

OpenEye
SCIENTIFIC

BROOD

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Introduction **1**
- 1.1 Applications . . . 1
- 1.2 Utility Programs . . . 1
- 1.3 Installing the BROOD Database . . . 2

2 vBROOD **3**
- 2.1 Overview . . . 3
- 2.2 Build and Run . . . 4
  - 2.2.1 Build a New Query . . . 4
  - 2.2.2 Setting up BROOD . . . 4
  - 2.2.3 Run BROOD . . . 5
- 2.3 Build a New Query . . . 5
  - 2.3.1 Loading a Molecule . . . 6
  - 2.3.2 Selecting a Fragment for Replacement . . . 6
  - 2.3.3 Fixing Issues with the Query Structure . . . 7
  - 2.3.4 Editing fragments . . . 10
  - 2.3.5 Finishing up . . . 12
- 2.4 Run BROOD . . . 12
  - 2.4.1 Load a Query . . . 13
  - 2.4.2 Select a Filter . . . 14
  - 2.4.3 Set Up BROOD . . . 15
  - 2.4.4 Run BROOD . . . 16
- 2.5 View Results . . . 16
- 2.6 Viewing Results in VIDA . . . 17
  - 2.6.1 Getting Started . . . 18
  - 2.6.2 Viewing Clusters . . . 18

3 BROOD **23**
- 3.1 Overview . . . 23
- 3.2 Input Files . . . 23
  - 3.2.1 Query Files . . . 23
  - 3.2.2 Fragment Database . . . 23
  - 3.2.3 Example Fragment Files . . . 24
  - 3.2.4 Molecular File Formats . . . 25
- 3.3 Example Commands . . . 25
- 3.4 Command Line Help . . . 26
- 3.5 Parameters . . . 27
  - 3.5.1 File Options . . . 27
- 3.6 Output Files . . . 32
  - 3.6.1 Info File . . . 33

3.6.2 Param File . . . 33
3.6.3 Log File . . . 36
3.6.4 Report Files . . . 36
3.6.5 Hitlist File . . . 36
3.6.6 Removed File . . . 37

4 CHOMP **39**
4.1 Overview . . . 39
4.2 Example Commands . . . 39
4.3 Command Line Help . . . 41
4.4 Required Parameters . . . 41
4.5 Optional Parameters . . . 41
4.5.1 Execute Options . . . 41

5 BroodDBMerge **45**
5.1 Overview . . . 45
5.2 Command Line Help . . . 45
5.3 Example Commands . . . 46
5.4 Required Parameters . . . 46
5.5 Optional Parameters . . . 46

6 Overview **49**

7 Human Complement C3a Receptor Example **51**
7.1 Specifying the Database . . . 51
7.2 Loading a Molecule . . . 51
7.3 Editing the Query . . . 53
7.4 Selecting Properties . . . 54
7.4.1 Using the Property Filter (Optional) . . . 54
7.5 Setting Up the BROOD Search . . . 54
7.6 Opening and Viewing Results . . . 55
7.7 Exploring the Results . . . 56
7.7.1 Keyboard Shortcuts for Navigating BROOD Results . . . 56
7.7.2 BROOD Results in the Results Viewer . . . 57
7.7.3 Working with Favorites in the BROOD Results Viewer . . . 59

8 Fragment Joining Tutorial **61**
8.1 BROOD Database . . . 61
8.2 Loading Two Molecules . . . 61
8.3 Loading the Protein . . . 62
8.4 Choosing a Query . . . 62
8.5 Setting up a Search . . . 64
8.6 Examining the Log Files . . . 65
8.7 Viewing Results in a Protein Site . . . 65

9 Generating a Database with Crystallographic Conformations **67**
9.1 Preparing Data Sources . . . 67
9.2 Generating a Database . . . 67
9.3 Advanced Topic: Editing the Fragments at an Intermediate Step . . . 69

10 Theory **71**
10.1 Theory . . . 71
10.1.1 3D Fragment similarity searching . . . 71
10.1.2 Clash detection and selectivity . . . 71
10.1.3 Strain energy . . . 72

10.1.4 Fragment joining . . . 72
10.1.5 Cyclization . . . 72
10.1.6 Molecular complexity . . . 72
10.1.7 Aromatic fragment comparison . . . 73
10.1.8 Shape-based belief model . . . 73
10.1.9 Abbott bioavailability score . . . 73
10.1.10 Similarity to available compounds . . . 74
10.1.11 BROOD Database Generation (CHOMP) . . . 74
10.1.12 BROOD Database . . . 75
10.2 Fragment joining . . . 75
10.3 Cyclization . . . 75
10.4 BROOD Database Generation (CHOMP) . . . 75

11 BROOD Fragment Database **77**
11.1 ChEMBL31 Fragment Database . . . 77
11.2 ChEMBL31_lite Fragment Database . . . 77
11.3 brood-database-chembl-3.0.0 . . . 77

12 Release History **79**
12.1 BROOD 5.0.0 . . . 79
12.1.1 Major Bug Fixes . . . 79
12.1.2 Minor Bug Fixes . . . 79
12.2 Bioisostere TK 5.0.0 . . . 79
12.2.1 New Features . . . 79
12.2.2 Minor Bug Fixes . . . 80
12.3 BROOD 4.2.1 . . . 80
12.3.1 Related Toolkit Information . . . 80
12.4 BROOD 4.2.0 . . . 81
12.4.1 New Features . . . 81
12.4.2 Related Toolkit Information . . . 83
12.5 BROOD 4.1.0 . . . 83
12.6 Bioisostere TK 4.1.0 . . . 83
12.6.1 New features . . . 83
12.6.2 Major bug fixes . . . 84
12.6.3 Minor bug fixes . . . 84
12.7 BROOD 4.0.0 . . . 84
12.7.1 Major bug fixes . . . 84
12.8 Bioisostere TK 4.0.0 . . . 84
12.8.1 New features . . . 84
12.9 BROOD 3.2.2 . . . 85
12.9.1 Minor bug fixes . . . 85
12.10 BROOD 3.2.1 . . . 85
12.10.1 Minor bug fixes . . . 85
12.11 BROOD 3.2.0 . . . 85
12.11.1 New Features . . . 86
12.11.2 Major bug fixes . . . 86
12.12 BROOD 3.1.7 . . . 86
12.13 BROOD 3.1.6 . . . 86
12.14 BROOD 3.1.5 . . . 86
12.15 BROOD 3.1.4.0 . . . 86
12.15.1 Documentation changes . . . 86
12.16 BROOD 3.1.3 . . . 86
12.16.1 Minor bug fixes . . . 86
12.17 BROOD 3.1.2 . . . 87

12.17.1 Major bug fixes . . . 87
12.17.2 Minor bug fixes . . . 87
12.18 BROOD 3.1.1 . . . 87
12.18.1 Major bug fixes . . . 87
12.19 BROOD 3.1.0 . . . 87
12.19.1 Minor bug fixes . . . 87
12.20 BROOD 3.0.0 . . . 87
12.20.1 Platform Support . . . 88
12.20.2 vBROOD . . . 88
12.20.3 Results Viewer . . . 88
12.20.4 BROOD . . . 88
12.20.5 CHOMP . . . 89
12.20.6 Merge . . . 89
12.21 BROOD 2.0.0 . . . 89
12.21.1 New Features: . . . 89
12.22 BROOD 1.1.2 . . . 89
12.23 BROOD 1.1.1 . . . 90
12.24 BROOD 1.1.0 . . . 90
12.25 BROOD 1.0.1 . . . 92
12.26 BROOD 1.0.0 . . . 92
12.27 BROOD 0.9.b2 . . . 92
12.28 BROOD 0.9.b1 . . . 93

13 Citation **95**

14 Copyright and Trademarks **97**

15 Sample Code **99**

16 Citation **101**
16.1 Orion® . . . 101
16.2 Orion Floes . . . 101
16.3 Toolkits and Applications . . . 101
16.4 Publications for Bibliographies . . . 103
16.4.1 Orion . . . 103
16.4.2 AFITT and FLYNN Applications . . . 103
16.4.3 OEDocking Application and Toolkit . . . 103
16.4.4 OMEGA Application and Toolkit . . . 103
16.4.5 ROCS Application . . . 104
16.5 OpenEye MMDS Web Service . . . 104

17 Technology Licensing **105**
17.1 GCC . . . 177
17.1.1 GCC RUNTIME LIBRARY EXCEPTION . . . 178
17.1.2 GNU GENERAL PUBLIC LICENSE . . . 179

Index **193**

CHAPTER

ONE

INTRODUCTION

BROOD is a lead generation tool designed to generate new and diverse compounds that satisfy isosteric and chemical feature constraints while also incorporating synthetic feasibility. Starting with a hit or lead molecule, **BROOD** generates bioisosteric analogs by replacing user-specified portions of the lead with fragments that have similar shape and electrostatics but with potentially novel connectivity and chemistry. Fragments and scaffold couplings are derived within the scope of a known chemical space, either user-specified or broadly relevant to medicinal chemistry (default **BROOD** database).

BROOD can also be used within the context of structure-based design if experimental or modeled cocrystal structures are available. Resulting **BROOD** analogs can be readily filtered and prioritized based on bioisosteric similarity, 2D chemical diversity, and 1D molecular properties. **BROOD** has multiple applications, including lead hopping, side-chain enumeration, library design, and expansion of composition of matter patents.

The **BROOD** distribution comprises two programs as well as a script that can be run in conjunction with OpenEye's molecular visualizer, **VIDA**.

1.1 Applications

The **BROOD** distribution contains three applications:

BROOD

The **BROOD** application creates analog molecules from a lead molecule by changing fragments of the lead, where the new analogs have potentially different properties.

vBROOD

vBROOD is the graphical interface of the **BROOD** application.

CHOMP

The **CHOMP** application creates a **BROOD** database by fragmenting molecules, filtering the fragments, generating 3D conformations, and organizing and indexing the fragments for rapid searching.

1.2 Utility Programs

The following utility program is also included in this distribution:

- *BroodDBMerge*: This utility creates a single database by merging two **BROOD** databases.

1.3 Installing the BROOD Database

Included in the **BROOD** product is a large database of fragments generated by fragmenting the ChEMBL molecular database [Bento-2014]. This database is available for download from db-downloads in a platform-independent format. For user convenience, the database is available in either ZIP or tar.gz format, though either file can be used on any platform.

The **BROOD** database is large (>5G) and takes 15–30 minutes to download with a good internet connection. When the download is complete, move the file to a convenient location and unpack (either unzip or untar/gzip) the database. The default **BROOD** database is a directory full of prepared files that are optimized to make common **BROOD** searches execute quickly.

After unpacking the default database, set it as the default database.

1. Launch **vBROOD** either by double-clicking the icon or by using the command line interface, depending on your platform.
2. To specify a database, select ->`Edit Preferences...` under the *Edit* menu. A preferences pop-up window will open.
3. Under the first line of the dialog, either type the path to your newly unpacked database or use the icon to open a file dialog that will allow you to browse to the database. In either case, specify the directory created in the unpacking process as the database rather than selecting one of the files contained in the database.

CHAPTER

TWO

VBROOD

2.1 Overview

vBROOD is the graphical interface that ships with **BROOD**. It gives access to all the command line features as well as some GUI-only features, such as query editing, and placing constraint and color atoms.

Upon completion, **BROOD** clusters the resultant molecules. **vBROOD** presents the cluster heads for viewing; the full clusters can be explored in **VIDA** (see *Viewing Results in VIDA*).

Most **BROOD** options can be changed from within **vBROOD**. **vBROOD** can easily run **BROOD** in multiprocessor mode; however, it cannot run **BROOD** across a cluster.

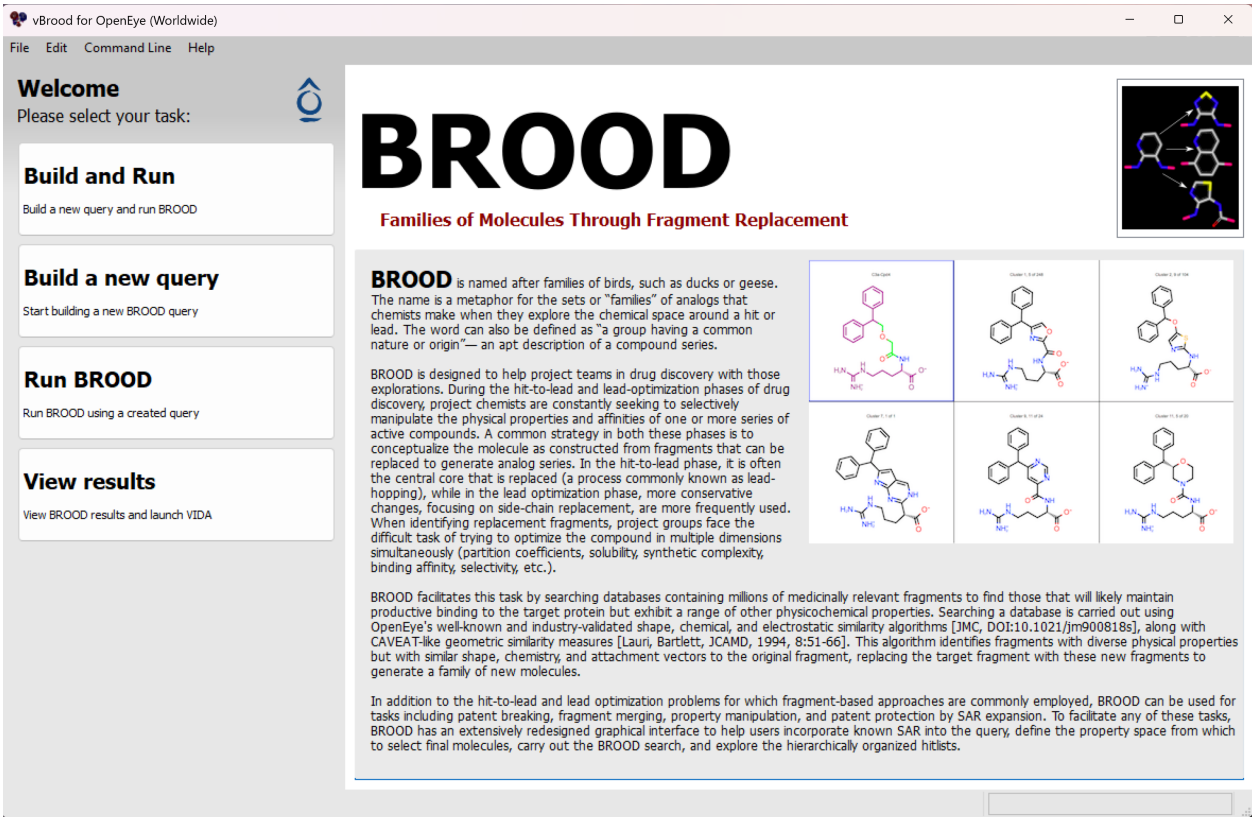


Fig. 1: **Figure 1.** The **vBROOD** task screen.

vBROOD offers five tasks to help run **BROOD**:

- *Build and Run*: Creates a query and immediately starts a **BROOD** run.
- *Build a New Query*: Creates and edits a query to save and send to a **BROOD** run.
- *Run BROOD*: Runs **BROOD** with a query.
- *View Results*: Browses results and launches the **BROOD** interface in **VIDA**.

2.2 Build and Run

Build and Run is the most direct way to run **BROOD**. This feature walks you through the entire process of constructing a query, setting up a **BROOD** run, and browsing results.

The *Build and Run* wizard consolidates the *Build a New Query* and *Run BROOD* steps. However, the *Query Selection* page in the *Run BROOD* wizard is bypassed, and queries will be saved only in the *Run Directory*.

2.2.1 Build a New Query

See the *Build a New Query* section.

2.2.2 Setting up BROOD

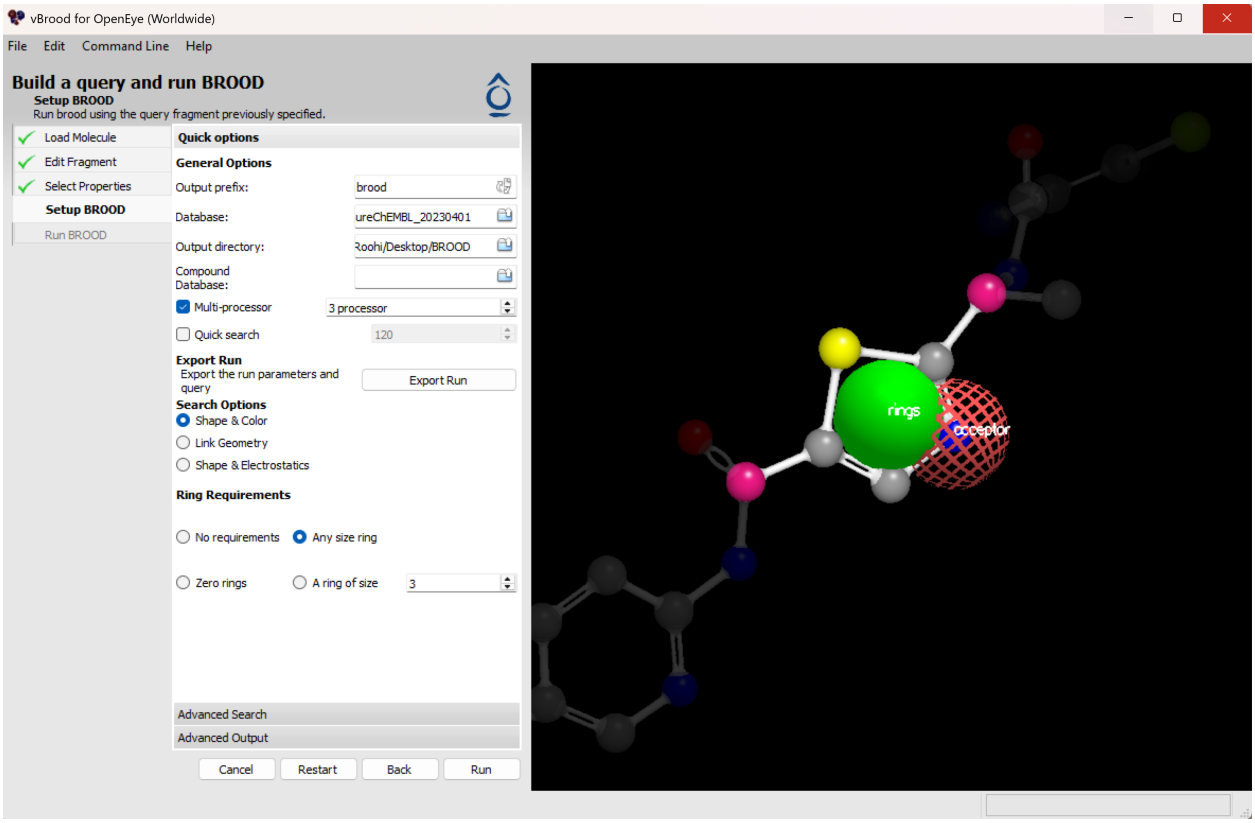


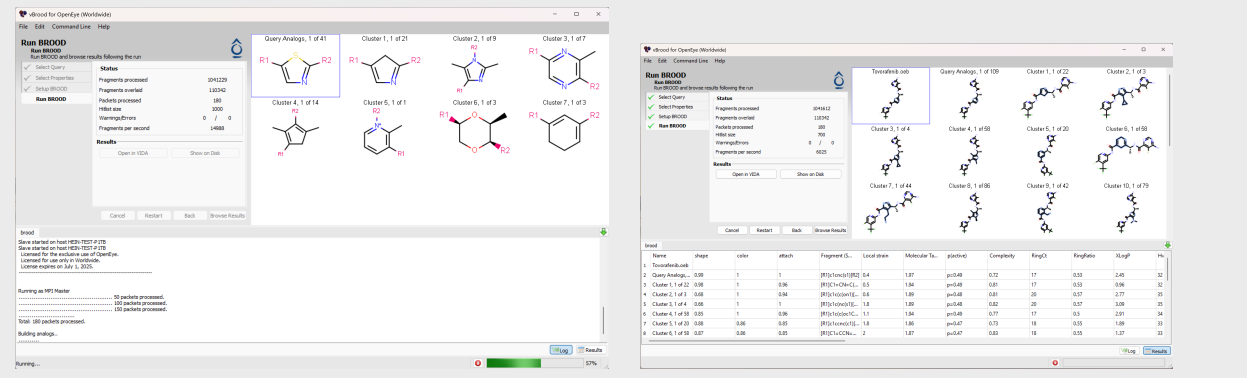
Fig. 2: **Figure 2.** Options for running **BROOD**.

The next step in running **BROOD** is choosing the run parameters. Most options correspond directly to a command line option. Hovering over most elements in the interface provides a brief description of the option as well as the command line flag it represents.

The multiprocessor options, found in the *Quick Options* section, are particularly useful. These options allow you to run multiple **BROOD** instances using OpenMPI on the same machine as **vBROOD**. This allows additional processors to be used if they are available.

When all options are set, click "Run" to start the **BROOD** run and advance to the *Run BROOD* page.

2.2.3 Run BROOD



A run in progress | A completed run

Figure 3. A comparison of in-progress and completed **BROOD** runs.

The screen will then switch to display a 2D hit list, the *Output* window, and the *Status* window. This display will dynamically update as the run progresses.

- The *Hit list* window shows the current cluster heads from the database.
- The *Status* window on the left shows information about how many fragments have been processed, how many were actually scored against the query, and how many were eliminated.
- The *vBROOD Output* window at the bottom displays the output of the **BROOD** run as if it were run from the command line. You can toggle between this output log and a spreadsheet of the top results by using the "Log" and "Results" buttons on the bottom right.
- At the bottom right of the **vBROOD** window is a progress bar and a "Stop" button for aborting the run.

As the run progresses, the top results are displayed in the 2D hit list and the spreadsheet continues to update. Both the spreadsheet and the 2D hit list provide scrollbars for browsing the hit list. Clicking on a 2D depiction highlights the same row in the spreadsheet and vice versa.

After a run completes, the buttons "Open in VIDA" and "Show on Disk" will be displayed and enabled. Clicking "Open in VIDA" opens **VIDA** with the **BROOD** *Results Viewer* tool loaded (see *Viewing Results in VIDA*). Clicking "Show on Disk" opens a window displaying the directory containing the results of the run. If the run did not complete successfully, **BROOD** displays a dialog box, and the *Output* window at the bottom shows the *Log* view. Scanning the log may provide an indication of the problem.

Click "Browse Results" to bring up the list of all runs for the current session. See *View Results* for more information.

2.3 Build a New Query

The process for building a query is as follows:

1. Load a molecule.
2. Select a fragment.
3. Modify the fragment to be replaced.

The workflow will stop after editing the fragment and allow you to save the query.

2.3.1 Loading a Molecule

The first step in creating a **BROOD** query is to load or create a source molecule. The molecule can be loaded from a file, entered by name, or sketched in the sketcher.

- To load a molecule from a file, either click the folder in the *Molecule* text entry box or choose a recently loaded molecule from the drop-down menu, which can be accessed from the arrow in the text entry box.
- To enter a molecule by name, type a structure name or a SMILES string in the text entry box. The 2D display will show a molecule if the entry is valid.
- To sketch a new molecule, click the + in the text entry box. This opens the sketcher for input. Sketch a molecule and click "OK" when you are satisfied with the sketch.

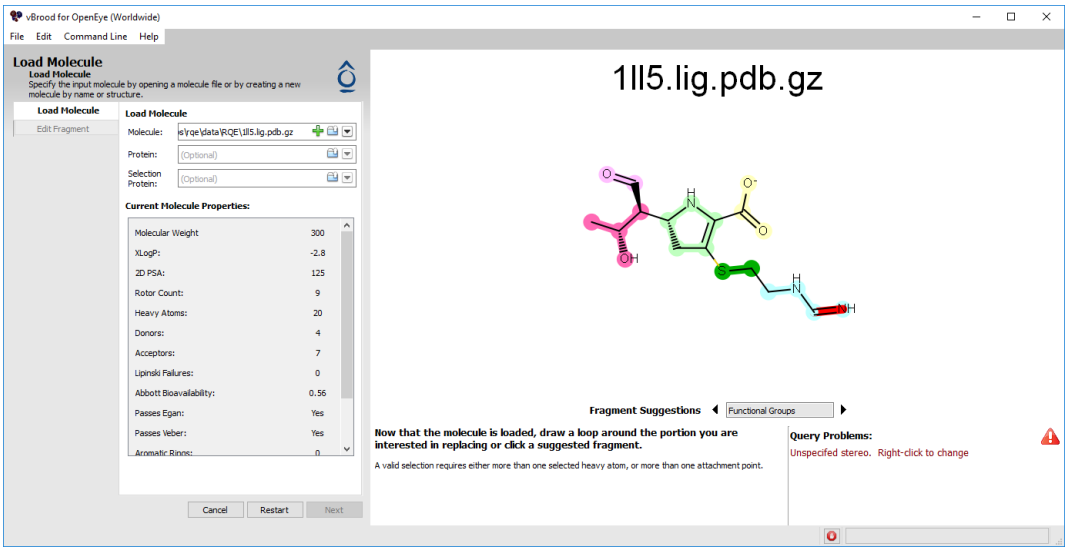


Fig. 3: **Figure 1.** Loading a molecule into **vBROOD**.

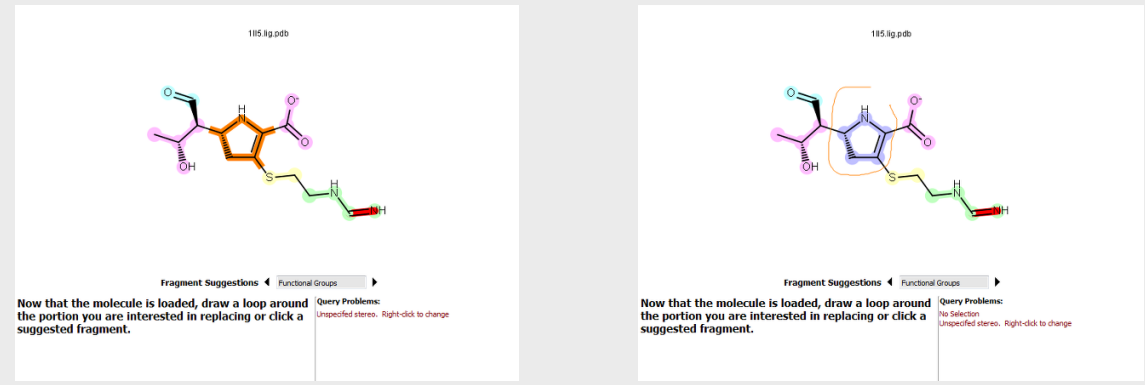
A protein can be loaded to provide a bump-check for generated results, although protein chemistry is not used as part of the score. You can also specify a selection protein, which requires a protein atom to be close to the query as a simplistic screen for selectivity.

Below the protein text entry boxes, **vBROOD** displays the properties of the loaded molecule. Any properties that might need attention are highlighted in red.

2.3.2 Selecting a Fragment for Replacement

After loading or creating a molecule, **vBROOD** will display the 2D structure. You must then select some portion of the molecule for replacement. You can select the fragment to be replaced in two ways:

- Click one or more of the highlighted fragments.
- Select a fragment by lassoing it.



Fragment-based selection | Lasso selection

Figure 2. How to select a fragment. For the fragment-based option, clicking on the pyrrolidine ring fragment highlighted it in orange.

2.3.3 Fixing Issues with the Query Structure

Some molecules may need additional fixes for problems found when they are loaded or selected. **vBROOD** highlights any problematic atom or bond and displays a message in the lower-right corner. If the message is red, the problem must be addressed before you can continue to build the query. Messages in yellow can be ignored, but should be addressed as they may lead to unexpected behaviors.

The following examples show some common issues and how to fix them.

Unspecified Stereochemistry

BROOD requires the specification of certain types of stereochemistry. Ignoring this problem can result in issues with coordinate generation or with property filtering. To fix unspecified stereochemistry, right-click on the red highlighted atom or bond and select *Set Chirality* or *Set Bond Stereo* and choose the appropriate stereochemistry.

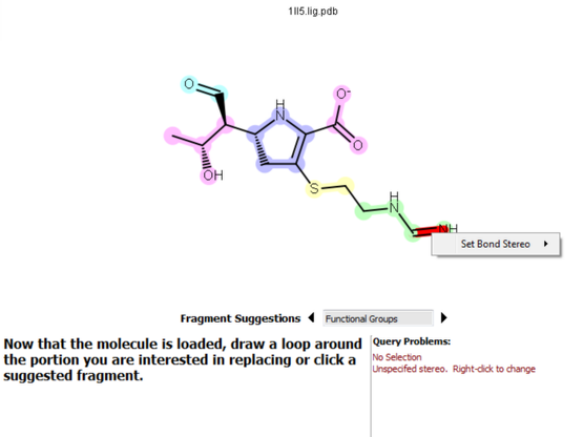


Fig. 4: **Figure 3.** Fixing bond stereo.

Broken Double Bonds

BROOD does not allow double bonds to be broken by selection. To correct this problem, either click *Select Bond* or redo the selection.

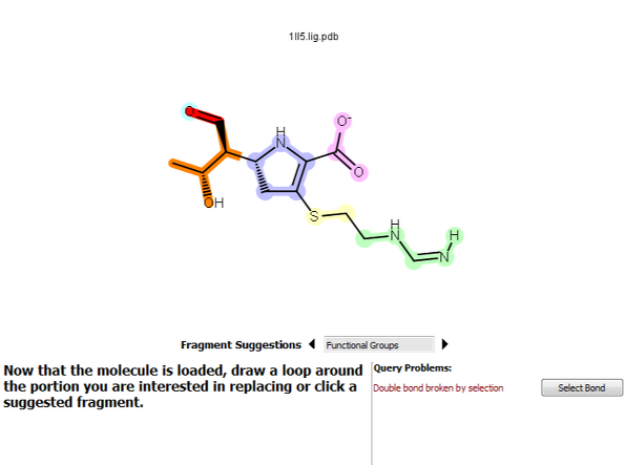


Fig. 5: **Figure 4.** A double bond broken by the selection.

Partially Selected Rings

Broken aliphatic rings can be ignored (but will always be closed), but broken aromatic rings must be fixed. To do this, either click the *Select Ring* button or redo the selection.

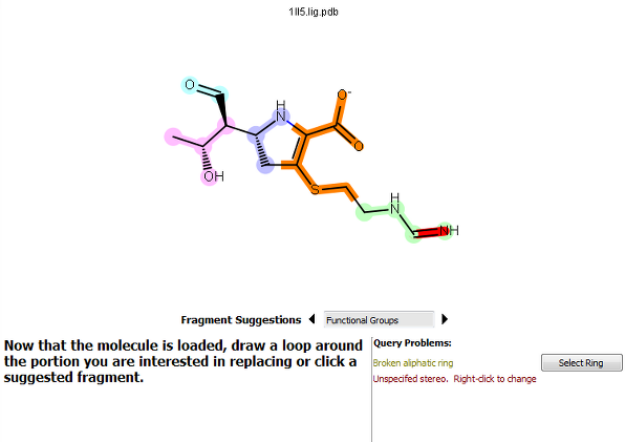


Fig. 6: **Figure 5.** A ring broken by the selection.

More than Three Attachment Points

BROOD databases have fragments containing between one and three attachment points. Thus with **BROOD** will not produce any results when you use a query with four or more attachments. Redo the selection so that it has between one and three attachment points.

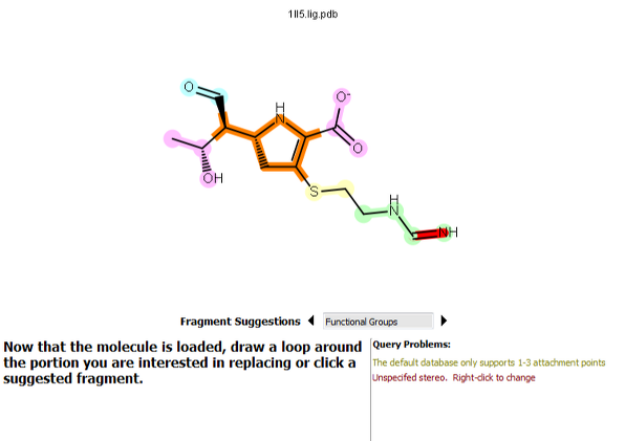


Fig. 7: **Figure 6.** A selection resulting in four attachment points.

Overlapping Attachment Points

Selecting two atoms that are each bonded to a third, unselected atom results in two attachment points overlapping. To correct this, click *Fix* or redo the selection.

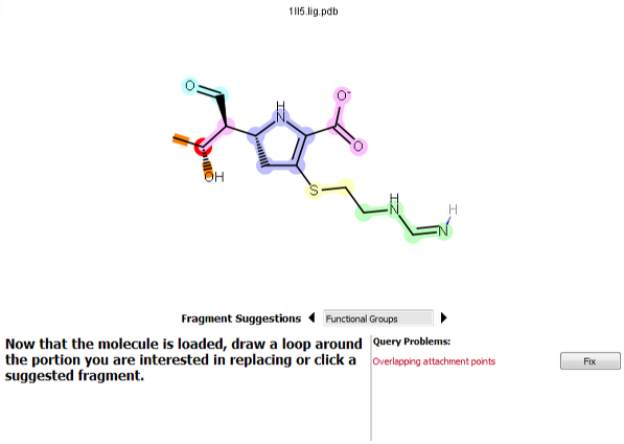


Fig. 8: **Figure 7.** A selection resulting in overlapping attachment points.

No Selection

vBROOD requires you to select some fraction of the molecule or it will not continue. To fix this, select a fragment of the molecule.

Once a molecule has a valid selection and all problems have been resolved, the “Next” button will be enabled. Click “Next” to advance to the *Edit Fragment* step.

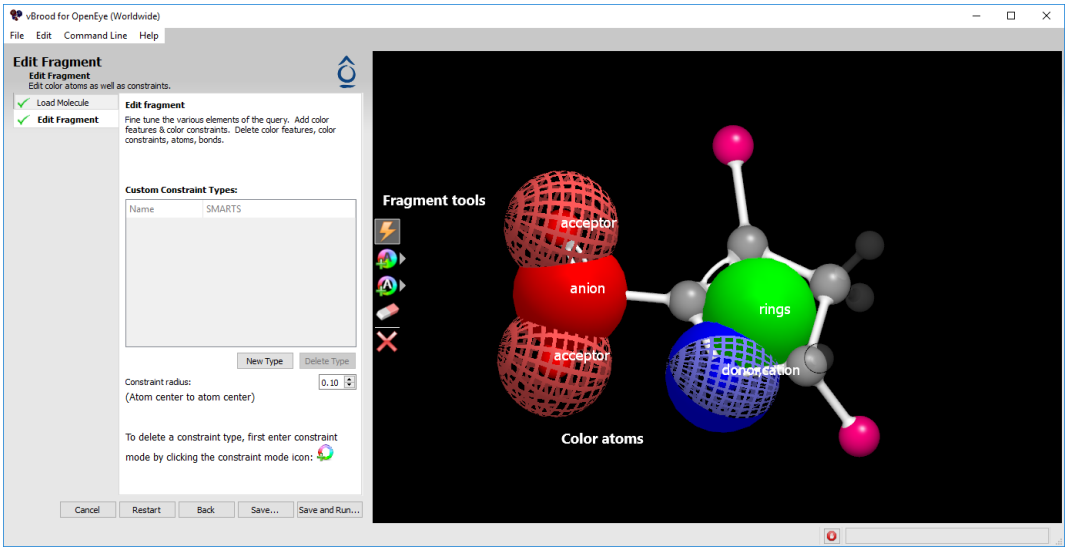


Fig. 9: **Figure 8.** Edit fragment screen.

2.3.4 Editing fragments

The *Edit Fragment* step allows users to edit some aspects of the replacement query. You can add and remove color atoms, add and remove constraints, or remove atoms in the query. This step also displays the query structure in 3D and the attachment vectors as R-group atoms. Color atoms are displayed as large spheres patterned and colored according to the color atom type. Constraint atoms are displayed as colored and patterned rings with a diameter corresponding to the diameter of the constraint.

The Fragment Editor provides five tools for editing a query, shown as icons on the left side of the 3D window.

- *Select*. This option can be used to select a group of atoms for deletion.
- *Add Color Atom*. A drop-down menu is provided to select the color atom type.
- *Add Constraint Atom*. A drop-down menu is provided to select the constraint type.
- *Eraser*. This option can be used to erase an atom, color atom, or constraint.
- *Delete Selected*. This option deletes all selected atoms.

If the query includes a protein, it is also possible to toggle the visibility of the protein with the *Show Protein* and *Hide Protein* buttons.

Adding a color atom

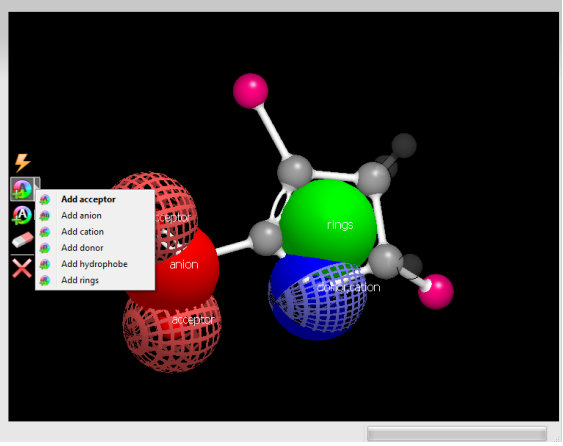


Fig. 10: **Figure 9.** The menu to add color atoms.

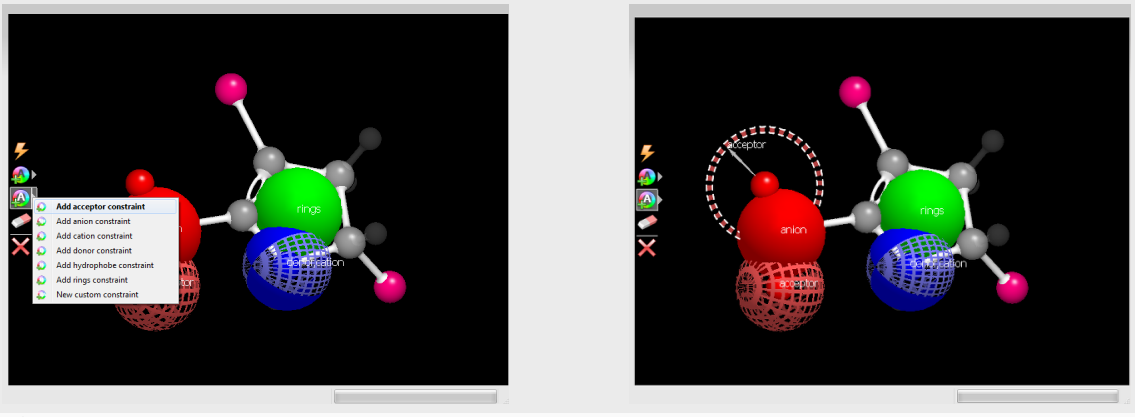
Color atoms allow users to specify desired chemistry at specific places on the fragment. Adding a color atom to a molecule is simple.

1. Select a color atom type from the *Color Atom* menu in the Fragment Editing toolbar.
2. Click an atom.

Either color atoms or molecular atoms can be selected. Multiple color atoms placed in the same location are displayed as slices of the same color atom sphere. By placing the same type of color atom at the same location, color atoms may be weighted to increase their influence on the score.

To place a color atom in between one or more atoms, hold the Shift key while clicking additional atoms. The color atom will be placed at the center of mass of the selected group.

Adding a constraint



The constraint menu | A constraint

Figure 10. Constraint options and example.

A constraint is much like a color atom in that it represents desired chemical features. Unlike a color atom, however, a constraint is a requirement, not a component of the score. Adding a constraint to a molecule is largely the same as adding a color atom.

1. Select a constraint type from the constraint menu in the Fragment Editing toolbar.
2. Click an atom. Multiple constraints in the same location will be displayed as segments of the same ring.

As with color atoms, to place a constraint in between one or more atoms, hold the Shift key while clicking additional atoms. The constraint will be placed at the center of mass of the selected group.

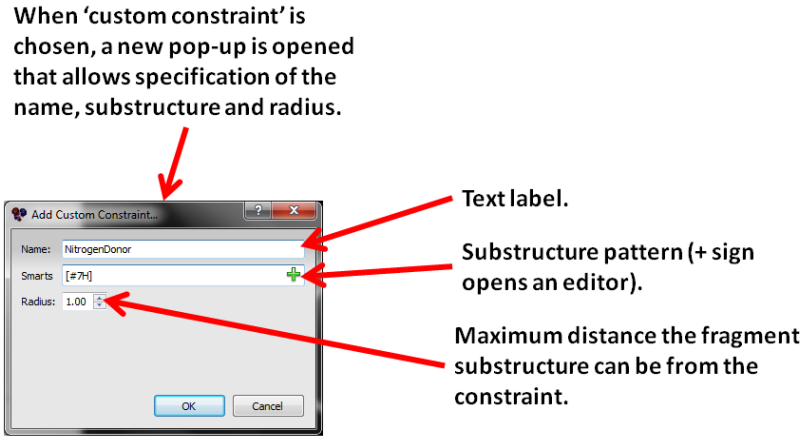


Fig. 11: **Figure 11.** The custom constraint dialog box.

