

[26] *i.e.* the current residue type matches the residue type to which you wish to mutate the residue

[27] *e.g.* after using Ca -> Mainchain.

[28] *i.e.* the residue atoms will remain untouched

5.17.3 Mutating to a Non-Standard Residue

Sometimes one might like to model post-translational or other such modifications. How is that done, if the new residue type is not one of the standard residue types?

There is a scripting function:

`(mutate-by-overlap` *`imol chain-id resno new-three-letter-code`*`)`

This imports a model residue for the new residue type and overlays it on to the given residue by using graph-matching to determine the equivalent atoms.

The GUI for this can be found under `Extensions -> Modelling -> Replace Residue...` (for this to work, you need to be centred on the residue you wish to replace).

Note that if you are replacing are conventional protein residue with a modified form (*e.g.* replacing a TYR with a phoso-tyrosine or a LYS with an acetyl-lysine) you will need to make sure that the group of the resulting restraints is an `L-peptide` (use `Edit -> Restraints` to check and modify the restraints group. Likewise for modified RNA/DNA nucleotides, you need to specify the group as `RNA` or `DNA` as appropriate.

5.17.4 Mutate and Autofit

The function combines Mutation and Auto Fit Rotamer and is the easiest way to make a mutation and then fit to the map. You can currently only “Mutate and Autofit” protein residues (*i.e.* things with a rotamer dictionary.

5.17.5 Renumbering

Renumbering is straightforward using the renumber dialog available under `Calculate -> Renumber Residue Range...`. There is also a scripting interface:

`(renumber-residue-range` *`imol chain-id start-res-no last-resno offset`*`)`

5.18 Importing Lignds/Monomers

You can import monomers (often ligands) using `File -> Get Monomer...`[29] by providing the 3-letter code of your monomer/ligand. The resulting molecule will be moved so that it placed at the current screen centre.

Typically, when you are happy about the placement of the ligand, you'd then use `Merge Molecules` to add the ligand/monomer to the main set of coordinates.

This procedure creates a pdb file ‘`monomer-XXX.pdb`’ and a dictionary file ‘`libcheck_XXX.cif`’ in the directory in which Coot was started.

A future invocation of Get Monomer uses these file so that the monomer appears quickly[30].

5.19 Ligand from SMILES strings

Similarly, you can generate ligands using `File -> SMILES...` and providing a SMILES string and a code for the residue name (this is your name for the residue type and a dictionary will be generated for the monomer of this type). This function is also a wrapper to LIBCHECK.

---

[29] this is a wrapper round LIBCHECK, so you must have CCP4 suite to installed for this function to work

[30] rather than running LIBCHECK again

5.20 Find Ligands

You are offered a selection of maps to search (you can only choose one at a time) and a selection of molecules that act as a mask to this map. Finally you must choose which ligand types you are going to search for in this map[31]. Only molecules with less than 400 atoms are suggested as potential ligands.

If you do not have any molecules with less that 400 atoms loaded in Coot, you will get the message:

"Error: you must have at least one ligand to search for!"

New ligands are placed where the map density is and protein (mask) atoms are *not*). The masked map is searched for clusters using a default cut-off of 1.0σ. In weak density this cut-off may be too high and in such a case the cut-off value can be changed using something such as:

```
(set-ligand-cluster-sigma-level 0.8)
```

However, if the map to be searched for ligands is a difference map, a cluster level of 2.0 or 3.0 would probably be more appropriate (less likely to generate spurious sites).

Each ligand is fitted with rigid body refinement to each potential ligand site in the map and the best one for each site selected and written out as a pdb file. The clusters are sorted by size, the biggest one first (with an index of 0). The output placed ligands files have a prefix “best-overall” and are tagged by the cluster index and residue type of the best fit ligand in that site.

By default, the top 10 sites are tested for ligands - to increase this use:

```
(set-ligand-n-top-ligands 20)
```

5.20.1 Flexible Ligands

If the “Flexible?” checkbutton is activated, coot will generate a number of variable conformations (default 100) by rotating around the rotatable bonds (torsions). Each of these conformations will be fitted to each of the potential ligand sites in the map and the best one will be selected (again, if it passes the fitting criteria above).

Before you search for flexible ligands you must have read the mmCIF dictionary for that particular ligand residue type (`File -> Import CIF dictionary`).

Use:

```
(set-ligand-flexible-ligand-n-samples n-samples)
```

where ***`n-samples`*** is the number of samples of flexibility made for each ligand. Generally speaking, The more the number of rotatable bonds, the bigger this number should be.

By default the options to change these values are not in the GUI. To enable these GUI options, use the scripting function:

```
(ligand-expert)
```

5.20.2 Adding Ligands to Model

After successful ligand searching, one may well want to add that displayed ligand to the current model (the coordinates set that provided the map mask). To do so, use Merge Molecules (Section Section 5.28 [Merge Molecules], page 42).

---

[31] you can search for many different ligand types.

5.21 Flip Ligand

Sometimes a ligand is placed more or less in the correct position, but the orientation is wrong - or at least you might want to explore other possible orientation. To do that easily a function has been provided:

`(flip-ligand imol chain-id residue-number)`

This will flip the orientation of the residue around the Eigen vector corresponding to the largest Eigen value, exploring 4 possible orientations.

This function has been further wrapped to provide flipping for the active residue:

`(flip-active-ligand)`

This function can easily be bound to a key.

5.22 Find Waters

As with finding ligands, you are given a chose of maps, protein (masking) atoms. A final selection has to be made for the cut-off level, note that this value is the number of standard deviation of the density of the map *before* the map has been masked. The default sigma level (water positions must have density above this level) is set for a “2Fo-Fc”-style map. If you want to use a difference map, you must change the sigma level (typically to 3 sigma) otherwise you run the risk of fitting waters to difference map noise peaks.

Then the map is masked by the masking atoms and a search is made of features in the map about the electron density cut-off value. Waters are added if the feature is approximately water-sized and can make sensible hydrogen bonds to the protein atoms. The new waters are optionally created in a new molecule called “Waters”.

You have control over several parameters used in the water finding:

`(set-write-peaksearched-waters)`

which writes `ligand-waters-peaksearch-results.pdb`, which contains the water peaks (from the clusters) without any filtering and `ligand-waters.pdb` which are a disk copy filtered waters that have been either added to the molecule or from which a new molecule has been created.

`(set-ligand-water-to-protein-distance-limits min-d max-d)` sets the minimum and maximum allowable distances between new waters and the masking molecule (usually the protein). Defaults are 2.4 and 3.2Å.

`(set-ligand-water-spherical-variance-limit varlim)` sets the upper limit for the density variance around water atoms. The default is 0.12.

The map that is marked by the protein and is searched to find the waters is written out in CCP4 format as `"masked-for-waters.map"`.

5.22.1 Refinement Failure

Sometimes as a result of water fitting, you may see something like:

```
WARNING:: refinement failure
          start pos: xyz = (      17.1,     34.76,     60.42)
          final pos: xyz = (     17.19,     34.61,     60.59)
```

When Coot finds a blob, it does a crude positioning of an atom at the centre of the grid points. It then proceeds to move to the peak of the blob by a series of translations. There are a certain number of cycles, and if it doesn't reach convergence by the end of those cycles then you get the error message.

