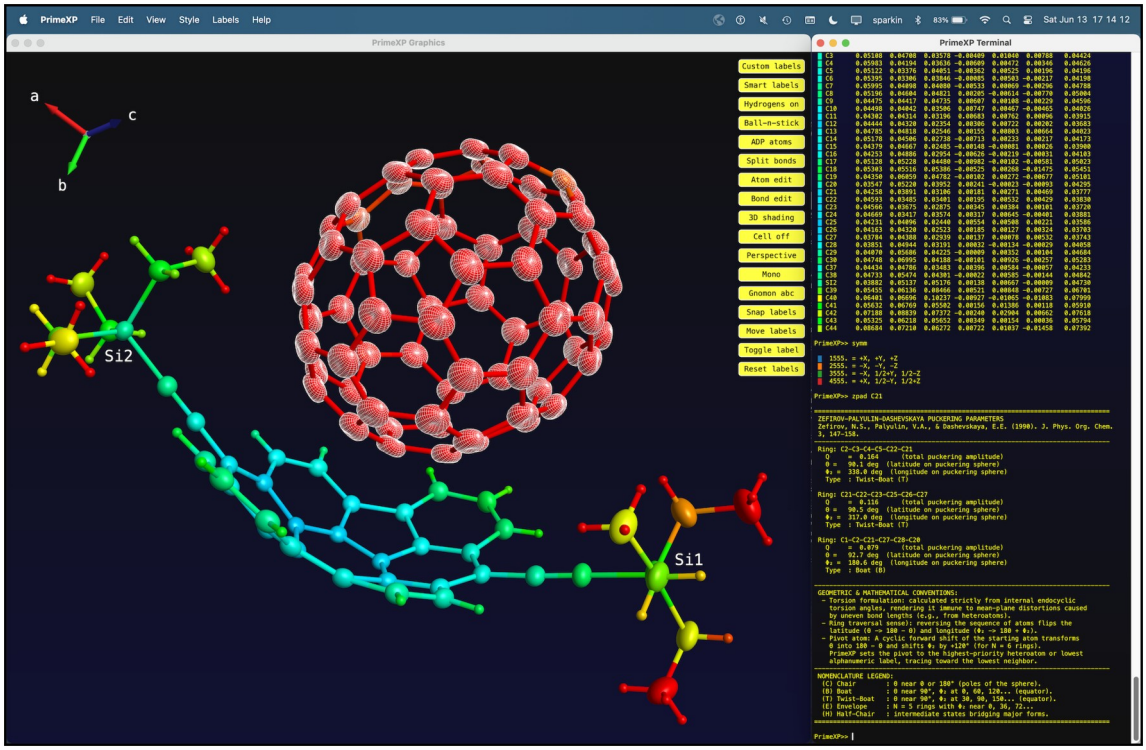
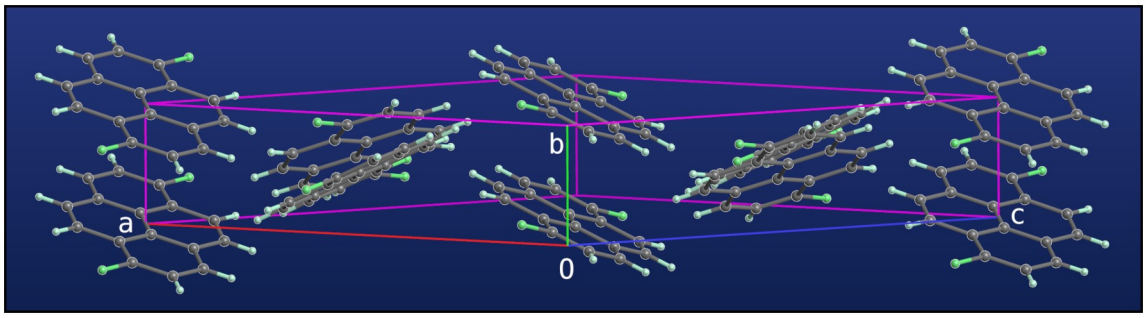
PrimeXP Reference Manual



PrimeXP © 2026 Sean R. Parkin
Department of Chemistry (retired)
University of Kentucky
Lexington, KY
40506–0055
s.parkin@uky.edu

Table of Contents

1. Introduction



PrimeXP [*Python* retro interactive model editor (inspired by) *XP*] is an interactive crystal structure editor with some intentional resemblance to the program that inspired it, George Sheldrick's *XP* (Sheldrick, 2008). It includes many of the same commands as *XP*, though some have altered and/or expanded meaning. There are also new commands that serve as enhancements or extensions. *PrimeXP* bridges the gap between *SHELXL* (Sheldrick, 2015) refinement and publication-quality diagrams, and also includes some analysis tools. It reads *SHELXL* *.res, writes *.ins files, extracts valid *SHELXL* files embedded within a .cif, and handles a few other common formats. It is not intended as a competitor to programs such as *ShelXle* (Hübschle *et al.*, 2011) or *Olex2* (Dolomanov *et al.*, 2009), but there is some overlap. Their model editing facilities are more modern and advanced than *PrimeXP*. For some complicated symmetry-related model manipulations, however, the command-line approach used by *XP* remains superior for experienced users in the same way that command-line computing can be dramatically more efficient than blundering through a graphical user interface (GUI). *PrimeXP* aims to replicate and build upon that type of functionality. *PrimeXP* is therefore much more closely related to *XP* than to *ShelXle* or *Olex2*. For example, similar to *XP*, *PrimeXP* is *not* a GUI to the *SHELX* suite. It does not call external programs like *SHELXL*. It does, however, provide a wide array of structure drawing and analysis tools that seek to match or surpass other small-molecule molecular graphics programs. Suggestions for improvements, new features, and especially bug reports, are welcome.

PrimeXP is built using *Python*, *PyQt6*, and *PyOpenGL*, using *NumPy*, *scikit-image*, *SciPy*, and *NetworkX* for geometric and topological mathematics, graphics, etc., and a fair bit of carefully vetted AI (mainly *Gemini*). The *PrimeXP* visuals are highly customizable. Virtually all control parameters, from default line widths to atom colours etc. can be individually configured. Any that were

accidentally left as hard-coded values will be changed to configurable parameters upon request. All the graphics are handled by *OpenGL 4.1*, enabling real-time rendering of features such as atomic displacement ellipsoids, a variety of atom and bond styles, electron density in 2D and 3D, 'void' space analysis and visualization, atom label collision avoidance, etc.

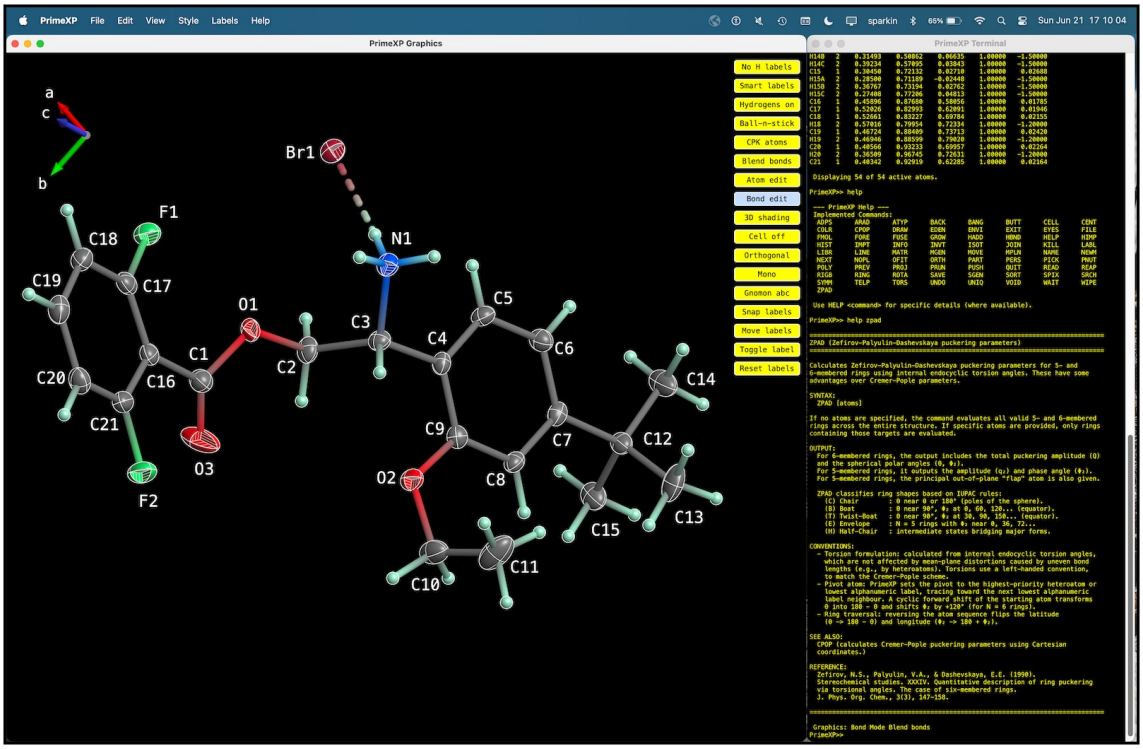


Figure 1.1. The *PrimeXP* GUI windows, graphics on the left, terminal on the right.

The *PrimeXP* graphical interface (Figure 1.1.) shows the structure of a bromide salt of an etoxazole metabolite (Mohan Kumar *et al.*, 2026) with 50% displacement ellipsoids for non-H atoms, and the terminal emulator (above right). The axis direction gnomon in the upper left corner has a 3-state toggle button (off / on /on + axis labels). The column of yellow buttons on the right side of the graphics window are an homage to George Sheldrick's classic *XP* program. They can be toggled off and on via the 'View' menubar dropdown. The *PrimeXP* code is modular, so it would be straightforward (though a bit tedious) to re-tool these as rows of drop-down menus, as per modern GUIs.

1.1. *PrimeXP* features

The program uses a hybrid interface, combining an interactive 3D graphics window with a *bash*-like terminal emulator, which should be familiar to experienced crystallographers (ok, the old fogeys maybe). Key features include:

- **Advanced rendering:** Customisable representations including wireframe, stick, ball-and-stick, anisotropic displacement ellipsoids, with or without major ellipses and/or cutout octants, and Bürgi-style 'peanut' plots (Hummel *et al.*, 1990).
- **Symmetry generation:** Real-time application of space-group operators and cell translations to expand the asymmetric unit and examine packing motifs.
- **Rigid-body analysis:** Translation-Libration-Screw (TLS) for analyzing rigid molecular fragments.
- **Geometric analysis:** Tools for calculating mean planes, torsion angles, interactive distance/angle measurements, and Cremer-Pople (Cremer & Pople, 1975) and Zefirov-Palyulin-Dashevskaya (Zefirov *et al.*, 1990) ring-puckering parameters.
- **Hydrogen modelling:** Automated hydrogen atom addition for various hybridization states (e.g., *sp*2, *sp*3) using standard riding model geometries. This is more sophisticated than HADD in *XP*, but much less so than the HFIX facilities in *SHELXL*.
- **Export:** Generation of high-quality, high-resolution diagrams suitable for publication, orthogonal coordinate files, or newly edited *.ins files etc. for further analysis.

1.2. A few *PrimeXP* examples

The program currently has 29 different 'ATYP' atom types for rendering of ellipsoids and spheres, with or without various kinds of shading/octant-cutout embellishments. It also includes anisotropic displacement parameters in the style of Hans-Beat Bürgi's *PEANUT* program plots. Unlike *XP*, TELP in *PrimeXP* is fully 3D-interactive, as are the 'peanuts' (*vide infra*). Figure 1.2. shows a traditional ellipsoid *ORTEP*-style (Johnson, 1965) plot for a Ru(II)-cymene sandwich complex (Pokharel *et al.*, 2026), drawn in 'TELP' mode by *PrimeXP*.

Other styles are also possible, as shown in Figure 1.3., which depicts the same projection as Figure 1.2., but with 3D-shaded ellipsoids and a blue gradient background. Ellipsoid probabilities from 1% to 99% are possible. Colours may be specified using 6-digit hexcodes (~16 million colours), 3-digit hexcodes (~4 thousand colours), or 148 named 'web' colours (see Appendix 2). The label colours may be set individually (via colour name or hexcode), but default to black or white, automatically switching based on the overall luminosity of the scene.

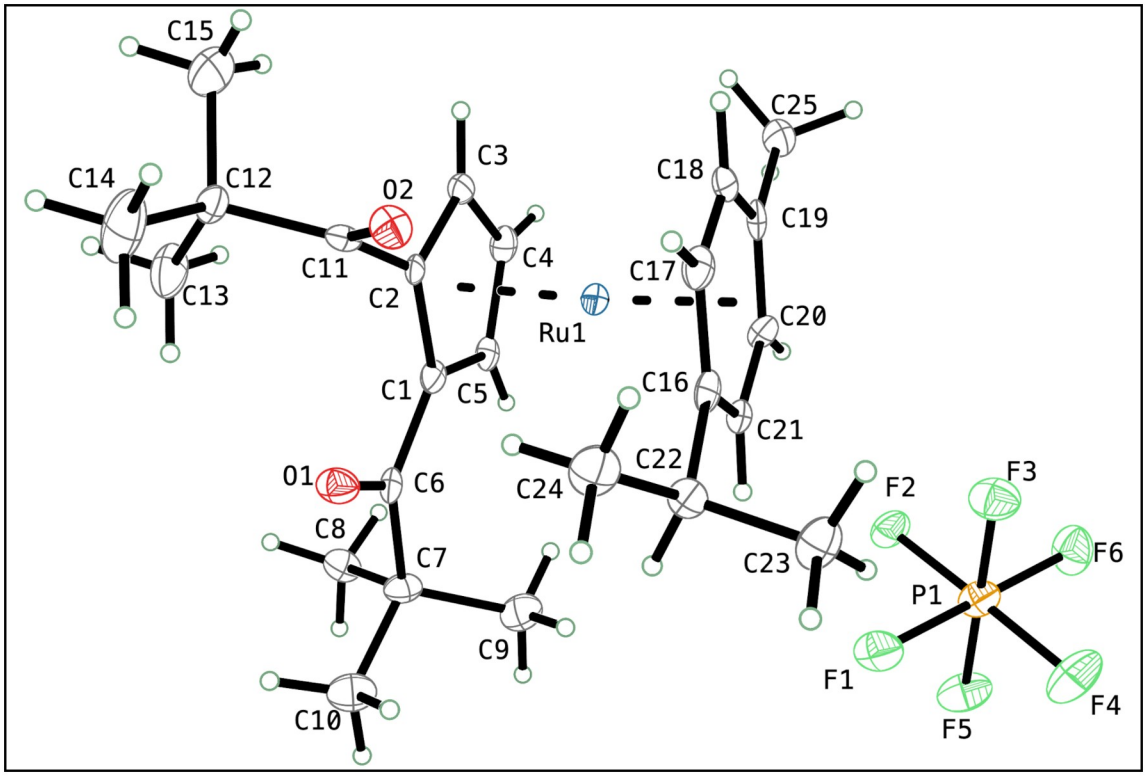


Figure 1.2. A traditional ellipsoid plot produced by *PrimeXP*.

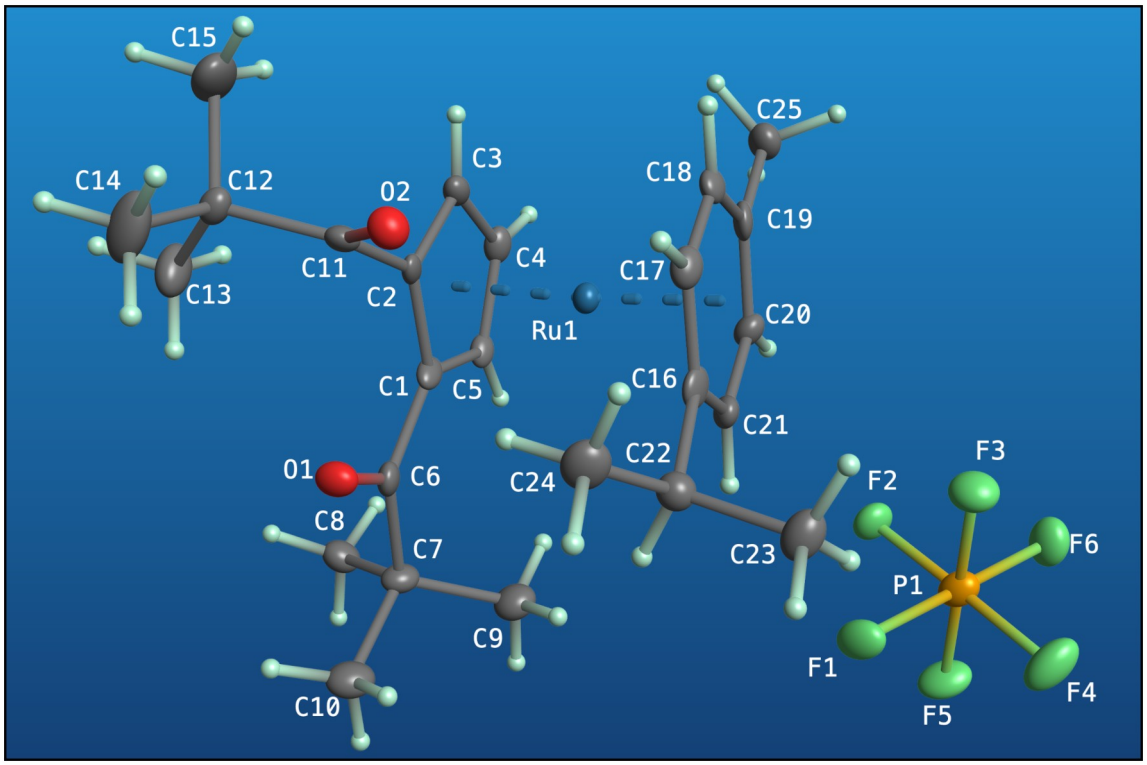


Figure 1.3. An ellipsoid plot with 3D-shaded atoms and a gradient background.

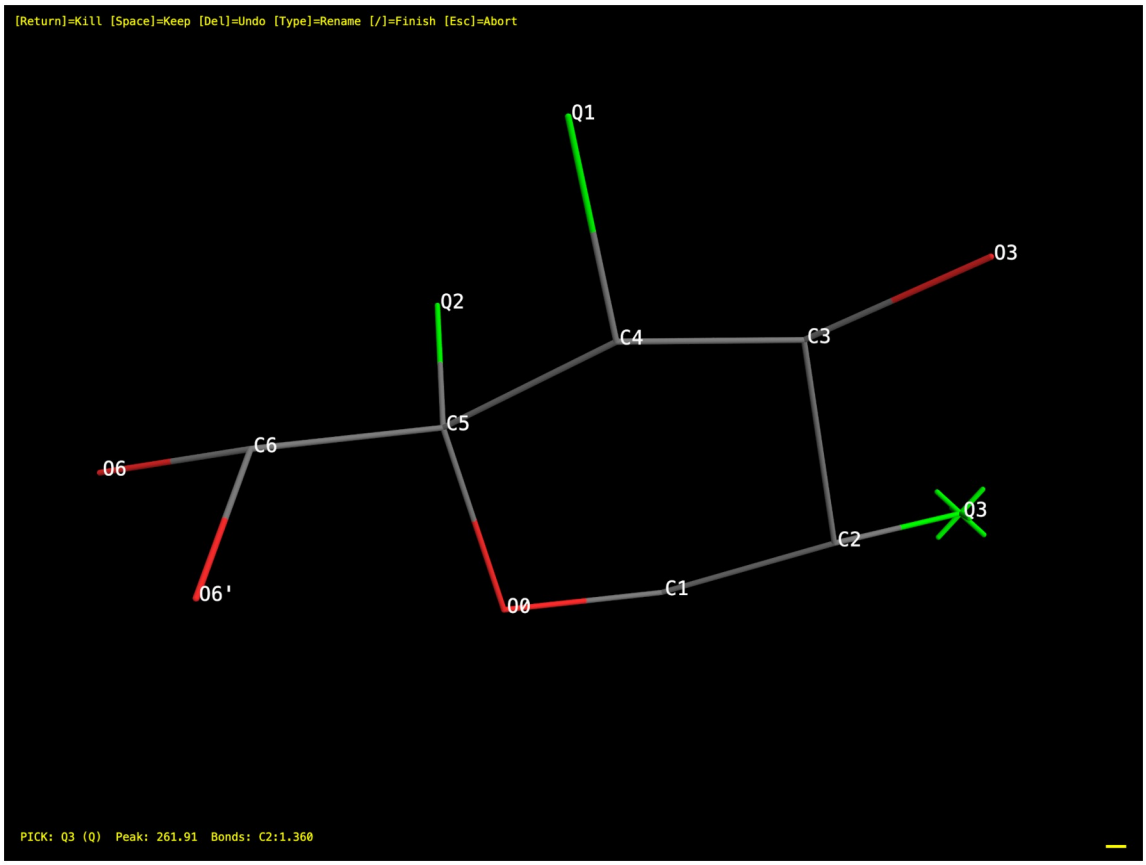


Figure 1.4. A view of the interactive model editor in *PrimeXP*.

Figure 1.4. is of the monosaccharide sorbose, which has a disordered hydroxyl group, undergoing atom assignment using PICK. Model editing using PICK is similar to its counterpart in the original *XP*. In *PrimeXP*, however, PICK is fully 3D interactive, and the PICK sequence can be interrupted by mouse clicks on any visible Q-peak or atom. Having said that, PICK is included in *PrimeXP* mainly to retain the model building and structure editing functionality of *XP*. Programs like *ShelXle* and *Olex2* are objectively more flexible for model editing.

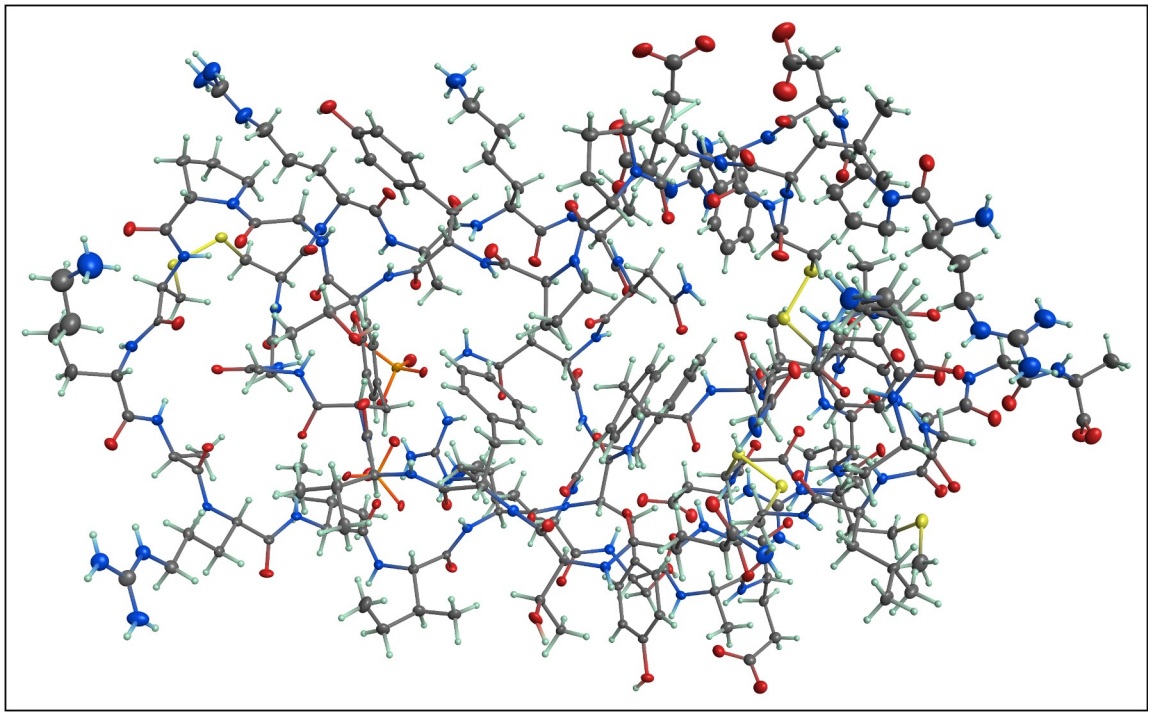


Figure 1.5. An ellipsoid plot of the 1.1 Å resolution structure of a small protein, BPTI, as rendered by *PrimeXP*.

Although it was not written with protein crystallographers in mind, *PrimeXP* is fully capable of rendering protein structures, as shown in Figure 1.5, which depicts a 1.1 Å resolution structure of bovine pancreatic trypsin inhibitor, BPTI (Parkin *et al.*, 1996), drawn after removal of its attendant water molecules. Atomic resolution structures of concanavalin A (Parkin *et al.*, 1996), sperm-whale myoglobin, and the industrial commodity enzyme subtilisin also render perfectly well as single molecules or as packed assemblies.

The RING/X command automatically places a dummy 'X' atom at the centroid of each ring (default 3-8 membered rings, but configurable), as in the C_{70} buckyball, buckybowl complex (Gao *et al.*, 2020) shown in Figure 1.6. In contrast to CENT/X in *XP*, that amounts to 46 dummy atoms placed by a *single* 'RING/X' command, without having to explicitly specify *any* ring atom names (the *XP* command CENT/X is also available, of course).

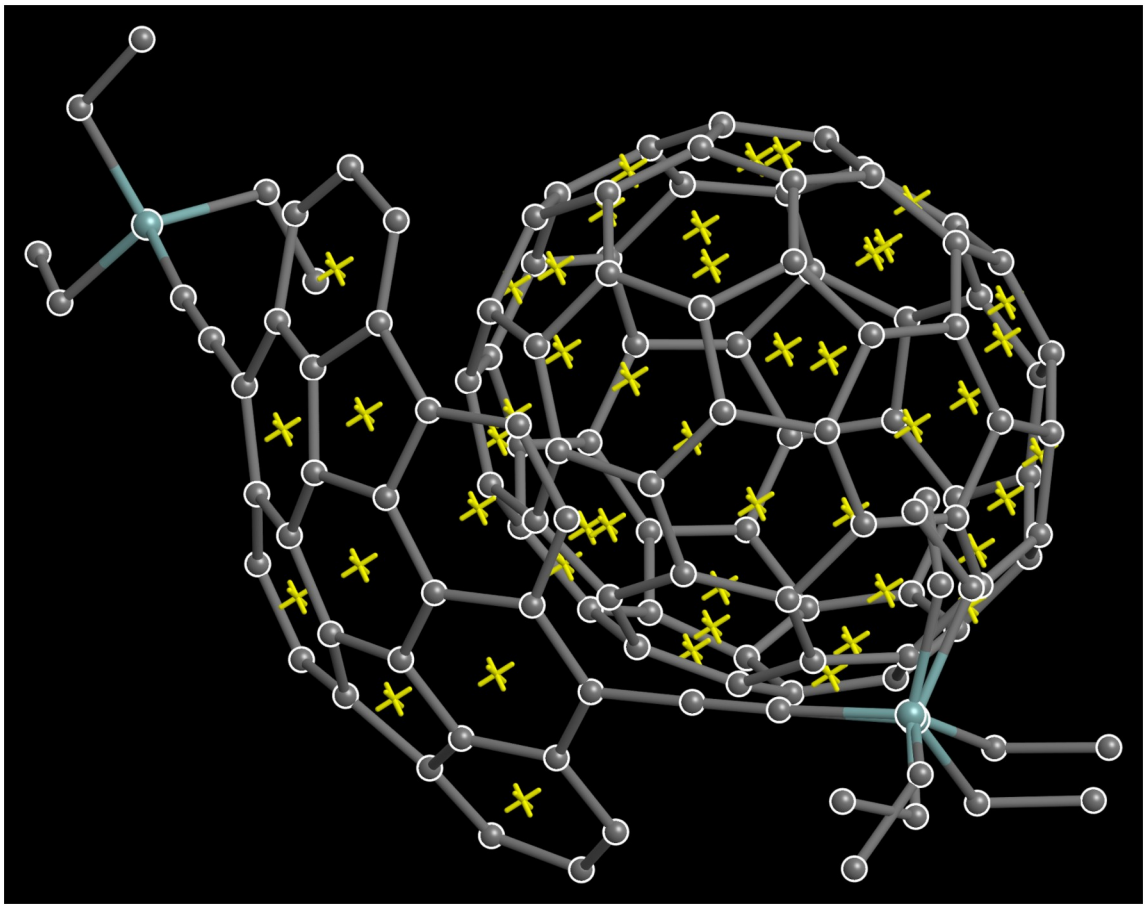


Figure 1.6. Automatic placement of 46 dummy atoms in a C_{70} fullerene complex.

Any of seven styles of bond representation (solid lines, open lines, solid dashes, open dashes, thin solid lines, thin dashes, and dots) may be used to connect real and/or dummy atoms (e.g., see Figures 1.2. and 1.3. above of the ruthenium-cymene pi-complex). *PrimeXP* includes variants of the JOIN command, such as 'star-join', 'nearest-neighbour join', 'pi-stacking join', which are used to rapidly add multiple connections to ellipsoid and/or packing plots.