Custom constraints can also be defined with user-supplied SMARTS patterns. To define a new constraint type, select *Add Custom Constraint*. The *Add Custom Constraint* dialog prompts for a constraint name, a constraint SMARTS pattern, and a radius. SMARTS patterns can either be typed into the SMARTS text editor or selected from the SMARTS dialog by clicking the + button.

Once these steps have been completed, click "Next" to progress to the next page.

2.3.5 Finishing up

After editing the query, either click "Save" or "Save and Run." "Save" will ask for the file name and return you to the task selection screen. "Save and Run" will ask for a file name, save, and then continue to the *Run BROOD* task.

2.4 Run BROOD

vBROOD runs the **BROOD** command line application using a query. It can run **BROOD** on multiple processors, provide access to the command line options, display the output of the command line in a log window, and display the best fragments found as the run progresses.

2.4.1 Load a Query

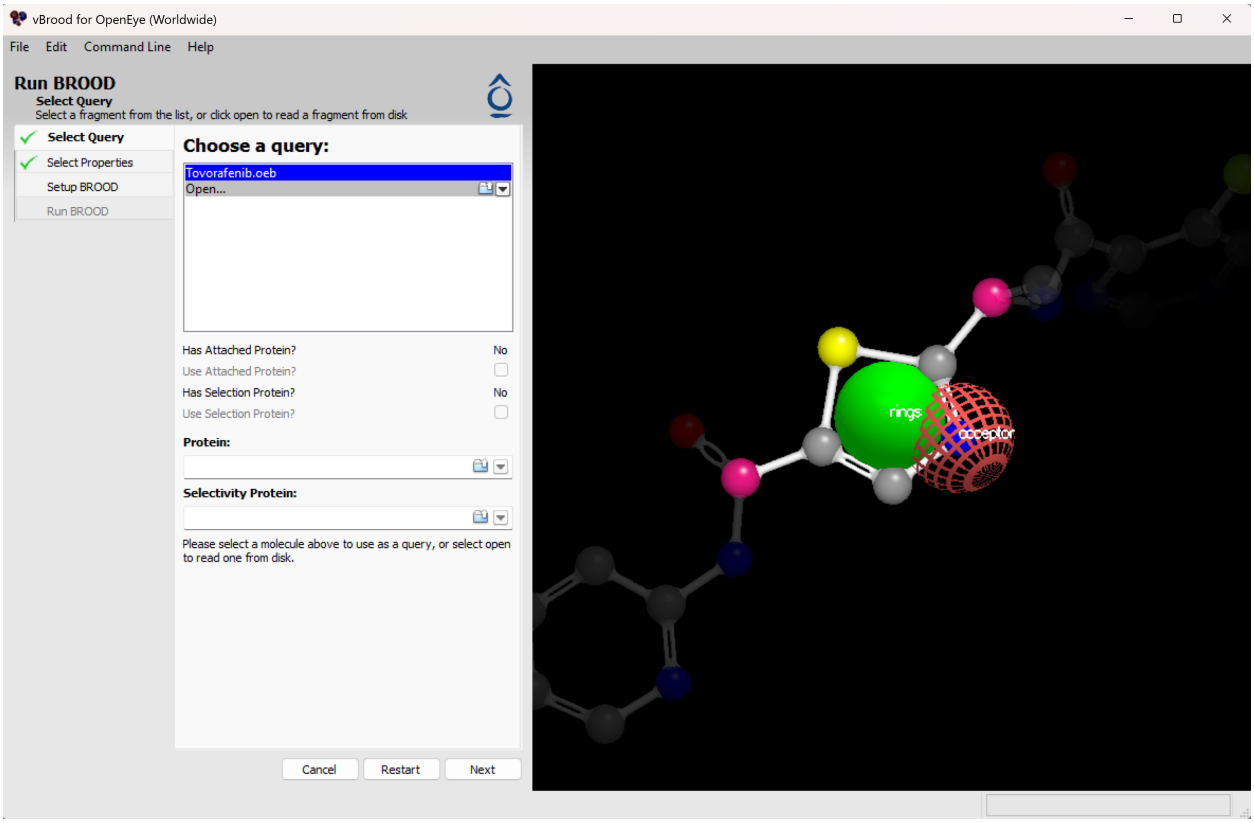


Fig. 12: **Figure 1.** Loading a query.

The first step to running **BROOD** is to load or select a query.

- Previously built queries appear in the query list.
- Queries can be loaded by clicking the *Open* option.
- If the query does not have an associated protein, clicking on the *Open* icon in the *Protein* text box will allow you to add one.

After a query is selected, it is displayed in the 3D window. Once it is loaded, clicking "Next" will take you to *Select a Filter*.

2.4.2 Select a Filter

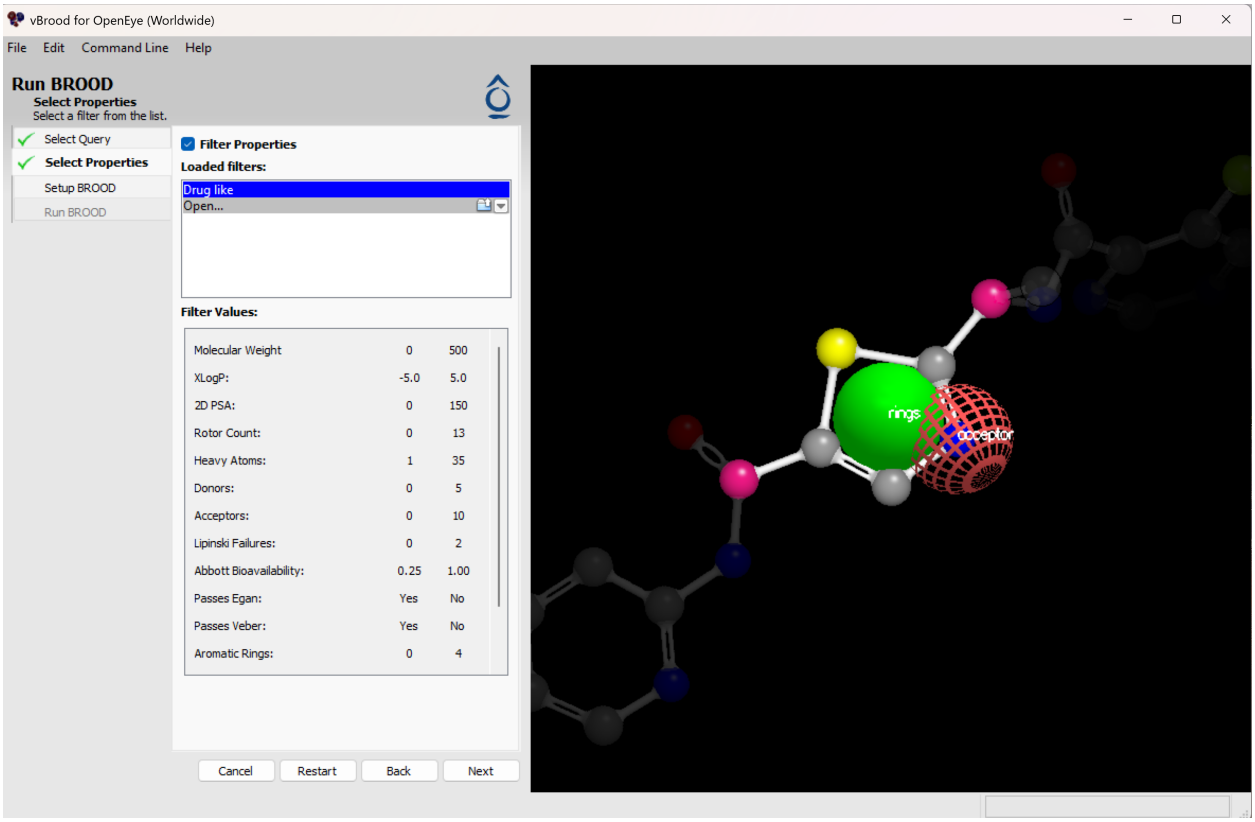


Fig. 13: **Figure 2.** Choosing a filter.

A filter can limit the **BROOD** search and can shorten the run time. **vBROOD** provides a built-in drug-like filter; you can also build additional filters. To use a filter, select one from the list or open a previously saved filter.

To open a previously saved filter:

- Click the *Open* item in the list. This brings up a dialog that can be used to select a filter.
- Select a filter file and click *Open*. This adds and selects a new entry in the list.

Alternatively, you can select a filter from the *Recents* menu. Selecting a filter will display its given properties in the *Filter Values* box below the filter list. When you are satisfied, click "Next" to continue to *Set up BROOD*.

2.4.3 Set Up BROOD

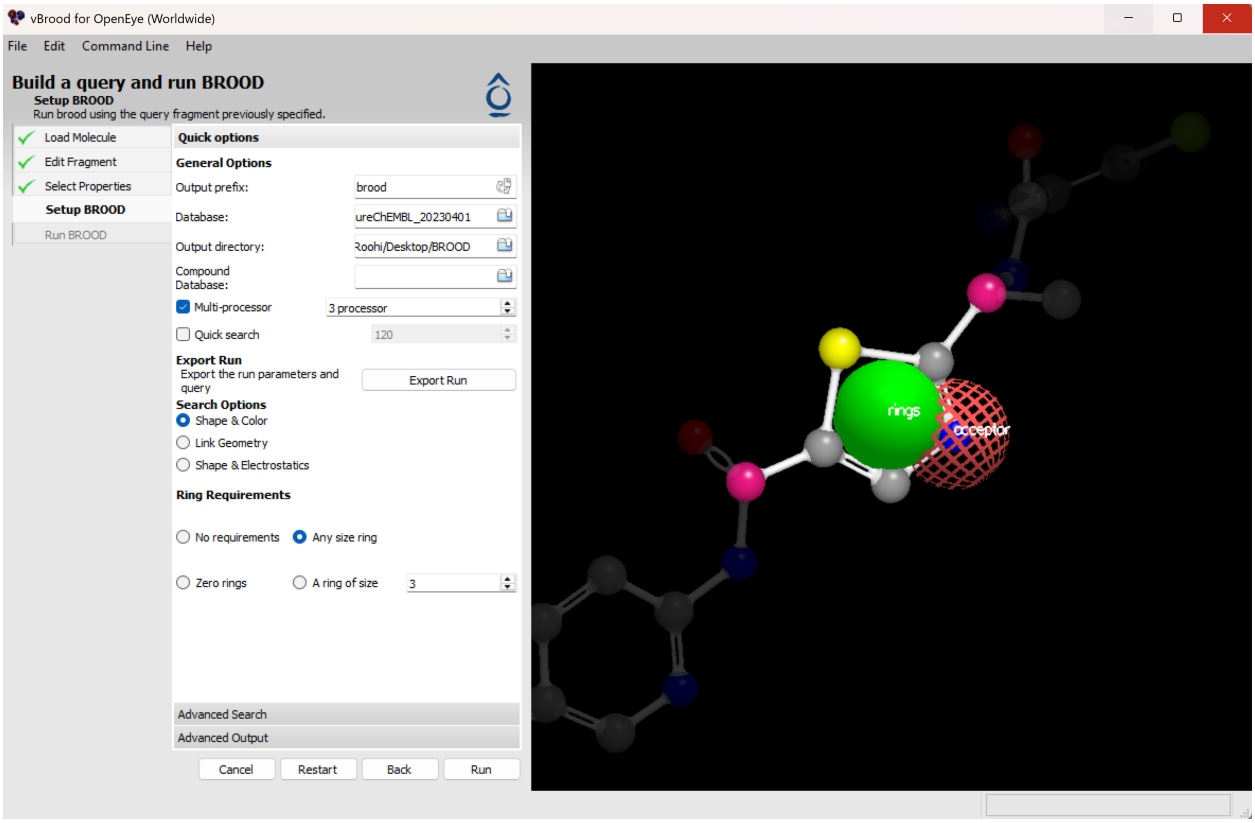


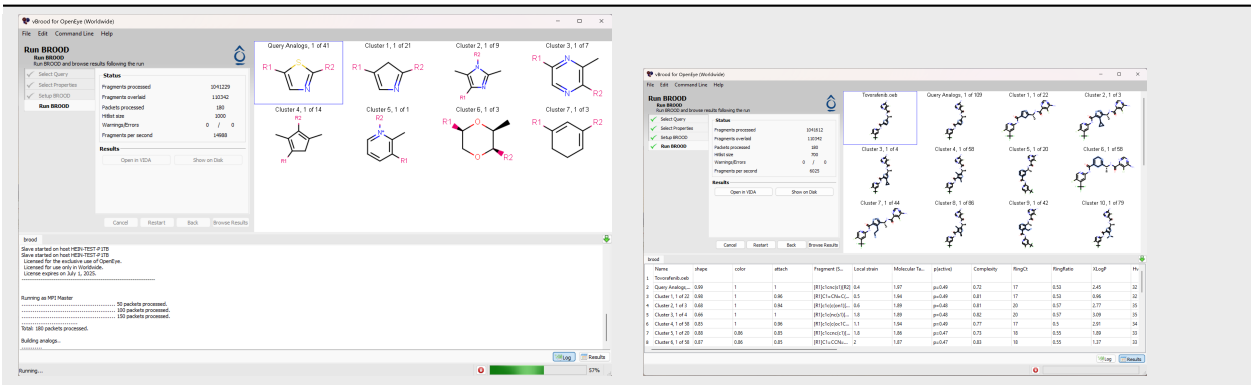
Fig. 14: **Figure 3.** Options for running **BROOD**.

The next step in running **BROOD** is choosing the run parameters. Most options correspond directly to a command line option. Hovering over most elements in the interface provides a brief description of the option as well as the command line flag it represents.

The multiprocessor options, found in the *Quick Options* section, are particularly useful. These options allow you to run multiple **BROOD** instances using OpenMPI on the same machine as **vBROOD**. This allows additional processors to be used if they are available.

When all options are set, click "Run" to start the **BROOD** run and advance to the *Run BROOD* page.

2.4.4 Run BROOD



A run in progress

A completed run

Figure 4. A comparison of in-progress and completed **BROOD** runs.

The screen will then switch to display a 2D hit list, the *Output* window, and the *Status* window. This display will dynamically update as the run progresses.

- The *Hit list* shows the current cluster heads from the database.
- The *Status* window on the left shows information about how many fragments have been processed, how many were actually scored against the query, and how many were eliminated.
- The *Output* window at the bottom displays the output of the BROOD run as if it were run from the command line. You can toggle between this output log and a spreadsheet of the top results by using by using the "Log" and "Results" buttons on the bottom right.
- At the bottom right of the **vBROOD** window is a progress bar and a "Stop" icon for aborting the run.

As the run progresses, the top results are displayed in the 2D hit list and the results spreadsheet continues to update. Both the spreadsheet and the 2D hit list provide scrollbars for browsing the hit list. Clicking on a 2D depiction highlights the same row in the spreadsheet and vice versa.

After a run completes, the buttons "Open in VIDA" and "Show on Disk" will be displayed and enabled. Clicking on "Open in VIDA" opens **VIDA** with the **BROOD** *Results Viewer* tool loaded (see *Viewing Results in VIDA*). Clicking "Show on Disk" opens a window displaying the directory containing the results of the run. If the run did not complete successfully, BROOD displays a dialog box, and the *Output* window at the bottom shows the *Log* view. Scanning the log may provide an indication of the problem.

Click "Browse Results" to bring up the list of all runs for the current session. See *View Results* for more information.

2.5 View Results

The final wizard launches the *Results Viewer*. This displays a list of all **BROOD** runs since launching **vBROOD**. Selecting a single run displays the cluster heads from the hit list file. It also allows you to locate the file on disk (using "Show on Disk") and to launch the **vBROOD** *Results Viewer*. Finally, previously generated results saved on disk may also be loaded.



	Name	shape	color	attach	Fragment (S...	Local strain	Molecular Ta...	p(active)	Complexity	RingCt	RingRatio	XLogP	Hv
1	Tovorafenib.oeb												
2	Query Analogs,...	0.99	1	1	[R1]c1cnc(s1)[R2]	0.4	1.97	p=0.49	0.72	17	0.53	2.45	32
3	Cluster 1, 1 of 22	0.98	1	0.96	[R1]C1=CN=C(...	0.5	1.94	p=0.49	0.81	17	0.53	0.96	32
4	Cluster 2, 1 of 3	0.68	1	0.94	[R1]c1c(c(on1)[...	0.6	1.89	p=0.48	0.81	20	0.57	2.77	35
5	Cluster 3, 1 of 4	0.66	1	1	[R1]c1c(nc(s1)[...	1.8	1.89	p=0.48	0.82	20	0.57	3.09	35
6	Cluster 4, 1 of 58	0.85	1	0.96	[R1]c1c(c(oc1C...	1.1	1.94	p=0.49	0.77	17	0.5	2.91	34
7	Cluster 5, 1 of 20	0.88	0.86	0.85	[R1]c1ccnc(c1)[...	1.8	1.86	p=0.47	0.73	18	0.55	1.89	33
8	Cluster 6, 1 of 58	0.87	0.86	0.85	[R1]C1=CCN=...	2	1.87	p=0.47	0.83	18	0.55	1.37	33
9	Cluster 7, 1 of 44	0.73	1	0.94	[R1]c1c(c(on1)[...	1.7	1.89	p=0.48	0.78	17	0.49	2.83	35
10	Cluster 8, 1 of 86	0.86	0.99	0.94	[R1]C1=C(C(=...	1.9	1.91	p=0.48	0.95	17	0.5	1.88	34
11	Cluster 9, 1 of 42	0.79	0.86	0.87	[R1]c1ccnc(c1C...	3.2	1.85	p=0.47	0.77	18	0.51	2.57	35
12	Cluster 10, 1 of 79	0.69	1	1	[R1]c1c(nc(s1)[...	0.9	1.89	p=0.48	0.79	17	0.49	2.93	35

Fig. 15: **Figure 1.** Browsing results from a **BROOD** run.

If the run was from the current session of **vBROOD**, the *Log* window will be populated with the output of **BROOD**. This includes detailed information regarding how the results were generated. Otherwise, it will be blank.

The **vBROOD** *Results Viewer* displays the cluster heads in 2D along with a spreadsheet containing medicinal chemistry properties and scores. The currently browsed run can be changed in the *Results* list box or by selecting the corresponding spreadsheet tab. Additional results can be loaded by selecting the *Open Results* option in the list box or choosing *Open Results* from the *File* menu and choosing a hit list file. *Results* can be closed by clicking on the red *X* in the *Results* list.

Once a result hit list is selected, you can browse the cluster heads via the spreadsheet or by scrolling the 2D window. Clicking on "Open in VIDA" will launch the **vBROOD** *Results Browser* for more full-featured exploration of the clusters (see *Viewing Results in VIDA*).

If a set of results came from a run in the current **vBROOD** session, the *Log* file may be displayed by clicking on the *Log* button in the lower right-hand corner. This displays the command line output of **BROOD** and can be useful in diagnosing unexpected output.

2.6 Viewing Results in VIDA

VIDA is OpenEye's molecular visualization tool. Although **vBROOD** provides some facility for browsing results, **VIDA** allows for more full-featured cluster browsing. **vBROOD** can launch the *Results Browser* tool from either the last page of a **BROOD** run or from the *Results Browser* in **vBROOD**. Click "Open in VIDA" to launch **VIDA** with the **BROOD** Cluster Browser loaded.

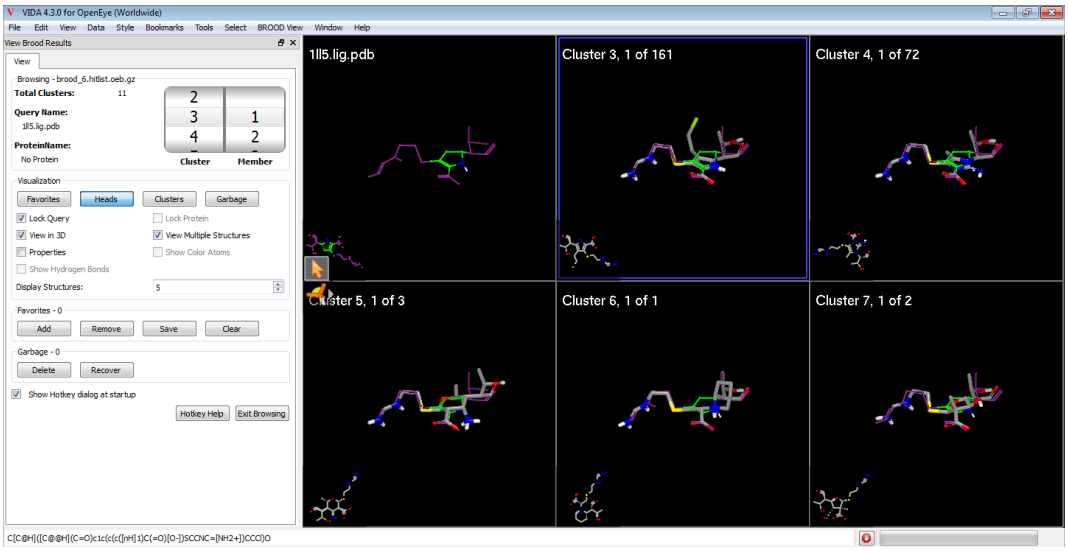


Fig. 16: **Figure 1.** The **BROOD** Cluster Browser in **VIDA**.

2.6.1 Getting Started

To open the **BROOD** Cluster Browser in **VIDA**, click "Open in VIDA" from either the *Browse Results* wizard or from the *Run BROOD* page in the either the *Run BROOD* wizard or the *Build and Run* wizard. If **VIDA** cannot be found or has not been previously launched, **vBROOD** will prompt you to locate it. The path to **VIDA** can also be specified in the **vBROOD** preferences dialog.

When **vBROOD** launches **VIDA** with the cluster browser, it first displays the new user interface and then loads the **BROOD** results. After the results have been loaded, the cluster browser will open the *Hotkey Help* dialog box displaying the various shortcut keys that can be used in the 3D window. This dialog can be closed by clicking "OK" and reopened by clicking on the *Hotkey Help* button in the **BROOD** Cluster Browser window.

The **BROOD** Cluster Browser window displays information about the hit list as well as tools for controlling the visualization of results, marking favorite molecules, and hiding other molecules in the *Trash*.

The **BROOD** Cluster Browser also adds a new menu, **BROOD** Viewer, which allows for loading and reloading **BROOD** results.

2.6.2 Viewing Clusters

The **BROOD** Viewer has four sections: *Browsing Information*, *Visualization Control*, *Favorites*, and *Garbage*.

Browsing

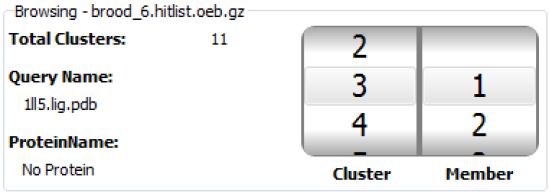


Fig. 17: **Figure 2.** The *Browsing* section of the **BROOD** Viewer.

The *Browsing Information* section provides general information about the displayed hit list. It shows which hit list, query, and protein are being displayed, information about the clusters in the file, and the current cluster being viewed.

The browsing window also contains a spinner that displays which cluster and which member you are currently browsing. The left spinner lists the clusters. If the selected cluster is blue, you are navigating the cluster heads. The right spinner shows the members of the current cluster. If the selected number in the right spinner is highlighted in blue, you are navigating with the cluster. This spinner can be useful for keyboard navigation.

Visualization

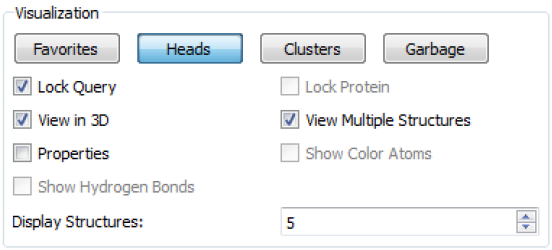


Fig. 18: **Figure 3.** The *Visualization* section of the **BROOD** Viewer.

The *Visualization* section contains the primary set of controls. The four buttons—*Favorites*, *Heads*, *Clusters* and *Garbage*—control which structures are being displayed in the main view.

- *Heads* navigates among the cluster heads.
- *Clusters* switches to browsing the current cluster members.
- *Favorites* displays any structures marked as favorites by the user.
- *Garbage* displays any molecules hidden in the garbage through the use of the Delete button.

The *Visualization* section also provides checkboxes to control the view.

- *Lock Query* displays the query alongside a result in each pane in 3D.
- *Lock Protein* displays the protein alongside a result in each pane in 3D.
- *View in 3D* toggles the cluster display between 3D and 2D.
- *View Multiple Structures* toggles between viewing a single result structure and a grid of structures.
- *Properties* toggles the display of a context-sensitive spreadsheet which displays medicinal chemistry properties for the current cluster.
- *Show Color Atoms* turns on the display of color atoms, if enabled by *Attach Color Atoms* under *Advanced Options* in **vBROOD**.
- *Show Hydrogen Bonds* toggles the display of hydrogen bonds between the protein and the result molecules.
- *Display Structures* controls the number of structures displayed in *Multiple Structure* view.

Favorites and Garbage

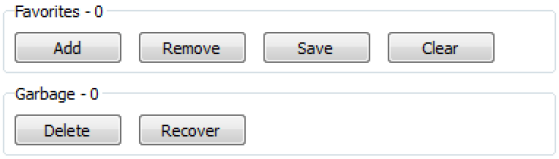


Fig. 19: **Figure 4.** The *Favorites* and *Garbage* sections.

The *Favorites* section allows users to tag individual molecules as *Favorites* for further follow-up or saving. The number of favorites are displayed in the caption. Below the caption are four buttons.

- *Add* puts the current structure (surrounded by the blue box) into the *Favorites* list.
- *Remove* takes the current structure (surrounded by the blue box) out of the *Favorites* list.
- *Save* gives a prompt for a file name, then saves the *Favorites* list to disk.
- *Clear* empties the *Favorites* list, moving all molecules back to their original location.

The *Garbage* section provides a way to hide structures from view. They are still available but are hidden from normal browsing. The number of structures in *Garbage* is displayed in the heading. The *Garbage* can be browsed by clicking *View Garbage*. Clicking *Recover* while viewing a structure in the *Garbage* will put the molecule back into its original cluster.

Keyboard Control

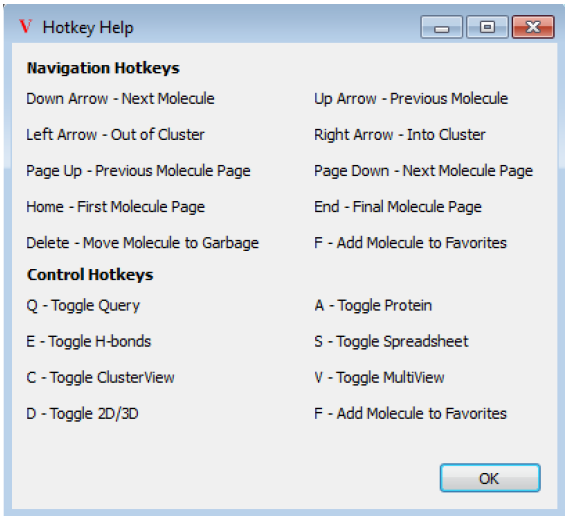


Fig. 20: **Figure 5.** The *Hotkey Help* dialog box.

The **VIDA** Cluster Browser script has key bindings for many common operations. The keyboard operations fall into two categories: cluster navigation and display control.

Cluster Navigation Hotkeys

The arrow keys are the primary navigation tool. They allow users to switch between navigating cluster heads and cluster members, as well as to navigate among the cluster heads or cluster members.

- The Right Arrow switches from navigating the cluster heads to the cluster of the currently selected head.
- The Left Arrow switches back from a given cluster to the cluster heads.
- The Up Arrow and Down Arrow navigate among the cluster or the cluster heads, whichever is currently being browsed.
- Page Up and Page Down navigate by several molecules at once when multiple structures are displayed.
- Delete moves a molecule to the *Garbage*.
- Home moves to the first page of molecules (in the cluster members or among the heads).
- End moves to the last page of molecules (in the cluster members or among the heads).

Display Control Hotkeys

Many of the controls available in the *Visualization* section are also keyboard accessible.

- *Q* toggles display of the query.
- *E* toggles the hydrogen bond display.
- *C* toggles between navigation of the heads and the clusters.
- *D* toggles between the 2D and 3D display.
- *A* toggles display of the protein (if any).
- *S* toggles display of the spreadsheet.
- *V* toggles display of multiple structures.
- *F* adds a molecule to favorites.

CHAPTER

THREE

BROOD

3.1 Overview

The basic idea behind **BROOD** is that you have a lead molecule and would like to generate analogs with somewhat different properties by changing a portion of the molecules. Normally, you will load the lead molecule into the GUI and select and edit the fragment you would like to replace. The GUI will also allow you to edit the chemical description of the query as well as define required motifs (constraints).

BROOD will search the database of fragments for fragments with similar shape, chemistry and electrostatics to your query fragment, replace them in your query molecule and generate a hitlist of analog molecules.

If you are just getting started with **BROOD**, we highly recommend you use the graphical interface. Even if you prefer to use the command line, walking through the graphical interface one time will give you a good overview of the workflow involved in running **BROOD**.

3.2 Input Files

Typically, a user will find it most convenient to generate the **BROOD** query using the graphical interface. Even if one wants to run **BROOD** on a cluster or other machine where it isn't convenient to use the graphical interface, the graphical interface can be used to generate the necessary input and then **BROOD** can be executed on a different machine after moving the input files. **BROOD** requires one input file, a query fragment file. With a default installation, the **BROOD** application can search either one of the default fragment database or a user-generated database.

3.2.1 Query Files

The entire query is encapsulated into a single query file, typically generated in the GUI. When generated, the GUI names the file *brood_1_query.oeb.gz* where the *1* is replaced with the next appropriate integer to maintain unique file names. This **BROOD** query includes the original molecule, the fragment that the user has chosen to replace, and the user-edited description of the shape, chemistry, and electrostatics of the query. In addition, if the query includes protein structures either for bump-check or selectivity-check, these too will be encapsulated in the query file.

BROOD carries out a 3D search, and a 3D search is required for the search and is usually included in the query file. The **BROOD** GUI can take either a 2D or a 3D query file. If the original input is 2D, a 3D structure will be generated in the GUI and used during the search.

3.2.2 Fragment Database

As mentioned in the theory section, **BROOD** comes with a pregenerated, multiconformer fragment database. This database is made from fragments of known molecules that contain 1-15 heavy atoms and 1-3 attachment sites.

BROOD can also search user-generated fragment files. A custom database can be created using *CHOMP*.

3.2.3 Example Fragment Files

The database file used for **BROOD** searches is in a highly optimized format with many precalculated properties and a complex structure intended to minimize disk access during the time-critical search. These database files can be generated by the **CHOMP** program included with your **BROOD** distribution. By default, **CHOMP** will generate fragments from complete molecules. On the other hand, if a user has their own fragment files they would like to search, these can be loaded directly into the **CHOMP** program and used to generate a searchable database.

In this case, an input database fragment for bioisosteric searching must be a molecular fragment with one or more "attachment points". An attachment point represents a bond from the fragment to the rest of the molecule. Inside **OEChem TK**, this is represented as a bond to an atom with atomic number 0 (termed a "Dummy Atom"). If it is necessary to uniquely distinguish these dummy atoms, Map Indices can be used via the SetMapIdx and GetMapIdx methods of OEAtomBase.

The input fragment does not need to have 3D coordinates (they can, optionally, be generated on-the-fly). This makes SMILES and 2D SDF file formats the most convenient for query fragment input. On the other hand, the fragment database should contain 3D coordinates with multiple conformers per fragment. In the examples below, a simple amide fragment with generic (amide) and uniquely labeled (amideR) dummy atoms are shown.

SMILES format:

```
*C(=O)N* amide
[*:1]C(=O)N[*:2] amideR
```

SDF format:

```
amide
  -OEChem-01040614232D

  5  4  0     0  0  0  0  0  0999 V2000
    0.0000    0.0000    0.0000 *   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 *   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  2  0  0  0  0
  2  4  1  0  0  0  0
  4  5  1  0  0  0  0
M  END
$$$$
amideR
  -OEChem-01040614232D

  5  4  0     0  0  0  0  0  0999 V2000
    0.0000    0.0000    0.0000 R#  0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 R#  0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  2  0  0  0  0
  2  4  1  0  0  0  0
  4  5  1  0  0  0  0
M  RGP  2   1   1   5   2
```

(continues on next page)

(continued from previous page)

```
M  END
$$$$
```

3.2.4 Molecular File Formats

BROOD can read and write a variety of molecular file formats. The file format is automatically interpreted from the filename suffix.

File Type	Input | Output	Extension
SMILES	Output	`.smi .ism .smi.gz .ism.gz`
CXSMILES	Output	`.cxsmiles .cxsmiles.gz`
SDF	Input and Output	`.sdf .sdf.gz`
MOL2	Input and Output	`.mol2 .mol2.gz`
OEBinary v2	Input and Output	`.oeb .oeb.gz .oez`
List files	Input	`.lst .list .lst.gz .list.gz`

Gzipped OEBinary version 2 (`oeb.gz`) is the recommended output format.

BROOD is capable of piping formatted input and output. The simple "-" can be used in place of a filename to indicate `std::cin` or `std::cout` with the default SMILES format.

```
prompt> brood -in .oeb.gz -db myDB < brood.query_1.oeb.gz
```

This execution will run **BROOD** with `std::cin` as the input with *.oeb.gz* format. The format is controlled by the suffix.

3.3 Example Commands

This section has a series of example **BROOD** command line executions. Most **BROOD** runs are carried out using **vBROOD**. Each example is followed by a brief description of its behavior.

If you would like to execute the following examples as written, the appropriate paths to the executable file and the database file must be included. In addition, the query file `q1.oeb` will need to be in the working directory.

```
prompt> brood -in q1.oeb -db chembl31
```

This command will execute BROOD with the default parameters. The file `q1.oeb` is opened as the query, and the database is read from the chembl31 folder. The default hit list will be written to `brood_hitlist.oeb.gz`, using the default -prefix argument "brood". Similarly, the informational output files `brood_hitlist_rpt.csv`, `brood_log.txt` and `brood_parm.txt` will also be written.

```
prompt> brood -in q1.oeb -db chembl31 -prefix 4dfr
```

This command is the same as the previous except that the prefix to all of the output files has been changed from "brood" to "4dfr" (for example, the log file will be written to `4dfr_log.txt` rather than `brood_log.txt`).

```
prompt> brood -param 4dfr_parm.txt
```

This execution of **BROOD** will read all the command line arguments from the file `4dfr_param.txt`. Every time **BROOD** is executed, a param file is generated that can be used to exactly reproduce the run (*vida infra*). This option is most useful when the job is set up in **vBROOD** and the query and param file are written out for later execution. Using

the command detailed here with a query and param file from **vBROOD** will give the same results as running the search from within **vBROOD**.

```
prompt> brood -param 4dfr_parm.txt -ringOnly 0
```

This example illustrates two points: that command line arguments take precedence over arguments in the `.param` file, and that the *-ringOnly* flag is an important flag in altering **BROOD** results. This command line will execute **BROOD** with the parameters from "`4dfr_parm.txt`", but the *-ringOnly* parameter will be overridden (taking precedence) over the *-ringOnly* parameter that is specified in `4dfr_parm.txt`. A similar outcome could be achieved by editing `4drf_parm.txt` in a text editor, but this command line alteration is a more direct means to the same behavior.

By default, **BROOD** only returns fragments that have at least two ring atoms on the shortest path between attachment points, or at least three ring atoms total if there is only one attachment point. This example shows that a user can turn off this constraint using the *-ringOnly* flag. If one wants to generate results that *only* avoid the rings as described above, one can pass **-1** to the *-ringOnly* parameter.

```
prompt> brood -in q1.oeb -db chembl31 -quickLook 120
```

This execution will use the query in `q1.oeb` to search the fragments in the database `chembl31`. The use of the *-quickLook* parameter will limit the search to approximately the number of seconds (120) provided after the flag. The parameter is a great way to get a quick notion of interesting results. The databases are organized to allow rapid identification of some interesting results. The full search can take minutes to hours, and is often more appropriate when designing molecules that may take days or weeks to synthesize; nevertheless, in an iterative design session, it can be useful to quickly generate ideas using the *-quickLook* parameter.

```
prompt> brood -in q1.oeb -db chembl31 -neutralpH false -tautomers false
```

When **BROOD** is building analogs by putting similar fragments into a molecular framework, sometimes the electronic environment of certain functional groups changes dramatically (amines converted to amides for instance). To properly handle these cases, it is useful for **BROOD** to normalize the ionization and tautomer states of the new molecules before generating the final hit list. In some cases, users are better served without these normalizations. The ionization and tautomer normalizations can be turned off with the *-neutralpH* and *-tautomers* flags, respectively.

```
prompt> brood -in q1.oeb -db chembl31 -scoreType linkOnly
```

By default, **BROOD** compares the shape, chemistry, electrostatics, and attachment geometry of fragments in order to suggest analog molecules. For some applications, such as joining fragments or closing rings, it is useful to build fragments into empty space. In these cases, one wants to only compare the attachment geometry of fragments, in a manner similar to the original CAVEAT searches [Lauri-1994]. The *-scoreType* `linkOnly` flag instructs **BROOD** to compare fragments using only their ability to form low-energy bridges between attachment points.

```
prompt> brood -in q1.oeb -db chembl31 -property false
```

By default, **BROOD** filters the final analog hit list with a series of property filters, each specified with a pair of range parameters. While this can be useful, on some occasions, users prefer to avoid all of the property filters. By passing *false* to the *-property* flag, all of the property filters can be turned off in one easy step.

3.4 Command Line Help

A description of the command line interface can be obtained by executing **BROOD** with the *–help* option.

```
> brood --help
```

will generate the following output:

```
Help functions:
 brood --help simple      : Get a list of simple parameters
 brood --help all         : Get a complete list of parameters
 brood --help defaults    : List the defaults for all parameters
 brood --help <parameter> : Get detailed help on a parameter
 brood --help html        : Create an html help file for this program
 brood --help versions    : List the toolkits and versions used in the application
```

3.5 Parameters

3.5.1 File Options

-log

The argument for this flag specifies the name of the log file. The level of detail for logfile information can be altered using the *-verbose* flag. Generation of an output log may be disabled by providing `nul` as a argument in Windows and `/dev/null` as an argument on Linux and macOS. [default = `prefix` _log.txt]

-prefix

The argument for this flag defines the prefix to be used for various information and data files generated. Most important among these is the '*prefix_parm.txt*' file which includes a copy of all the parameters used in the run. The prefix is also used to generate a default log file name if not explicitly specified with the *-log* flag. [default = <app_name>].

-verbose

This is a boolean flag that controls the level of detail written to the log file. By default only minimal information is written to the log file. Verbose logging will cause more information to be written to the log file in order to follow behavior during program execution. [default = false]

-molNames

This parameter takes a text file containing a list of molecule names (one name per line in the file). If this parameter is set then only molecules in the input file(s) (see parameter *-in*) with names that match those in the text files will be read in.

The general purpose of this flag is to provide an easy mechanism for reading a few specific molecule(s) that are contained in a large file, without having to extract those molecules by hand from the input molecules file. [default = NONE].

-in (-qMol, **-queryMol**)

Molecular file encapsulating the query. It contains the primary molecule of interest, the query fragment, description of the query potential function and constraints, and the protein structures for the active site and for selectivity.

-out (-o)

Output filename to be generated. Accepted file formats are: `cxsmiles`, `ism`, `mol2`, `oeb`, `oez`, `sdf`, `smi`.

Execute Options

-param

The argument for this flag is the name of a file containing control parameters. The control parameter file acts to either replace or augment the command line interface. All parameters necessary for program execution may be provided in the control parameter file, although any command given explicitly on the command line will supersede options found in the parameter file. The application generates a new parameter file containing the full set of execution parameters upon every execution. The name of the parameter file is created by combining the prefix base name with the *'_parm'* suffix and the *'.txt'* extension.

-mpi_np <n>

Specifies the number of processors n when the application is run in MPI mode.

-mpi_hostfile <filename>

Specifies the name of the file containing the processor's configuration. For every host, this file should contain a line `host_name slots=n` where n is the number of processors on the host.

BROOD App Options

-db (-database)

Specified the directory containing the **BROOD** database files. Additional details on the files contained in the directory can be found in the section on the *Fragment Database*.

-cpddb

The user can specify a file of available compounds (either commercial or internal) with this flag. If specified, molecules from this file that are similar to any created analogs will be used to annotate the newly suggested molecule. More details about analog similarity is described in *Similarity to available compounds* method.

