

OpenEye
SCIENTIFIC

OMEGA

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Introduction **1**
- 1.1 Overview . . . 1
- 1.2 Applications . . . 1
- 1.3 Utility Programs . . . 1

2 FILTER **3**
- 2.1 Overview . . . 3
- 2.2 Example Commands . . . 3
- 2.3 Command Line Help . . . 4
- 2.4 Required Parameters . . . 4
- 2.5 Optional Parameters . . . 5
  - 2.5.1 Execute Options . . . 5
  - 2.5.2 Control Options . . . 5
- 2.6 Filter Files . . . 7
  - 2.6.1 Physical Property Limits . . . 8
  - 2.6.2 Functional Group Rules . . . 18
  - 2.6.3 New Rules . . . 63
  - 2.6.4 Selection Statements . . . 63

3 OMEGA **65**
- 3.1 Overview . . . 65
- 3.2 GPU-Omega . . . 65
  - 3.2.1 Prerequisites . . . 65
  - 3.2.2 Supported Platforms . . . 65
  - 3.2.3 Supported GPUs . . . 66
  - 3.2.4 NVIDIA Drivers . . . 66
  - 3.2.5 Performance Tuning . . . 66
  - 3.2.6 Usage . . . 67
- 3.3 Example Commands . . . 67
  - 3.3.1 Basic Commands . . . 67
  - 3.3.2 Using Parameters and Param Files . . . 67
  - 3.3.3 Advanced Commands . . . 68
  - 3.3.4 CSV Report File for `macrocycle` Mode . . . 68
- 3.4 Command Line Help . . . 69
- 3.5 Required Parameters . . . 70
- 3.6 Optional Parameters . . . 70
  - 3.6.1 Execute Options . . . 70
  - 3.6.2 File Options . . . 70
  - 3.6.3 3D Construction Parameters . . . 71
  - 3.6.4 Structure Enumeration . . . 73

3.6.5 Torsion Driving Parameters . . . 74
3.6.6 Thompson Sampling Parameters . . . 75
3.6.7 Stereo Parameters . . . 76
3.6.8 General . . . 76
3.6.9 Options Specific to the Macrocycle Mode . . . 77
3.7 Additional Specifications . . . 78
3.7.1 OMEGA Modes and Parameters . . . 78
3.7.2 Force Fields . . . 79
3.7.3 Torsion Library Format . . . 80

4 Utilities **81**
4.1 Flipper . . . 81
4.1.1 Overview . . . 81
4.1.2 Example Commands . . . 81
4.1.3 Command Line Help . . . 82
4.1.4 Required Parameters . . . 82
4.1.5 Optional Parameters . . . 82
4.2 MakeFragLib . . . 83
4.2.1 Overview . . . 83
4.2.2 Example Commands . . . 84
4.2.3 Command Line Help . . . 84
4.2.4 Required Parameters . . . 84
4.2.5 Optional Parameters . . . 85
4.3 Oeb2SdConf . . . 86

5 Theory **87**
5.1 OMEGA Overview . . . 87
5.1.1 Two Approaches to Conformer Generation . . . 87
5.1.2 Torsion Driving (TD) in OMEGA . . . 88
5.1.3 Thompson Sampling in OMEGA . . . 89
5.1.4 OMEGA Sampling Modes . . . 90
5.1.5 Macrocycle Conformations . . . 91
5.2 Filtering Theory . . . 92
5.2.1 Introduction . . . 92
5.2.2 Filter Preprocessing . . . 96
5.2.3 Molecular Properties and Predictors . . . 98

6 Release History **103**
6.1 OMEGA 7.0.0 . . . 103
6.2 Related Toolkit Information . . . 103
6.2.1 Omega TK 7.0.0 . . . 103
6.2.2 MolProp TK 2.7.0 . . . 103
6.3 OMEGA 6.1.1 . . . 103
6.3.1 Related Toolkit Information . . . 104
6.4 OMEGA 6.1.0 . . . 104
6.4.1 Related Toolkit Information . . . 104
6.5 OMEGA 6.0.0 . . . 104
6.5.1 New Features . . . 104
6.5.2 Minor Bug Fixes . . . 104
6.6 Omega TK 6.0.0 . . . 104
6.6.1 New features . . . 104
6.6.2 Minor bug fixes . . . 105
6.7 MolProp TK 2.6.5 . . . 105
6.8 OMEGA 5.1.0 . . . 105

6.8.1 New Features . . . 105
6.8.2 Major bug fixes . . . 106
6.9 Omega TK 5.1.0 . . . 106
6.9.1 New features . . . 106
6.9.2 Minor bug fixes . . . 106
6.10 MolProp TK 2.6.4 . . . 106
6.11 OMEGA 5.0.0 . . . 106
6.11.1 New Features . . . 106
6.11.2 Major bug fixes . . . 106
6.11.3 Minor bug fixes . . . 107
6.12 Omega TK 5.0.0 . . . 107
6.12.1 New features . . . 107
6.12.2 Minor bug fixes . . . 107
6.13 MolProp TK 2.6.3 . . . 107
6.14 OMEGA 4.2.2 . . . 107
6.14.1 Minor bug fixes . . . 107
6.15 Omega TK 4.2.2 . . . 108
6.15.1 Minor bug fixes . . . 108
6.16 MolProp TK 2.6.2 . . . 108
6.17 OMEGA 4.2.1 . . . 108
6.17.1 New features . . . 108
6.17.2 Minor bug fixes . . . 108
6.17.3 New features . . . 108
6.17.4 Minor bug fixes . . . 109
6.17.5 New features . . . 109
6.17.6 Minor bug fixes . . . 109
6.18 OMEGA 4.2.0 . . . 109
6.18.1 New features . . . 109
6.18.2 Minor bug fixes . . . 109
6.19 Omega TK 4.2.0 . . . 110
6.19.1 New features . . . 110
6.19.2 Minor bug fixes . . . 110
6.20 MolProp TK 2.6.0 . . . 110
6.20.1 New features . . . 110
6.21 OMEGA 4.1.2 . . . 110
6.22 Omega TK 4.1.2 . . . 110
6.22.1 Minor bug fixes . . . 110
6.23 MolProp TK 2.5.7 . . . 111
6.23.1 New Features . . . 111
6.24 OMEGA 4.1.1 . . . 111
6.24.1 Minor Bug Fixes . . . 111
6.25 Omega TK 4.1.1 . . . 111
6.25.1 New features . . . 111
6.25.2 Major bug fixes . . . 111
6.25.3 Minor bug fixes . . . 111
6.26 MolProp TK 2.5.6 . . . 111
6.26.1 Minor bug fixes . . . 112
6.27 OMEGA 4.1.0.0 . . . 112
6.27.1 New Features . . . 112
6.27.2 Major Bug Fixes . . . 112
6.27.3 Documentation Changes . . . 112
6.28 Omega TK 4.1.0 . . . 112
6.28.1 New features . . . 113
6.28.2 Major bug fixes . . . 113

6.28.3 Minor bug fixes . . . 113
6.28.4 Documentation changes . . . 113
6.29 MolProp TK 2.5.5 . . . 113
6.30 OMEGA 4.0.0.6 . . . 113
6.31 OMEGA 4.0.0 . . . 113
6.31.1 New Features . . . 114
6.32 OMEGA 3.1.2 . . . 114
6.32.1 Minor bug fixes . . . 114
6.33 OMEGA 3.1.1 . . . 114
6.33.1 New Features . . . 114
6.33.2 Major bug fixes . . . 114
6.33.3 Minor bug fixes . . . 115
6.34 OMEGA 3.1.0 . . . 115
6.34.1 New features . . . 115
6.34.2 Major bug fixes . . . 115
6.34.3 Minor bug fixes . . . 115
6.35 OMEGA 3.0.1 . . . 115
6.35.1 Major bug fixes . . . 115
6.35.2 Minor bug fixes . . . 116
6.35.3 Other changes . . . 116
6.36 OMEGA 3.0.0 . . . 116
6.36.1 New features . . . 116
6.36.2 Major bug fixes . . . 117
6.36.3 Minor bug fixes . . . 117
6.36.4 Platform support . . . 117
6.37 OMEGA 2.5.1 . . . 118
6.37.1 New features . . . 118
6.37.2 Major bug fixes . . . 118
6.37.3 Minor bug fixes . . . 118
6.37.4 Other changes . . . 118
6.38 OMEGA 2.4.6 . . . 119
6.38.1 New features . . . 119
6.38.2 Bug fixes . . . 119
6.39 OMEGA 2.4.3 . . . 119
6.39.1 Bug fixes . . . 119
6.40 OMEGA 2.4.1 . . . 119
6.40.1 New features . . . 119
6.40.2 Bug fixes . . . 120
6.41 OMEGA 2.4.0 . . . 120
6.42 OMEGA 2.3.3 . . . 120
6.43 OMEGA 2.3.2 . . . 121
6.43.1 Bug fixes . . . 121
6.44 OMEGA 2.3.1 . . . 121
6.44.1 New features . . . 121
6.44.2 Bug fixes . . . 121
6.45 OMEGA 2.3.0 . . . 121
6.45.1 New features . . . 121
6.45.2 Bug fixes . . . 121
6.46 OMEGA 2.2.2 . . . 122
6.46.1 Bug fixes . . . 122
6.47 OMEGA 2.2.1 . . . 122
6.47.1 Bug fixes . . . 122

7 Citation **123**

8 Copyright and Trademarks **125**

9 Sample Code **127**

10 Citation **129**
10.1 Orion® . . . 129
10.2 Orion Floes . . . 129
10.3 Toolkits and Applications . . . 129
10.4 Publications for Bibliographies . . . 131
10.4.1 Orion . . . 131
10.4.2 AFITT and FLYNN Applications . . . 131
10.4.3 OEDocking Application and Toolkit . . . 131
10.4.4 OMEGA Application and Toolkit . . . 131
10.4.5 ROCS Application . . . 132
10.5 OpenEye MMDS Web Service . . . 132

11 Technology Licensing **133**
11.1 GCC . . . 205
11.1.1 GCC RUNTIME LIBRARY EXCEPTION . . . 206
11.1.2 GNU GENERAL PUBLIC LICENSE . . . 207

Index **221**

CHAPTER

ONE

INTRODUCTION

1.1 Overview

OMEGA is a tool for generating 3D conformations of molecules. **OMEGA** provides two different methods for conformation generation: torsion driving and distance geometry. The torsion driving method works best on molecules that do NOT contain large, flexible rings, while the distance geometry method works for all molecules but has been designed for molecules that do contain large, flexible rings (macrocycles).

The minimal input into **OMEGA** is a file of molecules, with all stereocenters defined (see flipper below) in any readable format. The minimal output is a file of molecules with conformations.

1.2 Applications

The **OMEGA** distribution comprises two applications:

OMEGA

- Generates conformations for molecules in the input file.

FILTER

- Removes molecules in the input file that do not conform to a set of physicochemical properties.
- Molecules should be processed through **FILTER** before using **OMEGA**.

1.3 Utility Programs

The following utility programs are also included in this distribution:

- *Flipper*: Enumerates all (unspecified) stereocenters in input molecules, ensuring that every stereocenter in a molecule is defined. These stereoisomers are then suitable for input to **OMEGA**.
- *MakeFraglib*: Generates 3D fragment geometries for rings and linkers and makes a custom library of these fragments for **OMEGA**.
- *Oeb2SdConf*: Converts **OMEGA** conformers from OEB format into a form usable by MOE. This is a helper utility for users of the MOE tool.

Conformations from **OMEGA** can be used as input into several other **OpenEye** tools, including **OEDocking**, **ROCS**, and **Freeform**.

CHAPTER

TWO

FILTER

2.1 Overview

Filtering based on graph and physical properties should always be carried out prior to generating multiconformer databases using **OMEGA**. Eliminating undesirable compounds prior to generating conformers saves execution time of both **OMEGA** and downstream applications, as well as storage space. Large polypeptides or proteins, very flexible molecules, or molecules that would never be considered useful for the ultimate modeling application are best eliminated from a dataset as soon as possible.

The most important graph filter to apply is rotatable bond count. Although **OMEGA** may be able to generate conformers for molecules with more than 20 rotatable bonds, the results of such an exercise would be dubious at best for any conformer generation method. The number of rings, especially flexible rings, should also be used to exclude irrelevant molecules. **OMEGA** will handle molecules that have many thousands of possible ring combinations, although the time expenditure may be prohibitive and the results equally questionable to molecules with an unreasonably large number of rotatable bonds. Simple element filters may be useful as well, although **OMEGA** will discard compounds for which no force field parameters exist. Using element filters beforehand may aid in tracking the rationale for discarding compounds instead of searching through **OMEGA** log files for failure modes.

The **FILTER** program from OpenEye, Cadence Molecular Sciences provides all of the functionality outlined above, and additional physical property filters. **FILTER** should be used for input file preparation of large datasets.

2.2 Example Commands

> **Note**
>
> `filter` may need to be replaced by `filter.bat` on Windows PowerShell.

This section includes several sample **FILTER** command line executions. Each example is followed by a brief description of its behavior.

```
prompt> filter drugs.smi drugs.oeb.gz
prompt> filter -in drugs.smi -out drugs.oeb.gz
```

These two commands will yield identical results. These execute **FILTER** with the default parameters. The file *drugs.smi* is opened in SMILES format for input, and the output is written to the file *drugs.oeb.gz* in gzipped OE-Binary version 2 format.

```
prompt> filter -in drugs.smi -out drugs.sdf -filter myfilter
```

This command is the same as the one above except for the *-filter* flag. It executes **FILTER** with the parameters found in the *myfilter* file. The file *drugs.smi* is opened in SMILES format for input, and the output is written to the file *drugs.sdf*.

```
prompt> filter -param myparameters drugs.smi drugs.oeb.gz
prompt> filter drugs.smi drugs.oeb.gz -param myparameters
```

The first of these two commands will yield exactly the same results as the example above. The file *drugs.smi* will be mapped to the *-in* flag and *drugs.oeb.gz* will be mapped to the *-out* flag, being the second to last and last command line arguments, respectively. Unfortunately, the second of these two commands will fail to parse because the implicit input and output arguments are not the final two arguments in the list.

```
prompt> filter -in drugs.smi -out drugs.oeb.gz -table drugs.table
```

This executes **FILTER** on the file *drugs.smi* and writes molecules that pass the filter to the file *drugs.oeb.gz*. It also writes all of the filter data to the tab-separated value file *drugs.table*.

```
prompt> cat maybridge.05-1.sdf |filter -in .sdf -out .ism|oeomega classic .ism m.oeb.gz
```

This command presumes that you have an SD format file called *maybridge.05-1.sdf*. That file is piped to the **FILTER** program. The *-in .sdf* flag indicates that **FILTER** should read .sdf format from std::in. Since no *-filter* flag is specified, the default filter will be used. The *-out .ism* flag indicates **FILTER** will write isomeric smiles format to std::out. The output would then be piped into **OMEGA**.

```
prompt> filter -in drugs.smi -out drugs.oeb.gz -select "[N;$(*-a)]"
```

This command will filter the compounds in *drugs.smi* with the default filter and write the output to *drugs.oeb.gz*. It also requires that molecules contain exactly one instance of the aniline substructure defined by the SMARTS pattern *"[N;$(*-a)]"*.

> **Note**
>
> `filter.bat -in drugs.smi -out drugs.oeb.gz "-select [N;\`$(*-a)]"` must be used instead on Windows PowerShell.

2.3 Command Line Help

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

```
prompt> filter --help
```

will generate the following output:

```
Help functions:
  filter --help simple      : Get a list of simple parameters
  filter --help all         : Get a complete list of parameters
  filter --help <parameter> : Get detailed help on a parameter
  filter --help html        : Create an HTML help file for this program
```

2.4 Required Parameters

-in

File containing one or more molecular connection tables from which you would like to remove the non-medicinal compounds. [Aliases=**-i**]

`-out`

File to fill with the molecules that pass all of the specified filters. [Aliases=**-o**]

2.5 Optional Parameters

2.5.1 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. **FILTER** generates a new parameter file containing the full set of execution parameters upon every execution. The name of the parameter file written by **FILTER** is created by combining the prefix base name with the *'.param'* extension.

2.5.2 Control Options

`-filter`

This optional parameter specifies a filter file to be used in place of the default filter. If only simple additions to the default filters are desired, please see the *-newrule* parameter. The file format for this file is described in the *Filter Files* chapter. There are several special reserved values for this parameter. If the *-filter* parameter is:

- **Blockbuster** This is the recommended and default filter
- **Lead** Corresponds to the [Oprea-2000] lead-like filters useful for preparing HTS databases.
- **Drug** Corresponds to the [Oprea-2000] drug-like filters useful for later stage drug development.This is provided as a backwards compatible option. Experience has shown it to be too restrictive
- **Macrocycle** Non-macrocycle molecules will be removed
- **Nonmacrocycle** Macrocycle molecules will be removed
- **Frag** Filter for extracting small molecules from a database that have attachment points
- **Pains** Corresponds to the [Baell-2010] filter useful for removing common substructures that interfere with bioassays

The default filter files for these reserved types can be found in the data directory included with the applications bundle. [Default=blockbuster]

`-fail`

This specifies an optional molecular output file where the molecules that fail to pass the filter will be written. If this parameter is specified, then every molecule from *-in* will either be written to *-out* or to *-fail*. [Default=null]

`-prefix`

For an execution of the FILTER program, three general purpose files are written in addition to the output file specified on the command-line. These file are the "info", "log" and "param" files. Normally, they all begin with the prefix "filter". However, this can be overridden with the *-prefix* parameter. This is particularly useful if you want to run multiple FILTER jobs in the same directory without overwriting files. [Default=filter]

`-info`

Normally, **FILTER** writes an info file during the progress of any execution. At regular intervals during the execution, the info file is updated to reflect the most recent progress. If you want to see the progress of a run, it is best to either use the *-dots* parameter or look at the info file. Normally, the info file is saved as *filename.info* where "filename" is the prefix specified by the *-prefix* flag. However, one may use the *-info* flag to specify an info file with a separate name. [Default=null]

`-interval`

This is the interval at which data is written to the *filter.info* file. The *filter.info* file contains running totals that are relevant to a **FILTER** run. Examining the *filter.info* file is the best means of checking on the progress of a **FILTER** execution. If this flag is 50, then the filter.info file is rewritten every 50 molecules. [default = 5000]

`-log`

This flag determines where the logging information is written. The logging information includes a listing of the filter used, followed by a one-line comment about why each molecule failed, or, if it passed, an assessment of the probability that the compound lies in drug-like space.

`-mmff`

This flag corresponds to the predefined force field, which is an exact reproduction of the published Merck Molecular Force Field (MMFF94) with additional atom types parameterized using the same algorithm ([Halgren-I-1996]; [Halgren-II-1996]; [Halgren-III-1996]; [Halgren-IV-1996]; [Halgren-V-1996]). If true, remove any compounds that cannot be MMFF atom-typed. [default = false]

`-newrule`

This optional parameter specifies a file that contains filter rules to supplement the default filter or the filter specified with the *-filter* parameter. This parameter can be used to extend the functional group list used to filter. New filter rules can also be added directly to the filter file specified with *-filter*. [Default=null]

`-typecheck`

This boolean flag controls whether the valence states of atoms will be checked. This check identifies molecules that are poorly specified, or represent nonsensical chemical states. For example, an oxygen with eight hydrogens attached or a carbon with a +9 formal charge would be rejected. [Default=true]

`-select`

This parameter is a SMARTS string that allows a user to require a specific functional group or substructure be present in all molecules that pass the filter. This feature is particularly useful for identification of reagents for library design. Selection items can also be added directly to the filter file specified with *-filter*. The command-line argument only allows specification of the SMARTS pattern, and exactly one copy of that functional group is required. If a user wants to specify a selection SMARTS with minimum and maximum number of occurrences other than 1, then they can use a SELECT statement inside the filter file. [Default=null]

`-log`

This flag determines where the logging information is written. The logging information includes a listing of the filter used, followed by a one line comment about why each molecule failed, or if it passed, an assessment of the probability that the compound lies in drug-like space.

`-table`

This flag specifies a file for a tab-separated format table that includes all of the filter data. These data files are ready for import into a spreadsheet program for easy examination. Each column of the table includes one of the filter categories (such as "Molecular Weight") and each row of the table corresponds to a single molecule. The table contains complete entries for all of the molecules in the input file regardless of whether they pass or fail. NOTE: Setting this flag will cause the program to slow down. [Default=null] [Aliases=**-Table**]

`-flagTableFailures`

This flag specifies that, if a table is being written, any values in the table that would cause a molecule to fail a filter will be flagged with an asterisk. This provides a means of seeing all the filters a molecule might fail, as the log file typically only provides the first failure. [Default=false] [Aliases=**-tableFlag**,**-tableflag**,**-TableFlag**,**-tflag**,**-tFlag**]

`-interval`

This is the interval at which data is written to the *filter.info* file. The *filter.info* file contains running totals that are relevant to a FILTER run. Examining the *filter.info* file is the best means of checking on the progress of a FILTER execution. If this flag is 50, then the filter.info file is re-written every 50 molecules. [Default=5000]

-pkanorm

This boolean flag determines whether compounds will be modified to reflect a pH=7.4 model. Notice, this will modify the molecules permanently. [Default=true] [Aliases=**-pKanorm**,**-pKaNorm**]

-normalize

This flag indicates an optional SMIRKS file. This file should contain the set of reactions you wish to use to normalize the connection table of your molecules. Please note: These reactions are applied before the filtering process and can significantly slow the filtering process. [Default=null] [Aliases=**-norm**,**-Norm**]

-salt

This flag specifies a molecule file that you consider to include salts. If any molecular entries contain multiple disconnected fragments, then any fragment contained in the salt file will be removed. If no file is specified, or if there are multiple disconnected fragments in a molecule record that are not in the salt file, then the first largest remaining fragment will be retained and all others discarded. [Default=null]

-sdtag

This boolean flag indicates whether you want the molecular properties used for the filtering run (see *-filter*) to be attached to output molecules as SD tag data. This parameter will only work for *.sdf* or *.oeb* formats. [Default=false] [Aliases=**-SDtag**,**-SDTag**]

-dots

Boolean flag that determines whether FILTER writes a single dot (.) to the terminal (stdout) for every 500 compounds that are processed. [Default=true]

-unique

This flag accepts the name of a file that contains molecules for **FILTER** to skip. Only unique molecules that do not appear in this file will be sent to the output. Molecules are checked for uniqueness after **FILTER** has processed them; therefore, a parameter such as -pkanorm could change an input structure from a duplicate to a unique molecule.

-mmff

If true, remove any compounds that cannot be MMFF atom-typed. [Default=false]

2.6 Filter Files

A user can provide two parameter files if they would like to override or augment the default parameter sets. The first is a *filter file*. It provides acceptable limits for all of the physical properties and functional groups in the default filter. The second is a `NEWRULE` file, which allows you to augment a filter with a set of additional rules.