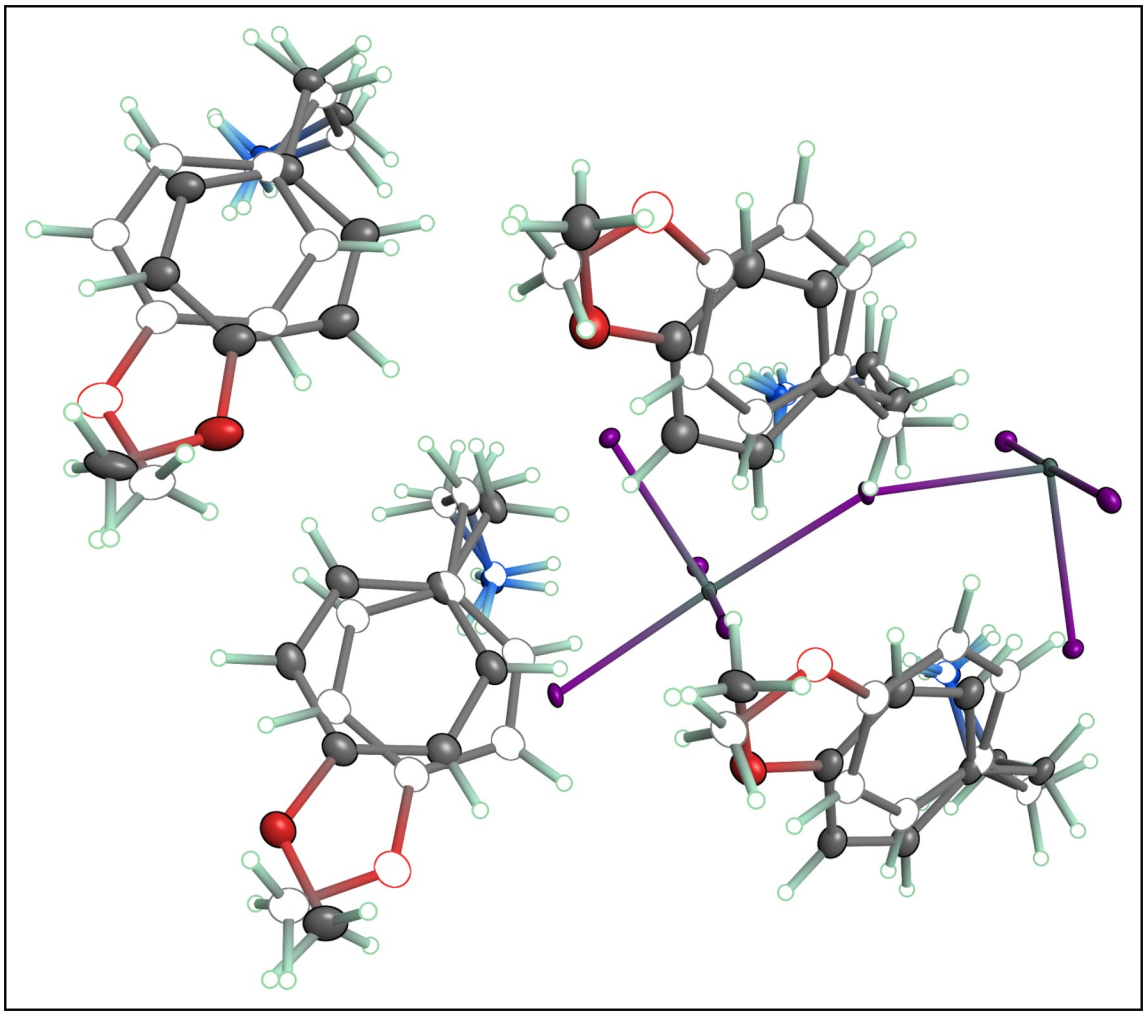


Figure 1.7. Depiction of disorder in a hybrid organic-inorganic perovskite structure by *PrimeXP*.

Disorder visualization in *PrimeXP* is flexible and can be shown in a number of ways. One such view, of the asymmetric unit of an extensively disordered di-periodic perovskite-like structure (Hossain *et al.*, 2024), is given in Figure 1.7. The non-disordered and major occupancy groups are drawn with 3D shading, while minor disorder parts are shown as boundary ellipses. Any of the available 'atom types' (ATYP command) are allowed. Note also the colour blending in the bonds. One of the yellow click buttons cycles through split-coloured bonds, blended bonds, and grey/black (depending on the scene).

There are also numerous ways to construct packing plots in *PrimeXP*. Figure 1.8. shows a perspective view down the *b*-axis of a hybrid organic-inorganic di-periodic perovskite-like structure [same structure as the disorder plot (*vide supra*), but with the minor disorder removed]. Orthogonal (parallel) projections are also available at the click of an on-screen button.

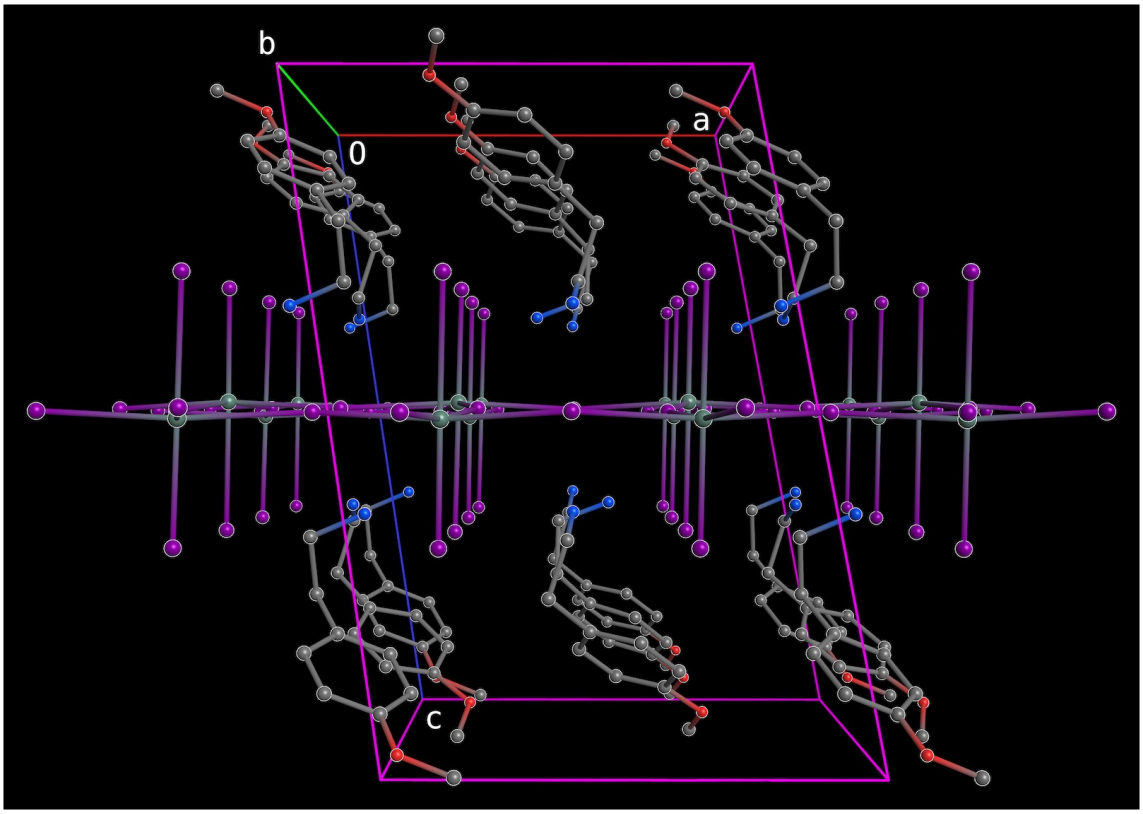


Figure 1.8. A packing view down the *b*-axis of a hybrid organic-inorganic perovskite structure.

Real-time translation-libration-screw (TLS) thermal-motion analysis following the methods described by Schomaker and Trueblood (1968), with tensor visualisation, is available via the LIBR command. The translation (T) and libration (rotary oscillation, L) tensors can be represented by 3x3 symmetric positive-definite matrices, and so may be visualized as ellipsoids, similar to atomic displacement ellipsoids. The screw (S) tensor, however, represents the coupling of translation and rotation, which requires an *asymmetric* 3x3 matrix. To visualize the S-tensor as an ellipsoid requires singular-value decomposition. The SVD process, however, does not preserve information about the handedness of screw motion. Nonetheless, that chirality is still present in the S-tensor. In *PrimeXP* the handedness is analytically recovered by projecting the screw pitch ($p = n^{T}Sn$) along the surface normal. Default colours are red (r, right handed) and lemon-yellow (l, left handed). Full numerical results, error trapping, and warnings are written to the terminal window, and an interactive graphical dashboard is displayed in the graphics window, Figure 1.9.

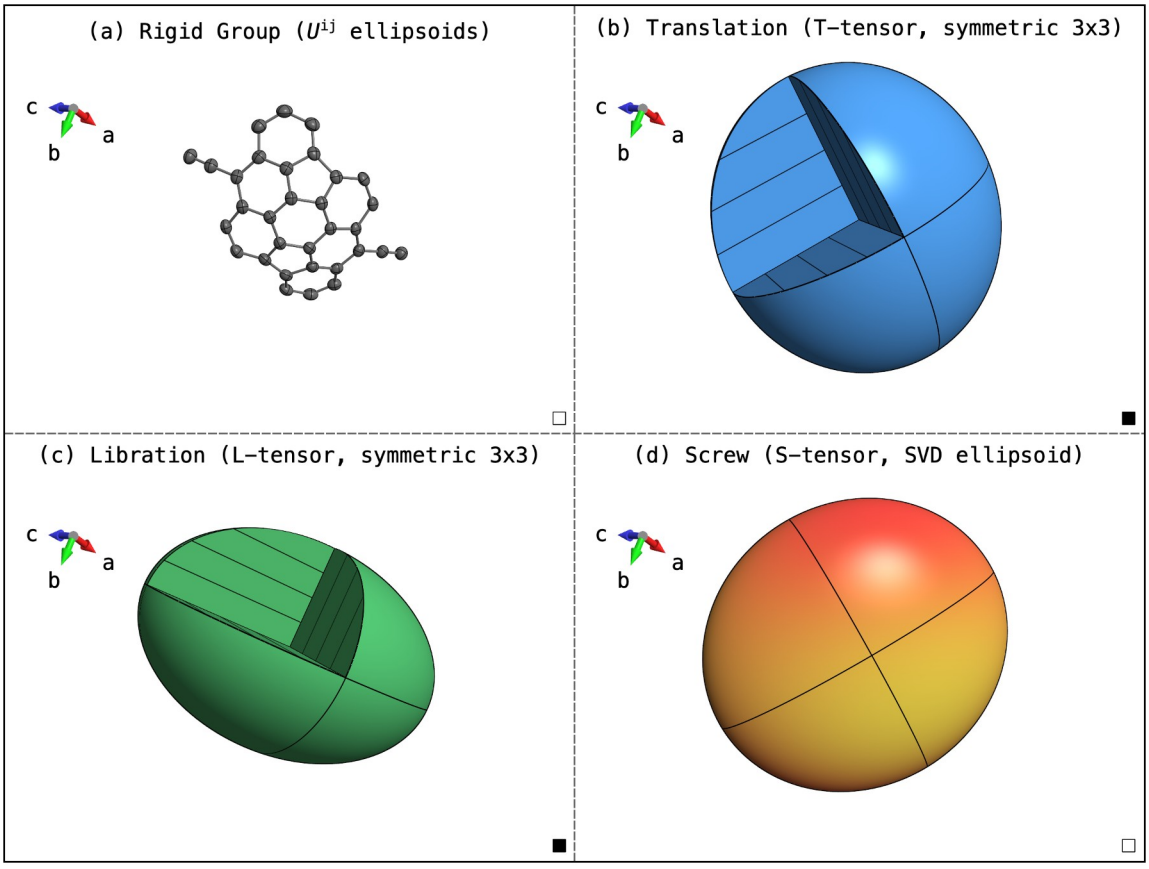


Figure 1.9. Graphical depiction of translation-libration-screw, TLS-tensors in *PrimeXP*.

Old-style 2D electron-density contour plots, similar those made by the EDEN command in *XP* or 'Contour' in *Platon* (Spek, A. L., 2009), are possible if suitable *.fcf files in 'LIST 6' format are available to *PrimeXP*'s version of EDEN. In Figure 1.10, the blue contours represent experimental electron density (Fobs) for the carbon and fluorine atoms. The model did not have hydrogen atoms included, so they show up as prominent positive (green) contours in the difference map (Fobs - Fcalc) at the expected positions. The small red contours indicate where the difference map happened to be slightly negative. These 2D electron density views may be zoomed in and out by mouse action.

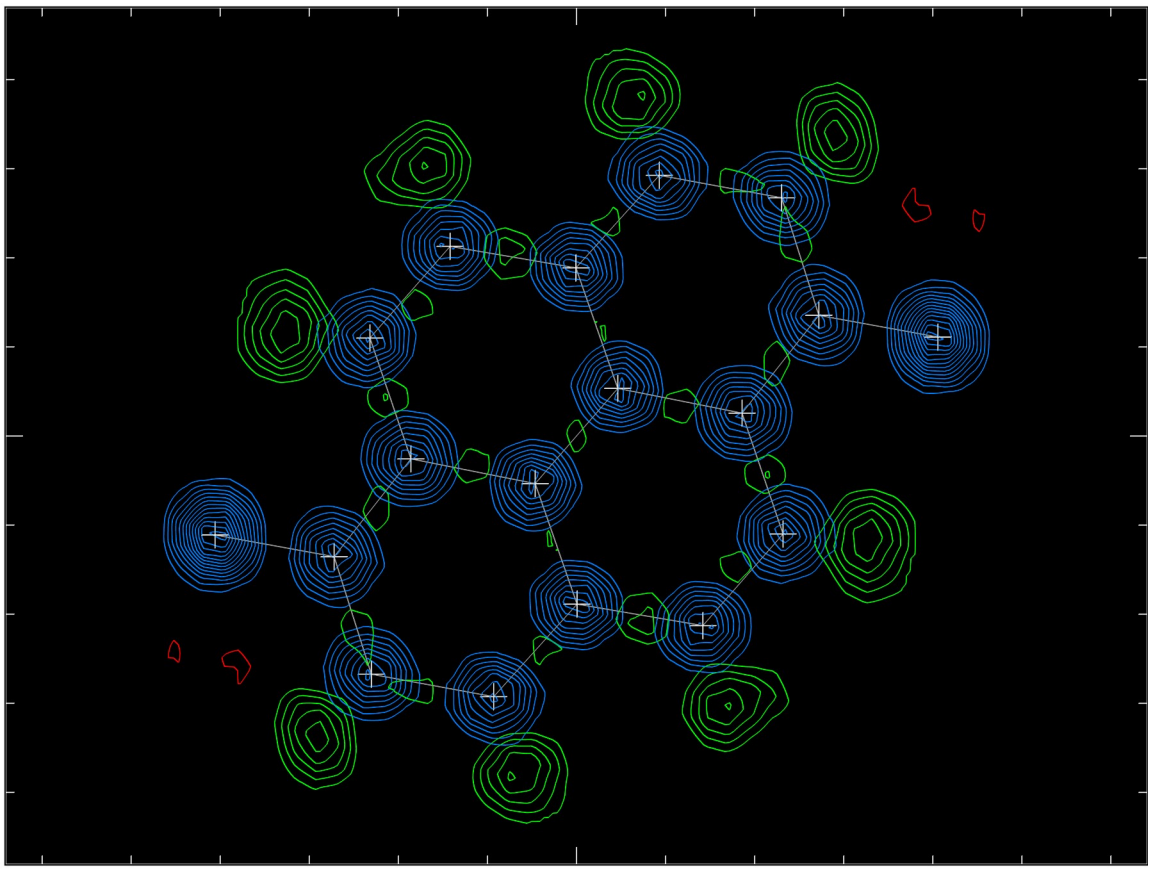


Figure 1.10. A 2D contour plot of electron density and difference-map density in a di-fluorinated pyrene.

These 2D contour plots are currently fairly rudimentary, but could be made more sophisticated if there is sufficient interest. Probably more useful nowadays than flat electron-density contour maps are interactive three-dimensional representations. Figure 1.11 shows such a plot of the same molecule, again showing observed electron density in blue and difference-map electron density in

green. In the 3D view, the electron density may be zoomed in/out and rotated in real time. Invoking the EDEN command also creates a conditional four-state click button on the graphical display window that cycles through different visualisation styles [cage contours, diffuse clouds, clouds and cages superimposed (as shown below), and off].

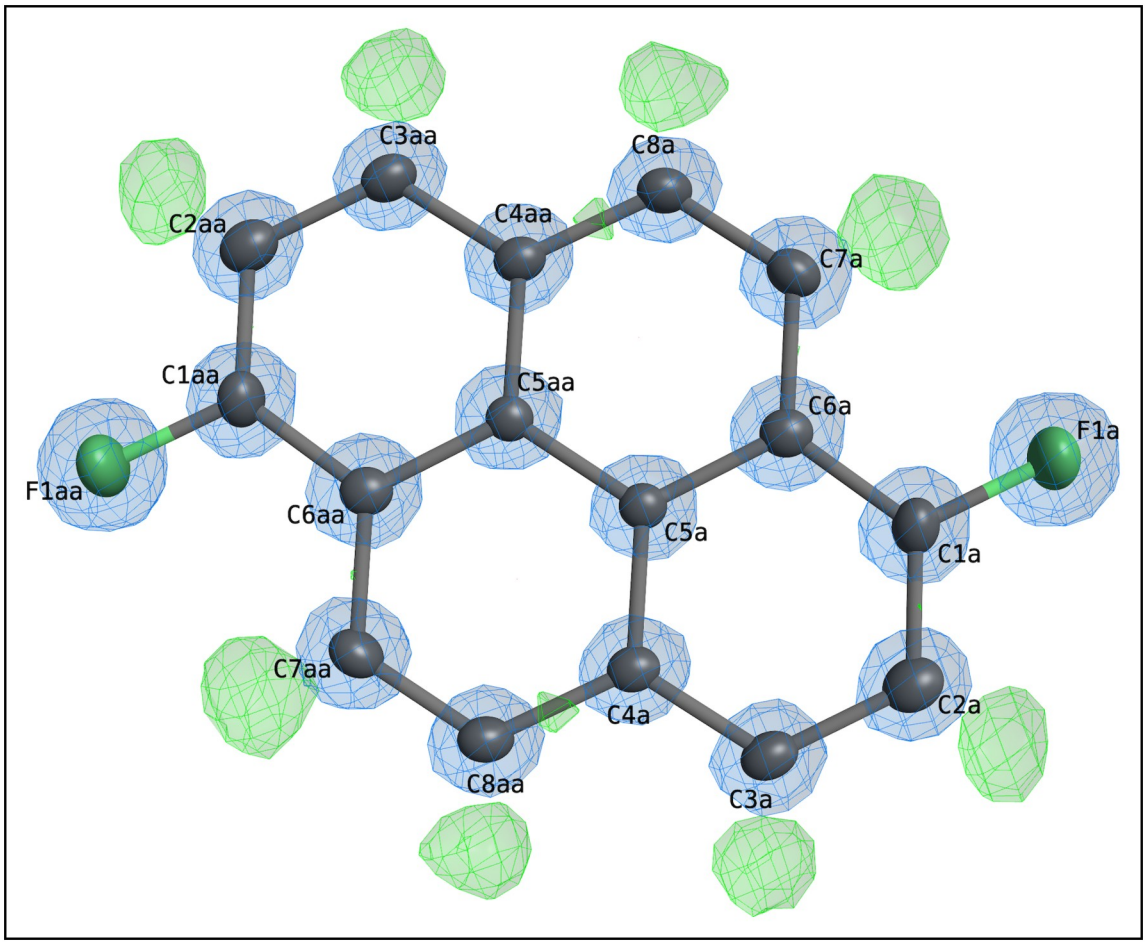


Figure 1.11. A 3D cage/cloud-contour plot of electron density and difference-map density in a di-fluorinated pyrene.

Anisotropic displacement parameter plots in the style of Bürgi's *PEANUT* program are also possible in *PrimeXP* via the PNUT command. Both RMSD (root-mean-square-displacement) and MSD (mean-square-displacement) peanuts, as either mesh-contours, 3D-solid rendered (as shown in Figure 1.12), or both (mesh superimposed on 3D solid) are possible. As with all other graphical structure depictions in *PrimeXP*, these are fully 3D interactive and can be freely mixed with other atom visualization styles, globally, by group selection, or on a per-atom basis. Difference 'peanuts' might follow if there is sufficient interest.

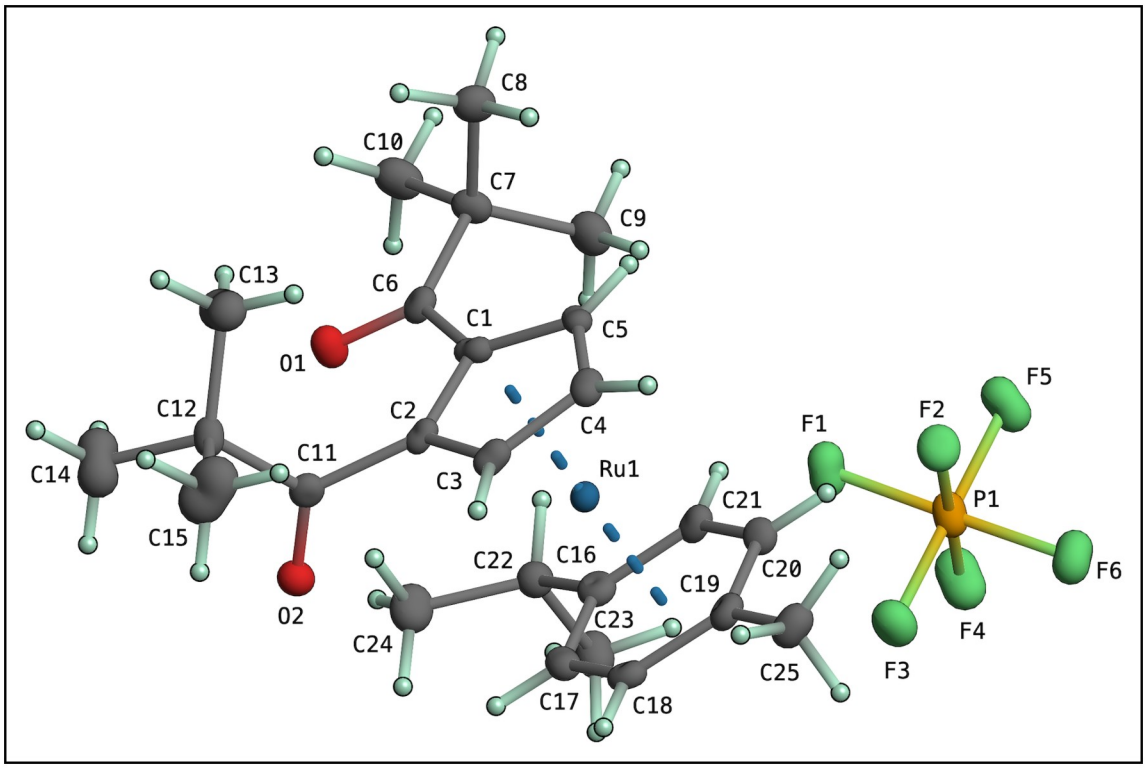


Figure 1.12. An example of RMSD *PEANUT*-style representation of anisotropic displacement parameters.

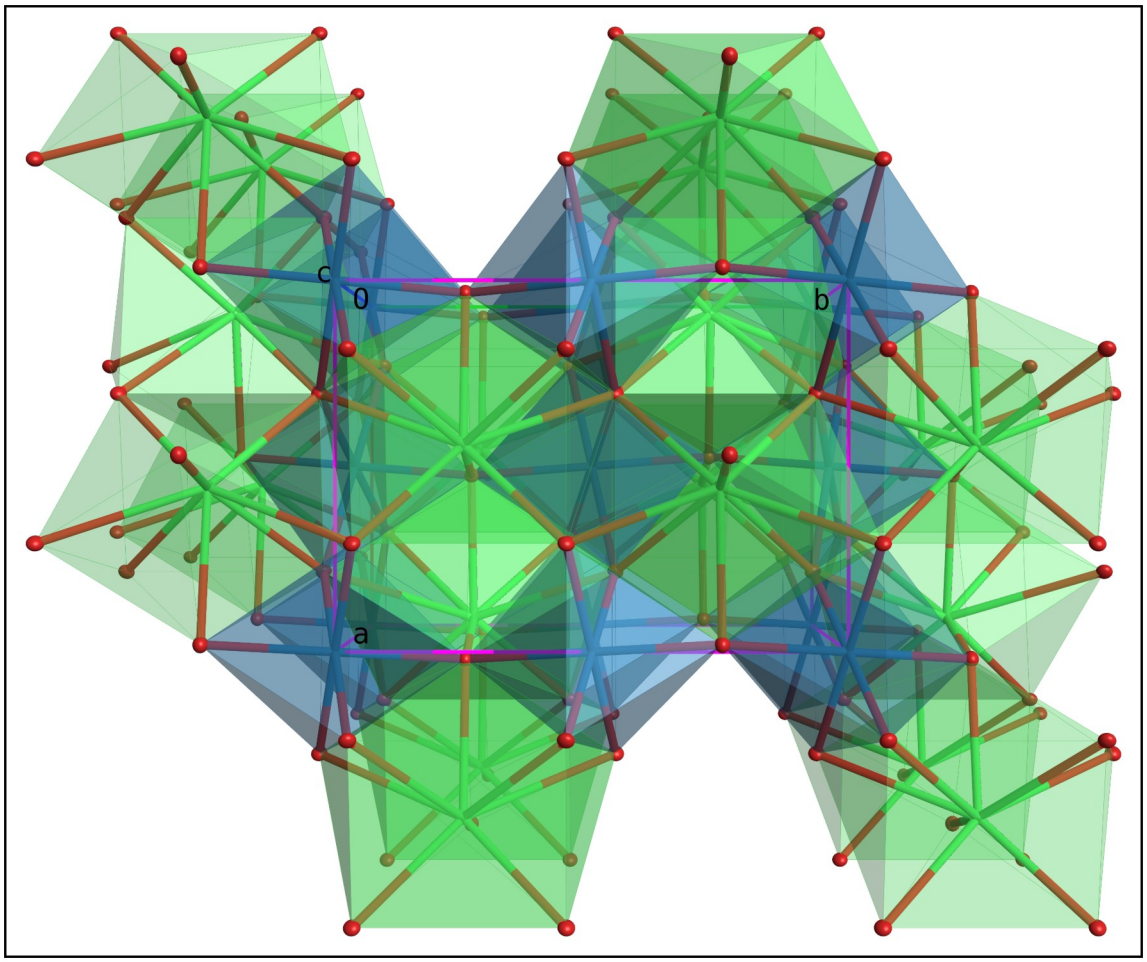


Figure 1.13. Convex polyhedra around calcium (green) and ruthenium (blue) in a calcium-ruthenium oxide structure.

Graphical representations of coordination geometries by polyhedral convex hulls around selected central atoms (Figure 1.13.) are implemented via the POLY command. The program identifies bonded ligands, calculates the 3D faces connecting them, and reports the coordination number, number of faces, and calculated volume for each generated polyhedron. The resulting polyhedra are rendered with semi-transparent faces and wireframe edges along with the atoms and bonds present in the structure. As with all other 3D view modes, these are fully rotatable and zoomable via mouse action.

Solvent-accessible channels or other regions of 'empty' space within a structure can be made visible using the VOID command. The colour, void probe radius and grid resolution are user definable. The terminal window gives an estimate of the fractional volume of the unmodelled 'void' space. For the structure in Figure 1.14 (Parkin & Behrman, 2011), the channels (visible as semi-transparent light-blue columns) were filled with solvent (acetone), thoroughly disordered about the $\bar{3}$ roto-inversion axis of space group ($R\bar{3}$). These disordered solvent molecules were removed from the model prior to drawing the diagram below.