-quickLook <n> (-quick)

This flag allows a user to do a quick **BROOD** search (in n number of seconds). **BROOD** is designed to identify as many interesting hits as soon as possible. Thus while a thorough search may take minutes to hours, useful preliminary results can be generated in approximately n seconds with this option.

BROOD Hit List Builder Options

-idea (-cluster)

This flag determines if the hit list will be organized according to a reduced graph hierarchy. This is quite useful for clustering similar analogs and thus allowing a user to quickly scan the different analog families and drill down into the most interesting clusters without needing to examine hundreds of analogs. [Default=true]

-maxHits (-maxHit)

This flag determines the number of compounds saved in the hit list. Allowable range is 1–50000. **The legal range of this parameter has been increased by an order of magnitude.** [Default=1000]

-buildType

This flag determines the type of building to perform. Accepted values are `2d` (2 dimensional build), `noopt3d` (3 dimensional build without optimization), and `optimized` (3 dimensional build with optimization). [Default=optimized]

-checkBond

If this flag is set to `True`, **BROOD** will check all bonds formed between new fragments and the rest of the query molecule and eliminate fragments that form bond types that are typically unstable. Rather than being discarded entirely, these hits will be written in the `brood_removed.oeb.gz` file by default. [Default=true]

-deltaLocalStrain

Boolean flag indicating if the relative local strain between query and analog should be used as a measure of strain. [Default=false]

-maxLocalStrain

Maximum local strain allowed to fit an analog molecule on top of a query. [Default=6.5]

-neutralpH

This flag controls whether the final molecules are set to an ionization state suitable at pH=7.4. Because new bonds are formed in the process of generating analogs, functional groups sometimes need adjustment even if their state was properly normalized prior to fragmentation. This flag allows the user to turn off pKa normalization. [Default=true]

-tautomers (-tautomer)

Similarly, the *-neutralpH* adjustment of tautomer states can be necessary after joining the new fragments with the static portions of the original molecule. This flag allows the user to turn off this normalization. [Default=true]

BROOD Match Options

-attachCutoff (-attachmentCutoff, **-attachCut)**

Minimum acceptable attachment point score cutoff. The cutoff is for the shape overlap score of the beginning and ending atom of each attachment bond. The default was chosen empirically to assure all fragments in the hit list would have sensible attachment geometries. This value does not affect runs when the *-scoreType* flag is set to `linkOnly`. [Default=0.78]

-bondOrder (-bo)

When this Boolean flag is set, only fragments with the same attachment bond orders as the query fragment are allowed. [Default=true]

chargeType (-fileChrg)

When this flag is set to `fileChrg`, the partial charges for electrostatic Tanimoto calculations are taken from the input files. If this flag is set to `mmff94`, or the input file does not contain partial charges, MMFF94 charges will be used. The partial charges on the database molecules are pregenerated. [Default=mmff94]

-checkGeometry

By default, regardless of the attachment score (used primarily to drive the optimization), the fit of a fragment at the attachment points is controlled by geometric constraints. These require that the two attachment vectors that will be jointed into a bond are overlapping and pointing in roughly opposite directions. In some cases, particularly when the required geometry is unusual or strained, it can be useful to turn off this constraint check. [Default=true]

-property (-prop)

This Boolean flag indicates whether ANY of the molecular properties should be used to eliminate compounds from consideration. While these filters can be useful, if a user expects to see a particular fragment or analog and it does not appear in the hit list, it will often have been eliminated by the property filters. The *Report Files* will indicate which, if any, property filter affected each fragment. If set to false, all of the related flags below become irrelevant. [Default=false]

-ringOnly (-ring)

This is a complex flag whose purpose is to allow the user to control the number of ring atoms in the selected fragments. It allows input of "-2, -1, 0, or an integer 1 <= x <= 12". This flag has to do with a count of the number of ring atoms in the shortest path between attachment points in a fragment. In cases with more than two attachment points, all shortest paths are calculated and the number of ring atoms is summed.

By default, the flag is set to "-2" (true) and requires at least two ring atoms in the path. If the flag is set to "0" (false), no ring atom filtering is done, while if the flag is set to "-1" (none), only fragments without ring atoms in the shortest paths are returned. If the flag is set to a number between 0 and 13, then that will be the **minimum** number of ring atoms required. [Default=-2]

-shapeCutoff

Indicates the minimum required shape Tanimoto score required for a fragment to appear in the color, elect, or queryAnalog hit lists. This cutoff is useful for cases when few shape-similar fragments exist in the database being searched. [Default=0.6]

BROOD Score Options

-attachScale (-attachmentScale, **-attach**, **-aScale**)

This floating point value determines the balance between the chemical color score and the attachment point scores. Higher values indicate more weighting for the attachment-point alignment. This parameter has complex effects; please use it with care. [Default=1.5]

-bumpRadius

This real number flag determines the minimum distance between ligand heavy atoms and protein heavy atoms in the final relaxed solution for there to be considered a clash. New analogs with atoms closer than this cutoff to the active site protein are discarded from the hit list. If a selectivity protein is being used, at least one clashing atom must be found in order for the hit list ligand to be retained. Because the molecular model of the protein is rigid, it may be advantageous for this cutoff to be shorter than in a system with a flexible protein model. [Default=2.25]

-ignoreProtein (-noqueryprot, **-noqueryprotein**)

When a query is created with the GUI and an active site protein is included, the section active site protein becomes part of the query. This flag makes **BROOD** ignore the active site protein even though it is part of the query. The flag makes it easy to run the **BROOD** job with and without the active site protein. [Default=false]

-ignoreProteinSelect (-noqueryselectionprotein, **-noqueryselection**)

When a query is created with the GUI and a selectivity protein is included, the selectivity protein becomes part of the query. This flag makes **BROOD** ignore the selectivity protein even though it is part of the query. The flag makes it easy to run the **BROOD** job with and without the selectivity protein. [Default=false]

-rangeOffset

By default, the range of fragment heavy atoms specified by the *-rangeSize* flag is centered around the same number of heavy atoms as the original query fragment. Instead, the user can use this flag to bias a search toward smaller or larger fragments with positive or negative values passed to this flag. [Default=0]

-rangeSize

By default, only fragments with a number of heavy atoms +/- N from the fragment being replaced are considered. This flag lets the user control this setting. It should not be necessary to change this value except in unusual circumstances. This flag determines the range of heavy atoms around query fragment's number of heavy atoms to examine. [Default=6]

-scoreType

This flag can have three legal values: `rocs`, `et`, and `linOnly`. If set to `rocs`, the shape and color Tanimoto similarities are calculated. If set to `et`, the electrostatic Tanimoto similarity is calculated on all of the fragments that also have a good shape score. If this flag is set to `linkOnly`, then the shape and chemistry of the fragment are ignored and ONLY the attachment point geometry (and constraints) are used to identify similar fragments. This is similar to Bartlett's Caveat algorithm, one of the first algorithms in this genre. This type of search can be useful when trying to bridge one or more fragments without any prior knowledge, such as when linking two fragments in fragment-based design.[Default=rocs]

Molecule Property Options

-eganEgg (-egan)

This Boolean parameter determines whether analog compounds will be required to fulfill the "Egan egg" measure of bioavailability. This measure was published by Bill Egan while at Pharmacopia [Egan-2000], and rejects compounds with a LogP > 5.88 or a PSA > 131.6. [Default=true]

-veber (-gsk)

This Boolean parameter determines whether analog compounds will be required to fulfill the measure of bioavailability Veber published at GSK [Veber-2002]. His measure of bioavailability eliminates compounds with a PSA > 140 or more than 10 rotatable bonds. [Default=false]

Extrinsic Property Values

-minAcceptors (-minLipinskiAcceptors, **-minLipinskiAcc)**

-maxAcceptors (-maxLipinskiAcceptors, **-maxLipinskiAcc)**

These flags indicate the upper and lower bound of the Lipinski hydrogen bond acceptor range of any analog. Analogs higher or lower in number of Lipinski hydrogen bond acceptors than these cutoffs will be eliminated from consideration. For the purpose of this measure, hydrogen bond acceptors are determined by the method of Lipinski [Lipinski-1997]. [Default=2, 11]

-minComplexity (-minComp)

-maxComplexity (-maxComp)

These flags indicate the upper and lower bound of the molecular complexity range of any analog. Analogs higher or lower in molecular complexity than these cutoffs will be eliminated from consideration. For more on molecular complexity, please see the theory section. [Default=0.0, 2.5]

-minDonors (-minLipinskiDonors, **-minLipinskiDon)**

-maxDonors (-maxLipinskiDonors, **-maxLipinskiDon)**

These flags indicate the upper and lower bound of the Lipinski hydrogen bond donor range of any analog. Analogs higher or lower in number of Lipinski hydrogen bond donors than these cutoffs will be eliminated from consideration. For the purpose of this measure, hydrogen bond donors are determined by the method of Lipinski [Lipinski-1997]. [Default=1,8]

-minFrequency (-minFreq)

-maxFrequency (-maxFreq)

These flags indicate the upper and lower bound of the frequency range of any analog. Analogs higher or lower in frequency than these cutoffs will be eliminated from consideration. For the purpose of this calculation, the frequency is a percentile number indicating the frequency of each fragment normalized relative to the frequency of fragments in ChEMBL. Frequency as assessed here is a measure of how common each fragment is among the source molecules. The most commonly occurring fragment would be in the 99th percentile, while the least commonly occurring fragments would be in the 1st percentile. [Default=0, 100]

-minLipinski

-maxLipinski

These flags indicate the upper and lower bound of the Lipinski violations range of any analog. Analogs higher or lower in number of Lipinski violations than these cutoffs will be eliminated from consideration. [Lipinski-1997]. In Lipinski's work, in order to segregate molecules that progressed through clinical trials, he determined that one violation was acceptable, but two were not. [Default=0, 1]

-minlogp

-maxlogp

These flags indicate the upper and lower bound of the calculated LogP range of any analog. Analogs higher or lower in LogP than these cutoffs will be eliminated from consideration. [Default=-1.0,5.0]

-minMartin (-minAbbott, **-minABS)**

-maxMartin (-maxAbbott, **-maxABS)**

These flags indicate the upper and lower bound of the Abbott Bioavailability Score (ABS) range of any analog. Analogs higher or lower in ABS than these cutoffs will be eliminated from consideration. This floating point ABS parameter (range 0.0-1.0) indicates the minimum allowable probability that F will be >10% in rats according the QSAR model developed and published by Yvonne Martin [Martin-2005]. A value of 0.0 will allow all compounds to pass. [Default=0.2, 1.0]

Intrinsic Property Values

-minAromFJCt

-maxAromFJCt

These flags indicate the upper and lower bound of the aromatic ring range of any analog. Analogs higher or lower in number of aromatic rings than these cutoffs will be eliminated from consideration. For the purpose of this calculation, the number of aromatic rings is the number of aromatic rings is #aromatic bonds - #aromatic atoms + 1. [Default=0, 5]

-minFsp3C

-maxFsp3C

These flags indicate the upper and lower bound of the fraction of carbons that are sp^3 range of any analog. Analogs higher or lower in the fraction of carbons that are sp^3 than these cutoffs will be eliminated from consideration. There is some evidence that increasing the fraction of carbons in a series that are sp^3 (escaping from "Flatland") improved success in clinical trials [Lovering-2009]. [Default=0.3, 1.0]

-minHvyAtom (-minHeavyAtom)

-maxHvyAtom (-maxHeavyAtom)

These flags indicate the upper and lower bound of the heavy atom range of any analog. Analogs higher or lower in number of heavy atoms than these cutoffs will be eliminated from consideration. [Default=7,35]

-minMolWt (-minmolecularweight)

-maxMolWt (-maxmolecularweight)

These flags indicate the upper and lower bound of the molecular weight range of any analog. Analogs higher or lower in molecular weight than these cutoffs will be eliminated from consideration. [Default=100,500]

-minRotBond (-minRotatableBond)

-maxRotBond (-maxRotatableBond)

These flags indicate the upper and lower bound of the rotatable bond range of any analog. Analogs higher or lower in number of rotatable bonds than these cutoffs will be eliminated from consideration. [Default=0,13]

-minpsa (-mintpsa)

-maxpsa (-maxtpsa)

These flags indicate the upper and lower bound of the topological polar surface area (TPSA) range of any analog. Analogs higher or lower in TPSA than these cutoffs will be eliminated from consideration. [Clark-1999] [Ertl-2000]. [Default=60,150]

3.6 Output Files

BROOD generates seven output files in addition to the hitlist. All of these files begin with the prefix specified by the *-prefix* flag, that by default is "brood". They also include an integer to retain unique file names in the working directory (in this example, we will use the integer "1"). These files include:

1. `brood_1_info.txt`, the info file (This file is only created if the run is initiated from the GUI.)
2. `brood_1_log.txt`, the log file
3. `brood_1_parm.txt`, the param file
4. `brood_1_hitlist_rpt.csv`, the hitlist report file
5. `brood_1_removed_rpt.csv`, the removed hits report file

By default, there is one hitlist and the contents of the hitlist will be determined by the command flags.

- `brood_1_hitlist.oeb.gz`

In postprocessing of the search, some fragments are removed from the hitlist. These are preserved in the "removed" file.

- `brood_1_removed.oeb.gz`

If the *-scoreType* `et` flag is used, the hitlist will include molecules electrostatically similar to the query.

Before executing the search, the GUI also writes the encapsulated query into the query file.

- `brood_1_query.oeb.gz`

3.6.1 Info File

By default, an information file titled `brood_1_info.txt` is written or updated for every 500 fragments that are processed. Reading this info file allows a user to monitor the progress of the database search while it is occurring and it serves as a record of the performance of that search after the execution is completed. During execution, the GUI uses this file as well as direct communication with the application to keep the user apprised of all information available in the info file.

An example info file is shown below.

```
********** Progress Update **********
Percent of Database Processed    = 100%
Number of Packets Processed      = 578
Number of Fragments Processed    = 1642376
Number of Fragments Overlayed    = 1094381
Number of Hits                   = 909
Number of Warnings               = 0
Number of Errors                 = 0
Processed fragments/sec          = 896
Elapsed Time (sec)               = 1832
*************************************
```

If any errors are noted in the info file, it is strongly suggested that a user check in the log file to determine the nature of the problem!

3.6.2 Param File

BROOD's command line interface can be efficiently run using the *-param* command line parameter followed by the name of a parameter file. Param files are files that contain one command line parameter on each line. Every execution of **BROOD**, including executions started within the GUI, generates a .param file called `brood_1_parm.txt`. This file contains all of the parameters used by **BROOD**. Further, this file can be used in subsequent runs with the *-param* flag either with or without user modifications.

The `_parm.txt` file is particularly useful if you want to use the graphical interface to set up a job, but execute the job on another machine (such as a cluster). The `_parm.txt` file can be moved to a different machine along with the *-in* -queryMol file it specifies, and it can be used to execute exactly the same run as would have been executed from within the graphical interface.

The following is an example of a `_parm.txt` file generated by the GUI (`brood_1_parm.txt`). It can be used to regenerate the run using the command line below.

This file could be edited or used as-is with subsequent runs. In addition, any explicit command line parameter takes precedence over parameters in the .param file, so this mechanism can be used to execute a series of similar jobs.

```
prompt> brood -param brood_1_parm.txt

prompt> brood -mpi_np 16 -param brood_1_parm.txt -prefix brood_mpi1
```

Listing of `brood_parm.txt` file contents:

```
#Interface settings

#Mode brood
#-failed (Not set, no default)
-in  /Users/Chemist/Desktop/brood.query.oeb.gz
#-log
#-molNames
-out  brood_hitlist.oeb
-prefix  brood
#-report
#-verbose  false

#Execute Options :
    -param  /Users/Chemist/Desktop/brood_parm.txt
    #-mpi_np (Not set, no default)
    #-mpi_hostfile (Not set, no default)

#Brood App Options :
    #-cpddb
    -db  /Users/Chemist/WorkingDirectory/chembl31
    #-quickLook  0
    -status  true

    #BROOD Hitlist Builder Options :
        -idea  true
        -maxHits  1000

        #Molecule Builder options :
            #-buildType  0
            -checkBond  true
            -deltaLocalStrain  false
            -maxLocalStrain  6.5
            -neutralPH  true
            -tautomers  false

    #BROOD Match Options :
        -attachCutoff  0.78
        -bondOrder  true
        -chargeType  5
        -checkGeometry  true
        -property  false
        -ringOnly  -2
        -shapeCutoff  0.6

        #Molecule Property Options :
            #-eganEgg  true
            #-veber  false
```

(continues on next page)

(continued from previous page)

```
        #Maximum allowed extrinsic property values :
            #-maxAcceptor  11
            #-maxComplexity  1.25
            #-maxDonor  8
            #-maxFreq  100
            #-maxLipinski  1
            #-maxLogp  5.0
            #-maxMartin  1.0

        #Maximum allowed intrinsic property values :
            #-maxAromFJCt  5
            #-maxFsp3C  1.0
            #-maxHvyAtom  35
            #-maxMolWt  500.0
            #-maxRotBond  13
            #-maxTpsa  150

        #Minimum required extrinsic property values :
            #-minAcceptor  2
            #-minComplexity  0.0
            #-minDonor  1
            #-minFreq  1
            #-minLipinski  0
            #-minLogp  -1.0
            #-minMartin  0.2

        #Minimum required intrinsic property values :
            #-minAromFJCt  0
            #-minFsp3C  0.3
            #-minHvyAtom  7
            #-minMolWt  100.0
            #-minRotBond  0
            #-minTpsa  60

    #BROOD Score Options :
        -attachScale  1.5
        -bumpRadius  2.25
        #-ignoreProtein  false
        #-ignoreProteinSelect  false
        -rangeOffset  0
        -rangeSize  6
        -scoreType  0
```

```
prompt> brood -param brood_parm.txt
```

This would result in exactly the same execution as the one which generated the file above.

3.6.3 Log File

The log file contains all of the critical information about the execution in one place. It begins with a copy of the param file and it finishes with the final info file output. In between, it contains all the warnings and errors that might have occurred during execution. The log file gives a user a single place they can check to determine what job was run, if it executed properly, and how long it took.

3.6.4 Report Files

Report files contain a detailed listing of the similarity scores for every molecule in the database that was sufficiently similar to warrant being in the hitlist. By default, the file titled `brood_1_hitlist_rpt.csv` contains information about every molecule that remained in the hitlist after postprocessing, and `brood_1_removed_rpt.csv` contains information about every molecule that was removed from the hitlist during the postprocessing.

Report files contain columns for

- database fragment SMILES
- database fragment title
- query SMILES
- number of attachment bonds
- structural rms value
- attachment score
- shape Tanimoto
- color Tanimoto
- combo score (shape + color)
- et attachment score
- et shape Tanimoto
- electrostatic Tanimoto
- et combo score (shape + et)
- a comment regarding the disposition of the fragment (if and why it failed to be built)

If a particular score was not calculated for any given fragment, "-" will be found in the report file under the corresponding column. Files are in comma-separated value format and can easily be imported into a spreadsheet program for further analysis of the results.

3.6.5 Hitlist File

As discussed above, a hitlist is generated for each execution of **BROOD**. All of the hitlists are written in the file format specified by the output file name *-out*, which defaults to gzipped OEB format `.oeb.gz`, if the *-out* is not specifically used. For instance, if *-out* `Output.oeb` is specified, the hitlist will have OEB format instead. The first entry in the hitlist is the query molecule. Each subsequent analog molecule in the hitlist is oriented in the optimum overlay on the query molecule. For the *-scoreType* `linkOnly` hitlist, this overlay is the optimum overlay of the attachment point atoms. In addition, by default, the similarity scores and physical property data for each molecule are attached as SDTags (OEB and SD format only).

By default, the hitlist files are written periodically while the search is being carried out. This allows a user to examine results at intermediate stages without waiting for the entire search to complete.

If the *-in* flag specified a 2D input molecule, then all of the analog molecules in the hitlist will likewise be in 2D format, though the fragment similarity search will have been carried out in 3D. When using the graphical interface with a 2D input molecule, a 3D molecule is generated prior to processing and the output format will always be 3D.

Below we provide a description of some of the SDtags attached to the hitlist molecules:

AroRingCt: Number of aromatic rings in the molecule

ClusterID/IdeaGroup: ClusterID of the molecule

color: Color Tanimoto score of the replacement fragment against the query fragment

combo: Shape + color Tanimoto combo score of the replacement fragment against the query fragment

Egan: Boolean specifying whether the molecule passes the Egan bioavailability model

Fragment: SMILES string of the replacement fragment

freq: Frequency of the replacement fragment

fsp3C: Fraction of sp^3 hybridized carbon atoms in the molecule

HvyAtoms: Number of heavy atoms in the molecule

LipinskiDon: Number of Lipinski donors in the molecule

LipinkskiAcc: Number of Lipinski acceptors in the molecule

LipinskiFail: Boolean specifying whether the molecule fails Lipinski's Rule of Five

Local strain: Calculated local strain of the molecule

Delta Local strain: Calculated delta local strain of the hit with respect to the query molecule

Molecular TanimotoCombo: Shape + color Tanimoto combo score of the molecule against the query molecule

MolWt: Molecular weight of the molecule

p(active): Belief score of the molecule [Muchmore-2008]

RingCt: Number of ring atoms

RingRatio: Ratio of the number of ring atoms to the total number of heavy atoms

Rotors: Number of rotatable bonds in the molecule

shape: Shape Tanimoto score of the replacement fragment against the query fragment

Source Mols: SMILES strings of the molecules the replacement fragment is part of

Source Mol Labels: Labels of the molecules the replacement fragment is part of

tPSA: Calculated topological polar surface area of the molecule

Veber: Boolean specifying whether the molecule passes the Veber bioavailability model

XlogP: Calculated LogP of the molecule

3.6.6 Removed File

At the end of the primary fragment search, a complete hitlist is generated. In postprocessing, the fragments are built into whole molecules and tested for strain, the newly formed bonds are examined for chemical stability, potential duplicates in the hitlist are removed, and the new molecules are checked for bumps with the protein environment and against the optional selectivity protein. At any of these stages, fragments that are similar according to the primary search criteria may be removed from the hitlist for these secondary criteria. In rare cases, few fragments may remain in the final hitlist and the user may want to examine the removed fragments without needing to rerun the entire search. For this reason, any fragment removed at the postprocessing stage is written to the removed hitlist `brood_1_removed.oeb.gz`.

CHAPTER

FOUR

CHOMP

4.1 Overview

BROOD includes a default database of approximately six million molecular fragments that are carefully selected to be medicinally relevant. While this database is quite thorough, we recognize that some users may want to supplement the default database with fragments generated from their own proprietary molecule collections. To facilitate this process, the **BROOD** distribution includes two programs, **CHOMP** and **BroodDBMerge**.

CHOMP allows users to fragment molecules, filter the fragments, generate 3D conformations, organize and index the fragments for rapid searching, and write a **BROOD** database. This process can be carried out from start to finish using a **BROOD** license and does not require access to **OMEGA**, **FILTER**, **MolProp**, or **OEChem** licenses. **BroodDB-Merge**, a utility that allows users to merge their own corporate databases with the OpenEye default database, will be covered in detail in the following chapter.

At its simplest, **CHOMP** takes molecules as input and fragments them to generate **BROOD** databases; simply specify the molecules you want to use to generate fragments using the `-in` flag and the output database you want to create with the `-out` flag.

By default, **CHOMP** assumes you want to create a database from your proprietary collection in order to augment the default database. **CHOMP** does not automatically remove any duplicates from the generated database. To remove duplicates, you can specify the database of fragments from which you would like to eliminate duplicates with the `-userUnique` flag.

The output from **CHOMP** will be a directory filled with a **BROOD** database. To search the database with **BROOD**, specify the *directory name* with **BROOD**'s `brood -db` flag.

While the basic use of **CHOMP** is straightforward, there are many options for controlling the **CHOMP** process and filtering the fragments.

> **Note**
>
> **CHOMP** input molecules are pKa normalized, that is, set to pH=7.4 ionization state.

4.2 Example Commands

This section gives a series of examples of **CHOMP** command-line executions. Unlike **BROOD**, **CHOMP** does not have a GUI and must be accessed from the command line. Each of these examples is followed by a brief description of its behavior.

```
prompt> chomp -in mymolecules.smi -out myfragDB
prompt> chomp -i mymolecules.smi -o myfragDB
prompt> chomp mymolecules.smi myfragDB
```

All three of these command lines specify exactly the same thing. In each case, **CHOMP** will read the molecules in `mymolecules.smi` and write the **BROOD** database in the directory `myfragDB`. This is the most basic and most common **CHOMP** execution. Since the *-dots* flag defaults to `True`, dots will be written to `std:cout` to indicate the progress of the molecular fragmentation as the job progresses. The `myfragDB` directory will be filled with just over 40 files that contain all the information critical for efficient 3D **BROOD** searches.

```
prompt> chomp -param oldrun.param
```

This execution of **CHOMP** will read the command-line parameters from the file `oldrun.param`. Every time **Chomp** is executed, a file called `chomp.param` is written that records the command-line parameters used. This is useful for recalling what was used in a specific execution or for repeating a previous calculation.

```
prompt> chomp -userFrags myFrags.sdf -out newDB
```

This reads the molecular fragments from the `myFrags.sdf` and, bypassing the fragmentation phase, generates 3D conformers and relates precalculated parameters, and writes the database into multiple files in the `newDB` directory.

```
prompt> chomp -in myMolecules.sdf -userUnique oldDB -out newDB
```

This execution of **CHOMP** behaves the same as the prior execution, except that every fragment generated by the fragmentation of `myFrags.sdf` will be compared to the fragments in the **BROOD** database `oldDB` (generated earlier). Fragments that exist in the old database will not be included in the `newDB`.

```
prompt> chomp -in myMolecules.sdf -omega false -out newDB
```

This reads and fragments the molecules from `myMolecules.sdf`, then writes all of the fragments (including source information) to the SMILES file `newDB.ism`. It will not generate conformers nor will it write a **BROOD** database. The newly generated SMILES file is available for examination and editing. After this partial run, the database preparation can be completed by passing the SMILES `newDB.ism` into **CHOMP** using the *-userFrags* parameter.

```
prompt> chomp -in myMolecules.sdf -readConfs xtals.sdf -out newDB
```

This execution of **CHOMP** reads and fragments the molecules from `myMolecules.sdf`. Then, rather than using OMEGA to generate conformers for the database, **CHOMP** uses the conformers read from the structures in `xtals.sdf` to generate conformers for the fragments. These conformers will be written into the new **BROOD** database `newDB`.

```
prompt> chomp -smarts bondids mymolecules.smi myfrags.oeb.gz
prompt> chomp mymolecules.smi myfrags.oeb.gz -smarts bondids
```

This example demonstrates two important principles. The first is that the first of these two command lines will work, but the second will result in an error. When specifying a command line with keyless arguments for the *-in* and *-out* files, these files must be the final two arguments on the command line.

The second principle involves use of non-default fragmentation patterns. In this example, **CHOMP** uses the SMARTS patterns in the file `bondids` to generate fragments rather than the default fragmentation scheme.

```
prompt> chomp -mpi_np 16 -in mymolecules.sdf -out newDB
```

In this execution, the `mymolecules.sdf` is read, the molecules are fragmented, 3D conformers are generated and the database is written to the directory `newDB`. The fragmentation and the more expensive conformer generation will be carried out on 16 local processors. On modern multicore workstations, this parameter is often a highly efficient means to execute **CHOMP**.

4.3 Command Line Help

A description of the command line interface can be obtained by executing **CHOMP** with the *–help* option.

```
> chomp --help
```

will generate the following output:

```
Help functions:
 chomp --help simple      : Get a list of simple parameters
 chomp --help all         : Get a complete list of parameters
 chomp --help defaults    : List the defaults for all parameters
 chomp --help <parameter> : Get detailed help on a parameter
 chomp --help html        : Create an html help file for this program
 chomp --help versions    : List the toolkits and versions used in the application
```

4.4 Required Parameters

-in (-i)

This flag specifies the input molecule file that contains the database of molecules you want to break up into a fragment database. If you have pregenerated fragments rather than whole molecules, please see the *-userFrags* flag (see below).

-out (-o)

This flag specifies the location in which to create the **BROOD** database. **CHOMP** will create a new directory using the specified flag and will fill that new directory with a series of files.

4.5 Optional Parameters

4.5.1 Execute Options

-param

The argument for this flag is the name of a file containing control parameters. The control parameter file acts to either replace or augment the command line interface. All parameters necessary for program execution may be provided in the control parameter file, although any command given explicitly on the command line will supersede options found in the parameter file. The application generates a new parameter file containing the full set of execution parameters upon every execution. The name of the parameter file is created by combining the prefix base name with the *'.param'* extension.

-mpi_np <n>

Specifies the number of processors n when the application is run in MPI mode.

-mpi_hostfile <filename>

Specifies the name of the file containing processors configuration. For every host this file should contain a line `host_name slots=n` where n is the number of processors on the host.

Input parameters

-userFrags

This option is similar to the *-in* flag except that it is for passing pregenerated fragments rather than for passing whole molecules. Using this flag skips the initial cutting apart stage; fragments are passed directly into the conformer generation and database construction phase. Fragments passed using this flag will be added to the

database in the order in which they are read (no resorting occurs). It is important to note that the *-in* and *-userFrags* flags can be used together.

-userUnique

This flag allows you to specify a database you would like to use for duplicate removal. The fragments in the specified database will be determined as unique; if the fragment is seen again in the current run, those fragments will be skipped rather than incorporated into the new database.

Output parameters

-prefix

This string flag determines the prefix of the `info`, `log`, `report`, `param`, and `output` files. For example, if *-prefix* is set to foo, then the output files will include `foo.info`, `foo.log`, `foo.report`, and `foo.param`. [Default=chomp]

-dots

When this flag is set, the program will write a series of dots (.) to the terminal (`stdout/cout`) to track the progress of the program. The dots are written in two phases: first as the molecules are cut apart and organized and second as the conformers are generated and the database is written. [Default=true]

Control parameters

-smarts

This flag specifies the built-in SMARTS pattern to use to identify bonds to break, or a file containing user-defined rules for identifying bonds to break.

Legal values are `rlf`, `recap`, `both` (both `rlf` and `recap`), `all`, or a filename containing user-defined rules.

Please see *CHOMP Theory* for a more detailed description.

-capAttach

When a fragment is generated with multiple attachment points, this flag controls whether all the related fragments can be generated by capping one or more of the attachment points with a hydrogen. This can generate many more fragments, but is quite efficient. [Default=true]

-flipper

This flag controls whether the flipper algorithm is used to generate the 2^N specific isomers by enumerating the stereo on unspecified stereocenters. The flipper flag takes an integer indicating the N in the 2^N isomers that can be generated from N stereocenters. If the stereocenters in the molecule exceed N, then 2^N random unique isomers are sampled from the available isomers. By default, no more than 32 isomers will be generated from any molecule. [Default=5]

-forceFlip

The *-forceFlip* flag causes all stereocenters in a ligand to be enumerated, even if they had specified stereo on the input structure. While this can generate fragments that are not specifically represented in the input molecules, it is also a means of exploring more of the chemical fragment space. [Default=false]

-flipN

This flag controls whether or not nitrogen atoms with potential stereochemistry are enumerated. Some nitrogens atoms have real stereochemistry, while most others are easily invertible in solution. In fragment replacement, specific invertible static nitrogen structures are often being compared to carbon-based structures that are not invertible. It can therefore be useful to enumerate the nitrogen structures in the database even though they are interconvertible in solution. For this reason, the flag here controls whether or not nitrogen atoms with potential stereochemistry are enumerated. [Default=true]

-omega

This flag controls whether the conformer generation using the **OMEGA** conformers is applied to the fragments. This flag has two important repercussions. First, by turning it off, the process can be stopped after the relatively quick fragmentation phase. This allows users to examine the fragments that are automatically written to an intermediate SMILES file before paying the high cost of generating conformers. Second, you can choose to replace the **OMEGA** algorithm with sampling conformers from known structures using the *-readConfs* flag.

-readConfs

This flag allows users to specify a 3D structure file from which the database generation algorithm will sample conformers. For each fragment generated in the cutting algorithm, all examples of the fragment's conformers that are present in the molecule file specified by this flag will be added as conformers and saved in the **BROOD** database. Identical conformers are recognized and duplicates are removed. If you have access to a large database of small-molecule crystal structures, this flag can be used to sample conformers from that database rather than using the **OMEGA** algorithm.

> **Note**
>
> The input molecules used to generate the fragments (from the *-in* or *-userFrags* flags) do not need to be the same as the input molecules specified with this flag.

-primaryFrag

This flag indicates that **CHOMP** should only process the smallest possible fragments with the given fragmentation pattern. It is primarily for fragmentation applications unrelated to **BROOD**.

Fragment selection parameters

-filter

This is a complex flag that specifies both whether the default filter should be applied to the fragments with the values (True and False). The *-filter* flag can also be used to specify a filter file describing a non-default filter to be used to filter the fragments. For additional information on filtering, please see the documentation for the filter product.

-minFrequency

During the breaking of input molecules into fragments, the number of source molecules from which a fragment can be extracted is tracked. This flag indicates the minimum number of source molecules that are required in order for the fragment to be retained. The frequency in this case is normalized as a percentile where 99 indicates the most common fragments and 0 indicates the least common fragments. By default, all fragments are retained. [Default=0]

-maxDegree

-minDegree

The *-minDegree* and *-maxDegree* flags specify the range of acceptable attachment points for fragments generated by **CHOMP**. The *-maxDegree* can not be more than 3 to produce a compatible fragment with the **BROOD** app. [Default=1,3]

-maxMolWt

This is a simple flag that allows the user to specify the maximum molecular weight of any fragment generated by **CHOMP**. [Default=350.0]

-maxHvy

-minHvy

These flags specify the range of heavy atoms allowed in fragments generated by the **CHOMP** algorithm. [Default=0,15]

-maxChiral

This flag allows you to specify the maximum number of chiral centers (both atom and bond centers) in a fragment. While there are many drugs and other useful molecules with a large number of stereocenters, they are often added for a specific purpose and should not be suggested lightly in a design setting. For this reason, fragments with many chiral centers may be undesirable in some databases. This flag allows the user to efficiently eliminate these fragments. [Default=3]

CHAPTER

FIVE

BROODDBMERGE

5.1 Overview

The **BROOD** distribution includes a utility, **BroodDBMerge**, that allows users to create a single database by merging two databases, such as BROOD's default database and their own corporate database. **BroodDBMerge** accomplishes this task using three required parameters: *-in1*, *-in2* and *-out*. This allows combining two databases in the directories specified by the two *in* options and writing a new database to the directory specified by the *-out* flag. By default, **BroodDBMerge** will not overwrite a database.

BROOD databases are complex structures with many precalculated parameters and indices organized into multiple binary files for fast access of multiple fragments in an efficient manner. Because of this complex structure, it is not possible to simply string two database files together in order to merge databases. Instead, **BroodDBMerge** reorganized the molecular fragments and recalculates the database indexes.

By default **BroodDBMerge** removes duplicate fragments from the new database (see *-removeDuplicates*). When **BroodDBMerge** identifies a unique fragment, it is written to the new database. If a second copy of a fragment is identified, it is discarded. The database passed in using the *-in1* parameter takes precedence in duplicate removal Duplicate fragments in the database specified by *-in2* will be discarded. In addition, if a fragment appears twice in either database, the later instance will be discarded. The user can use the database input order to control which duplicate fragments will be present in the final database.

5.2 Command Line Help

A description of the command line interface can be obtained by executing **BroodDBMerge** with the *–help* option.

```
> brooddbmerge --help
```

will generate the following output:

```
Help functions:
  brooddbmerge --help simple      : Get a list of simple parameters
  brooddbmerge --help all         : Get a complete list of parameters
  brooddbmerge --help defaults    : List the defaults for all parameters
  brooddbmerge --help <parameter> : Get detailed help on a parameter
  brooddbmerge --help html        : Create an html help file for this program
  brooddbmerge --help versions    : List the toolkits and versions used in the␣
↪application
```

5.3 Example Commands

This section has a brief series of examples of **BroodDBMerge** command-line executions. Each example is followed by a brief description of its behavior.

```
prompt> brooddbmerge -in1 dbA -in2 dbB -out dbAB
```

This is the most basic merge of the two databases dbA and dbB into the new database dbAB. By default, the new database dbAB will not contain any duplicate fragments, but will contain all of the unique fragments that were present in either dbA or dbB.

```
prompt> brooddbmerge -in1 dbA -in2 dbB -out dbB
```

In this execution, it is most likely the user accidentally used `dbB` for the output. Rather than allow this typo to accidentally erase or corrupt the `dbB` database, the execution will stop. This behavior can be overridden with the *-overwrite* flag.

```
prompt> brooddbmerge -in1 dbA -in2 dbB -out dbAB -removeDuplicates false
```

This execution of **BroodDBMerge** will not include duplicate removal. All of the fragments that are in `dbA` and `dbB` will occur in `dbAB` no matter how many times any fragment may appear in either or both databases. When **BROOD** searches the resulting database, each instance will be treated separately.

5.4 Required Parameters

-in1 (-i1)

This option is for loading the first **BROOD** database. The value passed with the flag should point at the directory of the database rather than any files inside the database.

-in2 (-i2)

This option is for loading the second **BROOD** database to be merged with the database specified by the *-in1* flag.

-out (-o)

This option indicates the new database to create based on merging the databases specified by the *-in1* and *-in2* flags. By default **BroodDBMerge** will require you to specify a new directory for the new database (see *-overwrite* for details).

5.5 Optional Parameters

-verbose (-v)

This flag controls the amount of output generated by the program. By default, all essential information is presented in a summary. When the flag is set, a large amount of detailed information is also written to stdout. [Default=false]

-overwrite (-over)

By default, **BroodDBMerge** will only write a new database (it will not overwrite another file or directory). Brood databases can be quite expensive to generate; to avoid accidental erasure of databases **BroodDBMerge** will not overwrite a previous database. If you *want* to overwrite a database, you can set the *-overwrite* flag. [Default=false]

-removeDuplicates (-duplicates)

When merging two databases, users can choose whether to remove duplicates. By default, any fragment that is present more than once in the combined databases will only be written one time. No fragment-level merging occurs; only the first instance of a fragment is written. [Default=true]

CHAPTER

SIX

OVERVIEW

The **BROOD** product contains four applications as well as a script that can be run in conjunction with OpenEye's molecular visualizer **VIDA**. Two of the programs, **vBROOD** and **BROOD**, and the **VIDA** script provide the ability to generate queries, search for similar fragments and visualize and triage the resulting analogs. The additional pair of applications, **CHOMP** and **BroodDBMerge**, are utilities that generate and merge **BROOD** fragment databases, respectively.

In these tutorials, we will primarily focus on generating and exploring analogs of an input molecule or molecules.

The basic workflow for **BROOD** is:

1. Build a query fragment from an input molecule.
2. Search for database fragments that are similar to the query fragment.
3. Examine the analog molecules built from putting the similar fragments into the initial molecule.

CHAPTER

SEVEN

HUMAN COMPLEMENT C3A RECEPTOR EXAMPLE

This example follows some of the work on Human Complement C3a Receptor (C3aR) from Reid and coworkers [Reid-2014]. C3aR is a G-protein coupled receptor important for human inflammatory processes and has been identified as a potential drug target for several inflammatory diseases. In their work, Reid replaced a bridging ether with a heterocycle to convert a known antagonist compound into a potent and selective agonist for Human C3aR. In this example, we will design analogs to the known compound. The results will include several compounds studied by Reid.

7.1 Specifying the Database

Open **vBROOD** by double-clicking the icon or typing **vBROOD** at a command prompt. The GUI will start, displaying the information screen and the five primary task buttons (*Build and Run*, *Build a new query*, *Run BROOD*, *Filtering*, and *View Results*).

- The first time you open **vBROOD**, you need to specify a default database. If you have not already done so, download the default database or build your own database and specify it.
  - For this example, be sure to specify the default **BROOD** database.
  - If you do not have the **BROOD** database available locally, you need to download it from the OpenEye Download Page (select your platform and then the product BROOD).
  - For help with installing a **BROOD** database, see *Database Installation*.
  - To specify a database, select *Edit Preferences* under the *Edit* menu and type the name of the database you want to use in the first line of the pop-up preferences dialog. Alternatively, click the *Browse* button to navigate to the database.
- If you have previously specified the database, you will not need to do this again unless you want to change the database.

Note

The **BROOD** database is a directory. After the default **BROOD** database has been downloaded, you need to unpack it and select the new directory. We do not recommend renaming the **BROOD** database, since you must then rename all the files contained within it. Consistency is required to prevent confusion between the name of the database and the files being searched under that name.

7.2 Loading a Molecule

Of the five primary tasks, the one most commonly used is *Build and Run*. This brings you to a five-step GUI wizard that takes you through building and editing a query, selecting an optional property filter, selecting parameters for the **BROOD** search, and visualizing the results.