Often when you go to the position indicated, you can see why Coot had a problem in the refinement.

5.22.2 Blobs

After a water search, Coot will create a blobs dialog (see Section Section 7.4 [sec_blobs], page 56).

5.23 Add Terminal Residue

This creates a new residue at the C or N terminal extension of the residue clicked by fitting to the map. ϕ, ψ angle pairs are selected at random based on the Ramachandran plot probability (for a generic residue) and fitted to the density. By default there are 100 trials. It is possible that a wrong position will be selected for the terminal residue and if so, you can reject this fit and try again with Fit Terminal Residue[32]. Each of the trial positions are scored according to their fit to the map[33] and the best one selected. It is probably a good idea to run "Refine Zone" on these new residues.

If you use the Extensions (`Dock Sequence... -> Associate Sequence with Model`) to apply a PIR sequence file to a model then `Add Terminal Residue` will use the sequence alignment to determine the residue type of the added residue.

Sometimes, particularly with low resolution maps, the added terminal residue will wander off to somewhere inappropriate. This can be addressed in a number of ways:

1. `(set-terminal-residue-do-rigid-body-refine 0)` will disable rigid body fitting of the terminal residue fragment for each trial residue position (the default is 1 (on)) - this may help if the search does not provide good results.
2. to anneal the newly added residue back to the clicked residue (no matter where it ended up being positioned): `(set-add-terminal-residue-do-post-refine 1)`
3. `(set-add-terminal-residue-n-phi-psi-trials 200)` will change the number of trials (default is 100). This is useful if you think that Coot needs to search harder to find a good solution to the positioning of the next residue.

5.24 Add OXT Atom to Residue

At the C-terminus of a chain of amino-acid residues, there is a "modification" so that the C-O becomes a carbonyl, *i.e.* an extra (terminal) oxygen (OXT) needs to be added. This atom is added so that it is in the plane of the Cα, C and O atoms of the residue.

Scripting usage:

`(add-OXT-to-residue imol residue-number insertion-code chain-id)`[34],

where `insertion-code` is typically `""`.

Note, in order to place OXT, the N, CA, C and O atoms must be present in the residue - if (for example) the existing carbonyl oxygen atom is called "OE1" then this function will not work.

---

[32] usually if this still fails after two repetitions then it never seems to work.

[33] The map is selected using "Refine/Regularize Control"

[34] *e.g.* `(add-OXT-to-residue 0 428 "" "A")`

5.25 Add Atom at Pointer

By default, “Add Atom At Pointer” will pop-up a dialog from which you can choose the atom type you wish to insert[35]. Using `(set-pointer-atom-is-dummy 1)` you can by-pass this dialog and immediately create a dummy atom at the pointer position. Use an argument of `0` to revert to using the atom type selection pop-up on a button press.

The atoms are added to a new molecule called “Pointer Atoms”. They should be saved and merged with your coordinates outside of Coot.

5.26 Place Helix

The idea is to place a helix more or less “here” (the screen centre) by fitting to the electron density map. The algorithm is straightforward. First we move to the local centre of density, then examine the density for characteristic directions and fit ideal helices (of length 20 residues) to these directions. The helix is then extended if possible (by checking the fit to the map of residues added in ideal helix conformation) and chopped back if not. If the fit is successful, the helix is created in a new molecule called “Helix”. If the fit is not successful, there is instead a message added to the status bar. You can build the majority of a helical protein in a few minutes using this method (you will of course have to assemble the helices and assign residue numbers and sequence later).

This is available as a scripting function `(place-helix-here)` and in the GUI (in the “Other Modelling Tools” dialog).

5.27 Building Ideal DNA and RNA

The interface to building ideal polynucleotides can be found by pressing the “Ideal RNA/DNA. . . ” button on the “Other Modelling Tools” dialog.

For a given sequence, a choice of DNA or RNA, A or B form, single or double stranded is presented.

The interface may not gracefully handle uracils in DNA, thymines in RNA or B form RNA[36].

5.28 Merge Molecules

This dialog can be found under “Calculate” in the main menubar. This is typically used to add molecule fragments or residues that are in one molecule to the “working” coordinates[37].

5.29 Temperature Factor for New Atoms

The default temperature factor for new atoms is 30.0. This can be changed by the following

```
(set-default-temperature-factor-for-new-atoms 50.0)
```

---

[35] including sulfate or phosphate ions (in such a case, it is probably useful to do a “Rigid Body Fit Zone” on that new residue).

[36] But you don’t want those things anyway, right?

[37] For example, after a ligand search has been performed.

5.30 Applying NCS Edits

Let's imagine that you have 3-fold NCS. You have molecule "A" as your master molecule and you make edits to that molecule. Now you want to apply the edits that you made to "A" (the NCS master chain ID) to the "B" and "C" molecules (i.e. you want the "B" and "C" molecules to be rotated/translated versions of the "A" molecule). How is that done?

There are now guis to NCS command to help you out (under Extensions). However, for completeness here are the scripting versions:

```
(copy-from-ncs-master-to-others imol master-chain-id)
```

If you have only a range of residues, rather than a whole chain to replace:

```
(copy-residue-range-from-ncs-master-to-others imol master-chain-id
start-resno end-resno)
```

e.g.

```
(copy-residue-range-from-ncs-master-to-others 0 "A" 1 5)
```

If you want to copy a residue range to a specific chain, or specific list of chains (rather than all NCS peer chains) then make a list of the chain-ids that you wish replaced:

```
(copy-residue-range-from-ncs-master-to-chains 0 "A" 1 5 (list "C"))
```

in this case, just the residues in the "C" chain is replaced.

5.31 Running Refmac

Use the "Run Refmac..." button to select the dataset and the coordinates on which you would like to run Refmac. Note that here Coot only allows the use of datasets which has Refmac parameters set as the MTZ file was read. By default, Coot displays the new coordinates and the new map generated from refmac's output MTZ file. Optionally, you can also display the difference map.

You can add extra parameters (data lines) to refmac's input by storing them in a file called `refmac-extra-params` in the directory in which you started coot.

You can also provide extra/replacement parameters for refmac by setting the variable `refmac-extra-params` to a list of strings, for example:

```
(set! refmac-extra-params (list "REFINE MATRIX 0.1" "MAKE HYDROGENS NO"))
```

Coot "blocks"[38] until Refmac has terminated[39].

The default refmac executable is `refmac5` it is presumed to be in the path. If you don't want this, it can be overridden using a re-definition either at the scripting interface or in one's `~/.coot` file *e.g.*:

- `(define refmac-exec "/e/refmac-new/bin/refmac5.6.3")`

After running refmac several times, you may find that you prefer if the new map that refmac creates (after refmac refinement) is the same colour as the previous one (from before this refmac refinement). If so, use:

```
(set-keep-map-colour-after-refmac 1)
```

---

[38] *i.e.* Coot is idle and ignores all input.