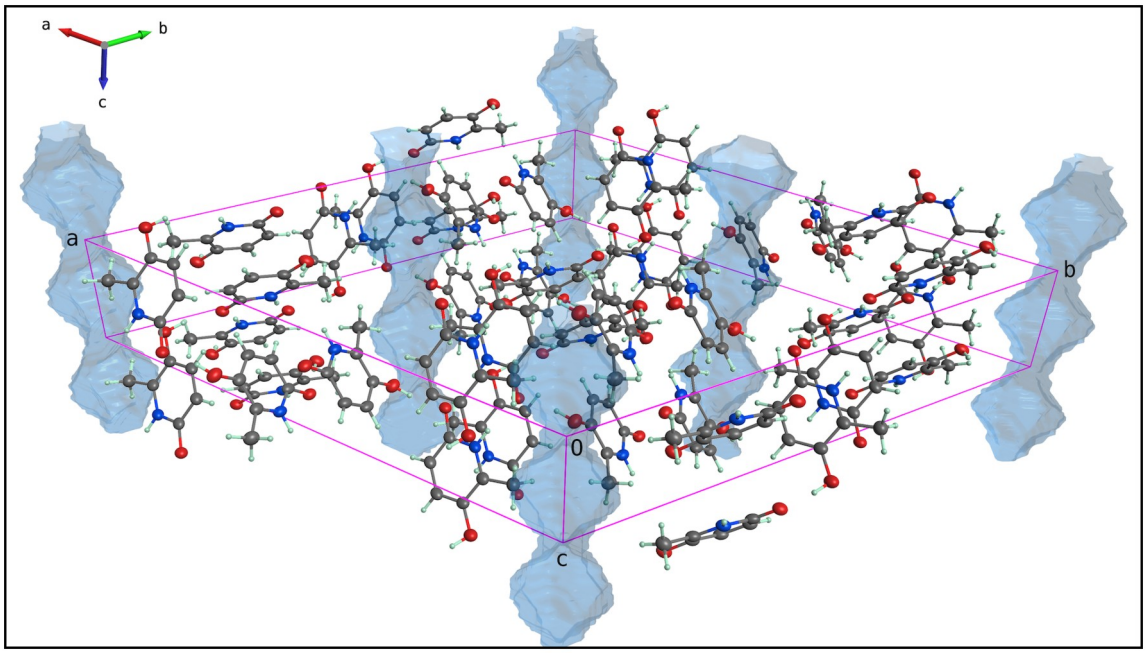


Figure 1.14. Solvent-accessible channels (light-blue columns) in a trigonal structure of 5-hydroxy-6-methyl-2-pyridone.

Molecules can be coloured according to their symmetry codes, as shown for 6-chloro-2,5-dihydroxypyridine (Parkin & Behrman, 2009) in Figure 1.15. These are activated either by mouse click in the GUI, the menubar, or via the COLR terminal command. The particular mapping of colours to symmetry operations is accessible via the SYMM command in the terminal window, which shows coloured blocks associated with each symmetry code and operation.

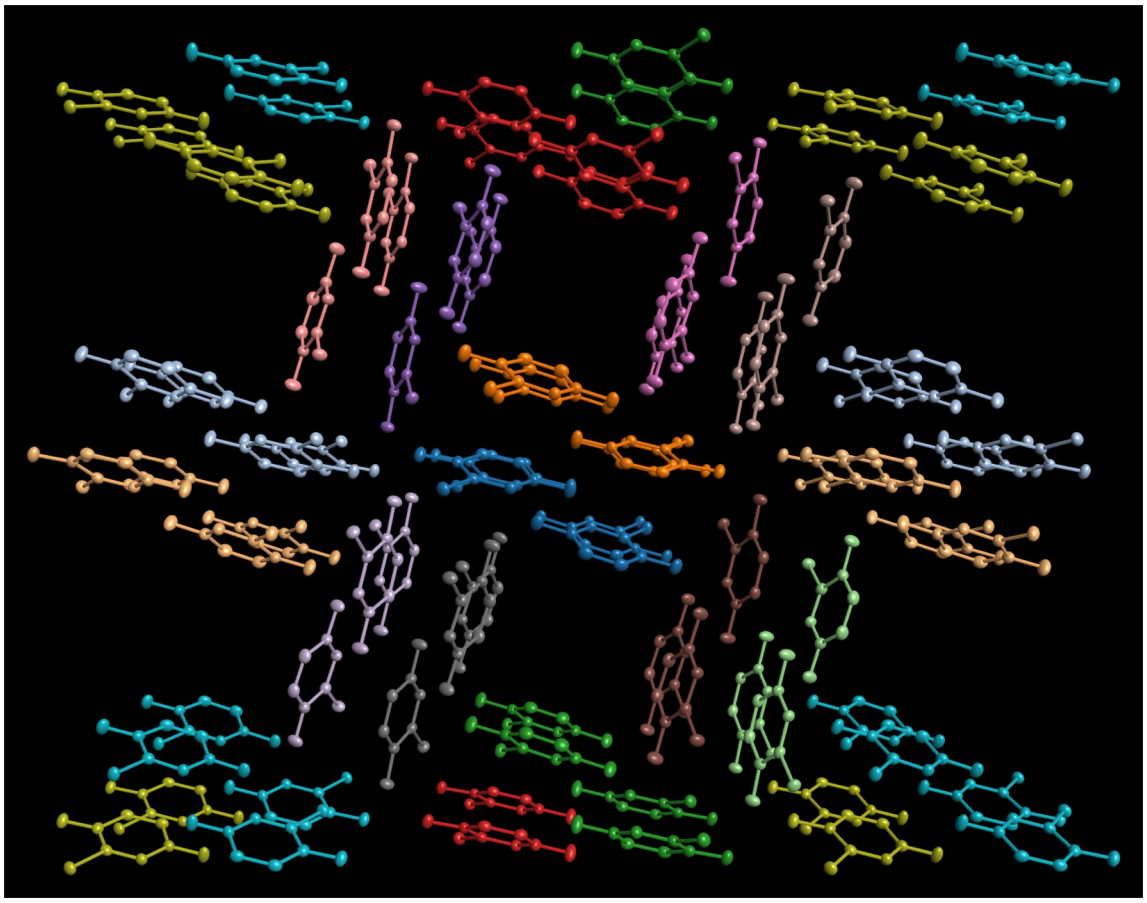


Figure 1.15. Molecules of 6-chloro-2,5-dihydroxypyridine, coloured according to their symmetry code.

Atoms may also be coloured on a rainbow spectrum from violet to red based on their U_{eq} values. An example, for a trigonal cobalt complex (Roecker & Parkin, 2024) is shown in Figure 1.16. Smaller U_{eq} being blue-shifted and larger U_{eq} being shifted into the red (the thresholds are configurable). In a similar vein, bonds can be coloured according to how well the atom U^{ij} obey Hirshfeld's rigid bond test, from blue (good fit) to red (poor fit).

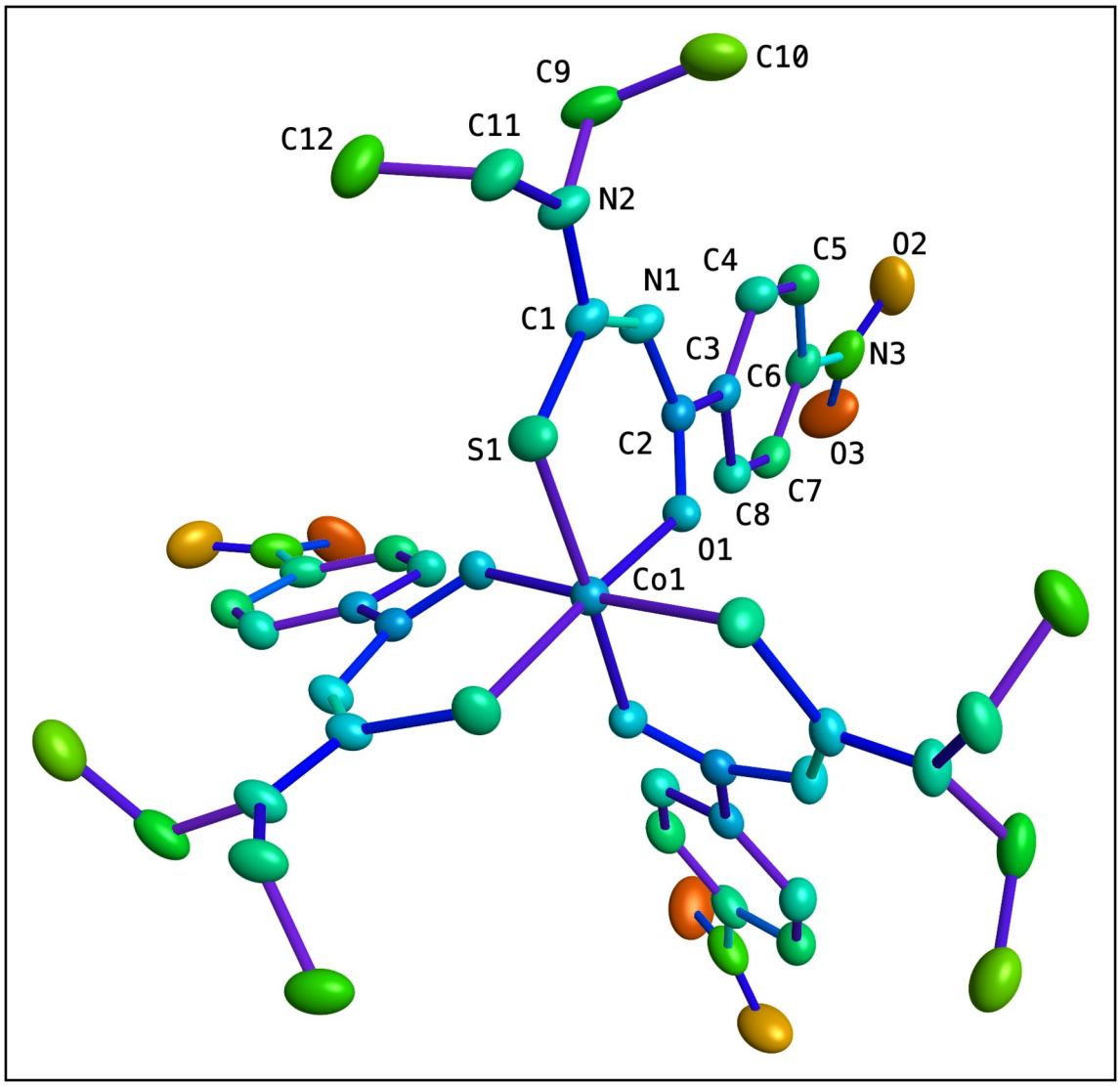


Figure 1.16. Molecules coloured according to their U_{eq} values.

The Δz^2 values from Hirshfeld's rigid-bond test can be extended to non-bonded atom pairs (RIGB command). Plotting these values for the current atom list on a square matrix (RIGB/H) generates a 'heatmap', which visualizes potential through-space atomic motion. A similar, but non-graphical, summary was available as a triangular matrix of numbers in the output logs of the *THMA11* program of Schomaker and Trueblood.

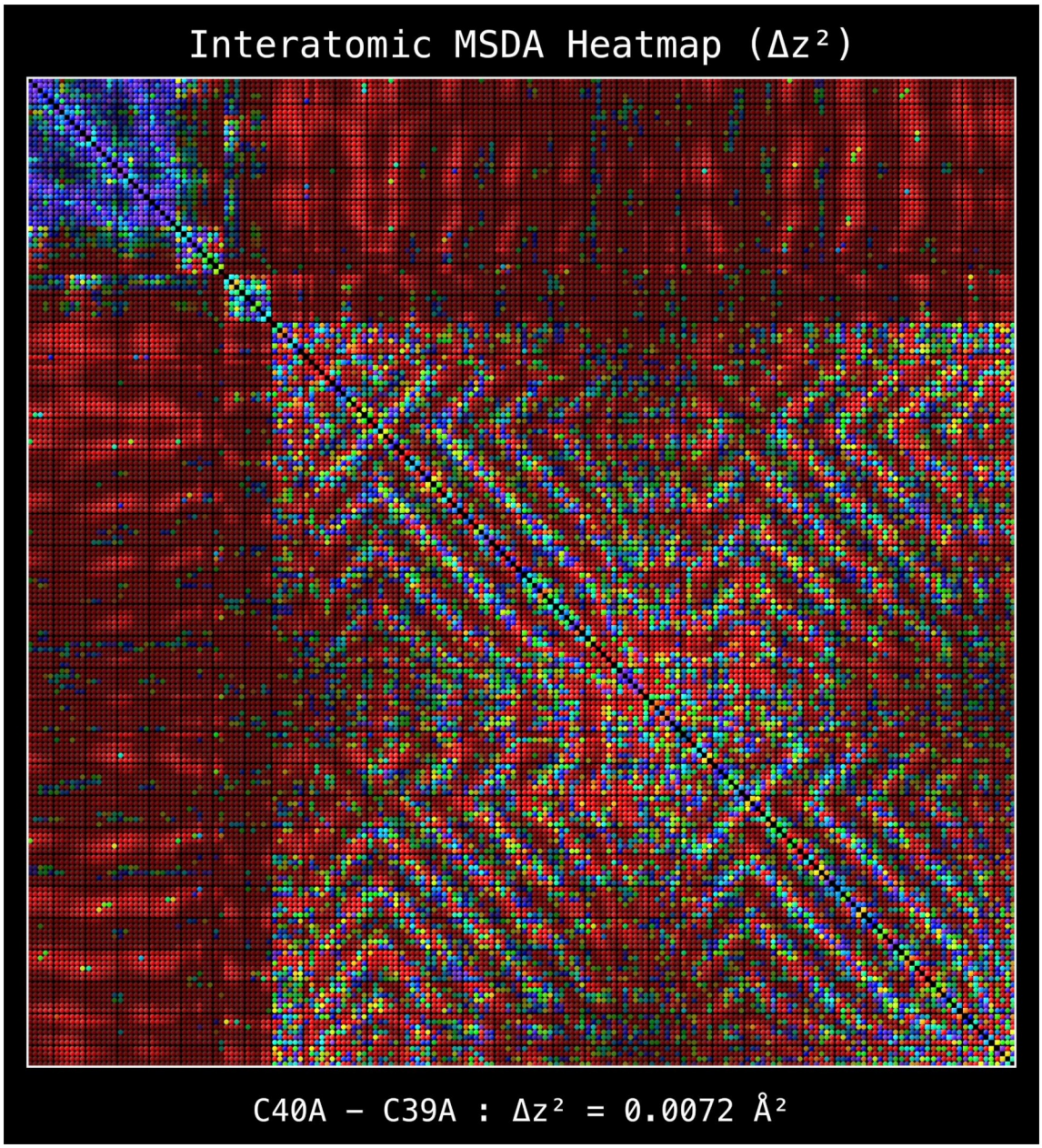


Figure 1.17. A Hirshfeld rigid-body inspired heatmap of Δz^2 values for both bonded and non-bonded pairs of atoms.

In *PrimeXP* heatmaps, bonded atom pairs are represented by squares and non-bonded pairs by circles. The colours are as per the U_{eq} values, but can be dimmed as a function of interatomic separation (4-state toggle covering distant, near, no dimming, off). The example in Figure 1.17 is from a structure with a pair of overlapping C_{70} fullerene cages (Gao *et al.*, 2020) and other junk (also used for RING/X in Figure 1.6). Clicking on a heatmap cell pops up a text box showing which atoms are involved and their particular Δz^2 value.

1.3. Availability and implementation

The modular architecture of *PrimeXP* separates graphical routines, command parsing, crystallographic symmetry, and mathematics. *PrimeXP* was developed on MacOS, and runs on both Linux (thanks to Paul Boyle and Steven Kelley for extensive testing and feedback) and on Windows, completely unmodified (thanks to Sean Gilpatrick of the Awuah group at UKy). Although *PrimeXP* will probably always be a work in progress, and new features are still being added, it is now available upon request. The Mac version has a clickable launcher that bundles *Python* and the external libraries. Linux and Windows currently require an established *Python* environment with the necessary libraries installed. Enquire by e-mail to s.parkin@uky.edu.

---

2. Cookbook

2.1. Atom and bond rendering styles

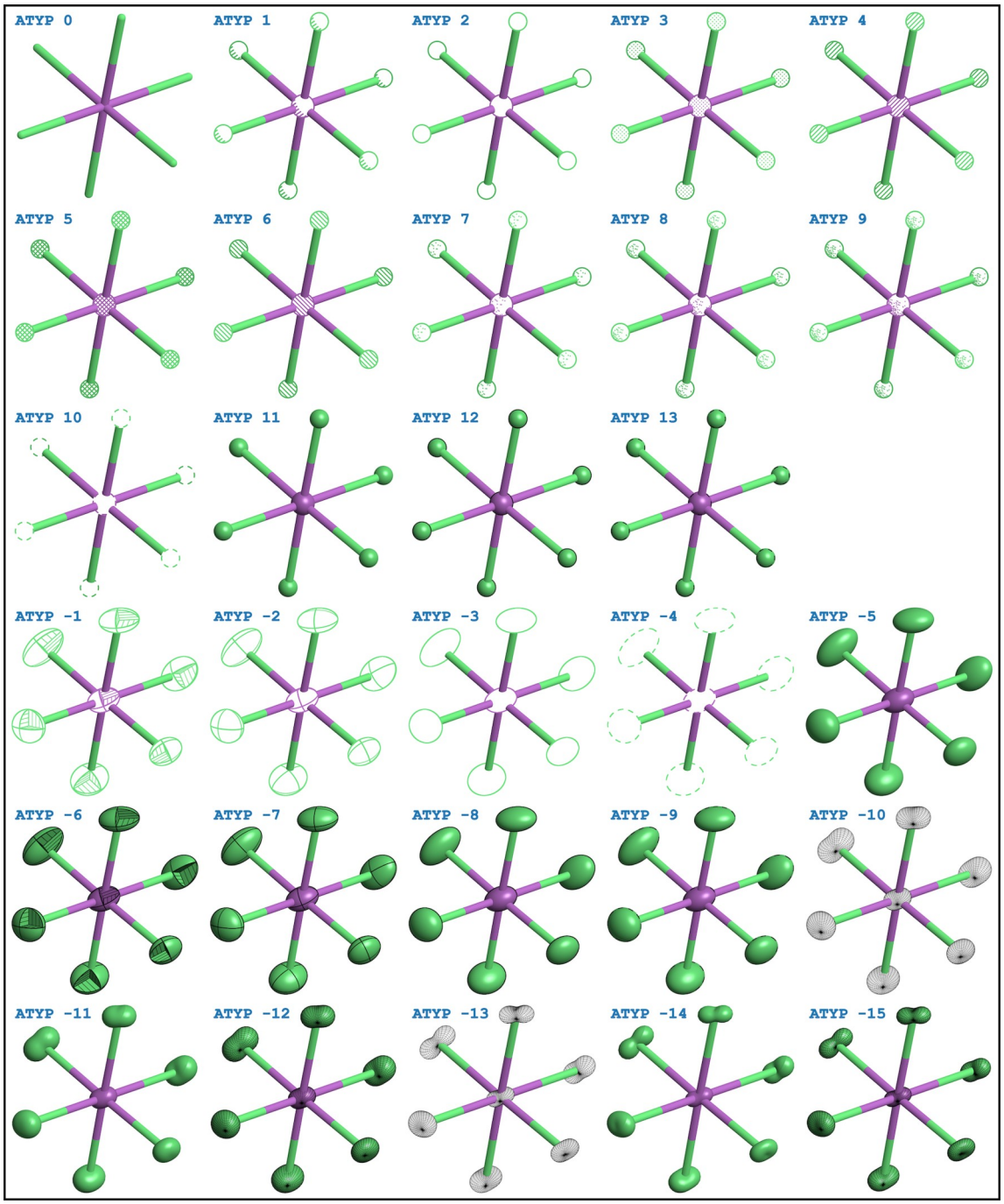


Figure 2.1a. The 29 available **ATYP** styles.

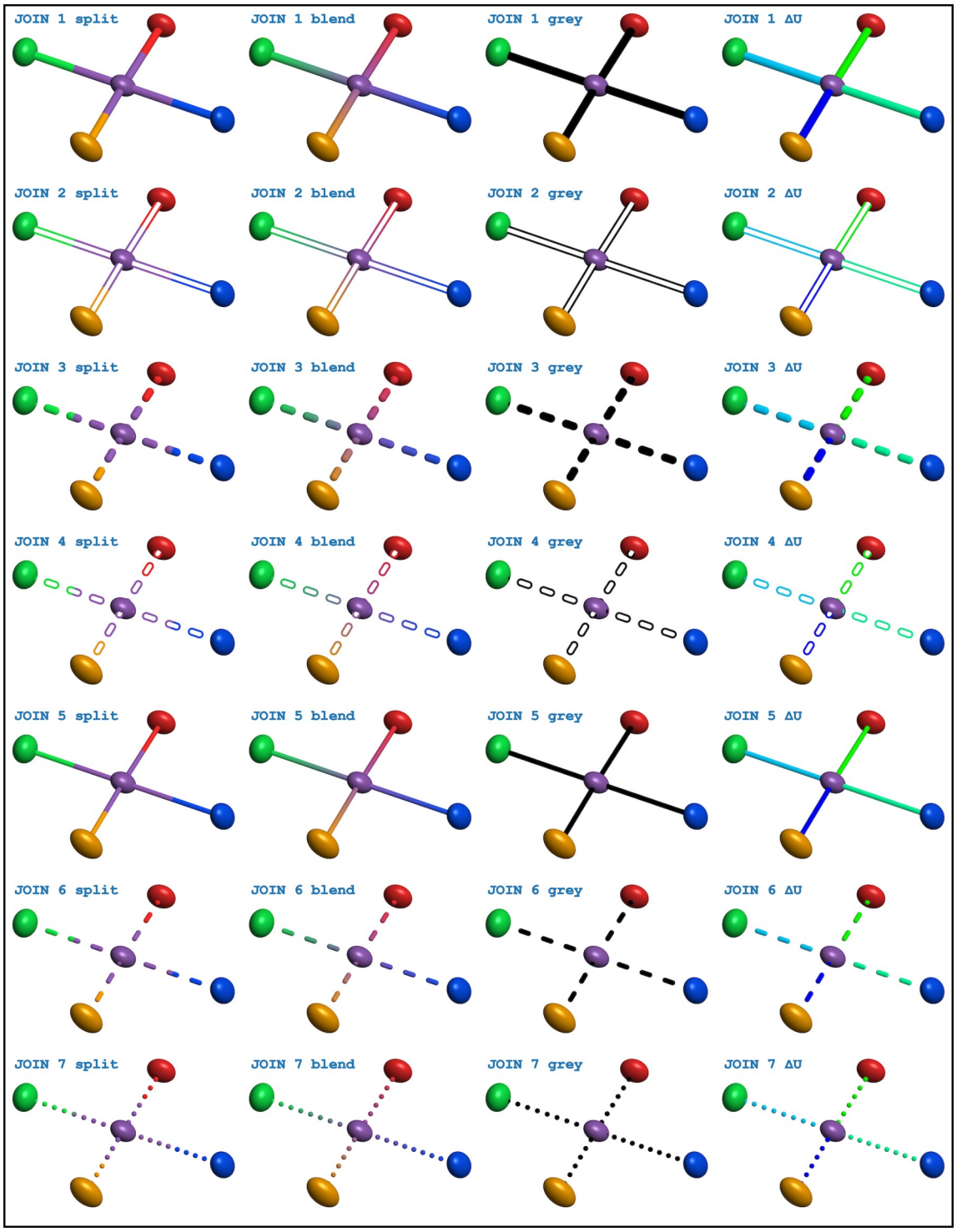


Figure 2.1b. The 7 **JOIN** and 4 bond colour styles.

2.2. Bog-standard *ORTEP*-style ellipsoid plots

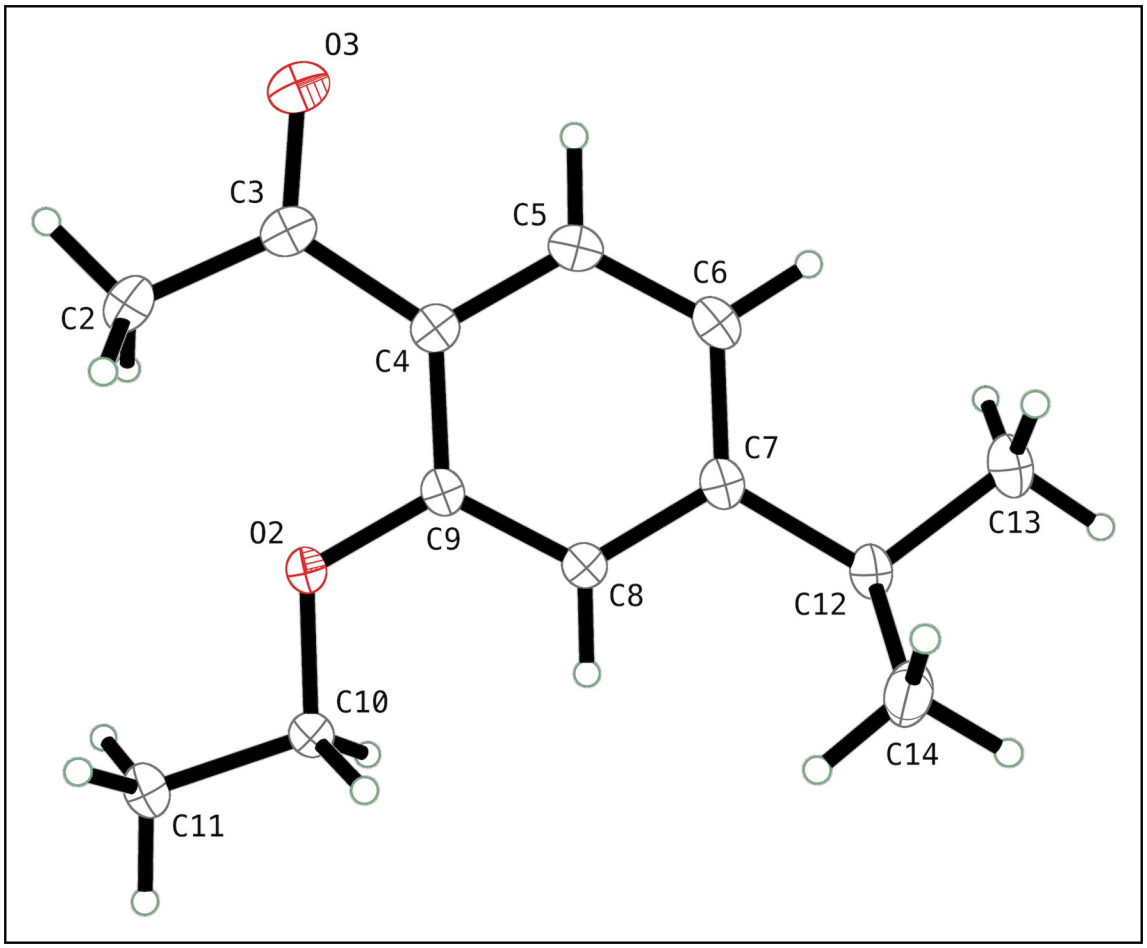


Figure 2.2. A generic 50% probability ellipsoid plot.

```
PrimeXP>> kill $q
PrimeXP>> telp
  [ Snap labels ]
PrimeXP>> atyp -1
PrimeXP>> atyp -2 $C
PrimeXP>> atyp 2 $H
  [ Grey bonds ]
  [ Gnomon off ]
PrimeXP>> draw Figure2.2
```

[Sowbhagya *et al.* (2026). *Acta Cryst.* E**82**, 530–533.]

2.3. Generic packing plots

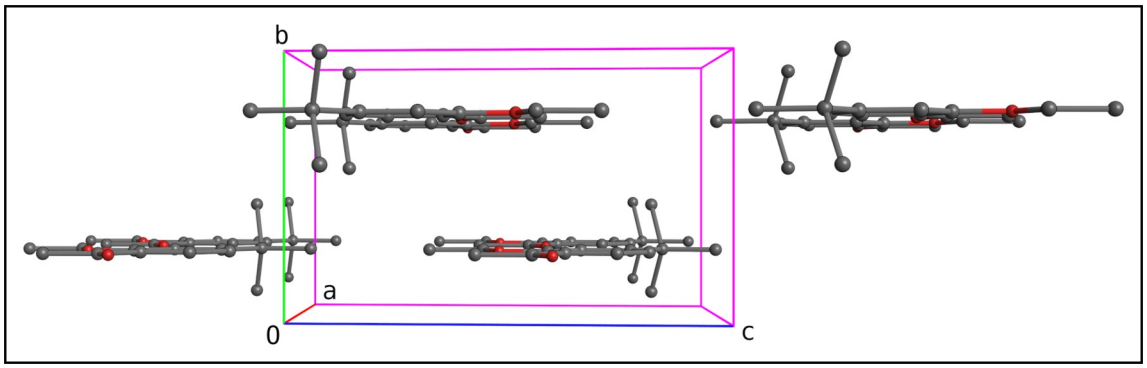


Figure 2.3a. View down the *a*-axis.