1. Begin by clicking the *Build and Run* button, which launches the *Load Molecule* interface.
2. Load the molecule `c3a-4.sdf` by clicking on the small folder icon in the molecule entry space, then navigate to the **BROOD** directory of the **OpenEye-applications** data installation and load the file `c3a-4.sdf`.
    a. On the right, the c3a compound 4 is displayed in a 2D image and is highlighted to suggest several fragments for possible replacement.
    b. On the left, the current molecular property table is now filled to reflect the properties of the molecule. These calculated properties can be used as a guide to consider some of the things that might need to be changed in the loaded molecule.

Note

The c3a compound **4** has appropriate aqueous formal charges. **BROOD** retains the state of the molecule as it is in the input file, so your query molecule *must* be in the charge and tautomer state you desire before loading it into **BROOD**.

3. Multiple fragments can be selected by shift-clicking. In this case, though, we will simply lasso the fragment we want to select. Hold down the Left button on your mouse and circle the fragment of the molecule shown in the image below. The lassoed fragment will then be highlighted (Figure 1).

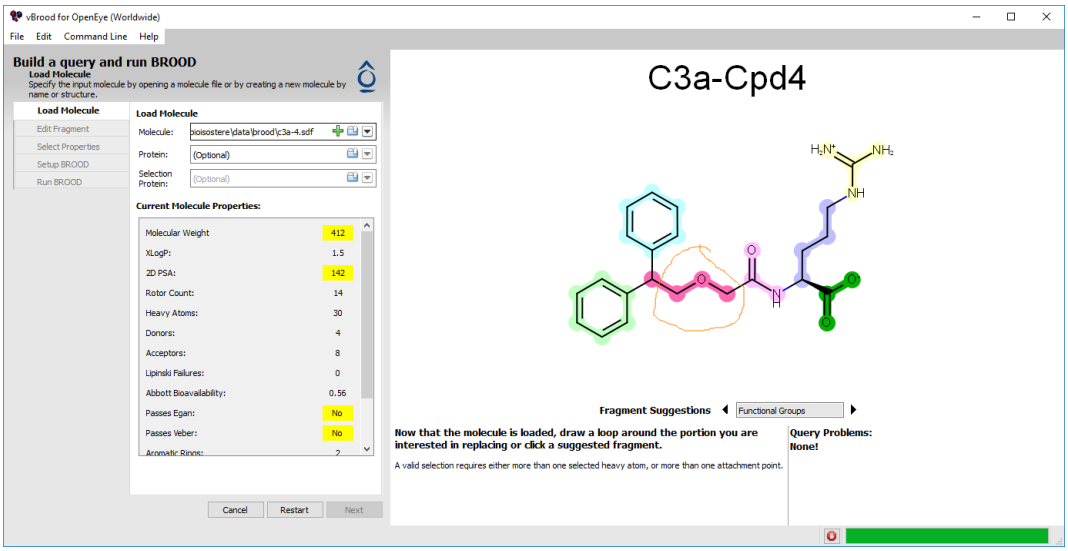


Fig. 1: **Figure 1.** Selecting a query fragment.

Note

When **BROOD** is run, the selected fragment will be replaced by fragments from the database with similar shape and chemistry.

4. After the fragment is selected, click the *Next* button to move on to query editing.

Note

In general, if the molecule you loaded in this step has 3D coordinates, **BROOD** will use them and will replace the

fragment in place. This allows fragment replacement to work in an active site. If the loaded molecule has only 2D coordinates, then a single low-energy conformer will be generated for the ligand using the **OMEGA** algorithm.

7.3 Editing the Query

As the wizard moves to the second stage, the fragment is extracted from the whole molecule and is displayed along with the color atoms generated from the input structure. This interface is designed to help users inspect and edit either the shape or color (i.e., chemistry) atoms of the query. Initially, the three heavy-atom, two attachment-point query will appear annotated with an acceptor color atom. In this example, Reid and coworkers discovered that this acceptor moiety is extremely important. Emphasis to the color force field at this point will be added as well as a constraint.

Note

For additional information about the theory of shape and color force fields, please see the Shape TK or ROCS Application theory sections.

The toolbar with query editing tools is located on the left-hand side of the 3D window. Hovering over each icon gives a brief description.

The first step is to increase the score for replacement fragments that match the constraint.

1. Select the *Color Editing* tool (second icon).
2. Click on the small arrow next to the icon and choose *Add Acceptor*. This prepares the *Selection* tool for adding color atoms.
3. Click on the current acceptor atom, thus adding a second color atom. The annotation on the 3D window should now read “acceptor x2” to reflect the change. This means that the color potential is doubly weighted in this position.

Second, a constraint must be added that requires all analogs identified by **BROOD** to include an acceptor near this position.

1. Select the *Constraint* tool (third icon).
2. Click on the small arrow next to the icon and choose *Add Acceptor Constraint*. This prepares the *Selection* tool to add a constraint.
3. Now click on the same acceptor atom we chose above. A dashed circle will appear around the atom, annotating the newly added constraint. The query should now look similar to Figure 2 below.

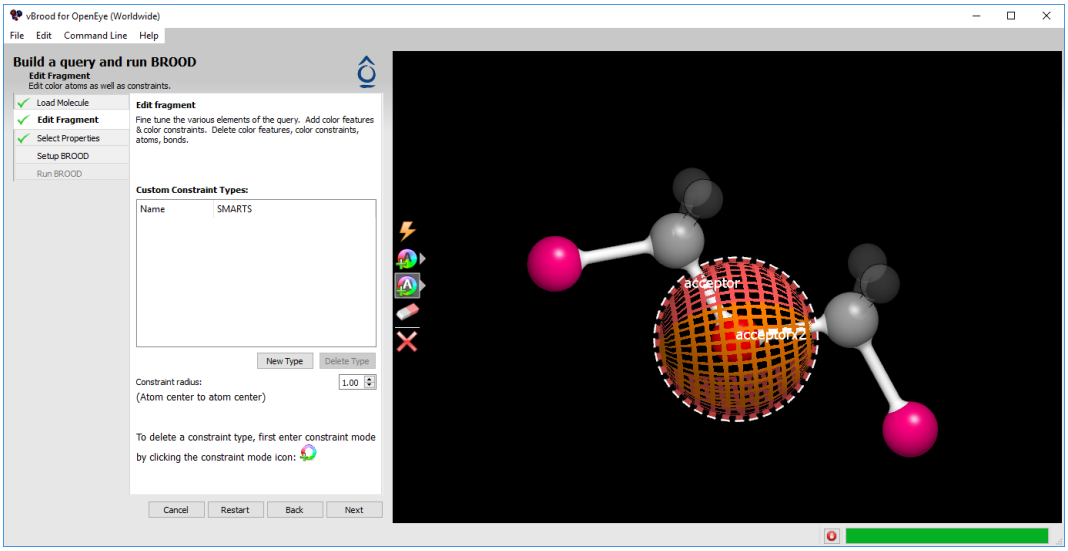


Fig. 2: **Figure 2.** Color and constraint on a query fragment.

Third, click *Next* to move to the next step in the wizard.

7.4 Selecting Properties

This interface allows **BROOD** to filter the output by the properties of constructed molecules with replacement fragments. For example, **BROOD** filters by simple rule-of-five, predicted bioavailability, and complexity. Any constructed molecules that fail to meet these property filters will not appear in the output. In all cases, the property filters consider the properties of the whole molecule (rather than only the fragment).

For this example, we will leave property filtering turned off. Click *Next* to move to the *Setup* stage in the wizard.

> **Note**
>
> There is a button for saving the query at this stage. In all cases, a copy of the query is written into the working directory when **BROOD** is run. This button is for saving another copy of the query or for saving a query when you don't want to follow through and set up a **BROOD** run. Saving at this point is for convenience only and not necessary.

7.4.1 Using the Property Filter (Optional)

At some point, you may want to use the property filter.

1. While still on the *Selecting Properties* step, check the *Property Filter* checkbox.
2. Choose a default *Drug-like* filter or a filter that was built earlier.

7.5 Setting Up the BROOD Search

The *Set Up BROOD* step has many options. Hovering over the options will reveal tooltips describing them. For this tutorial:

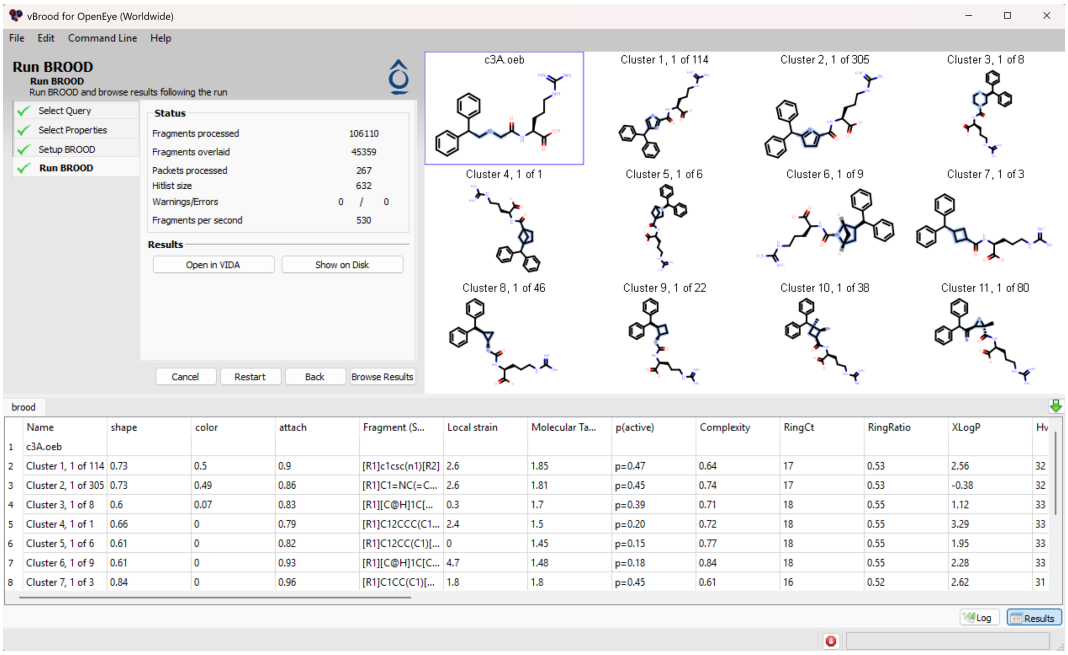
1. Select an *Output Prefix*. All output files will have this text pre-appended to their filenames. If the prefix has been used previously in the same directory, **BROOD** will automatically increment a trailing integer. If there is a danger of overwriting the previous output, the prefix will appear in red text.
2. Select an *Output Directory*. All output files will be written to this directory.
3. If you are running on a machine with more than two processors, click the multiprocessor box and set the number of processors.

> **Note**
>
> In general, **BROOD**'s speed scales by N-1, where N is the number of processors.

4. You should also verify that:
   a. The database you selected earlier in this tutorial appears in the *Database* setting.
   b. *Shape & Color* is selected under *Search Options*. This is the default, but if you changed it in a previous run, the GUI will remember that value.
   c. *Any Size Ring* is selected under *Ring Requirements*. This is the default, but if you changed it in a previous run, the GUI will remember that value. Setting the option to *Any Size Ring* requires that all substituted fragments have at least two ring atoms on the shortest path between the attachment points, thus providing a conformationally constrained replacement fragment.
   d. The annotated query fragment in the 3D window is the query you intend.
5. Click *Run* to start the search.

7.6 Opening and Viewing Results

As **BROOD** searches, the GUI writes log information to the window's lower portion and continuously updates the status information in the upper left. The lower right corner of the GUI displays a graphical progress bar. The search in this tutorial takes a few minutes to complete and results in 60 analog molecules in three clusters. At intermediate intervals, the current hit list of identified fragments is displayed in **vBROOD**'s 2D depiction area. A discrete progress bar in the lower right of the GUI also shows the progress. When the job is complete, the bottom half of the screen displays results in a spreadsheet while the upper-right portion gives a preliminary view of the 2D analogs (Figure 3 below).



	Name	shape	color	attach	Fragment (S...	Local strain	Molecular Ta...	p(active)	Complexity	RingCt	RingRatio	XLogP	Hv
1	c3A.oeb												
2	Cluster 1, 1 of 114	0.73	0.5	0.9	[R1]c1csc(n1)[R2]	2.6	1.85	p=0.47	0.64	17	0.53	2.56	32
3	Cluster 2, 1 of 305	0.73	0.49	0.86	[R1]C1=NC(=C...	2.6	1.81	p=0.45	0.74	17	0.53	-0.38	32
4	Cluster 3, 1 of 8	0.6	0.07	0.83	[R1][C@H]1C[...	0.3	1.7	p=0.39	0.71	18	0.55	1.12	33
5	Cluster 4, 1 of 1	0.66	0	0.79	[R1]C12CCC(C1...	2.4	1.5	p=0.20	0.72	18	0.55	3.29	33
6	Cluster 5, 1 of 6	0.61	0	0.82	[R1]C12CC(C1)[...	0	1.45	p=0.15	0.77	18	0.55	1.95	33
7	Cluster 6, 1 of 9	0.61	0	0.93	[R1][C@H]1C[C...	4.7	1.48	p=0.18	0.84	18	0.55	2.28	33
8	Cluster 7, 1 of 3	0.84	0	0.96	[R1]C1CC(C1)[...	1.8	1.8	p=0.45	0.61	16	0.52	2.62	31

Fig. 3: **Figure 3.** Search results.

Once the run is complete, **vBROOD** automatically passes the search results to **VIDA** with a script that allows convenient viewing of the **BROOD** results. This **VIDA** interface is simplified and specialized for viewing **BROOD** results.

Click *Open in VIDA* for further exploration and triage of the results.

7.7 Exploring the Results

BROOD results are organized into clusters; the interface is designed to help users explore these clusters. Along the way, you can identify analog molecules of particular interest and save them to a *Favorites* list that can be saved as a separate hit list. This portion of the tutorial explains how to navigate through the clusters, add molecules to your *Favorites* list, and save the *Favorites* list as a new molecule file.

7.7.1 Keyboard Shortcuts for Navigating BROOD Results

When you first open the **BROOD** *Results Viewer*, the keyboard shortcuts are immediately displayed (Figure 4). For this tutorial, we will use the Left, Right, Up, and Down Arrow keys as well as the Q, F, and Delete keys.

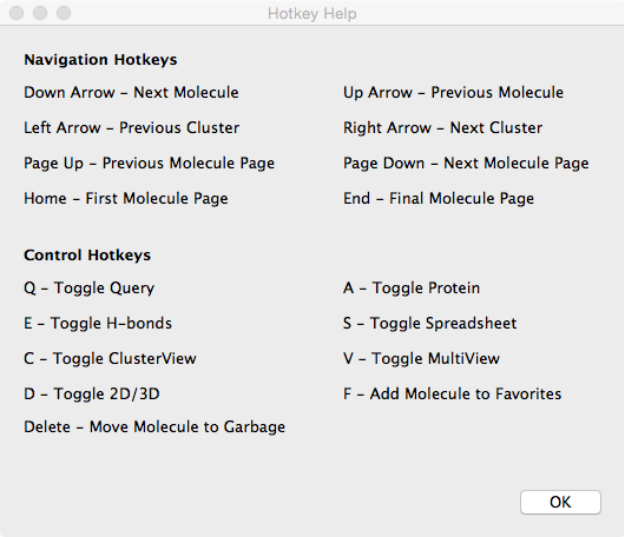


Fig. 4: **Figure 4.** Keyboard shortcuts.

Dismiss the keyboard shortcuts window. It can be redisplayed by clicking the *Hotkey Help* button on the left.

If this is your first time using the **BROOD** *Results Viewer*, you will see the results of the search in four 3D panes as well as a navigation and information pane on the left. The upper-left 3D Viewer pane always contains the query molecule with the replaced fragment highlighted in green and the constant part shaded in deep maroon.

The remaining panes each contain a cluster head of the three clusters identified in this query. By default, the new analog molecules are overlaid with a wireframe version of the query.

- Rotate and zoom the panes to examine some of the analogs. The panes will automatically be centered on the replaced fragment.
- Use the Q hotkey to toggle the wireframe query overlay on and off.

7.7.2 BROOD Results in the Results Viewer

The results of a **BROOD** search are automatically organized into clusters based on the similarity of their molecular framework. Particular attention is given to rings and aromaticity while different chemistries are allowed to be grouped. The Arrow keys can be used to explore the results for this search. Clusters found by **BROOD** are not hierarchical; instead, they are a series of collections, each represented by the compound from the cluster that receives the best **BROOD** score (Figure 5).

A look at the information pane of the **BROOD** *Results Viewer* shows the total number of clusters (three in the case of this tutorial). In addition, two tumblers identify the molecules that are currently visible. The left tumbler represents the cluster being viewed. When the blue highlight is on the left tumbler, the 3D panes are showing the cluster representatives. The Up and Down Arrow keys toggle through the cluster heads, with the selected cluster head highlighted in blue; and the matching blue square around the selected cluster representative in the 3D window will also change. If there are more than three clusters, the Up and Down Arrow keys will also handle scrolling through the all the cluster representatives.

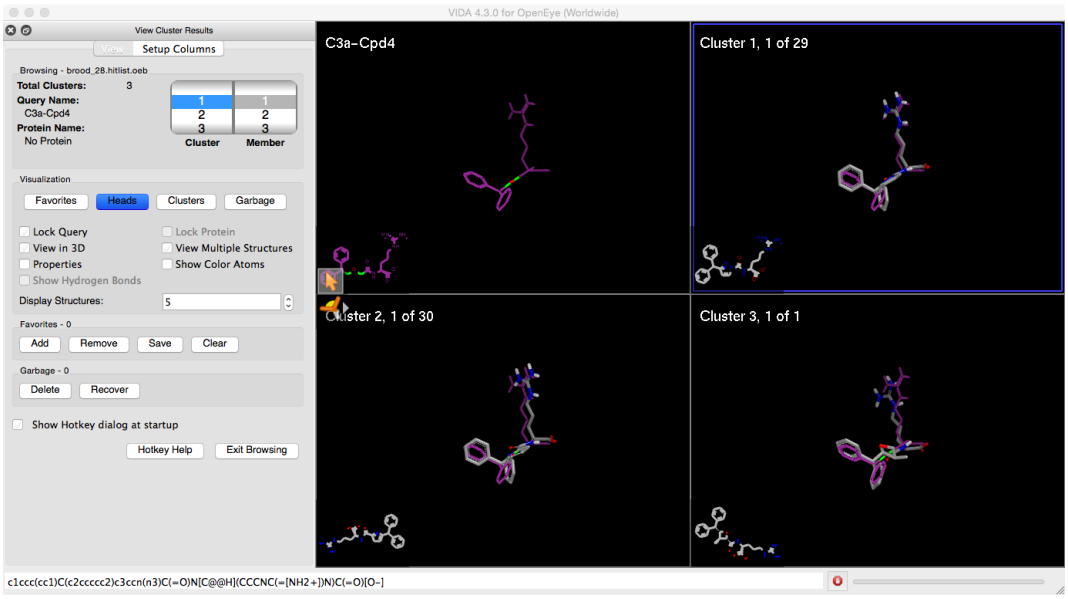


Fig. 5: **Figure 5.** Exploring clusters in the BROOD results.

1. Click the Down Arrow key. Note that the left tumbler changes from highlighting "1" to highlighting "2" and that the blue highlight box in the 3D window moves from highlighting the first cluster representative to highlighting the second representative. If you want to see all the members of a cluster, you can drill down into a cluster to see all of its members. In this section, we will explore navigation through Cluster 2 using the Arrow keys.
2. Use the Up and Down Arrows to select the representative molecule of Cluster 2. Note that it is labeled with "Cluster 2, 1 of 30," indicating that it is the first molecule in a cluster with 30 total members.
3. Press the Right Arrow key to expand Cluster 2. Notice that in the top-left pane, the blue highlight has moved onto the right tumbler, indicating that you are now viewing cluster members. It is highlighting "1," indicating that the highlighted molecule in the 3D window (the one surrounded by the blue box) is the first molecule in the cluster.
4. Use the Up and Down Arrows to move through the cluster members. By default, six panes will be visible and the Arrow keys will cause the molecules to scroll through the panes one by one. Look through some of the molecules in Cluster 2 using the Arrow keys. To toggle the number of molecules displayed, use the V hotkey.

> **Note**
>
> The Page Up and Page Down keys can be used for more efficient navigation. If multiple panes are visible, Page Up and Page Down will change all the visible molecules at one time. This navigation is analogous to the Up and Down Arrow keys, but more efficient.

5. Use the Page Up and Page Down keys to move through the cluster members.
6. The Home and End keys will navigate to the first page or the last page of the cluster of molecules you are currently examining. Use the Home and End keys to navigate to the beginning and end of Cluster 2.

The Up Arrow, Down Arrow, Page Up, Page Down, Home, and End keys are all available for navigation throughout all the cluster representatives in the same way they were for navigating the molecules inside Cluster 2.

7.7.3 Working with Favorites in the BROOD Results Viewer

When examining **BROOD** hit lists, some users like to select particular molecules for future sharing or examination by saving them into a separate culled list. To facilitate this, the **BROOD** *Results Viewer* includes *Favorites* and *Trash* features. While navigating the analogs as discussed in the previous section, you can choose any molecule and move it to your Favorites. In addition, if you identify molecules you would prefer not to examine again, you can move them to the *Trash*. Both the *Favorites* and the *Trash* can be opened if you inadvertently placed a molecule in the wrong category.

1. Using the navigation skills developed previously, choose a molecule in Cluster 2. To save it to your *Favorites* list, first select it by navigating until the blue box is highlighting the molecule (this can also be done by clicking on the molecule). Once highlighted, you can add it to your *Favorites* list either by using the F hotkey or by clicking the *Add* button under the *Favorites* menu. Select several more molecules and add them to your *Favorites* list.

2. To examine your current *Favorites* list, click the *Favorites* button in the navigation pane. The 3D window will then display the molecules you've added to your *Favorites* list. Navigate to your least favorite molecule in the *Favorites* list and press the Remove button. This action removes the molecule from the *Favorites* list and puts it back into its original position in the *Results* cluster.

3. When your list of favorite molecules is complete, you can save it to share with colleagues or for later examination. Click the *Save* button under the *Favorites* pane and a *File-Save* dialog will open. The default file format will be OpenEye Binary, although a variety of other file formats are also available. The query molecule will be saved in your *Favorites* list. If you have any proteins, color annotations, or property annotations, they will also be saved in your hit list. The data saved in your hit list is only limited by the file format you select. While OEB is preferred, SDF can also save much of the information.

In addition to keeping track of your favorite analogs, it can also be useful to get rid of some of the less desirable molecules. For this purpose, there is a garbage can that allows you to place molecules in the *Trash*.

1. Using the navigation skills developed previously in this tutorial, navigate to a molecule in Cluster 1 that is not useful. When the blue highlight box is around the molecule, hit the Delete key. The molecule will be removed from the list and the remaining molecules will be renumbered. For example, if you highlight and delete cluster member 15 (labeled "15 of 54"), it will be moved to the *Trash* and the molecules formerly labeled "16 of 54" to "54 of 54", will now be relabeled "15 of 53" through "53 of 53".

2. If you inadvertently put a desirable molecule in the *Trash*, you can open the trash can using the *Trash* button in the *Visualization* pane of the *Results Viewer*. When you click this button, the 3D view will change to display the molecules in the trash. These molecules can be navigated as described above, and can be returned to their original place in **BROOD**'s *Results* clusters by highlighting them and then clicking the *Recover* button under the *Trash* pane on the left panel.

CHAPTER

EIGHT

FRAGMENT JOINING TUTORIAL

In the fragment-based design setting, users may occasionally want to link two fragments with crystal orientations in the active site into a single molecule. Ideally, when the two fragments are linked without additional entropy, the binding energies can simply be added. In practice, this kind of additivity is often not possible for a number of reasons, including ligand strain, protein strain, entropy, and desolvation. Nevertheless, **BROOD** can be quite useful in this setting to suggest potential linker groups that might span the fragment–fragment gap in a low energy manner. This tutorial walks you through the steps for identifying linker fragments.

This tutorial follows the work of Ward et al. [Ward-2012], in which they designed linkers to join two crystallographic fragments to generate a lead.

8.1 BROOD Database

If you have not yet installed the **BROOD** database, please see the *Database Installation* section for further information before continuing this tutorial.

8.2 Loading Two Molecules

For this exercise, we will take two fragments positioned in the same active site and join them with an unstrained linker.

1. Open vBROOD and select *Build and Run* from the primary window.
2. Click on the small down arrow in the *Molecule* input box.
3. From the pop-up menu, choose *Open Fragments for Bridging*. This will open a two file dialog.
4. Choose two separate fragment files using the appropriate function for your operating system. For this tutorial, select the files `ward-22-21.oeb` and `ward-21-22.oeb` from the `brood` directory in Example and Tutorial Data.

When the files are properly loaded, the 2D window should display them as shown in Figure 6 below.

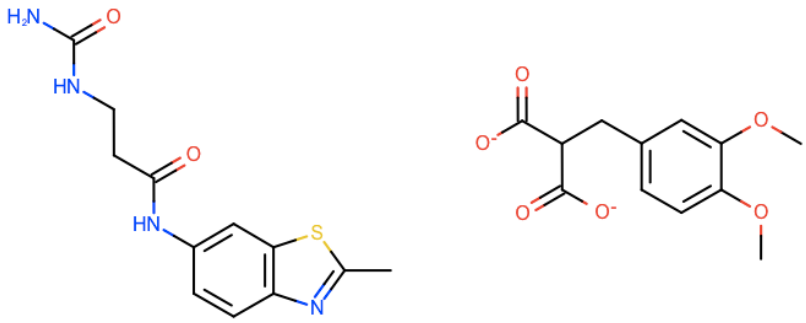


Fig. 1: **Figure 6.** Two molecules loaded for query selection.

8.3 Loading the Protein

In any **BROOD** run, whether fragment replacement or fragment joining, a *protein context* can be added. When the query molecule or molecules are from a protein active site and the protein is passed to **BROOD**, the new analogs will be constructed in the context of the protein active site. Before the final hit list is generated, all the new analogs will be tested for clashes against the protein.

1. After the two fragments are loaded, locate the *Protein* input box.
2. Click the folder icon to open an input dialog.
3. Select the file `ward-protein-join.oedu`. This will allow you to include the protein in the search and build joined fragment molecules in the active site of the protein.

8.4 Choosing a Query

When the two files are loaded, depictions of two separate molecules will appear in the 2D query window. As with a single query, the fragments for replacement can be lassoed, one from each molecule in this case (Figure 7).

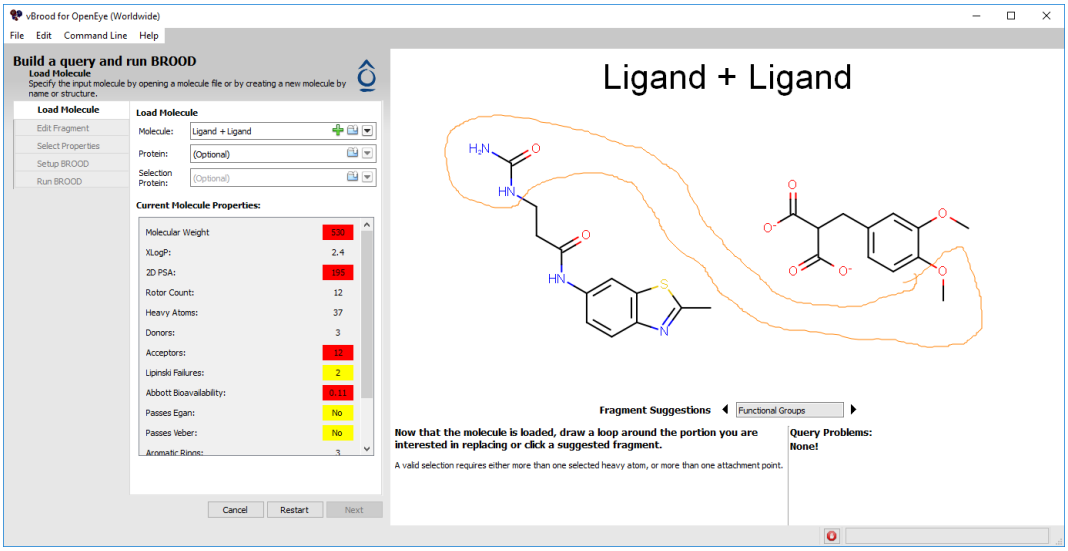


Fig. 2: **Figure 7.** Choosing a query to join two molecules.

1. Use your mouse to lasso the parahydroxyl and the terminal urea in a single circle. It doesn't matter that the fragments are not close to each other.
2. Click *Next* to move to the 3D window.
3. Since you loaded a protein in the previous query selection stage, there is now a *Show Protein* button beneath the window that allows you to toggle the visibility of the protein on and off. Try this now. Having no visible protein makes it much easier to work with the query fragments. On the other hand, being able to see the protein and how the current ligands interact with it can be quite helpful in designing a query. For this exercise, turn off protein visibility.
4. Select the *Eraser* icon and delete all the color atoms by clicking them. This will allow **BROOD** to bridge the two fragments independent of the influence of the terminal urea chemistry.
5. Now delete all the heavy atoms except for the pink attachment atoms and the oxygen and nitrogen atoms bonded to the attachment atoms. This will allow **BROOD** to bridge the two fragments independent of the influence of the shape of the terminal urea. In the end, you will have a minimal query required **BROOD** to carry out an unguided search (Figure 8).

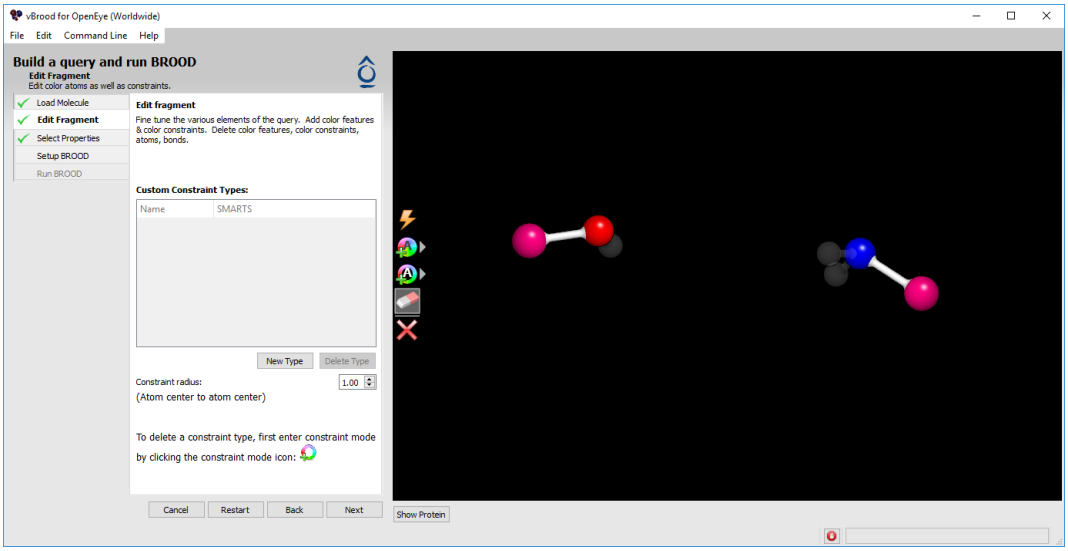


Fig. 3: **Figure 8.** Minimal query for molecule joining.

8.5 Setting up a Search

1. Click *Next* to move to the filter pane, then click *Next* again to move to the **BROOD** *Run Setup* window.
2. If you have a multiprocessor machine, click the *Multiprocessor* check box under *General Options* and select the maximum number of processors.

> **Note**
>
> **BROOD** can use from 1 to N+1 processors, where N is the number of logical cores detected on the machine. On many desktop systems, the OS remains responsive even with all the processors engaged by **BROOD**. Users should explore performance on their own systems.

3. Under *Search Options*, select the *Link Geometry* search type. This will carry out a CAVEAT-like search [Lauri-1994] that attempts to span the empty region between the two fragments.
4. Try this example with the *Quick Search* option (see below). Later you can also run without the *Quick Search* option to compare the calculation time and quality of results.
5. Next, consider ring selection in the search. By default, **BROOD** requires a ring in the atom path connecting every two attachment hit. For this exercise, under the *Ring Requirements* section, select the "Zero Rings" option. This will require that fewer than two ring atoms are located on the shortest atom-path connecting the fragments in each **BROOD** hit.
6. Finally, when a protein is included in a search (as in this case), all solutions are checked for bumps with the protein using a 2.25 Å cutoff by default. For this example, adjust the cutoff to 1.7 Å. To do this, click the *Advanced Search* tab and adjust the number labeled "Protein-ligand bump radius" to 1.7.
7. Now click *Run*. The view will switch to the *Results Tracking* view while the job is running. When the run is complete, the results will be viewable in the **vBROOD** GUI and in the **BROOD** *Results Viewer*. Navigation and exploration of those results work similarly to those in the first tutorial. If you are not familiar with results navigation, it may be useful to review the first tutorial. Here, special attention will be paid to features that are different in this tutorial.

8.6 Examining the Log Files

When the **BROOD** run is complete, click the *Log* button on the lower right of the **vBROOD** *Results* window. You can read the logging information to learn details about the number of molecules that were processed, how many were overlaid, how large the initial hit list was, and the reason that any molecules were eliminated from the hit list.

When running a standard search, the *Results* page shows every fragment that was overlaid and the hit list of those that passed all requirements. For this example, about 45,000 fragments were overlaid and a hit list of 300 was retained.

> **Note**
>
> The *Quick Search* function's goal is to complete its search in approximately two minutes. To accomplish this, the hit list is truncated up to the first 50 cluster representatives. Further, each cluster is represented only by its best scoring molecule. The *Quick Search* results are often interesting and contain a few attractive ideas. *Quick Search* allows users to assess whether a more time-consuming search is likely to be fruitful.

8.7 Viewing Results in a Protein Site

Now examine the results in 3D by clicking the *Open in VIDA* button. As in the previous example, this opens the **BROOD** *Results Viewing* script and automatically loads the results with the modified interface. As before, the results are organized by cluster and displayed in the context of the query. When you examine the *Quick Search* results, each cluster will have at most one member, so navigation is simple.

- Because we loaded a protein as part of the query, the protein structure is also available in the *Results Viewer*. Using the A hotkey or the *Lock Protein* check box in the visualization panel, the protein display can be turned on and off. When the protein is turned on, the protein residues near the query molecule are displayed.
- When the protein is visible, hydrogen bonds between the protein and the ligand annotations can be added and removed using the E hotkey or the *Show Hydrogen Bonds* check box. This works best if the query is not locked (use the Q hotkey or the *Lock Query* check box).
- When the protein is visible, a partial surface covering the area of the active site can be turned on and off using the R hotkey. The surface is colored with the electrostatic potential (Figure 9).

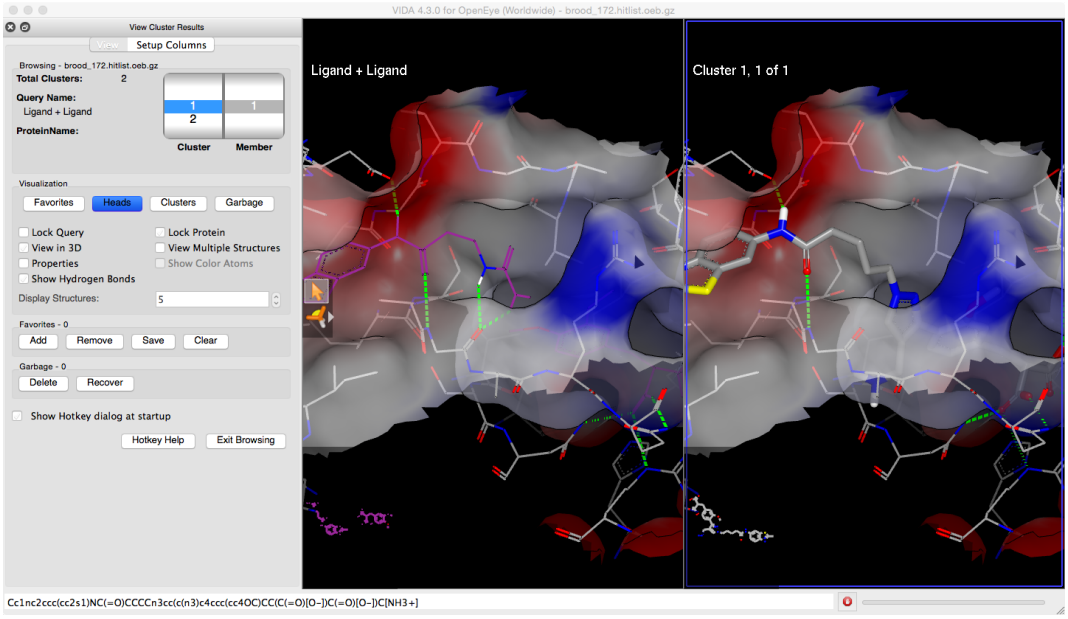


Fig. 4: **Figure 9.** BROOD results in an active site with surface and interactions.

The other features of the *Results Viewer* can be reviewed in the first tutorial.

CHAPTER

NINE

GENERATING A DATABASE WITH CRYSTALLOGRAPHIC CONFORMATIONS

This example demonstrates how users can create a **BROOD** database with fragment conformations pulled from small molecule crystal structures rather than conformations generated by **OMEGA**. For this task, we will be using **CHOMP**, **BROOD**'s application for fragment database generation.

9.1 Preparing Data Sources

For this exercise, **CHOMP** requires two molecular inputs. The first input will be the molecules you want **CHOMP** to use to generate molecular fragments for the **BROOD** database. The second input will be the 3D molecular structures **CHOMP** should use to extract example conformers for the fragments generated from the first input. The same file can be used for both inputs; for instance, if your company has a large collection of small molecule crystal structures, you can use it to both generate the molecular fragments for the **BROOD** database and to define the allowable conformations of those fragments.

The molecular file for fragmentation can be in any supported 1D, 2D, or 3D molecular file format (including SMILES or 2D SDF). If the molecules include titles, these can be used in the **BROOD** *Results* to track the most common sources of each fragment.

The molecular file for crystal conformer specification must be a 3D structure file. We recommend using SDF, MOL2, or OEB. These formats can be exported from common crystallographic database tools (refer to their manuals for details of the process). In our experience, the PDB format has too many ambiguities for reliable handling of 3D fragment conformations; we recommend avoiding the use of PDB files here if possible.

For this tutorial, example files have been included in the **BROOD** installation. For the first input, you will use the file `biotin.ism`; for the second input, you will use the structural file `biotin.xtal.oeb`.

> **Note**
>
> Using the same input for fragment generation and conformer specification assures that there will be a crystallographic conformation for every fragment generated by **CHOMP**. If different files are used, then fragments may be produced that have no representative conformation in the structural data.

9.2 Generating a Database

CHOMP generates **BROOD** databases in two phases. In the first phase, a unique set of fragments is extracted from the molecular input. In the second phase, conformers are generated for each fragment, the fragments are organized to optimize **BROOD** search times, and then the fragments are written into a database. For this exercise, each fragment from the first phase will be searched for in the 3D molecules. If the fragment is identified as a substructure of any 3D molecule, the coordinates will be extracted and added as a new fragment conformer. Finally, duplicate conformers

of each fragment will be removed before the fragment and its 3D conformations are passed on for organization and database writing.

> **Note**
>
> **CHOMP** is a command line only program. It is part of the **BROOD** product and is installed as part of the **BROOD** installation process. **CHOMP** is not available from within the **BROOD** GUI in this release but is available in form on all supported platforms.

1. Click on the command line menu and select **CHOMP** from the drop-down menu.
    a. At the prompt, type and execute the command line `chomp --help all`.
    b. This should generate a list of all the command line parameters and will test whether **CHOMP** is available in your path.
    c. Please check the first few lines generated and make sure you are running the current version of **CHOMP**.
2. Change directory to the `brood` directory under the Example and Tutorial Data of your installation. It should contain the files `biotin.ism` and `biotin.xtal.oeb`. You can also copy these files to another directory if you would prefer.
3. Execute command line: `prompt> chomp -in biotin.ism -out crystal -readconfs biotin.xtal.oeb`.
    a. The output is a directory named `crystal` with 45 paired files names `crystal.N.idx` and `crystal.N.db`, where N varies from 0 to 44. It also contains two metadata files, `crystal.all.idx` and `crystal.info`.
    b. Because this is a very small database, many of these files are 0 bytes in size. Only segments 0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 15, 16, 17, and 18 are used, and even these files are quite small. Use the appropriate file system listing mechanism on your platform to confirm that these files contain information.
4. The database is now ready for searching. There are two ways to select it for searching within **vBROOD**.
    a. To use the newly created database as the default database:
        1. Start vBROOD and select `Edit->Edit Preferences`.
        2. A preferences dialog will open; the first item will be the database directory.
        3. Either hand edit or use the *Browse* button to locate and select your newly created database.
    b. To use the newly created database without changing the default:
        1. Start vBROOD and select the *Build and Run* button.
        2. Open the crystallographic query file `4ggz.oeb` and select the central aliphatic chain connecting the acid to the ring.
        3. Click *Next* to move to the *Edit Fragment* tab.
        4. Click *Next* to move to the *Select Properties* tab.
        5. Click *Next* a third time to move to the *Setup BROOD* tab.
        6. Now, the *critical step* is in the second entry, labeled *Database*. Use the *Folder* icon to select the crystal database you just created with **CHOMP**. Selecting the database this way will not change your default database in **vBROOD**.
        7. Finally, change the *Ring Requirements* parameter to the *No Requirements* selection and click *Run BROOD*.

c. **BROOD** will quickly execute and return a single result that is nearly identical to the input structure. The inclusion of the unusual torsion in the distal aliphatic chain just before the carboxylate moiety is an indication that the fragment is in fact using crystallographic conformations.