[39] This is not an ideal feature, of course and will be addressed in future.... Digressive Musing: If only computers were fast enough to run Refmac interactively...

which will swap the colours of then new and old refmac map so that the post-refmac map has the same colour as the pre-refmac map and the pre-refmac map is coloured with a different colour.

5.32 Running SHELXL

Coot can read shelx `.res` files and write `.ins` files, and thus one can refine using SHELXL in a convenient manner using the function

`(shelxl-refine` *`imol . hkl-file-name`*`)`

(the *`hkl-file-name`* is an optional argument)

e.g.

`(shelxl-refine 0)`

or

`(shelxl-refine 0 "insulin.hkl")`

In the former case, coot will presume that there is a SHELX `hkl` file corresponding to the `res` file that you read in; if there is not coot will print a warning and not try to run shelxl. In the latter case, you can specify the location of the `hkl` file.

After shelxl has finished, coot will automatically read in the resulting `res` coordinates, the `fcf` file, convert the data to `mmCIF` format and read that, which generates a σ_A map and a difference map.

Coot creates a time stamped `ins` file and a time-stamped sym-link to the `hkl` file in the `coot-shelxl` directory.

Please note that the output `ins` file will not be particularly useful (and thus shelxl will fail) if the input file was not in SHELX `ins` format.

There is a GUI for this operation under the “Extensions” menu item.

5.33 Clear Pending Picks

Sometimes one can click on a button[40] unintentionally. This button is there for such a case. It clears the expectation of an atom pick. This works not only for modelling functions, but also geometry functions (such as Distance and Angle).

5.34 Delete

Single atoms or residues can be deleted from the molecule using “Delete...” from the “Model/Fit/Refine”dialog. Pressing this button results in a new dialog, with the options of “Residue” (the default), “Atom” and “Hydrogen Atoms”. Now click on an atom in the graphics - the deleted object will be the whole residue of the atom if “Residue” was selected and just that atom if “Atom” was selected. Note that if a residue has an alternative conformation, then “Delete Residue” will delete only the conformation that matches that alternative conformation specifier of the clicked atom.

Only waters are deletable if the "Water" check button is active and waters are not deletable if the "Residue/Monomer" check button is active. This is to reduce mis-clicking.

---

[40] such that Coot would subsequently expect an atom selection “pick” in the graphics window.

To rotate the view when in “Delete Mode”, use Ctrl left-mouse.

If you want to delete multiple items you can use check the “Keep Delete Active” check-button on this dialog This will will keep the dialog open, ready for deletion of next item.

5.35 Sequence Assignment

You can assign a (FASTA format) sequence to a molecule using:

```
(assign-fasta-sequence imol chain-id fasta-seq)
```

This function has been provided as a precursor to functions that will (as automatically as possible) mutate your current coordinates to one that has the desired sequence. It will be used in automatic side-chain assignment (at some stage in the future).

5.36 Building Links and Loops

Coot can make an attempt to build missing linking regions or loops[41]. This is an area of Coot that needs to be improved, currently O does it much better. We will have several different loop tools here[42]. For now there is `Calculate -> Fit Gap` or the scripting function:

```
(fit-gap imol chain-id start-resno stop-resno)
```

and

```
(fit-gap imol chain-id start-resno stop-resno sequence)
```

the second form will also mutate and try to rotamer fit the provided sequence.

Example usage: let’s say for molecule number 0 in chain `"A"` we have residues up to 56 and then a gap after which we have residues 62 and beyond:

```
(fit-gap 0 "A" 57 61 "TYPWS")
```

5.37 Fill Partial Residues

After molecular replacement, the residues of your protein could well have the correct sequence but be chopped back to CG or CB atoms. There is a function to fill such partially-filled residues:

```
(fill-partial-residues imol)
```

This identifies residues with missing atoms, then fills them and does a rotamer fit and real-space refinement.

If you want to fill the side chain of just one residue

```
(fill-partial-residue imol chain-id res-no ins-code)
```

this does a auto-fit-best-rotamer and a refinement on the resulting side-chain position.

5.38 Changing Chain IDs

You can change the chain ids of chains using `Calculate -> Change Chain IDs...`. Coot will block an attempt to change the whole of a chain and the target chain id already exists in the molecule.

If you use the "Residue Range" option then you can insert residues with non-conflicting residue number into pre-existing chains.

---

[41] the current single function doesn’t always perform very well in tests

[42] I suspect that there is not one tool that fits for all.

5.39 Setting Occupancies

As well as the editing "Residue Info" to change occupancies of individual atoms, one can use a scripting function to change occupancies of a whole residue range:

- `(zero-occupancy-residue-range` *`imol chain-id resno-start resno-last`*`)`

  example usage:

  `(zero-occupancy-residue-range 0 "A" 23 28)`

  This is often useful to zero out a questionable loop before submitting for refinement. After refinement (with refmac) there should be relatively unbiased density in the resulting 2Fo-Fc-style and difference maps.

  Similarly there is a function to reverse this operation:

- `(fill-occupancy-residue-range` *`imol chain-id resno-start resno-last`*`)`

5.40 Fix Nomenclature Errors

Currently this is available only in scripting form:

`(fix-nomenclature-errors` *`imol`*`)`

This will fix atoms nomenclature problems in molecule number *`imol`* according to the same criteria as WATCHECK[43] *e.g.* Chi-2 for Phe, Tyr, Asp, and Glu should be between -90 and 90 degrees. Note that Val and Leu nomenclature errors are also corrected.

5.41 Rotamer Fix Whole Protein

There is an experimental scripting function

`(fit-protein` *`imol`*`)`

which does a auto-fit rotamer and Real Space Refinement for each residue. The graphics follow the refinement.

5.42 Refine All Waters

All the waters in a model can be refined (that is, moved to the local density peak) using

`(fit-waters` *`imol`*`)`

This is a non-interactive function (the waters are moved without user intervention).

5.43 Moving Molecules/Ligands

Often you want to move a ligand (or some such) from wherever it was read in to the position of interest in your molecule (*i.e.* the current view centre). There is a GUI to do this: `Calculate -> Move Molecule Here`.

There are scripting functions available for this sort of thing:

`(molecule-centre` *`imol`*`)`

will tell you the molecule centre of the *`imol`*th molecule.

`(translate-molecule-by imol x-shift y-shift z-shift)` will translate all the atoms in molecule *`imol`* by the given amount (in Ångströms).

---

[43] R.W.W. Hooft, G. Vriend, C. Sander, E.E. Abola, Errors in protein structures. *Nature* (1996) **381**, 272-272.

```
(move-molecule-to-screen-centre imol)
```

will move the *`imol`*th molecule to the current centre of the screen (sometimes useful for imported ligands). Note that this moves the atoms of the molecule - not just the view of the molecule.

5.44 Modifying the Labels on the Model/Fit/Refine dialog

If you don't like the labels "Rotate/Translate Zone" or "Place Atom at Pointer" and rather they said something else, you can change the button names using:

```
(set-model-fit-refine-rotate-translate-zone-label "Move Zone")
```

and

```
(set-model-fit-refine-place-atom-at-pointer "Add Atom")
```

6 Map-Related Features

6.1 Maps in General

Maps are “infinite,” not limited to pre-calculated volume (the “Everywhere You Click - There Is Electron Density” (EYC-TIED) paradigm) symmetry-related electron density is generated automatically. Maps are easily re-contoured. Simply use the scroll wheel on you mouse to alter the contour level (or `-`/`+` on the keyboard).