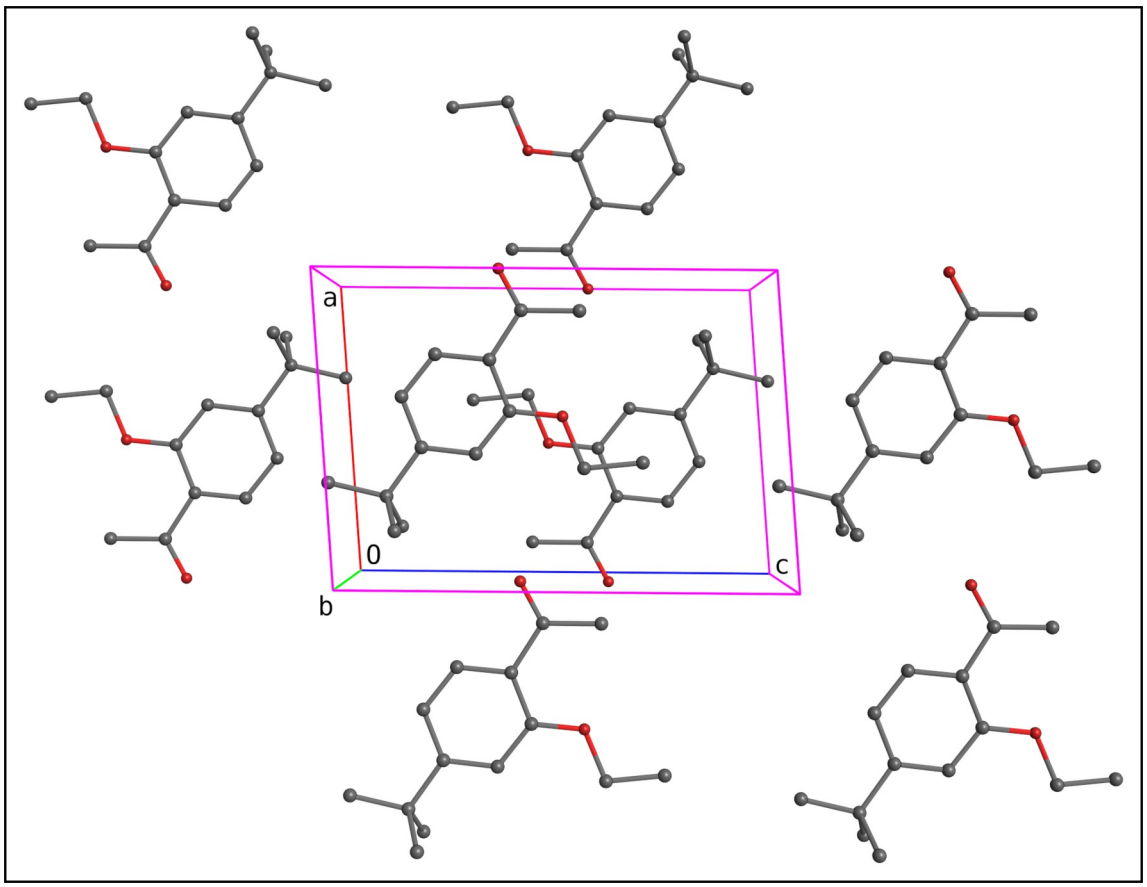


Figure 2.3b. View down the *b*-axis.

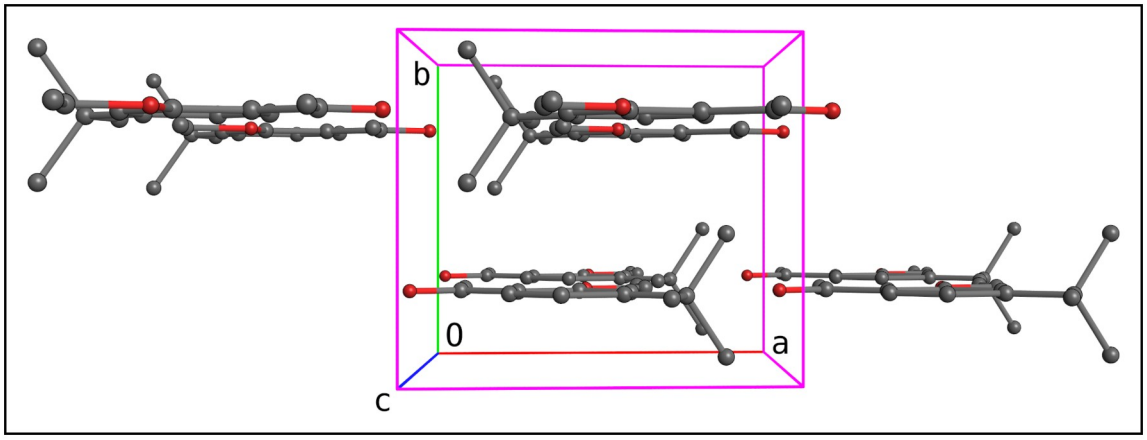


Figure 2.3c. View down the *c*-axis.

```
PrimeXP>> fmol less $h
PrimeXP>> mgen
  [ Labels off ]
PrimeXP>> matr 1
PrimeXP>> telp
PrimeXP>> atyp 12
  [ Smart labels ]
  [ Gnomon off ]
  [ Cell rgb ]
  [ Manual tweak of axis and origin labels ]
PrimeXP>> draw Figure2.3a
PrimeXP>> matr 2
  [ Manual tweak of axis and origin labels ]
PrimeXP>> draw Figure2.3b
PrimeXP>> matr 3
  [ Manual tweak of axis and origin labels ]
PrimeXP>> draw Figure2.3c
```

2.4. Star-join

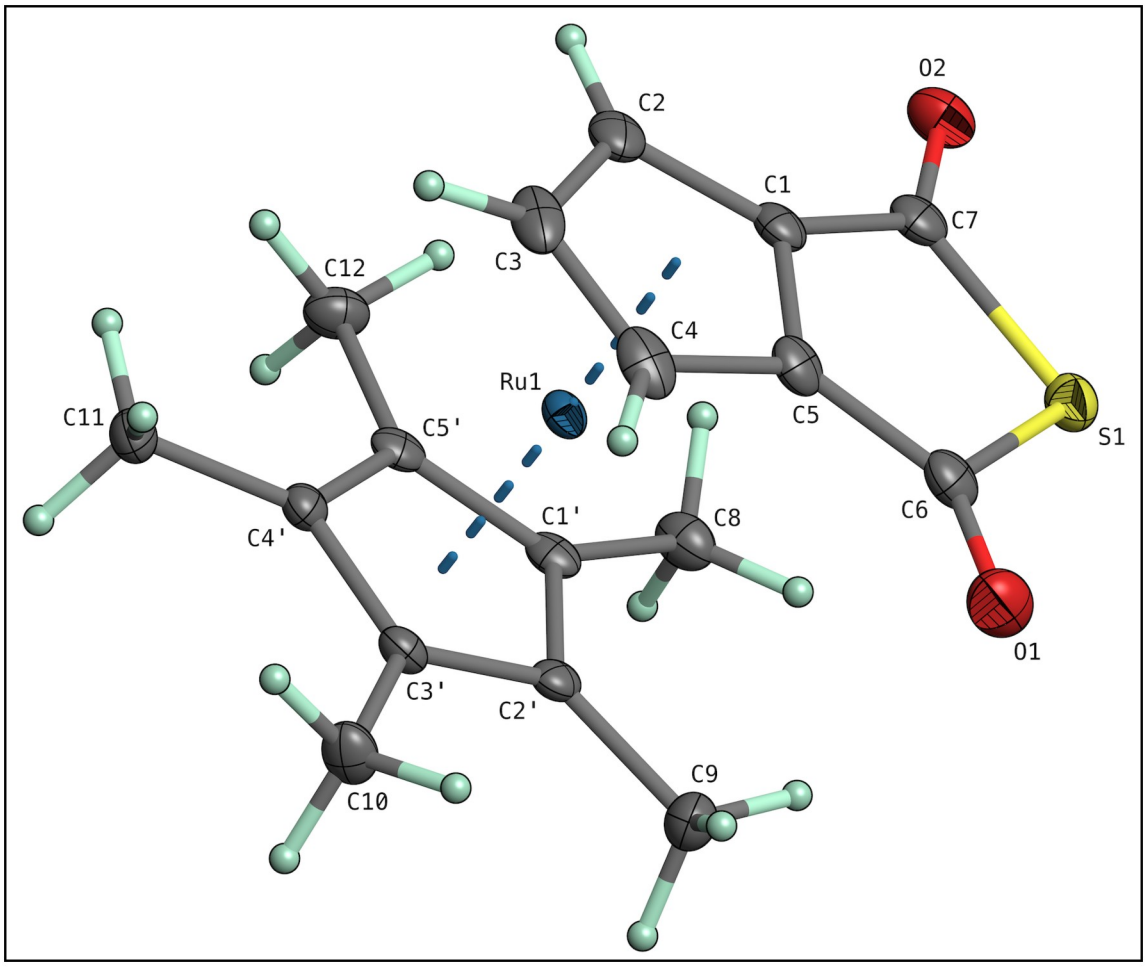


Figure 2.4a. A simple example of the star-join, **JOIN/s**.

```
    [ Gnomon off ]
    [ Kill Q-peaks ]
PrimeXP>> prun Ru1
PrimeXP>> ring/x
PrimeXP>> kill x2
PrimeXP>> prun Ru1
PrimeXP>> join/s 6 1 ru1 x1 x3
PrimeXP>> atyp 0 x*
    [ Manual re-orientation by mouse drag ]
    [ Snap labels ]
PrimeXP>> telp
```

[Pokharel *et al.* (2026). Crystals **16**, 409.]

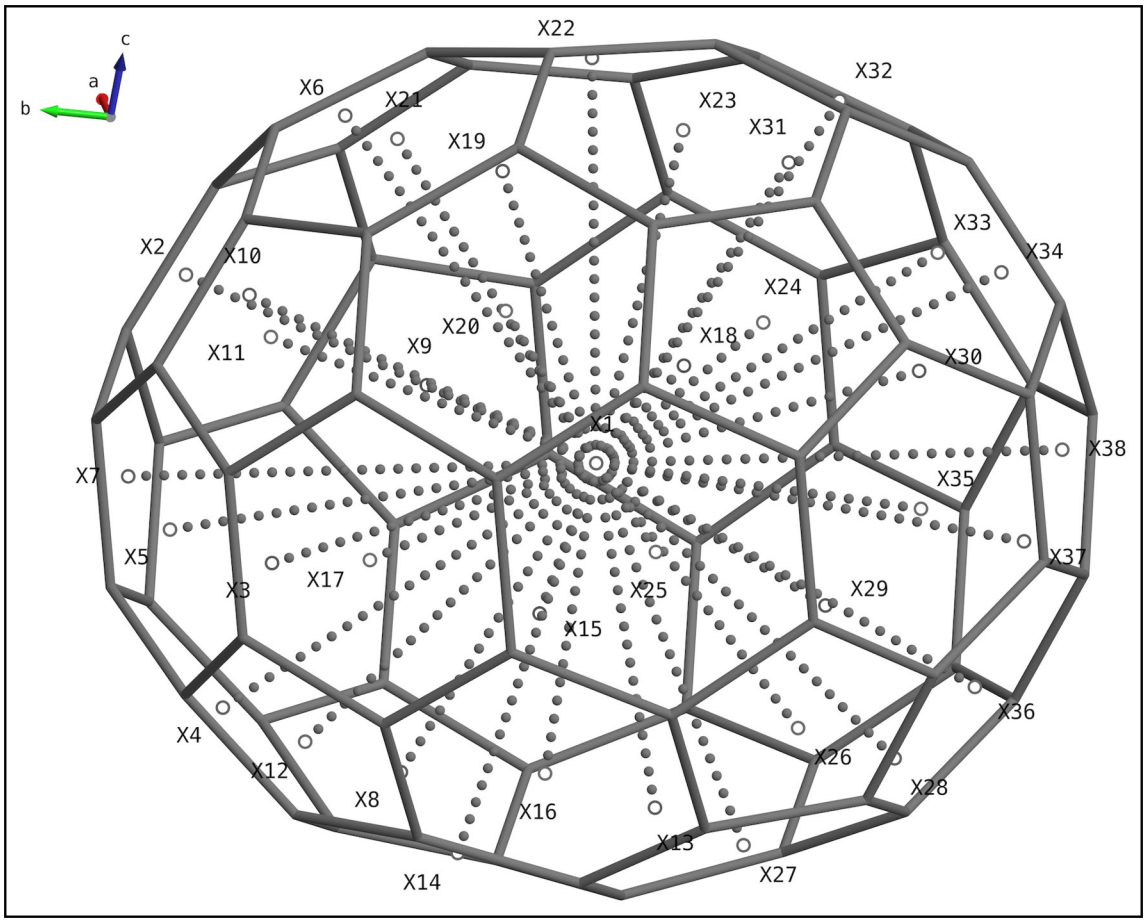


Figure 2.4b. Star join in C_{70}, showing cage centroid to each ring face centroid.

```
PrimeXP>> atyp 0 $c
PrimeXP>> cent/x
PrimeXP>> ring/x
PrimeXP>> join 5
PrimeXP>> join/s 7 x1 x*
PrimeXP>> arad 0.15 x*
PrimeXP>> atyp 2 x*
PrimeXP>> back white
PrimeXP>> bang/l x1
  X1    X2          3.947
  X1    X3          3.582
  X1    X4          3.711
etc.
```

[Gao *et al.* (2020). *J. Am. Chem. Soc.* 142, 2460–2470.]

2.5. Nearest-neighbour join

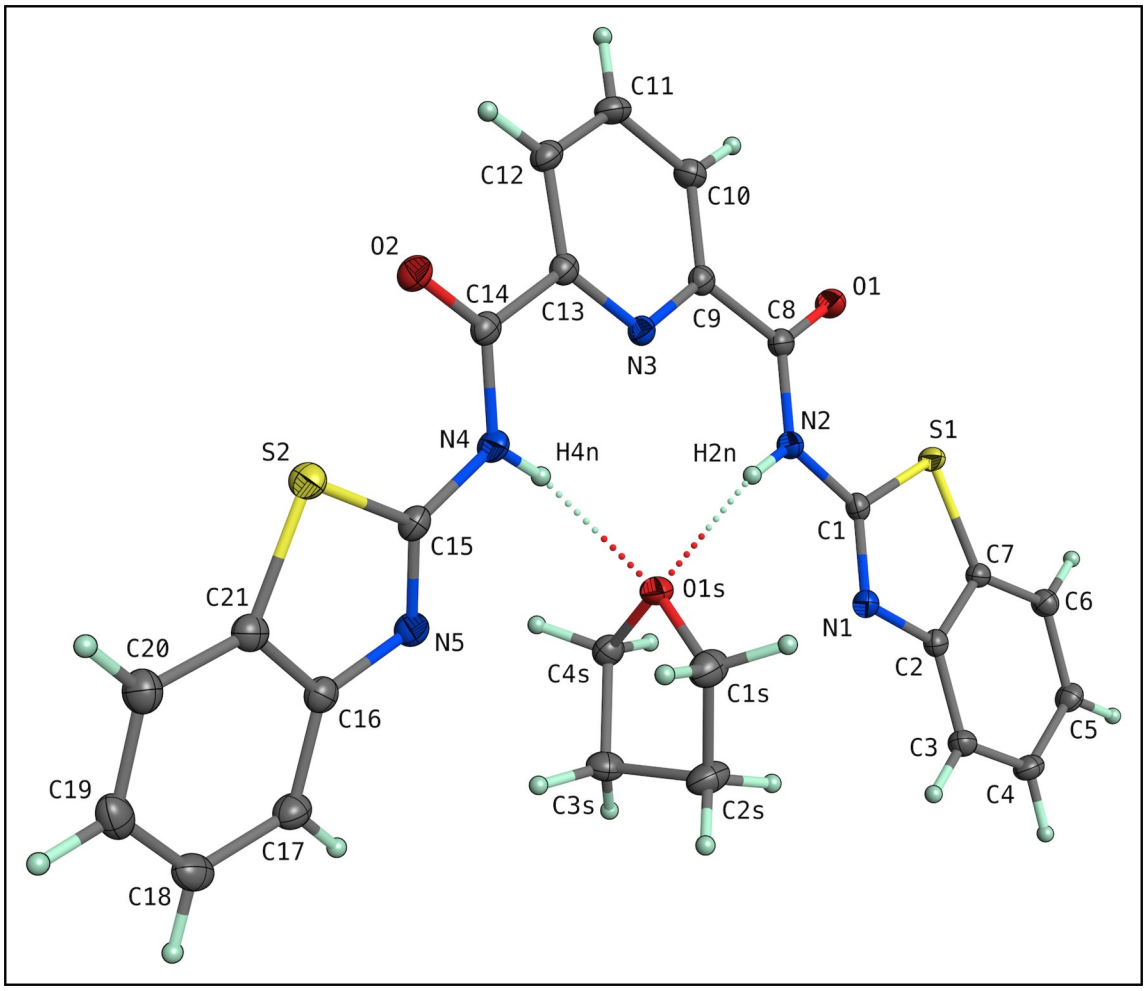


Figure 2.5. An example of nearest-neighbor join, **JOIN/x**. Note that the **HBND** command could also be used in this case.

```
    [ Gnomon off ]
    [ Kill Q-peaks ]
PrimeXP>> telp
PrimeXP>> join/x 7 1 h?n o1s
    [ All minus CH labels ]
    [ Smart labels ]
    [ Snap labels ]
    [ Manual tweak of some labels ]
PrimeXP>> draw Figure2.5
```

[Basu Baul *et al.* (2025). *Inorg. Chim. Acta* **585**, 122754.]

2.6. Pi-stacking join

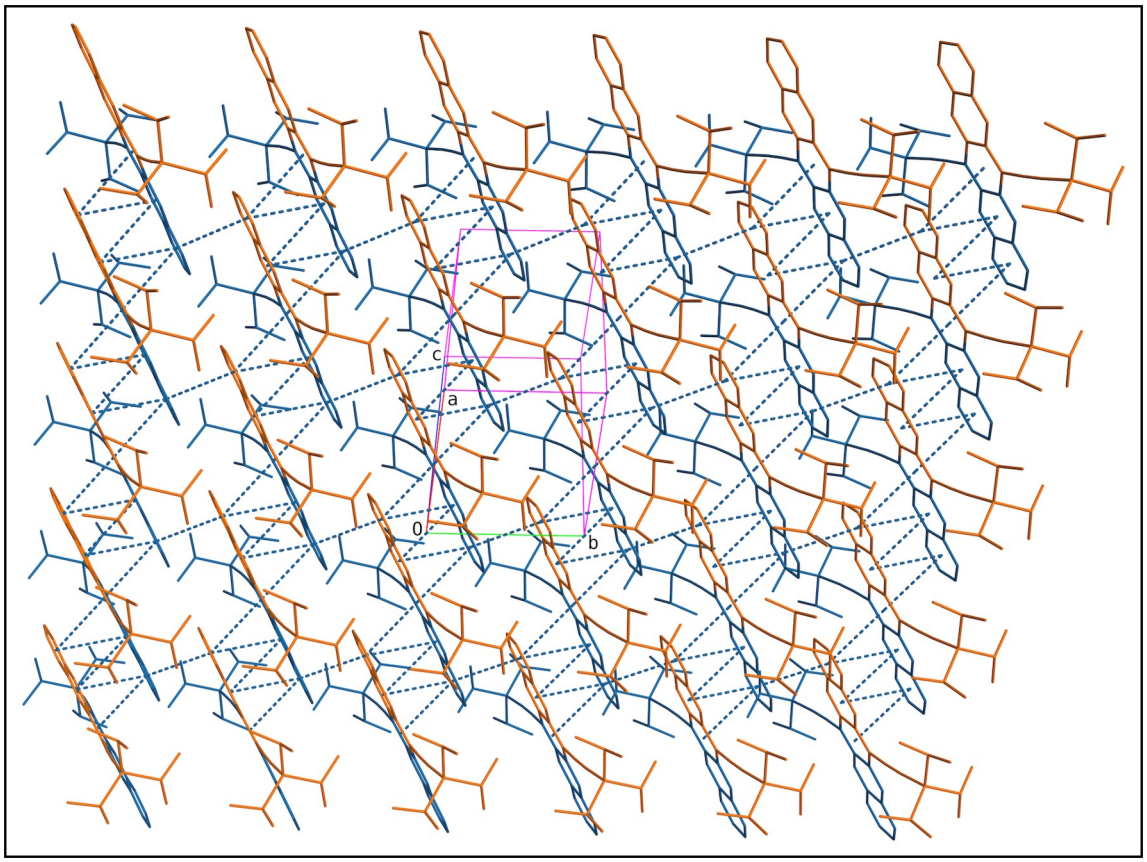


Figure 2.6. An example of the pi-stacking join, **JOIN/p** in tips-pentacene.

```
PrimeXP>> kill part2
PrimeXP>> grow
   [ Hydrogens off, Sticks, Labels off ]
PrimeXP>> mgen 2 2 0 0.5 0.5 0
PrimeXP>> mpln/n
PrimeXP>> ring/x
PrimeXP>> join/p 3 3 x* x*
atyp 0 x*
fmol/n
   [ SYMM atoms (just for the hell of it) ]
   [ Gnomon off ]
PrimeXP>> rota 2 -10
PrimeXP>> draw Figure2.6
```

[Anthony *et al.* (2001). *J. Am. Chem. Soc.* **123**, 9482–9483.]

2.7. Hydrogen bonds

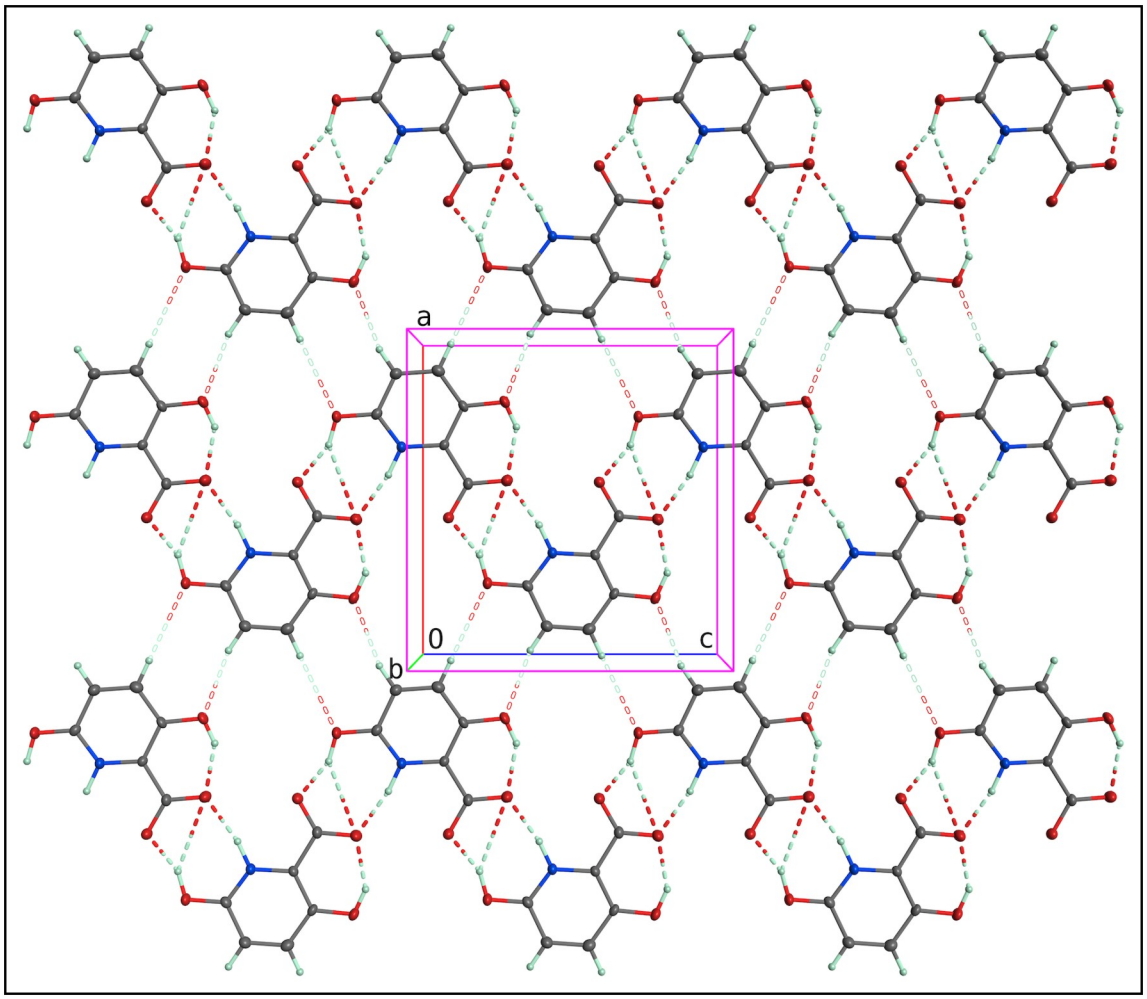


Figure 2.7. Conventional (solid blips) and weak (open blips) hydrogen bonds in picolinic acid.

```
PrimeXP>> kill part2
PrimeXP>> back white
PrimeXP>> hbnd 2 3 4
PrimeXP>> matr 2
PrimeXP>> rota 3 90
   [ Gnomon off, Labels off ]
PrimeXP>> draw Figure2.7
```

[Behrman & Parkin (2021). *Acta Cryst.* E**77**, 623–628.

2.8. Two-fold disorder

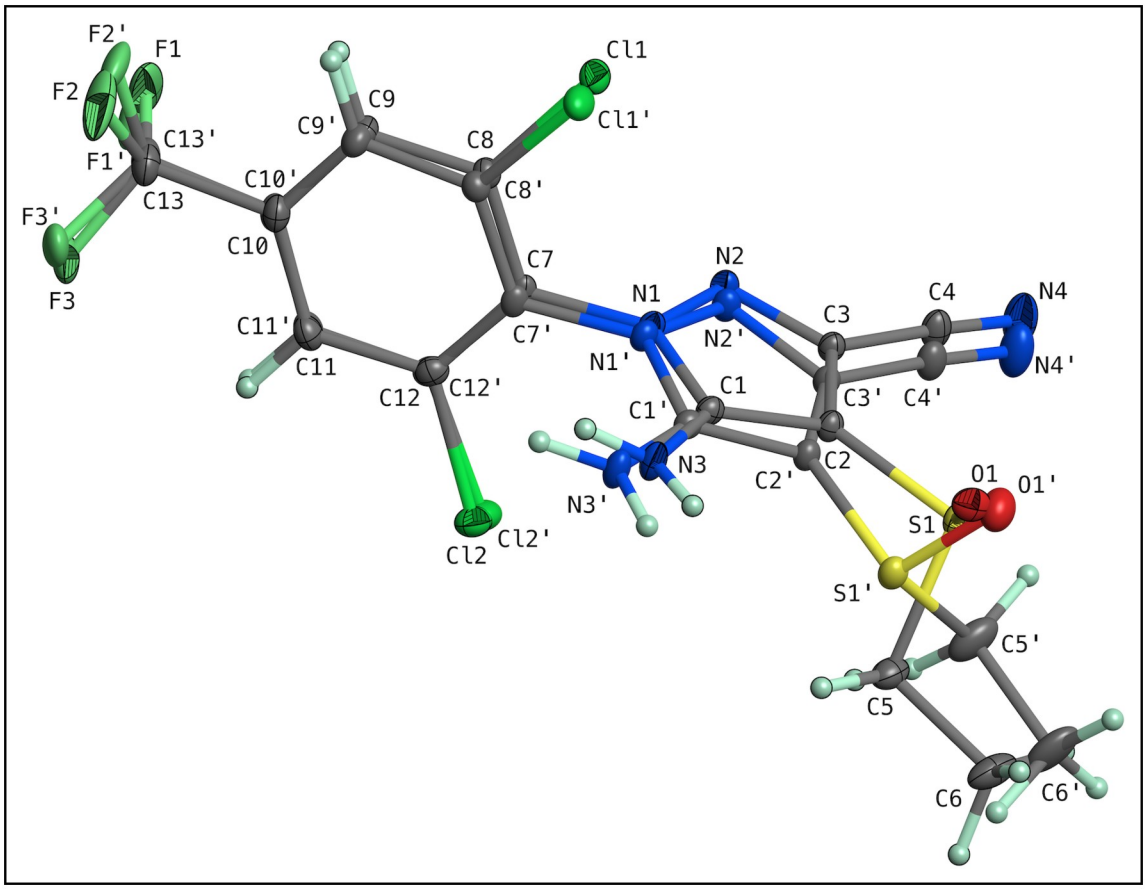


Figure 2.8. Two-fold whole molecule disorder in the insecticide ethiprole. Major component shown with cutout octants.

```
PrimeXP>> telp
PrimeXP>> adps 30
    [ Orthogonal ]
    [ Gnomon off ]
    [ Smart labels ]
    [ Snap labels ]
    [ Manual tweak of a few labels ]
PrimeXP>> draw Figure2.8
```

[Vinaya *et al.* (2023). *Acta Cryst.* E**79**, 54–59.]