9.3 Advanced Topic: Editing the Fragments at an Intermediate Step

There are many fragment generation and fragment filtering parameters that control the number and type of fragments generated from your input molecules by **CHOMP**. The defaults for these parameters mimic our best practices for the general case. If you are generating a specific database for a special case, you may want to examine the fragments generated by **CHOMP** to determine in detail if you want to modify the default parameters. This section of the tutorial explains how to make this part of the database preparation workflow.

The following steps will quickly lead you to an intermediate *Results* file containing all the fragments generated for your database. Since this phase is typically significantly faster than full database generation, you can repeat it, adjusting the parameters until you are satisfied. Alternatively, you can hand edit the intermediate file using your own criteria and then pass it into the final 3D database generation phase when you are satisfied with the results.

- Consider the *command line used above* but without the 3D molecule file with the `-readConfs` flag. Without the `-readConfs` flag, by default, OMEGA conformations will be used. Turn off this option by passing False to the `-omega` parameter.

Whenever **CHOMP** is executed (regardless of 3D options), an intermediate file is generated that contains all the fragments generated from the input molecules. With both of these options for conformations turned off, **CHOMP** will generate the fragment file and then stop.

1. In the command line described above, use the output designation crystal with the `-out` flag. The intermediate file will be called `crystal.ism`.

   a. `prompt> chomp -in biotin.ism -omega false -out crystal`

   b. The output file `crystal.ism` includes 18 fragments numbered one to 18.

2. This file contains the output fragments along with a complex molecular title. The title includes information about the source molecules for the fragment and can be disregarded for this exercise.

3. Load this fragment file into your preferred molecular viewer, noting that it may contain a very large number of fragments. By examining the fragments, you can determine if they fit your preferences.

4. Rerun the command line, using the flag `-maxHvy` 8 and change the output to avoid overwriting the previous output (`-out` crystal2). Now you can compare the output of the `crystal.ism` and `crystal2.ism` files to see the effects of this parameter change.

   a. `prompt> chomp -in biotin.ism -omega false -out crystal2 -maxHvy 8`

   b. Examining the output file `crystal2.ism` reveals that only 13 fragments are present: the five that are larger than eight heavy atoms have been removed.

In addition to modifying the *Fragment Selection* parameters, as with `-maxHvy` above, the `-smarts` parameter can have a powerful effect on the fragments that are generated by **CHOMP**. This can be an effective tool regardless of whether the 3D conformers for your database are being extracted from crystallographic conformations or generated by **OMEGA**.

CHAPTER

TEN

THEORY

10.1 Theory

There is a long history to fragment and bioisosteric replacement (see [Chen-2003]). Most medicinal chemists are well versed in standard sets of bioisosteric fragments. Likewise, there is a long history of computational approaches to fragment replacement (see [Verloop-1987] and [Lauri-1994]). There have been several attempts to examine sets of known active compounds to empirically identify bioisosteric fragments (see [Ujváry-2003] and [Sheridan-2002]). While this is an interesting exercise, it has two drawbacks. First, it can only identify bioisosteric fragment pairs that are already known. While these provide interesting study, they are often already familiar to experienced medicinal chemists and modelers. Second, it identifies many incidental rather than meaningful fragment pairs. These result from the fact that simply because two molecules bind to the same site does not mean they differ only by bioisosteric replacement. For instance, chemists may analog a compound by substituting an N-methyl group with an N-benzyl group in order to identify new binding pockets. However, just because both of these compounds are bioactive does not mean that methyl and benzyl are similar fragments (though they would be identified as such by some methods). While one may apply various heuristics, such as size, to avoid this problem, we hope to explore methods that are more robust.

An alternative approach has been to use an algorithm that would predict whether two fragments are similar in relevant ways. Several groups including Bartlett, Verloop, and Willett (see [Lauri-1994], [Verloop-1987], and [Watson-2001]) have developed methods in this area. Here we seek to capitalize on and extend the ideas developed by these workers.

10.1.1 3D Fragment similarity searching

BROOD allows users to enter a single query fragment and search a very large database of known molecular fragments in order to identify fragments that are similar. Each database fragment is compared to the query fragment in 3D with regard to shape, chemistry, electrostatics, and geometric presentation of attachment vectors. The fragments that are most similar to the query fragment will appear in a hitlist.

All similar fragments in a **BROOD** hitlist are organized into clusters. The clusters are organized so that molecules with the same ring structures and core framework (reduced-graph) are placed in the same cluster. The first cluster in **BROOD** hitlists is always the molecules that are similar to the query, sharing the same core atomic framework. While some of these analogs may be obvious, they often include alternative interesting chemistries. The remaining clusters are each organized around a unique core atomic framework. Each cluster is represented by its best scoring member and the clusters themselves are ranked by the score of their best member.

10.1.2 Clash detection and selectivity

BROOD allows users to specify protein structures for the purpose of testing whether newly constructed analogs fit into the active site. When the **BROOD** query ligand is based upon a crystallographic co-crystal structure, **BROOD** builds the newly created analog molecules in the same shape and orientation as the query. If the crystallographic ligand was originally in an active site and the protein is passed to **BROOD** as the bump protein, the new analogs will be built in poses that are also in the bump protein's active site. When a bump protein is passed, **BROOD** checks for clashes between the bump protein and each analog. By default, if any ligand heavy atom is less than 2.25 Angstroms from a protein heavy atom, the analog is removed from the hitlist.

Users can also specify another protein for selectivity testing (referred to here as the selectivity protein). In order for an analog to remain in the final hitlist, it must clash with the selectivity protein. The bump protein and the selectivity protein should be aligned and the **BROOD** query should be based on a molecule that fits within the bump protein active site. In this case, analogs in the final **BROOD** hitlist will also fit into the bump protein's active site, while they will also have clashes with the selectivity protein. This combination is a simple model for analogs that have a chance to be active against the bump protein, but have a very low chance of being active against the selectivity protein. This model assumes the analog molecules bind to the bump protein in a manner that is similar to the original ligand and that the most favorable pose for the ligand class in the selectivity protein is similar to that in the bump protein.

10.1.3 Strain energy

BROOD's output includes newly constructed analog molecules that are intended to have a similar 3D shape to the query molecule. These new analogs are constructed partially from the original molecule and partially from new fragments. When these new molecules are generated and built into a conformation that has good shape and chemistry overlap with the query molecules, some strain may be introduced. To produce high-quality results, it is essential that the analogs are optimized while maintaining the query shape and that little strain is introduced in the process.

The **BROOD** search process guarantees that each of the molecular fragments alone is in a low energy state. After the fragments are joined, this may no longer be true. For every **BROOD** analog in the final hitlist, two optimizations are carried out to determine the local strain introduced by maintaining a shape similar to the original query. In the first optimization, the ligand is allowed to relax into a local minimum. In the second optimization, the ligand atoms are only allowed to move a fraction of an Angstrom, keeping the same overall shape of the molecule. In both calculations, the OEMMFF [Halgren-I-1996], [Halgren-II-1996], [Halgren-III-1996], [Halgren-IV-1996], [Halgren-V-1996], [Halgren-VI-1999], [Halgren-VII-1999] potential is used with a Sheffield solvation function [Grant-2007]. The local strain energy is the difference in ligand energy between the two calculations. By default, the maximum strain for any successful **BROOD** analog is limited to 6.5 kCal/M.

10.1.4 Fragment joining

One approach to lead identification and development is based on the identification and expansion of physically very small molecule inhibitors that are commonly termed "fragments" (see [Hajduk-2007]). Fragments in this sense are molecules with few atoms and should not be confused with the term fragment used elsewhere in this document that refers to part of a molecule. Nevertheless, the fragment replacement algorithm in **BROOD** can be useful in the modeling of fragment-based design. In fragment-based design, one strategy is to combine two non-overlapping inhibitory fragments to form a single molecule. While this is sometimes empirically done with a series of flexible linear linkers, the linkers can also be modeled.

10.1.5 Cyclization

One of the first applications of **BROOD** many users want to explore is the replacement of a flexible portion of a molecule with a more rigid fragment that fills the same space. **BROOD** excels at this application. This type of exercise can be considered a local cyclization. In some molecules, rather than a local cyclization, some chemists prefer to design a bridge between two portions of a molecule that do not have a local link. This is another task where **BROOD** can be quite useful. Long-distance cyclization, like fragment joining, is about finding a chemical fragment that can bridge to moieties given a particular 3D orientation.

10.1.6 Molecular complexity

In the early 1980's, Bertz first published a measure of the complexity of molecules and asserted that his calculated complexity could be related to synthetic ease ([Bertz-1981], [Bertz-1982]). Bertz built complexity terms that are similar to a Shannon entropy ([Shannon-1949]) but with regard to the elements in a molecule and the diversity of small fragments that make up the structure of the molecule. While the actual synthetic accessibility can be heavily influenced by the availability of complex synthetic building blocks and advances in stereo synthetic methods, molecular complexity remains a useful tool for prioritizing which compounds chemists should look at first when primary modeling methods don't readily distinguish between them.

In 2007, Boda and coworkers extended Bertz's idea and compared it to experimental chemists' predictions of synthetic accessibility ([Boda-2007]).

Boda made the significant advance of adding stereo complexity to Bertz's elemental, graph, and ring complexity. We noted in the paper that nearly all the signal was generated by the molecular complexity and stereo complexity. In fact, each of these measures, without the linear fitting presented in the paper, correlated with chemists' predictions of synthetic accessibility as well as the chemists' predictions correlated with one another. Thus, in **BROOD**, we have implemented a molecular complexity that is a normalized sum of the graph and size complexity, elemental complexity, and stereo complexity. The normalized complexity score starts at zero for the simplest, smallest molecule and grows to values that generally don't exceed 1.0 for medicinally relevant small molecules.

BROOD uses molecular complexity to sort analog molecules in the final hitlist that have very similar shape and color scores.

10.1.7 Aromatic fragment comparison

This is an example of applying **BROOD** as a technology to explore fragment similarity outside the direct context of a specific drug-discovery project. Tu and coworkers at Pfizer explored the chemical space of aromatic ring systems ([Tu-2012]). In their work, they used **BROOD** at the core of the NEAT (Novel and Electronically Equivalent Aromatic Templates) tool. This tool uses a combination of high-level QM-derived partial charges along with **BROOD**'s electrostatic similarity calculation to explore potentially aromatic system replacements. It allows medicinal chemists to explore the large space of complex aromatic ring systems for replacements that are both electronically and sterically analogous. We present this example as an illustration of the more diverse applications of **BROOD**'s fragment-matching technology.

10.1.8 Shape-based belief model

It is a well-known concept in medicinal chemistry that compounds with greater similarity have a higher probability of having shared properties. It is by this premise that project teams seek to explore chemical space around a lead compound in order to discover new active molecules. Based on this concept, Muchmore and colleagues at Abbott attempted to quantify the relationship between various measures of ligand similarity and binding activity ([Muchmore-2008]). A large number of ligands with activities measured across a large number and wide variety of targets were used to generate the probability that two molecules with a given similarity would have activity within one log unit of one another (the p[active]). Several well-regarded ligand similarity techniques showed sigmoidal curves where, at very low similarity, the probability of shared activity was related to the prevalence of inhibitors in the underlying data; whereas, as two molecules approached very high similarity, the probability of similar activity approached something like 35-55%. While perhaps surprising at first, both of these results are sensible. It is well known that while similar molecules are much more likely to have shared activity than two random molecules, similarity is by no means a guarantee of shared activity.

As reported in their paper, one of the similarity techniques with the highest maximum probability was the combo score (shape + color score) from the OpenEye tool ROCS. The critical change from color score to color Tanimoto occurred in ROCS since the publication of Muchmore's paper. The change to color Tanimoto resulted in an even higher maximum probability of 0.512 ([Brown-Muchmore-2008]) for shape + color Tanimoto. The curve that was refit using the Abbott data is used in **BROOD** to convert the overall shape + color Tanimoto similarity of the analog molecule to the query molecule to generate a p(Active) value that indicates the probability, according the Abbott's belief model, that the analog compound will have activity within one log unit of the query compound.

10.1.9 Abbott bioavailability score

In 2005, Martin published a predictive model for the probability that a compound will have bioavailability (f) > 10% in rats ([Martin-2005]). Despite attempts to identify a useful model using straightforward linear combinations of simple parameters, such as logP, logD, donors, acceptors, PSA, or flexibility, Martin noted that different predictive models were required for different ionization states. Anions require a bioavailability model that depends strongly on PSA. By contrast, neutral species, cations, and zwitterions require a model based on the rule-of-five. The combination of these models provides a single model for bioavailability in rats.

10.1.10 Similarity to available compounds

Sometimes it may be desirable to check whether compounds similar to the analogs generated by **BROOD** already exist in commercial or internal databases, allowing a chemist to consider molecules available for purchase before undertaking the synthesis of a given analog.

BROOD uses similarity searching between compounds, based on two-dimensional circular fingerprints, to access compound similarity. The cricular fingerprints encode the presence of molecular substructures based on atom neighborhoods. This similarity is quantified using the Tanimoto coefficient, which measures the ratio of the intersection to the union of the fingerprint bit vectors. A Tanimoto similarity threshold of 0.15 is used as the cutoff to determine whether two compounds are considered structurally similar.

10.1.11 BROOD Database Generation (CHOMP)

BROOD's database generation program, **CHOMP**, fragments molecules by identifying critical bonds that can be broken. **CHOMP** includes three sets of bond-breaking patterns: *RLF (ring, linkers & functional groups)*, RECAP rules [Lewell-1998], and ALL (indicating breaking all non-ring, non-resonance single bonds). By default, **CHOMP** breaks all bonds identified using any of these methods. Users can also specify a SMARTS file of their own bond identifiers. This file should include a series of SMARTS patterns (one per line) that each define two atoms on opposite ends of the bond to be broken. For example, a line with the SMARTS "-" will cause all single bonds to be broken, while a line with the SMARTS "[R]-[!R]" will cause all bonds between ring atoms and non-ring atoms to be broken and "[#6]-!@[!#6!#1]" will cause CHOMP to break every single non-ring bond between a carbon atom and a heteroatom.

The RLF chemical heuristics seek to break compounds into three types of primary fragments; contiguous ring systems, functional groups, and linkers. Contiguous ring systems include any set of atoms that are bonded together by at least one ring bond. Thus, fused rings and spiro rings are included as a single ring system, but biphenyl is broken into two ring systems. Functional groups are defined as any collection of bonded atoms including one or more heteroatoms or unsaturated carbons separated by at most a single fully saturated carbon atom. The linkers are the remaining saturated carbon skeletons. It should be noted that linkers, like functional groups and ring systems, can be terminal (i.e., degree 1).

CHOMP systematically identifies all the molecular fragments that can be generated by breaking one or more of the specified bonds. **CHOMP** first eliminates fragments that have more than 15 heavy atoms or that have more than three attachment points. **CHOMP** also filters the fragments based on commonly used molecular filters, eliminating unstable, reactive, or toxic functional groups. As a final fragment generation step, for every fragment with two attachment points, **CHOMP** generates two single attachment fragments; for every fragment with three attachment points, **CHOMP** generates six fragments with two attachment points and three fragments with single attachment points by capping the open attachment valence with hydrogens. **CHOMP** then eliminates any duplicate fragments.

This first portion of the **CHOMP** algorithm involves only graph processing and can proceed very quickly. Please note, however, that when processing molecular databases of more than a few million compounds, **CHOMP** will use significant amounts of memory. Memory usage is especially aggressive early in the execution, when many new fragments are being identified, and slows somewhat as the algorithm progresses.

The next stage of the **CHOMP** algorithm is to generate 3D conformers using a specialized version of the **OMEGA** algorithm. **CHOMP**'s version of **OMEGA** has a modified TORLIB in order to sample more generously around the attachment points. It also differs from default **OMEGA** parameters in having tighter constraints for removal of RMSD duplicates (as the measure is very sensitive to size) and for having scaled MAXCONFS and EWINDOW parameters based on the size of the fragments. Finally, **CHOMP** writes the fragments into a **BROOD** database. To accomplish this, **CHOMP** carries out several precalculations, including generating physical properties and adding color atoms. CHOMP also segregates the fragment conformers into groups that are likely to match the same types of queries. All these processes help **BROOD** search the databases more quickly.

10.1.12 BROOD Database

BROOD databases are currently in the format of a directory or folder that contains many files. The databases can be easily compressed and uncompressed with standard compression algorithms. They can also be moved, but not easily renamed. To rename the **BROOD** database, you must also rename all the files contained within it. Consistency is required to prevent confusion between the name of the database and the files being searched under that name.

Of the files in the database directory, only one is human-readable. This is the `.info` file. If you open the `.info` file with a text reader, you will see some information about the database, its version, and a manifest of the files that are supposed to be in the database. Whenever a database is read, the manifest is checked to ensure the database has not been corrupted.

10.2 Fragment joining

In the **BROOD** GUI, it is possible to load two fragments and use the *-scoreType* `linkOnly` search to identify potential linkers that can join the fragments with an energetically favorable, medicinally relevant linker in a low-energy conformation. This search can be carried out in a protein's active site, taking account of the need for the linker to fit into the active site as well as join the fragments. For more information on this application, please see the *tutorials section*.

10.3 Cyclization

Long-distance cyclization, like fragment joining, is about finding a chemical fragment that can bridge to moieties given a particular 3D orientation. Use the *-scoreType* `linkOnly` option for long-distance cyclization.

10.4 BROOD Database Generation (CHOMP)

CHOMP offers additional options for duplicate removal. To eliminate a set of potential fragments, such as those from a previous proprietary fragment database, you can specify the file containing those fragments with the *-userUnique* flag.

In many instances, users already have sets of fragments generated by their own means. **CHOMP** will read these fragments directly using the *-userFrags* flag. This flag can be used to generate a **BROOD** database using only these fragments or in conjunction with fragments generated using the molecular fragmentation process discussed above. In either case, **BROOD** will process the user-defined fragments first and process them in the order they are read from the *-userFrags* file. This feature is important for use in conjunction with **BROOD**'s *brood -quickLook* flag because those fragments will be searched first.

This first portion of the **CHOMP** algorithm involves only graph processing and can proceed very quickly. Please note, however, that when processing molecular databases of more than a few million compounds, **CHOMP** will use significant amounts of memory. Memory usage is especially aggressive early in the execution, when many new fragments are being identified, and slows somewhat as the algorithm progresses. For this reason, about 1G of memory for every million molecules you wish to process is recommended. If necessary, you can break a large molecular database into several groups, run each individually, and save their intermediate results before recombining them prior to the final stages of the **CHOMP** algorithm.

To write intermediate files, whether for the purpose of breaking a database into chunks or simply to investigate the fragments **CHOMP** is generating, a user can set the *-omega* flag to `False`. This will cause **CHOMP** to fragment the molecules but not generate a 3D database. The 2D output of this calculation is specified with the *-out* flag, but with the ISM suffix, which contains all the fragments processed so far. These molecules can be examined, filtered, sorted, joined with additional SMILES, and otherwise manipulated. The same molecules can be used with the *-userFrags* flag to generate a **BROOD** database. Other than the changes made by manipulating this intermediate output file, the databases generated by stopping at this intermediate step will be the same as one generated by executing **CHOMP** with the *-omega* flag set to `True` from the start.

Like the **OMEGA** application, the **OMEGA** algorithm inside **CHOMP** can take up to one or occasionally two seconds per fragment. For this reason, **CHOMP** can be run in multi-processor mode using MPI.

As an alternative to using **OMEGA** to generate conformers, **CHOMP** can extract the fragment conformers from a small-molecule structural database. When a small-molecule structural file is passed to **CHOMP** using the *-readConfs* flag, it signals that **CHOMP** should read the conformers rather than generating them. For every fragment identified by **CHOMP**, it searches the structural file and identifies all instances of the fragment. For each instance, the coordinates are extracted and added as a conformer of the fragment. Finally, duplicate conformers for each fragment are removed. In this manner, fragment conformers from any source can be included in the **BROOD** database.

CHAPTER

ELEVEN

BROOD FRAGMENT DATABASE

An essential feature of these methods is the generation of a database of potential fragments. While it may be tempting to generate fragments *de novo*, these approaches often generate unrealistic chemical fragments. Particularly in regards to a method that is related to a common medicinal chemistry technique, we feel it is important to propose known fragments.

The following databases are available with **BROOD**.

11.1 ChEMBL31 Fragment Database

The ChEMBL31 fragment database is derived from the ChEMBL 31 molecule database. The compounds are fragmented and passed through the OpenEye fragment filter, resulting in 20.7 million fragments.

11.2 ChEMBL31_lite Fragment Database

The ChEMBL31_lite fragment database is also derived from the ChEMBL 31 molecule database. It is more size tractable and contains 8.8 million fragments. In addition to applying the OpenEye fragment filter, further filters based on the number of heavy atoms and the minimum frequency of fragments were applied in creating this database.

11.3 brood-database-chembl-3.0.0

brood-database-chembl-3.0.0 is derived from the ChEMBL20 molecule database. The compounds are fragmented resulting in approximately 11 million unique molecular fragments after standard property filtering. The fragments are prioritized according to their medicinal and geometric relevance to fragment replacement and a final collection of approximately six million fragments are retained for the database.

Users may also provide their own fragment database for searching. These fragments databases can be prepared from molecule collections using the **CHOMP** program. **CHOMP** breaks the molecules into fragments, filters the fragments, enumerates undefined stereochemistry, tracks molecules from which the fragments came, and identifies the unique collection of fragments. Once this set is generated, **CHOMP** generates or reads multiconformer representations for each fragment. The conformers can either be generated by **OMEGA** technology within **CHOMP** or extracted from small-molecule crystal structure databases passed into **CHOMP**. As a final step in database generation, **CHOMP** precalculates physical and geometric properties, organizes the fragments for efficient retrieval, and writes a database format that is optimized for efficient **BROOD** searching.

CHAPTER

TWELVE

RELEASE HISTORY

12.1 BROOD 5.0.0

12.1.1 Major Bug Fixes

- **vBROOD** now respects the user defined color atoms and color constraints in the generated `Brood Query`.

12.1.2 Minor Bug Fixes

- The `-quickLook` option in **BROOD** now properly enforces a valid range, reducing the possibility of getting an empty hit list as output.
- An issue has been fixed where occasional high memory usage of **BROOD** occurred while running the application in the `ET` mode.

Related Toolkit Information

12.2 Bioisostere TK 5.0.0

12.2.1 New Features

- The following new classes have been added as preliminary API:
  - OEBroodBuildResult
  - OEBroodCluster
  - OEBroodClusterBuilder
  - OEBroodComboBuilder
  - OEHitlistBuilderOptions
  - OEHitlistBase
  - OEScoreHitlist
  - OEBroodHitlist
  - OEMolCTBuilder
  - OESimilarity
- The following new methods have been added to OEBroodQuery:
  - GetQueryID
  - SetQueryID

- A new constructor overload has been added to OEHitlistBuilder that takes a OEHitlistBuilderOptions as an argument.
- A new overload of Setup has been added to OEBroodDBFilter that takes a molecule as argument.
- The following new methods have been added to OEBroodQuery:
  - GetBuildRes
  - GetScores
- The following new methods have been added to OEBroodScore:
  - Add
  - GetQueryIDs
- The following new functions have been added:
  - OECreateFlipperOptions
  - OECreateFragFilter
  - OEIsFragment
- The following methods have been removed from the OEHitlistBuilder, as they are no longer relevant:
  - GetTrimmedCount
  - TrimHitlist
  - An overload of GetHits that took a vector of OEBroodHit as an argument.
  - GetRemovedHits

12.2.2 Minor Bug Fixes

- The OECreateBroodQuery method now keeps any incoming color atoms associated with the query selection, when available, and does not add any additional color atoms to the query. The behavior is unchanged when no color atoms are present with the incoming selection.
- A bug in fragment property filtering has been resolved. Now by setting the SetUseProperty option in OEBroodScoreOptions to true, you can activate the fragment filtering based on the properties defined by OEExtrinsicPropOptions and OEIntrinsicPropOptions. Previously, the filter was not being applied correctly, which could lead to unexpected fragments being included in the results.
- An issue has been fixed related to proper usage of SetRangeSize and SetRangeOffset so that fragment retrieval from the database during the overlay step now properly respects the specified range boundaries.
- A bug has been fixed where the color score reported by GetColorScore could exceed its valid range in ROCS mode, causing the combined score to exceed its range as well. The color Tanimoto is now computed consistently from the scored conformer.

12.3 BROOD 4.2.1

Minor internal improvements have been made.

12.3.1 Related Toolkit Information

OEBioisostere TK 4.2.1

New Features

- A new overload has been added to the SetSmartsFile method in OEFragmentOptions, enabling the ability to pass in a file name directly.
- A new overload has been added to the OEGenerateConformers, enabling the option for `rotorOffset` compression during conformer generation. The existing overload no longer performs `rotorOffset` compression.
- A new method, HasProteinSelect, has been added to the OEBroodQuery class that can be called on a query to check whether it has a selectivity protein.

Minor Bug Fixes

- Both OEBroodOverlay and OEFragOverlay now transfer any residue information of query fragments into the overlaid fragments within the Overlay method.
- The Build method in OEHitlistBuilder now properly updates the tracer when provided.
- The Init method in OEBroodQuery now properly adds the protein structures for a selectivity check to the query when provided.

12.4 BROOD 4.2.0

12.4.1 New Features

- **BROOD** has been rewritten to take advantage of newly released API and functionality in **Bioisostere TK**. This ensures that the results from the **BROOD** application are consistent with those from **Bioisostere TK**.
- Some flags have been modified for more consistency with the toolkit and other applications. Old flags continue to be functional unless they are *REMOVED*. A complete list of added and *REMOVED* flags can be found in the following table. The list of all existing flags could be looked up by running *brood –help all* in **BROOD** CLI app.

OLD **BROOD**	NEW **BROOD**
-qMol, *-queryMol*	*-in*
-queryFrag	**REMOVED**
-prot, *-select*	**REMOVED**
-noqueryprot	*-ignoreProtein*
-noqueryselectprot	*-ignoreSelectionProtein*
-csv	**REMOVED**
-maxFrag	**REMOVED**
-hitlistProt	**REMOVED**
-info	**REMOVED**
-format	*-out*
-dots	**REMOVED**
-quickLook (bool)	*-quickLook (int)* - **Old Flag REMOVED**
-ET, *-linkonly*	*-scoreType rocs/et/linkonly* - **Old Flags REMOVED**
-sdTag	**REMOVED**
-interval , *-hitinterval*	**REMOVED**
-ringOnly (string->true, none, false,[1,12])	*-ringOnly (int [-2,12]) - true -> (-2)*, *none -> (-1)*, *false -> (0)*
-attachColor	**REMOVED**
-attachFrag	**REMOVED**
-maxHit	*-maxHits*
-attachmentCutoff	*-attachCutoff*
-forcefield	**REMOVED**
-	*-buildType (2d, noopt3d, optimized)*
-attachmentScale	*-attachScale*
-title	**REMOVED**
-fromCT	**REMOVED**
-fileChrg (bool)	*-chargeType (mmff94, fileChrg)* - **Old Flag REMOVED**
-	*-deltaLocalStrain*

- **vBROOD** has been updated to work with OEDesignUnit contained in *.oedu* files for both active site and selectivity proteins. Molecule file formats, such as *.oeb* files, for proteins are no longer supported.
- Two new parameters have been added to both **BROOD** and **vBROOD**:
  - `-quickLook`: Has been changed from being a *boolean* to an *integer* option, enabling users to perform timed searches.
  - `-deltaLocalStrain`: Enables filtering based on relative local strain, instead of absolute local strain.
- The naming pattern of output files has been changed. The following files are now created as a result of a **BROOD** run:
  1. `brood_log.txt`: the log file
  2. `brood_parm.txt`: the param file
  3. `brood_hitlist_rpt.csv`: the hitlist report file
  4. `brood_removed_rpt.csv`: the removed hits report file
  5. `brood_hitlist.oeb.gz`: the hitlist
  6. `brood_removed.oeb.gz`: the removed hitlist

  In addition, the following two files are generated when the run is initiated from **vBROOD**:
  - `brood_info.txt`: the info file

- `brood_query.oeb.gz`: a copy of the ran query

12.4.2 Related Toolkit Information

Bioisostere TK 4.2.0

New Features

- The new class OECPDDatabase has been added as a preliminary API, which can be used to perform 2D similarity comparisons between a **Brood** hit list and an external compound database.
- Eight new overloads have been added to the existing OECreateBroodQuery function. The new overloads extend the API for query generation in the presence of an active site protein and/or a selectivity protein. Additionally, there are new overloads to create a bridging query between two molecules.
- The following new methods have been added to OEHitlistBuilder:
  - GetRemovedHits
  - GetFailedCount
  - TrimHitlist
  - GetTrimmedCount
- The Build method in the OEHitlistBuilder has been expanded to accept an OETracerBase as an optional argument.
- The following new methods have been added to OEBroodHit:
  - GetBuildStatus
  - GetDeltaLocalStrain

Minor Bug Fixes

- Some memory management issues have been fixed in AddScores of the OEHitlistBuilder class.
- The OEBroodScore returned from the Overlay calculation now always contains the fragment, irrespective of whether the overlay was a success.
- The GetFrags method in the OEBroodDBPacket now properly returns an empty iterator if there are no fragments in a packet.
- The SetupRef method in OEFragOverlay is now substantially faster for subsequently initializing it with new reference fragments.
- The Init method in OEDBReader is now returning an unsigned int (instead of bool), reflecting the OEBioisostere OEBroodStatusCode Success or other error codes for database initialization.

12.5 BROOD 4.1.0

Minor internal improvements have been made.

12.6 Bioisostere TK 4.1.0

12.6.1 New features

- A new algorithm, based on OEFlexiOverlay, has been adapted in Build for optimization of generated poses for molecules obtained from fragment replacement. The new algorithm is more robust for optimization of poses for all geometries, including cyclic peptides.

12.6.2 Major bug fixes

- Query building algorithms in OECreateBroodQuery have been improved for robustness. The improved API automatically includes any associated hydrogen atoms as part of the query fragment. Behavior has also been improved to provide better error behavior against invalid selections.
- The new methods SetSingleSourceMol and GetSingleSourceMol have been added to OEFragmentOptions to provide flexibility regarding source molecule storage during database building.

12.6.3 Minor bug fixes

- Some memory management issues have been fixed in Build and Generate.

12.7 BROOD 4.0.0

12.7.1 Major bug fixes

- The `Filtering` button was removed from the **vBROOD** home page to prevent irregular crashes and inconsistent UI behavior.

12.8 Bioisostere TK 4.0.0

This is the first commercial release of **Bioisostere TK**.

12.8.1 New features

The following classes are available as preliminary API with the first release of this toolkit.

- OEBroodBuildOptions
- OEBroodDBFilter
- OEBroodDBPacket
- OEBroodFragPrep
- OEBroodGeneralOptions
- OEBroodHit
- OEBroodHitlistOptions
- OEBroodOverlay
- OEBroodOverlayBase
- OEBroodQuery
- OEBroodScore
- OEBroodScoreOptions
- OEDBBuffer
- OEDBBuilder
- OEDBReader
- OEDBScreenOptions
- OEDBWriter
- OEFragConfReader

- OEFragmentOptions
- OEFragOverlay
- OEHitlistBuilder
- OEMolPropertyOptions
- OEIntrinsicPropOptions
- OEExtrinsicPropOptions

The following functions are available as preliminary API with the first release of this toolkit.

- OEBioisostereGetArch
- OEBioisostereGetLicensee
- OEBioisostereGetPlatform
- OEBioisostereGetRelease
- OEBioisostereGetSite
- OEBioisostereGetVersion
- OEBioisostereIsLicensed
- OECreateBroodQuery
- OEGenerateConformers
- OEGetBroodStatus
- OEGetDBIndices
- OEReadBroodQuery
- OEWriteBroodQuery

12.9 BROOD 3.2.2

12.9.1 Minor bug fixes

- **BroodDBMerge** can now merge databases built from any version of `3.x`.

Minor internal improvements have been made.

12.10 BROOD 3.2.1

12.10.1 Minor bug fixes

- The default for the `-property` parameter in **BROOD** has been changed to `False`, so it is consistent with **vBROOD**.

12.11 BROOD 3.2.0

Fall 2022

12.11.1 New Features

- Two new fragment databases, ChEMBL31_lite and ChEMBL31, are now available. Refer to the *BROOD Fragment Database* section for details about these new fragment databases.

12.11.2 Major bug fixes

- An issue that caused *BROOD* to generate incorrect `Molecular Tanimoto Combo` with corresponding `p(active)` has been fixed.
- An issue that caused *CHOMP* to generate a much larger database than expected, by generating too many conformers per fragment, has been fixed.

12.12 BROOD 3.1.7

Spring 2022

- Minor internal improvements have been made.

12.13 BROOD 3.1.6

Fall 2021

- Minor internal improvements have been made.

12.14 BROOD 3.1.5

July 2021

- Minor internal improvements have been made.

12.15 BROOD 3.1.4.0

Fall 2020

- Minor internal improvements have been made.

12.15.1 Documentation changes

- Documentation and command line help for the `-smarts` parameter have been updated with information on built-in fragmentation methods.

12.16 BROOD 3.1.3

Spring 2020

- This version of **BROOD** has been built using **OEToolkits 2020.0**.

12.16.1 Minor bug fixes

- **BROOD** no longer fails when stereos are not specified for query fragments.
- **BROOD**'s warning messages about invalid query fragments are now clearer.
- Clicking *Show on Disk* in **vBrood** on macOS no longer uses Apple Script. This avoids OS-generated log messages in the terminal window.

- **vBrood** no longer generates *libpng* warning messages upon startup.
- Clicking *Run* after creating a query no longer uses the deprecated `OEFileRandomName` function. This avoids warning messages in the terminal window.

12.17 BROOD 3.1.2

Nov 2019

- This version of **BROOD** has been built using **OEToolkits 2019.Oct**. The previous version was built using **OEToolkits 2019.Apr**.

12.17.1 Major bug fixes

- A bug that caused **CHOMP** to crash in some cases when using `-userFrag` has been fixed.

12.17.2 Minor bug fixes

- A bug that caused the `-userUnique` flag of **CHOMP** to not parse absolute or relative paths correctly has been fixed.
- **CHOMP** now properly runs in multi-node MPI mode when passing in a custom filter file.
- **CHOMP** now honors `-maxHvy` values greater than the default value of 15 for post-fragmentation filtering.
- A segmentation fault caused by **CHOMP**'s failure to construct a valid molecule fragment has been fixed.

12.18 BROOD 3.1.1

May 2019

- This version of **BROOD** has been built using **OEToolkits 2019.Apr**. The previous version was built using **OEToolkits 2018.Oct**.

12.18.1 Major bug fixes

- **BROOD** now correctly finds analogs for queries that does not contain any color atoms.
- A segmentation fault caused by **CHOMP**'s failure to construct a valid molecule fragment has been fixed.

12.19 BROOD 3.1.0

November 2018

- This version of **BROOD** has been built using **OEToolkits 2018.Oct**. The previous version was built using **OEToolkits 2015.Jun**.

12.19.1 Minor bug fixes

- The `Documentation` sub-menu of **vBROOD** now correctly points to the online documentation.

12.20 BROOD 3.0.0

July 2015

This is a significant release of **BROOD**. It primarily to add user-requested features and to fix user-reported bugs. It includes important improvements in all four programs included in the **BROOD** suite as well as numerous bug fixes. This release of **BROOD** also include support for several new Linux platforms as well as support for MPI multi-processing on Windows.

12.20.1 Platform Support

- Added support for RedHat 6, 7, Ubuntu 14, OSX 10.7, 10.8, 10.9
- Dropped support for RedHat 5 32 bit and OSX 10.6
- Added MPI support for Windows
- This will be the last release for SuSe 11

12.20.2 vBROOD

- Added auto-suggestion of fragments for replacement in query
- Reduced arbitrary strict stereo in query build
- Streamlined query building, searching and results view
- Improved cluster handling in hitlist filtering
- Added 2D alignment of results viewing
- Added upper and lower bound to filter properties

12.20.3 Results Viewer

- Annotation of hitlist molecules with color atoms
- Improved cluster navigation
- Cleaned up cluster naming and manipulation
- Improved viewer widget layout
- Improved visibility of keyboard shortcuts and navigation

12.20.4 BROOD

- Substantial improvement to the calculation of local strain
- Improved elimination of over-strained analogs
- Added the ability to suppress protein output in hitlists
- Added the ability to annotate results with similar molecules from a user collection
- Added protein selectivity
- Fixed problems with property flag
- Fixed bug in frequency data
- Added upper and lower bound to filter properties

12.20.5 CHOMP

- Added the ability to pull fragment conformations from user-supplied structure database
- Added new fragment database derived from ChEMBL [Bento-2014]
- Mapping of top 5 source molecules onto each fragment
- Added enumeration of stereochemistry to CHOMP
- Fixed inconsistency between single-processor and MPI duplicate removal

12.20.6 Merge

- Improved user interface with clearer output and verbose output
- Added support for duplicate recognition and removal
- Added ability to overwrite existing databases if necessary

12.21 BROOD 2.0.0

April 2010

This is a major rewrite of **BROOD** including a new database format, faster searching, a much larger database and a graphical interface. The graphical interface greatly simplifies selection of a molecular fragment for replacement and facilitates specification of constraints.

12.21.1 New Features:

- **BROOD** now comes with a new graphical application, **vBROOD**, for setting up searches.
- Includes specification and graphical editing of color constraints. Pre-defined standard constraints and user-defined constraints are available.
- New database of more than 3.6 million, common, medicinal fragments.
- New database format for faster search times.
- Improved hitlist organization, including a hierarchy based on reduced graphs.
- Optimized analog molecule overlay.
- Allows incorporation of a protein structure for clash detection.
- Graphical interface allows designing of some physical properties.
- Assessment of synthetic accessibility of generated structures.
- Implementation of activity belief model for analog molecules.
- Includes a VIDA extension for ease of exploring results.
- 1-step database generation, including fragment generation and filtering, charge and property calculation, conformer generation, and buffered database creation.

12.22 BROOD 1.1.2

February 2009

This is a major bug-fix release for **BROOD** v1.1. This fix includes a major fix to the licensing system that allows **BROOD** to use licenses that expire in the year 2010 and later.

Release Notes:

1. Updated the license system to work with licenses that expire in 2010 and later.
2. BROOD installations now support having multiple versions and platform architectures in the same directory structure.

12.23 BROOD 1.1.1

August 2007

This is a minor bug-fix release for **BROOD** v1.1. This release includes a handful of fixes to bugs that have been reported over the past three months. Several of the most important bugs related to the *-ET* flag. We strongly recommend that anyone using electrostatic similarity should upgrade to this new version.

Release Notes:

1. Fixed major bug that caused molecules in the queryAnalog.build hitlist to be disconnected fragments when the *-ET* flag was true.
2. Fixed bug that allowed the electrostatic Tanimoto to still include the attachment score.
3. Fixed bug that failed to include the total electrostatic Tanimoto score to be recorded in the report file, even when all the components were present.
4. Added explicit hydrogens to the entire rebuilt molecule when the *-build* flag is used with 3D input.
5. Removed fragmented partial charges on fragments and built molecules before they are written as output.
6. Fixed counting error for the number of warnings reported in the information and log files.
7. Categorized and ordered the appearance of parameters in the auto-help.

12.24 BROOD 1.1.0

May 2007

This is a major new release of **BROOD**, OpenEye's fragment replacement search program. This release fixes all known bugs from v1.0. It includes significant re-working of the scoring methods. Most notable among these are that the attachment geometry score is now used as a cutoff to assure that all poses have reasonable attachment geometries, but fragments are ranked according to their shape plus chemical similarity or their shape plus electrostatic similarity. Further, several steps have been taken to eliminate the bias towards "over-colored" fragments (including implementing Tanimoto color rather than scaled color). Last, but not least, there are dramatic improvements to the databases and database content, as well as **CHOMP**, the program for user-generated databases. The two databases provided with **BROOD** now contain the most common fragments found in vendor compounds, are more complete, are better filtered, have improved conformer sampling and contain proper MMFF partial charges. We believe this release is a major step toward maturation of the fragment searching methods available from OpenEye.