Maps follow the molecule. As you recentre or move about the crystal, the map quickly follows. If your computer is not up to re-contouring all the maps for every frame, then use `Draw -> Dragged Map...` to turn off this feature.

6.1.1 Map Reading Bug

Unfortunately, there is a bug in map-reading. If the map is not a bona-fide CCP4 map[1], then coot will crash. Sorry. A fix is in the works but “it’s complicated”. That’s why maps are limited to the extension ".ext" and ".map", to make it less likely a non-CCP4 map is read.

6.2 Create a Map

From MTZ, mmCIF and .phs data use `File -> Open MTZ, CIF or phs...`. You can then choose the MTZ columns for the Fourier synthesis. The button “Expert mode” also adds to the options any anomalous columns you may have in the MTZ file (a -90 degree phase shift will be applied). It also provides the option to apply resolution limits.

From a CCP4 map use `File -> Read Map`. After being generated/read, the map is immediately contoured and centred on the current rotation centre.

6.2.1 Auto-read MTZ file

This function allows Coot to read an MTZ file and make a map directly (without going through the column selection procedure). The default column labels for auto-reading are "FWT" and "PHWT" for the 2Fo-Fc-style map, "DELFWT" and "PHDELWT" for the difference map. You can change the column labels that Coot uses for auto-reading - here is an example of how to do that:

`(set-auto-read-column-labels "2FOFCWT" "PHIWT" 0) (set-auto-read-column-labels "FOFCWT" "DELPHIWT" 1)`

By default the difference map is created in auto-reading the MTZ file. If you don’t want a difference map, you can use the function:

`(set-auto-read-do-difference-map-too 0)`

6.2.2 Reading CIF data

There are several maps that can be generated from CIF files that contain observed Fs, calculated Fs and calculated phases:

- `(read-cif-data-with-phases-fo-alpha-calc` *`cif-file-name`*`)` Calculate an atom map using F_obs and α_calc

---

[1] *e.g.* it’s a directory or a coordinate filename.

- `(read-cif-data-with-phases-2fo-fc` *`cif-file-name`*`)` Calculate an atom map using F_obs, F_calc and α_calc
- `(read-cif-data-with-phases-fo-fc` *`cif-file-name`*`)` Calculate an difference map using F_obs, F_calc and α_calc.

6.2.3 Reading PHS data

There are 2 ways to read data by scripting:

`(read-phs-and-make-map-using-cell-symm` *`phs-file-name space-group-name a b c alpha beta gamma`*`)`

`(read-pdb-and-make-map-with-reso-limits` *`imol-previous phs-file-name reso-limit-low reso-limit-high`*`)`

The first specifies the cell explicitly, and *`alpha, beta`* and *`gamma`* are specified in degrees.

The second form allows the specification of resolution limits and takes the cell and symmetry from a previous molecule (typically a pdb file).

6.3 Map Contouring

Maps can be re-contoured using the middle-mouse scroll-wheel (buttons 4 and 5 in X Window System(TM) terminology). Scrolling the mouse wheel will change the map contour level and the map it redrawn. If you have several maps displayed then the map that has its contour level changed can be set using `HID -> Scrollwheel -> Attach scroll-wheel to which map?`. If there is only one map displayed, then that is the map that has its contour level changed (no matter what the scroll-wheel is attached to in the menu). The level of the electron density is displayed in the top right hand corner of the OpenGL canvas.

Use keyboard `+` or `-` to change the contour level if you don't have a scroll-wheel[2].

If you are creating your map from an MTZ file, you can choose to click on the "is difference map" button on the Column Label selection widget (after a data set filename has been selected) then this map will be displayed in 2 colours corresponding to `+` and `-` the map contour level.

If you read in a map and it is a difference map then there is a checkbutton to tell Coot that.

If you want to tell Coot that a map is a difference map after it has been read, use:

`(set-map-is-difference-map` *`imol`*`)`

where *`imol`* is the molecule number.

By default the change of the contour level is determined from the sigma of the map. You can change this in the map properties dialog or by using the scripting function:

`(set-contour-by-sigma-step-by-mol` *`step on/off? imol`*`)`

where

`step` is the difference in sigma from one level to the next (typically 0.2)

`on/off?` is either 0 (sigma stepping off) or 1 (sigma stepping on)

---

[2] like I don't on my Mac.

By default the map radius[3] is 10Å. The default increment to the electron density depends on whether or not this is a difference map (0.05 $e^{-}/Å^3$ for a "2Fo-Fc" style map and 0.005 $e^{-}/Å^3$ for a difference map). You can change these using `Edit -> Map Parameters` or by using the "Properties" button of a particular map in the Display Control (Display Manager) window.

6.4 Map Extent

The extent of the map can be set using the GUI (`Edit -> Map Parameters -> Map Radius`) or by using the scripting function, *e.g.*:

```
(set-map-radius 13.2)
```

6.5 Map Contour "Scrolling" Limits

Usually one doesn't want to look at negative contour levels of a map[4], so Coot has by default a limit that stops the contour level going beyond (less than) 0. To remove the limit:

`(set-stop-scroll-iso-map 0)` for a 2Fo-Fc style map

`(set-stop-scroll-diff-map 0)` for a difference map

To set the limits to negative (*e.g.* -0.6) levels:

```
(set-stop-scroll-iso-map-level -0.6)
```

and similarly:

```
(set-stop-scroll-diff-map-level -0.6)
```

where the level is specified in $e^{-}/Å^3$.

6.6 Map Line Width

The width of the lines that describe the density can be changed like this:

```
(set-map-line-width 2)
```

The default line width is 1.

6.7 "Dynamic" Map colouring

By default, maps get coloured according to their molecule number. The starting colour (*i.e.* for molecule 0) is blue. The colour of a map can be changed by `Edit -> Map Colour...` The map colour gets updated as you change the value in the colour selector[5]. Use "OK" to fix that colour.

As subsequent maps are read, they are coloured by rotation round a colour wheel. The default colour map step is 31 degrees. You can change this using:

`(set-colour-map-rotation-for-map` *`step`*`)`

---

[3] actually, it's a box.

[4] in a coot difference map you will get to see the negative level contoured at the inverted level of the positive level, what I mean is that you don't want to see the "positive" level going less than 0.

[5] takes you right back to the good old Frodo days, no?

6.8 Difference Map Colouring

For some strange reason, some crystallographers[6] like to have their difference maps coloured with red as positive and green as negative, this option is for them:

```
(set-swap-difference-map-colours 1)
```

This option will allow the “blue is positive, red is negative” colour scheme on “Edit `->` Map Colour”.

6.9 Make a Difference Map

Using the “Make a Difference Map” function in the Extensions menu, one can make a difference from two arbitrary maps. The maps need not be on the same griding, or in the same space group even. The resulting map will be on the same griding and space group as the “Reference” map.