2.9. Three-fold disorder

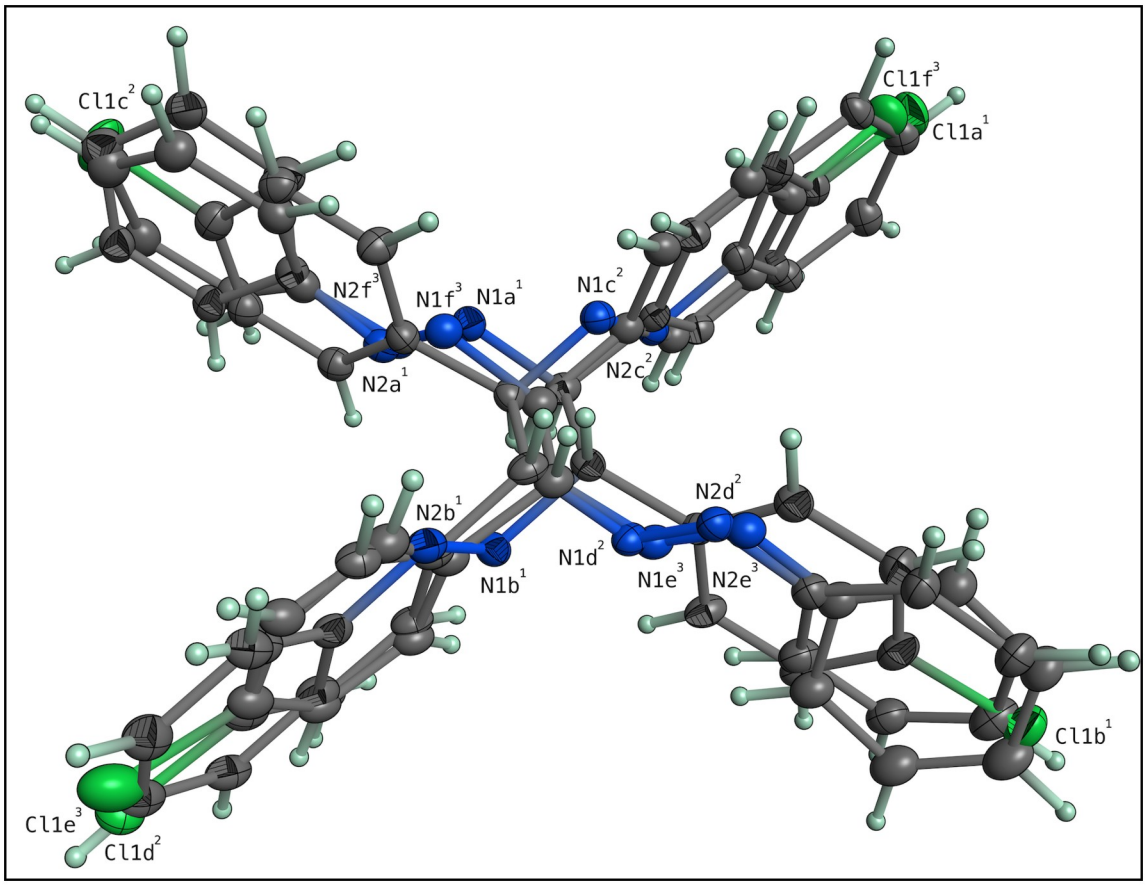


Figure 2.9. Three-fold whole molecule disorder.

```
PrimeXP>> kill $q
PrimeXP>> grow
PrimeXP>> back fff
PrimeXP>> matr 1 0.25 -0.25 -0.25 0 -1 -.25 1 0.25
PrimeXP>> atyp -6 part 1
PrimeXP>> atyp -7 part 2
PrimeXP>> atyp -8 part 3
PrimeXP>> atyp 12 $h
PrimeXP>> labl 1 300 $n $cl
   [ Part tags on ]
   [ Manual tweak of a few labels ]
PrimeXP>> draw Figure2.9
```

[Parkin *et al.* (2023). *Acta Cryst.* C**79**, 77–82.]

2.10. Electron density

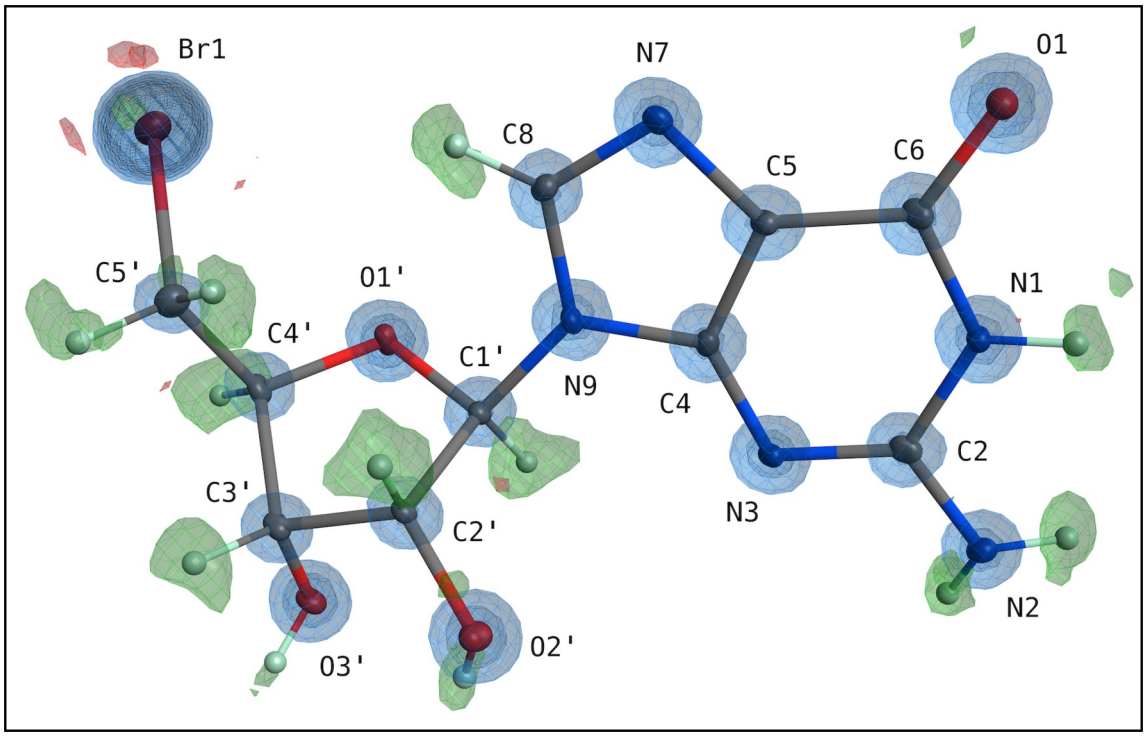


Figure 2.10. Multi-level electron density, Fobs (blue) and Fobs-Fcalc (green).

```
PrimeXP>> kill $q
   [ Gnomon off ]
   [ Smart labels, Snap labels ]
PrimeXP>> eden 1 2 4 6 8 10
PrimeXP>> eden 2 m18122d.fcf 3 6
   [ Manual tweak of atom labels ]
   [ Fobs map toggled to Combo ]
   [ Fobs-Fcalc map toggled to Combo ]
PrimeXP>> atyp 11 $h
PrimeXP>> join 5 $h
PrimeXP>> draw Figure2.10
```

[Parkin *et al.* (2018). *Cryst. Growth Des.* **18**, 6995–7005.]

2.11. Coordination polyhedra

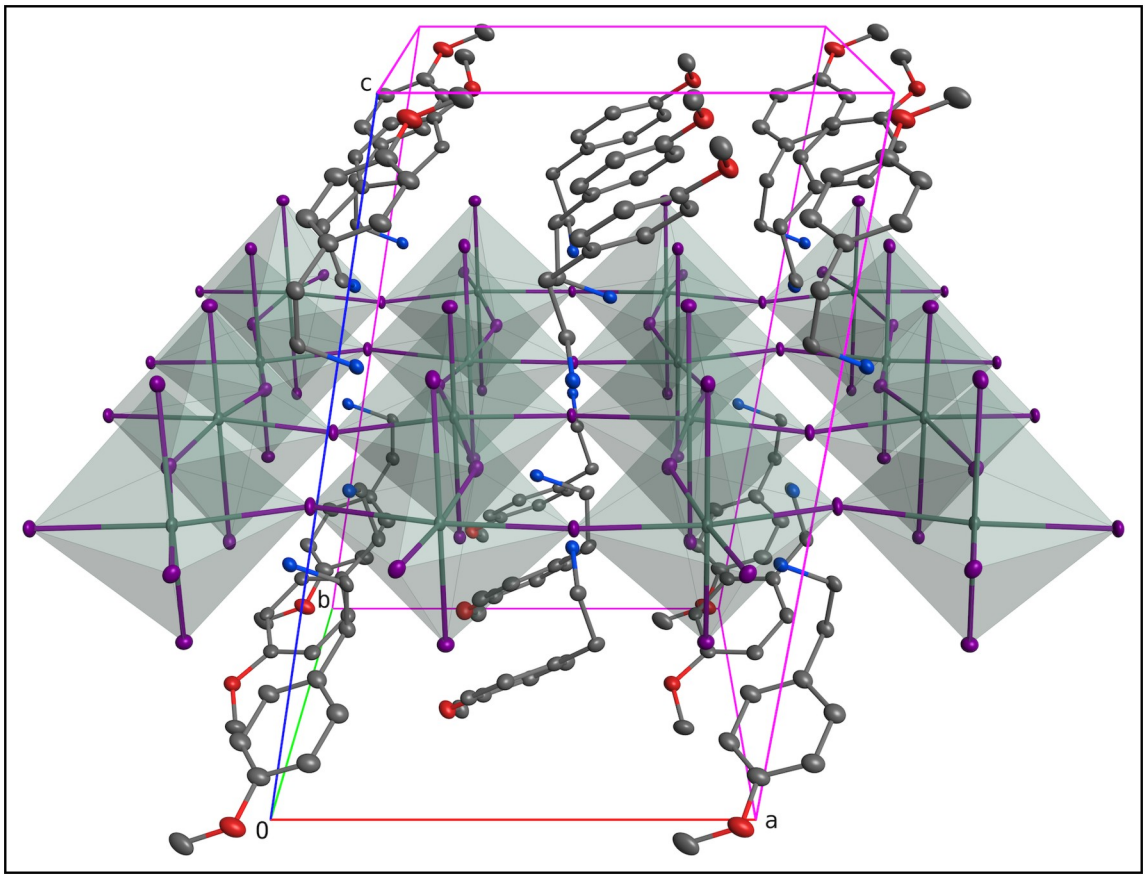


Figure 2.11. Coordination octahedra in a layered hybrid organic-inorganic pseudo-perovskite structure.

```
PrimeXP>> kill $q $h part2
  [ Gnomon off ]
PrimeXP>> arad .5 1.7 $I
PrimeXP>> mgen .76 .76 .35
  [ Labels off ]
  [ Manual kill of a few Sn and I (Atom edit button) ]
  [ Cell rgb ]
PrimeXP>> back fff
PrimeXP>> poly/k $sn
PrimeXP>> matr 1 0 0 0 0.333 1 0 -1 0.333
PrimeXP>> draw Figure2.11
```

[Hossain *et al.* (2024). *Chem. Mater.* **36**, 11004.]

2.12. Solvent-accessible voids

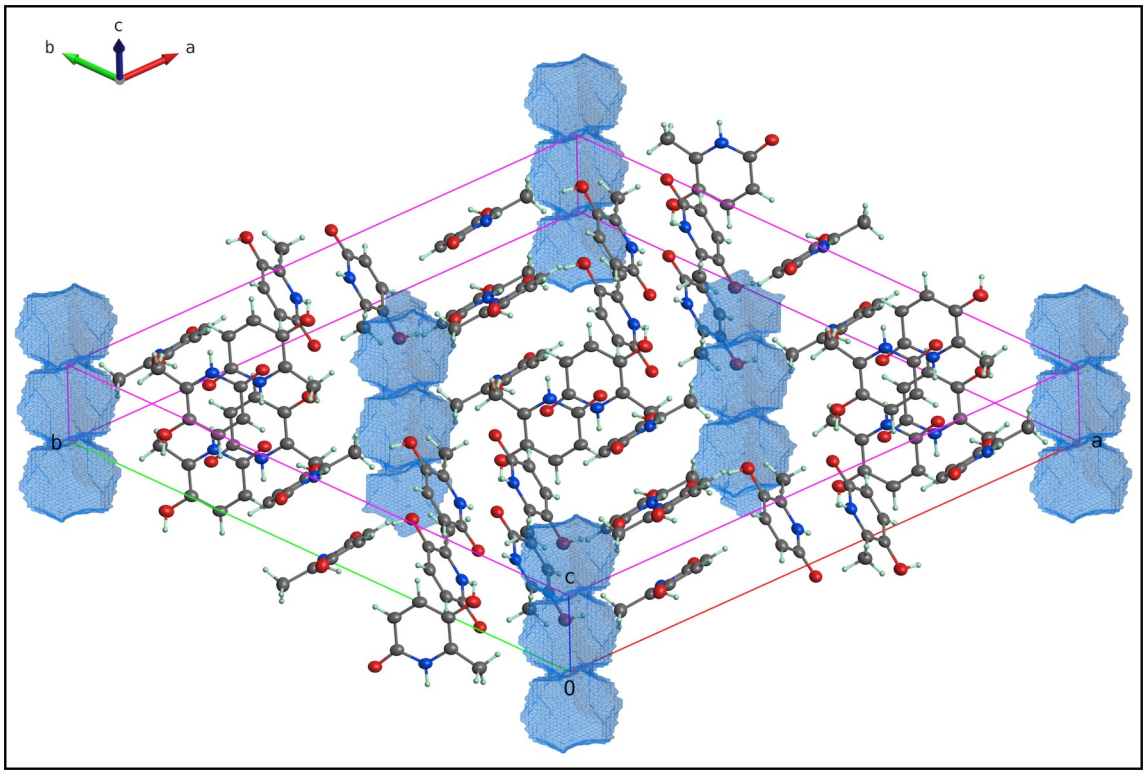


Figure 2.12. Solvent-accessible voids (light blue) in a trigonal substituted pyridone.

```
PrimeXP>> mgen
   [ Labels off ]
   [ Cell rgb ]
PrimeXP>> void 1.2 0.2 0.2 0.2 0.75
PrimeXP>> back white
   [ Void: combo ]
   [ Orthogonal ]
   [ Snap labels ]
PrimeXP>> draw Figure2.12
```

[Parkin & Behrman (2011). *Acta Cryst.* C**67**, o324–o328.]

2.13. A small protein (BPTI) at atomic resolution

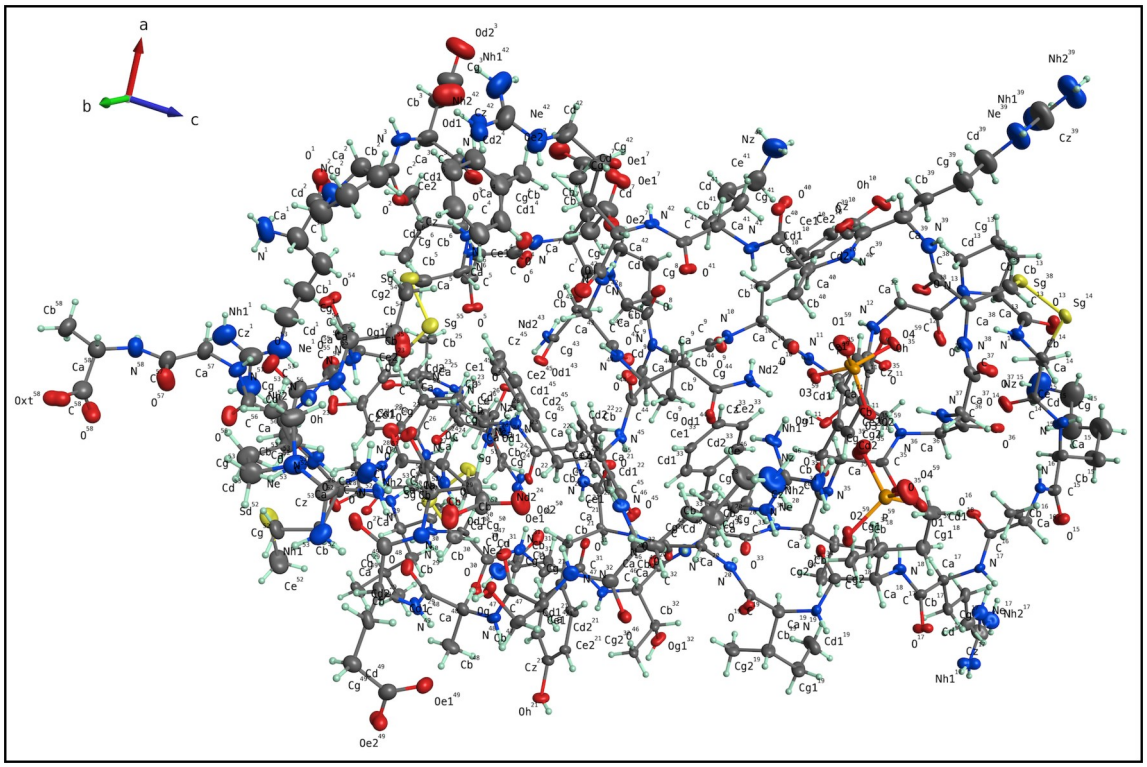


Figure 2.13. Structure of BPTI at 1.1 Å, with mean-square displacement peanuts and **RESI**-tagged labels.

```
PrimeXP>> kill resi hoh
PrimeXP>> pnut 2
PrimeXP>> labl 1 250
   [ RESI tags shown ]
   [ Snap labels ]
   [ note: ~17 seconds for 470 atoms on an M3 MacBook Pro ]
PrimeXP>> draw bpti
PrimeXP>> hist bpti.txt
```

[bpti: Parkin *et al.* (1996). *Acta Cryst.* D**52**, 18–20.]
[peanuts: Hummel *et al.* (1990). *J. Mol. Graphics* **8**, 214–220.]

2.14. Water channels in a protein (conA) crystal

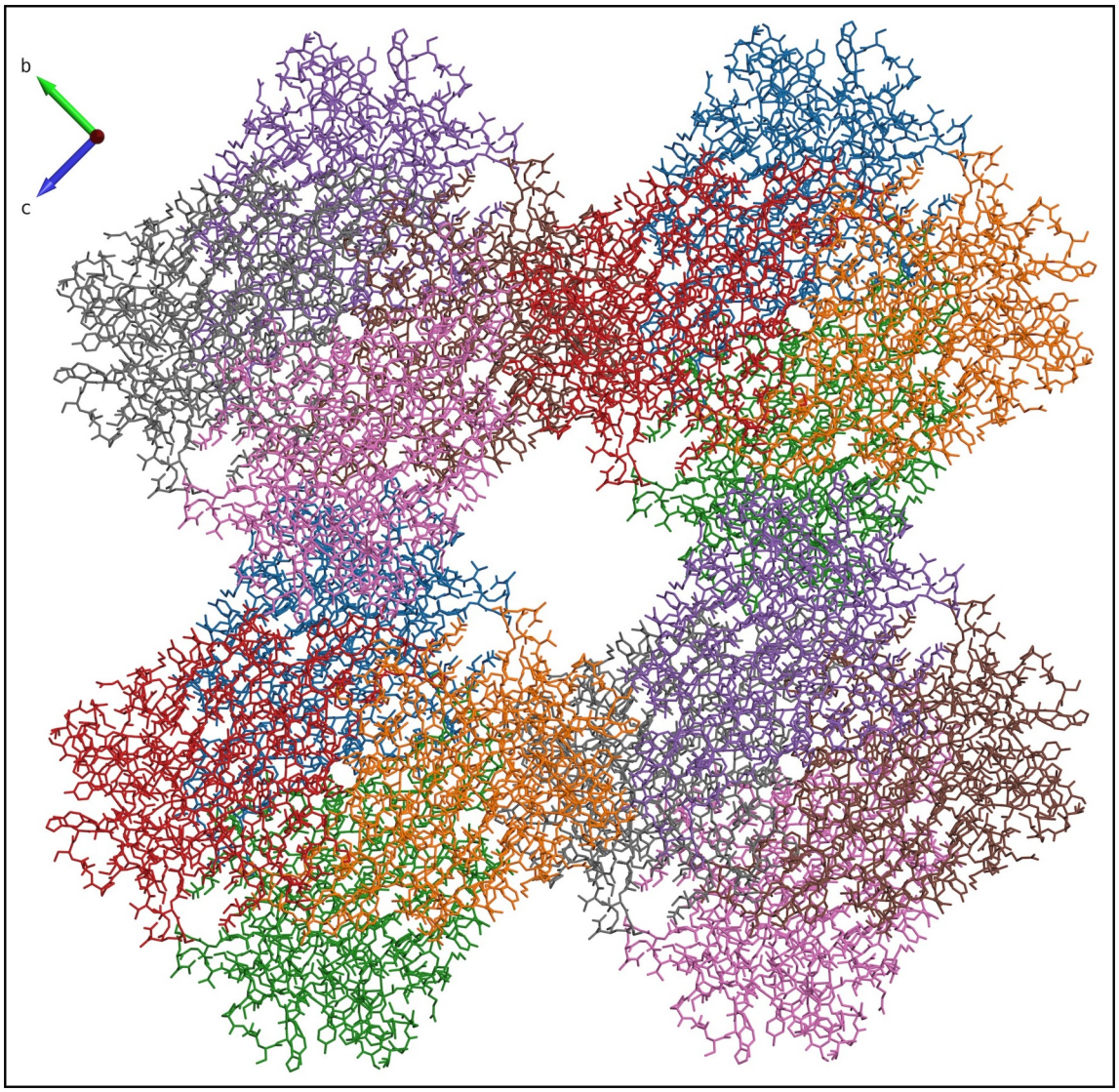


Figure 2.14. A view down the *a*-axis of *I*222 concanavalin A, showing ~30 Å wide water channels. Colours are by symmmetry operation.

```
PrimeXP>> kill resi hoh
PrimeXP>> matr 1
PrimeXP>> rota 3 45
   [ Orthogonal, Atom colour mode set to SYMM atoms ]
```

[Parkin *et al.* (1996). *Acta Cryst.* D**52**, 1161–1168.]
[Parkin & Hope (2003). *Acta. Cryst.* D**59**, 2228–2236.]

3. Command list

ADPS (Atomic displacement parameters and probability)

Displays a table of atomic displacement parameters (ADPs) for the specified atoms, and/or sets the global probability level for ellipsoid plots.

USAGE

ADPS [prob] [atom_list]
ADPS/H [prob] [atom_list]

ARGUMENTS

[prob] : Optional. A number between 1.0 and 99.0 sets the global probability level for drawing displacement ellipsoids.

[atom_list] : Can be specific labels, ranges, or element types. If omitted, applies to ALL atoms.

QUALIFIER

/H : Includes hydrogen atoms.

OUTPUT

- A table with rectangles coloured by the U_{eq} (blue = low, red = high)
- Atom label, U^{11}, U^{22}, U^{33}, U^{23}, U^{13}, U^{12}, U_{eq} (or U_{iso})

NOTES

- If only a probability value is provided (e.g., **'ADPS 50'**), the ellipsoid probability is updated and the table is suppressed.
- For atoms refined isotropically, the individual U^{ij} components are dashed out, with the U_{iso} value in the final column.
- Riding hydrogen atoms show their *SHELXL*-style specifier (e.g., -1.2) rather than a derived value.
- Ellipsoid colours can be mapped to ADP values or to symmetry codes.

ARAD (Atomic radii)

Changes or lists atomic radii.

USAGE

ARAD [ar] [br] [sr] [atom_list]

ARGUMENTS

[ar] : The radius (in ångström units) used to represent atoms as spheres (i.e, positive **ATYP** values) in **PROJ**, **TELP**, **PNUT** etc. modes.

[br] : Bonding radius. Used by **FMOL**, **SRCH**, **ENVI**, **GROW**, and **UNIQ**. A bond is generated between two atoms when the distance is shorter than br(1) + br(2) + δ (default 0.5).

[sr] : The radius used in **SPIX** space-filling plots.

[atom_list] : Can be specific labels, ranges, or element types. If omitted, applies to ALL atoms.

BEHAVIOUR

- 0 numbers : Lists the current ar, br, and sr for the specified atoms.
- 1 number : Sets ar.
- 2 numbers : Sets ar and br.
- 3 numbers : Sets ar, br, and sr.

NOTES

- The element symbol determines the default values of ar, br, and sr when reading in atoms or renaming them (via **PICK** or **NAME**). The initial values are in file "rsrc_radii.py"
- When generating symmetry equivalent atoms, the current radii are copied.
- If any bonding radii (br) are changed, the connectivity table is updated.

`ATYP` (Atom drawing type)

Sets the visual representation mode for the specified atoms.

USAGE

```
ATYP type [atom_list]
```

ARGUMENTS

type : An integer defining the drawing style.

`[atom_list]` : The atoms to modify. If omitted, applies to ALL atoms.

AVAILABLE TYPES

--- Anisotropic styles (displacement ellipsoids) ---

`-1` : Line-drawn ellipsoid with nearest octant cutout.

`-2` : Line-drawn ellipsoid with principal equators and outer boundary.

`-3` : Line-drawn ellipsoid outer boundary only.

`-4` : Line-drawn ellipsoid dashed outer boundary.

`-5` : 3D shaded ellipsoid.

`-6` : 3D shaded ellipsoid with nearest octant cutout.

`-7` : 3D shaded ellipsoid with principal equators and boundary.

`-8` : 3D shaded ellipsoid with outer boundary.

`-9` : 3D shaded ellipsoid with dashed outer boundary.

--- Anisotropic styles (*PEANUT*-style plots) ---

`-10` : Mesh-drawn rmsd peanuts.

`-11` : 3D shaded rmsd peanut.

`-12` : Mesh-drawn rmsd superimposed on 3D shaded rmsd peanuts.

`-13` : Mesh-drawn msd peanuts

`-14` : 3D shaded msd peanuts.

`-15` : Mesh-drawn msd superimposed on 3D shaded peanuts.

`0` : Invisible. Useful for dummy atoms such as centroids (e.g., centre of a ring). Note: Invisible atoms are made visible during `PICK` mode.

--- Isotropic styles (spheres & 2D circles) ---

`1` : Circle with a parallel line shading wedge.

`2` : Open circle.

`3` : Circle with a regular dot grid.

`4` : Circle with parallel lines.

`5` : Circle with a crosshatch mesh.

`6` : Circle with parallel lines (opposite diagonal to `ATYP 4`).

`7` : Stippled circle (Low density dots).

`8` : Stippled circle (Medium density dots).

9 : Stippled circle (High density dots).
10 : Dashed open circle.
11 : 3D shaded sphere.
12 : 3D shaded sphere with an outer boundary line.
13 : 3D shaded sphere with a dashed outer boundary line.

NOTES

- Unlike the original *XP*, this command does not change the atom colour.
- On loading a file, defaults are assigned based on rsrc_config.py (e.g., non-H atoms default to **ATYP -5**, and H atoms default to **ATYP 11**).
- Symmetry-generated atoms inherit the **ATYP** of their parent.
- Hidden-line removal for line-drawn *PEANUT* meshes (**ATYP -10**, -13) is enabled by default and can be changed in the rsrc_config.py file.

BACK (Background colour)

Changes the background colour or gradient of the graphics window. The program accepts six-digit hexcodes, so 16,777,216 colours are possible. Three-digit hexcodes are also allowed (4096 colours), and 148 named **'web colours'** are available (see "rsrc_colours.py"). Based on the luminance of the background, atom, cell, and gnomon labels are set to white or black. This may be overridden with **'LABL colour'** (see **HELP LABL**).

USAGE

```
BACK
BACK [colour1] [[TO] colour2]
BACK/T [colour1] [[TO] colour2]
BACK FLIP
BACK/T FLIP
```

ARGUMENTS

[colour1] : A 3 or 6-digit hexcode (e.g., #000000 or #000 for black, #FFF or #FFFFFF for white) or one of the 148 named colours in "rsrc_colours.py". If only one colour is given, the background is uniform.

[colour2] : If a second colour is provided, the background becomes a smooth linear gradient between the two colours. The **'TO'** keyword is optional.

FLIP : Reverses the gradient by swapping top and bottom colours.

QUALIFIER

/T : Applies the colour changes to the Terminal window.

EXAMPLES

```
BACK                  -> Reports the current background colour/gradient
BACK #000000          -> Solid black
BACK white            -> Solid white
BACK black TO white   -> black to white gradient
BACK flip             -> Reverses the existing gradient.
```

BANG (Bond distances and angles)

Calculates and displays bond distances and angles for specified atoms based on the current connectivity table.

USAGE

```
BANG [atom_list]
BANG/L [atom_list]
BANG/H [atom_list]
```

ARGUMENTS

[atom_list] : The atoms to be analyzed. If omitted, applies to all atoms currently in the **FMOL** list.

QUALIFIERS

/L : Suppresses output of bond angles.

/H : Includes bond lengths and angles involving hydrogen atoms.

OUTPUT

For each atom in the list, the command generates:

- The distance to every bonded neighbour (in ångströms).
- All unique bond angles (A-B-C) formed by those neighbours (unless the **/L** qualifier is used).

NOTES

- By default, hydrogen atoms are excluded from the output. Use the **/H** qualifier or explicitly specify them in the [atom_list] to include them.
- Calculations uses the current connectivity table.
- Unlike **ENVI**, which searches for all atoms within a physical radius, **BANG** only reports on atoms that are chemically linked in the model.

BUTTons (Graphical user interface controls)

BUTT is not really a command as such, it just displays this help file entry. The *PrimeXP* graphics window includes a column of interactive buttons on the right side. When clicked, these buttons cycle through different states or a dropdown menu appears, allowing adjustment of the 3D view without typing.

AVAILABLE BUTTONS

Labels : Opens a dropdown menu to select specific label visibility states (Off, No H, ASU-H, All, ASU+H, All-CH, ASU-CH, Custom).

Smart/Dumb : Toggles **'Smart Labels'**. When enabled, the program uses an anti-collision engine to push labels apart and prevent them from obscuring atoms and bonds. (Note: Enabling this reveals additional buttons for manual label tweaking, including a toggle to click any atom/label to manually hide or restore it, which triggers **'Custom'** mode). This is slow for large structures.