BROOD and **CHOMP** Features:

1. Improved conformers of constructed analog molecules to be as close to the original build molecule conformers as possible.
2. Added each score component to the report file as well as to the SDTags.
3. Added the frequency information to the SDTags.
4. Added bondOrder flag that by default requires attachment point bond order to be the same as those in the query fragment.
5. Changed color scoring from scaled-color to Tanimoto-color in order to prevent over-scoring of highly colored fragments.

6. Added shapeCut parameter to prevent very bad overlaps from appearing in hitlist when no good matches are found in the database.
7. Added attachCut parameter to eliminate fragments with poor attachment geometries regardless of how good their shape and chemical similarity is.
8. Removed the attachment score as a component to the color combo and ET combo score. Color combo is now the sum of the shape and chemical color Tanimoto scores. ET combo is now the sum of the sum of the shape and the electrostatic Tanimoto scores.
9. Improved and clarified the handling of 2D and 3D coordinates in the query fragment and the build molecule. This includes generating depiction coordinates for the built analog molecules when the input build molecule doesn't contain 3D coordinates.
10. Improved documentation of hitlists and database generation.
11. Updated OEChem to include the ability to read SDTags from the .oeb formatted database file.
12. Added info file to CHOMP runs.
13. Added warnings when short hitlists are encountered.
14. Added fragment filtering progression and hitlist size to info file.
15. Cleaned up the column ordering in the report file.
16. Properly handle cases with more than three attachment points.
17. Changed :option: *-sdtag* default from "true" to "verbose".
18. Improved reporting of early initialization problems.

BROOD and CHOMP Bug Fixes:

1. Built analog molecules are now aligned with the input build molecule.
2. Fixed bug that caused the query fragment to be moved before the search began and never replaced.
3. Fixed bug that prevented the proper behavior of the :option: *-build* flag in conjunction with the :option: *-ET* flag.
4. Corrected bug related to the :option: *-fileCharge* flag.
5. Fixed query fragment to build molecule matching bug related to differing aromaticity models.
6. Fixed two CHOMP bugs that prevented the :option: *-smarts* flag from working properly.
7. Removed potential for double-counting non-default attachment scale parameter when used with the :option: *-ET* flag.
8. Fixed memory leak in large hitlist management system.

Database Features:

1. Improved database fragment content to include common fragments of all sizes up to 15 heavy atoms (or 350 MW).
2. Generate, use and attach fragment frequency information to each fragment.
3. Developed new methods for filtering fragments rather than filtering whole molecules before fragmentation.
4. Eliminated primary-fragment database.
5. Added simple chemical complexity filters.
6. Improved perception of anionic hydrogen-bond perception.

Database Bug Fixes:

1. Fixed partial charge errors that occurred on some attachment point methyl groups.

2. Fixed polar-proton partial charge error.
3. Removed duplicate fragments.
4. Fixed ring-opening bug that fragmented most aliphatic heterocycles, thus preventing them from occurring in the original database.
5. Fixed branching bug that prevented some unusual branching patterns from occurring in the database.
6. Improved fragmentation schemes for identifying functional groups.

12.25 BROOD 1.0.1

May 2006

This version fixes several minor bugs.

12.26 BROOD 1.0.0

April 2006

This is the first official release of the bioisostere search program **BROOD**. This release is a significant overhaul of the beta2 release. It includes dramatic feature enhancements, bug fixes and optimizations. Most notable are improvements in the manner in which fragments are ranked and addition of electrostatic similarity. Further, this release has been applied to many bioisostere examples from the medicinal chemistry literature that have helped to refine its performance. Finally, this release makes generation of a custom fragment database much easier.

Release Notes:

1. Fixed major bug in initial fragment orientation.
2. Added electrostatic similarity measure
3. Significantly refined the evaluation functions for ranking bioisosteres.
4. Added the ability to replace the query fragment with bioisosteric fragments in an initial molecule.
5. Added separate handling of query analog fragments to avoid filling the hitlists with obvious answers.
6. Allow user specified partial charges and query conformer.
7. Allow user defined scaling of the contribution to similarity resulting from the attachment points.
8. Added the ability to write intermediate hitlists during execution.
9. Allow user to limit results to ring fragments only.
10. Refinement of default fragment databases.
11. Execution speed optimizations.
12. Addition of two applications for generating a database of molecular fragments.

12.27 BROOD 0.9.b2

January 2006

This beta release of the bioisostere program includes several significant improvements despite coming soon after the first beta release. Most importantly, this release fixes a licensing bug, a best-overlay bug, an RMS sorting bug, and a hydrogen placement bug in the fragment library. New features include, improved sampling for fragments with a single attachment point and more complete SDTag data.

Release Notes:

1. Improved sampling and initial orientation of fragments with a single attachment point.
2. Incorporated color score file into the application.
3. Improved proton placement in the fragment library.
4. SDTag data now includes all the scores of a fragment.
5. Fixed RMS sorting bug.
6. Fixed best-overlay NaN bug.
7. Resolved internal licensing problems.
8. Improved documentation of attachment point representation in SMILES and SDF file formats.

12.28 BROOD 0.9.b1

December 2005

This is the first public release of the bioisostere program. The primary purpose of this release is to garner feedback concerning the functional aspects of the scientific product. Critical questions to address include; does it work in prospective studies, are users satisfied with the fragment database, are the three types of bioisosteres incomplete, sufficient or overwhelming. This release has a minimal feature list, but should be stable and robust. While it is not exceptionally slow, it has not been optimized for speed nor has it been adapted to PVM or MPI.

New Features:

1. Multiconformer database of over 90 thousand unique molecular fragments generated from commercially available compounds.
2. Automatic 3D conformer generation for query fragments.
3. Automated generation of hitlists for geometric, chemical and balanced classes of bioisosteres.

CHAPTER

THIRTEEN

CITATION

Note

To cite BROOD please use the following:

BROOD 5.0.0.0. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.eyesopen.com.

CHAPTER

FOURTEEN

COPYRIGHT AND TRADEMARKS

© 1997-2024 Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA. All rights reserved.

OpenEye, Lexichem, ROCS, Grapheme, and Orion are registered trademarks of Cadence Design Systems, Inc.

This material contains proprietary information of Cadence Design Systems, Inc. Use of copyright notice is precautionary only and does not imply publication or disclosure.

The information supplied in this document is believed to be true, but no liability is assumed for its use or the infringement of the rights of others resulting from its use. Information in this document is subject to change without notice and does not represent a commitment on the part of Cadence Design Systems, Inc.

This package is sold/licensed/distributed subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out, or otherwise circulated without Cadence Design Systems' prior consent, in any form of packaging or cover other than that in which it was produced. No part of this manual or accompanying documentation, may be reproduced, stored in a retrieval system on optical or magnetic disk, tape, CD, DVD or other medium, or transmitted in any form or by any means, electronic, mechanical, photocopying recording, or otherwise for any purpose other than for the purchaser's personal use without a legal agreement or other written permission granted by Cadence.

This product should not be used in the planning, construction, maintenance, operation or use of any nuclear facility nor the flight, navigation, or communication of aircraft or ground support equipment. Cadence Design Systems, Inc., shall not be liable, in whole or in part, for any claims arising from such use, including death, bankruptcy or outbreak of war.

Windows is a registered trademark of Microsoft Corporation. Apple, OS X, macOS, and Macintosh are registered trademarks of Apple Inc. UNIX is a registered trademark of the Open Group. Linux is a registered trademark of Linus Torvalds. Red Hat is a registered trademark of Red Hat, Inc. SUSE, SLED, and SLES are registered trademarks of Novell, Inc. Ubuntu is a registered trademark of Canonical Ltd.

SYBYL is a registered trademark of Certara, L.P. MDL, BIOVIA and ISIS are registered trademarks of Dassault Systemes and/or its affiliates. SMARTS and SMIRKS are registered trademarks of Daylight Chemical Information Systems, Inc. MacroModel is a registered trademark of Schrodinger, LLC.

Python is a registered trademark of the Python Software Foundation. Django is a registered trademark of the Django Software Foundation. Java is a registered trademark of Oracle and/or its affiliates.

Other products and software packages referenced in this document are trademarks and registered trademarks of their respective vendors or manufacturers.

CHAPTER

FIFTEEN

SAMPLE CODE

For all of the code examples and samples within documents published by OpenEye, Cadence Molecular Sciences, the following terms of use apply.

TERMS FOR USE OF SAMPLE CODE

The software below ("Sample Code") is provided to current licensees or subscribers of Cadence products or SaaS offerings (each a "Customer"). Customer is hereby permitted to use, copy, and modify the Sample Code, subject to these terms. Cadence claims no rights to Customer's modifications. Modification of Sample Code is at Customer's sole and exclusive risk. Sample Code may require Customer to have a then current license or subscription to the applicable Cadence offering.

THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. CADENCE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be liable for any damages or liability in connection with the Sample Code or its use.

CHAPTER

SIXTEEN

CITATION

Please follow the guidelines in this document to cite use of our products in research publications and other documents.

For more information, please contact OpenEye Support.

16.1 Orion®

To cite use of Orion, please use the following:

```
Orion <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

For example:

```
Orion 2025.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.eyesopen.com.
```

The version number for Orion appears in the product's release notes, for example, https://docs.eyesopen.com/orion-user-interface/releasenotes/index.html.

16.2 Orion Floes

To cite use of an Orion-based floe package, please use the following:

```
OpenEye <package-name> <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe,␣
↪NM. http://www.eyesopen.com.
```

For example:

```
OpenEye Large-Scale Floes 4.2.3. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

The version number for a floe package is displayed on the first page of the package's release notes. For example: https://docs.eyesopen.com/floe/modules/large_scale_floes/docs/source/releasenotes_2025_1.html.

16.3 Toolkits and Applications

To cite OpenEye toolkits or applications used in your work, please use the following:

```
OpenEye Toolkits [or Applications] <version-number>. OpenEye, Cadence Molecular Sciences,
↪ Santa Fe, NM. http://www.eyesopen.com.
```

For example:

```
OpenEye Toolkits 2024.2.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

The Applications and Toolkits are released as a package, multiple times per year. The first part of the version number designates the calendar year. The version number appears on the summary document for the package.

If you want to cite an individual application or toolkit, you can use the syntax:

```
<product name> <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

where <*product name*> is:

- Bioisostere TK
- OEChem TK
- OEDepict TK
- OEDocking TK
- OEFF TK
- Eon TK
- FastROCS TK
- Grapheme TK
- GraphSim TK
- Lexichem TK
- OEMedChem TK
- MolProp TK
- Omega TK
- Quacpac TK
- Shape TK
- Spicoli TK
- Spruce TK
- Szmap TK
- Szybki TK
- Zap TK
- AFITT Application
- BROOD Application
- OEDocking Application
- EON Application
- OMEGA Application
- PICTO Application
- pKa-Prospector Application
- QUACPAC Application

- ROCS Application
- SiteHopper Application
- SPRUCE Application
- SZMAP Application
- SZYBKI Application
- VIDA Application

You can use the version number of the individual Application or Toolkit. For example:

```
SZYBKI Application 2.8.0.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://
↪www.eyesopen.com.

Szybki TK 2.8.0.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

The first documentation pages give the product names and version numbers, as in https://docs.eyesopen.com/applications/szybki/index.html and https://docs.eyesopen.com/toolkits/python/szybkitk/index.html.

16.4 Publications for Bibliographies

Please also cite these publications if the corresponding product was used in your research.

16.4.1 Orion

Sørensen, J., Bannan, C.C., Calabrò, G., Jain, V., Ovanesyan, G., Smith, A., Zhang, S., Bayly, C.I., Darden, T.A., Geballe, M.T., LeBard, D.N., McGann, M., Moon, J.B., Muddana, H.S., Shewmaker, A., LaFon, J., Tolbert, R.W., Skillman, A.G., Nicholls, A. Orion® A Cloud-Native Molecular Design Platform. In *Computational Drug Discovery*. V. Poongavanam and V. Ramaswamy, Eds. Wiley-VCH, 2024; pp. 579–615. DOI: 10.1002/9783527840748.ch24

16.4.2 AFITT and FLYNN Applications

Wlodek, S.; Skillman, A.G.; Nicholls, A. Automated Ligand Placement and Refinement with a Combined Force Field and Shape Potential. *Acta Crystallogr., Sect. D: Struct. Biol.* **2006**, *62* (7), 741–749. DOI: 10.1107/s0907444906016076

16.4.3 OEDocking Application and Toolkit

Kelley, B.P.; Brown, S.P.; Warren, G.L.; Muchmore, S.W. POSIT: Flexible Shape-Guided Docking For Pose Prediction. *J. Chem. Inf. Model.* **2015**, *55* (8), 1771-1780. DOI: 10.1021/acs.jcim.5b00142

McGann, M. FRED Pose Prediction and Virtual Screening Accuracy. *J. Chem. Inf. Model.* **2011**, *51* (3), 578-596. DOI: 10.1021/ci100436p

McGann, M. FRED and HYBRID Docking Performance on Standardized Datasets. *J. Comput.-Aided Mol. Des.* **2012**, *26* (8), 897-906. DOI: 10.1007/s10822-012-9584-8

16.4.4 OMEGA Application and Toolkit

If you are using OMEGA in torsion driving mode, please cite the following two publications.

Hawkins, P.C.D.; Skillman, A.G.; Warren, G.L.; Ellingson, B.A.; Stahl, M.T. Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and the Cambridge Structural Database. *J. Chem. Inf. Model.* **2010**, *50* (4), 572-584. DOI: 10.1021/ci100031x

Hawkins, P.C.D. and Nicholls, A. Conformer Generation with OMEGA: Learning from the Data Set and the Analysis of Failures. *J. Chem. Inf. Model.* **2012**, *52* (11), 2919–2936. DOI: /10.1021/ci300314k

If you are using OMEGA in macrocyle mode, please cite the following.

Hawkins, P.C.D. and Wlodek, S. Decisions with Confidence: Application to the Conformation Sampling of Molecules in the Solid State. *J. Chem. Inf. Model.* **2020**, *60* (7), 3518–3533. DOI: /10.1021/acs.jcim.0c00358

16.4.5 ROCS Application

Hawkins, P.C.D.; Skillman, A.G.; Nicholls, A. Comparison of Shape-Matching and Docking as Virtual Screening Tools. *J. Med. Chem.* **2007**, *50* (1), 74-82. DOI: 10.1021/jm0603365

16.5 OpenEye MMDS Web Service

To cite use of the Macromolecular Data Service (MMDS) web service, please use the syntax:

```
Macromolecular Data Service <version-number>. OpenEye, Cadence Molecular Sciences, Santa␣
↪Fe, NM. http://www.eyesopen.com.
```

For example:

```
Macromolecular Data Service 1.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

The MMDS version number appears on the web service's release notes.

CHAPTER

SEVENTEEN

TECHNOLOGY LICENSING

OpenEye products use the following technology under license.

Some of the open source technologies we use are licensed under the GNU Public License (GPL) version 2 or 3. In all instances, these technologies are communicated with using the "at arms length" principle, using pipes and command line arguments.

While the software is in some instances assembled into a container, that assembly does not invoke the GPL copyleft scope. For each Floe package where these are used, the environment log clearly details the installed programs and their version numbers. With the links below for each technology, the source for each of those programs is available.

Table 1: Licensed Technology

Name of Project	Website	License URL
_libgcc_mutex	https://github.com/conda-forge/_libgcc_mutex-feedstock	https://github.com/conda-forge/_libgcc_mutex-feedstock/blob/master/LICENSE.txt
_openmp_mutex	https://github.com/conda-forge/_openmp_mutex-feedstock	https://github.com/conda-forge/_openmp_mutex-feedstock/blob/main/LICENSE.txt
ablang2	https://github.com/TobiasHeOl/AbLang2	https://github.com/TobiasHeOl/AbLang2/blob/main/LICENSE
abseil-cpp	https://github.com/abseil/abseil-cpp	https://github.com/abseil/abseil-cpp/blob/master/LICENSE
absl-py	https://github.com/abseil/abseil-py	https://github.com/abseil/abseil-py/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
aiohappyeyeballs	https://github.com/aio-libs/aiohappyeyeballs	https://github.com/aio-libs/aiohappyeyeballs/blob/main/LICENSE
aiohttp	https://docs.aiohttp.org/en/stable/	https://github.com/aio-libs/aiohttp/blob/master/LICENSE.txt
aiosignal	https://github.com/aio-libs/aiosignal	https://github.com/aio-libs/aiosignal/blob/master/LICENSE
Amazon Linux 2	https://aws.amazon.com/amazon-linux-2	N/A
AmberUtils	http://ambermd.org	N/A
ambit	https://github.com/khimaros/ambit	https://github.com/khimaros/ambit/blob/master/LICENSE
amqp	https://github.com/celery/py-amqp	https://github.com/celery/py-amqp/blob/master/LICENSE
anaconda-anon-usage	https://github.com/anaconda/anaconda-anon-usage	https://github.com/anaconda/anaconda-anon-usage/blob/main/LICENSE
angular	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md
angular-animate	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md
angular-cache	https://github.com/jmdobry/angular-cache	https://github.com/jmdobry/angular-cache/raw/master/LICENSE
angular-cookies	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
angular-loggly-logger	https://github.com/ajbrown/angular-loggly-logger	https://github.com/ajbrown/angular-loggly-logger/raw/master/LICENSE.md
angular-mocks	https://github.com/angular/angular.js	https://raw.githubusercontent.com/angular/angular.js/master/LICENSE
angular-resource	https://github.com/angular/angular.js	https://raw.githubusercontent.com/angular/angular.js/master/LICENSE
angular-toggle-switch	https://github.com/cgarvis/angular-toggle-switch	https://github.com/cgarvis/angular-toggle-switch/blob/master/LICENSE
angular-ui-grid	https://github.com/angular-ui/ui-grid	https://github.com/angular-ui/ui-grid
angular-ui-router	https://github.com/angular-ui/ui-router	https://github.com/angular-ui/ui-router/raw/master/LICENSE
angular-ui-tree	https://github.com/angular-ui-tree/angular-ui-tree	https://github.com/angular-ui-tree/angular-ui-tree/raw/master/LICENSE
angular-vs-repeat	https://github.com/kamilkp/angular-vs-repeat	https://raw.githubusercontent.com/kamilkp/angular-vs-repeat/master/LICENSE
aniso8601	https://bitbucket.org/nielsenb/aniso8601	https://bitbucket.org/nielsenb/aniso8601/src/master/LICENSE
annotated-types	https://github.com/annotated-types/annotated-types	https://github.com/annotated-types/annotated-types/blob/main/LICENSE
antlr4-python3-runtime	http://www.antlr.org OR https://pypi.org/project/antlr4-python3-runtin	https://github.com/conda-forge/antlr4-python3-runtin/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
anyio	https://github.com/agronholm/anyio	https://github.com/agronholm/anyio/blob/master/LICENSE
appdirs	http://github.com/ActiveState/appdirs	http://github.com/ActiveState/appdirs/blob/master/LICENSE.txt
appengine	https://google.golang.org/appengine	https://github.com/golang/appengine/blob/master/LICENSE
arabic-reshaper	https://github.com/mpcabd/python-arabic-reshape	https://github.com/mpcabd/python-arabic-reshape/blob/master/LICENSE
archspec	https://github.com/archspec/archspec/blob/master/README.md	https://github.com/archspec/archspec/blob/master/COPYRIGHT
argon2-cffi	https://github.com/hynek/argon2-cffi	https://github.com/hynek/argon2-cffi/blob/master/LICENSE
argon2-cffi-bindings	https://github.com/hynek/argon2-cffi-bindings	https://github.com/hynek/argon2-cffi-bindings/blob/master/LICENSE
arpack	https://www.caam.rice.edu/software/ARPACK/	https://www.caam.rice.edu/software/ARPACK/
arrow	https://github.com/arrow-py/arrow	https://github.com/arrow-py/arrow?tab=Apache-2.0-1-ov-file#readme
ascii85	https://github.com/huandu/node-ascii85	https://github.com/huandu/node-ascii85/raw/master/LICENSE
ase	https://wiki.fysik.dtu.dk/ase/	https://wiki.fysik.dtu.dk/ase/development/licenseinfo.html#license-info
asgiref	https://github.com/django/asgiref/	https://github.com/django/asgiref/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
asn1crypto	https://github.com/wbond/asn1crypto	https://github.com/wbond/asn1crypto/blob/master/LICENSE
assertions go-render	https://github.com/smartystreets/assertions/internal/go-render/render	https://github.com/smartystreets/assertions/blob/master/internal/go-render/LICENSE
assertions oglmatchers	https://github.com/smartystreets/assertions/internal/oglematchers	https://github.com/smartystreets/assertions/blob/master/internal/oglematchers/LICENSE
assertions	https://github.com/smartystreets/assertions	https://github.com/smartystreets/assertions/blob/master/LICENSE.md
asttokens	https://github.com/gristlabs/asttokens	https://github.com/gristlabs/asttokens/blob/master/LICENSE
astunparse	https://github.com/simonpercivall/astunparse	https://github.com/simonpercivall/astunparse/blob/master/LICENSE
async-lru	https://github.com/aio-libs/async-lru	https://github.com/aio-libs/async-lru/blob/master/LICENSE
async-timeout	https://github.com/aio-libs/async-timeout	https://github.com/aio-libs/async-timeout/blob/master/LICENSE
atk-1.0	https://docs.gtk.org/atk/	https://gitlab.gnome.org/GNOME/atk/-/blob/master/COPYING
atomic	https://github.com/uber-go/atomic	https://github.com/uber-go/atomic/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
atomicwrites	https://github.com/untitaker/python-atomicwrites	https://github.com/untitaker/python-atomicwrites/blob/master/LICENSE
attrs	https://www.attrs.org/	https://github.com/python-attrs/attrs/blob/main/LICENSE
aws-sdk-go	https://github.com/aws/aws-sdk-go	https://github.com/aws/aws-sdk-go/blob/master/LICENSE.txt
babel	https://babel.pocoo.org/ OR https://github.com/python-babel/babel	https://github.com/python-babel/babel/blob/master/LICENSE OR https://babel.pocoo.org/en/latest/license.html#babel-license
Babel	https://github.com/python-babel/babel	https://github.com/python-babel/babel/blob/master/LICENSE
backcall	https://github.com/takluyver/backcall	https://github.com/takluyver/backcall/blob/master/LICENSE
backports	https://github.com/brandon-rhodes/backports	https://github.com/brandon-rhodes/backports/blob/master/LICENSE.TXT
backports.functools-lru-cache	https://github.com/jaraco/backports.functools_lru_cache	https://github.com/jaraco/backports.functools_lru_cache/blob/main/LICENSE
backports.zstd	https://github.com/rogdham/backports.zstd	https://github.com/rogdham/backports.zstd?tab=License-1-ov-file#readme
base62	https://github.com/kare/base62	https://github.com/kare/base62/blob/main/LICENSE
beautifulsoup4	https://www.crummy.com/software/BeautifulSoup/	https://www.crummy.com/software/BeautifulSoup/

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
billiard	https://github.com/celery/billiard	https://github.com/celery/billiard/blob/master/LICENSE.txt
biopython	https://biopython.org	https://github.com/biopython/biopython/blob/master/LICENSE.rst
biotite	https://github.com/biotite-dev/biotite/blob/master/README.rst	https://github.com/biotite-dev/biotite/blob/master/LICENSE.rst
bitarray	https://github.com/ilanschnell/bitarray	https://github.com/ilanschnell/bitarray/blob/master/LICENSE
bitset	https://github.com/willf/bitset	https://github.com/willf/bitset/blob/master/LICENSE
blas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
bleach	https://github.com/mozilla/bleach	https://github.com/mozilla/bleach/blob/main/LICENSE
blessings	https://github.com/erikrose/blessings	https://github.com/erikrose/blessings/blob/master/LICENSE
blinker	https://pythonhosted.org/blinker/	https://github.com/pallets-eco/blinker/blob/master/LICENSE
bokeh	https://github.com/bokeh/bokeh	https://github.com/bokeh/bokeh/blob/branch-4.0/LICENSE.txt
blosc	https://github.com/Blosc/python-blosc	https://github.com/Blosc/python-blosc/blob/master/LICENSE.txt
blosc2	https://github.com/Blosc/python-blosc2	https://github.com/Blosc/c-blosc2/blob/main/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
boltons	https://github.com/mahmoud/boltons	https://github.com/mahmoud/boltons/blob/master/LICENSE
Boltz	https://boltz.bio	https://github.com/jwohlwend/boltz/blob/main/LICENSE
boost	https://www.boost.org/doc/libs/1_71_0/libs/python/doc/html/index.html	https://www.boost.org/LICENSE_1_0.txt
boost-cpp	https://www.boost.org/doc/libs/1_71_0/libs/python/doc/html/index.html	https://www.boost.org/LICENSE_1_0.txt
bootstrap-vue	https://github.com/bootstrap-vue/bootstrap-vue	https://github.com/bootstrap-vue/bootstrap-vue/raw/master/LICENSE
boto3	https://github.com/boto/boto3	https://github.com/boto/boto3/blob/develop/LICENSE
botocore	https://github.com/boto/botocore	https://github.com/boto/botocore/blob/develop/LICENSE.txt
Bottleneck	https://bottleneck.readthedocs.io/en/latest/index.html	https://bottleneck.readthedocs.io/en/latest/license.html
Brotli	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
brotli-bin	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
brotli-python	http://python-hyper.org/projects/brotlipy/en/latest/	https://github.com/python-hyper/brotlicffi/blob/main/LICENSE
brotlicffi	https://github.com/python-hyper/brotlicffi	https://github.com/python-hyper/brotlicffi?tab=MIT-1-ov-file#readme
brotlipy	https://github.com/python-hyper/brotlicffi	https://github.com/python-hyper/brotlicffi/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
bson	http://github.com/py-bson/bson	https://github.com/py-bson/bson/blob/master/LICENSE_APACHE
bytefmt	https://code.cloudfoundry.org/bytefmt	https://github.com/cloudfoundry/bytefmt/blob/master/LICENSE
bzip2	https://www.bzip.org	https://spdx.org/licenses/bzip2-1.0.6.html
c-ares	https://github.com/c-ares/c-ares	https://github.com/c-ares/c-ares/blob/main/LICENSE.md
ca-certificates	https://github.com/conda-forge/ca-certificates-feedsto	https://wiki.mozilla.org/CA/Included_Certificates
cached-property	https://github.com/pydanny/cached-property	https://github.com/pydanny/cached-property/blob/master/LICENSE
cachetools	https://github.com/tkem/cachetools/	https://github.com/tkem/cachetools/blob/master/LICENSE
cairo	https://pycairo.readthedocs.io/en/latest/	https://github.com/pygobject/pycairo/blob/master/COPYING
canvas	https://github.com/Automattic/node-canvas	https://github.com/Automattic/node-canvas
cctbx	https://github.com/cctbx/cctbx_project	https://github.com/cctbx/cctbx_project/blob/master/cctbx/LICENSE_2_0.txt
celery	https://github.com/celery/celery	https://github.com/celery/celery/blob/master/LICENSE
Cerberus	https://docs.python-cerberus.org/en/stable/	https://docs.python-cerberus.org/en/stable/license.html
certifi	https://certifiio.readthedocs.io/en/latest/	https://wiki.mozilla.org/index.php?title=CA/IncludedCertificates&redirect=no

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cffi	https://github.com/python-cffi	https://github.com/cffi/cffi/blob/master/COPYRIGHT
cftime	https://pypi.org/project/cftime/	https://github.com/Unidata/cftime/blob/master/LICENSE
chardet	https://github.com/chardet/chardet	https://github.com/chardet/chardet/blob/master/LICENSE
charset-normalizer	https://github.com/ousret/charset_normalizer	https://github.com/Ousret/charset_normalizer/blob/master/LICENSE
ChEMBL Structure Pipeline	https://github.com/chembl/ChEMBL_Structure_Pipeline/tree/master	https://github.com/chembl/ChEMBL_Structure_Pipeline/blob/master/LICENSE
choreographer	https://github.com/mazpie/choreographer	https://github.com/mazpie/choreographer/blob/main/LICENSE
chunkreader	https://github.com/jackc/chunkreader/v2	https://github.com/jackc/chunkreader/blob/master/LICENSE
click	https://palletsprojects.com/p/click/	https://github.com/pallets/click/blob/main/LICENSE.rst
click-completion	https://github.com/click-contrib/click-completion	https://github.com/click-contrib/click-completion/blob/master/LICENSE
click-didyoumean	https://github.com/click-contrib/click-didyoumean	https://github.com/click-contrib/click-didyoumean/blob/master/LICENSE
click-plugins	https://github.com/click-contrib/click-plugins	https://github.com/click-contrib/click-plugins/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
click-repl	https://github.com/untitaker/click-repl	https://github.com/click-contrib/click-repl/blob/master/LICENSE
cloudpickle	https://github.com/cloudpipe/cloudpickle	https://github.com/cloudpipe/cloudpickle/blob/master/LICENSE
cmake	https://cmake.org/	https://cmake.org/licensing/
colorama	https://github.com/tartley/colorama	https://github.com/tartley/colorama/blob/master/LICENSE.txt
comm	https://github.com/ipython/comm	https://github.com/ipython/comm/blob/master/LICENSE
compose	https://github.com/docker/compose	https://github.com/docker/compose/blob/v2/LICENSE
conda-content-trust	https://github.com/conda/conda-content-trust	https://github.com/conda/conda-content-trust/blob/main/LICENSE
conda-libmamba-solver	https://github.com/conda/conda-libmamba-solv	https://github.com/conda/conda-libmamba-solv/blob/main/LICENSE
conda-pack	https://conda.github.io/conda-pack/	https://github.com/conda/conda-pack/blob/main/LICENSE.txt
conda-package-handling	https://github.com/conda/conda-package-handli	https://github.com/conda/conda-package-handli/blob/main/LICENSE
conda-package-streaming	https://anaconda.org/conda-forge/conda-package-stream	https://github.com/conda/conda-package-stream/blob/main/LICENSE
conda-token	https://anaconda.org/anaconda/conda-token	https://anaconda.org/anaconda/conda-token