There are four types of statements that can occur in a filter file:

- Physical property limits
- Rules
- New rules
- Selections

The statements should occur one per line in the filter file.

> **Note**
>
> If the appropriate line is not in the filter file, or the value is false, the respective measure will not be used in filtering and its value will not be included in any table-based output.

2.6.1 Physical Property Limits

There are a large number of physical property limits. They occur as three fields on a line. For example:

```
MIN_HETEROATOMS 2 "Minimum number of heteroatoms"
```

The first field is the property keyword, the second field is the value assigned to that keyword, and the third field is a brief informational message. There are a fixed number of physical property keywords. No additional physical property keywords can be added by the user. The current keywords and brief definitions of each are listed below.

Hint

The values listed below are those found in the default *BlockBuster* filter.

Basic Properties

Molecular Weight

Isotopic molecular weight.

```
MIN_MOLWT 130 "Minimum molecular weight"
MAX_MOLWT 781 "Maximum molecular weight"
```

Heavy Atom Count

Number of non-hydrogen atoms.

```
MIN_NUM_HVY 9 "Minimum number of heavy atoms"
MAX_NUM_HVY 55 "Maximum number of heavy atoms"
```

Carbon Count

Number of carbons.

```
MIN_CARBONS 3 "Minimum number of carbons"
MAX_CARBONS 41 "Maximum number of carbons"
```

Hetero Count

Number of non-carbon and non-hydrogen atoms.

```
MIN_HETEROATOMS 1 "Minimum number of heteroatoms"
MAX_HETEROATOMS 14 "Maximum number of heteroatoms"
```

Heteroatom to Carbon Ratio

Hetero count/carbon count.

```
MIN_Het_C_Ratio 0.04 "Minimum heteroatom to carbon ratio"
MAX_Het_C_Ratio 4.0 "Maximum heteroatom to carbon ratio"
```

Chiral Count

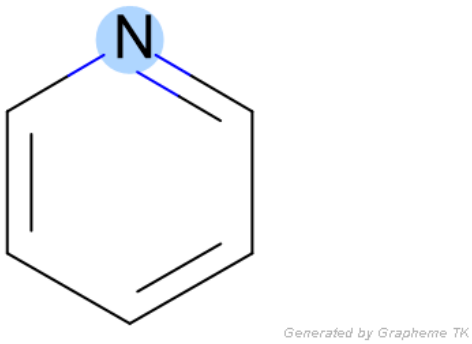
Number of chiral atoms.

```
MIN_CHIRAL_CENTERS 0 "Minimum chiral centers"
MAX_CHIRAL_CENTERS 21 "Maximum chiral centers"
```

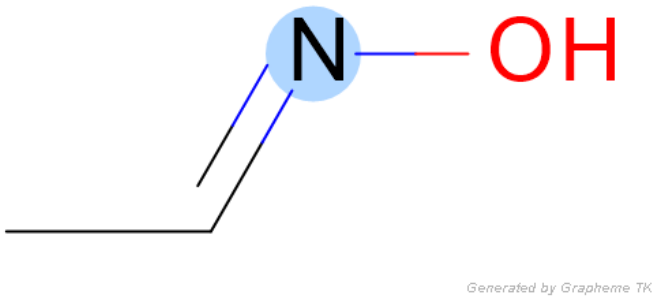
Hydrogen Bond Acceptors

Number of atoms which match any of the following:

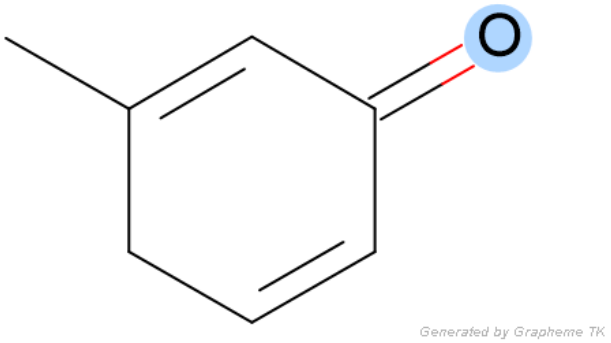
- Degree 2, aromatic, nonpositive nitrogens. An example is shown below.



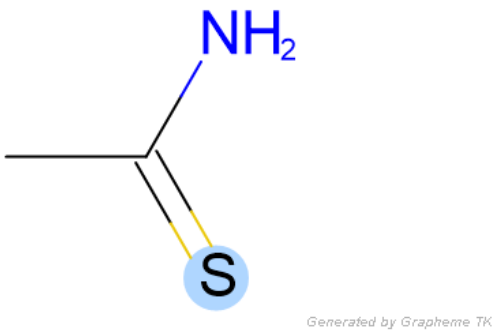
- Electron rich (negative), valence less than 4, nonaromatic nitrogens. An example is shown below.



- Negatively charged (does not withdraw electrons), neutral, nonaromatic oxygens. An example is shown below.



- Degree 1, double bonded, electron rich, nonaromatic sulfur. An example is shown below.



This definition is from the work of Mills and Dean [Mills-Dean-1996] and also by Jeffrey [Jeffrey-1997].

```
MIN_HBOND_ACCEPTORS 0 "Minimum number of hydrogen-bond acceptors"
MAX_HBOND_ACCEPTORS 13 "Maximum number of hydrogen-bond acceptors"
```

See also

- OEGetHBondAcceptorCount function
- *Lipinski and Hydrogen Bonds* section

Hydrogen Bond Donors

Number of hydrogen atoms on nitrogen, oxygen, or sulfur atoms. This definition is from the work of Mills and Dean ([Mills-Dean-1996]) and also the book by Jeffrey ([Jeffrey-1997]).

```
MIN_HBOND_DONORS 0 "Minimum number of hydrogen-bond donors"
MAX_HBOND_DONORS 9 "Maximum number of hydrogen-bond donors"
```

See also

- OEGetHBondDonorCount function

- *Lipinski and Hydrogen Bonds* section

Lipinski Acceptors

Number of nitrogens or oxygens. This definition is from the work of Lipinski [Lipinski-1997].

```
MIN_LIPINSKI_ACCEPTORS 1 "Minimum number of oxygen & nitrogen atoms"
MAX_LIPINSKI_ACCEPTORS 14 "Maximum number of oxygen & nitrogen atoms"
```

See also

- OEGetLipinskiAcceptorCount function
- *Lipinski and Hydrogen Bonds* section

Lipinski Donors

Number of nitrogens and oxygens with at least one hydrogen attached. This definition is from the work of Lipinski [Lipinski-1997].

```
MIN_LIPINSKI_DONORS 0 "Minimum number O & N atoms with hydrogens"
MAX_LIPINSKI_DONORS 6 "Maximum number O & N atoms with hydrogens"
```

See also

- OEGetLipinskiDonorCount function
- *Lipinski and Hydrogen Bonds* section

Halide Fraction

Percent of molecular weight from halides.

```
MIN_HALIDE_FRACTION 0.0 "Minimum Halide Fraction"
MAX_HALIDE_FRACTION 0.66 "Maximum Halide Fraction"
```

See also

- OEGetHalideFraction function

Formal Count

Number of atoms with a formal charge (excludes dative).

```
MIN_COUNT_FORMAL_CRG 0 "Minimum number formal charges"
MAX_COUNT_FORMAL_CRG 4 "Maximum number of formal charges"
```

Formal Sum

Total formal charge.

```
MIN_SUM_FORMAL_CRG -2 "Minimum sum of formal charges"
MAX_SUM_FORMAL_CRG 2 "Maximum sum of formal charges"
```

Connected Non-Ring

Considers sets of contiguous (bonded) non-ring atoms.

```
MIN_CON_NON_RING 0 "Minimum number of connected non-ring atoms"
MAX_CON_NON_RING 19 "Maximum number of connected non-ring atoms"
```

Unbranched Chains

The size of the longest chain of either heavy atoms or of all carbons. An unbranched atom is an atom with at most two connections to other heavy atoms and is not in a ring. A set of unbranched atoms which are connected together form a chain. A molecule may contain multiple chains which are isolated from each other by non-chain atoms (e.g., ring or branched atoms). The longest chain of heavy atoms and the longest chain of carbons are identified, and the MIN and MAX parameter filters are applied.

```
MIN_UNBRANCHED 1 "Minimum number of connected unbranched non-ring atoms"
MAX_UNBRANCHED 13 "Maximum number of connected unbranched non-ring atoms"
MIN_UNBRANCHED_C 0 "Minimum number of connected unbranched non-ring carbon"
MAX_UNBRANCHED_C 6 "Maximum number of connected unbranched non-ring carbon"
```

See also

- OEGetLongestUnbranchedHeavyAtomsChain function
- OEGetLongestUnbranchedCarbonsChain function

Total Functional Group Count

Total number of functional groups. Does not count any ring systems as functional groups. Degree 1 heteroatoms, particularly those with double bonds or dative bonds are considered part of ring systems and do not count as functional groups.

```
MIN_FCNGRP 0 "Minimum number of functional groups"
MAX_FCNGRP 7 "Maximum number of functional groups"
```

Note

This is different than the functional group rules.

Ring Systems

Number of ring systems (contiguous systems of ring atoms and bonds).

```
MIN_RING_SYS 0 "Minimum number of ring systems"
MAX_RING_SYS 5 "Maximum number of ring systems"
```

Ring Size

Maximum size of any single ring system.

```
MIN_RING_SIZE 0 "Minimum atoms in any ring system"
MAX_RING_SIZE 20 "Maximum atoms in any ring system"
```

Rotor Count

Number of rotatable bonds. Allows optional adjustment for aliphatic rings following the method of Oprea [Oprea-2000].

```
MIN_ROT_BONDS 0 "Minimum number of rotatable bonds"
MAX_ROT_BONDS 16 "Maximum number of rotatable bonds"
ADJUST_ROT_FOR_RING true "BOOLEAN for whether to estimate degrees of freedom in rings"
```

See also

- OEGetRotatableBondCount function

Rigid Count

Number of rigid bonds (non-rotatable bonds).

```
MIN_RIGID_BONDS 4 "Minimum number of rigid bonds"
MAX_RIGID_BONDS 55 "Maximum number of rigid bonds"
```

Unspecified Atom Stereo

Number of unspecified atom stereos

```
MIN_UNSPECIFIED_ATOM_STEREOS 0 "Minimum number of unspecified atom stereos"
MAX_UNSPECIFIED_ATOM_STEREOS 2 "Maximum number of unspecified atom stereos"
```

Note

`MIN_UNSPECIFIED_ATOM_STEREOS` and `MAX_UNSPECIFIED_ATOM_STEREOS` are not used for filtering in the default filters.

See also

- OEGetNumUnspecifiedAtomStereos function
- stereo atom
- Stereochemistry Perception chapter

- `OEAtomBase.HasStereoSpecified`

Unspecified Bond Stereo

Number of unspecified bond stereos

```
MIN_UNSPECIFIED_BOND_STEREOS 0 "Minimum number of unspecified bond stereos"
MAX_UNSPECIFIED_BOND_STEREOS 2 "Maximum number of unspecified bond stereos"
```

Note

`MIN_UNSPECIFIED_BOND_STEREOS` and `MAX_UNSPECIFIED_BOND_STEREOS` are not used for filtering in the default filters.

See also

- OEGetNumUnspecifiedBondStereos function
- stereo bond
- Stereochemistry Perception chapter
- `OEBondBase.HasStereoSpecified`

Atom Type Checks

Checks the validity of atom charges, valences or MMFF atom types for the entire molecule.

```
TYPECHECK     false "Screen for unusual valences or charges"
MMFFTYPECHECK false "Screen for atoms with unknown MMFF atom types"
```

See also

- *Type Checking*
- SetTypeCheck
- *MMFF94 Atom Type Checking*
- SetMMFFTypeCheck

LogP

The logP calculation is a derivative of the published XLOGP algorithm [Wang-R-1997] but is reparameterized without the dependence on 3D coordinates or the SYBYL/Mol2 aromaticity model.

XLogP

Calculated LogP.

```
MIN_XLOGP -3.0 "Minimum XLogP"
MAX_XLOGP 6.85 "Maximum XLogP"
```

Solubility

The solubility predictions are based on the atom types from the XLOGP algorithm (see [Wang-R-1997]) and on reparameterized atom types based on available solubility data. Rather than a quantitative cutoff, the solubility uses six allowable categories:

1. Insoluble
2. Poorly
3. Moderately
4. Soluble
5. Very
6. Highly

These categories are keywords used in the following filter files.

Solubility

Calculated solubility class.

```
MIN_SOLUBILITY insoluble "Minimum solubility"
```

Pharmacokinetic Predictors

Several secondary filters that are built upon published combinations of simpler properties are available.

Note

All of these properties are used for filtering in the default filters.

Anionic Carbon Count

Number of anionic carbons

```
MIN_ANION_C 0 "Minimum number of anionic carbons"
MAX_ANION_C 2 "Maximum number of anionic carbons"
```

Note

`MIN_ANION_C` and `MAX_ANION_C` are not used for filtering in the default filters.

See also

- OEGetAnionicCarbonCount function

Lipinski Violations

Number of allowable Lipinski violations. A single Lipinski violation is considered acceptable. The published work allows compounds to pass with a single violation but not multiple violations [Lipinski-1997].

```
MAX_LIPINSKI 3 "Maximum number of Lipinski violations"
```

See also

The *Lipinski theory* section in the Molecular Properties and Predictors chapter.

PSA

Peter Ertl's topological polar surface area (phosphorus and sulfur areas are optional) [Ertl-2000].

```
PSA_USE_SandP false "Count S and P as polar atoms"
MIN_2D_PSA 0.0 "Minimum 2-Dimensional (SMILES) Polar Surface Area"
MAX_2D_PSA 205.0 "Maximum 2-Dimensional (SMILES) Polar Surface Area"
```

See also

The *PSA theory* section in the Molecular Properties and Predictors chapter.

GSK/Veber

Veber's measure of bioavailability (PSA > 140 or Rotatable bonds >10) [Veber-2002].

```
GSK_VEBER false "PSA>140 or >10 rot bonds"
```

Abbott/Martin

Yvonne Martin's Abbott Bioavailability Score [Martin-2005]. This is reported as a probability that F>10% in rats.

```
MIN_ABS 0.11 "Minimum probability F>10% in rats"
```

Pharmacopia/Egan

"Egan egg" measure of bioavailability (LogP >5.88 or PSA > 131.6) [Egan-2000].

```
PHARMACOPIA false "LogP > 5.88 or PSA > 131.6"
```

Aggregators

Aggregators are small molecules that can interfere with assay results by sequestering proteins in an aggregation of small molecules in solution. They appear to have activity in many assays, but in fact are usually not specific inhibitors of the protein in question. This includes two measures of whether a molecule is one of the aggregators defined by Shoichet et al. ([McGovern-2003]; [Seidler-2003]). The first measure, AGGREGATORS, states whether the molecule is an exact match to one of the approximately 400 published aggregators. The second measure, PRED_AGG, states whether the molecule is included in Shoichet's QSAR model for predicting aggregators.

Aggregators

Whether a compound is known or predicted to aggregate in concentrations common in virtual screening.

```
AGGREGATORS true "Eliminate known aggregators"
PRED_AGG false "Eliminate predicted aggregators"
```

Elemental Filters

The elemental filters are applied in this order:

1. Test for the existence of any of the metals in the `ELIMINATE_METALS` filter in the molecule.
2. Remove salts by stripping away all the disconnected components except for the largest.
3. Test to make sure only atoms specified in `ALLOWED_ELEMENTS` filter are in the molecule.

See also

- *Metal Removal*
- *Salt Removal*

The format of the two elemental filter fields is the keyword followed by a comma-delimited list of atomic symbols.

Eliminate Metals

Any molecule with the atoms indicated in `ELIMINATE_METALS` fail to pass the filter.

```
ELIMINATE_METALS Sc,Ti,V,Cr,Mn,Fe,Co,Ni,Cu,Zn,Y,Zr,Nb,Mo,Tc,Ru,Rh,Pd,Ag,Cd
```

Allowed Elements

Molecules with atoms other than those specified by `ALLOWED_ELEMENTS` fail to pass the filter.

```
ALLOWED_ELEMENTS H,C,N,O,F,P,S,Cl,Br,I
```

Aromatic Ring Count

Uses the result of the function OEGetAromaticRingCount (see toolkit docs) to filter molecules based on their number of aromatic rings.

```
MIN_AROMATIC_RING_COUNT 1 "Minimum aromatic ring count"
MAX_AROMATIC_RING_COUNT 6 "Maximum aromatic ring count"
```

CSP3 Carbon Fraction

Uses the result of the OEGetFractionCsp3 function from MolProp TK to filter molecules based on their CSP3 carbon fraction.

```
MIN_FRACTION_CSP3 0.1 "Minimum fraction CSP3 carbons"
MAX_FRACTION_CSP3 0.5 "Maximum fraction CSP3 carbons"
```

2.6.2 Functional Group Rules

Rules statements set the limits for the maximum number of the specified type of functional group that may be allowed in the molecule.

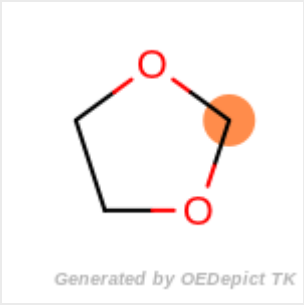
The first field of a rule statement is the word RULE in all capital letters. The second field is a number indicating the maximum number of the group allowed in a molecule. The third field is the functional group keyword. Functional group keywords are case sensitive.

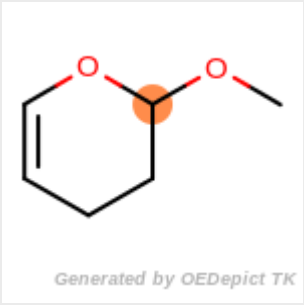
```
RULE 0 acid_halide
```

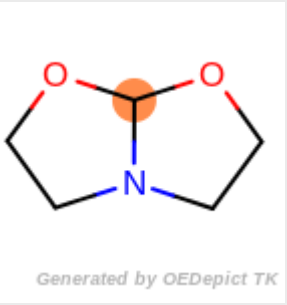
The following is a list of functional groups which the filter recognizes by default. Three example matches are provided with the atoms that correspond to each other highlighted.

> **Note**
>
> Due to the highly complex nature of the patterns, in particular recursive SMARTS, it is not possible to fully highlight every atom that was included as part of the match.