H atoms : Toggles the global graphical visibility of hydrogen atoms.

Style : Cycles the geometric drawing style of the model: (Wireframe -> Sticks -> Hybrid -> Ball-n-stick).

Atom colour : Cycles atom rendering in the graphics window between CPK (elemental), **SYMM** (symmetry equivalence), and ADP (thermal parameters). The button text dynamically updates to display the currently active mode.

Bond colour : Cycles the bond colour style: (Split -> Blended -> Grey -> ΔU-based colours).

Atom edit : Toggles interactive Edit mode. When enabled, clicking an atom opens a contextual menu to hide or kill that atom, or the entire molecule it belongs to.

Bond edit : Opens a dropdown menu for interactive bond creation. Select a bond type (1 through 7), then click two atoms in the viewer to join them. Select **'Edit off'** to exit this mode.

Shading : Cycles the depth representation: (3D Shaded -> 2D flat with depth gaps -> 2D flat no gaps). The 2D modes are ideal for traditional publication drawings.

Cell : Cycles unit cell visibility: (Off -> Standard colour -> RGB coloured a,b,c axes).

Projection : Toggles the camera lens between perspective (realistic depth) and orthogonal (flat, no depth distortion) projection.

Stereo : Toggles anaglyph (red/cyan) 3D stereo mode (see **EYES**).

Gnomon : Cycles the coordinate axis indicator in the top-left: (Off -> axis vectors only -> labelled a,b,c vectors).

Eden : Cycles the display style of loaded electron density maps (Off -> Cages -> Clouds -> Combined). This button only appears if an **EDEN** map is actively loaded.
Void : Cycles the display style of the accessible void surface map (Off -> Cages -> Clouds -> Combined). This button dynamically updates its label and only appears if a **VOID** map has been generated.
Kill Q : Instantly removes all Q-peaks from the model. (This button only appears if Q-peaks are present in the current list).
Quit : Exits the program (only visible if enabled in rsrc_config.py).

NOTES

- The state of these buttons is saved when using the **SAVE** command and restored when using **NEXT**.
- Many of these toggle states are linked directly to terminal commands, meaning clicking a button will often execute the corresponding command silently in the background.

CELL (Unit cell parameters and display colour)

Displays the unit-cell parameters, or updates the colour of the unit cell box in the graphics window.

USAGE

CELL

CELL colour

ARGUMENTS

colour : A valid hex code (e.g., #FF00FF) or a recognized web colour name (e.g., magenta, teal) to set the unit cell box rendering colour.

OUTPUT

When called without arguments, it displays the unit cell information:
- Cell edges: a, b, c
- Cell angles: α, β, γ
- Calculated cell volume (V)

When called with a colour argument, it confirms the new colour selection, updates the live graphics window, and writes the unit cell information.

NOTES

- Usually, these parameters are loaded from the .res or .ins file during the **READ** or **REAP** command.
- This is a stripped down version of **CELL** compared to *XP*. More functionality will be added if needed.

CENT (Centroid calculation)

Calculates the unweighted centroid of a specified group of atoms.

USAGE

```
CENT [atom_list]
CENT/X [atom_list]
CENT/H [atom_list]
```

ARGUMENTS

[atom_list] : The atoms used to calculate the centroid. If omitted, uses all non-hydrogen atoms currently in the **FMOL** list.

QUALIFIERS

/X : Creates a dummy atom (named X1, X2, etc.) at the calculated centroid position and adds it to the active **FMOL** list.

/H : Includes hydrogen atoms in the centroid calculation.

NOTES

- By default, hydrogen atoms are excluded from the calculation. Use the **/H** qualifier or explicitly specify them in the [atom_list] to include them.
- The calculated fractional and orthogonal coordinates are written to the terminal.
- Centroids generated with **/X** are not automatically bonded to any other atoms. Use the **JOIN** command to link the centroid to a metal centre or other reference point.
- This can be useful for organometallic complexes (e.g., calculating the centre of a pi-bonded ring), but see also the **RING/X** command, which adds dummy atoms to the centres of a variety of ring types.

COLR (Set atom colour mode)

Changes the colouring scheme of atoms in the graphics window to highlight different chemical, symmetrical, or thermal properties.

USAGE

COLR [mode]

ARGUMENTS

mode : The specific colour scheme to apply. Valid options are: CPK (or 1) - colours atoms by their standard chemical element. **SYMM** (or 2) - colours atoms according to their macroscopic symmetry operation (matches the **SYMM** command output). ADP (or 3) - colours atoms based on their Atomic Displacement Parameter probabilities.

(no arguments) : Cycles through the available colour modes (e.g., CPK -> **SYMM** -> ADP -> CPK...).

NOTES

- Interactive update: Changing the colour mode immediately refreshes the graphics window and updates the corresponding UI button text to reflect the currently active mode.
- Symmetry mapping: When using the **SYMM** mode, atoms generated via **MGEN**, **GROW**, or **SGEN** will perfectly preserve and display their internally mapped symmetry equivalence.

CPOP (Cremer-Pople puckering parameters)

Calculates Cremer-Pople puckering parameters for 5- and 6-membered rings using Cartesian out-of-plane displacements.

SYNTAX

CPOP [atoms]

- If no atoms are specified, **CPOP** evaluates all valid 5- and 6-membered rings in the current **FMOL** list.
- If atom names are provided, only rings containing those atoms are evaluated.

OUTPUT

- For 6-membered rings, output includes the total puckering amplitude (Q) and spherical polar angles (Θ, Φ_2).
- For 5-membered rings, it outputs the amplitude (q_2) and phase angle (Φ_2).
- For 5-membered rings, the principal out-of-plane "flap" atom is also given.

CPOP classifies ring shapes based on IUPAC rules:

(C) Chair : Θ near 0 or 180° (poles of the sphere).
(B) Boat : Θ near 90°, Φ_2 at 0, 60, 120... (equator).
(T) Twist-Boat : Θ near 90°, Φ_2 at 30, 90, 150... (equator).
(E) Envelope : N = 5 rings with Φ_2 near 0, 36, 72...
(H) Half-Chair : intermediate states bridging major forms.

CONVENTIONS

- Mean plane normal: calculated using the left-handed cross product (n = R_{sin} x R_{cos}), as per the original Cremer-Pople paper to ensure agreement with other crystallographic software, such as *Platon*.
- Pivot atom: sets the starting point at the highest-priority non-carbon or lowest alphanumeric label, tracing toward the next lowest label. A cyclic forward shift of the pivot transforms Θ to 180 - Θ and shifts Φ_2 by +120° (for N = 6 rings).
- Ring traversal: reversing the sequence of atoms flips the latitude (Θ -> 180 - Θ) and longitude (Φ_2 -> 180 + Φ_2).

SEE ALSO

ZPAD (calculates Zefirov-Palyulin-Dashevskaya puckering parameters using internal torsion angles.)

REFERENCE

Cremer, D., & Pople, J. A. (1975). A general definition of ring puckering coordinates. *J. Am. Chem. Soc.*, 97(6), 1354-1358.

DRAW (Save graphics view to PNG file)

Saves the current 3D graphics view to a PNG image file. The output image is cropped to the visible content and a border is added.

USAGE

```
DRAW
DRAW [filename] [target_width]
```

ARGUMENTS

[filename] : The name of the output image file. If the .png extension is omitted, it will be added.

[target_width] : Optional. Specifies the horizontal resolution in pixels (default 5000 px) for a high-resolution export that bypasses the physical monitor constraints. The final .png file will usually be a bit smaller than 5000 px wide because blank space around the view are removed and a border is added.

NOTES

- **DRAW** without a filename leads to a prompt in the terminal.
- If [target_width] is not explicitly provided, the export resolution defaults to the DRAW_HORIZONTAL_PX setting in your configuration file.
- The image captures the exact current state of the graphics window, including the background colour (or gradient), atom representations (**PROJ**, **TELP**, **PNUT**, etc.), labels, and viewing orientation.

EDEN (Electron density)

Loads structure factor data from a *.fcf file (LIST 6 format) and calculates electron density maps via inverse fast Fourier transform. The resulting density can be visualised in 3D space or as a 2D slice. The 2D view includes a bounding box with 1Å tick marks.
Arguments can be provided in any order. The command supports multi-level sigma contouring, allowing several density levels for nested iso-surfaces. The densest core is brightest, and outer shells are progressively dimmed.
Multiple map types (e.g., Fobs, Fobs-Fcalc, 2Fobs-Fcalc) can be displayed simultaneously with their own distinct contour levels by issuing sequential **EDEN** commands.

USAGE

```
EDEN [1|2|3] [3D|2D] [CAGES|CLOUDS|COMBINED] [file.fcf]
[levels...] [OFF]
```

ARGUMENTS

[1|2|3] : Specifies the type of map. 1 = Fobs (observed electron density, default) 2 = Fobs-Fcalc (difference map) 3 = 2Fobs-Fcalc

[3D|2D] : The dimensional rendering mode. 3D renders the calculated volume (default). 2D calculates a cross-section slice.

[style] : The graphical representation of the density. Valid options are CAGES (wireframe/mesh, default), CLOUDS (solid/volume), or COMBINED (both).

[filename.fcf] : The target .fcf file. If omitted and only a single .fcf file exists in the working directory, it is loaded automatically. If multiple .fcf files are present, it looks for a .fcf with the same filename stem as the active structure (.res or .ins) file.

[levels...] : One or more floating-point values defining the sigma contour levels (e.g., 1.5 3.0 4.5). For Fobs-Fcalc (2), both positive and negative surfaces are generated. Levels exceeding the map's maximum density are ignored. If omitted or completely invalid, defaults are applied (1.5 for Fobs/2Fobs-Fcalc, 3.0 for Fobs-Fcalc).

[OFF] : Clears all active density maps, parameters, and visuals from the screen and memory.

TERMINAL OUTPUT

EDEN generates a status report with:
- Confirmation of LIST 6 data extraction from the parsed *.fcf file.
- Progress of the selected map's density grid calculation.

- Warnings if any specified sigma levels exceed the maximum/minimum density available in the calculated map.
- The final sorted list of valid sigma levels applied to the visualisation.
- Errors are reported if multiple *.fcf files are present without explicit specification, if cell parameters are invalid, or if the file cannot be located.

GRAPHICS

A successful **EDEN** command displays the calculated density in the graphics view and creates the density control buttons.
- In 3D mode, the map is rendered as isosurface overlays on the current structural model, with nested levels dimmed.
- In 2D mode, the density is projected onto a predefined geometrical plane with dynamic multi-level contour lines. 2D mode is zoomable, but can be a bit slow to respond.

NOTES

- Argument parsing is order-independent; commands, files, and contour levels can be provided in any sequence.
- Multiple map types can be active simultaneously. Issuing sequential **EDEN** commands (e.g., **EDEN 1** followed by **EDEN 2**) overlays them while respecting their respective contour levels.
- Rendering in 2D mode requires a defined plane. Thus, an **MPLN** command first establish the necessary mean plane, centroid, and normal vector for the slice.

ENVI (Environment)

Calculates and prints the coordination environment of the specified atoms, identifying all unique neighbours within a defined search radius.

USAGE

```
ENVI [delta] [atom_list]
```

ARGUMENTS

[delta] : A tolerance value (default 0.5 Å) added to the sum of the bonding radii (br1 + br2). The search radius is defined as (br1 + br2 + δ).
[atom_list] : The central atoms to investigate. If omitted, applies to all atoms in the current **FMOL** list.

OUTPUT

For each specified atom, **ENVI** lists:
- Distances to all neighbouring atoms within the search radius.
- All unique angles between these neighbours.
- The symmetry codes used to generate the ligand atoms.

NOTES

- Unlike **BANG**, this command does not use the existing connectivity table; it calculates proximity based strictly on Cartesian distance.
- **ENVI** is essential for exploring inorganic coordination polyhedra or identifying intermolecular contacts.
- Symmetry codes printed by **ENVI** can be used with the **SGEN** command to permanently add specific neighbouring atoms to the model.

EXIT (Exit program)

Terminates the session and returns to the operating system.

USAGE

```
EXIT
QUIT
```

NOTES

- In this version, **EXIT** and **QUIT** perform the same function.

- Ensure any important coordinate exports or structural changes are noted before exiting, as the current session state is not automatically saved.

EYES (Stereo/mono toggle)

Sets the default viewing mode for graphical representation.

USAGE

```
EYES n
```

ARGUMENTS

n : An integer defining the viewing mode. (Required)

- **1** : Mono (default)
- **2** : Anaglyph stereo (red/cyan)

NOTES

- This setting affects the 3D representation in the graphics window.
- The viewing mode can also be toggled instantly using the dedicated button in the graphical user interface.

FILE (Write FMOL list to SHELX file)

Writes the atoms in the current **FMOL** list to a new .ins file for the next round of *SHELXL* refinement.

USAGE

FILE [/F] filename

ARGUMENTS

filename : The name of the file to be written (e.g., project.ins).

QUALIFIER

/F : Force write. Bypasses the safety prompts and forces the program to write the file even if mathematical symmetry equivalents or Q-peaks are detected. Use with caution!

DETAILS

- Header instructions (**TITL**...**UNIT**) and residual instructions (**HKLF**, **END**, **WGHT**) are preserved from the original source file.
- Interspersed restraints/constraints (e.g., SAME, SADI, EADP) are bound to their respective atom(s) and preserved even if the list has been reordered. Nevertheless, it is always wise to check the new .ins file.
- Topology (**RESI**, **PART**, **AFIX**) is dynamically generated based on the current state of the atoms in memory.
- DISORDER FVAR LINKAGE: If the program detects unlinked atoms assigned to **PART 1** and **PART 2**, it will launch an interactive prompt to estimate the occupancy and link them via a free variable (FVAR).

SAFETY TRAPS

- OVERWRITE: If the target filename already exists, the program will prompt for confirmation before overwriting it.
- Symmetry equivalents: If symmetry equivalents (e.g., from **GROW**) are detected, the command will prompt for confirmation. Writing these above the **HKLF** line is potentially disastrous for *SHELXL*.
- Q-peaks: These are written below the **HKLF** line, so should be benign.

FMOL (Form molecule)

Defines the "active" atom list and generates the connectivity table (bonds) used for calculations and graphical representation.

USAGE

```
FMOL [atom_list] [delta]
FMOL/N [atom_list] [delta]
FMOL REST [delta]
```

ARGUMENTS

[atom_list] : The atoms to include in the active model. Can be specific labels, element types (e.g., $C), or ranges.

[delta] : A tolerance value (default 0.5 Å) added to the sum of the bonding radii (br1 + br2).

QUALIFIER

/N : "No output" mode. Suppresses the listing of atoms and bonds in the terminal.

KEYWORDS

REST : Swaps the current list. All atoms NOT currently in the **FMOL** list become the new active list.

NOTES

- *PrimeXP* performs a silent **FMOL** (**FMOL/N**) upon reading a file, creating an initial connectivity table.
- Only atoms in the **FMOL** list are available for subsequent commands like **BANG** or **ENVI**.
- Bonding is calculated based on Cartesian distance versus the current bonding radii (br) defined in the **ARAD** table.
- The resulting connectivity can be manually edited using **JOIN**, **PRUN**, and **UNDO**, or interactively modified using **PICK**.
- **FMOL** clears any previous active list before building the new one.
- Parameter parsing: A trailing number is interpreted as [delta] unless the selection string includes **RESI** or **PART** keywords.

FORE (Foreground colour)

Changes the text colour in the terminal.

USAGE

```
FORE
FORE [colour]
```

ARGUMENTS

`[colour]` : A 3 or 6-digit hexcode (e.g., #000 or #000000 for black, #FFF or #FFFFFF for white) or one of the 148 named colours in "rsrc_colours.py".

NOTES

- Entering **FORE** without arguments reports the current foreground colour to the terminal.
- **FORE** only affects the text in the terminal window, not the 3D graphics window (see **BACK** for background settings).

FUSE (Symmetry reduction and cluster packing)

Reduces the current atoms to a unique asymmetric unit by identifying and removing symmetry equivalents. This restores a connected asymmetric unit, but the option to find a tighter cluster via fragment docking is available.

USAGE

```
FUSE [delta]
```

ARGUMENTS

[delta] : Optional. A distance tolerance in ångströms (default 0.5) used to determine if two atoms are symmetry equivalents overlapping at the same site.

NOTES

- Dummy atoms: dummy atoms (labelled **'X'**) are purged before **FUSE** executes.
- Hydrogens: Hydrogen atoms are strictly grouped with their parent heavy atoms to ensure they are not separated during symmetry transformations.
- Q-Peaks: These are excluded from the initial reduction and are instead snapped as a single block to the main cluster centroid at the very end of the process.
- Fragment docking: if the asymmetric unit consists of disjointed fragments, the heaviest fragment can be frozen as a static anchor, and all remaining fragments are translated via symmetry to dock close to it. If a fragmented asymmetric unit is detected, a prompt allows the user to bypass this docking to preserve the original spatial arrangements.
- **FUSE** overwrites the master atom list and rebuilds the connectivity table.

GROW (Grow symmetry equivalents)

Generates symmetry-equivalent atoms that are chemically bonded to the specified seed atoms, effectively completing molecules that lie across special positions or symmetry elements.

USAGE

```
GROW [delta] [atom_list]
```

ARGUMENTS

[delta] : A tolerance value (default 0.5 Å) added to the sum of the bonding radii (br1 + br2).

[atom_list] : The seed atoms to grow from. If omitted, applies to all atoms currently in the active **FMOL** list.

NOTES

- New atoms have a letter appended to the label (e.g., C1 generates C1A, then C1B, etc.).
- The newly generated atoms are added to both the active **FMOL** list and the master atom pool.
- **GROW** contains safety checks to prevent generating overlapping atoms or runaway polymer chains.
- Graphical reset: Because **GROW** recalculates the connectivity table, it automatically turns off full-radii drawing modes (like **PERS**).
- For assembling disconnected molecular fragments using symmetry without relying on bonding radii, see the **FUSE** command.

HADD (Add hydrogen atoms)

Generates idealized hydrogen atoms attached to specified targets (C, N, O, B, P, S). *PrimeXP* uses a "virtual grow" environment via a KD-Tree, allowing it to correctly determine the coordination sphere of atoms on special positions or near symmetry boundaries without requiring a manual **GROW** command.

USAGE

```
HADD
HADD [type] [selection]
HADD [type] [distance] [selection]
```

ARGUMENTS

[type] : An integer defining the geometry. If omitted, *PrimeXP* enters Auto-Fill mode and guesses based on geometry and valency.

- **1** : Tertiary C-H (R_3CH).
- **2** : Secondary CH_2 (R_2CH_2).
- **3** : Methyl CH_3 or tetrahedral ammonium $-NH_3^+$.
- **4** : Aromatic/Vinyl sp^2 C-H.
- **8** : Hydroxyl O-H. Searches for nearest H-bond acceptor.
- **9** : Terminal $=CH_2$ or planar amine/amide $-NH_2$.
- **15** : Borane/Cage radial H (Negative vector sum).
- **16** : Acetylenic sp C-H.

[distance] : Optional specific X-H bond length (Å). If omitted, *PrimeXP* uses standard low-temperature X-ray riding model defaults configured in rsrc_config.py (e.g., C-H = 0.95 to 1.00 Å).

[selection]: Atoms to protonate. If omitted, targets all heavy atoms.

NOTES

- File preparation: If hydrogen is not already present in the structure, the program appends **'H'** to the SFAC and **UNIT** instructions.
- Auto-fill: Intelligent guessing based on bond distances and topological angles. For example, if an aromatic C-C bond is unrefined and appears too long (acting like a single bond), **HADD** will double-check the bond angle. If it falls within the sp^2 range (configurable in rsrc_config.py, default 115°-130°), it overrides the distance check and assigns a single planar hydrogen instead of a methylene group.
- **AFIX** codes: Generated hydrogens are assigned a suitable *SHELXL* **AFIX** constraint (e.g., 13, 23, 137, 43, 83, 153) corresponding to their geometric type, so they are ready for refinement.

- Hydroxyls (type 8): *PrimeXP* searches the virtual crystal structure for nearby O/N/Halogen H-bond acceptors. If none are found, it falls back to a staggered geometry. You might want to replace any **`AFIX 83`** with **`AFIX 147`** prior to refinement.
- Cages/clusters (type 15): Boron atoms with 4+ bonds, and carbon atoms bonded to 3+ borons, bypass standard valency checks and are assigned radial hydrogens (**`AFIX 15`**) in Auto-fill mode.
- Ionic filter: If metallic/ionic neighbors (e.g., Na, K, Ru) are detected near the target, the program will prompt you [Y/N] to ignore them so they do not distort covalent vector sums.
- Inheritance: Generated hydrogens inherit the **`PART`** and **`RESI`** numbers of their parent atom.
- U_{iso}: Isotropic displacement parameters are assigned based on type (-1.5 for methyl/hydroxyl, -1.2 for all others).
- Parsing: The argument order is flexible; the command distinguishes integer types, float distances, and atom labels.

HBND (Hydrogen bonds)

Searches for and generates bonds representing hydrogen bonding interactions based on geometric criteria (distance and angle).

USAGE

```
HBND
HBND 1 [join_code]
HBND 2 [join_code_1] [join_code_2]
```

ARGUMENTS

`[mode]`	**:**	An integer defining the search mode: **`1`** **:** Searches only for conventional H-bonds (default). **`2`** **:** Searches for both conventional and weak H-bonds.
`[join_code_1]`	**:**	Bond type for normal H-bonds (default solid dashes).
`[join_code_2]`	**:**	Bond type for weak H-bonds (default hollow dashes).

NOTES

- Geometric tolerances: Conventional: D-H...A angle >= 120.0°, H...A distance <= 3.0 A. Weak (C-H...A): D-H...A angle >= 110.0°, H...A distance <= 3.0 A.
- Donors and acceptors: Conventional donors include O, N, F, S, Cl, Br, I. Weak donors include C. Acceptors include O, N, F, S, Cl, Br, I.

- Configuration: The exact distance thresholds, angle cutoffs, and lists of valid donor/acceptor elements can be customized in the "rsrc_config.py" file.

`HELP` (Command documentation)

Provides detailed information and usage instructions for *PrimeXP* commands.

USAGE

```
HELP [command name]
```

ARGUMENTS

`[command]` **`:`** Four-letter command name.

NOTES

- **`HELP`** without a command name displays a list of available commands.
- The documentation includes syntax, argument descriptions, and practical tips for command usage.

HIMP (Hydrogen improve)

Adjusts hydrogen atom positions along their existing bond vectors to a target distance.

USAGE

HIMP [distance] [atom_list]

ARGUMENTS

[distance] : The desired bond length (in ångströms).

[atom_list] : The atoms to be processed. If omitted, applies to all hydrogen atoms currently in the **FMOL** list.

DEFAULT DISTANCES

If [distance] is not specified, the following defaults are used based on the atom to which the hydrogen is bonded:

- **C-H** : 0.96 Å
- **N-H** : 0.90 Å
- **O-H** : 0.85 Å

NOTES

- Smart selection: You do not need to explicitly list the hydrogen atoms. If you specify a non-hydrogen parent atom (e.g., **'HIMP C1'**), *PrimeXP* finds and adjusts all hydrogen atoms bonded to it.
- The command requires an existing connectivity table to determine the bond vector.
- Cognoscenti hack: To regularise the bond length of a non-hydrogen atom, temporarily rename it to H, remove all but one bond, run **HIMP**, and then change the name back.
- Hydrogens bonded to elements other than C, N, or O will only be moved if an explicit distance is provided in the command.

`HIST` (Save terminal history)

Writes all terminal output since the start of the session to a text file.

USAGE

```
HIST [filename]
```

ARGUMENTS

`[filename]` **`:`** The filename for the history dump. If omitted, *PrimeXP* names it based on the currently loaded structure file (e.g., "my_project-history.txt"). If no file is loaded, it defaults to "primexp_history.txt".

NOTES

- If no extension is provided, ".txt" is appended.
- Useful for keeping a record of your structural refinement steps, geometry calculations, command history, etc.

`IMPT` (Import configuration)

Reloads configuration settings and user preferences, and dynamically updates the graphics window.

USAGE

```
IMPT [ALL]
```

NOTES

- **`IMPT`** (no argument): Reloads only your personal user preferences defined in the "~/.*PrimeXP*.cfg" file. This lightweight refresh applies your custom settings while preserving the current dynamic display parameters, custom colours, and **`ATYP`** assignments.
- **`IMPT`** ALL: Performs a complete "Scorched Earth" factory reset. This reloads only the base global settings ("rsrc_config.py", "rsrc_colours.py", and "rsrc_radii.py") and intentionally ignores your user preferences, returning all display parameters and logic to their original shipped defaults.
- Both modes trigger a recompilation of the OpenGL shaders, making it useful for testing custom GLSL code or advanced rendering tweaks during a live session.

INFO (Model information)

Displays a summary of atom parameters, coordinates, *SHELXL*-style occupancies, U_{eq}, and Q-peak heights for the specified list.

USAGE

```
INFO [atom_list]
```

ARGUMENTS

[atom_list] : If omitted, applies to all atoms in the current **FMOL** list.

OUTPUT

The command prints a formatted table containing:
- Atom : The atom name (label).
- SFAC : The scattering factor index.
- x, y, z : Fractional coordinates.
- sof : Site occupancy factor.
- U_{eq} : Equivalent isotropic displacement parameter.
- Peak : Q-peak heights (displayed only if Q-peaks are in the selection).

NOTES

- This command provides a direct output of the model's positional data.
- At the end of the listing, the program reports the number of atoms displayed versus the total number of active atoms in the session.

INVT (Invert structure)

Inverts the coordinates of specified atoms through a defined point in fractional space.

USAGE

```
INVT [Xi Yi Zi] [atom_list]
```

ARGUMENTS

[Xi Yi Zi] : The fractional coordinates of the inversion centre. The default is (0, 0, 0).

[atom_list] : The atoms to be inverted. If omitted, applies to all atoms currently in the **FMOL** list.

NOTES

- **INVT** can produce the mirror image of a structure or correct absolute configuration.
- Atoms outside the current **FMOL** list or not specified are left unchanged.

ISOT (Isotropic conversion)

Converts the displacement parameters of anisotropic atoms to U_{eq}, or forces them to a specified U_{iso} value.

USAGE

```
ISOT [val] [atom_list]
```

ARGUMENTS

[val] : Optional. A numeric value to assign as the isotropic displacement parameter (U_{iso}) for the selected atoms.

[atom_list] : If omitted, applies to all atoms in the current **FMOL** list.

NOTES

- If [val] is omitted, anisotropic atoms have their new isotropic displacement parameter (U_{iso}) set equal to the calculated equivalent value (U_{eq}). Atoms that are already isotropic remain unaffected.
- If [val] is provided, ALL specified atoms are forced to this new U_{iso}.

- This is a common step before exporting coordinates for certain types of structural analysis or when simplifying a model.

JOIN (Create or modify bonds)

Adds a bond to the connectivity table or modifies the visual style of existing bonds for the specified atoms.

USAGE

```
JOIN [bondtype] [atom_list]
JOIN/S [bondtype] [atom1] [target_list]
JOIN/X [bondtype] [number] [hub_atoms] [target_atoms]
JOIN/P [bondtype] [number] [hub_atoms] [target_atoms]
```

ARGUMENTS

[bondtype] : An integer (1-9) defining the line style. Default is 1.
[atom_list] : The atoms to be joined or their bond style modified.
[atom1] : Central "hub" atom (required when using **/S**).
[target_list] : Atoms that connect to the hub (required for **/S**).
[number] : Max number of bonds to form per atom (for **/X** and **/P**).
[hub_atoms] : Source atoms for **/X** and **/P** joins.
[target_atoms]: Destination atoms for **/X** and **/P** joins.

QUALIFIERS

/S : Creates a "star" join. Bonds the single hub atom ([atom1]) to every atom in the subsequent [target_list].
/X : Nearest-neighbor join. Connects each hub atom to its [number] nearest target atoms based on Cartesian distance.
/P : Pi-stacking join. Connects hubs and targets globally, up to [number] bonds per atom, within configured pi-stacking distance limits (default 2.9 Å - 4.1 Å).

BOND TYPES

1 : Standard thick (default)
2 : Hollow/halo thick (background coloured interior, coloured perimeter)
3 : Standard dashed thick
4 : Hollow/halo dashed thick
5 : Standard thin solid
6 : Dashed thin solid
7 : Dotted
8 : Darker thick
9 : Darker thick dashes

BEHAVIOUR

- Star **JOIN** (**/S**): Creates a bond from the hub atom to every target atom. The target list fully supports standard selection logic.
- Nearest **JOIN** (**/X**): Evaluates distances dynamically and generates bonds to the closest specified targets.
- Pi-stack **JOIN** (**/P**): Assigns bonds between hub and target pools while respecting the [number] quota per atom and distance limits.
- Exactly two atoms: Creates a bond or updates the bondtype.
- Other atom counts: Changes the bond type for ALL existing bonds connected to the listed atoms.

NOTES

- **JOIN** is used to create bonds that fall outside the threshold (br1 + br2 + δ), such as long coordination bonds.
- These styles are respected in both terminal calculations and the 3D graphics window (**PROJ**, **TELP**, **PNUT**, etc.).
- To remove a bond(s) entirely, use **UNDO** or **PRUN**.
- The **'Bond edit'** button may also be used.