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
conda_package_strear	https://github.com/conda/conda-package-streamtree/main/conda_package_streaming	https://github.com/conda/conda-package-streamtab=License-1-ov-file
confuse	https://github.com/beetbox/confuse	https://github.com/beetbox/confuse/blob/main/LICENSE
contourpy	https://github.com/contourpy/contourpy	https://github.com/contourpy/contourpy/blob/main/LICENSE
cpp-expected	https://github.com/TartanLlama/expected	https://github.com/TartanLlama/expected?tab=CC0-1.0-1-ov-file#readme
cpp-peglib	https://github.com/yhirose/cpp-peglib	https://github.com/yhirose/cpp-peglib/blob/master/LICENSE
cryptography	https://github.com/pyca/cryptography	https://github.com/pyca/cryptography/blob/main/LICENSE
cssselect2	https://github.com/Kozea/cssselect2/	https://github.com/Kozea/cssselect2/blob/master/LICENSE
cuda-bindings	https://nvidia.github.io/cuda-python/cuda-bindings/latest/install.html	https://github.com/NVIDIA/cuda-python/blob/main/cuda_bindings/LICENSE
cuda-pathfinder	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/license.html
cudatoolkit	https://developer.nvidia.com/cuda-toolkit	https://docs.nvidia.com/cuda/eula/index.html
cuequivariance-ops-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cuequivariance-ops-torch-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cuequivariance-torch	https://docs.nvidia.com/cuda/cuequivariance/	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cupy-cuda113	https://cupy.dev/	https://github.com/cupy/cupy/blob/main/LICENSE
cupy-cuda11x	https://cupy.dev/	https://github.com/cupy/cupy/?tab=MIT-1-ov-file#readme
curl	https://curl.se	https://curl.se/docs/copyright.html
cycler	https://github.com/matplotlib/cycler	https://github.com/matplotlib/cycler/blob/main/LICENSE
cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl/blob/master/COPYING
Cython	https://cython.org	https://www.apache.org/licenses/LICENSE-2.0.html
d3	https://github.com/mbostock/d3	https://github.com/mbostock/d3/raw/master/LICENSE
dataclasses	https://github.com/ericvsmith/dataclasses	https://github.com/ericvsmith/dataclasses/blob/master/LICENSE.txt
ddsketch	http://github.com/datadog/sketches-py	https://github.com/DataDog/sketches-py/blob/master/LICENSE
debugpy	https://aka.ms/debugpy	https://github.com/microsoft/debugpy/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
decimal	https://github.com/shopspring/decimal	https://github.com/shopspring/decimal/blob/master/LICENSE
decorator	https://github.com/micheles/decorator	https://github.com/micheles/decorator/blob/master/LICENSE.txt
deepdiff	https://github.com/seperman/deepdiff	https://github.com/seperman/deepdiff/blob/master/LICENSE
deeptime	https://github.com/deeptime-ml/deeptime	https://github.com/deeptime-ml/deeptime/blob/main/LICENSE.txt
defusedxml	https://github.com/tiran/defusedxml	https://github.com/tiran/defusedxml/blob/main/LICENSE
dftd3	https://github.com/dftd3/tad-dftd3	https://github.com/dftd3/tad-dftd3/blob/main/LICENSE
dill	https://github.com/uqfoundation/dill	https://github.com/uqfoundation/dill/blob/master/LICENSE
distlib	https://distlib.readthedocs.io/en/latest/	https://github.com/pypa/distlib/blob/master/LICENSE.txt
distro	https://github.com/python-distro/distro	https://github.com/python-distro/distro/blob/master/LICENSE
Django	https://www.djangoproject.com/	https://github.com/django/django/blob/main/LICENSE
django-classy-tags	https://github.com/django-cms/django-classy-tags	https://github.com/django-cms/django-classy-tags/blob/master/LICENSE
django-cors-headers	https://github.com/adamchainz/django-cors-headers	https://github.com/adamchainz/django-cors-headers/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
django-csp	https://github.com/mozilla/django-csp	https://github.com/mozilla/django-csp/blob/main/LICENSE
django-extensions	https://github.com/django-extensions/django-extensions	https://github.com/django-extensions/django-extensions/blob/main/LICENSE
django-filter	https://github.com/carltongibson/django-filter/tree/master	https://github.com/carltongibson/django-filter/blob/main/LICENSE
django-redis	https://github.com/jazzband/django-redis	https://github.com/jazzband/django-redis/blob/master/LICENSE
django-taggit	https://github.com/jazzband/django-taggit	https://github.com/jazzband/django-taggit/blob/master/LICENSE
django-taggit-serializer	https://github.com/glemmaPaul/django-taggit-serializ	https://github.com/glemmaPaul/django-taggit-serializ blob/master/LICENSE
django-taggit-templatetags2	https://github.com/fizista/django-taggit-templat	https://github.com/fizista/django-taggit-templat blob/master/LICENSE.txt
djangorestframework	https://www.django-rest-framewor org/	https://github.com/encode/django-rest-framewor blob/master/LICENSE.md
dkh	https://psicode.org/psi4manual/master/dkh.html	https://github.com/psi4/dkh/blob/master/COPYING.LESSER
dm-tree	https://github.com/deepmind/tree	https://github.com/deepmind/tree/blob/master/LICENSE
docker-py	https://github.com/docker/docker-py/	https://github.com/docker/docker-py/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
docker-pycreds	https://github.com/shin-/dockerpy-creds	https://github.com/shin-/dockerpy-creds/blob/master/LICENSE
docopt	https://docopt.org	https://github.com/docopt/docopt/blob/master/LICENSE-MIT
docutils	https://docutils.sourceforge.io	https://docutils.sourceforge.io/COPYING.html
drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields/blob/master/LICENSE
DuckDB	https://duckdb.org/	https://github.com/duckdb/duckdb?tab=MIT-1-ov-file#readme
edgembar	https://rutgerslbsr.gitlab.io/fe-toolkit/edgembar/index.html	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
editdistance	https://github.com/roy-ht/editdistance	https://github.com/roy-ht/editdistance/blob/master/LICENSE
eigen	https://eigen.tuxfamily.org/index.php?title=Main_Page	https://eigen.tuxfamily.org/index.php?title=Main_Page#License
einops	https://github.com/arogozhnikov/einops	https://github.com/arogozhnikov/einops/blob/master/LICENSE
einx	https://github.com/fferflo/einx	https://github.com/fferflo/einx/blob/master/LICENSE
entrypoints	https://github.com/takluyver/entrypoints	https://github.com/takluyver/entrypoints/blob/master/LICENSE
errors	https://github.com/pkg/errors	https://github.com/pkg/errors/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
eslint-plugin	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile/-/blob/branch/default/LICENCE.rst
exceptiongroup	https://github.com/agronholm/exceptiongroup	https://github.com/agronholm/exceptiongroup/blob/main/LICENSE
execnet	https://execnet.readthedocs.io/en/latest/	https://github.com/pytest-dev/execnet/blob/master/LICENSE
executing	https://github.com/alexmojaki/executing	https://github.com/alexmojaki/executing/blob/master/LICENSE.txt
expat	https://libexpat.github.io	https://github.com/libexpat/libexpat/blob/master/expat/COPYING
fairscale	https://github.com/facebookresearch/fairscale	https://github.com/facebookresearch/fairscale/blob/main/LICENSE
fairseq	https://github.com/pytorch/fairseq	https://github.com/facebookresearch/fairseq/blob/main/LICENSE
fastjsonschema	https://github.com/horejsek/python-fastjsonschem	https://github.com/horejsek/python-fastjsonschem/blob/master/LICENSE
fastrlock	https://github.com/scoder/fastrlock	https://github.com/scoder/fastrlock/blob/master/LICENSE
fftw	https://www.fftw.org	commercial license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
filebuffer	https://github.com/mattetti/filebuffer	https://github.com/mattetti/filebuffer/blob/master/LICENSE
filelock	https://py-filelock.readthedocs.io/en/latest/index.html	https://github.com/tox-dev/py-filelock/blob/main/LICENSE
finufft	https://github.com/flatironinstitute/finufft	https://github.com/flatironinstitute/finufft/blob/master/LICENSE
Flask	https://flask.palletsprojects.com/en/2.1.x/	https://flask.palletsprojects.com/en/2.1.x/license/
flatbuffers	https://google.github.io/flatbuffers/	https://github.com/google/flatbuffers/blob/master/LICENSE.txt
flexcache	https://github.com/hgrecco/flexcache	https://github.com/hgrecco/flexcache/blob/main/LICENSE
flexparser	https://github.com/hgrecco/flexparser	https://github.com/hgrecco/flexparser/blob/main/LICENSE
flit-core	https://github.com/pypa/flit	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
FLTK	https://www.fltk.org/index.php	https://www.fltk.org/COPYING.php
fmt	https://fmt.dev/latest/index.html	https://github.com/fmtlib/fmt/blob/master/LICENSE.rst
font-awesome	https://github.com/FortAwesome/Font-Awesome	https://github.com/FortAwesome/Font-Awesome
font-ttf-dejavu-sans-mono	https://dejavu-fonts.github.io	https://dejavu-fonts.github.io/License.html
font-ttf-inconsolata	https://fonts.google.com/specimen/Inconsolata	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
font-ttf-source-code-pro	https://fonts.google.com/specimen/Source+Code+Pro	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL
font-ttf-ubuntu	https://fonts.google.com/specimen/Ubuntu	https://ubuntu.com/legal/font-licence
fontconfig	https://www.freedesktop.org/wiki/Software/fontconfig/	https://gitlab.freedesktop.org/fontconfig/fontconfig/-/blob/main/COPYING
fonts-conda-ecosystem	https://anaconda.org/conda-forge/fonts-conda-ecosyster	https://docs.conda.io/en/latest/license.html
fonts-conda-forge	https://anaconda.org/conda-forge/fonts-conda-forge/	https://creativecommons.org/licenses/by/4.0/
fonttools	https://github.com/fonttools/fonttools	https://github.com/fonttools/fonttools/blob/main/LICENSE
forcebalance	https://simtk.org/home/forcebalance OR https://github.com/leeping/forcebalance	https://github.com/leeping/forcebalance/blob/master/LICENSE
fqdn	https://github.com/ypcrts/fqdn	https://github.com/ypcrts/fqdn?tab=MPL-2.0-1-ov-file#readme
freetype	https://freetype.org	https://freetype.org/license.html
freetype-py	https://github.com/rougier/freetype-py	https://github.com/rougier/freetype-py?tab=License-1-ov-file#readme
fribidi	https://github.com/fribidi/fribidi	https://github.com/fribidi/fribidi/blob/master/COPYING
frozendict	https://github.com/Marco-Sulla/python-frozendict	https://github.com/Marco-Sulla/python-frozendict/blob/master/LICENSE.txt
frozenlist	https://github.com/aio-libs/frozenlist	https://github.com/aio-libs/frozenlist/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
fsmlite	https://github.com/tkem/fsmlite	https://github.com/tkem/fsmlite/blob/master/LICENSE
fsspec	https://github.com/fsspec/filesystem_spec	https://github.com/fsspec/filesystem_spec/blob/master/LICENSE
funcy	https://github.com/Suor/funcy	https://github.com/Suor/funcy/blob/master/LICENSE
gast	https://github.com/serge-sans-paille/gast/	https://github.com/serge-sans-paille/gast/blob/master/LICENSE
gau2grid	https://github.com/dgasmith/gau2grid	https://github.com/dgasmith/gau2grid/blob/master/LICENSE
gax-go	https://github.com/googleapis/gax-go/v2	https://github.com/googleapis/gax-go/blob/main/LICENSE
gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf/-/blob/master/COPYING
gemmi	https://gemmi.readthedocs.io/en/latest/	https://github.com/project-gemmi/gemmi/blob/master/LICENSE.txt
genproto	https://google.golang.org/genproto/googleapis	https://github.com/googleapis/go-genproto/blob/main/LICENSE
geometric	https://openbase.com/python/geometric	https://openbase.com/python/geometric
giflib	https://giflib.sourceforge.net	https://github.com/mirrorer/giflib/blob/master/COPYING
gitdb	https://github.com/gitpython-developers/gitdb	https://github.com/gitpython-developers/gitdb/blob/master/LICENSE
GitPython	https://github.com/gitpython-developers/GitPython	https://github.com/gitpython-developers/GitPython/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
glib	https://docs.gtk.org/glib/	https://gitlab.gnome.org/GNOME/glib/-/blob/main/LICENSES/LGPL-2.1-or-later.txt
GLM Library	https://github.com/g-truc/glm	https://github.com/g-truc/glm/blob/master/manual.md#section0
gls	https://github.com/jtolds/gls	https://github.com/jtolds/gls/blob/master/LICENSE
gmpy2	https://github.com/aleaxit/gmpy	https://github.com/gmpy2/gmpy2/blob/master/COPYING.LESSER
go-cleanhttp	https://github.com/hashicorp/go-cleanhttp	https://github.com/hashicorp/go-cleanhttp/blob/master/LICENSE
go-connections	https://github.com/docker/go-connections	https://github.com/docker/go-connections/blob/master/LICENSE
go-cpio	https://github.com/cavaliercoder/go-cpio	https://github.com/cavaliercoder/go-cpio/blob/master/LICENSE
go-getter	https://github.com/hashicorp/go-getter	https://github.com/hashicorp/go-getter/blob/master/LICENSE
go-homedir	https://github.com/mitchellh/go-homedir	https://github.com/mitchellh/go-homedir/blob/master/LICENSE
go-ini	https://github.com/go-ini/ini	https://github.com/go-ini/ini/blob/master/LICENSE
go-jmespath	https://github.com/jmespath/go-jmespath	https://github.com/jmespath/go-jmespath/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go-junit-report	https://github.com/jstemmer/go-junit-report	https://github.com/jstemmer/go-junit-report/blob/master/LICENSE
go-netrc	https://github.com/bgentry/go-netrc/netrc	https://github.com/bgentry/go-netrc/blob/master/LICENSE
go-ole	https://github.com/go-ole/go-ole	https://github.com/go-ole/go-ole/blob/master/LICENSE
go-pg	https://github.com/go-pg/pg	https://github.com/go-pg/pg/blob/master/LICENSE
go-redis	https://github.com/go-redis/redis/v8	https://github.com/go-redis/redis/blob/master/LICENSE
go-rendezvous	https://github.com/dgryski/go-rendezvous	https://github.com/dgryski/go-rendezvous/blob/master/LICENSE
go-safetemp	https://github.com/hashicorp/go-safetemp	https://github.com/hashicorp/go-safetemp/blob/master/LICENSE
go-sysconf	https://github.com/tklauser/go-sysconf	https://github.com/tklauser/go-sysconf/blob/master/LICENSE
go-testing-interface	https://github.com/mitchellh/go-testing-interface	https://github.com/mitchellh/go-testing-interface/blob/master/LICENSE
go-units	https://github.com/docker/go-units	https://github.com/docker/go-units/blob/master/LICENSE
go-version	https://github.com/hashicorp/go-version	https://github.com/hashicorp/go-version/blob/master/LICENSE
go-zglob	https://github.com/mattn/go-zglob	https://github.com/mattn/go-zglob/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go.opencensus	https://go.opencensus.io	https://github.com/census-instrumentatio opencensus-go/blob/master/LICENSE
gobrake.v2	https://gopkg.in/airbrake/gobrake.v2	https://github.com/airbrake/gobrake/blob/v2.0.9/LICENSE
goconvey	https://github.com/smartystreets/goconvey	https://github.com/smartystreets/goconvey/blob/master/LICENSE.md
golden-layout	https://github.com/deepstreamIO/golden-layout	https://github.com/deepstreamIO/golden-layout/raw/master/LICENSE
google-auth	https://google-auth.readthedocs.io/en/master/	https://github.com/GoogleCloudPlatform google-auth-library-p blob/master/LICENSE
google-auth-oauthlib	https://github.com/googleapis/google-auth-library-p	https://github.com/googleapis/google-auth-library-p blob/main/LICENSE
google-cloud-go	https://cloud.google.com/go	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-cloud-go/storage	https://cloud.google.com/go/storage	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-pasta	https://github.com/google/pasta	https://github.com/google/pasta/blob/master/LICENSE
google.golang.org/api	https://google.golang.org/api	https://github.com/googleapis/google-api-go-client/blob/main/LICENSE
gopsutil	https://github.com/shirou/gopsutil	https://github.com/shirou/gopsutil/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
gorilla websocket	https://github.com/gorilla/websocket	https://github.com/gorilla/websocket/blob/master/LICENSE
gpytorch	https://gpytorch.ai	https://github.com/cornellius-gp/gpytorch?tab=MIT-1-ov-file#readme
graphite2	https://github.com/silnrsi/graphite	https://github.com/silnrsi/graphite/blob/master/LICENSE
graphviz	https://graphviz.org/	https://graphviz.org/license/
greenlet	https://greenlet.readthedocs.io/en/latest/	https://github.com/python-greenlet/greenlet/blob/master/LICENSE
groupcache	https://github.com/golang/groupcache	https://github.com/golang/groupcache/blob/master/LICENSE
grpc	https://google.golang.org/grpc	https://github.com/grpc/grpc-go/blob/master/LICENSE
grpc-cpp	https://github.com/grpc/grpc	https://github.com/grpc/grpc/blob/master/LICENSE
grpcio	https://github.com/grpc/grpc.io/blob/main/LICENSE	https://github.com/grpc/grpc.io/blob/main/LICENSE
gtk2	https://gitlab.gnome.org/GNOME/gtk	https://gitlab.gnome.org/GNOME/gtk/-/blob/main/COPYING
gts	https://sourceforge.net/projects/gts/	https://www.gnu.org/licenses/lgpl-3.0.html
h11	https://github.com/python-hyper/h11	https://github.com/python-hyper/h11/blob/master/LICENSE.txt
h2	https://github.com/python-hyper/h2	https://github.com/python-hyper/h2?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
h5py	https://www.h5py.org	https://docs.h5py.org/en/stable/licenses.html
harfbuzz	https://github.com/harfbuzz/harfbuzz	https://github.com/harfbuzz/harfbuzz/blob/main/COPYING
hdbscan	https://hdbscan.readthedocs.io/en/latest/	https://github.com/scikit-learn-contrib/hdbscan/blob/master/LICENSE
hdf4	https://www.hdfgroup.org/solutions/hdf4/	https://github.com/HDFGroup/hdf4/blob/master/COPYING
hdf5	https://www.hdfgroup.org/solutions/hdf5/	https://github.com/HDFGroup/hdf5/blob/master/COPYING
he	https://github.com/mathiasbynens/he	https://github.com/mathiasbynens/he/raw/master/LICENSE-MIT.txt
hpack	https://github.com/python-hyper/hpack	https://github.com/python-hyper/hpack?tab=MIT-1-ov-file#readme
html-loader	https://github.com/webpack-contrib/html-loader	https://github.com/webpack-contrib/html-loader/raw/master/LICENSE
html5lib	https://github.com/html5lib/html5lib-python	https://github.com/html5lib/html5lib-python/blob/master/LICENSE
htslib	https://www.htslib.org	https://github.com/samtools/htslib/blob/develop/LICENSE
httpcore	https://www.encode.io/httpcore/	https://github.com/encode/httpcore/blob/master/LICENSE.md
httpx	https://github.com/encode/httpx	https://github.com/encode/httpx/blob/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
humanize	https://github.com/jmoiron/humanize	https://github.com/jmoiron/humanize/blob/master/LICENCE
hydra-core	https://github.com/facebookresearch/hydra	https://github.com/facebookresearch/hydra/blob/main/LICENSE
hyperframe	https://github.com/python-hyper/hyperframe	https://github.com/python-hyper/hyperframe?tab=MIT-1-ov-file#readme
icu	https://github.com/unicode-org/icu	https://github.com/unicode-org/icu/blob/main/icu4c/LICENSE
idna	https://github.com/kjd/idna	https://github.com/kjd/idna/blob/master/LICENSE.md
ihm	https://github.com/ihmwg/python-ihm	https://github.com/ihmwg/python-ihm/blob/main/LICENSE
imageio	https://github.com/imageio/imageio	https://github.com/imageio/imageio/blob/master/LICENSE
imagesize	https://github.com/shibukawa/imagesize_py	https://github.com/shibukawa/imagesize_py/blob/master/LICENSE.rst
ImmuneBuilder	https://github.com/oxpig/ImmuneBuilder/tree/main	https://github.com/oxpig/ImmuneBuilder/blob/main/LICENSE
importlib-metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE
importlib_metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
importlib_resources	https://github.com/python/importlib_resources	https://github.com/python/importlib_resources/blob/main/LICENSE
InChI	https://iupac.org/who-we-are/divisions/division-details/inchi/	https://iupac.org/wp-content/uploads/2021/06/iupac-inchi-license_2020.pdf
inflection	https://github.com/jinzhu/inflection	https://github.com/jinzhu/inflection/blob/master/LICENSE
ini.v1	https://gopkg.in/ini.v1	https://github.com/go-ini/ini/blob/v1.66.2/LICENSE
iniconfig	https://github.com/pytest-dev/iniconfig	https://github.com/pytest-dev/iniconfig/blob/main/LICENSE
innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill/raw/master/LICENSE
intel-openmp	https://github.com/hermitcore/libomp_oss	https://github.com/hermitcore/libomp_oss/blob/hermit/LICENSE.txt
ipykernel	https://ipython.org	https://ipython.org
ipython	https://ipython.org	https://github.com/ipython/ipython/blob/master/LICENSE
ipython-genutils	http://ipython.org	https://github.com/ipython/ipython_genutils/blob/master/COPYING.md
ipython_pygments_le	https://github.com/ipython/ipython-pygments-lex	https://github.com/ipython/ipython-pygments-lex blob/main/LICENSE
ipywidgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
isodate	https://github.com/gweis/isodate/	https://github.com/gweis/isodate/blob/master/LICENSE
isoduration	https://github.com/bolsote/isoduration	https://github.com/bolsote/isoduration?tab=ISC-1-ov-file#readme
itsdangerous	https://palletsprojects.com/p/itsdangerous/	https://itsdangerous.palletsprojects.com/en/2.1.x/license/
jax	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxlib	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxtyping	https://github.com/patrick-kidger/jaxtyping	https://github.com/patrick-kidger/jaxtyping?tab=License-1-ov-file#readme
jedi	https://jedi.readthedocs.io/en/latest/	https://github.com/davidhalter/jedi/blob/master/LICENSE.txt
Jinja2	https://palletsprojects.com/p/jinja/	https://github.com/pallets/jinja/blob/main/LICENSE.rst
jmespath	https://github.com/jmespath/jmespath.py	https://github.com/jmespath/jmespath.py/blob/develop/LICENSE.txt
joblib	https://joblib.readthedocs.io/en/latest/	https://github.com/joblib/joblib/blob/master/LICENSE.txt
jpeg	https://www.ijg.org	https://www.ijg.org/files/README
json5	https://github.com/dpranke/pyjson5	https://github.com/dpranke/pyjson5/blob/main/LICENSE
jsonfield	https://github.com/dmkoch/django-jsonfield/	https://github.com/rpkilby/jsonfield/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jsonpatch	https://github.com/stefankoegl/python-json-patch	https://github.com/stefankoegl/python-json-patch/blob/master/LICENSE
jsonpickle	https://github.com/jsonpickle/jsonpickle	https://github.com/jsonpickle/jsonpickle/blob/main/LICENSE
jsonpointer	https://github.com/stefankoegl/python-json-pointer	https://github.com/stefankoegl/python-json-pointer/blob/master/LICENSE.txt
jsonschema	https://github.com/python-jsonschema/jsonschema	https://github.com/python-jsonschema/jsonschema/blob/master/COPYING
jsonschema-specifications	https://github.com/python-jsonschema/jsonschema-specificat	https://github.com/python-jsonschema/jsonschema-specificat blob/main/COPYING
jstat	https://github.com/jstat/jstat	https://www.opensource.org/licenses/mit-license.php
jupyter-client	https://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/COPYING.md
jupyter-core	https://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/COPYING.md
jupyter-events	https://github.com/jupyter/jupyter_events	https://github.com/jupyter/jupyter_events/blob/main/LICENSE
jupyter-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp
jupyter-serverhttp://jupyter.or	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE	
jupyter_client	http://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jupyter_core	http://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/LICENSE
jupyter_server	https://github.com/jupyter-server/jupyter_server	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE
jupyter_server_termin	https://github.com/jupyter-server/jupyter_server_terminals	https://github.com/jupyter-server/jupyter_server_terminals/blob/main/LICENSE
jupyterlab	https://github.com/jupyterlab/jupyterlab	https://github.com/jupyterlab/jupyterlab/blob/main/LICENSE
jupyterlab-pygments	http://jupyter.org	https://github.com/jupyterlab/jupyterlab_pygments/blob/main/LICENSE
jupyterlab-widgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/master/LICENSE
jupyter-lab_pygments	https://github.com/jupyterlab/jupyterlab_pygments	https://github.com/jupyterlab/jupyterlab_pygments?tab=BSD-3-Clause-1-ov-file#readme
jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server/blob/main/LICENSE
jupyterlab_widgets	https://github.com/jupyter-widgets/ipywidgets	https://github.com/jupyter-widgets/ipywidgets?tab=BSD-3-Clause-1-ov-file#readme
kaleido	https://github.com/plotly/Kaleido	https://github.com/plotly/Kaleido/blob/master/LICENSE.txt
keras	https://github.com/keras-team/keras	https://github.com/keras-team/keras/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
Keras-Preprocessing	https://github.com/keras-team/keras-preprocessing	https://github.com/keras-team/keras-preprocessing/blob/master/LICENSE
keras-tuner	https://github.com/keras-team/keras-tuner	https://github.com/keras-team/keras-tuner/blob/master/LICENSE
keyring	https://github.com/jaraco/keyring	https://github.com/jaraco/keyring/blob/main/LICENSE
keyutils	https://github.com/sassoftware/python-keyutils	https://github.com/sassoftware/python-keyutils/blob/master/LICENSE
kiwisolver	https://kiwisolver.readthedocs.io/en/latest/	https://github.com/nucleic/kiwi/blob/main/LICENSE
kombu	https://kombu.readthedocs.io	https://github.com/celery/kombu/blob/master/LICENSE
krb5	https://web.mit.edu/kerberos/	https://github.com/krb5/krb5/blob/master/src/lib/gssapi/LICENSE
kt-legacy	https://github.com/haifeng-jin/kt-legacy	https://github.com/haifeng-jin/kt-legacy/blob/main/LICENSE
lark	https://github.com/lark-parser/lark	https://github.com/lark-parser/lark?tab=MIT-1-ov-file#readme
lazy_loader	https://github.com/scientific-python/lazy_loader	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
lcms2	https://www.littlecms.com	https://github.com/sk1project/python-lcms2/blob/master/LICENSE
legacy-cgi	https://github.com/jackrosenthal/legacy-cgi	https://github.com/jackrosenthal/legacy-cgi?tab=License-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lerc	https://github.com/Esri/lerc	https://github.com/Esri/lerc/blob/master/LICENSE
libarchive	http://www.libarchive.org	https://raw.githubusercontent.com/libarchive/libarchive/master/COPYING
libblas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
libbrotlicommon	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlidec	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlienc	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libcblas	https://anaconda.org/conda-forge/libcblas	N/A
libclang	https://github.com/sighingnow/libclang	https://github.com/sighingnow/libclang/blob/master/LICENSE.TXT
libcurl	https://curl.se/libcurl/	https://curl.se/docs/copyright.html
libcxx	https://github.com/llvm-mirror/libcxx	https://github.com/llvm-mirror/libcxx/blob/master/LICENSE.TXT
libdb	https://www.oracle.com/technology/software/products/berkeley-db/index.html	https://github.com/berkeleydb/libdb/blob/master/LEGAL
libdeflate	https://github.com/ebiggers/libdeflate	https://github.com/ebiggers/libdeflate/blob/master/COPYING
libedit	https://thrysoee.dk/editline/	http://www.netbsd.org/Goals/redistribution.html
libev	https://software.schmorp.de/pkg/libev.html	https://software.schmorp.de/pkg/libev.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libexpat	https://libexpat.github.io/	https://github.com/libexpat/libexpat?tab=MIT-1-ov-file#readme
libffi	https://github.com/libffi/libffi	https://github.com/libffi/libffi/blob/master/LICENSE
libgcc	https://gcc.gnu.org/onlinedocs/gccint/Libgcc.html	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libgcrypt	https://gnupg.org/software/index.html	https://git.gnupg.org/cgi-bin/gitweb.cgi?p=libgcrypt.git;a=blob;f=COPYING.LIB;hb=HEAD
libgd	https://libgd.github.io	https://libgd.github.io/manuals/2.2.1/files/license-txt.html
libglib	https://github.com/PolMine/libglib	https://github.com/PolMine/libglib/blob/master/include/glib-2.0/glib.h
libiconv	https://www.gnu.org/software/libiconv/	https://www.gnu.org/software/libiconv/#downloading
libint	https://tinyurl.com/yvw97wbw	https://tinyurl.com/yvw97wbw
liblapack	http://www.netlib.org/lapack/	https://www.netlib.org/lapack/LICENSE.txt
liblapacke	https://anaconda.org/conda-forge/liblapacke	https://www.netlib.org/lapack/LICENSE.txt
liblzma	https://tukaani.org/xz/	https://git.tukaani.org/?p=xz.git;a=blob;f=COPYING.0BSD
libmamba	https://medium.com/@QuantStack/open-software-packag	https://github.com/mamba-org/mamba/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libmambapy	https://www.anaconda.com/blog/a-faster-conda-for-a-g	https://github.com/conda/conda-libmamba-solv/blob/main/LICENSE
libnetcdf	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
libnghttp2	https://github.com/nghttp2/nghttp2	https://github.com/nghttp2/nghttp2/blob/master/COPYING
libnsl	https://github.com/thkukuk/libnsl	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libopenblas	https://www.openblas.net/	https://github.com/xianyi/OpenBLAS/blob/develop/LICENSE
libpng	https://www.libpng.org/pub/png/libpng.html	https://www.libpng.org/pub/png/src/libpng-LICENSE.txt
libpq	https://www.postgresql.org/	https://www.postgresql.org/about/licence/
librsvg	https://gitlab.gnome.org/GNOME/librsvg	https://gitlab.gnome.org/GNOME/librsvg/-/blob/main/COPYING.LIB
libsolv	https://github.com/openSUSE/libsolv	https://github.com/openSUSE/libsolv/blob/master/LICENSE.BSD
libsqlite	https://github.com/mackyle/sqlite	https://github.com/mackyle/sqlite?tab=License-1-ov-file#readme
libssh2	https://github.com/libssh2/libssh2	https://github.com/libssh2/libssh2/blob/master/COPYING
libstdcxx	https://gcc.gnu.org/	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libtiff	https://www.libtiff.org/	https://www.libtiff.org/misc.html
libtrust	https://github.com/docker/libtrust	https://github.com/docker/libtrust/blob/master/LICENSE
libuuid	https://sourceforge.net/projects/libuuid/	https://sourceforge.net/p/libuuid/code/ci/master/tree/COPYING
libuv	https://github.com/libuv/libuv	https://github.com/libuv/libuv/blob/v1.x/LICENSE
libwebp	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libwebp-base	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libxc	https://www.tddft.org/programs/libxc/	https://gitlab.com/libxc/libxc/-/blob/master/COPYING
libxcb	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-libxcb/blob/master/COPYING
libxcrypt	https://github.com/besser82/libxcrypt	https://github.com/besser82/libxcrypt?tab=LGPL-2.1-1-ov-file#readme
libxml2	https://git.gnome.org/browse/libxml2/	https://gitlab.gnome.org/GNOME/libxml2/-/blob/master/Copyright
libxmlsec1	https://github.com/lsh123/xmlsec	https://github.com/lsh123/xmlsec/blob/master/README
libxslt	https://gitlab.gnome.org/GNOME/libxslt/-/wikis/home	https://gitlab.gnome.org/GNOME/libxslt/-/blob/master/Copyright

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libzlib	https://zlib.net	https://zlib.net/zlib_license.html
lightning-utilities	https://github.com/Lightning-AI/utilities	https://github.com/Lightning-AI/utilities?tab=Apache-2.0-1-ov-file#readme
lime	https://github.com/marcotcr/lime	https://github.com/marcotcr/lime/blob/master/LICENSE
linear-operator	https://github.com/cornellius-gp/linear_operator	https://github.com/cornellius-gp/linear_operator?tab=MIT-1-ov-file#readme
lit	http://llvm.org	https://releases.llvm.org/14.0.0/LICENSE.TXT
llvm-openmp	https://github.com/llvm-mirror/openmp	https://github.com/llvm-mirror/openmp/blob/master/LICENSE.txt
llvmlite	http://llvmlite.readthedocs.io	https://github.com/numba/llvmlite/blob/main/LICENSE
loader-utils	https://github.com/webpack/loader-utils	https://github.com/webpack/loader-utils/raw/master/LICENSE
logistro	https://github.com/geopozo/logistro	https://github.com/geopozo/logistro?tab=MIT-1-ov-file#readme
logomaker	https://logomaker.readthedocs.io/en/latest/	https://github.com/jbkinney/logomaker/blob/master/LICENSE
logrus	https://github.com/sirupsen/logrus	https://github.com/sirupsen/logrus/blob/master/LICENSE
logrus-airbrake-hook.v2	https://gopkg.in/gemnasium/logrus-airbrake-hook.v2	https://github.com/gemnasium/logrus-airbrake-hook/blob/v2.1.2/LICENSE
lxml	https://lxml.de	https://lxml.de/index.html#license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lz4-c	https://www.lz4.org/	https://github.com/lz4/lz4/blob/dev/LICENSE
lzo	https://www.oberhumer.com/opensource/lzo/	https://www.oberhumer.com/opensource/gpl.html
mamba	https://github.com/mamba-org/mamba	https://github.com/mamba-org/mamba?tab=BSD-3-Clause-1-ov-f readme
markdown	https://github.com/evilstreak/markdown-js	https://www.opensource.org/licenses/mit-license.php
markdown-it-py	https://github.com/executablebooks/markdown-it-py	https://github.com/executablebooks/markdown-it-py/blob/master/LICENSE
markupsafe	https://github.com/pallets/markupsafe/tree/main/src/markupsafe	https://github.com/pallets/markupsafe/tree/main?tab=BSD-3-Clause-1-ov-f
MarkupSafe	https://palletsprojects.com/p/markupsafe/	https://github.com/pallets/markupsafe/blob/main/LICENSE.rst
mashumaro	https://github.com/Fatal1ty/mashumaro	https://github.com/Fatal1ty/mashumaro
matplotlib	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-base	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-inline	https://github.com/ipython/matplotlib-inline	https://github.com/ipython/matplotlib-inline/blob/master/LICENSE
mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mda_xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib?tab=License-1-ov-file#readme
mdtraj	https://www.mdtraj.org/	https://www.mdtraj.org/1.9.8.dev0/index.html#license
mdurl	https://github.com/markdown-it/mdurl	https://github.com/markdown-it/mdurl/blob/master/LICENSE
menuinst	https://github.com/conda/menuinst	https://github.com/conda/menuinst/blob/main/LICENSE.txt
mergo	https://github.com/imdario/mergo	https://github.com/imdario/mergo/blob/master/LICENSE
mistune	https://github.com/lepture/mistune	https://github.com/lepture/mistune/blob/master/LICENSE
mkl	https://github.com/rust-math/intel-mkl-src	https://github.com/rust-math/intel-mkl-src/blob/master/License.txt
mkl-fft	https://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft/blob/master/LICENSE.txt
mkl-random	https://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random/blob/master/LICENSE.txt
mkl-service	https://github.com/IntelPython/mkl-service	https://github.com/IntelPython/mkl-service/blob/master/LICENSE.txt
mkl_fft	http://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft?tab=BSD-3-Clause-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mkl_random	http://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random?tab=BSD-3-Clause-1-ov-f readme
ml-dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
ml_dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
modelcif	https://github.com/ihmwg/python-modelcif	https://github.com/ihmwg/python-modelcif/blob/main/LICENSE
molecupy	https://molecupy.readthedocs.io/en/latest/	https://github.com/samirelanduk/atomium/blob/master/LICENSE
moment	https://github.com/moment/moment	https://github.com/moment/moment/raw/master/LICENSE
moment-precise-range-plugin	https://github.com/codebox/moment-precise-range	https://github.com/codebox/moment-precise-range raw/master/LICENSE.md
more-itertools	https://github.com/more-itertools/more-itertools	https://github.com/more-itertools/more-itertools/blob/master/LICENSE
mpiplus	https://github.com/choderalab/mpiplus	https://github.com/choderalab/mpiplus/blob/main/LICENSE
mpmath	http://mpmath.org/	https://github.com/mpmath/mpmath/blob/master/LICENSE
mrcfile	https://github.com/ccpem/mrcfile	https://github.com/ccpem/mrcfile/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
msgpack	https://msgpack.org/	https://github.com/msgpack/msgpack-python/blob/main/COPYING
multidict	https://github.com/aio-libs/multidict	https://github.com/aio-libs/multidict/blob/master/LICENSE
multierr	https://go.uber.org/multierr	https://github.com/uber-go/multierr/blob/master/LICENSE.txt
multiprocess	https://github.com/uqfoundation/multiprocess	https://github.com/uqfoundation/multiprocess/blob/master/LICENSE
munkres	https://software.clapper.org/munkres/	https://github.com/bmc/munkres/blob/master/LICENSE.md
myesui uuid	https://github.com/myesui/uuid	https://pkg.go.dev/github.com/myesui/uuid?tab=licenses
namex	https://github.com/bcgov/namex	https://github.com/bcgov/namex/blob/main/LICENSE
narwhals	https://github.com/narwhals-dev/narwhals	https://github.com/narwhals-dev/narwhals?tab=MIT-1-ov-file#readme
nbclassic	http://jupyter.org	https://github.com/jupyter/nbclassic/blob/main/LICENSE
nbclient	https://jupyter.org	https://github.com/jupyter/nbclient/blob/main/LICENSE
nbconvert	https://jupyter.org	https://github.com/jupyter/nbconvert/blob/main/LICENSE
nbformat	http://jupyter.org	https://github.com/jupyter/nbformat/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ncurses	https://invisible-island.net/ncurses/	https://invisible-island.net/ncurses/ncurses-license.html
ndfes	https://rutgerslbsr.gitlab.io/fe-toolkit/	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
ndindex	https://quansight-labs.github.io/ndindex/	https://github.com/Quansight-Labs/ndindex?tab=MIT-1-ov-file#readme
nest-asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio/blob/master/LICENSE
nest_asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio?tab=BSD-2-Clause-1-ov-f readme
netcdf-fortran	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
netCDF4	http://github.com/Unidata/netcdf4-python	https://github.com/Unidata/netcdf4-python/blob/master/LICENSE
nettle	https://git.lysator.liu.se/nettle/nettle	https://git.lysator.liu.se/nettle/nettle/-/blob/master/AUTHORS?ref_type=heads
networkx	https://networkx.org	https://github.com/networkx/networkx/blob/main/LICENSE.txt
nfpm	https://github.com/goreleaser/nfpm	https://github.com/goreleaser/nfpm/blob/master/LICENSE.md
ng-tags-input	https://github.com/mbenford/ngTagsInput	https://github.com/mbenford/ngTagsInput
ng-toast	https://github.com/tameraydin/ngToast	https://github.com/tameraydin/ngToast