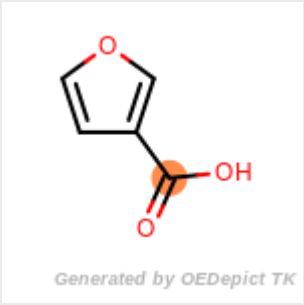
acetal

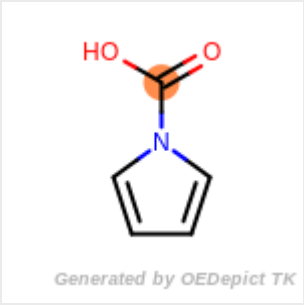


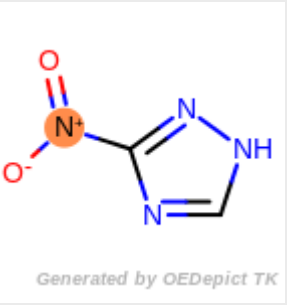




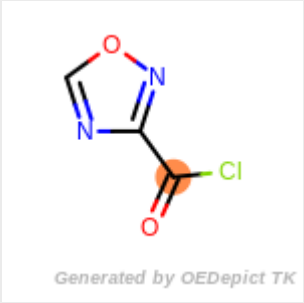
acid

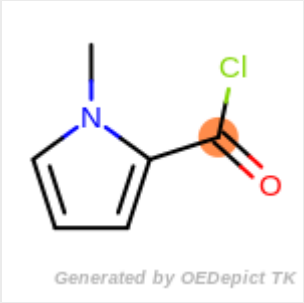


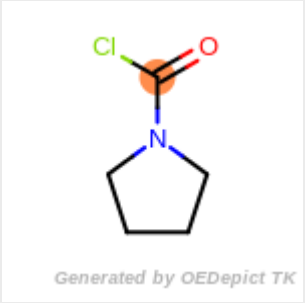




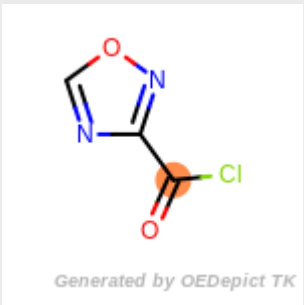
acid_chloride

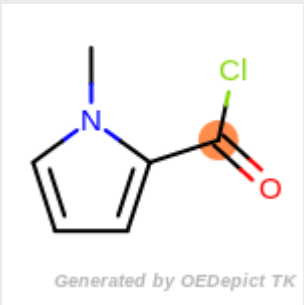


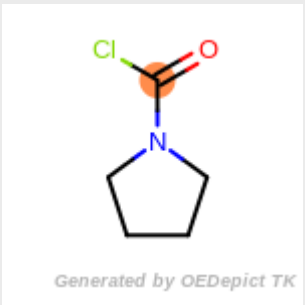




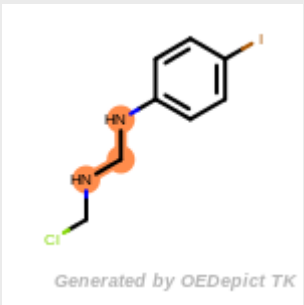
acid_halide

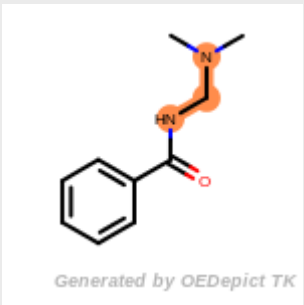


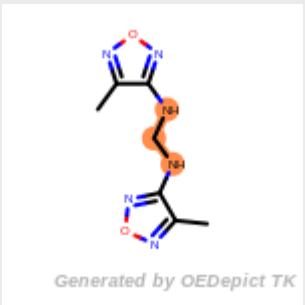




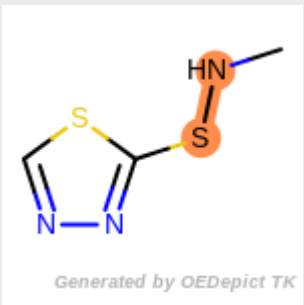
acyclic_NCN

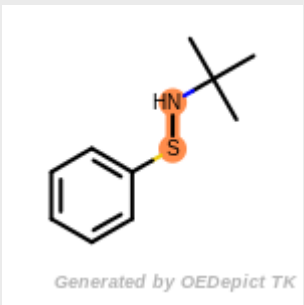


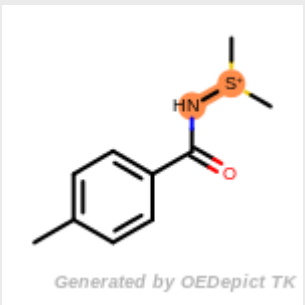




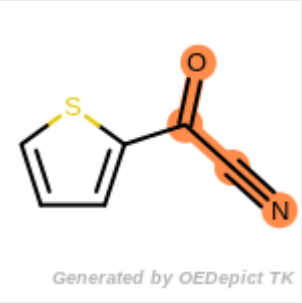
acyclic_NS

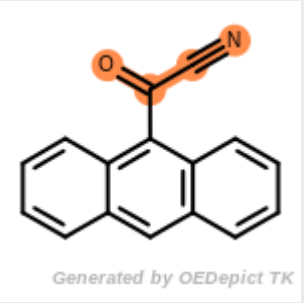


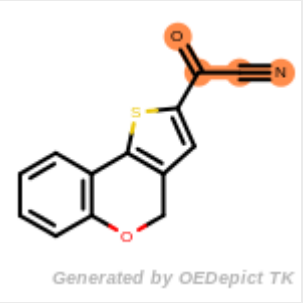




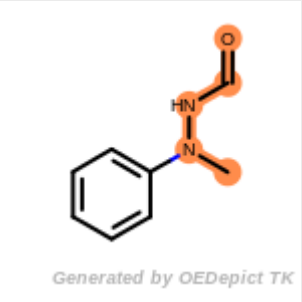
acyl_cyanides

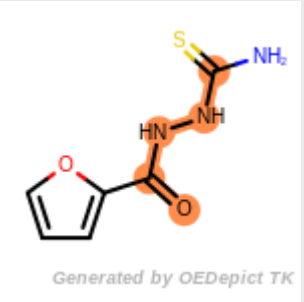


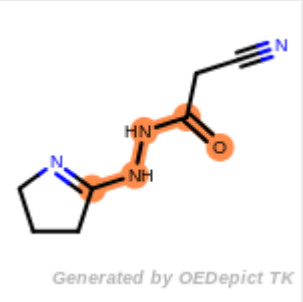




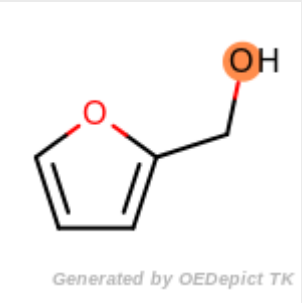
acylhydrazide

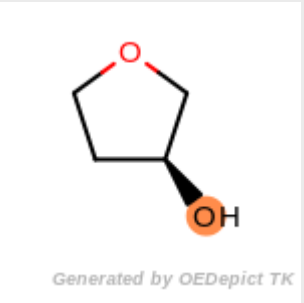


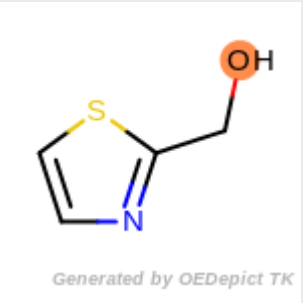




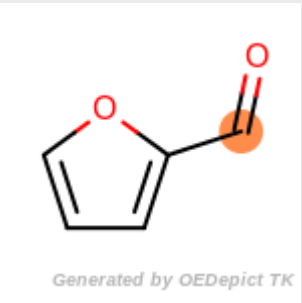
alcohol

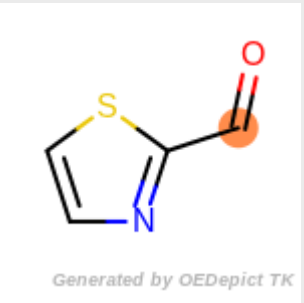


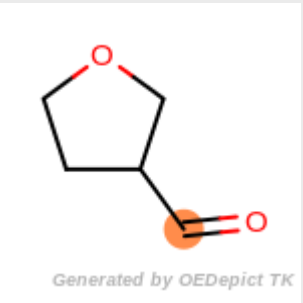




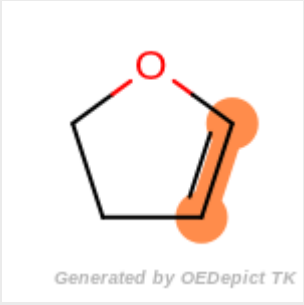
aldehyde

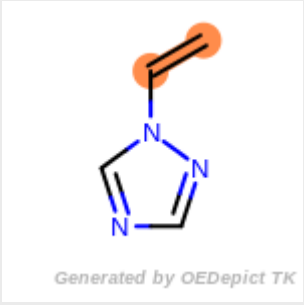


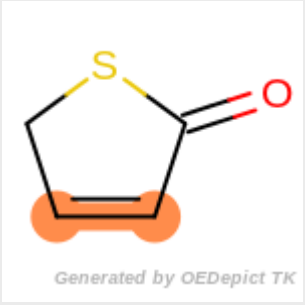




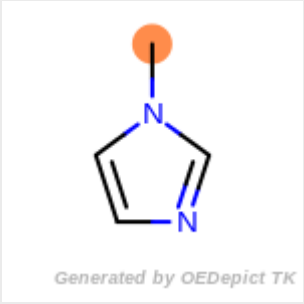
alkene

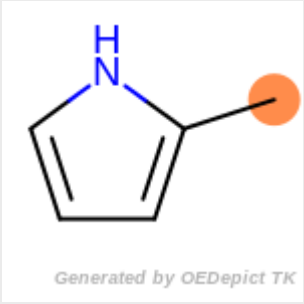


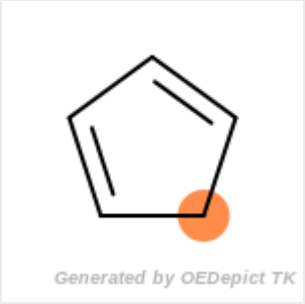




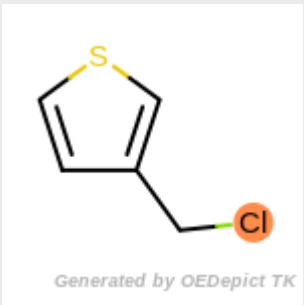
alkyl

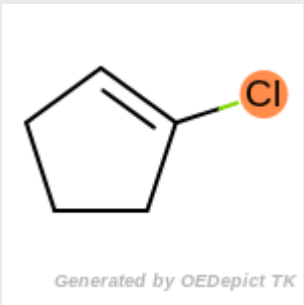


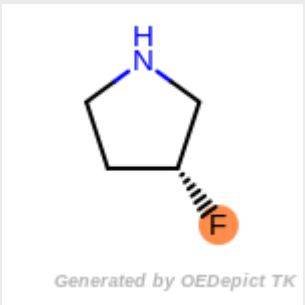




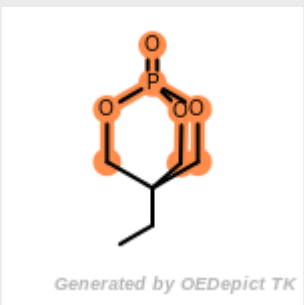
alkyl_halide

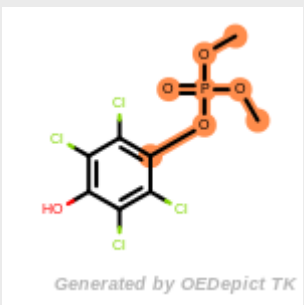


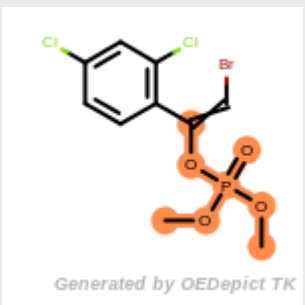




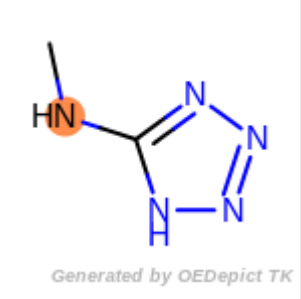
alkyl_phosphate

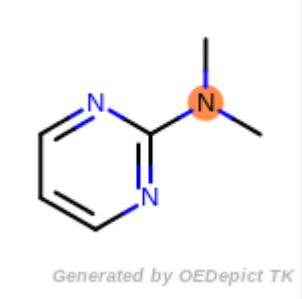


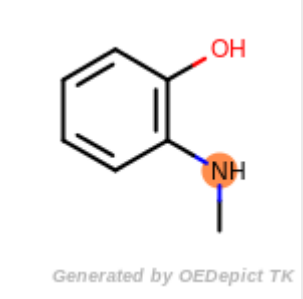




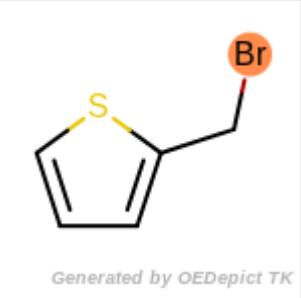
alkylaniline

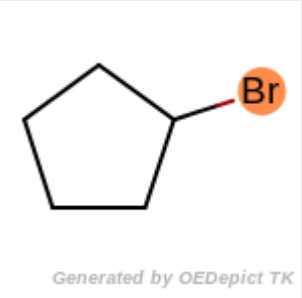


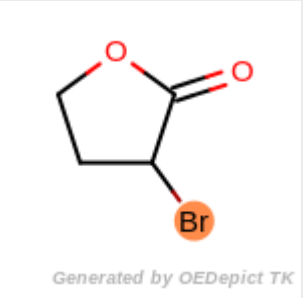




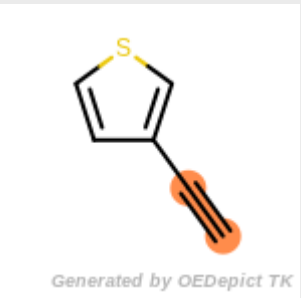
alkylating_agent

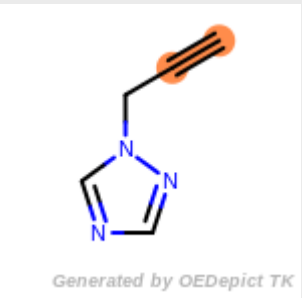


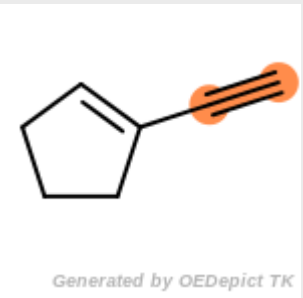




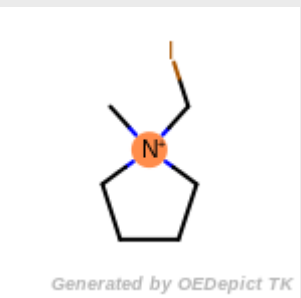
alkyne

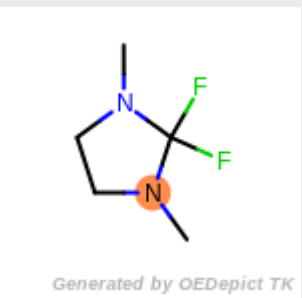


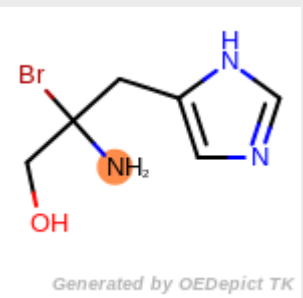




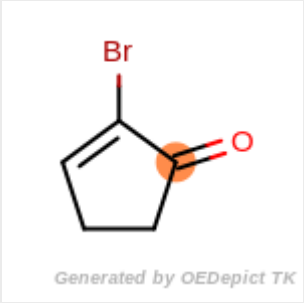
alphahalo_amine

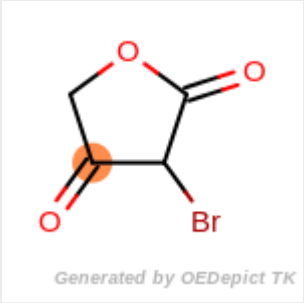


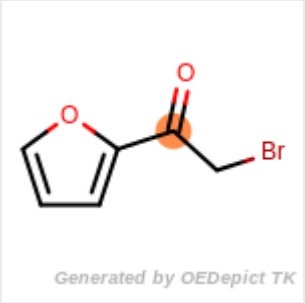




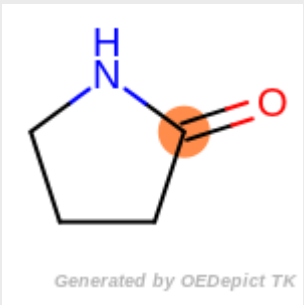
alphahalo_ketone

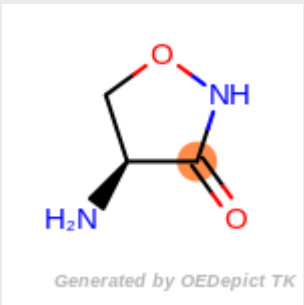


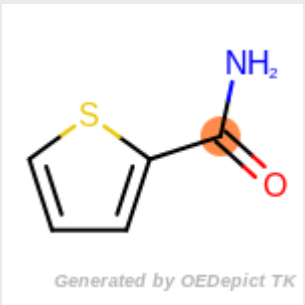




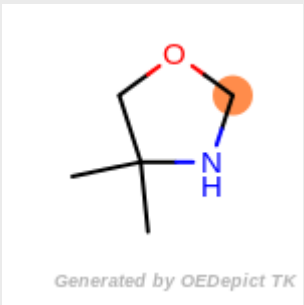
amide

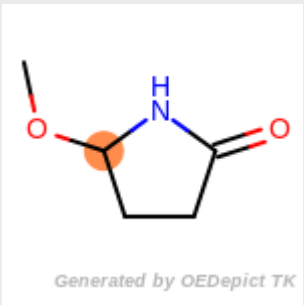


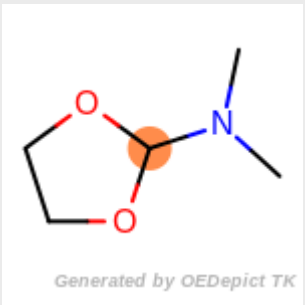




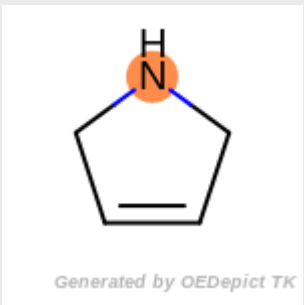
aminal

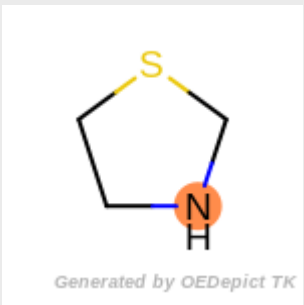


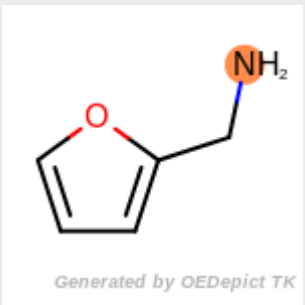




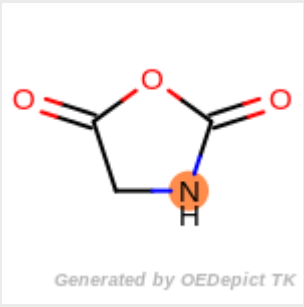
amine

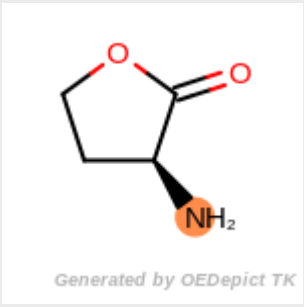


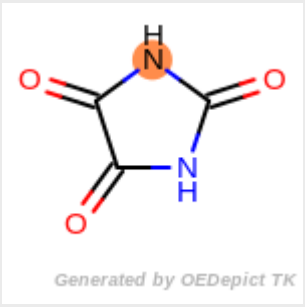




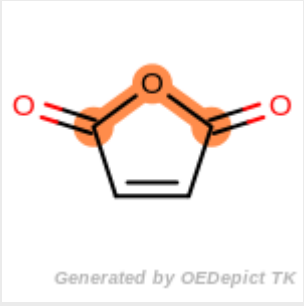
amino_acid

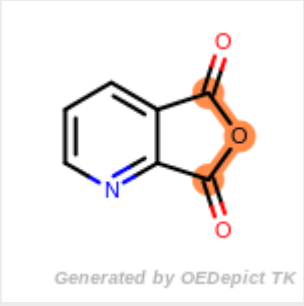


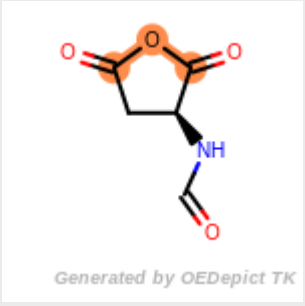




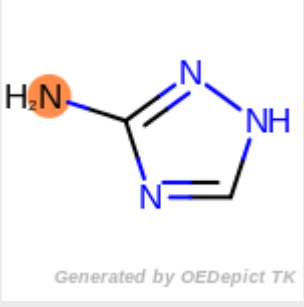
anhydride

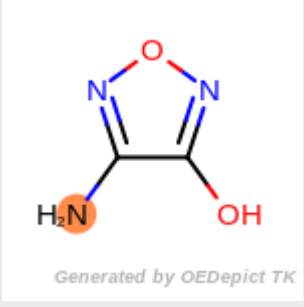


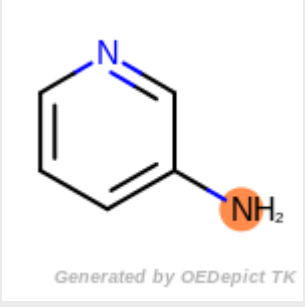




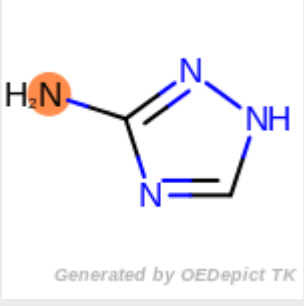
aniline

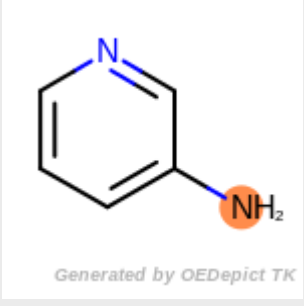


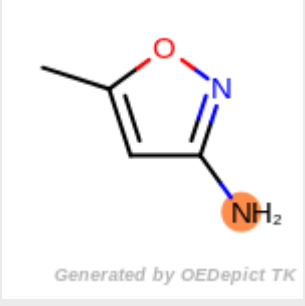




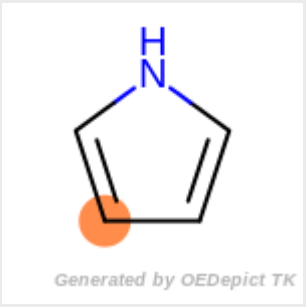
aniline_unsubstituted

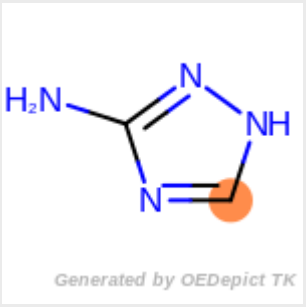


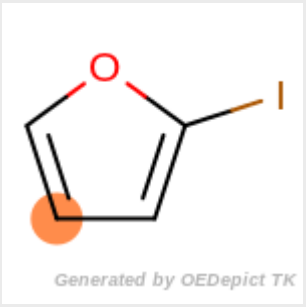




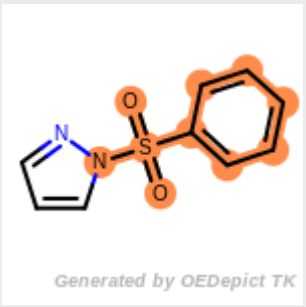