KILL (Remove atoms)

Removes the specified atoms from the model and the connectivity table.

USAGE

```
KILL [atom_list]
KILL REST
KILL ALL
```

ARGUMENTS

[atom_list] : The atoms to be deleted.

KEYWORDS

REST : Eradicates all atoms that are NOT in the current **FMOL** list. Useful for purging symmetry-generated junk or unwanted solvent molecules after isolation.

ALL : Eradicates ALL atoms in the session memory.

NOTES

- When an atom is removed, all associated bonds in the connectivity table are deleted.
- The atom list is re-indexed to fill the gaps left by the removed atoms.
- Caution: **KILL** is permanent for the current session.
- If the model includes Q-peaks, a **'Kill $Q'** button is present.

LABL (Atom labels)

Controls the display and size of atom labels in the graphics window. Also changes the colour (but not size) of the gnomon axis labels. When a light or dark background is set via the **BACK** command, the label colour automatically changes to black or white, which may be overridden with **LABL**.

USAGE

```
LABL [code] [size] [colour] [selection]
(Arguments can be provided in any order, e.g., 'LABL
yellow 400 1', or
'LABL c? f*')
```

ARGUMENTS

[code] : An integer defining the labelling mode:

- **0** : No labels.
- **1** : Non-hydrogen atoms (default).
- **2** : Non-hydrogen atoms in the asymmetric unit only.
- **3** : All atoms, including H.
- **4** : All atoms, including H, but only the asymmetric unit.
- **5** : Heavy atoms and H (or D) bonded to non-carbon.
- **6** : Heavy atoms and H (or D) bonded to non-carbon, but only the asymmetric unit.
- **7** : Custom labels (enabled automatically when specifying a selection or when toggling labels via mouse clicks in the graphics window).

[size] : The vertical dimension of the labels. The default is 400 (equivalent to 24pt), so 600 is about 36pt, but these are also affected by the field-of-view angle and the initial zoom level, which are all set via the "rsrc_config.py" file.

[colour] : A three or six digit hexcode or one of the named colours in the file "rsrc_colours.py".

[selection] : Standard atom selection tokens (e.g., C? N1 O*). If provided, labels are turned on exclusively for these atoms, and the mode switches to 7 (Custom labels).

NOTES

- Flexible parsing: The **LABL** command accepts size, colour, mode, and selection tokens in any order.

- Interactive toggling: Clicking an atom or its label in the graphics window while in toggle mode will hide/restore it, pivoting the view to mode 7 (Custom labels) to preserve currently visible labels.
- Persistence: Label positions update if atoms are moved, e.g. by **`HIMP`**.
- Be careful with synchronization of the **`'H atom on/off/no H'`** button.

LIBR (Thermal motion analysis)

Performs rigid-body thermal motion analysis using the Schomaker and Trueblood method [*Acta Cryst.*, B**24** (1968) 63-76] to calculate libration-corrected bond lengths. The rigid group criteria are strict. Impossible "corrections" for inappropriate (i.e., non-rigid) groups give a warning, and further analysis is aborted by default.

USAGE

```
LIBR [atom_list]
LIBR OFF | CLEAR
```

ARGUMENTS

[atom_list] : At least three atoms with U^{ij} parameters (ADPs) are needed. Isotropic atoms are ignored. If omitted, applies to all atoms in the current **FMOL** list.

OFF : Deactivates the graphical TLS dashboard and restores the standard graphics view.

TERMINAL OUTPUT

- The generalised R-index for agreement between observed and calculated orthogonalised U^{ij} values.
- The T, L, S tensors (Translation, Libration, Screw) evaluated at the rigid-group centroid.
- The shifted T, S tensors evaluated at the idealised **'Centre of Reaction'** (CoR). Note: The L-tensor is origin-invariant.
- The T, L, S tensors projected onto a Local Inertial Frame (LIF), as per established tools like *Platon* and *THMA11*. (Note: to reproduce mass-weighted results (as in *Platon*), use **WAIT/M** to assign weighting by atomic mass prior to **LIBR**).
- Libration-corrected bond lengths for all unique bonds between the specified atoms.

GRAPHICAL DASHBOARD

A successful **LIBR** command opens the interactive 3D TLS dashboard, splitting the graphics window into four quadrants. The visualizer centres the origin on the Centre of Reaction (CoR):

- Q1 (Top Left) : Rigid group (model) ellipsoid view.
- Q2 (Top Right) : Translation (T-tensor) - sliding motion of the rigid body (dimensions are distance²).

- Q3 (Bottom Left) : Libration (L-tensor) - rotary oscillation (wobble) of the body (dimensions are angle²).
- Q4 (Bottom Right) : Screw (S-tensor) - correlated coupling between translation/libration (dimensions distance·angle). Shifting the origin to the CoR, symmetrizes the S-tensor. The tensor **`'shape'`** is visualised as an SVD-ellipsoid (Martin & Porter, 2011; Kindlmann, 2004). This SVD-ellipsoid portrays the the S-tensor relative magnitude in 3D, but loses the screw chirality. The handedness is analytically recovered by projecting the screw pitch ($p = n^T S n$) along the surface normal and mapped to surface colouring by **`'Directionally Encoded Colour'`** (DEC) mapping (Pajevic & Pierpaoli, 1999): right-handed (red by default) and left-handed (lemon-yellow). The colours can be set in rsrc_config.py.

Note: By default, T and L are drawn with an octant cutout to reveal their principal axes, but for the S tensor, this can partially obscure the subtle colour variation. The cutouts can each be toggled on/off by a button in the lower right of each quadrant.

REFERENCES

Kindlmann, G. (2004). Superquadric tensor glyphs. *Proc. Joint Eurographics-IEEE* TCVG Symposium on Visualization.

Martin, C. D., & Porter, M. A. (2011). The extraordinary SVD. *arXiv.* https://doi.org/10.48550/arxiv.1103.2338

Pajevic, S., & Pierpaoli, C. (1999). Color schemes to represent the orientation of anisotropic tissues from diffusion tensor data. *Mag. Res. Med.*, 42(3), 526-540.

Schomaker, V., & Trueblood, K. N. (1968). On the rigid-body motion of molecules in crystals. *Acta Cryst.* B**24**(1), 63-76.

LINE (Calculate line between two atoms)

Calculates the line passing through two specified atoms in both crystal and orthogonal coordinates.

USAGE

```
LINE atom1 atom2
LINE/N atom1 atom2
```

ARGUMENTS

atom1, atom2 : The two atoms defining the line.

QUALIFIER

/N : "No output" mode. Stores the line but does not write details.

OUTPUT

- The length of the line segment between the two atoms (in ångströms).
- The equation of the line in fractional (crystal) and orthogonal (Å) coordinates.
- Angles between this new line and any previously defined lines, planes (see **MPLN**), or orientations (**MATR**) are written to the terminal.

NOTES

- View orientation: Upon execution, the graphical view is oriented to look directly down the calculated vector.
- All defined lines are stored for subsequent angle calculations. Use the **NOPL** command to clear the stored geometry list.

MATR (Orientation matrix)

Resets or displays the orientation matrix for graphical views.

USAGE

```
MATR [9 numbers]
MATR [n]
MATR [h k l]
```

ARGUMENTS

`[9 numbers]` : Manually sets the 3x3 orientation matrix P (p11...p33).

`[n]` : Sets a standard view down a real crystal axis:

- `1` : View down the crystal a (x) axis.
- `2` : View down the crystal b (y) axis.
- `3` : View down the crystal c (z) axis.

`[h k l]` : Sets the view perpendicular to the lattice plane defined by the Miller indices (h, k, l).

OUTPUT

- When called with no arguments, the current 3x3 orientation matrix is written to the terminal.
- The angles between the specified orientation matrix and any previously defined vectors (from **LINE**), or mean planes (**MPLN**) are written to the terminal.

NOTES

- **MATR** displays or changes the orientation matrix, not the actual atomic coordinates.

MGEN (Molecule generation/packing)

Generates symmetry-equivalent molecules to fill a defined volume in fractional space. Primarily used to prepare models for packing plots.

USAGE

MGEN [atom_list] [delX delY delZ] [Xc Yc Zc]

ARGUMENTS

[atom_list] : The atoms used as the basis for the search. If omitted, defaults to ALL (can be slow for large **FMOL** lists).

[delX delY delZ] : The half-widths of the search volume (default 0.5).

[Xc Yc Zc] : The centre of the search volume (default 0.5, 0.5, 0.5).

MECHANISM

1. Molecule definition: **MGEN** identifies discrete molecules based on the current connectivity table (all defined bonds). These connections are used to "assemble" equivalent units.
2. Generation: Symmetry transformations are applied. If any atom in the specified list falls within the box (Xc ± delX, etc.), the entire associated molecule is generated.
3. Replacement: The generated contents of the packing box completely replace the previous session memory. The **FMOL** list will consist entirely of the newly generated assembly.

NOTES

- Default volume: Without arguments, **MGEN** fills exactly one unit cell, plus the rest of any molecule that penetrates the defined volume.
- Symcodes & Visuals: The program prints the symmetry codes used for each transformation. **MGEN** combines symmetry operations to preserve the internal symmetry (and thus the distinct **COLR SYMM** colours) of the parent molecules.
- Pre-requisites: For optimal rendering performance and accurate mapping, it is best to run **FUSE** and generate any **VOID** maps prior to **MGEN**.
- Efficiency: For large structures, specify only one central atom per unique molecule/ion to speed up the search.
- Special positions: Ensure molecules on special positions are completed using **GROW** before invoking **MGEN**.

- Iteration: If the result is not right, run **FMOL** and/or **FUSE** to reset your base molecule before calling **MGEN** again.

MPLN (Mean plane calculation)

Calculates the least-squares plane through a specified set of atoms and sets the view orientation to be perpendicular to that plane.

USAGE

```
MPLN [atom_list]
MPLN/N [atom_list]
```

ARGUMENTS

[atom_list] : The atoms used to define the plane. If omitted, applies to all atoms in the current **FMOL** list.

QUALIFIER

/N : "No output" mode. Calculates the plane and resets the view without writing details to the terminal.

OUTPUT

- The equation of the plane and the deviations of individual atoms from the plane (in ångströms).
- The angles between the current plane normal and any previously defined normals or vectors (from **LINE**), or orientation matrices (**MPLN**) are written to the terminal.

NOTES

- View orientation: Upon execution, the graphical view (**PROJ/TELP/PNUT**) is oriented to look down the normal to the plane.
- This is the standard method for obtaining a "face-on" view of a planar molecule or ring system.
- Like **MATR**, this command updates the orientation matrix.
- All defined planes are held in memory for subsequent angle calculations. Use the **NOPL** command to clear the stored geometry list.

NAME (Name or rename atoms)

Changes the names of specified atoms and updates their physical and graphical properties based on the element symbol.

USAGE

```
NAME oldname newname [sfac]
```

ARGUMENTS

oldname : The current name of the atom(s). Supports ? and * wildcards.
newname : The new name for the atom(s). Also supports the wildcards.
[sfac] : Optional. An integer SFAC index of the renamed atom(s).

BEHAVIOUR

- Element detection: The program deduces the new element from the first one or two characters of **'newname'**.
- Upon renaming, the atom's colour, **ATYP** code, and radii (**ARAD**) are reset to the defaults for the new element.
- Validation: **NAME** will fail if it would result in duplicate labels.
- Truncation: New labels are truncated to the configured maximum length (default is 4 characters for standard *SHELX* compatibility).

WILDCARDS

- The **'?'** character matches any single character in a specific position. Example: **'NAME Q?? C??'** changes Q10 to C10, but leaves Q1 untouched. Note: When using **'?'**, the lengths of the old and new patterns must match.
- The **'*'** character acts as a prefix/suffix wildcard. Example: **'NAME C* N*'** changes all atoms starting with **'C'** to start with **'N'** (e.g., C12 becomes N12, C2A becomes N2A).

NOTES

- This is the standard way to convert Q-peaks (from a structure solution or a difference map) into bona fide atoms.
- Residue information is not affected by **NAME**.

NEWM (Newman projection)

Calculates the geometry and torsional angles for a Newman projection along a specified bond. The orientation is not set to view down the specified bond, but this can be achieved with **LINE**.

USAGE

```
NEWM atom1 atom2
```

ARGUMENTS

atom1 : The front (top) atom of the projection.
atom2 : The rear (bottom) atom of the projection.

OUTPUT

- Lists all torsional angles for substituents attached to the bond.

NOTES

- Rotational convention: This implementation follows the IUPAC standard.

REFERENCE

Newman, M. S. (1952). A convention for the representation of spatial arrangement of substituent groups. *Record Chem. Progr.*, 13, 111-116.

NEXT (Restore graphical view and state)

Restores a previously saved graphical session state from a .sav file.

USAGE

```
NEXT
NEXT [filename]
```

ARGUMENTS

[filename] : The name of the saved session file to load. If omitted, the command defaults to "view_state.sav".

BEHAVIOUR

- Data restoration: **NEXT** reads the designated file and reconstructs the saved scene (coordinates, occupancies, U-values, element types).
- Connectivity: It completely reinstates the saved connectivity table, preserving custom bond types and unmodified distances.
- Full view state: Re-instates all aesthetic and viewing parameters, including:
    - The 3D orientation matrix and projection mode.
    - The background colour or gradient.
    - Global drawing styles (atom mode, bond blending, cell visibility).
    - Displacement ellipsoid probability (ADP) settings.
    - Any manual label tweaks (custom dragged positions or hidden labels).

NOTES

- A .sav extension is appended if missing.
- **NEXT** reads both the default compressed .sav files and uncompressed long-form JSON files (created via **SAVE/L**).
- Overwrites the active session and refreshes the graphics window.
- **NEXT** pairs specifically with the **SAVE** command.

NOPL (Clear lines and planes)

Clears all currently stored geometric lines and planes from the session memory.

USAGE

NOPL

NOTES

- **NOPL** resets the internal list used for calculating angles between structural features.
- Use **NOPL** before defining a new set of vectors (via **LINE**) or least-squares planes (via **MPLN**) to ensure that subsequently calculated angles only refer to the current group of interest.
- **NOPL** is purely a housekeeping command for geometric data, it does not affect atom coordinates or connectivity.

OFIT (Structure overlays)

A version of **OFIT** is not included here. The original *XP* **'OFIT'** command was limited to just two structures. Far superior third party tools are available:
Mercury: Part of the CCDC CSD-System, widely used for structural visualization and comparison.
Over-lay: Dedicated programs (including *over-rip* and *over-rot*) for multiple structure overlays. See: https://xray.uky.edu/Tutorials/structure-overlays/1-Introduction.html

---

ORTH (Write file with orthogonal coordinates)

Exports the current **FMOL** list to a file using orthogonal coordinates (Å). The output is centred with the first atom at the origin and reflects the current orientation (see **MATR**).

USAGE

```
ORTH filename
ORTH/F filename
```

ARGUMENTS

filename : The target file for export.
- No extension: Defaults to **'.ort'** (*XP* ASCII format).
- **.ort :** Writes atoms, **ATYP** codes, and the connectivity table.
- **.xyz :** Writes a standard XYZ Cartesian coordinate file.
- **.mol :** Writes a MDL Molfile.

QUALIFIER

/F : (Full) Includes the crystallographic unit cell, symmetry operators, and transformation matrices.

NOTES

- Multi-format support: Unlike the original *XP*, this implementation switches formats based on the file extension provided.
- Integration: Use **'.xyz'** or **'.mol'** for transferring models to external molecular mechanics or visualization software.

PART (Assign atoms to a disorder component)

Assigns the specified atoms to component n of a disordered group. Affects bonding and symmetry generation behaviour.

USAGE

```
PART n [atomnames]
```

ARGUMENTS

n : The disorder component number (an integer). The default for atoms is 0 (not disordered).

[atomnames] : If omitted, applies to all currently selected/active atoms.

NOTES

- Bonding rules: No bonds are formed between atoms with different **PART** numbers unless one of the **PART** numbers is 0. Atoms with **PART 0** can bond to any component.
- Symmetry rules: When the **GROW** command is applied to atoms with negative **PART** numbers, no symmetry equivalents are generated.
- Selection: The **PART** keyword can be used within atom selection strings to filter groups. For example, **'FMOL C11 > N2 LESS PART 2'** includes all atoms in that range except those in disorder component 2.

PERS (Perspective ball-and-stick view)

A 3D view for visualizing the current **FMOL** list with spheres scaled as the first **ARAD** parameter. Unlike *XP*, real-time rotation and zoom by mouse action. The unit cell may be toggled via the "Unit cell" button. By default, non-H atoms are labelled.

USAGE

PERS [atom_list]

ARGUMENTS

[atom_list] : If omitted, defaults to the entire **FMOL** list.

REPRESENTATION MODES

While in **PERS** mode, the user can cycle through four distinct visual styles:

1. Wireframe : Simple line-based connections.
2. Stick : Solid cylinders connecting atom centres.
3. Hybrid : A combination of sticks and scaled spheres.
4. Ball & Stick : Full spheres and cylinders.

NOTES

- Scaling: The "Hybrid" and "Ball & Stick" modes use the individual atom radii (arad) to scale the spheres relative to the model.
- Performance: The view is updated in real-time, allowing for immediate visual feedback when modifying the orientation or selection.
- See also **PROJ**, **SPIX**, **TELP**, **PNUT**.

PICK (Interactive editing)

Enters an interactive graphical mode for identifying, renaming, or removing atoms from the current **FMOL** list. **PICK** starts from the bottom of the current atom list and works up, allowing the user to keep, reject, or rename atoms. This sequence may be interrupted with a mouse click on an atom, which makes that atom active, thereby resetting the **PICK** sequence.

USAGE

```
PICK [atom_list] [/H]
```

ARGUMENTS

[atom_list] : The atoms to be processed. Defaults to the entire **FMOL** list.
/H : Includes H atoms in the picking sequence (default: off).

INTERACTIVE FEEDBACK and COMMANDS

- [Blink] : Each atom in the list (and its bonds) blinks in turn to indicate it is the active atom.
- [Enter] : Deletes the blinking atom and its associated bonds.
- [Space] : Retains the blinking atom with its existing name.
- [New name] : Renames the atom (e.g., typing "C12" and hitting Enter). Properties like colour and radii are updated.
- [Delete] : Steps back to the previous atom, reinstating it if it was deleted. (Note: Names are not reverted).
- [/] : Exits **PICK** mode early while saving all changes made.
- [Esc] : Emergency exit; cancels all deletions from the current **PICK** operation.

STATUS LINE

The status line shows real-time data for the active atom, including its label, peak height (for Q-peaks), and bond lengths to its neighbours.

PNUT (Bürgi PEANUT plots)

Anisotropic atoms are represented by Bürgi-style *PEANUT* plots, which portray displacements as radius vectors, creating pinched dumbbell (prolate) or corpuscle (oblate) ovaloid shapes that visually highlight the directions of maximum and minimum displacement. Like **TELP**, **PNUT** is fully 3D interactive.

USAGE

```
PNUT [mode] [scale]
```

ARGUMENTS

[mode] : Optional. Determines the representation style:
1 : Line-drawn root-mean-square displacement (RMSD) (default).
2 : Line-drawn mean-square displacements (MSD).

[scale] : Optional. A numeric multiplier to adjust the surface size. If only one number is provided and it is not 1 or 2, it is treated as the scale factor.

NOTES

- Anisotropy: Atoms with anisotropic displacement parameters are drawn as *PEANUT* surfaces [RMSD (4th order) or MSD (6th order)].
- Scaling: Plots display true physical displacements independent of thermal probability levels. Surface sizes can be independently adjusted by passing a scale factor directly (e.g., **'PNUT 1.5'** or **'PNUT 2 1.5'**), which updates the PNUT_SCALE_RMSD or PNUT_SCALE_MSD configuration parameters.
- Isotropic fallback: Isotropic atoms and hydrogens do not have *PEANUT* representations and are rendered as small spheres/circles (e.g., **ATYP 2** for H atoms) based on your configuration.
- Publication styling: The **PNUT** command prepares the scene for publication by switching to a white background, setting bond colours to grey, and adjusting to publication-sized labels.
- Orientation: The plot uses the current viewing orientation as defined by the last **ROTA**, **LINE**, **MPLN**, **MATR**, etc. command.

REFERENCE

Hummel, W., Hauser, J., & Bürgi, H. B. (1990). *PEANUT*: Computer graphics program to represent atomic displacement parameters. *J. Mol. Graphics*, 8(4), 214-220.

POLY (Polyhedral plots)

Generates graphical representations of coordination geometries by drawing polyhedral convex hulls around selected central atoms. The command identifies bonded ligands, calculates the 3D faces connecting them, and reports the coordination number, number of faces, and volume for each polyhedron. The resulting polyhedra are rendered with semi-transparent faces and wireframe edges along with the atoms and bonds present in the structure.

USAGE

```
POLY [central_atoms] [/K]
POLY OFF
```

ARGUMENTS & FLAGS

[central_atoms] : Defines the atoms to be used as polyhedral centres (e.g., $Ru $Ca). If omitted, **POLY** will evaluate all active atoms in the current fragment.

/K : Kills (removes) any active atoms that are "dangling" (i.e., not a central atom or a vertex of a polyhedron).

OFF | CLEAR : Turns off polyhedral rendering and clears the shapes.

NOTES

- Auto-expand: If a central atom sits on a symmetry element or at the edge of the active fragment, **POLY** will attempt to **GROW** the missing symmetry-equivalent ligands to complete the coordination sphere.
- Topological exclusion: To prevent spurious metal-metal edges (e.g., in oxides), atoms of the same element(s) as the selected central atoms are prevented from acting as polyhedral vertices.
- Geometric rules: Polyhedra require a minimum number of non-coplanar ligands (default 4, configurable via POLY_MIN_LIGANDS). Additionally, the central atom must be strictly contained within the resulting hull.
- Customization: Face opacity, edge visibility, and edge thickness can be adjusted in the configuration file (rsrc_config.py).

PREV (Restore previous orientation)

Re-instates a previous viewing orientation by popping the last entry from the orientation history stack.

USAGE

PREV

MECHANISM

- History stack: The program saves the orientation matrix whenever a command changes the view (e.g., **ROTA**, **MATR**, **MPLN**, **LINE**).
- Limit: The history limit defaults to 10 previous orientations (defined by PREV_HISTORY_LIMIT in rsrc_config.py).

NOTES

- Repeatedly calling **PREV** allows you to step backward through your viewing history.
- This information is session-specific and is not stored in project files or external exports.
- If the stack is empty, the command will notify you that no more previous orientions are available.

PROJ (Interactive projection)

A dynamic 3D view for visualizing the current **FMOL** list. **PROJ** defaults to stick mode, but wireframe, hybrid and ball & stick are available via the click button. For the latter two, sphere radii (in **PERS**) are further scaled down (default = 0.5, set in rsrc_config.py). Unlike *XP*, real-time rotation and zoom via the mouse, are available. By default, only non-hydrogen atoms are labelled.

USAGE

```
PROJ [atom_list]
PROJ [CELL | /CELL] [atom_list]
```

ARGUMENTS

[atom_list]	:	The atoms to include. Defaults to the entire **FMOL** list.
CELL	:	Displays the unit-cell outlines. (Also accepts **/CELL** flag).

REPRESENTATION MODES

- Default style: **PROJ** defaults to Ball & Stick mode.
- Scaling: When using Hybrid or Ball & Stick modes, sphere radii (first **ARAD** parameter) are scaled by ATOM_RADIUS_SCALE_SMALL (0.5), which is set in rsrc_config.py
- Labelling: By default, non-hydrogen atoms are labelled.

NOTES

- **ATYP** restoration: **PROJ** restores any atom drawing styles (**ATYP** codes) that were overridden by **PERS** or **SPIX**, but not by **TELP** or **PNUT**.
- Flexibility: The projection type itself can be managed globally via the user interface buttons rather than **PROJ**.
- See also: **PERS**, **SPIX**, **TELP**, **PNUT**.

PRUN (Prune connectivity)

Removes unwanted bonds from the connectivity table based on coordination number or bond length.

USAGE

```
PRUN [nb] [atom_list]
PRUN [d1] [d2] [atom_list]
```

ARGUMENTS

[nb] : (Integer) The maximum number of bonds allowed for the specified atoms. The shortest [nb] bonds are retained; all others are deleted. Default is 0 (deletes ALL bonds to the specified atoms).

[d1] [d2] : (Floats) A distance range in ångströms. All bonds involving at least one atom from the list that fall outside the range [d1] to [d2] are deleted.

[atom_list] : The atoms to be pruned. Defaults to the entire **FMOL** list.

NOTES

- Shortest bond rule: When using the [nb] mode, the program prioritizes shorter distances, ensuring the strongest chemical interactions are preserved.
- Permanent: Unlike purely visual settings, **PRUN** modifies the internal connectivity table used for calculations and file output.

PUSH (Direct coordinate shift and/or inversion)

Applies a coordinate translation and/or inversion to the current **FMOL** list.

USAGE

PUSH dx dy dz [sign]

ARGUMENTS

dx, dy, dz : The translation vector in fractional coordinates to be added to the current atom positions.

[sign] : Multiplier for the coordinates (+1 or -1). Defaults to +1.

MECHANISM

- Transformation: The new coordinates are calculated as: x' = (x * sign) + dx y' = (y * sign) + dy z' = (z * sign) + dz

NOTES

- Inversion: To invert the structure through the origin, use **PUSH 0 0 0 -1**.
- Inversion through centre: The command **PUSH 1 1 1 -1** inverts the structure through the fractional coordinate (0.5, 0.5, 0.5).
- Origin shifts: This is the standard method for shifting a molecule by integer cell translations (e.g., **PUSH 1 0 0**).
- Caution: **PUSH** alters the coordinates in your current session.

QUIT

Terminates the session and returns to the operating system.

USAGE

QUIT (or EXIT)

READ (Input atoms and instructions)

Reads atomic coordinates, unit cell parameters, and instructions from the specified file and adds them to the current session. If there is already a structure loaded, the program asks if the current display parameters should be reset or retained.

USAGE

```
READ
READ filetitle
```

ARGUMENTS

filetitle : The name (or base name) of the file to be read.
- If no extension is provided, the program looks for **'.res'**.
- If **'.res'** is not found, *PrimeXP* tries **'.ins'**.

NOTES

- Without an argument, a file search window appears. While *PrimeXP* does not actively parse .cif files, it can extract an embedded .res from within a *SHELXL*-generated cif.
- Working directory: **READ** updates the current working directory to the folder containing the loaded file.
- After a **READ** command, the program issues a (silent) **FMOL** to define the active molecule and generate bonds for display.
- Note the use of **REAP**, which also loads in difference map (Q) peaks.

REAP (Input atoms, instructions, and Q-peaks)

Reads atomic coordinates, unit cell parameters, instructions, and difference Fourier (Q) peaks from the specified file. If there is already a structure loaded, the program asks if the current display parameters should be reset or retained.

USAGE

```
REAP
REAP filetitle
```

ARGUMENTS

filetitle : The name (or base name) of the file to be read.
- If no extension is provided, the program looks for **'.res'**.
- If **'.res'** is not found, it tries **'.ins'**.

NOTES

- Without an argument, a file search window appears. While *PrimeXP* does not actively parse .cif files, it can shred an embedded .res from within a *SHELXL*-generated cif.
- Working directory: **REAP** updates the program's current working directory to the folder containing the loaded file.
- Difference map peaks: This is the primary distinction from **READ**. **REAP** imports the Q-peaks (found after the **'END'** instruction) so they can be identified and renamed using **NAME** or **PICK**.
- Like **READ**, **REAP** issues a silent **FMOL** upon completion to generate the initial connectivity and display.

RIGB (Hirshfeld rigid-bond analysis)

Calculates the absolute difference in mean-square displacement amplitude (MSDA) for pairs of atoms projected along their interatomic vector. For bonded atoms, this is used for the Hirshfeld rigid-bond test.

USAGE

```
RIGB
RIGB/H
RIGB/H ALL
RIGB/H [atom_list]
RIGB OFF
```

QUALIFIER

/H : Triggers the generation of an interactive 2D NxN heatmap overlay in the graphics window instead of the standard terminal table.

ARGUMENTS

ALL : Includes hydrogen atoms in the output (table or heatmap).

[atom_list] : Generates the heatmap for a specific selection of atoms (e.g., **'C1 C2 C3'** etc.).

OFF : Dismisses the active graphical heatmap.

OUTPUT

[Terminal table] Generates a formatted table for the evaluated bonds containing:

- A colour block indicating bond rigidity on a continuous spectrum from 0.000 Å² (violet) to the configured ceiling (red, default 0.010 Å²), configurable in rsrc_config.py.
- Atom 1, Atom 2, interatomic distance (Å), and the calculated Δz^2 (Å²).

[Graphical Heatmap] If the **/H** qualifier is used, an interactive NxN matrix is overlaid on the 3D viewport:

- Bonded pairs are drawn as squares.
- Non-bonded pairs are drawn as circles.
- For massive structures, individual cell borders are suppressed to maintain performance.
- Click and drag a marquee box with the left mouse button to zoom into specific regions of the matrix. Right-click to step back or reset the zoom.
- On clicking any heatmap cell, a text legend displays the atom labels (including **PART**, **RESI**, and **SYMM** superscripts if enabled) and the Δz^2 value.