6.10 Make an Averaged Map

There is a scripting interface to the generation of map averages. As above, the maps need not be on the same grid or in the same space group. The resulting map will have the same gridding and space group as the first map in the list. Typical usage:

```
(average-map '((1 1.0) (2 1.0))))
```

The argument to `(average-map` is a list of lists, each list element is a list of the map number and a weighting factor (1.0 in this case).

6.11 Map Sampling

By default, the Shannon sampling factor is the conventional 1.5. Use larger values (`Edit -> Map Parameters -> Sampling Rate`) for smoother maps[7].

This value can be set by the scripting command

```
(set-map-sampling-rate 2.5)
```

6.12 Dragged Map

By default, the map is re-contoured at every frame during a drag (Ctrl Left-mouse). Sometimes this can be annoyingly slow and jerky so it is possible to turn it off: `Draw -> Dragged Map -> No`.

To change this by scripting:

```
(set-active-map-drag-flag 0)
```

6.13 Dynamic Map Sampling and Display Size

If activated (`Edit -> Map Parameters -> Dynamic Map Sampling`) the map will be re-sampled on a more coarse grid when the view is zoomed out. If “Display Size” is also activated, the box of electron density will be increased in size also. In this way, you can see electron density for big maps (many unit cells) and the graphics still remain rotatable.

---

[6] Jan Dohnalek, for instance.

[7] a value of 2.5 is often sufficient.

If you want to have these functions active for all maps, add the following to your initialization file Section 3.10.2 [Scheme], page 12:

```
(set-dynamic-map-sampling-on) (set-dynamic-map-size-display-on)
```

6.14 Skeletonization

The skeleton (also known as "Bones"[8]) can be displayed for any map. A map can be skeletonized using `Calculate -> Map Skeleton...`. Use the option menu to choose the map and click "On" then "OK" to the generate the map (the skeleton is off by default).

The level of the skeleton can be changed by using `Edit -> Skeleton Parameters... -> Skeletonization Level...` and corresponds to the electron density level in the map. By default this value is 1.2 map standard deviations. The amount of map can be changed using `Edit -> Skeleton Parameters... -> Skeleton Box Radius...`[9]. The units are in Ångströms, with 40 as the default value.

The skeleton is often recalculated as the screen centre changes - but not always since it can be an irritatingly slow calculation. If you want to force a regeneration of the displayed skeleton, simply centre on an atom (using the middle mouse button) or press the S key.

6.15 Map Sharpening

It can be educational (even useful at lower resolutions) to sharpen or blur a map. This can be achieved with the sharpening tool `Calculate -> Map Sharpening...`. By default, the maximum and minimum sharpness is +/- $30Å^2$, this can be changed (in this case to 80) using:

```
(set-map-sharpening-scale-limit 80)
```

This currently only works on maps created by reading an MTZ (or other) reflection data file.

6.16 Pattersons

Pattersons can be generated using the `make-and-draw-patterson` function. Example usage:

```
(make-and-draw-patterson mtz-file-name f-col sig-f-col weight-col
use-weights-flag)
```

where *`use-weights-flag`* is either `0` or `1`.

6.17 Masks

A map can be masked by a set of coordinates. Use the scripting function:

```
(mask-map-by-molecule imol-map imol-model invert-mask?)
```

If *`invert-mask?`* is 0, this will create a new map that has density only where there are no (close) coordinates. If *`invert-mask?`* is 1 then the map density values will be set to zero everywhere *except* close to the atoms of molecule number *imol-model*.

The radius of the mask around each atom is 2.0Å by default. You can change this using:

---

[8] If you're living in Sweden... or Captain Kirk, that is.

[9] you may think it strange that a box has a radius, this is an idiosyncrasy of Coot.

`(set-map-mask-atom-radius` ***`radius`***`)`

There is a GUI interface to Map Masking under the Extensions menu.

6.17.1 Example

If one wanted to show just the density around a ligand:

1. Make a pdb file the contains just the ligand and read it in to Coot - let's say it is molecule 1 and the ligand is residue 3 of chain "L".
2. Get a map that covers the ligand (*e.g.* from refmac). Let's say this map is molecule number 2.
3. Mask the map:

   `(mask-map-by-molecule 2 1 1)`

   This creates a new map. Turn the other maps off, leaving only the masked map.

To get a nice rendered image, press F8 (see Section Section 3.6 [Raster3D], page 10).

6.18 Trimming

If you want to remove all the atoms[10] that lie "outside the map" (*i.e.* in low density) you can use

`(trim-molecule-by-map` ***`imol-coords imol-map density-level delete/zero-occ?`***`)`

where ***`delete/zero-occ?`*** is `0` to remove the atoms and `1` to set their occupancy to zero.

There is a GUI interface for this feature under the "Extensions" menu item.

6.19 Map Transformation

If you want to transform a map, you can do it thusly:

`(transform-map` ***`imol rotation-matrix trans point radius`***`)`

where:

`rotation-matrix` is a 9-membered list of numbers for an orthogonal rotation matrix.

`trans` is a 3-membered list of numbers (distances in Ångstöms).

`point` is a 3-membered list of numbers (centre point in Ångstöms).

`radius` is a single number (also in Ångstöms).

This applies the rotation ***`rotation-matrix`*** and a translation ***`trans`*** to a map fragment, so that when the transformation is applied the centre of the new map is at ***`point`***.

Example usage:

`(transform-map 2 '(1 0 0 0 1 0 0 0 1) '(0 0 1) (rotation-centre) 10)`

which transforms map number 2 by a translation of 1Å along the Z axis, centred at the screen centre for 10Å around that centre.

Here's a more real-world example:

Let's say we want to tranform the density over the "B" molecule to a position over the "A" molecule. First we do a LSQ transformation to get the rotation and translation that moves the "B" coordinates over the "A" coordinates:

In the terminal output we get:

---

[10] or set their occupancy to zero

```
|    0.9707,    0.2351,   0.05033|
|  -0.04676,      0.39,   -0.9196|
|   -0.2358,    0.8903,    0.3896|
(    -33.34,     21.14,     18.82)
```

The centre of the “A” molecule is at (58.456, 5.65, 11.108). So we do:

`(transform-map 3 (list 0.9707 0.2351 0.05033 -0.04676 0.39 -0.9196 -0.2358 0.8903 0.3896) (list -33.34 21.14 18.82) (list 58.456 5.65 11.108) 8)`

Which creates a map over the middle of the “A” molecule. Note that using a too high ***radius*** can cause overlap problems, so try with a small ***radius*** (*e.g.* 5.0) if the resulting map looks problematic.

Alternatively, instead of typing the whole matrix, you can use a coordinates least-squares fit to generate the matrix for you. `(transform-map-using-lsq-matrix)` does just that.

Heres how to use it:

`(transform-map-using-lsq-matrix` *`imol-ref ref-chain ref-resno-start ref-resno-end imol-mov mov-chain mov-resno-start mov-resno-end imol-map about-pt radius`*`)`

Hopefully the arguments are self explanatory (***ref*** refers to the reference molecule, of course and ***about-pt*** is a 3-number list such as is returned by `(rotation-centre)`).

We can now export that map, if we want.