arene

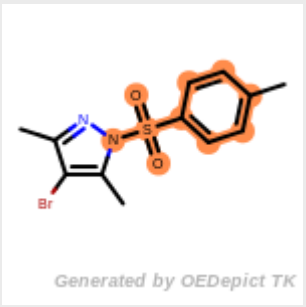


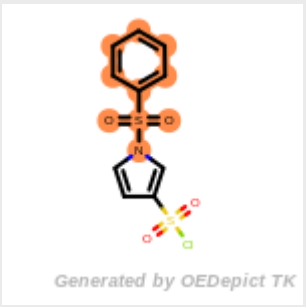




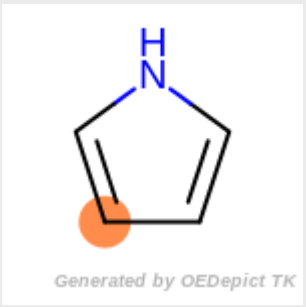
arenesulfonyl

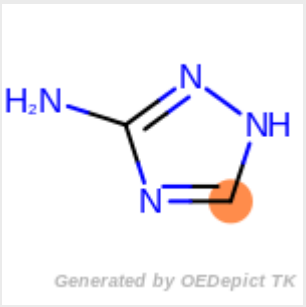


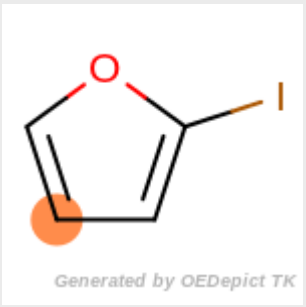




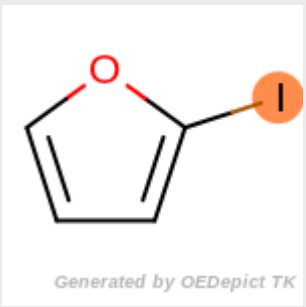
aryl

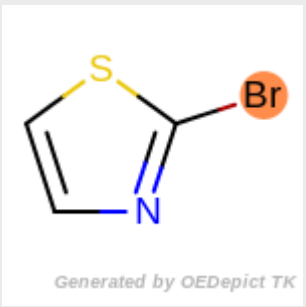


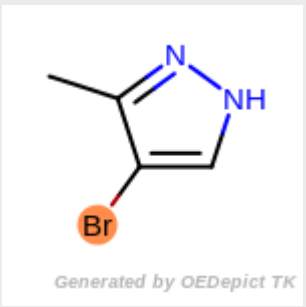




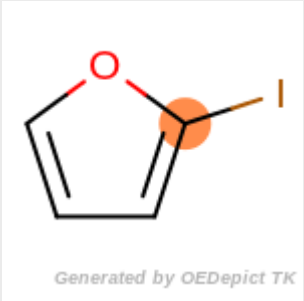
aryl_halide

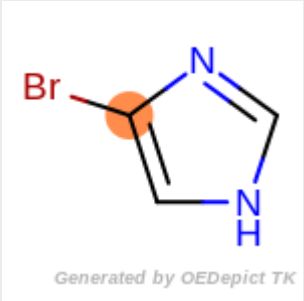


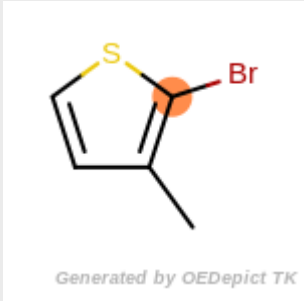




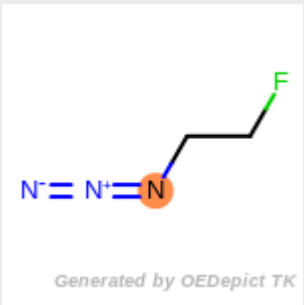
aryl_mono_BrI

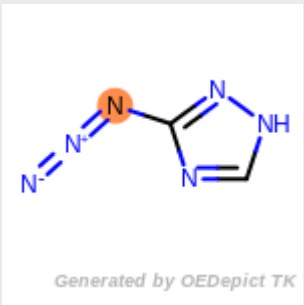


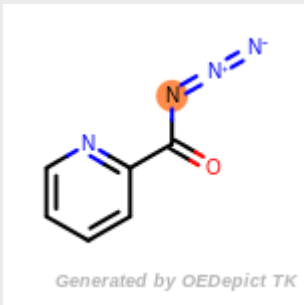




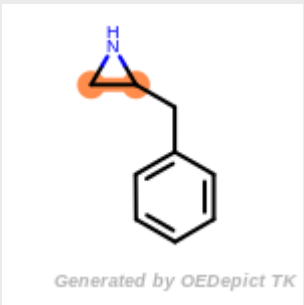
azide

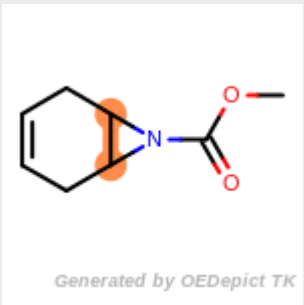


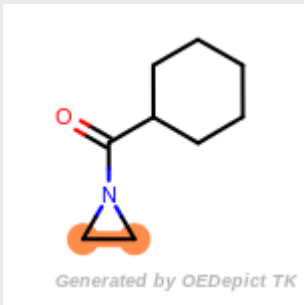




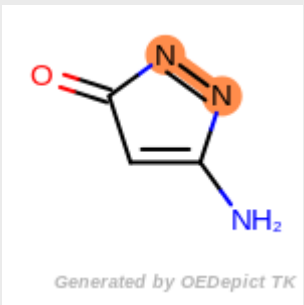
aziridine

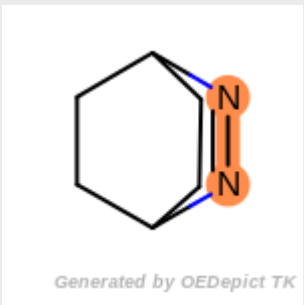


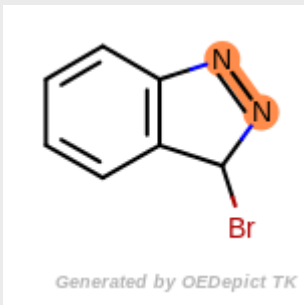




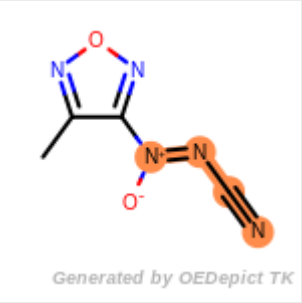
azo

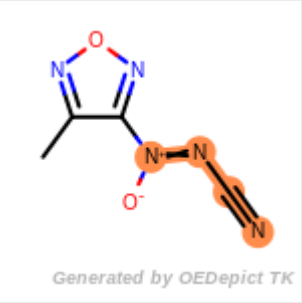


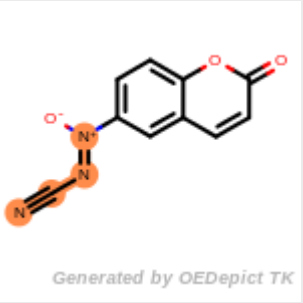




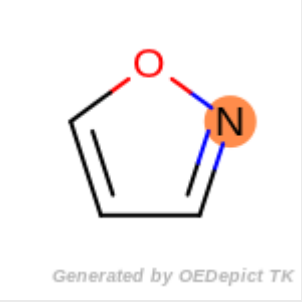
azocyanamides

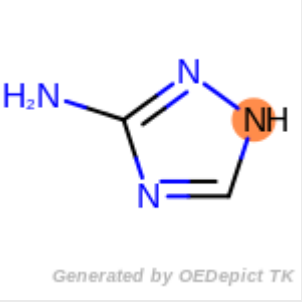


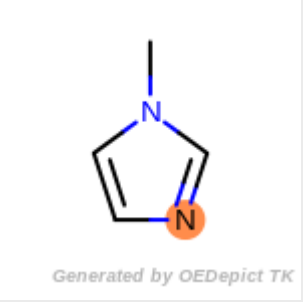




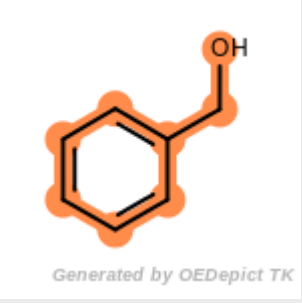
base

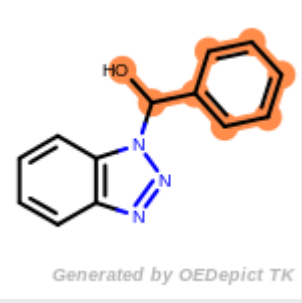


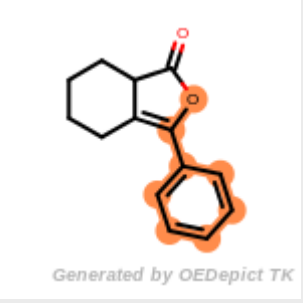




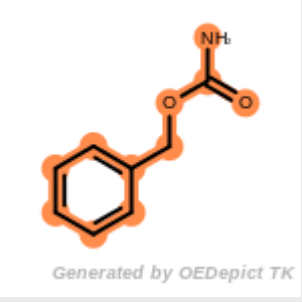
benzyl_ether

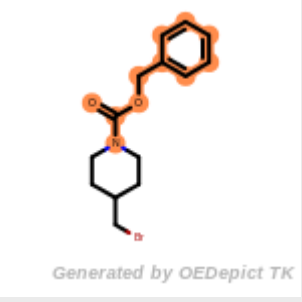


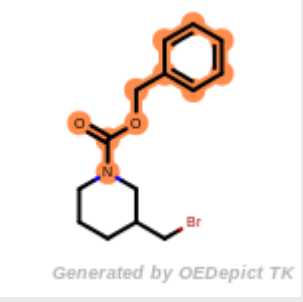




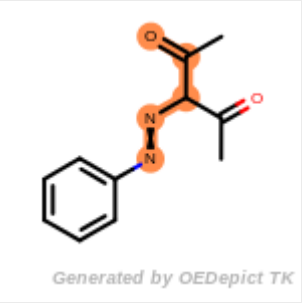
benzyloxycarbonyl_CBZ

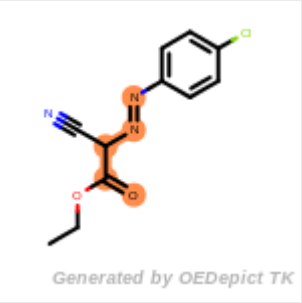


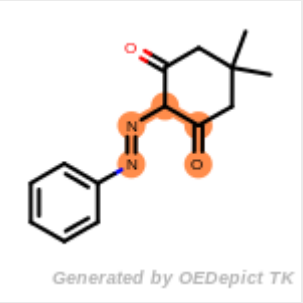




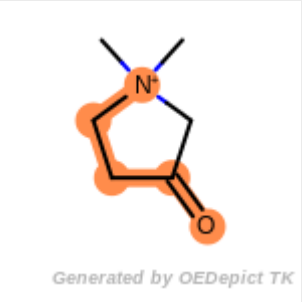
beta_azo_carbonyl

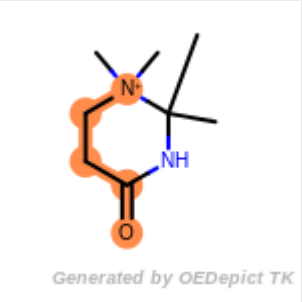


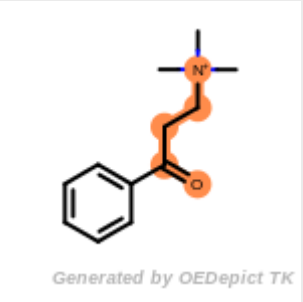




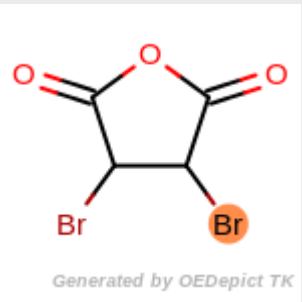
beta_carbonyl_quat_nitrogen

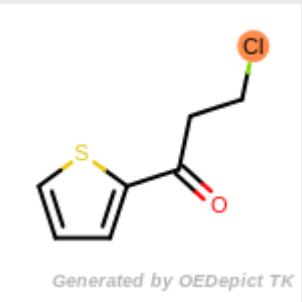


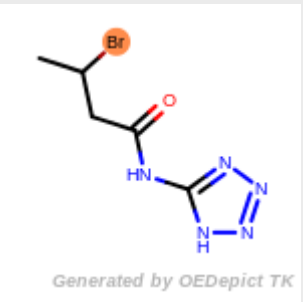




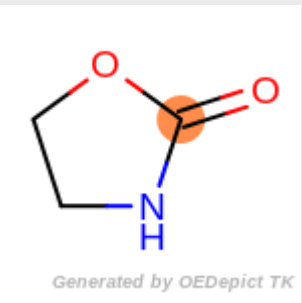
beta_halo_carbonyl

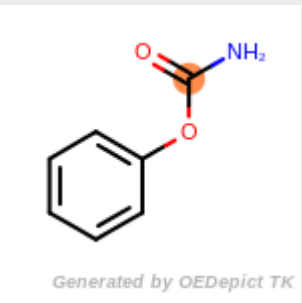


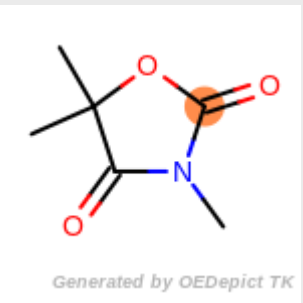




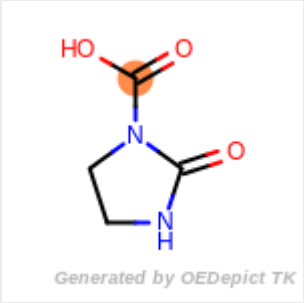
carbamate

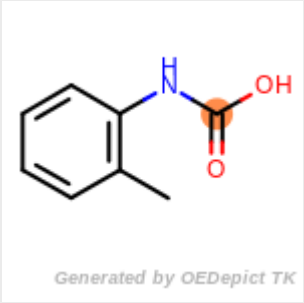


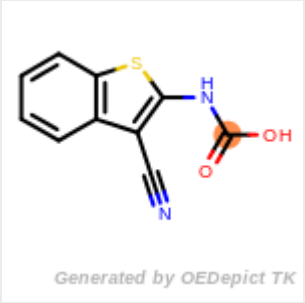




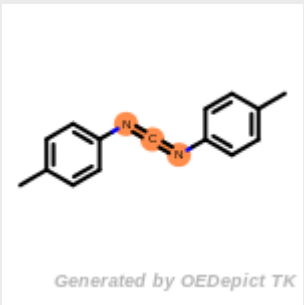
carbamic_acid

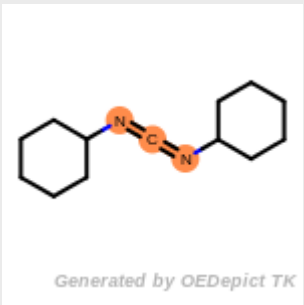


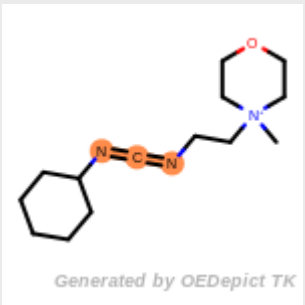




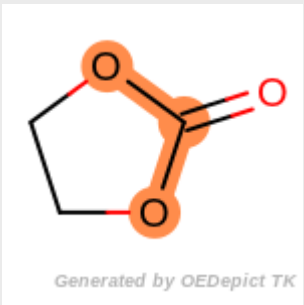
carbodiimide

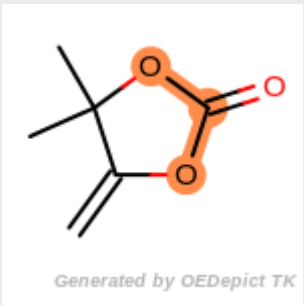


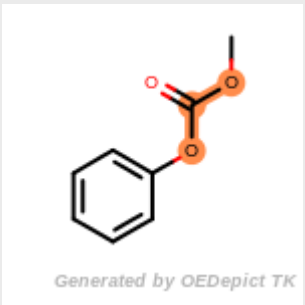




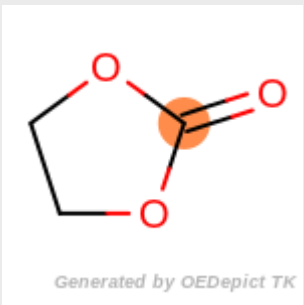
carbonate

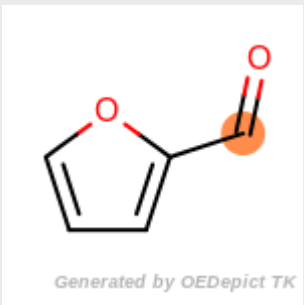


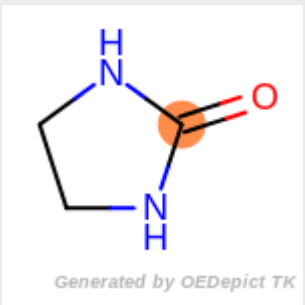




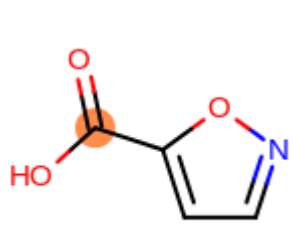
carbonyl

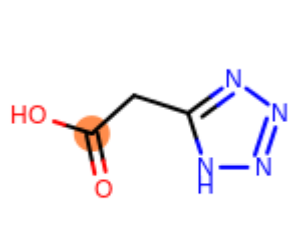


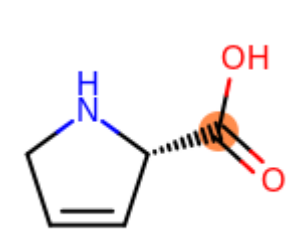




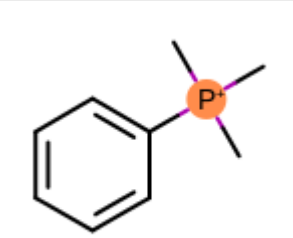
carboxylic_acid

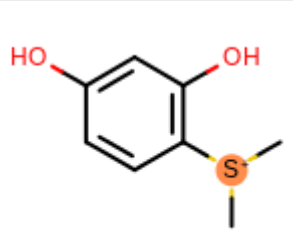


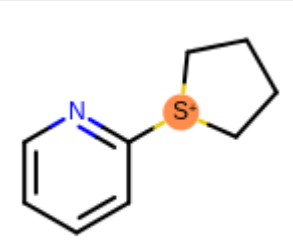




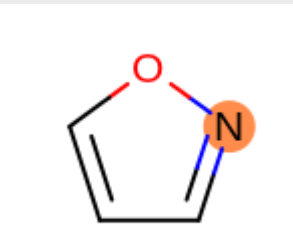
cation_C_Cl_I_P_or_S

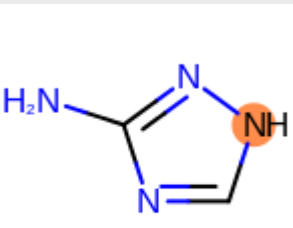


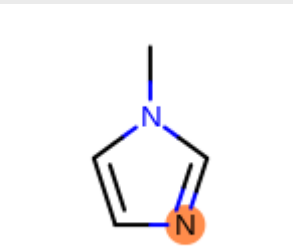




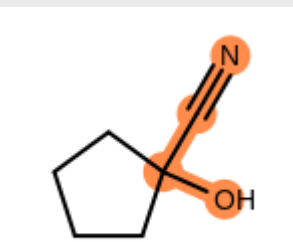
charge

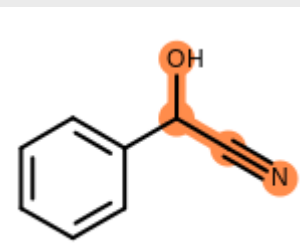


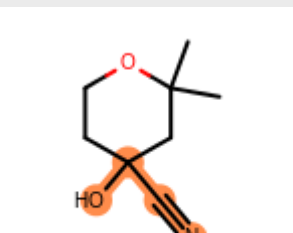




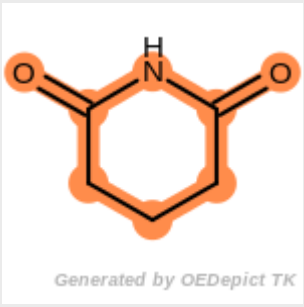
cyanohydrins

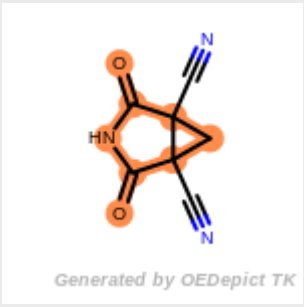


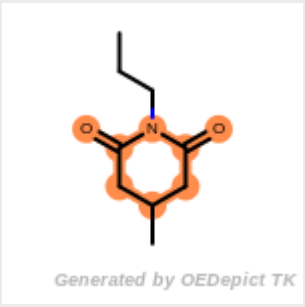




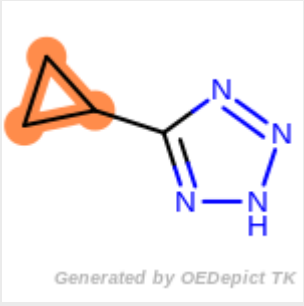
cycloheximide_derivatives



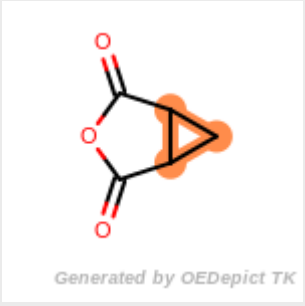




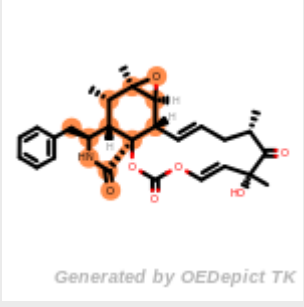
cyclopropyl

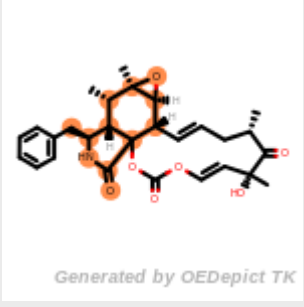


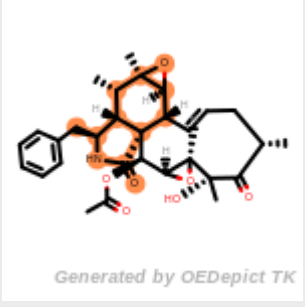




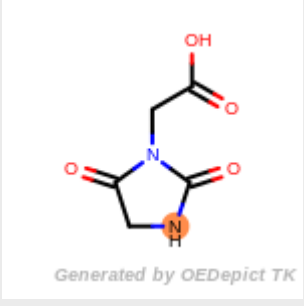
cytochalasin_derivatives