- A dashboard dropdown menu appears, allowing the user to filter the heatmap visibility using four states: **`'Heatmap off'`**, **`'Dim distant'`** (default), **`'Dim close'`**, and **`'Full heatmap'`**.

NOTES

- In the terminal, atoms refined isotropically are marked with an asterisk (*). While the rigid-bond postulate does not strictly apply to isotropic models, large differences in U_{iso} between bonded atoms are usually suspect and are included in the analysis to alert the user.
- Calculations are performed dynamically on-the-fly to ensure the data reflects the current model state.
- You can visualize the **`RIGB`** thresholds directly on the 3D model by cycling the bond mode button in the graphics window to **`'ΔU bonds'`**.
- By default, the heatmap applies a dynamic, distance-based α fade (Dim distant), bringing bonds and close contacts to the foreground. This can be inverted (Dim close) to emphasize through-space non-bonded correlations, or bypassed completely (Full heatmap) via the dashboard.
- Extensive visual customization for the heatmap (ceiling limits, background scrim wash, distance fade limits, line thickness, etc.) can be defined in the configuration file and applied via the **`IMPT`** command.

RING (Ring highlighting and centroid finder)

Identifies chordless chemical rings [between 3 and 8 (default)] atoms in size based on the current connectivity graph of the active molecule, but the maximum may be changed in "rsrc_config.py". It can either assign a specific bond type to the ring bonds or generate dummy atoms at their centroids.

USAGE

```
RING [bond_type] [atom_list]
RING/X [atom_list]
```

ARGUMENTS

[bond_type] : Optional. The bond type to assign to the bonds within the identified rings. Defaults to 2. This argument is ignored if the **/X** qualifier is used.

[atom_list] : Optional. Acts as a filter. If specified, the command will only search for rings that contain at least one of the listed atoms. If omitted, all detected rings in the molecule are processed.

QUALIFIER

/X : Calculates the centroid of each matching ring and creates a dummy atom (named X1, X2, etc.) at that position. These atoms are added to the **FMOL** list. Does not modify existing bond types.

NOTES

- Optimization: To ensure high performance on large structures, **RING** uses an ego-graph search to locate rings locally around target atoms.
- Atom exclusion: Q-peaks and existing dummy (X) atoms are excluded from the ring detection graph.
- Dummy atoms generated with **/X** are useful for defining the centroids of pi-bonded rings (e.g., Cp or benzene rings) in organometallic complexes.
- Use the **JOIN** command if you need to explicitly link a newly created dummy centroid to a metal centre.

ROTA (Rotates the view)

Rotates the current orientation matrix. This can be done by via sequential rotations around the Cartesian screen axes, aligning a specific vector horizontally, or spinning the model around a defined atomic bond axle.

USAGE

```
ROTA n1,phi1; [n2,phi2; ...]
ROTA atom1 atom2
ROTA angle atom1 atom2
```

ARGUMENTS

[n] : The Cartesian screen axis around which the rotation is applied. Accepts values: 1 = Vertical (Y) axis 2 = Horizontal (X) axis 3 = Depth (Z) axis pointing out of the screen

[phi] : Rotation extent (°).

[angle] : The angle to spin the model around a specific bond/vector.

[atom1] : The first atom defining the alignment vector or rotational axle.

[atom2] : The second atom defining the alignment vector or rotational axle.

NOTES

- Coordinate rotation (**ROTA** n,phi): A positive **'phi'** value rotates the structure anticlockwise when viewed looking towards the origin from the positive end of the specified axis.
- Vector alignment (**ROTA** atom1 atom2): Rotates the plot about the Z axis to make the vector between the two specified atoms horizontal (i.e., parallel to the X axis). The first named atom ([atom1]) will be positioned on the left-hand side.
- Bond/axle rotation (**ROTA** angle atom1 atom2): Uses the vector between the two atoms as an axle and spins the model around it by the specified angle. Useful for viewing Newman projections after using **MATR** or **LINE** to look down the bond.

SAVE (Save graphical view and state)

Saves the complete current session state, atom parameters, connectivity, and full graphical view configuration into a compressed JSON (.sav) file.

USAGE

```
SAVE
SAVE [filename]
SAVE/L [filename]
```

ARGUMENTS

/L	:	Forces output to an uncompressed, long-form JSON file.
[filename]	:	If omitted, the command defaults to saving "view_state.sav".

BEHAVIOUR

- Atom properties: Writes out the full detailed property dictionary for every atom in the active list (**FMOL**), including coordinates, radii, colours, and U-values.
- Connectivity: Saves the active connectivity table, preserving bond styles set by the **JOIN** command.
- View state: Captures the exact visual appearance of the graphics window at the moment of saving, including:
    - The current 3D orientation matrix (**MATR/ROTA**/mouse spins).
    - Background colours or gradients (**BACK**).
    - Rendering modes (**PROJ/TELP/PNUT/VOID/EDEN**, flat shading, blending).
    - Manually dragged label positions and label visibility.
    - Thermal ellipsoid probability settings (**ADPS**).

NOTES

- A .sav extension is appended to the filename if missing.
- By default, **SAVE** creates a minified archive.
- Use the **/L** (Long-form) flag if you need the file to be formatted as standard **'pretty-printed'** json, making it human-readable and more easily editable.
- **NEXT** restores graphics scenes created with **SAVE**.

SGEN (Symmetry generation)

Generates new atoms for all permutations of the specified atoms and symmetry operation codes (symcodes). The program assigns unique names to the new atoms and copies properties such as atom radii, colours, and types. Appropriate transformations are rigorously applied to anisotropic temperature factors.

USAGE

```
SGEN symcodes [atom_list]
SGEN -symcode [atom_list]
SGEN/N symcodes [atom_list]
```

ARGUMENTS

[symcodes] : One or more symmetry operation codes defining the transformations to apply. (See **'SYMM'**, and **'SRCH'** for more information on format and usage). If a negative symcode is provided, the named atoms are transformed in place rather than duplicated. When using a negative symcode, only ONE symcode is permitted.

[atom_list] : Atoms to be processed. Defaults to ALL if omitted.

QUALIFIER

/N : Suppresses terminal output of the newly generated coordinates and parameters.

OUTPUT

- Coordinate list: Unless **/N** is used, the **SGEN** prints the label, symcode, and fractional coordinates of each generated atom.
- Anchors: The listing gives the names of already existing atoms to which the new atom is bonded, facilitating immediate identification of the generated group's orientation.

NOTES

- Duplicate prevention: Atoms are NOT generated if the symmetry operation produces an atomic position that is already occupied. This check is performed against the entire structure.
- Connectivity: **SGEN** updates the connectivity table (bonds) to include the newly generated atoms or reflect in-place transformations.
- Auto-naming: The program attempts to assign unique names by varying the last character of the parent atom's name.
- Renaming: To manually change the names of the newly generated atoms after using **SGEN**, use the **NAME** or **PICK** commands.

- Visual mapping: When the graphics window is set to the **'Colour: Symm'** mode (see **COLR**), **SGEN** dynamically updates the colours of the generated or transformed atoms to match their new symmetry equivalents.

SORT (Sort and reorder the atom list)

Re-orders the current atom list. By default, hydrogen atoms are grouped with the heavy atom to which they are attached.

USAGE

```
SORT
SORT atom1
SORT atom1 atom2 [atom3 ... atomn]
SORT/H [atom_list]
SORT/N [atom_list]
SORT/R [atom_list]
SORT/P [atom_list]
```

ARGUMENTS

[None]	:	Defaults to a numerical sort.
[atom1]	:	The atom is moved to the top of the atom list.
[atom1, atom2, ...]	:	Atoms are sorted in the supplied order.

QUALIFIERS

/H : Treats hydrogen atoms as standard atoms. They will NOT automatically move with the heavy atoms to which they are bonded.

/N : Sorts the specified atoms numerically, ignoring any alphabetic or non-numerical characters in their labels. Because the sort is stable, atoms with the same number will retain their original relative order from the file (e.g., F1, O1, N1, C1, F2, O2, N2, C2).

/R : Sorts the specified atoms in numerical residue order.

/P : Sorts atoms in numerical residue order, and within each residue, sorts them into standard PDB order (this assumes relatively standard PDB atom naming conventions are used).

NOTES

- Subset insertion: When using the **/N**, **/R**, or **/P** qualifiers on a subset of atoms, the newly sorted block is placed into the main list starting at the original position of the atom that comes first in the newly sorted group. If that "winning" atom was originally located further down the file, the entire block will be shifted to that lower position.
- Default hydrogen behaviour: Without the **/H** qualifier, hydrogens are placed immediately after the atom to which they are bonded, based on the current connectivity table. If a hydrogen is bonded to multiple atoms (e.g., if an H-bond

has been explicitly added), the shortest distance determines its primary attachment for sorting purposes.
- Iterative sorting: The **SORT/N** command is often applied iteratively to selected subsets of the full atom list, sometimes in combination with standard **SORT** commands, to achieve the desired final sequence.

SPIX (Space-filling projection)

Switch to a space-filling (CPK) representation of the current **FMOL** list. This mode is used for visualizing molecular packing, van der Waals surfaces, and void spaces (but see **VOID**).

USAGE

```
SPIX [atom_list]
SPIX [CELL | /CELL] [atom_list]
```

ARGUMENTS

[atom_list] : Defaults to all non-hydrogen atoms in the current **FMOL** list.
CELL : Displays the unit-cell box. (Also accepts **/CELL** flag).

RADII AND REPRESENTATION

- Radius selection: **SPIX** uses the third **ARAD** parameter (SRAD) for each atom. These are set to van der Waals radii in the configuration files.
- Drawing type: **SPIX** temporarily overrides the current **ATYP** settings to render atoms as solid spheres. Original ATYPs are restored when returning to **PROJ** mode.

NOTES

- Hydrogen atoms: By default, H atoms are omitted from **SPIX** for clarity unless they are explicitly included in the [atom_list].

SRCH (Symmetry search)

Searches for atoms that are within a specified distance of the current **FMOL** list. A standard method for finding intermolecular contacts, hydrogen bonds, or completing the coordination sphere of a metal centre.

USAGE

```
SRCH [dist] [atom_list]
SRCH/S [dist] [atom_list]
```

ARGUMENTS

[dist] : The search radius in ångströms. Defaults to 3.5 Å.

[atom_list] : The target atoms to search around. Defaults to all atoms in the current **FMOL** list.

QUALIFIER

/S : "Shortest only" mode. For each external atom found, only the single closest symmetry-equivalent position is reported.

OUTPUT

- **SRCH** prints a table of all atoms found within the search radius.
- Data includes: The atom label, the symmetry code (symcode), and the calculated distance to the target atom.
- Symcode format: The standard *SHELX* format (e.g., 1555) where **'1'** is the symmetry operator index and **'555'** represents the unit cell translation.

NOTES

- Structure completeness: Unlike **FMOL**, which only looks for covalently bonded networks, **SRCH** looks through the entire structure.
- Integration: Symcodes found with **SRCH** are used as input for **SGEN** to generate the symmetry-related atoms to the model.
- To clear generated symmetry atoms, use **KILL** or **FMOL**.

SYMM (List symmetry operations)

Lists the symmetry operations for the current crystal structure, used by **GROW**, **SGEN**, **MGEN**, and **SRCH** to describe or generate symmetry equivalents.

USAGE

SYMM

ARGUMENTS

None : Lists the symmetry operators of the structure.

NOTES

- Origin: Symmetry operations are parsed directly from the .ins/.res file loaded at the start of the session. They cannot be defined interactively.
- Terminal output: Lists the symmetry operations and symcodes with a colour block next to each operation. This colour identifies the atoms generated by that symmetry operation when the graphics window is set to "Colour: Symm" mode (via the **COLR** command, or the UI button).
- The identity operation (X, Y, Z) is always assumed by the program.

TELP (Thermal ellipsoid plots)

Generates a high-quality graphical representation of the molecule where atoms are represented by their displacement ellipsoids, in the style pioneered by *ORTEP* (Johnson, 1965). By default, **TELP** sets the background to white. Unlike *ORTEP* and the original *XP* (Sheldrick, 2008), **TELP** in *PrimeXP* is fully 3D interactive. When you have a view that you like and well positioned labels, use the **DRAW** command to save it as a PNG file.

USAGE

TELP [prob]

ARGUMENTS

[prob] : Optional. A number between 1.0 and 99.0 sets the global probability level for drawing displacement ellipsoids.

NOTES

- Anisotropy: Atoms with anisotropic displacement parameters are drawn as ellipsoids.
- Isotropic fallback: Atoms defined isotropically are rendered as spheres scaled to the specified probability level based on their U(iso) values.
- Orientation: The plot uses the current viewing orientation as defined by the last **ROTA**, **LINE**, **MPLN**, **MATR**, etc. command.
- Labels: Non-hydrogen atom labels are included by default, but their appearance can be toggled by screen button or menu bar action.

REFERENCES

Johnson, C. K. (1965). *ORTEP*: A Fortran thermal-ellipsoid plot program for crystal structure illustrations. Oak Ridge National Laboratory Report ORNL-3794.

Sheldrick, G. M. (2008). A short history of *SHELX*. *Acta Cryst.* A64, 112-122.

TORS (Torsion angle calculation)

Calculates the torsion angle for a specified sequence of four atoms. If more than four atoms are given, it gives the torsion angles for each successive group of four specified.

USAGE

```
TORS
TORS ALL
TORS atom1 atom2 atom3 atom4 [atom5 ...]
```

ARGUMENTS

[no arguments] : Defaults to all non-hydrogen atoms (**TORS** ALL LESS $H).

ALL : Calculates all torsion angles, including hydrogen atoms.

atom1...atom4+ : The atoms defining the torsion angle(s). The angle is calculated for the rotation of the 1-2 bond relative to the 3-4 bond around the 2-3 axis.

OUTPUT

- The torsion angle in degrees (°).

NOTES

- Sign convention: The program follows the IUPAC convention. Looking down the 2-3 bond, the angle is positive if the 1-2 bond must be rotated clockwise to eclipse the 3-4 bond.
- Connectivity: The atoms do not strictly need to be bonded in the connectivity table for this calculation to be performed, though meaningful chemical torsions usually are.
- For multiple torsions along a backbone, consider using the **NEWM** command for a tabular summary.

UNDO (Remove specific bonds)

Deletes all bonds from the connectivity table that involve two atoms included in the supplied list. To delete e.g., all bonds to one specific atom or type of atom, use **PRUN**.

USAGE

```
UNDO [atoms]
```

ARGUMENTS

[atoms] : A standard atom selection (e.g., C1 C2, $N, or a residue range). The command targets bonds where bonded atoms are part of the selection.

OUTPUT

- Prints the number of bonds removed from the connectivity table.
- Updates the graphical display to reflect the removed bonds.

NOTES

- Logic: A bond is only removed if both partners are in the selection.
- Comparison with **PRUN**: To delete all bonds connected to a single atom (regardless of what it is bonded to), use the **PRUN** command instead.
- Connectivity: This command only affects the "bonds" list; it does not remove the atoms themselves from the **FMOL** list.

UNIQ (Unique molecule assembler)

Assembles complete, unique molecule(s) starting from a specified seed atom (or atoms). **UNIQ** searches for and generates necessary symmetry equivalents to build molecules outward, while preventing spatial overlaps and duplicates.

USAGE

UNIQ atom_list

ARGUMENTS

atom_list : The seed atom(s) for assemblage of unique fragments.

BEHAVIOUR

- Assembly: The program uses the seed atoms as a starting frontier and evaluates Cartesian distances against all available symmetry operations to find valid, chemically bonded neighbours.
- Overlap Prevention: Strict spatial overlap filters and KD-Tree tracking ensure that duplicate symmetry equivalents are rejected and only a single contiguous molecule is built.
- Connectivity: After assembling the atoms, the program recalculates the connectivity table. All disconnected or overlapping atoms are removed from the active **FMOL** list.
- View Update: The graphics window is refreshed to show only the newly assembled unique fragment.

NOTES

- Unlike **KILL**, which deletes atoms from the master structure, **UNIQ** manages the active **FMOL** display list.

VOID (Solvent accessible voids)

Generates and visualises solvent accessible spaces. **VOID** uses a rolling probe algorithm to map continuous regions of **'empty'** (i.e., unmodelled) space in the structure.
To ensure mathematical and crystallographic rigour, **VOID** silently generates a virtual unit cell packed with all necessary symmetry equivalents before running the calculation. This prevents false voids from being generated.

USAGE

```
VOID [probe_radius] [resolution] [extend_a] [extend_b]
[extend_c]
VOID flip
VOID off
```

ARGUMENTS

[probe_radius] : A float specifying the radius of the rolling probe in ångströms. Smaller probes will find smaller pores and channels. If omitted, defaults to 1.2 Å.
[resolution] : A float specifying the grid spacing of the **VOID** map in ångströms. If omitted, defaults to 0.25 Å.
[extend_a/b/c] : Fractional extensions along the a, b, and c axes beyond the primary cell. If omitted, they default to 0.0.
[flip] : Inverts the **'void'** cloud so that it represents the molecular envelope rather than the void space. Acts as a toggle, switching back and forth.
[off] : Clears the void visualisation from the screen and memory.

TERMINAL OUTPUT

The command generates a brief status report detailing:
- Parameters used for the generation (probe radius and grid resolution).
- Any extensions applied to the plotting boundaries.
- The approximate void volume calculated in cubic ångströms (Å³).
- The percentage of void space in the unit cell.
- A notification if no accessible void space is detected.
- Performance notes regarding **MGEN** and **FUSE**.

GRAPHICS

A successful **VOID** command injects the calculated void space directly into the active graphics view.

- By default, voids are depicted as **`'clouds'`** (semi opaque).
- An UI button appears that cycles the representation through CLOUDS, CAGES, COMBO, and OFF modes (can be slow for large structures).
- By default, the visualisation is cropped at the boundaries of the unit cell (plus any specified extensions).

NOTES

- For optimal performance, it is best to run **`VOID`** before invoking **`MGEN`**. If **`MGEN`** has already been run, it might help to run **`FUSE`** prior to **`VOID`**.
- The extraction of float arguments is sequential; if multiple floats are provided, they define the probe radius, resolution, and a/b/c extensions in that exact order.
- System defaults and behaviour can be modified in rsrc_config.py.

WAIT (Assign weights for MPLA etc.)

Assigns numerical weights to selected atoms. These weights are used to influence subsequent geometric and analytical calculations, such as mean planes or rigid-body libration analysis.

USAGE

```
WAIT [wt] [atoms]
WAIT/Z [atoms]
WAIT/M [atoms]
```

ARGUMENTS

[wt] : The numerical weight to assign. Defaults to 1.0 if not specified. This argument is ignored if the **/Z** or **/M** qualifiers are used.

[atoms] : The selection of atoms to receive the weight. Defaults to ALL active atoms if omitted.

QUALIFIERS

/Z : Assigns weights based on the atomic number (Z).

/M : Assigns weights based on atomic masses (M).

OUTPUT

- Writes a confirmation message indicating the weight assigned and the number of atoms updated.

NOTES

- Default state: All atoms are initialized with unit weights (1.0) when first loaded (**READ/REAP**), named (**NAME**), or picked (**PICK**).
- Use in calculations: **MPLN**: Uses weights for centroid and SVD plane fitting. **LIBR**: Uses weights for Schomaker & Trueblood TLS rigid-body analysis.
- Chemical properties: atomic numbers and masses are resolved using the unified chemical properties registry (rsrc_properties.py), which also supports dummy atoms and isotopes deuterium (D) and tritium (T).

WIPE (Clear terminal)

Clears all text from the terminal display buffer.

USAGE

WIPE

OUTPUT

- Wipes all command history and output text from the terminal window.

NOTES

- Functionality: **WIPE** only clears the terminal display. It does not affect the internal state of the session, the atom list (**FMOL**), or the graphics window.
- Use case: Useful for decluttering the terminal after running verbose commands like **ENVI**, **SRCH**, or **LIBR**.

ZPAD (Zefirov-Palyulin-Dashevskaya puckering parameters)

Calculates Zefirov-Palyulin-Dashevskaya puckering parameters for 5- and 6-membered rings using internal endocyclic torsion angles. These have some advantages over Cremer-Pople parameters.

SYNTAX

ZPAD [atoms]

If no atoms are specified, the command evaluates all valid 5- and 6-membered rings across the entire structure. If specific atoms are provided, only rings containing those targets are evaluated.

OUTPUT

For 6-membered rings, the output includes the total puckering amplitude (Q) and the spherical polar angles (Θ, Φ_2). For 5-membered rings, it outputs the amplitude (q_2) and phase angle (Φ_2). For 5-membered rings, the principal out-of-plane "flap" atom is also given.

ZPAD classifies ring shapes based on IUPAC rules:

(C) Chair : Θ near 0 or 180° (poles of the sphere).
(B) Boat : Θ near 90°, Φ_2 at 0, 60, 120... (equator).
(T) Twist-Boat : Θ near 90°, Φ_2 at 30, 90, 150... (equator).
(E) Envelope : N = 5 rings with Φ_2 near 0, 36, 72...
(H) Half-Chair : intermediate states bridging major forms.

CONVENTIONS

- Torsion formulation: calculated from internal endocyclic torsion angles, which are not affected by mean-plane distortions caused by uneven bond lengths (e.g., by heteroatoms). Torsions use a left-handed convention, to match the Cremer-Pople scheme.
- Pivot atom: *PrimeXP* sets the pivot to the highest-priority heteroatom or lowest alphanumeric label, tracing toward the next lowest alphanumeric label neighbour. A cyclic forward shift of the starting atom transforms Θ into 180 - Θ and shifts Φ_2 by +120° (for N = 6 rings).
- Ring traversal: reversing the atom sequence flips the latitude (Θ -> 180 - Θ) and longitude (Φ_2 -> 180 + Φ_2).

SEE ALSO

CPOP (calculates Cremer-Pople puckering parameters using Cartesian coordinates.)

REFERENCE

Zefirov, N.S., Palyulin, V.A., & Dashevskaya, E.E. (1990). Stereochemical studies. XXXIV. Quantitative description of ring puckering via torsional angles. The case of six-membered rings. *J. Phys. Org. Chem.*, 3(3), 147-158.

Appendix 1. *PrimeXP* source hierarchy

```
src/
├── main.py              # Execution wrapper script
├── PrimeXP.py           # Main application GUI class and
                           primary entry point
├── chemistry/           # Atomic property and coordinate
                           handling
├── commands/            # Core logic for user commands
├── data/                # Internal data models and active
                           selection states
├── graphics/            # PyOpenGL rendering pipeline
│   └── shaders/         # GLSL vertex and fragment shader
                           source files
├── mathematics/         # Matrix transforms, puckering, and
                           topological algebra
├── resources/           # Application assets, GUI icons,
                           colours, configuration
├── system/              # Session management and terminal
                           I/O routing
└── xtallography/        # Space group symmetry and
                           fractional coordinate maths
```

Appendix 2. *PrimeXP* named colours

aliceblue
antiquewhite
aqua
aquamarine
azure
beige
bisque
black
blanchedalmond
blue
blueviolet
brown
burlywood
cadetblue
chartreuse
chocolate
coral
cornflowerblue
cornsilk
crimson
cyan
darkblue
darkcyan
darkgoldenrod
darkgray
darkgreen
darkgrey
darkkhaki
darkmagenta
darkolivegreen
darkorange
darkorchid
darkred
darksalmon
darkseagreen
darkslateblue
darkslategray
darkslategrey
darkturquoise
darkviolet
deeppink
deepskyblue
dimgray
dimgrey
dodgerblue
firebrick
floralwhite
forestgreen
fuchsia
gainsboro
ghostwhite
gold
goldenrod
gray
green
greenyellow
grey
honeydew
hotpink
indianred
indigo
ivory
khaki
lavender
lavenderblush
lawngreen
lemonchiffon
lightblue
lightcoral
lightcyan
lightgoldenrodyellow
lightgray
lightgreen
lightgrey
lightpink
lightsalmon
lightseagreen
lightskyblue
lightslategray
lightslategrey
lightsteelblue
lightyellow
lime
limegreen
linen
magenta
maroon
mediumaquamarine
mediumblue
mediumorchid
mediumpurple
mediumseagreen
mediumslateblue
mediumspringgreen
mediumturquoise
mediumvioletred
midnightblue
mintcream
mistyrose
moccasin
navajowhite
navy
oldlace
olive
olivedrab
orange
orangered
orchid
palegoldenrod
palegreen
paleturquoise
palevioletred
papayawhip
peachpuff
peru
pink
plum
powderblue
purple
rebeccapurple
red
rosybrown
royalblue
saddlebrown
salmon
sandybrown
seagreen
seashell
sienna
silver
skyblue
slateblue
slategray
slategrey
snow
springgreen
steelblue
tan
teal
thistle
tomato
turquoise
violet
wheat
white
whitesmoke
yellow
yellowgreen

References

Anthony, J. E., Brooks, J. S., Eaton, D. L. & Parkin, S. R. (2001).
J. Am. Chem. Soc. **123**, 9482–9483.

Basu Baul, T. S., Brahma, S., Duthie, A., & Parkin, S. (2025).
Inorg. Chim. Acta **585**, 122754

Behrman, E. J. & Parkin, S. (2021).
Acta Cryst. E**77**, 623–628.

Cremer, D. & Pople, J. A. (1975).
J. Am. Chem. Soc. **97**, 1354–1358.

Dolomanov, O. V., Bourhis, L. J., Gildea, R. J., Howard, J. A. K., & Puschmann, H. (2009).
J. Appl. Cryst. **42**, 339–341.

Gao, G., Chen, M., Roberts, J., Feng, M., Xiao, C., Zhang, G., Parkin, S., Risko, C., & Zhang, L. (2020)
J. Am. Chem. Soc. **142**, 2460–2470.

Hirshfeld, F. L. (1976).
Acta Cryst. A**32**, 239–244.

Hossain, T., Atapattu, H. R., Pruett, H., Rahman, M. T., Pedersen, K. R., Huckaba, A. J., Parkin, S. R., & Graham, K. R. (2024)
Chem. Mater. **36**, 11004.

Hübschle, C. B., Sheldrick, G. M., & Dittrich, B. (2011).
J. Appl. Cryst. **44**, 1281–1284.

Hummel, W., Hauser, J., & Bürgi, H. B. (1990).
J. Mol. Graphics **8**(4), 214–220.

Johnson, C. K. (1965).
ONRL Report #3794. Oak Ridge National Laboratory, Oak Ridge, TN.

Kindlmann, G. (2004).
Proc. Joint Eurographics-IEEE TCVG Symposium on Visualization.

Martin, C. D., & Porter, M. A. (2011).
arXiv. https://doi.org/10.48550/arxiv.1103.2338

Mohan Kumar, T. M., Sowbhagya, C., Bhavya, P., Yathirajan, H. S., & Parkin, S. (2026).
Acta Cryst. E**82**, 326–330.

Newman, M. S. (1952).
Record Chem. Progr. **13**, 111–116.

Pajevic, S., & Pierpaoli, C. (1999).
Mag. Res. Med. **42**, 526–540.

Parkin, S. R. & Behrman, E. J. (2009).
Acta Cryst. C**65**, o529–o533.

Parkin, S. R. & Behrman, E. J. (2011).
Acta Cryst. C**67**, o324–o328.

Parkin, S. R., Coldren, W. H., Fernandez, J. P. Hadad, C. M, & Behrman, E. J. (2018).
Cryst. Growth Des. **18**, 6995-7005.

Parkin, S., Glidewell, C. & Horton, P. N. (2023).
Acta Cryst. C**79**, 77–82.

Parkin, S. & Hope, H. (2003).
Acta. Cryst. D**59**, 2228–2236.

Parkin, S., Rupp, B., & Hope, H. (1996).
Acta Cryst. D**52**, 18–29.

Parkin, S., Rupp, B., & Hope, H. (1996).
Acta Cryst. D**52**, 1161–1168.

Pokharel, U. R., Parkin, S., & Selegue, J. S. (2026).
Crystals **16**, 201.

Pokharel, U. R., Parkin, S., & Selegue, J. S. (2026).
Crystals **16**, 409.

Roecker, L. & Parkin, S. (2024).
Acta Cryst. E**80**, 713–716.

Schomaker, V. and Trueblood, K. N. (1968).
Acta Cryst. B**24** 63–76.

Sheldrick, G. M. (2008).
Acta Cryst. A**64**, 112–122.

Sheldrick, G. M. (2015).
Acta Cryst. C**71**, 3–8.

Sowbhagya, C., Mohan Kumar, T. M., Bhavya, P., Shankara Prasad, H. J., Chinthal, C. H., Yathirajan, H. S., Parkin, S.
Acta Cryst. (2026). E**82**, 530–533.

Spek, A. L. (2009).
Acta Cryst. D**65**, 148–155.

Vinaya, Basavaraju, Y. B., Srinivasa, G. R., Shreenivas, M. T., Yathirajan, H. S. & Parkin, S. (2023).
Acta Cryst. E**79**, 54–59.

Zefirov, N. S., Palyulin, V. A., & Dashevskaya, E.E. (1990).
J. Phys. Org. Chem. **3**(3), 147–158.