6.20 Export Map

You can write out a map from Coot (e.g. one from NCS averaging, or masking or general transformation) using the export map function:

`(export-map` *`imol filename`*`)`

e.g.

`(export-map 4 "ncs-averaged.map")`

7 Validation

The validation functions are still being added to from time to time. In future there will be more functions, particularly those that will interface to other programs.

7.1 Ramachandran Plots

Ramachandran plots are "dynamic". When you edit the molecule (*i.e.* move the coordinates of some of atoms) the Ramachandran plot gets updated to reflect those changes. Also the underlying ϕ/ψ probability density changes according to the selected residue type (*i.e.* the residue under the mouse in the plot). There are 3 different residue types: GLY, PRO, and not-GLY-or-PRO[1].

When you mouse over a representation of a residue (a little square or triangle[2]) the residue label pops up. The residue is "active" *i.e.* it can be clicked. The "graphics" view changes so that the Cα of the selected residue is centred. In the Ramachandran plot window, the current residue is highlighted by a green square.

The underlying distributions are taken from the Richardson's Top500 structures `http://kinemage.biochem.duke.edu/databases/top500.php`.

The probability levels for acceptable (yellow) and preferred (red) are 0.2% and 2% respectively.

You can change the contour levels:

```
(set-ramachandran-plot-contour-levels 0.025 0.003)
```

You can change the "blocksize" (the default is 10 degrees) of the contours using

```
(set-ramachandran-plot-background-block-size 5)
```

These comes into effect when a new plot is created (it doesn't change plots currently displayed).

7.2 Geometry Analysis

A restraints-based geometry analysis of the molecule. The distortion is weighted by atom occupancy. The distortion of the geometry due to links is shared between the contributing residues.

Note that only the first model of a multi-model molecule is analysed.

7.3 Chiral Volumes

The dictionary is used to identify the chiral atoms of each of the model's residues. A clickable list is created of atoms whose chiral volume in the model is of a different sign to that in the dictionary.

During refinement and regularization, Coot will pop-up dialogs warning about chiral volume errors - if you have them. This can be annoying[3]. You can inhibit this dialog like this:

```
(set-show-chiral-volume-errors-dialog 0)
```

---

[1] the not-GLY-or-PRO is the most familiar Ramachandran plot.

[2] prolines have a grey outline rather than a black one, triangles are glycines.

[3] but that's partly the idea, I suppose.

7.3.1 Fixing Chiral Volume Errors

There are two obvious ways:

1) mutate and auto-fit rotamer (mutate it to the residue type that it is)

2) RS Refine the residue and invert the chiral centre by pulling an atom. Usually you can pull the CA to the other side of the plane made by the chiral neighbouring atoms (using ctrl left-click). Sometimes giving the CB a good old tweak is the easier way.

Inverting the CB of THR is easier, just move the OG so that the plane of the neighbours is on the other side of the CB (again with ctrl left-click).

7.4 Blobs: a.k.a. Unmodelled density

This is an interface to the Blobs dialog. A map and a set of coordinates that model the protein are required.

A blob is region of relatively high residual election density that cannot be explained by a simple water. So, for example, sulfates, ligands, mis-placed sidechains or unbuilt terminal residues might appear as blobs. The blobs are in order, the biggest[4] at the top.

7.5 Difference Map Peaks

This is one of the fastest ways to validate a model and its data (presuming that the difference map comes from a post-refinement mFo-DFc map). It highlights regions where the model and the data do not agree.

Lesser peaks within a certain distance (by default, 2.0Å) of a large peak are not shown. This cuts down on the number of times one is navigated to a particular region because of ripple or other noise peaks around a central peak.

This value can be queried:

```
(difference-map-peaks-max-closeness)
```

and adjusted:

```
(set-difference-map-peaks-max-closeness 0.1)
```

7.6 Check Waters by Difference Map

Sometimes waters can be misplaced - taking the place of sidechains or ligands or crystallization agents such as phosphate for example[5]. In such cases the variance of the difference map can be used to identify these problems.

This function is also useful to check anomalous maps. Often waters are placed in density that is really a something else, perhaps a cation, anion, sulphate or a ligand. If such an atom diffracts anomalously this can be identified and corrected.

By default the waters with a map variance greater than 3.5σ are listed. One can be more rigorous by using a lower cut-off:

```
(set-check-waters-by-difference-map-sigma-level 3.0)
```

The scripting interface is:

---

[4] and therefore most interesting

[5] or the water should be more properly modelled as anistrotropic or a split partial site

```
(check-waters-by-difference-map imol-coords imol-diff-map)
```

where *imol-coords* is the molecule number of the coordinates that contain the waters to be checked

imol-diff-map is the molecule number of the difference map (it must be a difference map, not an "ordinary" map). This difference map must have been calculated using the waters. So there is no point in doing this check immediately after "Find Waters". You will need to run Refmac or some other refinement first first[6].

7.7 Molprobity Tools Interface

The molprobity tools '`probe`' and '`reduce`' have been interfaced into Coot (currently, the interface is not as slick as it might be). However, the tools are useful and can be used in the following way:

first we need to tell Coot where to find the relevant executables (typically you would add the following lines to you '`~/.coot`' file):

```
(define *probe-command* "/path/to/probe/executable")
(define *reduce-command* "/path/to/reduce/executable")
```

now the probe hydrogens and probe dots can be generated using `Validate -> Probe Clashes` (or in the Scripting Window):

```
(probe imol)
```

where *imol* is the molecule number of coordinates to be probed. A new molecule with Hydrogens is created (by '`reduce`') and read in.

By default Coot creates a new molecule for the molecule that now has hydrogens. To change this:

```
(set! reduce-molecule-updates-current #t)
```

and that, as you can guess, replaces, rather than adds to the "probed" molecule.

This gives a "static" view of the molecule's interactions.

To get a dynamic view (which is currently only enabled on rotating chi angles) add these to your '`~/.coot`' file:

```
(set-do-probe-dots-on-rotamers-and-chis 1)
```

To get a semi-static view (dots are regenerated in the region of zone after a "Real Space Refinement"):

```
(set-do-probe-dots-post-refine 1)
```

7.8 GLN and ASN B-factor Outliers

It is often difficult to detect by eye the correct orientation of the amino-carbonylo group of GLN and ASNs. However, we can use (properly refined) temperature factors to detect outliers. We take the Z value as half the difference between the B-factor of the NE2 and OE1 divided by the standard deviation of the B-factors of the rest of the residue. An analysis of GLNs and ASNs of high resolutions structures indicates that a Z value of greater than 2.25 indicates a potential (if not probable) flip. A "Fix" button is provided in the resultant dialog make this easy to do.

---

[6] and remember to check the difference map button in the "Run Refmac" dialog

This analysis was added after discussions with Atsushi Nakagawa and so is called "Nakagawa's Bees".

The analysis does not check residues with multiple conformations.

7.9 Validation Graphs

Coot provides several graphs that are useful for model validation (on a residue by residue basis): residue density fit, geometry distortion, temperature factor variance, peptide distortion and rotamer analysis.

7.9.1 Residue Density Fit

The density fit graph shows the density fit for residues. The score is the average electron density level at the atom centres of the atoms in the residue. The height of the blocks is inversely proportional to the density average.