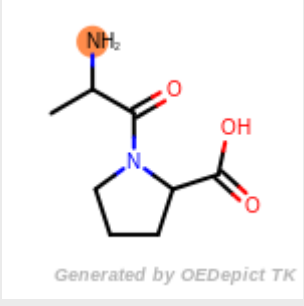


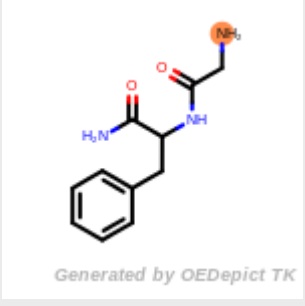




di_peptide

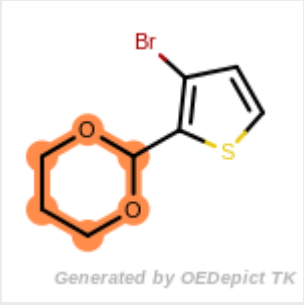


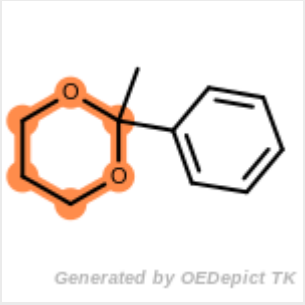




dioxane_6MR

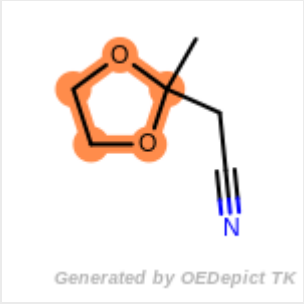


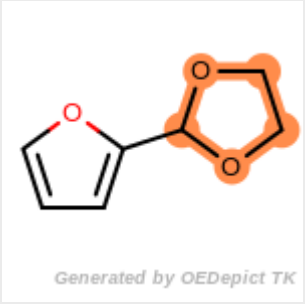




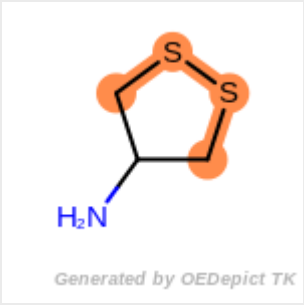
dioxolane_5MR

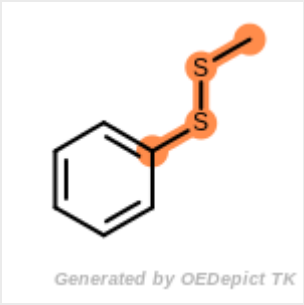


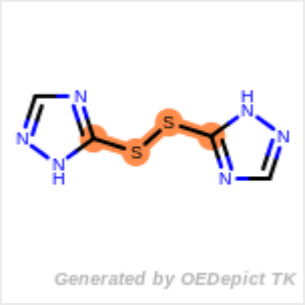




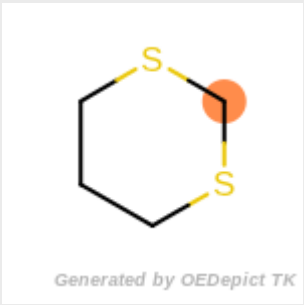
disulfide

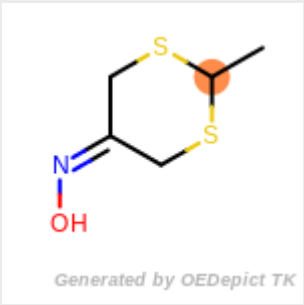


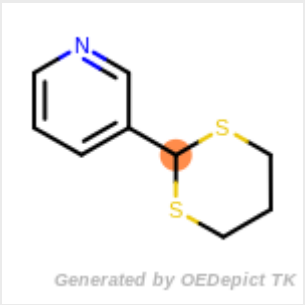




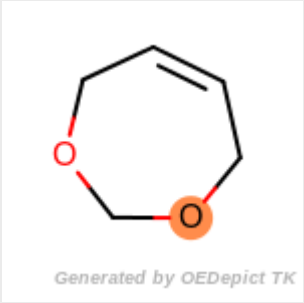
dithioacetal

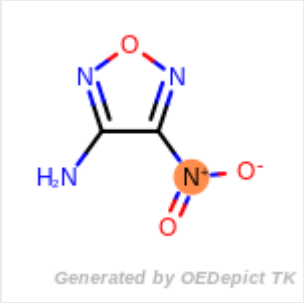


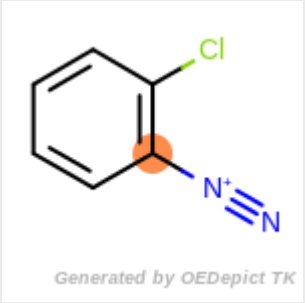




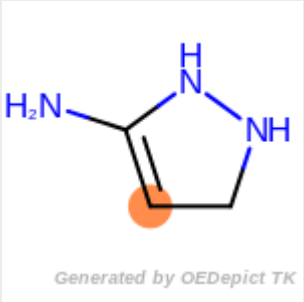
dye

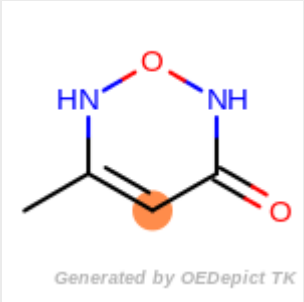


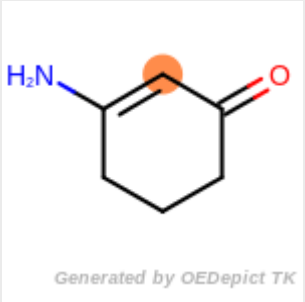




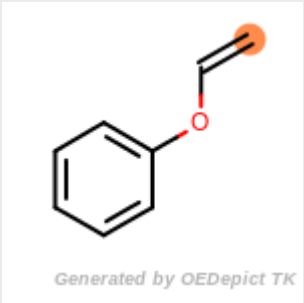
enamine

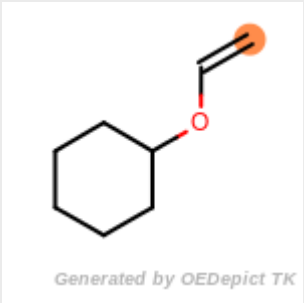


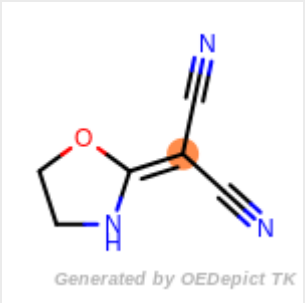




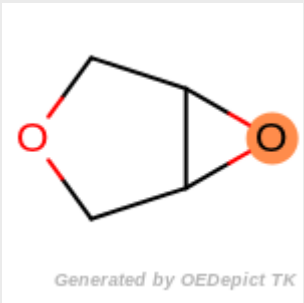
enol_ether

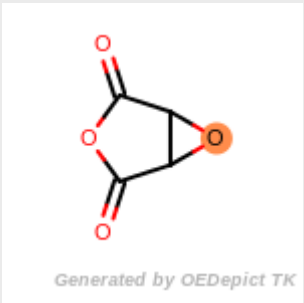


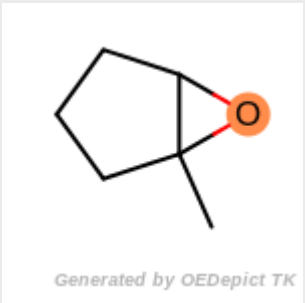




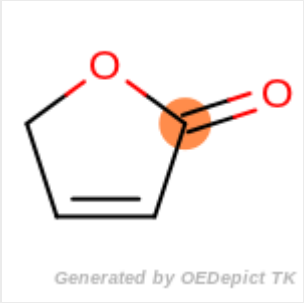
epoxide

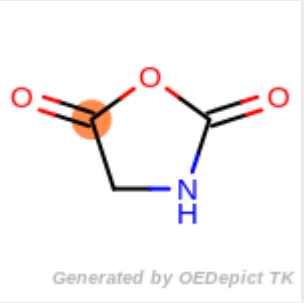


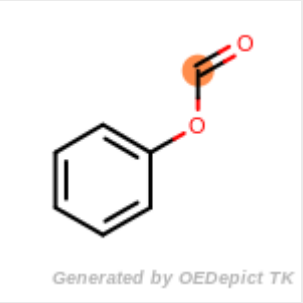




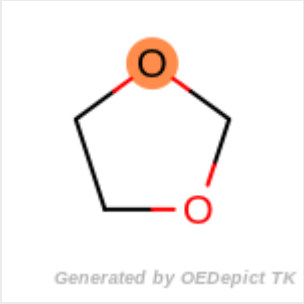
ester

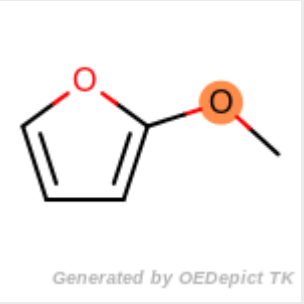


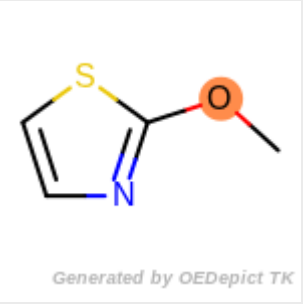




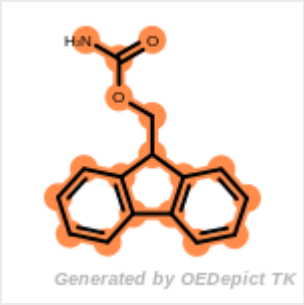
ether



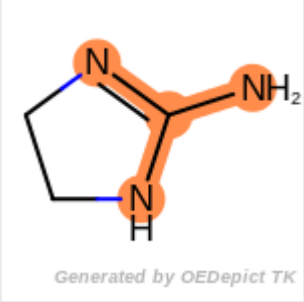


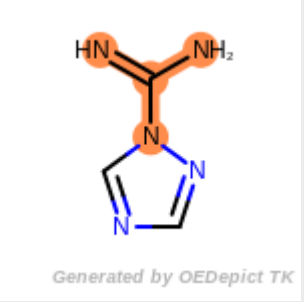


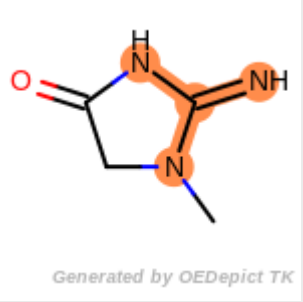
fluorenylmethoxycarbonyl_Fmoc



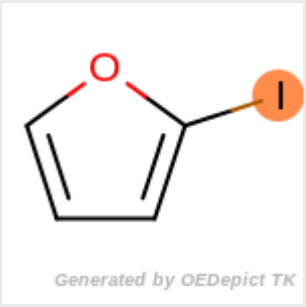
guanidine

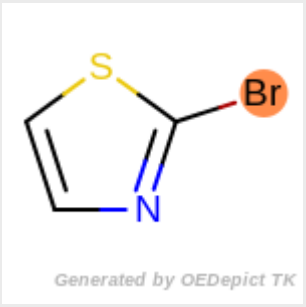


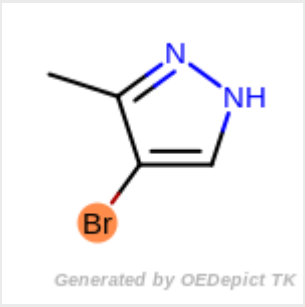




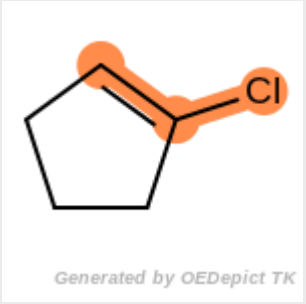
halide

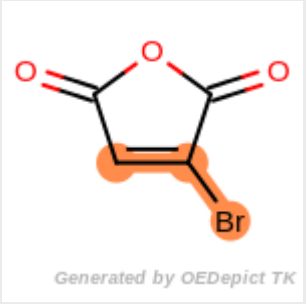


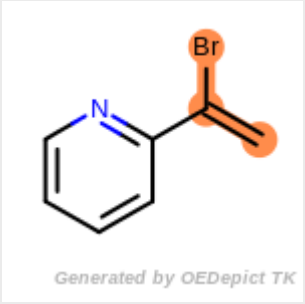




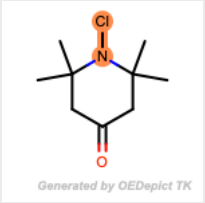
halo_alkene

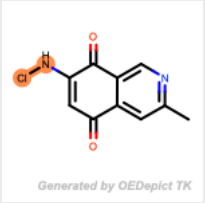


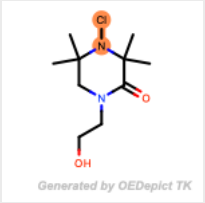




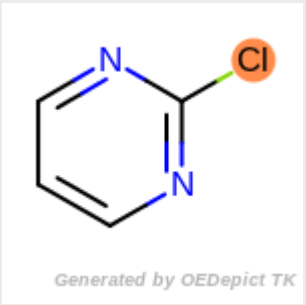
halo_amine

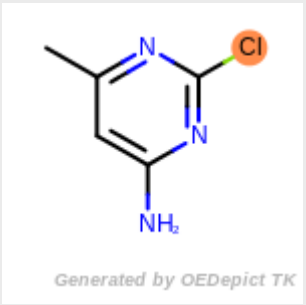


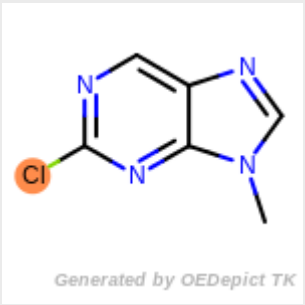




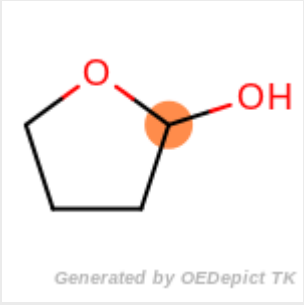
halopyrimidine

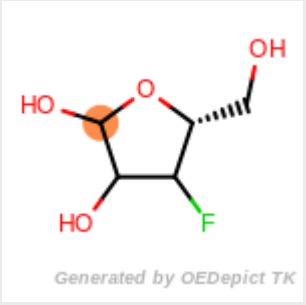


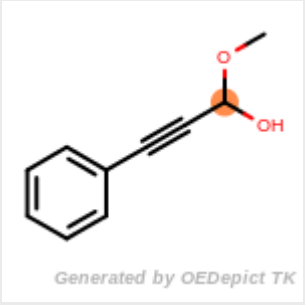




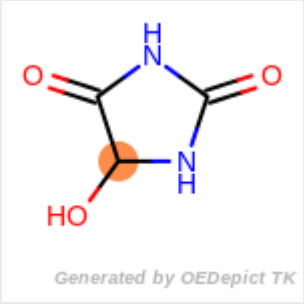
hemiacetal

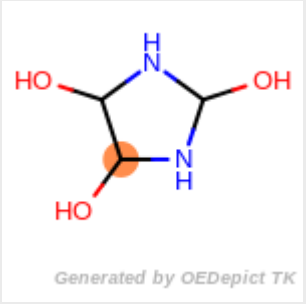


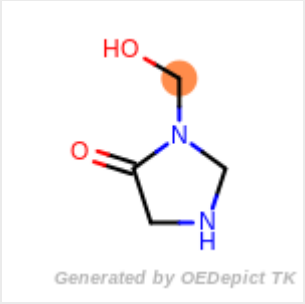




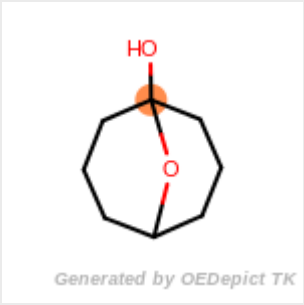
hemiaminal

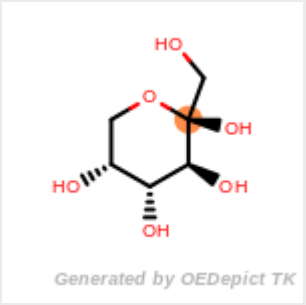


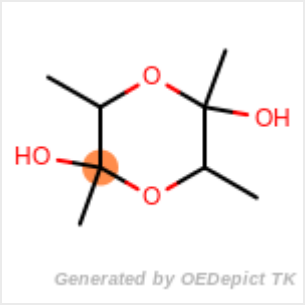




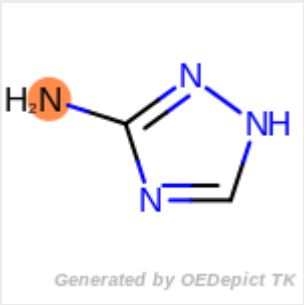
hemiketal

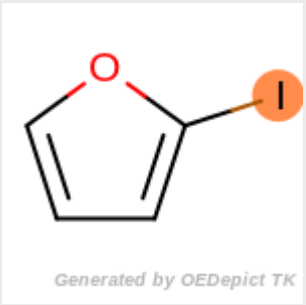


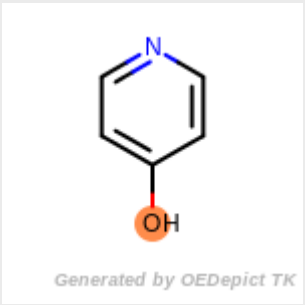




hetatm

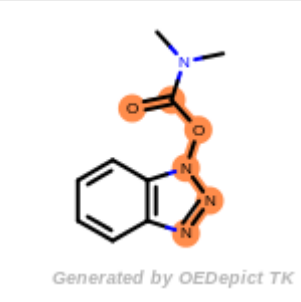


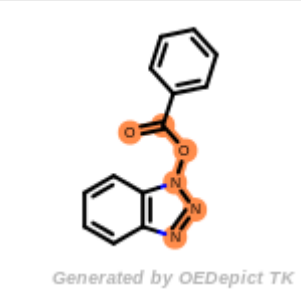




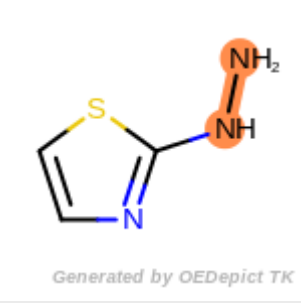
hetero_hetero

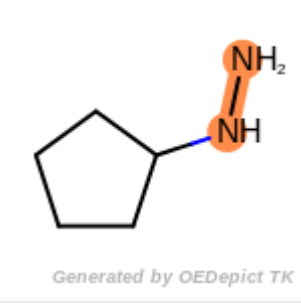
HOBT_esters

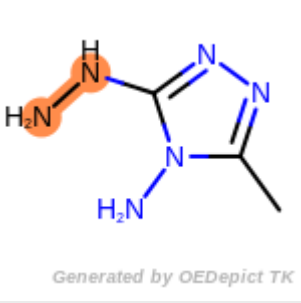




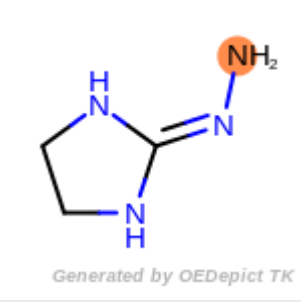
hydrazine

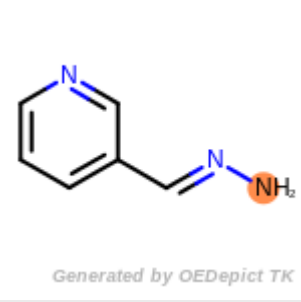


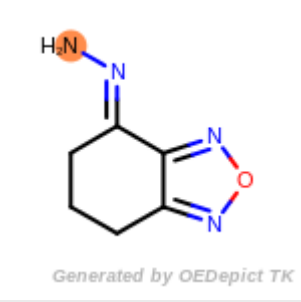




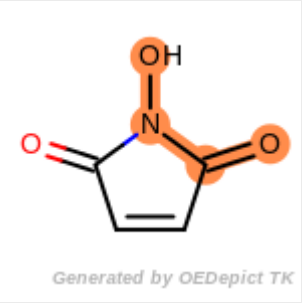
hydrazone

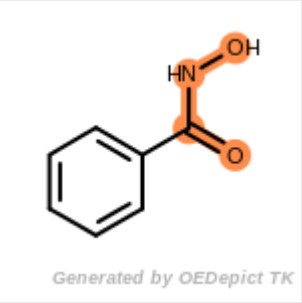


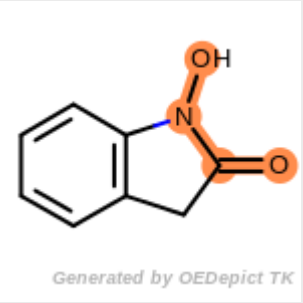




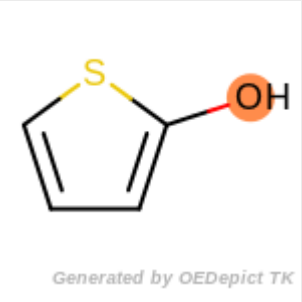
hydroxamic_acid

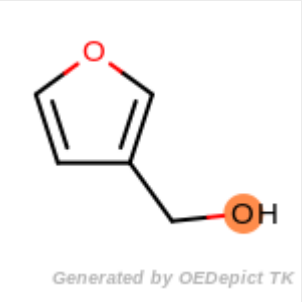


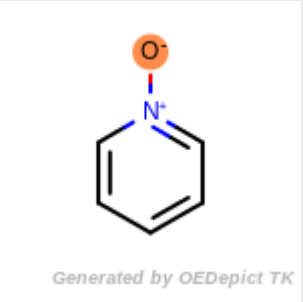




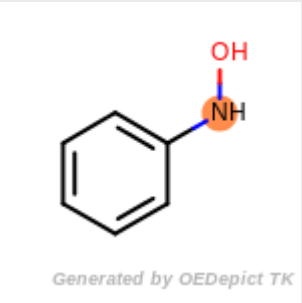
hydroxyl

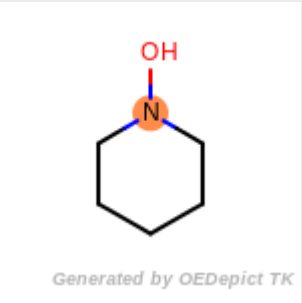


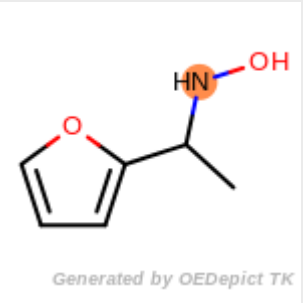




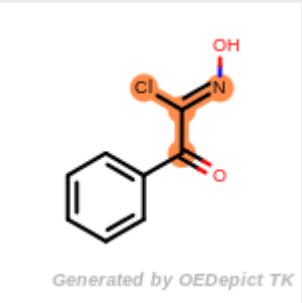
hydroxylamine

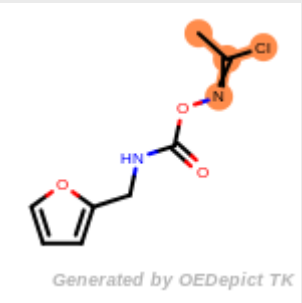


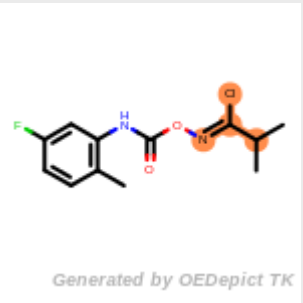