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ngdraggable	https://github.com/fatlinesofcode/ngDraggable	https://github.com/fatlinesofcode/ngDraggable/raw/master/LICENSE.txt
ngVue	https://github.com/ngVue/ngVue	https://github.com/ngVue/ngVue/raw/master/LICENSE
nh3	https://nh3.readthedocs.io/en/latest/	https://github.com/messense/nh3/blob/main/LICENSE
nlohmann_json	https://github.com/nlohmann/json	https://github.com/nlohmann/json?tab=MIT-1-ov-file#readme
nlopt	https://nlopt.readthedocs.io/en/latest/NLopt_Python_Reference/	https://nlopt.readthedocs.io/en/latest/NLopt_License_and_Copyright/
nodejs	https://nodejs.org/en/	https://raw.githubusercontent.com/nodejs/node/master/LICENSE
nomkl	https://github.com/conda-forge/nomkl-feedstock	https://github.com/conda-forge/nomkl-feedstock/blob/main/LICENSE.txt
notebook	http://jupyter.org	https://github.com/jupyter/notebook/blob/main/LICENSE
notebook-shim	https://github.com/jupyter/notebook_shim	https://github.com/jupyter/notebook_shim
notebook_shim	http://jupyter.org	https://github.com/jupyter/notebook_shim/blob/main/LICENSE
numba	https://numba.pydata.org	https://github.com/numba/numba/blob/main/LICENSE
numcpus	https://github.com/tklauser/numcpus	https://github.com/tklauser/numcpus/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
numexpr	https://github.com/pydata/numexpr	https://github.com/pydata/numexpr/blob/master/LICENSE.txt
numpy	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpy-base	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpydoc	https://numpydoc.readthedocs.io/en/latest/index.html	https://github.com/numpy/numpydoc/blob/main/LICENSE.txt
nvidia-cublas-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cublas-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cudnn-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html
nvidia-cudnn-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-cuEquivariance	https://github.com/NVIDIA/cuEquivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
nvidia-cufft-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufft-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufile-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html#
nvidia-curand-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-curand-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparselt-cu12	https://docs.nvidia.com/cuda/cusparselt/index.html	https://docs.nvidia.com/cuda/cusparselt/license.html
nvidia-ml-py	https://pypi.org/project/nvidia-ml-py/	https://pypi.org/project/nvidia-ml-py/
nvidia-nccl-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nccl-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nvjitlink-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-nvshmem-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
Oat++	https://oatpp.io/	https://github.com/oatpp/oatpp/blob/master/LICENSE
oauthlib	https://github.com/oauthlib/oauthlib	https://github.com/oauthlib/oauthlib/blob/master/LICENSE
ocl-icd	https://github.com/OCL-dev/ocl-icd	https://github.com/OCL-dev/ocl-icd/blob/master/COPYING
ocl-icd-system	https://github.com/conda-forge/ocl-icd-system-feedst	https://github.com/conda-forge/ocl-icd-system-feedst blob/main/LICENSE.txt
olefile	https://www.decalage.info/python/olefileio	https://olefile.readthedocs.io/en/latest/License.html
omegaconf	https://github.com/omry/omegaconf	https://github.com/omry/omegaconf/blob/master/LICENSE
OmegaFold	https://github.com/HeliXonProtein/OmegaFold/tree/main	https://github.com/HeliXonProtein/OmegaFold/blob/main/LICENSE
omnicanvas	https://omnicanvas.readthedocs.io/en/latest/	https://github.com/samirelanduk/omnicanvas/blob/master/LICENSE
OpenFF	https://openforcefield.org/	https://github.com/openforcefield/openff-forcefields
openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openff-forcefields	https://openforcefield.org	https://github.com/openforcefield/openff-forcefields/blob/master/LICENSE
openff-fragmenter	https://github.com/openforcefield/openff-fragmenter	https://github.com/openforcefield/openff-fragmenter/blob/main/LICENSE
openff-interchange	https://github.com/openforcefield/openff-interchange	https://github.com/openforcefield/openff-interchange/blob/main/LICENSE
openff-models	https://github.com/openforcefield/openff-models	https://github.com/openforcefield/openff-models/blob/main/LICENSE
openff-nagl	https://docs.openforcefield.org/projects/nagl/en/latest/?badge=latest	https://github.com/openforcefield/openff-nagl/blob/main/LICENSE
openff-nagl-models	https://github.com/openforcefield/openff-nagl-models	https://github.com/openforcefield/openff-nagl-models/blob/main/LICENSE
openff-toolkit	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-toolkit-base	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-units	https://github.com/openforcefield/openff-units	https://github.com/openforcefield/openff-units/blob/main/LICENSE
openff-utilities	https://github.com/openforcefield/openff-utilities	https://github.com/openforcefield/openff-utilities/blob/main/LICENSE
openjpeg	https://github.com/uclouvain/openjpeg	https://github.com/uclouvain/openjpeg/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openldap	https://www.openldap.org/software/repo.html	https://git.openldap.org/openldap/openldap/-/blob/master/LICENSE
OpenMM	https://openmm.org	https://docs.openmm.org/7.1.0/userguide/library.html#license
openmmtools	https://github.com/choderalab/openmmtools	https://github.com/choderalab/openmmtools/blob/main/LICENSE
openmoltools	https://github.com/choderalab/openmoltools	https://github.com/choderalab/openmoltools/blob/master/LICENSE
openpyxl	https://openpyxl.readthedocs.io/en/stable/	https://foss.heptapod.net/openpyxl/openpyxl/-/blob/branch/3.0/LICENCE.rst
openssl	https://www.openssl.org	https://github.com/openssl/openssl/blob/master/LICENSE.txt
opt-einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/master/LICENSE
opt_einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/main/LICENSE
OptKing	https://github.com/psi-rking/optking	https://github.com/psi-rking/optking/blob/master/LICENSE
optree	https://optree.readthedocs.io/en/latest/	https://github.com/metaopt/optree/blob/main/LICENSE
oscrypto	https://github.com/wbond/oscrypto	https://github.com/wbond/oscrypto/blob/master/LICENSE
overrides	https://github.com/mkorpela/overrides	https://github.com/mkorpela/overrides/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
packaging	https://github.com/pypa/packaging	https://github.com/pypa/packaging/blob/main/LICENSE
packmol	https://leandro.iqm.unicamp.br/m3g/packmol/home.shtml	https://github.com/m3g/packmol/blob/master/LICENSE
packmol_memgen	http://cpclab.uni-duesseldorf.de/	https://gitlab.com/cbjh/memgen/py-memgen/-/blob/main/LICENSE
pandas	https://pandas.pydata.org	https://github.com/pandas-dev/pandas/blob/main/LICENSE
pandocfilters	http://github.com/jgm/pandocfilters	https://github.com/jgm/pandocfilters/blob/master/LICENSE
panedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pango	https://pango.gnome.org	https://gitlab.gnome.org/GNOME/pango/-/blob/main/COPYING
ParmEd	https://parmed.github.io/ParmEd/html/index.html	https://github.com/ParmEd/ParmEd#license
parser	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
parso	https://parso.readthedocs.io/en/latest/	https://github.com/davidhalter/parso/blob/master/LICENSE.txt
pathos	https://github.com/uqfoundation/pathos	https://github.com/uqfoundation/pathos/blob/master/LICENSE
patsy	https://patsy.readthedocs.io/en/latest/	https://github.com/pydata/patsy/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pbkdf2	https://golang.org/x/crypto/pbkdf2	https://cs.opensource.google/go/x/crypto/+/master:LICENSE
pbr	https://docs.openstack.org/pbr/latest/	https://opendev.org/openstack/pbr/src/branch/master/LICENSE
pcmsolver	https://pcmsolver.readthedocs.io/en/stable/	https://github.com/PCMSolver/pcmsolver/blob/master/LICENSE
pcre	https://www.pcre.org	https://www.pcre.org/licence.txt
pcre2	https://www.pcre.org	https://github.com/PCRE2Project/pcre2/blob/master/LICENCE
pdb4amber	https://github.com/Amber-MD/pdb4amber	https://github.com/Amber-MD/pdb4amber/blob/master/LICENSE
pdbfixer	https://github.com/openmm/pdbfixer	https://github.com/openmm/pdbfixer/blob/master/LICENSE
pexpect	https://pexpect.readthedocs.io/	https://github.com/pexpect/pexpect/blob/master/LICENSE
pgconn	https://github.com/jackc/pgconn	https://github.com/jackc/pgconn/blob/master/LICENSE
pgio	https://github.com/jackc/pgio	https://github.com/jackc/pgio/blob/master/LICENSE
pgpassfile	https://github.com/jackc/pgpassfile	https://github.com/jackc/pgpassfile/blob/master/LICENSE
pgproto3	https://github.com/jackc/pgproto3/v2	https://github.com/jackc/pgproto3/blob/master/LICENSE
pgtype	https://github.com/jackc/pgtype	https://github.com/jackc/pgtype/blob/master/LICENSE
pgx	https://github.com/jackc/pgx/v4	https://github.com/jackc/pgx/blob/v4.0.1/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
phonopy	https://github.com/phonopy/phono3py	https://github.com/phonopy/phono3py/blob/develop/LICENSE
pickleshare	https://github.com/pickleshare/pickleshare	https://github.com/pickleshare/pickleshare/blob/master/LICENSE
Pillow	https://python-pillow.org	https://github.com/python-pillow/Pillow/blob/main/LICENSE
Pint	https://pint.readthedocs.io/en/stable/	https://github.com/hgrecco/pint/blob/master/LICENSE
pip	https://pip.pypa.io/	https://github.com/pypa/pip/blob/main/LICENSE.txt
pip-licenses	https://github.com/raimon49/pip-licenses	https://github.com/raimon49/pip-licenses/blob/master/LICENSE
pixman	https://pixman.org	https://cgit.freedesktop.org/pixman/tree/COPYING
pkginfo	https://launchpad.net/pkginfo	https://bazaar.launchpad.net/~tseaver/pkginfo/trunk/view/head:/LICENSE.txt
pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
pkgutil_resolve_name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
platformdirs	https://github.com/platformdirs/platformdirs	https://github.com/platformdirs/platformdirs/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
plotly	https://plotly.com/python/	https://github.com/plotly/plotly.py/blob/master/LICENSE.txt
plotly-orca	https://github.com/plotly/orca	https://github.com/plotly/orca/blob/master/LICENSE
plotly.js	https://github.com/plotly/plotly.js	https://github.com/plotly/plotly.js/raw/master/LICENSE
pluggy	https://pluggy.readthedocs.io/en/stable/index.html	https://github.com/pytest-dev/pluggy/blob/main/LICENSE
pooch	https://github.com/fatiando/pooch	https://github.com/fatiando/pooch/blob/main/LICENSE.txt
portalocker	https://github.com/WoLpH/portalocker	https://github.com/wolph/portalocker/blob/develop/LICENSE
pox	https://github.com/uqfoundation/pox	https://github.com/uqfoundation/pox/blob/master/LICENSE
ppft	https://github.com/uqfoundation/ppft	https://github.com/uqfoundation/ppft/blob/master/LICENSE
pq	https://github.com/lib/pq	https://github.com/lib/pq/blob/master/LICENSE.md
ProDy	http://www.csb.pitt.edu/ProDy	https://github.com/prody/ProDy/blob/master/LICENSE.rst
prometheus-client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prometheus_client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prompt-toolkit	https://python-prompt-toolkit.readthedocs.io/en/stable/	https://github.com/prompt-toolkit/python-prompt-toolkit/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
prompt_toolkit	https://github.com/prompt-toolkit/python-prompt-toolki	https://github.com/prompt-toolkit/python-prompt-toolki tab=BSD-3-Clause-1-ov-f readme
propcache	https://github.com/aio-libs/propcache	https://github.com/aio-libs/propcache?tab=Apache-2.0-1-ov-file#readme
protobuf	https://google.golang.org/protobuf	https://github.com/protocolbuffers/protobuf-go/blob/master/LICENSE
psi4	https://psicode.org	https://github.com/psi4/psi4/blob/master/COPYING.LESSER
psutil	https://psutil.readthedocs.io/en/latest/	https://github.com/giampaolo/psutil/blob/master/LICENSE
psycopg2	https://psycopg.org/	https://github.com/psycopg/psycopg2/blob/master/LICENSE
PTable	https://github.com/kxxoling/PTable	https://github.com/kxxoling/PTable/blob/master/COPYING
pthread-stubs	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-pthread-stubs/blob/master/COPYING
ptyprocess	https://ptyprocess.readthedocs.io/en/latest/	https://github.com/pexpect/ptyprocess/blob/master/LICENSE
pure-eval	http://github.com/alexmojaki/pure_eval	http://github.com/alexmojaki/pure_eval/blob/master/LICENSE.txt
pure_eval	http://github.com/alexmojaki/pure_eval	https://github.com/alexmojaki/pure_eval?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
py	https://py.readthedocs.io/en/latest/	https://github.com/pytest-dev/py/blob/master/LICENSE
py-cpuinfo	https://github.com/workhorsy/py-cpuinfo	https://github.com/workhorsy/py-cpuinfo/blob/master/LICENSE
pyasn1	https://github.com/etingof/pyasn1	https://github.com/etingof/pyasn1/blob/master/LICENSE.rst
pyasn1-modules	https://github.com/etingof/pyasn1-modules	https://github.com/etingof/pyasn1-modules/blob/master/LICENSE.txt
pyasn1_modules	https://pyasn1.readthedocs.io/en/latest/contents.html	https://github.com/pyasn1/pyasn1-modules?tab=BSD-2-Clause-1-ov-file#readme
pybind11	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11-abi	https://github.com/pybind/pybind11	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11_global	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pycairo	https://pycairo.readthedocs.io/en/latest/index.html	https://github.com/pygobject/pycairo/blob/master/COPYING
pycosat	https://github.com/conda/pycosat	https://github.com/conda/pycosat/blob/main/LICENSE
pycparser	https://github.com/eliben/pycparser	https://github.com/eliben/pycparser/blob/master/LICENSE
pydantic	https://pydantic-docs.helpmanual.io	https://github.com/samuelcolvin/pydantic/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pydantic-core	https://github.com/pydantic/pydantic-core	https://github.com/pydantic/pydantic-core
pyedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pyEMMA	http://github.com/markovmodel/PyEMMA	https://github.com/markovmodel/PyEMMA/blob/devel/LICENSE.txt
Pygments	https://pygments.org	https://github.com/pygments/pygments/blob/master/LICENSE
pygments	https://pygments.org/	https://github.com/pygments/pygments?tab=BSD-2-Clause-1-ov-file#readme
pygraphviz	https://pygraphviz.github.io	https://github.com/pygraphviz/pygraphviz/blob/main/LICENSE
pygtop	https://pygtop.readthedocs.io/en/latest/	https://github.com/samirelanduk/pygtop/blob/master/LICENSE
pyHanko	https://github.com/MatthiasValvekens/pyHanko	https://github.com/MatthiasValvekens/pyHanko/blob/master/LICENSE
pyhanko-certvalidator	https://github.com/MatthiasValvekens/certvalidator	https://github.com/MatthiasValvekens/certvalidator/blob/master/LICENSE
PyJWT	https://github.com/jpadilla/pyjwt	https://github.com/jpadilla/pyjwt/blob/master/LICENSE
pymbar	https://pymbar.org	https://github.com/choderalab/pymbar/blob/master/LICENSE
pynvml	https://github.com/gpuopenanalytics/pynvml	https://github.com/gpuopenanalytics/pynvml/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pyOpenSSL	https://pyopenssl.org/	https://github.com/pyca/pyopenssl/blob/main/LICENSE
pyparsing	https://pyparsing-docs.readthedocs.io/en/latest/	https://github.com/pyparsing/pyparsing/blob/master/LICENSE
PyPDF3	https://github.com/sfneal/PyPDF3	https://github.com/sfneal/PyPDF3/blob/master/LICENSE
pyrsistent	http://github.com/tobgu/pyrsistent/	https://github.com/tobgu/pyrsistent/blob/master/LICENSE.mit
pysam	https://github.com/pysam-developers/pysam	https://github.com/pysam-developers/pysam/blob/master/COPYING
PySocks	https://github.com/Anorov/PySocks	https://github.com/Anorov/PySocks/blob/master/LICENSE
pytables	https://www.pytables.org	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
python	https://www.python.org/	https://docs.python.org/3/license.html
python-bidi	https://github.com/MeirKriheli/python-bidi	https://github.com/MeirKriheli/python-bidi/blob/master/COPYING
python-constraint	https://github.com/python-constraint/python-constraint	https://github.com/python-constraint/python-constraint/blob/master/LICENSE
python-dateutil	https://dateutil.readthedocs.io	https://github.com/dateutil/dateutil/blob/master/LICENSE
python-json-logger	http://github.com/madzak/python-json-logger	https://github.com/madzak/python-json-logger/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
python-pptx	https://github.com/scanny/python-pptx	https://github.com/scanny/python-pptx/blob/master/LICENSE
python-ldap	https://www.python-ldap.org/	https://github.com/python-ldap/python-ldap/blob/python-ldap-3.4.0/LICENCE
python3-saml	https://github.com/onelogin/python3-saml	https://github.com/onelogin/python3-saml/blob/master/LICENSE
python_abi	https://github.com/conda-forge/python_abi-feedstock	https://github.com/conda-forge/python_abi-feedstock/blob/main/LICENSE.txt
pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning/blob/master/LICENSE
pytz	https://pythonhosted.org/pytz	https://git.launchpad.net/pytz/tree/LICENSE.txt
pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shim/blob/master/LICENSE
PyWavelets	https://github.com/PyWavelets/pywt	https://github.com/PyWavelets/pywt/blob/master/LICENSE
PyYAML	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
pyyml	No longer available	No longer available
pyzmq	https://pyzmq.readthedocs.io/en/latest/	https://github.com/zeromq/pyzmq/blob/main/COPYING.BSD
qcelemental	https://github.com/MolSSI/QCElemental	https://github.com/MolSSI/QCElemental/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
qcengine	https://github.com/MolSSI/QCEngine	https://github.com/MolSSI/QCEngine/blob/master/LICENSE
qrcode	https://github.com/lincolnloop/python-qrcode	https://github.com/lincolnloop/python-qrcode/blob/main/LICENSE
ramda	https://github.com/ramda/ramda	https://github.com/ramda/ramda/raw/master/LICENSE.txt
rapidjson	https://rapidjson.org/	https://github.com/Tencent/rapidjson/blob/master/license.txt
rdkit	https://www.rdkit.org	https://github.com/rdkit/rdkit/blob/master/license.txt
re2	https://github.com/google/re2	https://github.com/google/re2/blob/main/LICENSE
readme-renderer	https://github.com/pypa/readme_renderer	https://github.com/pypa/readme_renderer/blob/main/LICENSE
redis-py	https://github.com/andymccurdy/redis-py	https://github.com/redis/redis-py/blob/master/LICENSE
referencing	https://github.com/python-jsonschema/referencing	https://github.com/python-jsonschema/referencing
regex	https://github.com/mrabarnett/mrab-regex	https://github.com/mrabarnett/mrab-regex/blob/hg/LICENSE.txt
reportlab	https://www.reportlab.com	https://www.reportlab.com/dev/docs/faq/#1.3
reproc	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE
reproc-cpp	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
requests	https://requests.readthedocs.io	https://github.com/psf/requests/blob/main/LICENSE
requests-oauthlib	https://github.com/requests/requests-oauthlib	https://github.com/requests/requests-oauthlib/blob/master/LICENSE
requests-toolbelt	https://toolbelt.readthedocs.org	https://github.com/requests/toolbelt/blob/master/LICENSE
resumable	https://github.com/stevvooe/resumable	https://github.com/stevvooe/resumable/blob/master/LICENSE
retrying	https://github.com/rholder/retrying	https://github.com/rholder/retrying/blob/master/LICENSE
rfc3339-validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator/blob/master/LICENSE
rfc3339_validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator?tab=MIT-1-ov-file#readme
rfc3986	https://rfc3986.readthedocs.io/en/latest/	https://github.com/python-hyper/rfc3986/blob/main/LICENSE
rfc3986-validator	https://github.com/naimetti/rfc3986-validator	https://github.com/naimetti/rfc3986-validator/blob/master/LICENSE
rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax?tab=MIT-1-ov-file#readme
rich	https://github.com/Textualize/rich	https://github.com/Textualize/rich/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
rich-argparse	https://github.com/hamdanal/rich-argparse	https://github.com/hamdanal/rich-argparse/blob/main/LICENSE
rlPyCairo	https://pypi.org/project/rlPyCairo/	https://hg.reportlab.com/hg-public/rlPyCairo/file/tip/rlPyCairo/LICENSE.txt
RocksDict	https://github.com/rocksdict/RocksDict	https://github.com/rocksdict/RocksDict?tab=MIT-1-ov-file#readme
roman-numerals	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
roman-numerals-py	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
rotary-embedding-torch	https://github.com/lucidrains/rotary-embedding-tor	https://github.com/lucidrains/rotary-embedding-tor blob/main/LICENSE
rpds-py	https://github.com/crate-py/rpds	https://github.com/crate-py/rpds
rpmpack	https://github.com/google/rpmpack	https://github.com/google/rpmpack/blob/master/LICENSE
rsa	https://stuvel.eu/rsa	https://github.com/sybrenstuvel/python-rsa/blob/main/LICENSE
ruamel-yaml	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/LICENSE
ruamel.yaml.clib	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
s3transfer	https://github.com/boto/s3transfer	https://github.com/boto/s3transfer/blob/develop/LICENSE.txt
sacrebleu	https://github.com/mjpost/sacrebleu	https://github.com/mjpost/sacrebleu/blob/master/LICENSE.txt
sapiens	https://github.com/Merck/Sapiens	https://github.com/Merck/Sapiens/blob/main/LICENSE
sasl	https://mellium.im/sasl	https://github.com/mellium/sasl/blob/main/LICENSE
scikit-gstat	https://mmaelicke.github.io/scikit-gstat/	https://github.com/mmaelicke/scikit-gstat/blob/main/LICENSE
scikit-image	https://scikit-image.org	https://scikit-image.org/docs/dev/license.html
scikit-learn	https://scikit-learn.org/stable/	https://github.com/scikit-learn/scikit-learn/blob/main/COPYING
scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra/blob/main/LICENSE
scikit-network	https://github.com/sknetwork-team/scikit-network	https://github.com/sknetwork-team/scikit-network?tab=License-1-ov-file#readme
scipy	https://scipy.org	https://github.com/scipy/scipy/blob/main/LICENSE.txt
seaborn	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE
seaborn-base	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
semver	https://github.com/Masterminds/semver/v3	https://github.com/Masterminds/semver/blob/v3.0.3/LICENSE.txt
Send2Trash	https://github.com/arsenetar/send2trash	https://github.com/arsenetar/send2trash/blob/master/LICENSE
sentry-sdk	https://github.com/getsentry/sentry-python	https://github.com/getsentry/sentry-python/blob/master/LICENSE
setproctitle	https://github.com/dvarrazzo/py-setproctitle	https://github.com/dvarrazzo/py-setproctitle/blob/master/LICENSE
setuptools	https://github.com/pypa/setuptools	https://github.com/pypa/setuptools/blob/main/LICENSE
setuptools-scm	https://github.com/pypa/setuptools_scm/	https://github.com/pypa/setuptools_scm/blob/main/LICENSE
sh	https://github.com/amoffat/sh	https://github.com/amoffat/sh/blob/develop/LICENSE.txt
shellingham	https://github.com/sarugaku/shellingham	https://github.com/sarugaku/shellingham/blob/master/LICENSE
simdjson	https://github.com/simdjson/simdjson/tree/master	https://github.com/simdjson/simdjson/blob/master/LICENSE
simint	https://www.bennyp.org/research/simint/	https://github.com/simint-chem/simint-generator/blob/master/LICENSE
simplejson	https://github.com/simplejson/simplejson	https://github.com/simplejson/simplejson/blob/master/LICENSE.txt
six	https://github.com/benjaminp/six	https://github.com/benjaminp/six/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
smirnoff99frosst	https://github.com/openforcefield/smirnoff99frosst	https://github.com/openforcefield/smirnoff99Frosst/blob/master/LICENSE
smmap	https://github.com/gitpython-developers/smmap	https://github.com/gitpython-developers/smmap/blob/master/LICENSE
snappy	https://github.com/google/snappy	https://github.com/google/snappy/blob/main/COPYING
sniffio	https://github.com/python-trio/sniffio	https://github.com/python-trio/sniffio/blob/master/LICENSE.MIT
snowballstemmer	https://github.com/snowballstem/snowball	https://github.com/snowballstem/snowball/blob/master/COPYING
soupsieve	https://github.com/facelessuser/soupsieve	https://github.com/facelessuser/soupsieve/blob/main/LICENSE.md
spdlog	https://github.com/gabime/spdlog	https://github.com/gabime/spdlog?tab=License-1-ov-file#readme
spglib	https://spglib.readthedocs.io/	https://github.com/spglib/spglib/blob/develop/COPYING
sphinx	https://github.com/sphinx-doc/sphinx	https://github.com/sphinx-doc/sphinx/blob/5.x/LICENSE
sphinxcontrib-applehelp	https://github.com/sphinx-doc/sphinxcontrib-applehe	https://github.com/sphinx-doc/sphinxcontrib-applehe blob/master/LICENSE
sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp blob/master/LICENSE
sphinxcontrib-htmlhelp	https://github.com/sphinx-doc/sphinxcontrib-htmlhel	https://github.com/sphinx-doc/sphinxcontrib-htmlhel blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath/blob/master/LICENSE
sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp/blob/master/LICENSE
sphinxcontrib-serializinghtml	https://github.com/sphinx-doc/sphinxcontrib-serializ	https://github.com/sphinx-doc/sphinxcontrib-serializ blob/master/LICENSE
SQLAlchemy	https://www.sqlalchemy.org	https://docs.sqlalchemy.org/en/14/copyright.html
SQLAlchemy	https://www.sqlalchemy.org/	https://github.com/sqlalchemy/sqlalchemy?tab=MIT-1-ov-file#readme
sqlite	https://sqlite.org/index.html	https://sqlite.org/copyright.html
sqlparse	https://github.com/andialbrecht/sqlparse	https://github.com/andialbrecht/sqlparse/blob/master/LICENSE
stack-data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data/blob/master/LICENSE.txt
stack_data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data?tab=MIT-1-ov-file#readme
starfile	https://github.com/alisterburt/starfile	https://github.com/teamtomo/starfile/blob/main/LICENSE
statsmodels	https://github.com/statsmodels/statsmodels	https://github.com/statsmodels/statsmodels/blob/main/LICENSE.txt
structlog	https://www.structlog.org/	https://github.com/hynek/structlog/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
streamlit	https://github.com/streamlit/streamlit	https://github.com/streamlit/streamlit/blob/develop/LICENSE
svglib	https://github.com/deeplook/svglib	https://github.com/deeplook/svglib/blob/master/LICENSE.txt
sympy	https://sympy.org	https://github.com/sympy/sympy/blob/master/LICENSE
tables	https://www.pytables.org/	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
tabulate	https://github.com/astanin/python-tabulate	https://github.com/astanin/python-tabulate/blob/master/LICENSE
tbb	https://github.com/oneapi-src/oneTBB	https://github.com/oneapi-src/oneTBB/blob/master/LICENSE.txt
tenacity	https://github.com/jd/tenacity	https://github.com/jd/tenacity/blob/master/LICENSE
tensorboard	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-data-server	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-plugin-wit	https://github.com/pair-code/what-if-tool	https://github.com/PAIR-code/what-if-tool/blob/master/LICENSE
tensorboardX	https://github.com/lanpa/tensorboardX	https://github.com/lanpa/tensorboardX/blob/master/LICENSE
tensorflow	https://github.com/tensorflow/tensorflow	https://github.com/tensorflow/tensorflow/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tensorflow-estimator	https://github.com/tensorflow/estimator	https://github.com/tensorflow/estimator/blob/master/LICENSE
tensorflow-io-gcs-filesystem	https://github.com/tensorflow/io	https://github.com/tensorflow/io/blob/master/LICENSE
tensorflow-probability	https://github.com/tensorflow/probability	https://github.com/tensorflow/probability/blob/main/LICENSE
termcolor	https://github.com/hugovk/termcolor	https://github.com/hugovk/termcolor/blob/main/COPYING.txt
terminado	https://github.com/jupyter/terminado	https://github.com/jupyter/terminado/blob/main/LICENSE
testpath	https://github.com/jupyter/testpath	https://github.com/jupyter/testpath/blob/master/LICENSE
textangular	https://github.com/fraywing/textAngular	https://github.com/fraywing/textAngular
texttable	https://github.com/foutaise/texttable/	https://github.com/foutaise/texttable/?tab=MIT-1-ov-file#readme
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
threadpoolctl	https://github.com/joblib/threadpoolctl	https://github.com/joblib/threadpoolctl/blob/master/LICENSE
three	https://github.com/mrdoob/three.js	https://github.com/mrdoob/three.js/raw/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tiffile	https://github.com/cgohlke/tiffile/	https://github.com/cgohlke/tiffile/blob/master/LICENSE
tinycss2	https://github.com/Kozea/tinycss2/	https://github.com/Kozea/tinycss2/blob/master/LICENSE
tinyxml2	https://github.com/leethomason/tinyxml2	https://github.com/leethomason/tinyxml2/blob/master/LICENSE.txt
tk	https://www.tcl.tk/	https://www.tcl.tk/software/tcltk/license.html
toml	https://github.com/toml-lang/toml	https://github.com/toml-lang/toml/blob/main/LICENSE
tomli	https://github.com/hukkin/tomli	https://github.com/hukkin/tomli/blob/master/LICENSE
toolz	https://github.com/pytoolz/toolz	https://github.com/pytoolz/toolz/blob/master/LICENSE.txt
torch	https://pytorch.org/	https://pypi.org/project/torch/#license
torch-geometric	https://pyg.org/	https://github.com/pyg-team/pytorch_geometric?tab=MIT-1-ov-file#readme
torcheval	https://github.com/pytorch/torcheval	https://github.com/pytorch/torcheval?tab=License-1-ov-file#readme
torchmetrics	https://github.com/Lightning-AI/torchmetrics	https://github.com/Lightning-AI/torchmetrics?tab=Apache-2.0-1-ov-file#readme
tornado	https://www.tornadoweb.org	https://github.com/tornadoweb/tornado/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL		
tqdm	https://github.com/tqdm/tqdm	https://github.com/tqdm/tqdm/blob/master/LICENCE		
tracy	https://github.com/gear-genomics/tracy	https://github.com/gear-genomics/tracy/blob/main/LICENSE		
traitlets	https://github.com/ipython/traitlets	https://github.com/ipython/traitlets/blob/main/COPYING.md		
triton	https://github.com/openai/triton/	https://github.com/openai/triton/blob/main/LICENSE		
truststore	https://github.com/sethmlarson/trus	https://github.com/sethmlarson/truststore/blob/main/LICENSE		
truststore	https://github.com/sethmlarson/truststore	https://github.com/sethmlarson/truststore/blob/main/LICENSEtruststore	https://github.com/sethmlarson/truststore	https://github.com/sethmlarson/truststore/blob/main/LICENSE
ts-jest	https://github.com/kulshekhar/ts-jest	https://github.com/kulshekhar/ts-jest/raw/master/LICENSE.md		
ts-loader	https://github.com/TypeStrong/ts-loader	https://github.com/TypeStrong/ts-loader/raw/master/LICENSE		
twine	https://github.com/pypa/twine	https://github.com/pypa/twine/blob/main/LICENSE		
twinj uuid	https://github.com/twinj/uuid	https://github.com/twinj/uuid/blob/master/LICENSE		
types	https://github.com/babel/babel	https://github.com/babel/babel/raw/master/LICENSE		
types-python-dateutil	https://github.com/python/typeshed	https://github.com/python/typeshed?tab=License-1-ov-file#readme		
types-requests	https://pypi.org/project/types-requests/	https://github.com/python/typeshed/blob/main/LICENSE		

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
typescript	https://github.com/Microsoft/TypeScript	https://github.com/Microsoft/TypeScript/raw/master/LICENSE.txt
typing-inspection	https://github.com/pydantic/typing-inspection	https://github.com/pydantic/typing-inspection?tab=MIT-1-ov-file#readme
typing-utils	https://github.com/bojiang/typing_utils	https://github.com/bojiang/typing_utils?tab=Apache-2.0-1-ov-file#readme
typing_extensions	https://github.com/python/typing	https://github.com/python/typing/blob/master/LICENSE
typing_utils	https://github.com/bojiang/typing_utils	https://github.com/bojiang/typing_utils?tab=Apache-2.0-1-ov-file#readme
tzdata	https://github.com/python/tzdata	https://github.com/python/tzdata/blob/master/LICENSE
tzlocal	https://github.com/regebro/tzlocal	https://github.com/regebro/tzlocal/blob/master/LICENSE.txt
umf	https://github.com/oneapi-src/unified-memory-fram	https://github.com/oneapi-src/unified-memory-fram tab=License-1-ov-file#readme
umi-tools	https://github.com/CGATOxford/UMI-tools	https://github.com/CGATOxford/UMI-tools/blob/master/LICENSE
unicodedata2	https://github.com/fonttools/unicodedata2	https://github.com/fonttools/unicodedata2/blob/master/LICENSE
uri-template	https://gitlab.linss.com/open-source/python/uri-template	https://gitlab.linss.com/open-source/python/uri-template/-/blob/main/LICENSE
uritools	https://github.com/tkem/uritools/	https://github.com/tkem/uritools/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL		
urllib3	https://urllib3.readthedocs.io/	https://github.com/urllib3/urllib3/blob/main/LICENSE.txt		
vine	https://github.com/celery/vine	https://github.com/celery/vine/blob/master/LICENSE		
vue	https://github.com/vuejs/vue	https://github.com/vuejs/vue/raw/master/LICENSE		
wadler_lindig	https://github.com/patrick-kidger/wadler_lindig	https://github.com/patrick-kidger/wadler_lindig?tab=Apache-2.0-1-ov-file#readme		
wadler_lindig	https://github.com/patrick-kidger/wadler_lindig	https://github.com/patrick-kidger/wadler_lindig?tab=Apache-2.0-1-ov-file#readmewcwidth	https://github.com/jquast/wcwidth	https://github.com/jquast/wcwidth/blob/master/LICENSE
wandb	https://github.com/wandb/wandb	https://github.com/wandb/wandb/blob/main/LICENSE		
webcolors	https://github.com/ubernostrum/webcolors	https://github.com/ubernostrum/webcolors?tab=BSD-3-Clause-1-ov-file#readme		
webencodings	https://github.com/gsnedders/python-webencodings	https://github.com/gsnedders/python-webencodings/blob/master/LICENSE		
websocket-client	https://github.com/websocket-client/websocket-client.git	https://github.com/websocket-client/websocket-client/blob/master/LICENSE		
Werkzeug	https://palletsprojects.com/p/werkzeug/	https://werkzeug.palletsprojects.com/en/2.1.x/license/		
westpa	http://github.com/westpa/westpa	https://github.com/westpa/westpa/blob/westpa2/LICENSE		
wheel	https://github.com/pypa/wheel	https://github.com/pypa/wheel/blob/main/LICENSE.txt		

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
widgetsnbextension	https://github.com/jupyter-widgets/ipywidgets#readme	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE
wrapt	https://github.com/GrahamDumpleton/wrapt	https://github.com/GrahamDumpleton/wrapt/blob/develop/LICENSE
wsutil	https://github.com/yhat/wsutil	https://github.com/yhat/wsutil/blob/master/LICENSE
x/lint	https://golang.org/x/lint	https://github.com/golang/lint/blob/master/LICENSE
x/mod	https://golang.org/x/mod/semver	https://cs.opensource.google/go/x/mod/+/master:LICENSE
x/net	https://golang.org/x/net	https://cs.opensource.google/go/x/net/+/master:LICENSE
x/oauth2	https://golang.org/x/oauth2	https://cs.opensource.google/go/x/oauth2/+/master:LICENSE
x/sys	https://golang.org/x/sys	https://cs.opensource.google/go/x/sys/+/master:LICENSE
x/text	https://golang.org/x/text	https://cs.opensource.google/go/x/text/+/master:LICENSE
x/tools	https://golang.org/x/tools	https://cs.opensource.google/go/x/tools/+/master:LICENSE
x/xerrors	https://golang.org/x/xerrors	https://cs.opensource.google/go/x/xerrors/+/master:LICENSE
xgboost	https://github.com/dmlc/xgboost	https://github.com/dmlc/xgboost/blob/master/LICENSE
xhtml2pdf	http://github.com/xhtml2pdf/xhtml2pdf	https://github.com/xhtml2pdf/xhtml2pdf/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
xlrd	https://github.com/python-excel/xlrd	https://github.com/python-excel/xlrd/blob/master/LICENSE
xlsxwriter	https://github.com/jmcnamara/XlsxWriter	https://github.com/jmcnamara/XlsxWriter/blob/main/LICENSE.txt
xmlsec	https://github.com/mehcode/python-xmlsec	https://github.com/mehcode/python-xmlsec/blob/master/LICENSE
xmltodict	https://github.com/martinblech/xmltodict	https://github.com/martinblech/xmltodict/blob/master/LICENSE
xorg-kbproto	https://gitlab.freedesktop.org/xorg/proto/kbproto	https://gitlab.freedesktop.org/xorg/proto/kbproto/-/blob/master/COPYING
xorg-libice	https://gitlab.freedesktop.org/xorg/lib/libice	https://gitlab.freedesktop.org/xorg/lib/libice/-/blob/master/COPYING
xorg-libsm	https://gitlab.freedesktop.org/xorg/lib/libsm	https://gitlab.freedesktop.org/xorg/lib/libsm/-/blob/master/COPYING
xorg-libx11	https://gitlab.freedesktop.org/xorg/lib/libx11	https://gitlab.freedesktop.org/xorg/lib/libx11/-/blob/master/COPYING
xorg-libxau	https://gitlab.freedesktop.org/xorg/lib/libxau	https://gitlab.freedesktop.org/xorg/lib/libxau/-/blob/master/COPYING
xorg-libxdmcp	https://gitlab.freedesktop.org/xorg/lib/libxdmcp	https://gitlab.freedesktop.org/xorg/lib/libxdmcp/-/blob/master/COPYING

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
xorg-libxext	https://gitlab.freedesktop.org/xorg/lib/libxext	https://gitlab.freedesktop.org/xorg/lib/libxext/-/blob/master/COPYING
xorg-libxrender	https://gitlab.freedesktop.org/xorg/lib/libxrender	https://gitlab.freedesktop.org/xorg/lib/libxrender/-/blob/master/COPYING
xorg-libxt	https://gitlab.freedesktop.org/xorg/lib/libxt	https://gitlab.freedesktop.org/xorg/lib/libxt/-/blob/master/COPYING
xorg-renderproto	https://gitlab.freedesktop.org/xorg/proto/renderproto	https://gitlab.freedesktop.org/xorg/proto/renderproto/-/blob/master/COPYING
xorg-xextproto	https://gitlab.freedesktop.org/xorg/proto/xextproto	https://gitlab.freedesktop.org/xorg/proto/xextproto/-/blob/master/COPYING
xorg-xproto	https://gitlab.freedesktop.org/xorg/proto/xproto	https://gitlab.freedesktop.org/xorg/proto/xproto/-/blob/master/COPYING
xxhash	https://github.com/cespare/xxhash/v2	https://github.com/cespare/xxhash/blob/main/LICENSE.txt
xz	https://github.com/ulikunitz/xz	https://github.com/ulikunitz/xz/blob/master/LICENSE
yaml	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
yaml-cpp	https://github.com/jbeder/yaml-cpp	https://github.com/jbeder/yaml-cpp/blob/master/LICENSE
yaml.v2	https://gopkg.in/yaml.v2	https://github.com/go-yaml/yaml/blob/v2.4.0/LICENSE
yaml.v3	https://gopkg.in/yaml.v3	https://github.com/go-yaml/yaml/blob/v3/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
yarl	https://github.com/aio-libs/yarl/	https://github.com/aio-libs/yarl/blob/master/LICENSE
yaspin	https://github.com/pavdmyt/yaspin	https://github.com/pavdmyt/yaspin/blob/master/LICENSE
yfiles	https://www.yworks.com/products/yfiles	commercial license
yml	https://pypi.org/project/yml/	N/A
zap	https://go.uber.org/zap	https://github.com/uber-go/zap/blob/master/LICENSE.txt
zipp	https://github.com/jaraco/zipp	https://github.com/jaraco/zipp/blob/main/LICENSE
zlib	https://zlib.net/	https://zlib.net/zlib_license.html
zstandard	https://github.com/indygreg/python-zstandard	https://github.com/indygreg/python-zstandard/blob/main/LICENSE
zstd	https://facebook.github.io/zstd/	https://github.com/facebook/zstd/blob/dev/LICENSE
click-option-group	https://github.com/click-contrib/click-option-group	https://github.com/click-contrib/click-option-group/blob/master/LICENSE

17.1 GCC

On the Linux platform, The OpenEye toolkits are linked with the GCC libraries using an eligible compilation process under the terms of the GCC RUNTIME LIBRARY EXCEPTION (see below).

The OpenEye toolkits are NOT open source software and are NOT governed by GPL or other open source licenses. Your right to use the toolkits are governed by the OpenEye license which is binding to You.

For certain Linux distributions, GCC library components are included as object files. Your rights to use GCC library components (libstdc++, libgcc, and libgomp) included with these distributions are governed by the GPLv3 license as follows:

The components libstdc++, libgcc, and libgomp are part of GCC.

- GCC is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
- GCC is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the

implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with GCC (see below). If not, see Licenses GNU Project .

Per GPLv3, section 6d, the source for GCC is available for download from the following location: GNU. The GCC library components included here are from version gcc-4.9.3 and are built using the standard instructions at Installing GCC.

17.1.1 GCC RUNTIME LIBRARY EXCEPTION

```
Version 3.1, 31 March 2009

Copyright © 2009 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document,␣
↪but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional permission under␣
↪section 7 of the GNU General Public License,
version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a␣
↪notice placed by the copyright holder
of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header␣
↪files and runtime libraries with
the compiled program. The purpose of this Exception is to allow compilation of non-GPL␣
↪(including proprietary) programs
to use, in this way, the header files and runtime libraries covered by this Exception.

0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for␣
↪execution after a Compilation Process,
or makes use of an interface provided by the Runtime Library, but is not otherwise based␣
↪on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications,␣
↪governed by version 3 (or a specified
later version) of the GNU General Public License (GPL) with the option of using any␣
↪subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and␣
↪use would permit combination with
GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor␣
↪architecture, in executable
form or suitable for input to an assembler, loader, linker and/or execution phase.␣
↪Notwithstanding that, Target Code
does not include data in any format that is used as a compiler intermediate␣
↪representation, or used for producing a
compiler intermediate representation.
```

(continues on next page)

(continued from previous page)

```
The "Compilation Process" transforms code entirely represented in non-intermediate␣
↪languages designed for human-written
code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use␣
↪of source code generators and
preprocessors need not be considered part of the Compilation Process, since the␣
↪Compilation Process can be understood
as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-
↪compatible software, or if it is
done without using any work based on GCC. For example, using non-GPL-compatible Software␣
↪to optimize any GCC intermediate
representations would not qualify as an Eligible Compilation Process.

1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by combining the Runtime␣
↪Library with Independent Modules,
even if such propagation would otherwise violate the terms of GPLv3, provided that all␣
↪Target Code was generated by
Eligible Compilation Processes. You may then convey such a combination under terms of␣
↪your choice, consistent with the
licensing of the Independent Modules.

2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general presumption that third-
↪party software is unaffected by
the copyleft requirements of the license of GCC.
```

See also

- http://www.gnu.org/licenses/gcc-exception.html

17.1.2 GNU GENERAL PUBLIC LICENSE

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

 The GNU General Public License is a free, copyleft license for
 software and other kinds of works.

   The licenses for most software and other practical works are designed
```

(continues on next page)

(continued from previous page)

```
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

  Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

  For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

  Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
```

(continues on next page)

(continued from previous page)

```
patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and
modification follow.

                    TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.
```

(continues on next page)

(continued from previous page)

```
  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
```

(continues on next page)

(continued from previous page)

```
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.


  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

   a) The work must carry prominent notices stating that you modified
   it, and giving a relevant date.

   b) The work must carry prominent notices stating that it is
   released under this License and any conditions added under section
   7.  This requirement modifies the requirement in section 4 to
   "keep intact all notices".
```

(continues on next page)

(continued from previous page)

```
    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

  a) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by the
  Corresponding Source fixed on a durable physical medium
  customarily used for software interchange.

  b) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by a
  written offer, valid for at least three years and valid for as
  long as you offer spare parts or customer support for that product
  model, to give anyone who possesses the object code either (1) a
  copy of the Corresponding Source for all the software in the
  product that is covered by this License, on a durable physical
  medium customarily used for software interchange, for a price no
  more than your reasonable cost of physically performing this
  conveying of source, or (2) access to copy the
  Corresponding Source from a network server at no charge.

  c) Convey individual copies of the object code with a copy of the
  written offer to provide the Corresponding Source.  This
  alternative is allowed only occasionally and noncommercially, and
  only if you received the object code with such an offer, in accord
```

(continues on next page)

(continued from previous page)

```
   with subsection 6b.

   d) Convey the object code by offering access from a designated
   place (gratis or for a charge), and offer equivalent access to the
   Corresponding Source in the same way through the same place at no
   further charge.  You need not require recipients to copy the
   Corresponding Source along with the object code.  If the place to
   copy the object code is a network server, the Corresponding Source
   may be on a different server (operated by you or a third party)
   that supports equivalent copying facilities, provided you maintain
   clear directions next to the object code saying where to find the
   Corresponding Source.  Regardless of what server hosts the
   Corresponding Source, you remain obligated to ensure that it is
   available for as long as needed to satisfy these requirements.

   e) Convey the object code using peer-to-peer transmission, provided
   you inform other peers where the object code and Corresponding
   Source of the work are being offered to the general public at no
   charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
```

(continues on next page)

(continued from previous page)

```
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
```

(continues on next page)

(continued from previous page)

```
   authors of the material; or

   e) Declining to grant rights under trademark law for use of some
   trade names, trademarks, or service marks; or

   f) Requiring indemnification of licensors and authors of that
   material by anyone who conveys the material (or modified versions of
   it) with contractual assumptions of liability to the recipient, for
   any liability that these contractual assumptions directly impose on
   those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.
```

(continues on next page)

(continued from previous page)

```
  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
```

(continues on next page)

(continued from previous page)

```
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
```

(continues on next page)

(continued from previous page)

```
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
```

(continues on next page)

(continued from previous page)

```
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>
```

(continues on next page)

(continued from previous page)

```
    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

See also

- http://www.gnu.org/licenses/gpl.txt

INDEX

Symbols

-aScale)
    brood command line option, 30
-attach
    brood command line option, 30
-attachCut)
    brood command line option, 29
-attachCutoff
    brood command line option, 29
-attachScale
    brood command line option, 30
-bondOrder
    brood command line option, 29
-buildType
    brood command line option, 28
-bumpRadius
    brood command line option, 30
-capAttach
    chomp command line option, 42
-checkBond
    brood command line option, 28
-checkGeometry
    brood command line option, 29
-cpddb
    brood command line option, 28
-db
    brood command line option, 28
-deltaLocalStrain
    brood command line option, 28
-dots
    chomp command line option, 42
-eganEgg
    brood command line option, 30
-filter
    chomp command line option, 43
-flipN
    chomp command line option, 42
-flipper
    chomp command line option, 42
-forceFlip
    chomp command line option, 42
-idea
    brood command line option, 28
-ignoreProtein
    brood command line option, 30
-ignoreProteinSelect
    brood command line option, 30
-in
    brood command line option, 27
    chomp command line option, 41
-in1
    brooddbmerge command line option, 46
-in2
    brooddbmerge command line option, 46
-log
    brood command line option, 27
-maxABS)
    brood command line option, 31
-maxAcceptors
    brood command line option, 31
-maxAromFJCt
    brood command line option, 32
-maxChiral
    chomp command line option, 44
-maxComplexity
    brood command line option, 31
-maxDegree
    chomp command line option, 43
-maxDonors
    brood command line option, 31
-maxFrequency
    brood command line option, 31
-maxFsp3C
    brood command line option, 32
-maxHits
    brood command line option, 28
-maxHvy
    chomp command line option, 43
-maxHvyAtom
    brood command line option, 32
-maxLipinski
    brood command line option, 31
-maxLipinskiAcc)
    brood command line option, 31

-maxLipinskiDon)
    brood command line option, 31
-maxLocalStrain
    brood command line option, 28
-maxMartin
    brood command line option, 31
-maxMolWt
    brood command line option, 32
    chomp command line option, 43
-maxRotBond
    brood command line option, 32
-maxlogp
    brood command line option, 31
-maxpsa
    brood command line option, 32
-minABS)
    brood command line option, 31
-minAcceptors
    brood command line option, 31
-minAromFJCt
    brood command line option, 32
-minComplexity
    brood command line option, 31
-minDegree
    chomp command line option, 43
-minDonors
    brood command line option, 31
-minFrequency
    brood command line option, 31
    chomp command line option, 43
-minFsp3C
    brood command line option, 32
-minHvy
    chomp command line option, 43
-minHvyAtom
    brood command line option, 32
-minLipinski
    brood command line option, 31
-minLipinskiAcc)
    brood command line option, 31
-minLipinskiDon)
    brood command line option, 31
-minMartin
    brood command line option, 31
-minMolWt
    brood command line option, 32
-minRotBond
    brood command line option, 32
-minlogp
    brood command line option, 31
-minpsa
    brood command line option, 32
-molNames
    brood command line option, 27
-mpi_hostfile
    brood command line option, 28
    chomp command line option, 41
-mpi_np
    brood command line option, 27
    chomp command line option, 41
-neutralpH
    brood command line option, 28
-noqueryprotein)
    brood command line option, 30
-noqueryselection)
    brood command line option, 30
-omega
    chomp command line option, 42
-out
    brood command line option, 27
    brooddbmerge command line option, 46
    chomp command line option, 41
-overwrite
    brooddbmerge command line option, 46
-param
    brood command line option, 27
    chomp command line option, 41
-prefix
    brood command line option, 27
    chomp command line option, 42
-primaryFrag
    chomp command line option, 43
-property
    brood command line option, 29
-queryMol)
    brood command line option, 27
-quickLook
    brood command line option, 28
-rangeOffset
    brood command line option, 30
-rangeSize
    brood command line option, 30
-readConfs
    chomp command line option, 43
-removeDuplicates
    brooddbmerge command line option, 46
-ringOnly
    brood command line option, 29
-scoreType
    brood command line option, 30
-shapeCutoff
    brood command line option, 29
-smarts
    chomp command line option, 42
-tautomers
    brood command line option, 28
-userFrags
    chomp command line option, 41

-userUnique
    chomp command line option, 42
-veber
    brood command line option, 30
-verbose
    brood command line option, 27
    brooddbmerge command line option, 46

B

brood command line option
    -aScale), 30
    -attach, 30
    -attachCut), 29
    -attachCutoff, 29
    -attachScale, 30
    -bondOrder, 29
    -buildType, 28
    -bumpRadius, 30
    -checkBond, 28
    -checkGeometry, 29
    -cpddb, 28
    -db, 28
    -deltaLocalStrain, 28
    -eganEgg, 30
    -idea, 28
    -ignoreProtein, 30
    -ignoreProteinSelect, 30
    -in, 27
    -log, 27
    -maxABS), 31
    -maxAcceptors, 31
    -maxAromFJCt, 32
    -maxComplexity, 31
    -maxDonors, 31
    -maxFrequency, 31
    -maxFsp3C, 32
    -maxHits, 28
    -maxHvyAtom, 32
    -maxLipinski, 31
    -maxLipinskiAcc), 31
    -maxLipinskiDon), 31
    -maxLocalStrain, 28
    -maxMartin, 31
    -maxMolWt, 32
    -maxRotBond, 32
    -maxlogp, 31
    -maxpsa, 32
    -minABS), 31
    -minAcceptors, 31
    -minAromFJCt, 32
    -minComplexity, 31
    -minDonors, 31
    -minFrequency, 31
    -minFsp3C, 32
    -minHvyAtom, 32
    -minLipinski, 31
    -minLipinskiAcc), 31
    -minLipinskiDon), 31
    -minMartin, 31
    -minMolWt, 32
    -minRotBond, 32
    -minlogp, 31
    -minpsa, 32
    -molNames, 27
    -mpi_hostfile, 28
    -mpi_np, 27
    -neutralpH, 28
    -noqueryprotein), 30
    -noqueryselection), 30
    -out, 27
    -param, 27
    -prefix, 27
    -property, 29
    -queryMol), 27
    -quickLook, 28
    -rangeOffset, 30
    -rangeSize, 30
    -ringOnly, 29
    -scoreType, 30
    -shapeCutoff, 29
    -tautomers, 28
    -veber, 30
    -verbose, 27
    chargeType, 29
brooddbmerge command line option
    -in1, 46
    -in2, 46
    -out, 46
    -overwrite, 46
    -removeDuplicates, 46
    -verbose, 46

C

chargeType
    brood command line option, 29
chomp command line option
    -capAttach, 42
    -dots, 42
    -filter, 43
    -flipN, 42
    -flipper, 42
    -forceFlip, 42
    -in, 41
    -maxChiral, 44
    -maxDegree, 43
    -maxHvy, 43
    -maxMolWt, 43
    -minDegree, 43

-minFrequency, 43
-minHvy, 43
-mpi_hostfile, 41
-mpi_np, 41
-omega, 42
-out, 41
-param, 41
-prefix, 42
-primaryFrag, 43
-readConfs, 43
-smarts, 42
-userFrags, 41
-userUnique, 42