The residue density fit is by default scaled to a map that is calculated on the absolute scale. Sometimes you might be using a map with density levels considerably different to this, which makes the residue density fit graph less useful. To correct for this you can use the scripting function:

`(set-residue-density-fit-scale-factor` *`factor`*`)`

where *`factor`* would be $1/(4\sigma_{map})$ (as a rule of thumb).

`(residue-density-fit-scale-factor)` returns the current scale factor (default 1.0).

There is also a GUI to this:

`Extensions -> Refine... -> Set Density Fit Graph Weight...`

7.9.2 Rotamer Analysis

Residue rotamers are scored according to the prior likelihood. Note that when CD1 and CD2 of a PHE residue are exchanged (simply a nomenclature error) this can lead to large red blocks in the graph (apparently due to very unlikely rotamers). There are several other residues that can have nomenclature errors like this. To fix these problems use

`(fix-nomenclature-errors` *`imol`*`)`

7.9.3 Temperature Factor Variance

This idea is from Eleanor Dodson, who liked to use the standard deviation of a residue's temperature factors to highlight regions of questionable structure.

Note that Hydrogens are ignored in this analysis.

7.9.4 Peptide Omega Angle Distortion

Some variability of the ω is to be expected in the peptide bond. But not too much. Anything more than 13 degrees is suspicicous. Unexpected peptide bonds show up red by default. If cis peptides *are* to be expected, and should not marked as bad, then you can tell this to Coot using:

`Edit -> Preferences -> Geometry -> Cis-Peptides -> No`

8 Representation

8.1 Surfaces

Coot uses the surface code from Gruber and Noble (2004).

Coot uses the partial charges of the atoms (the *partial_charge* field in the *_chem_comp_atom* block) from the charge dictionary item in the refmac (or other) cif dictionary. However, partial charges are only used under certain conditions

1) the molecule consists of less than 100 atoms

or

2) the number of atoms in the molecule that are hydrogens is at least 15% of the total number of atoms in the molecule

If partial charges are not used, then the fall-back is to use charges from side-chains charged at physiological pH (Arg, Lys, Asp, Glu).

9 Hints and Usage Tips

9.1 Documentation

This manual is on the web where it can be searched:

- `http://www.biop.ox.ac.uk/coot/doc/user-manual.html` monolithic version
- `http://www.biop.ox.ac.uk/coot/doc/chapters/user-manual_toc.html` which is split into sections

In the Menu item “About”, under “Online Docs URL...” there is a entry bar that can be used to search the Coot documentation via Google. The results are returned as a web page in web browser. The browser type can be specified as in this example:

```
(set-browser-interface "firefox")
```

Example usage can be found in ‘`xxx/share/coot/scheme/group-settings.scm`’

9.2 Low Resolution

Building structures using low resolution data is a pain. We hope to make it less of a pain in future, but there are some things that you can do now.

- [Add Planar Peptide Restraints] Add restraints via scripting command
- [Use Secondary Structure Restraints] where appropriate under Refinement Control
- [Check Chirals] Check Chiral Volumes regularly
- [Change the Weighing Scheme] `(set-matrix 20.0)` [Default is 60, the lower the number the more the geometry is idealised]

9.3 Coot Droppings

This describes the files and directory that coot leaves behind after it has been fed (sorry, I mean “used”). Everything except the `0-coot.state.scm` state file can comfortably be deleted if needed after coot has finished.

You can stop the state and history files being written if you start coot with the `--no-guano` option.

- `0-coot.state.scm` The most important file. This contains the state of coot when you last exited. It contains things like which molecules were read, the maps, the colours of the molecules and map, the screen centre, map size and so on. When restarting a coot session, this file should usually be used.
- `0-coot-history.scm` The history of coot commands you used in your last coot session in scheme format. Incomplete history. One day this will be a complete history of the session suitable for uploading into a database describing the model modification.
- `0-coot-history.py` The history of coot commands you used in your last coot session in python format.
- `coot-download` directory where the files downloaded from the network (e.g. from the EBI and EDS) go.
- `coot-backup` Each model modification generates the saving of coordinates as a pdb file in this directory.

- `coot-refmac` When running REFMAC using the Coot interface, the input to refmac and the output go in this directory.
- `coot-molprobity` When running Molprobity's Probe and Reduce using the Coot interface, the input and output go in this directory.

9.4 Clearing Backups

Coot will occasionally ask you to clear up the '`coot-backup`' directory. You can adjust the behaviour in a number of ways:

- `(define *clear-out-backup-run-n-days* 3)` will run the backup clearance every 3 days (the default is every 7).
- `(define *clear-out-backup-old-days* 1)` will clear out files older then 1 day (rather than the default 7 days).
- You can create your own version of the function that is run on exiting Coot: `(clear-backups-maybe)`

So, if you wanted to clear out everything more than 1 day old, every time, without Coot asking you about it:

```
(define *clear-out-backup-run-n-days* 0)
(define *clear-out-backup-old-days* 1)
(define (clear-backups-maybe)
  (delete-coot-backup-files 'delete)
  (coot-real-exit 0))
```

9.5 Getting out of "Translate" Mode

If you get stuck in "translate" mode in the GL canvas (*i.e.* mouse does not rotate the view as you would expect) simply press and release the Ctrl key to return to "rotate" mode.

9.6 Getting out of "Continuous Rotation" Mode

The keyboard I key toggles the "continuous rotation" mode. The menu item `Draw -> Spin View On/Off` does the same thing.

9.7 Getting out of "Label Atom Only" Mode

Similarly, if you are stuck in a mode where the "Model/Fit/Refine" buttons don't work (the atoms are not selected, only the atom gets labelled), press and release the Shift key.

9.8 Button Labels

Button labels ending in "..." mean that a new dialog will pop-up when this button is pressed.

9.9 Picking

Note that left-mouse in the graphics window is used for both atom picking and rotating the view, so try not to click over an atom when trying to rotate the view when in atom selection mode.

9.10 Resizing View

Click and drag using right-mouse (up and down or left and right) to zoom in and out.

9.11 Scroll-wheel

To change the map to which the scroll-wheel is attached, use the scroll check button in the Display Manager or use `HID -> Scrollwheel -> Attach Scrollwheel to which map?`

9.12 Slow Computer Configuration

Several of the parameters of Coot are chosen because they are reasonable on my "middle-ground" development machine. However, these parameters can be tweaked so that slower computers perform better:

- `(set-smooth-scroll-steps 4) ; default 8`
- `(set-smooth-scroll-limit 30) ; Angstroms`
- `(set-residue-selection-flash-frames-number 3);`
- `(set-skeleton-box-size 20.0) ; A (default 40).`
- `(set-active-map-drag-flag 0) ; turn off recontouring every step`
- `(set-idle-function-rotate-angle 1.5) ; continuous spin speed`

10 Other Programs

10.1 findligand

`findligand` is a stand-alone command-line program that uses the libraries of Coot.