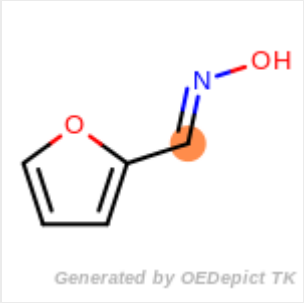
imidoyl_chlorides

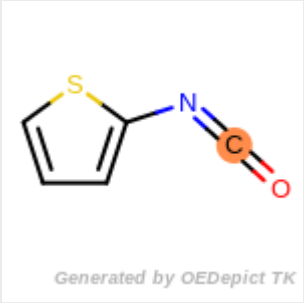


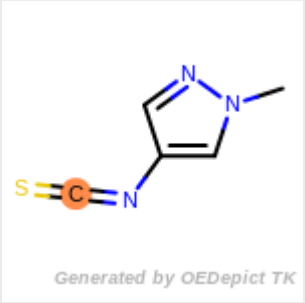




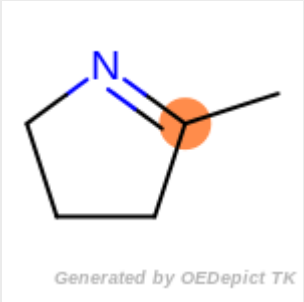
imine

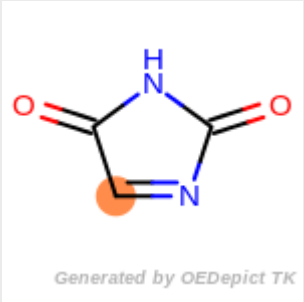


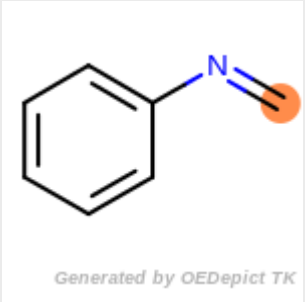




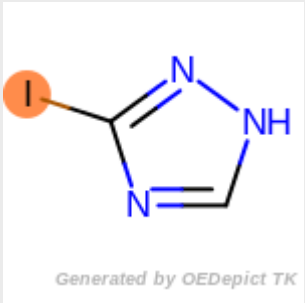
imino

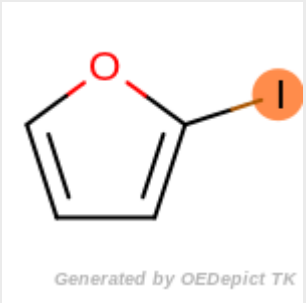


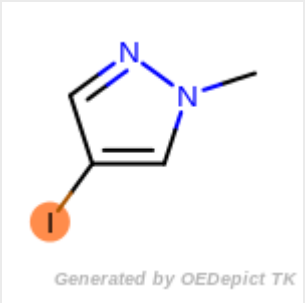




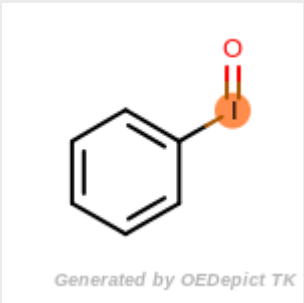
iodine

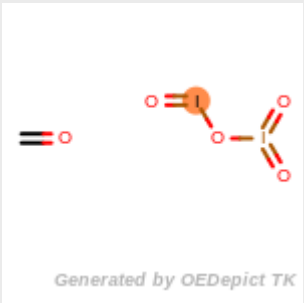


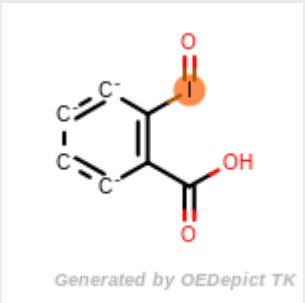




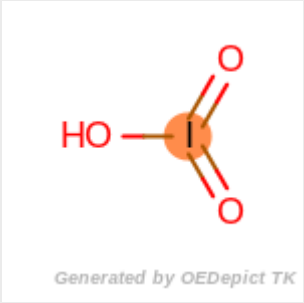
iodoso

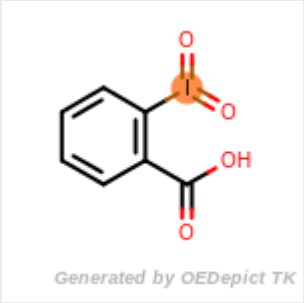


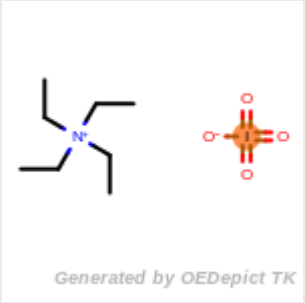




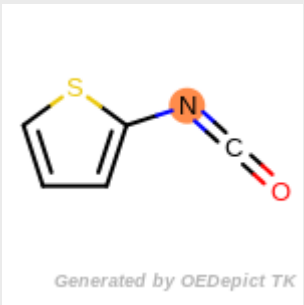
iodoxy

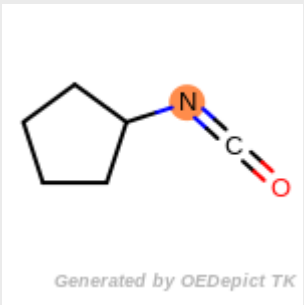


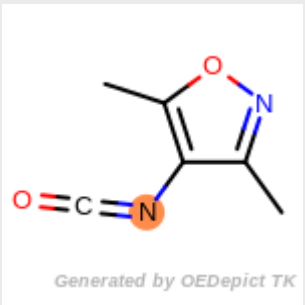




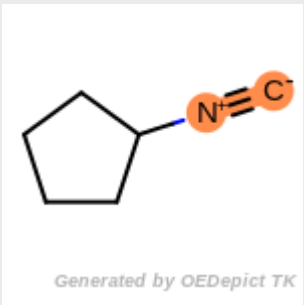
isocyanate

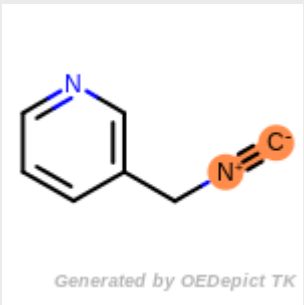


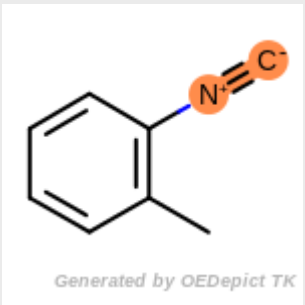




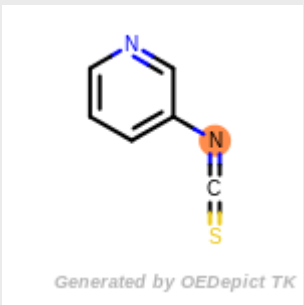
isonitrile

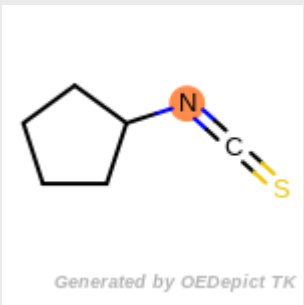


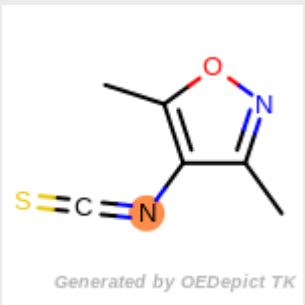




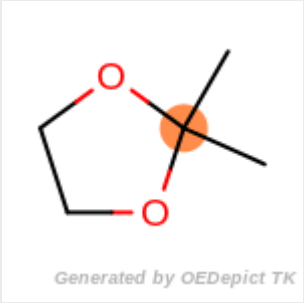
isothiocyanate

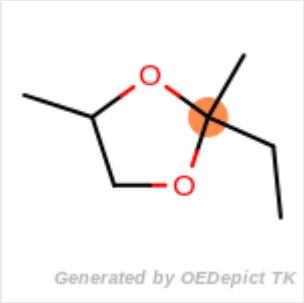


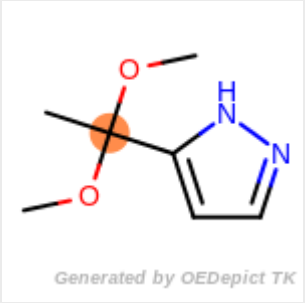




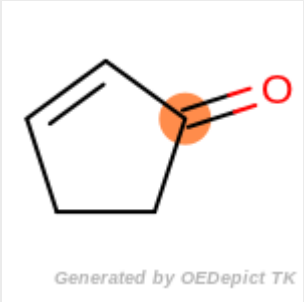
ketal

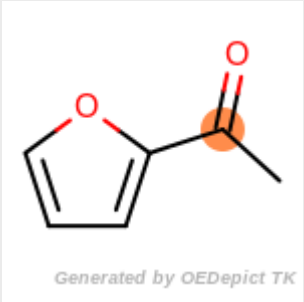


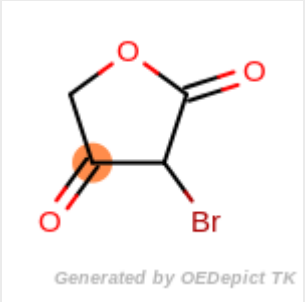




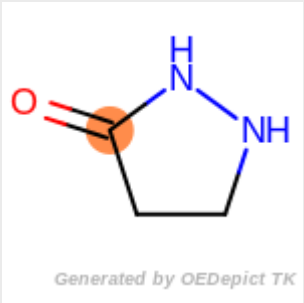
ketone

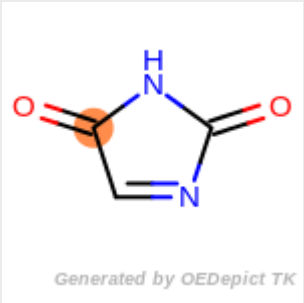


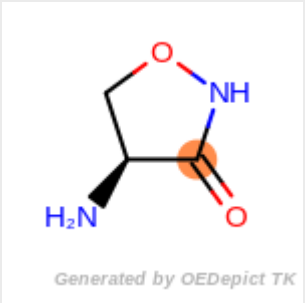




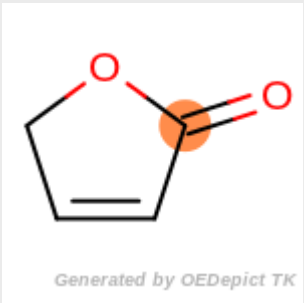
lactam

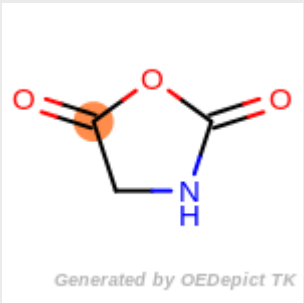


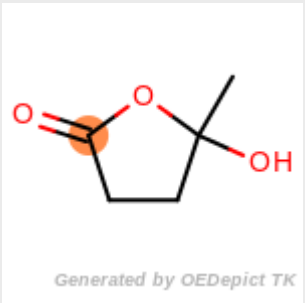




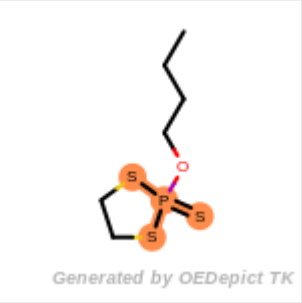
lactone

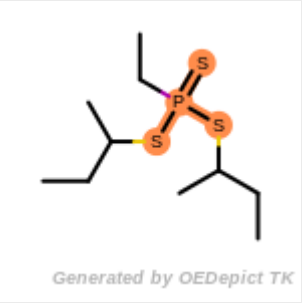


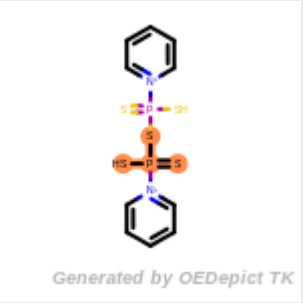




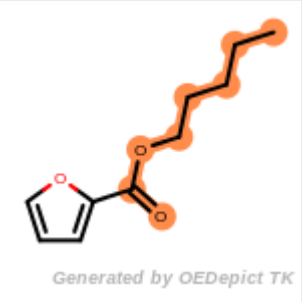
lawesson_s_reagent

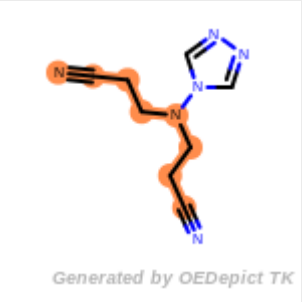


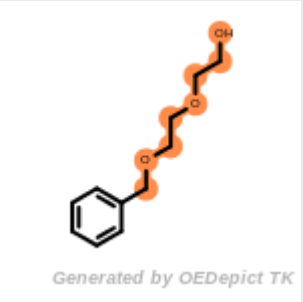




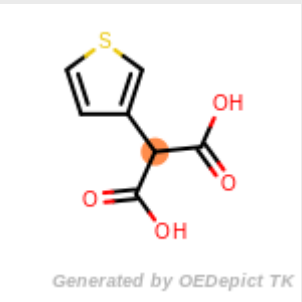
long_aliphatic_chain

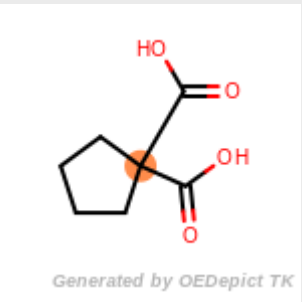


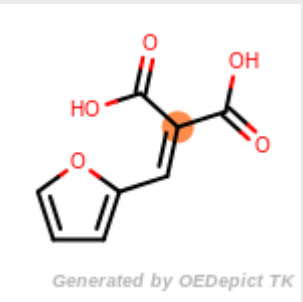




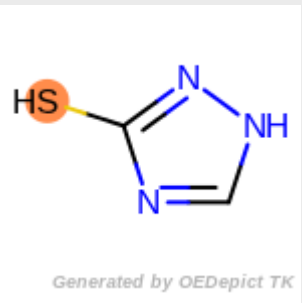
malonic

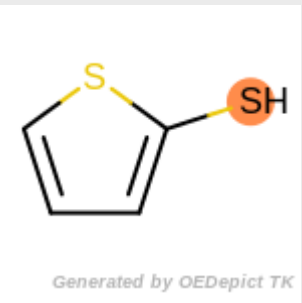


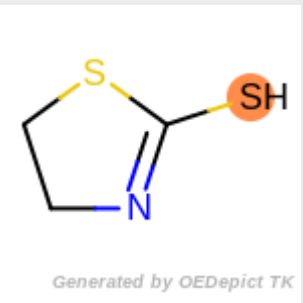




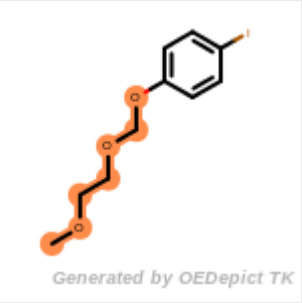
mercapto



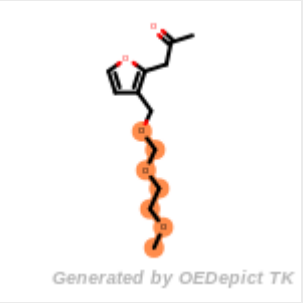




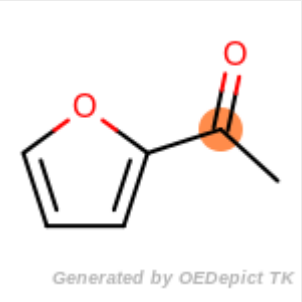
methoxyethoxymethyl_MEM

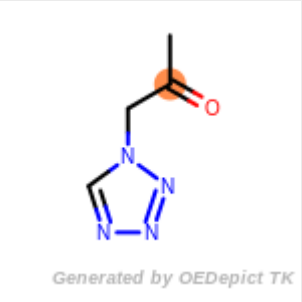


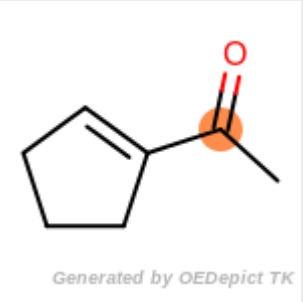




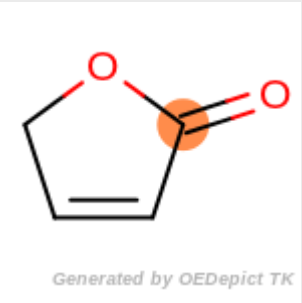
methyl_ketone

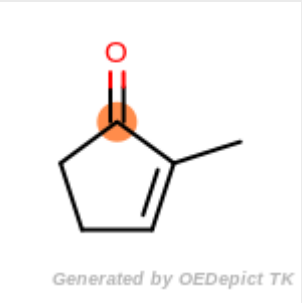


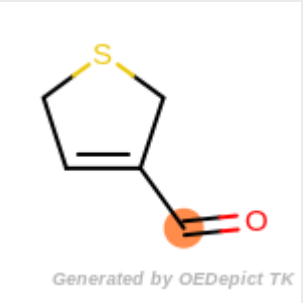




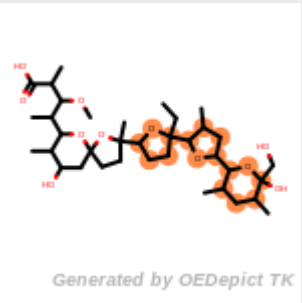
michael_acceptor



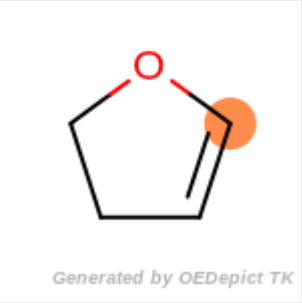


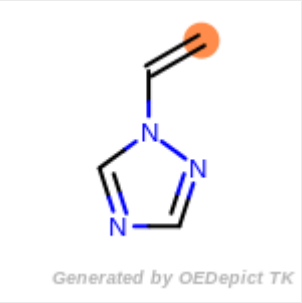


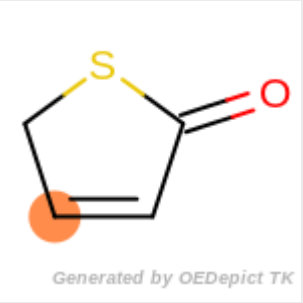
monensin_derivatives



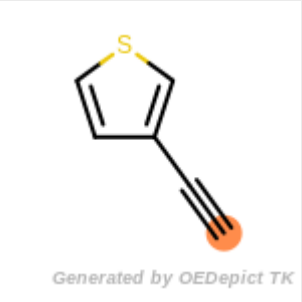
mono_alkene

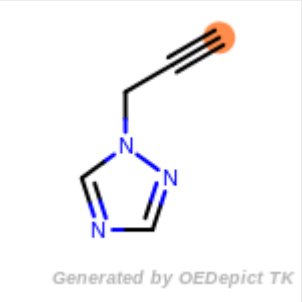


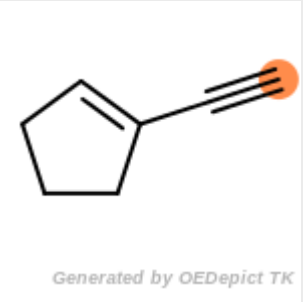




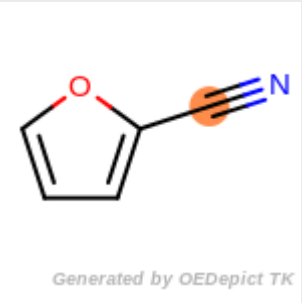
mono_alkyne

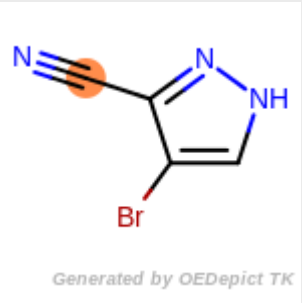


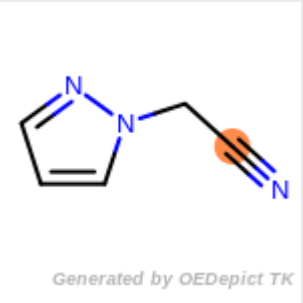




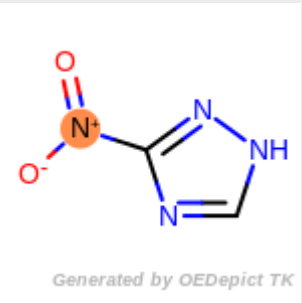
nitrile

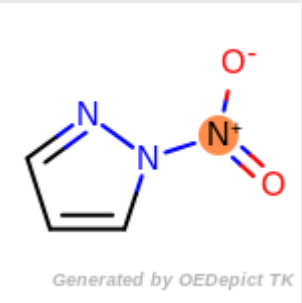


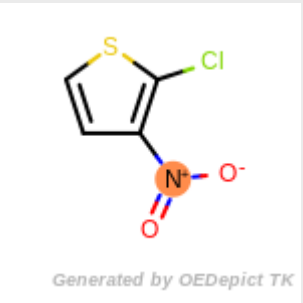




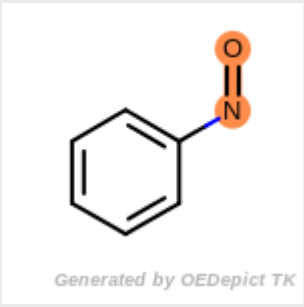
nitro

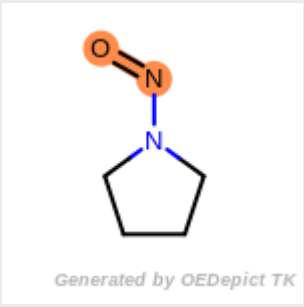


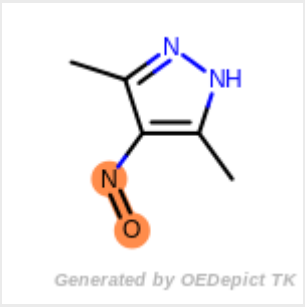




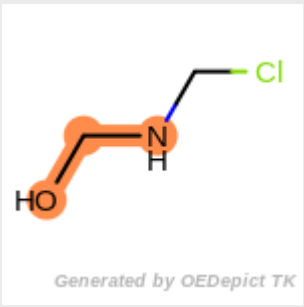
nitroso

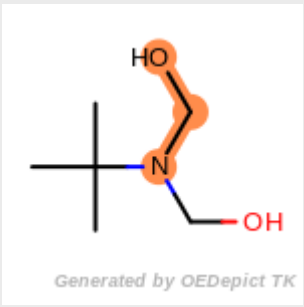


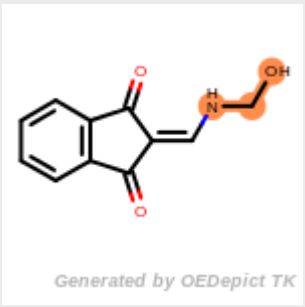




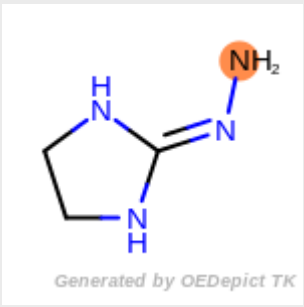
N_methoyl

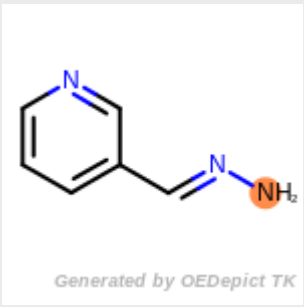


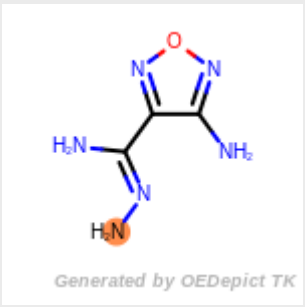




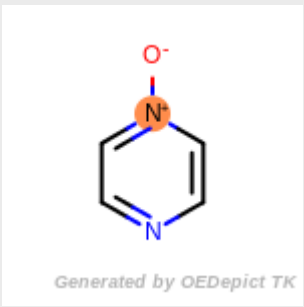
nonacylhydrazone

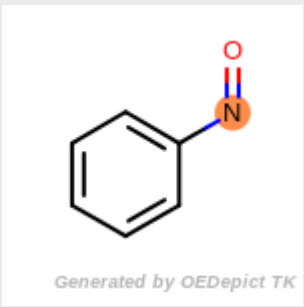


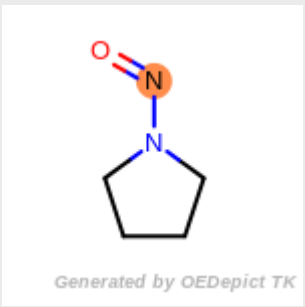




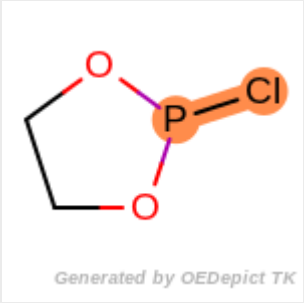
noxide

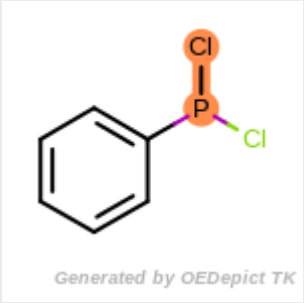


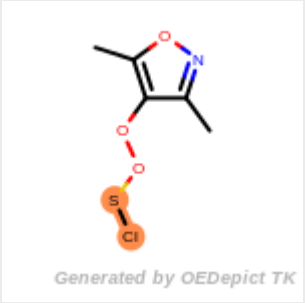




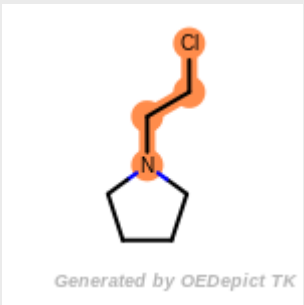
N_P_S_Halides

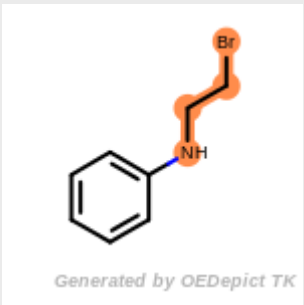


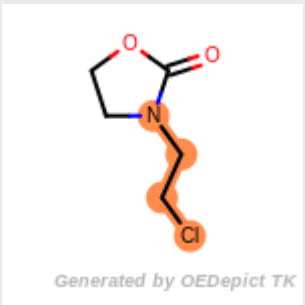




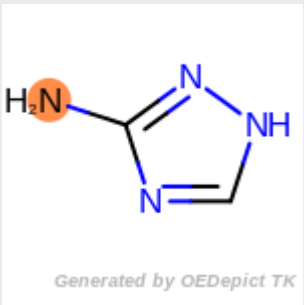
NS_beta_halothyl

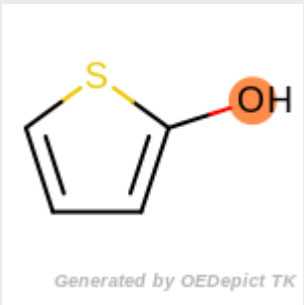


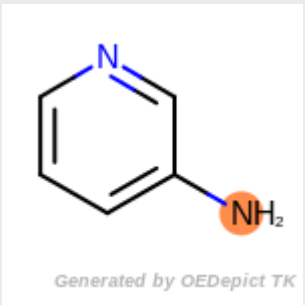




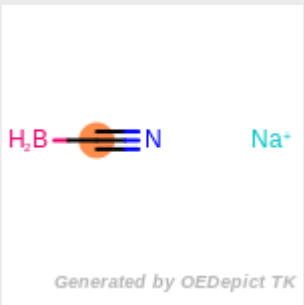
nucleophile

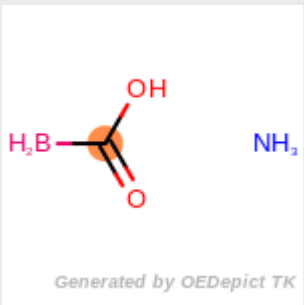


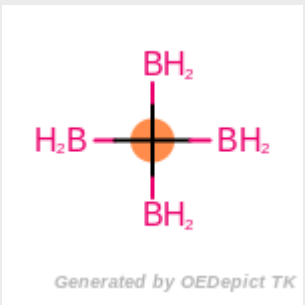




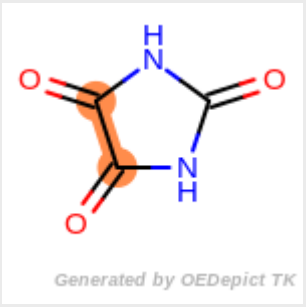
organometallic

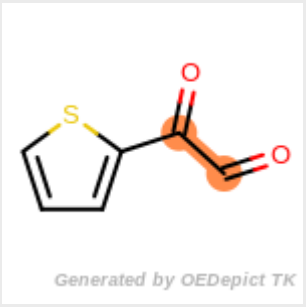


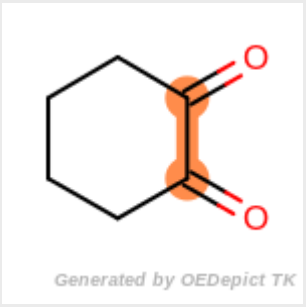




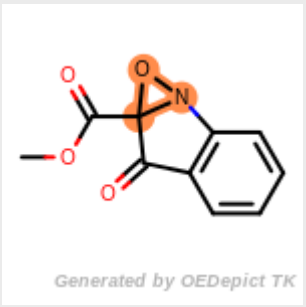
oxalyl

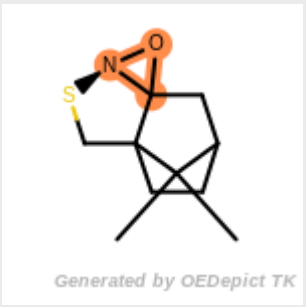


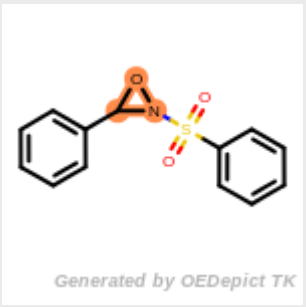




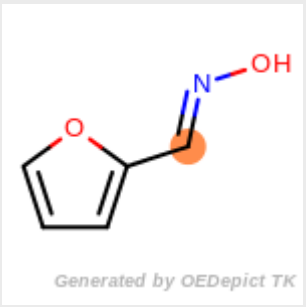
oxaziridine

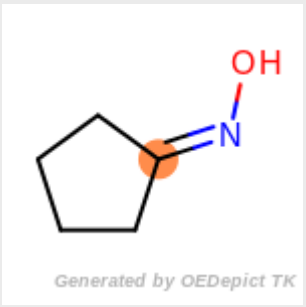


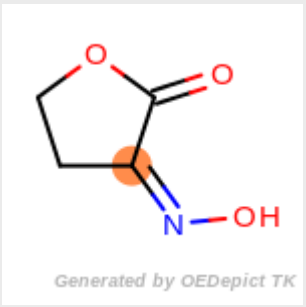




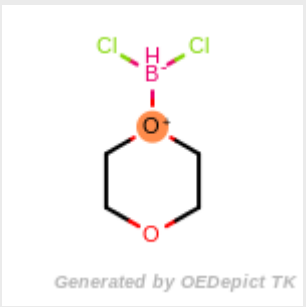
oxime

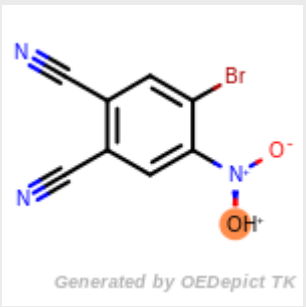


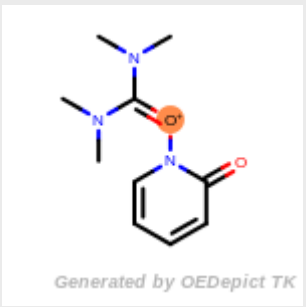




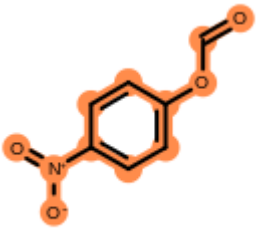
oxygen_cation

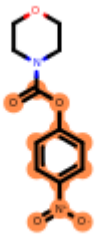


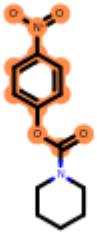




paranitrophenyl_esters



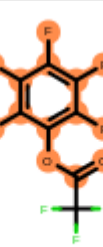




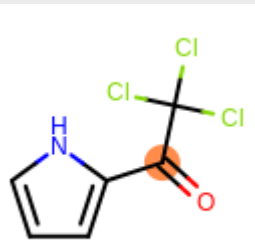
pentafluorophenyl_esters

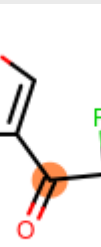


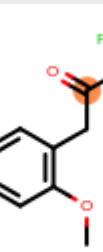




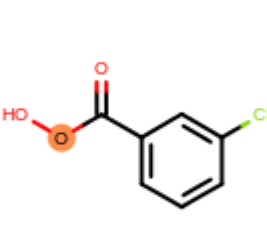
perhalo_ketone

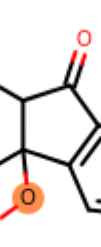


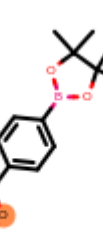




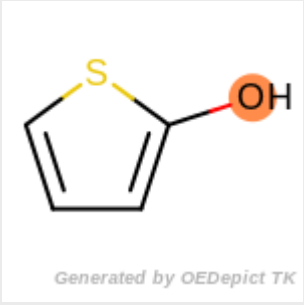
peroxide

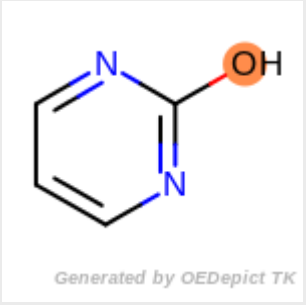


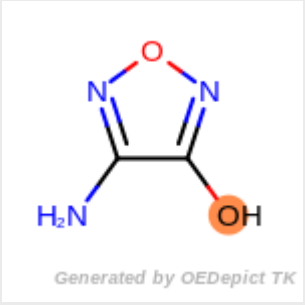




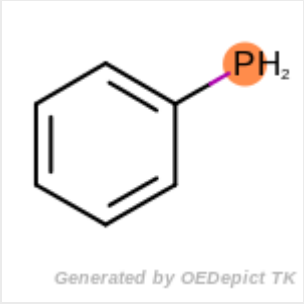
phenol

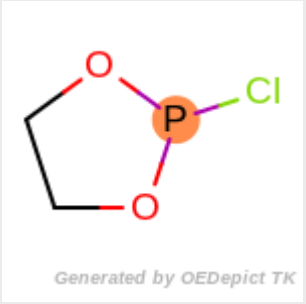


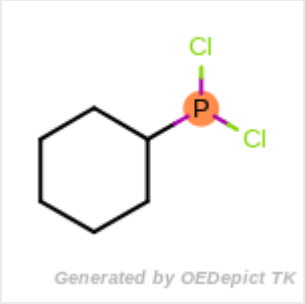




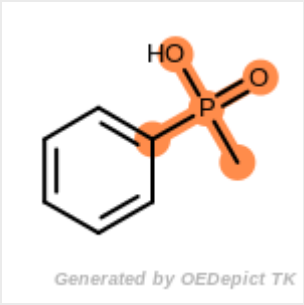
phosphanes

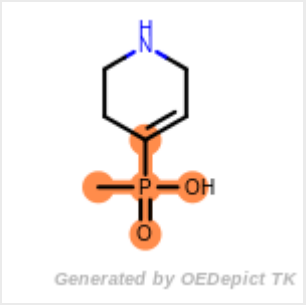


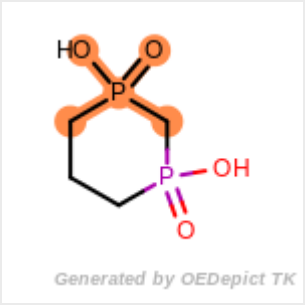




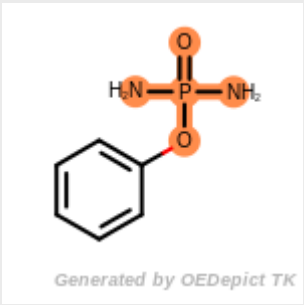
phosphinic_acid

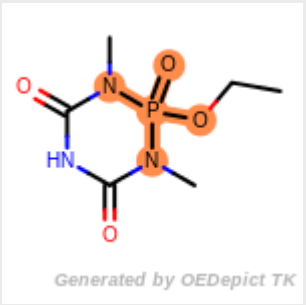


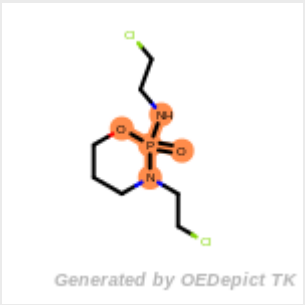




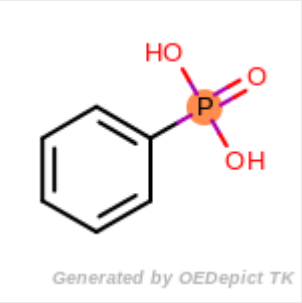
phosphonamide

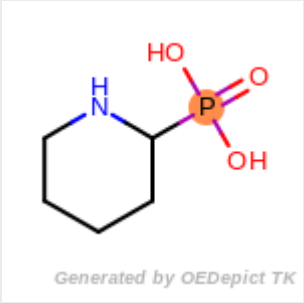


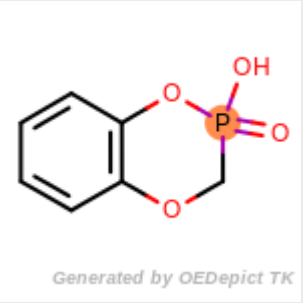




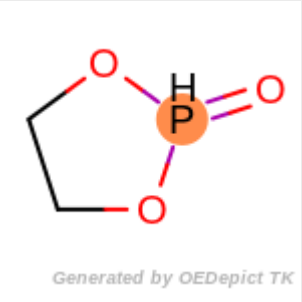
phosphonic_acid

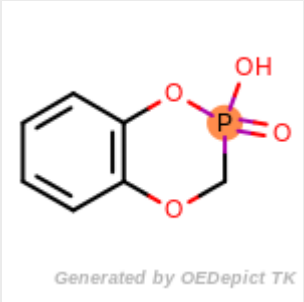


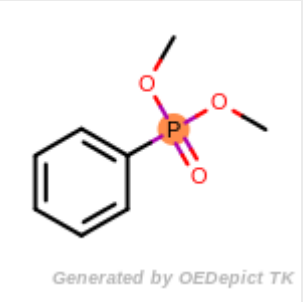




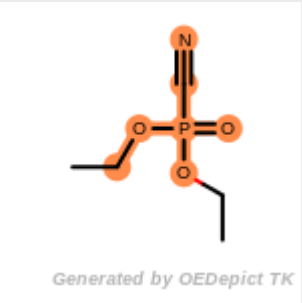
phosphonic_ester



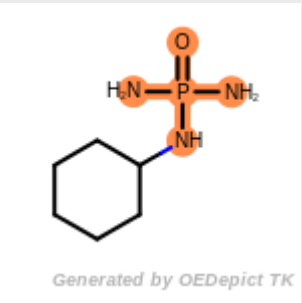


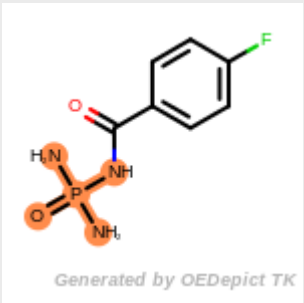


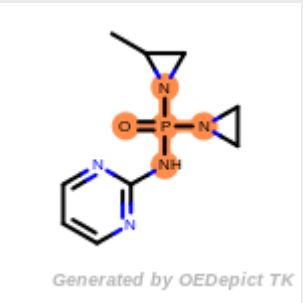
phosphonylnitrile



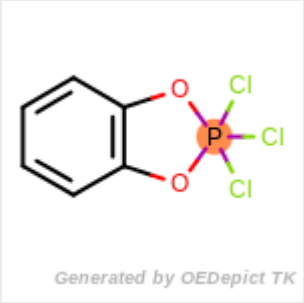
phosphoramides

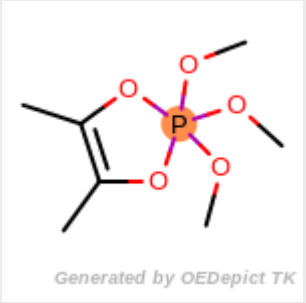


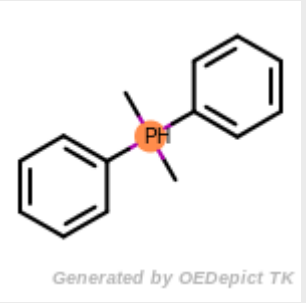




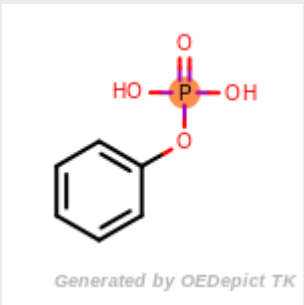
phosphoranes

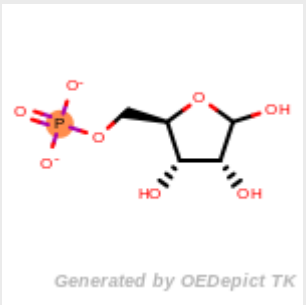


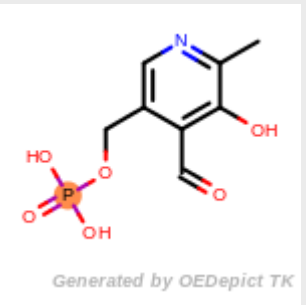




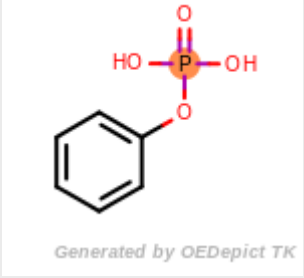
phosphoric_acid

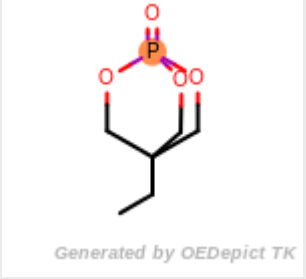


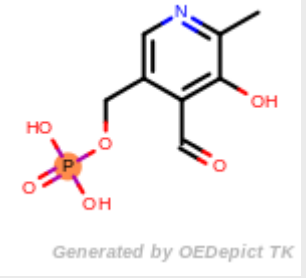




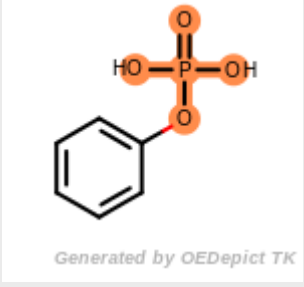
phosphoric_ester

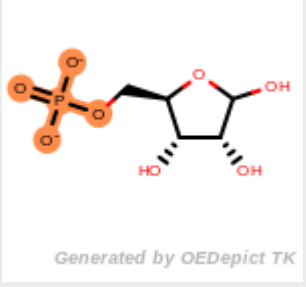


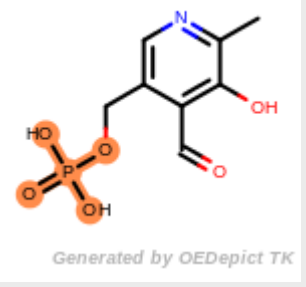




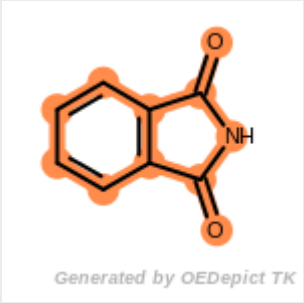
phosphoryl

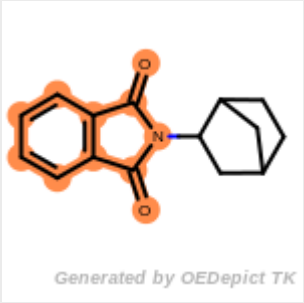


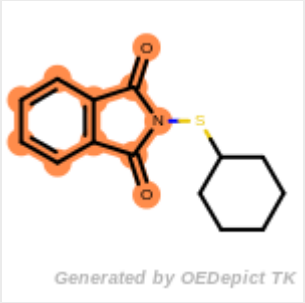




phthalimides_PHT

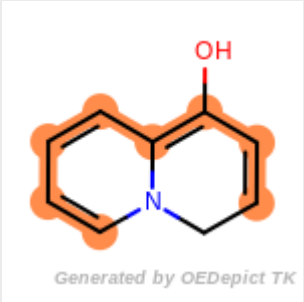


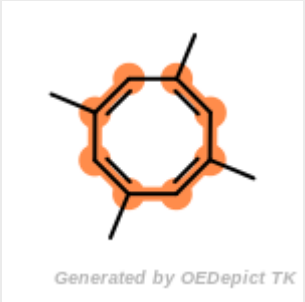




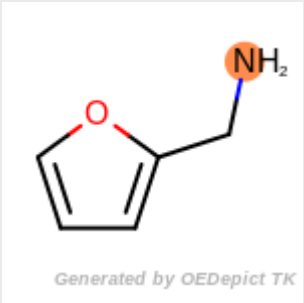
polyenes