`findligand` provides a number of command line arguments for increased flexibility:

- `--pdbin` *`pdb-in-filename`*

  where *`pdb-in-filename`* is the protein (typically)
- `--hklin` *`mtz-filename`*
- `--f` *`f_col_label`*
- `--phi` *`phi_col_label`*
- `--clusters` *`nclust`*

  where *`nclust`* is the number of density clusters (potential ligand sites) to search for
- `--sigma` *`sigma-level`*

  where *`sigma-level`* the density level (in sigma) above which the map is searched for ligands
- `--fit-fraction` *`frac`*

  where *`frac`* is the minimum fraction of atoms in density allowed after fit [default 0.75]
- `--flexible`

  means use torsional conformation ligand search
- `--samples` *`nsamples`*

  `nsamples` is the number of flexible conformation samples [default 30]
- `--dictionary` *`cif-dictionary-name`*

  the file containing the CIF ligand dictionary description

One uses `findligand` like this:

```
$ findligand various-args ligand-pdb-file-name(s)
```

i.e. the example ligand pdb files that you wish to search for are given at the end of the command line.

11 Scripting Functions

12 More Scripting Functions

13 Scheme Scripting Functions

Concept Index

.

`.coot` 12
`.coot.py` 12

A

alternate conformation 36
anisotropic atoms 22
annotations 17
atom colouring 10, 19
atom info 19
atom label, brief 19
atom labeling 19
atom picking 44
atomic dots 24
auto-fit rotamer 34
average map 51

B

B-Factor, New Atoms 42
background colour 16
backrub rotamers 35
backups 13
backups, clearing 61
ball and stick 24
baton build 31
baton mode 31
big maps 51
blobs 56
bond thickness 20
bones 52

C

C-terminus 41
C\alpha representation 10
C\alpha symmetry 23
carbohydrates 29
chain id 45
change contour level 48
changing chain ids 45
changing the Refinement Map 30
chi angles 35
cif dictionary 28
citing coot 1
clashing residues 35
Clear Pending Picks 44
clearing backups 61
clipping 16
colour by chain 10
colouring, atoms 19
colouring, map 50
command line arguments 3
contouring, map 6, 49
coordinates format 18
coot droppings 60
crash 4
crash recovery 4
crosshairs 16

D

debugger 4
default refmac executable 43
delete 44
density line thickness 50
depth-cueing 9
dictionary, cif 2, 39
dictionary, ligands 3
difference map 49, 51
difference map colours 49
directory for save coordinates 11
Display Manager 10
DNA, ideal 42
DNA, mutating 37
docking sidechains 33
droppings 60
dual conformations 36

E

edit \chi angles 35
edit B-factors 19
edit occupancy 19
Electron Density Server 21
executing commands 12
export map 54
EYC-TIED 48

F

file-name filtering 11
fill partial residues 45
findligand 63
flip peptide 36
fragment direction change 33
frame rate 17

G

gdb 4
GLN and ASN B-factor Outliers 57
goose 21
guano 60
guile 12

H

helix placement . . . 42
hydrogens . . . 20

I

initialization file . . . 12

L

Least Squares Fitting . . . 25
ligand orientation . . . 40
ligand torsion angles . . . 36
ligand, overlay . . . 25
ligands . . . 38, 39
ligands, flexible . . . 39

M

mainchain . . . 33
mainchain torsions . . . 33
map box radius . . . 50
map changing (for refinement/building) . . . 30
map extent . . . 50
map line width . . . 50
map scrolling . . . 6
map transformation . . . 53
masks . . . 52
mean B-factor . . . 24
median B-factor . . . 24
merge molecules . . . 42
missing symmetry . . . 23
mmCIF dictionary . . . 28
modelling toolbar . . . 11
modified labels . . . 47
molecule centre . . . 46
molecule number . . . 8
Molprobity Tools . . . 57
monomers . . . 38
mouse . . . 2
mouse buttons . . . 5
moving ligands . . . 46
moving molecules . . . 46
Moving Zero Occupancy Atoms . . . 30
multi-mutate . . . 37
multiple coordinates files . . . 18
mutate . . . 37
mutating DNA . . . 37
mutating RNA . . . 37

N

NCS . . . 20
NCS averaging . . . 21
NCS edits . . . 43
negative contour levels . . . 50

O

OCA . . . 21
OpenGL . . . 9
Orientation Axes . . . 16
origin marker . . . 9
output . . . 17
overlaying ligands . . . 25
OXT atom . . . 41

P

packing diagram . . . 23
Patterson . . . 52
pepflip . . . 36
peptide restraints, planar . . . 29
PHS data . . . 49
PHS data format . . . 48
picking . . . 61
pink pointer . . . 16
planar peptide restraints . . . 29
planes . . . 27
polynucleotides . . . 42
post-manipulation-hook . . . 31
Print Sequence . . . 23
probe . . . 57
python . . . 12

R

Ramachandran plot . . . 55
Raster3D . . . 10
reading multiple pdb files . . . 18
recentring view . . . 15
recover session . . . 4
redo . . . 14
reduce . . . 57
reference manual . . . 1
reference structures . . . 2
refine single click . . . 27
refinement . . . 27
refinement weight . . . 27
refinement, rigid body . . . 30
refinement, simplex . . . 31
refining residues . . . 29
refmac . . . 43
refmac map colour . . . 43
refmac parameters . . . 43
refmac, default . . . 43
regularization . . . 27
render . . . 10
renumbering residues . . . 38
residue info . . . 19
resizing view . . . 62
restore after crash . . . 14
reverse direction . . . 33
rigid body fit . . . 30
RNA, ideal . . . 42
RNA, mutating . . . 37

rotamers 34
rotate/translate, manual 30
rotation centre 15
rotation centre pointer 16
running refmac 43
running SHELXL 44

S

save coordinates directory 11
scheme 12
screenshot. 9
scripting 11
Scroll 10
scroll wheel, map for 62
scroll, map contour change by 6
scroll-wheel 6
Secondary Structure Matching (SSM) 24
sequence view 23
`set-matrix` 27
set-rotation-centre 15
setting space group 22
sharpening, map 52
SHELX .ins 18
SHELXL 44
sidechains 33
simplex refinement 31
single click refine 27
skeleton updating 52
skeleton, missing 32
skeletonization 52
slab 16
sliding 15
slow computer 62
SMILES strings 38
space group names 18
space group operators 15
sphere refinement 28
startup dialog (state) 13
startup settings (python) 12
startup settings (scheme) 12
state 12
superposition 24
symmetry 22
symmetry operators 15

T

terminal oxygen 41
terminal residue 41
thickness of density lines 50
tooltips 8
torsion angles, ligand 36
torsion general 36
torsion restraints 27
torsions 35
trackball, virtual 6
traffic lights 28
translate molecule 47
translation, keyboard 6
translation, mouse 5
trimming atoms 53

U

undo 13
unexplained density 56
unit cell 16
`UNK` residue 30
unmodelled density 56

V

Validation Graphs 58
version number 8
view matrix 14

W

waters, finding 40
web page 4
weight, real space refinement 27
width, bonds 20
write map 54
writing PDBs 26

Y

yellow box 9

Z

z-rotation 6
zero occupancy 23
zoom 62
zoom, slider 7

Function Index

(Index is nonexistent)