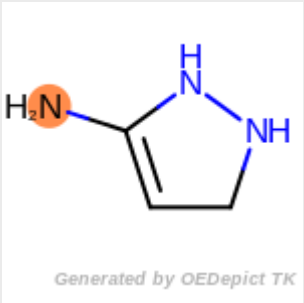


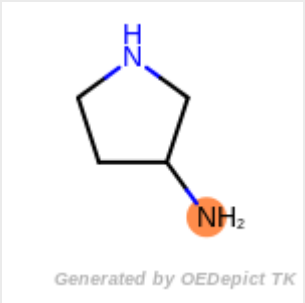




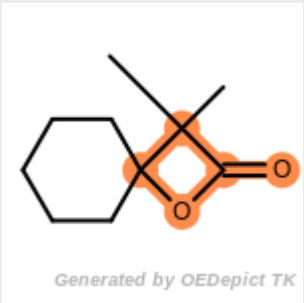
primary_amine

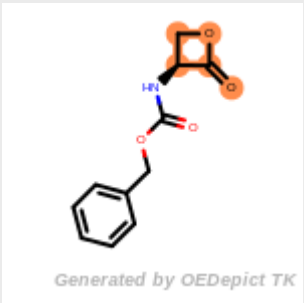


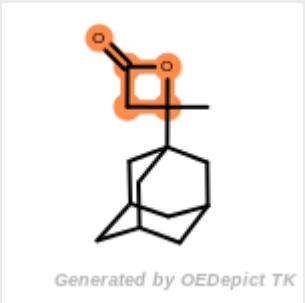




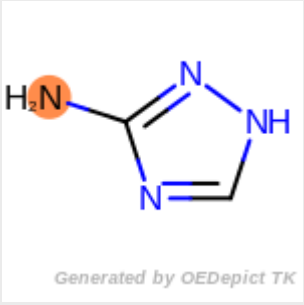
propiolactones

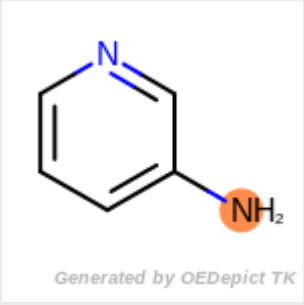


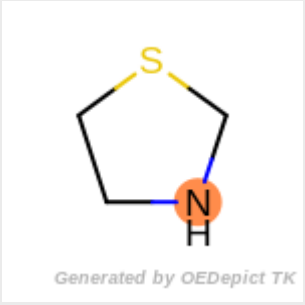




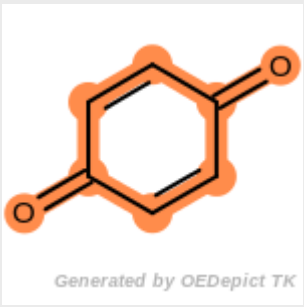
pseudo_amine

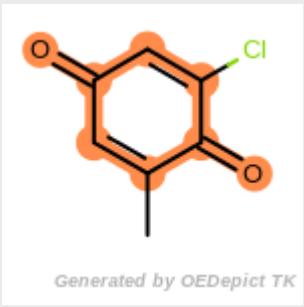


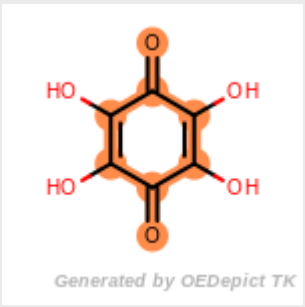




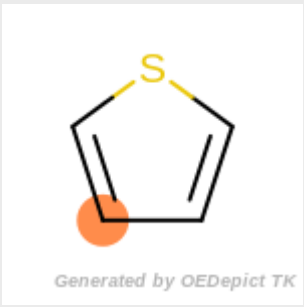
quinone

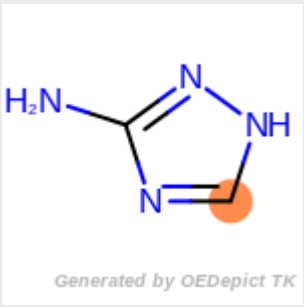


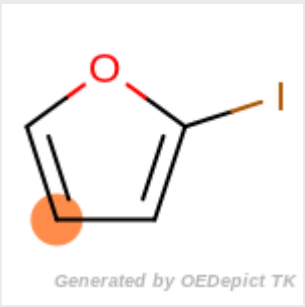




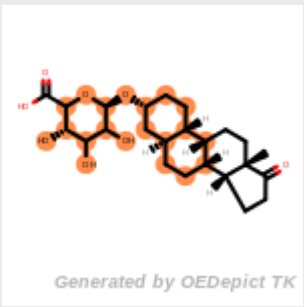
ring



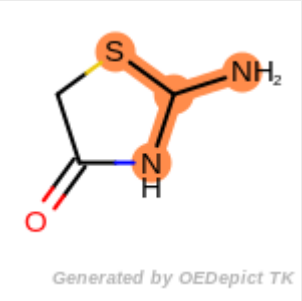


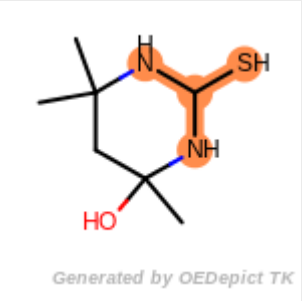


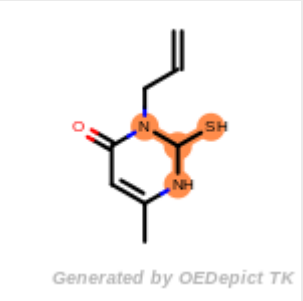
saponin_derivatives



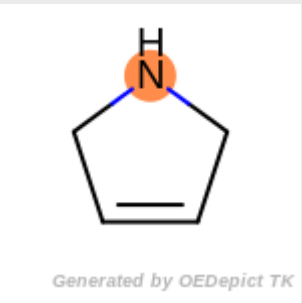
SCN2

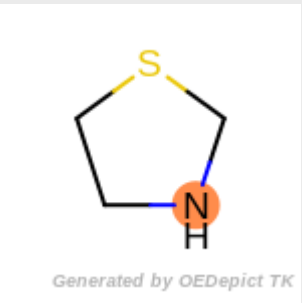


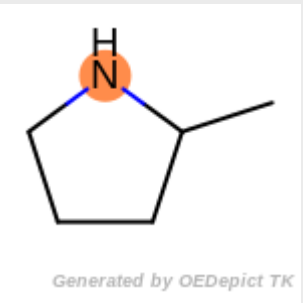




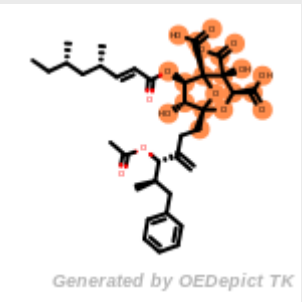
secondary_amine



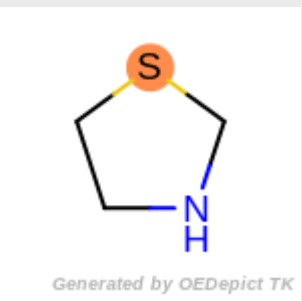


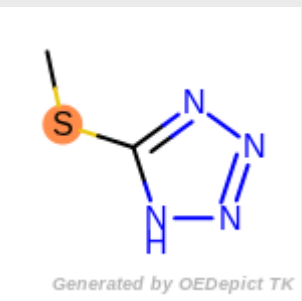


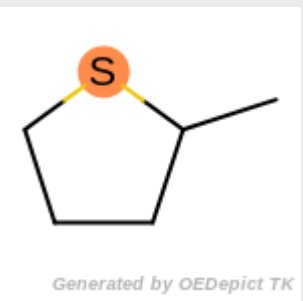
squalestatin_derivatives



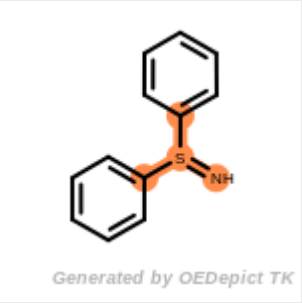
sulfide

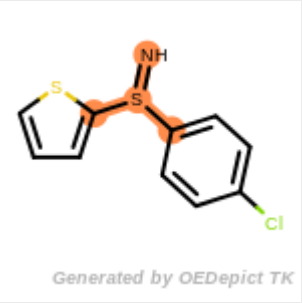


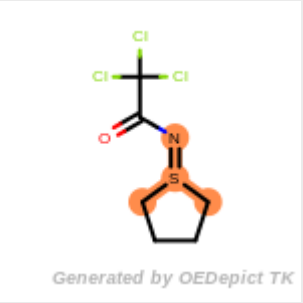




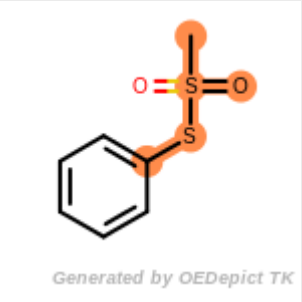
sulfinimine

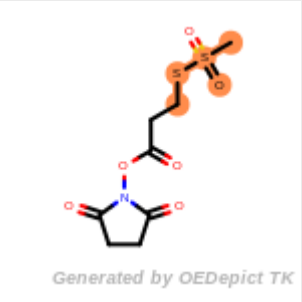


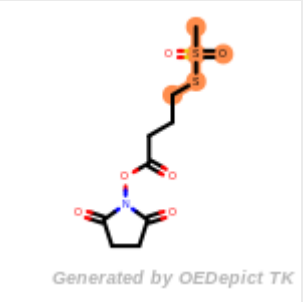




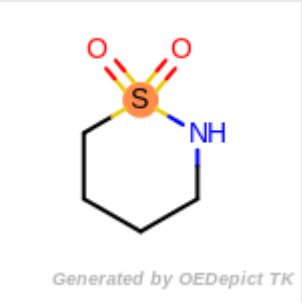
sulfinylthio

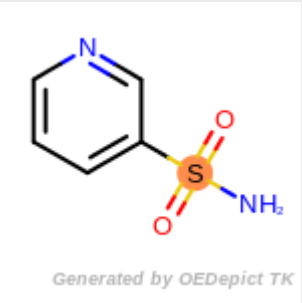


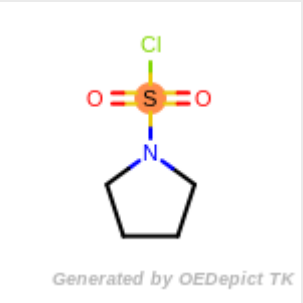




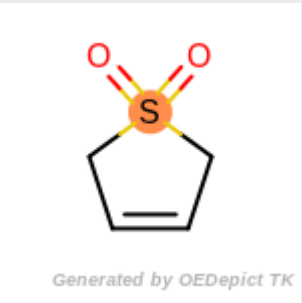
sulfonamide

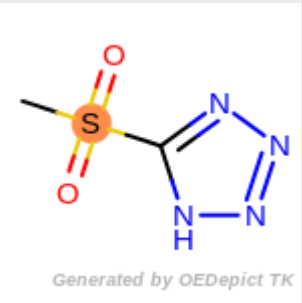


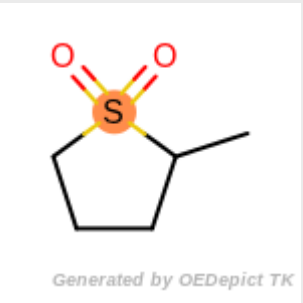




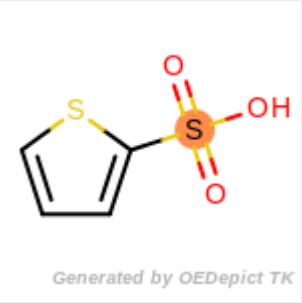
sulfone

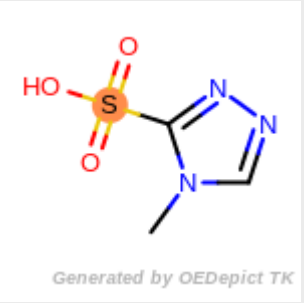


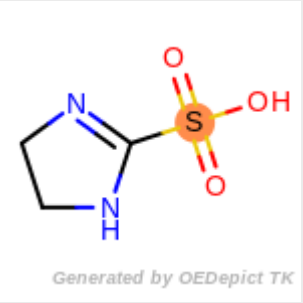




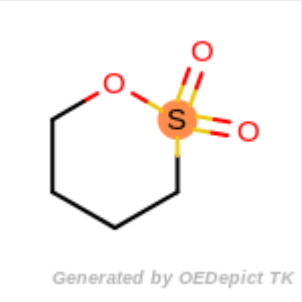
sulfonic_acid

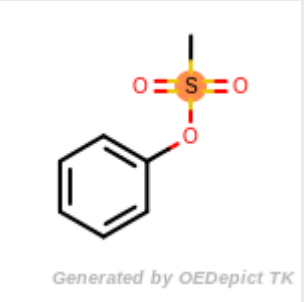


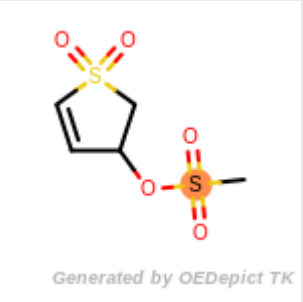




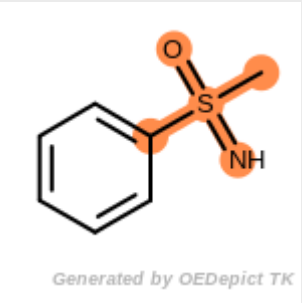
sulfonic_ester

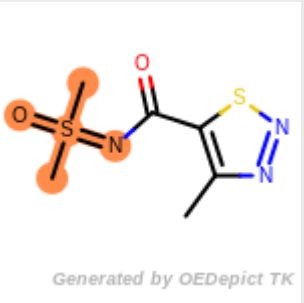


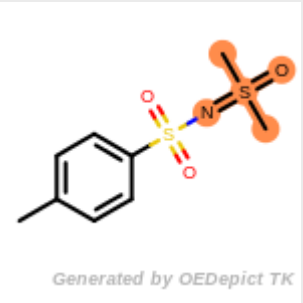




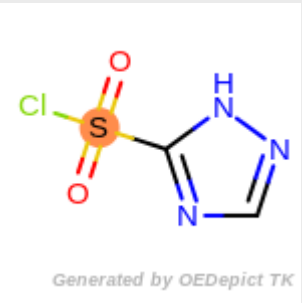
sulfonimine

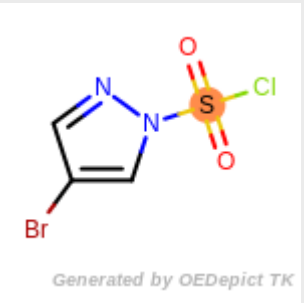


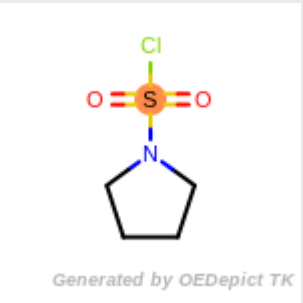




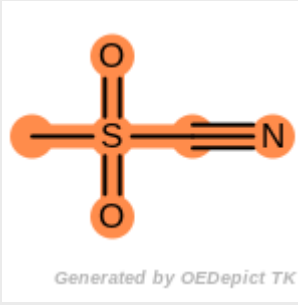
sulfonyl_halide

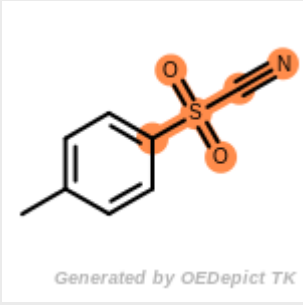


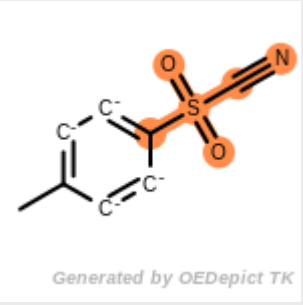




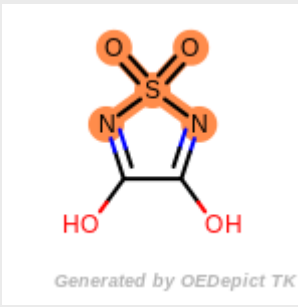
sulfonylnitrile

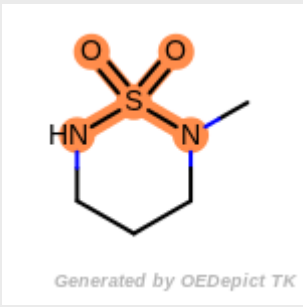


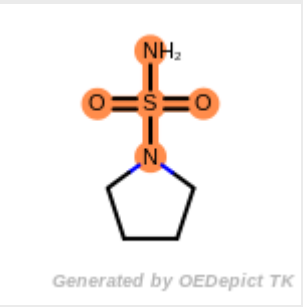




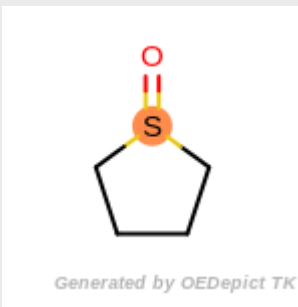
sulfonylurea

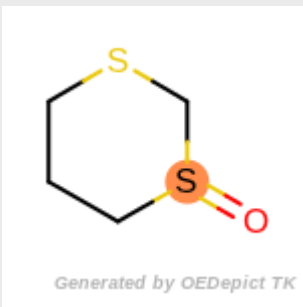


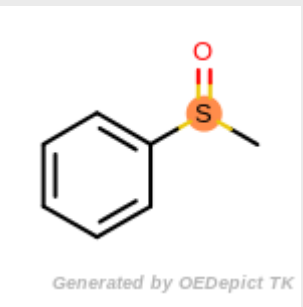




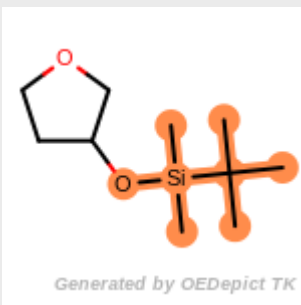
sulfoxide

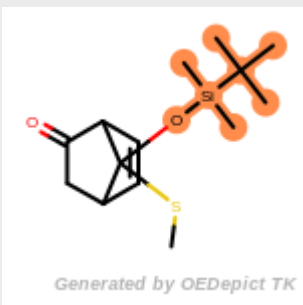


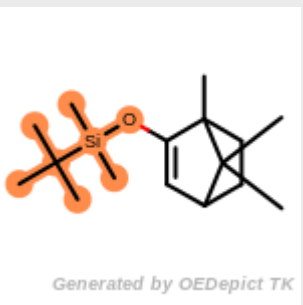




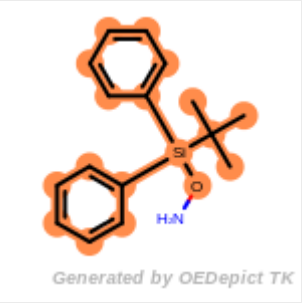
t_butyldimethylsilyl_TBDMS

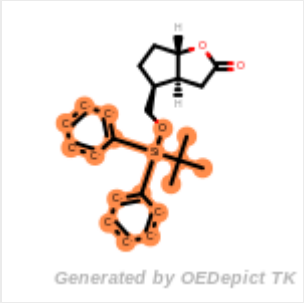




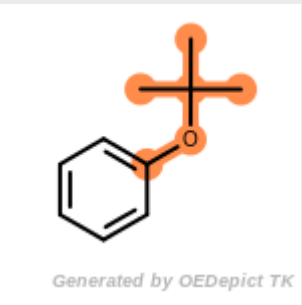


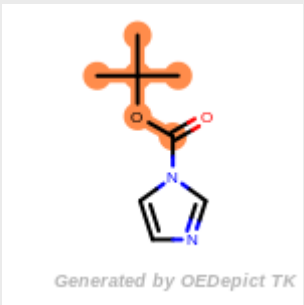
t_butyldiphenylsilyl_TBDPS

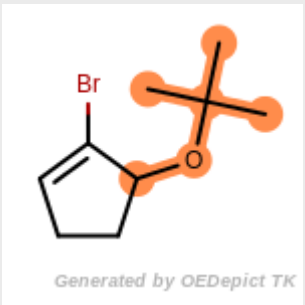




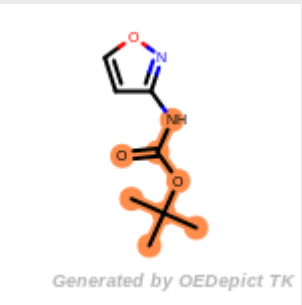
t_butyl_ether

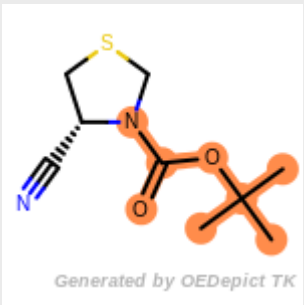


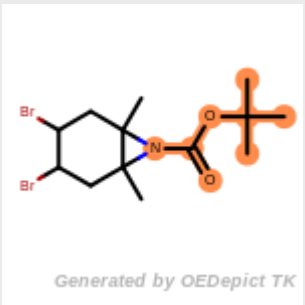




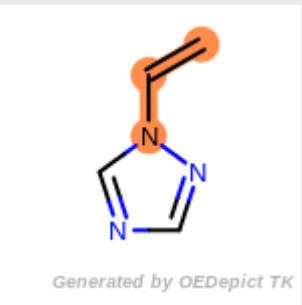
t_butoxycarbonyl_tBOC

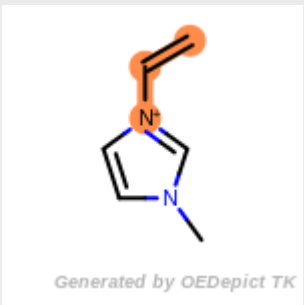


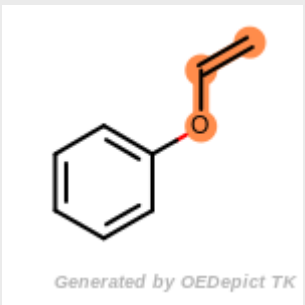




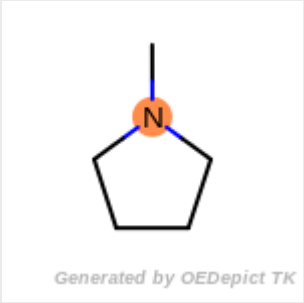
terminal_vinyl

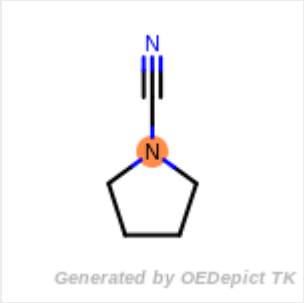


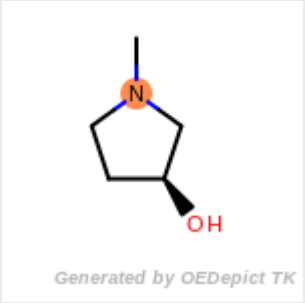




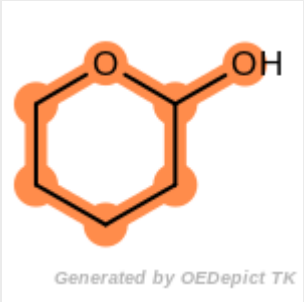
tertiary_amine

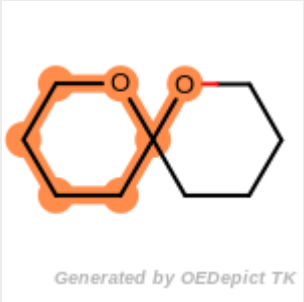


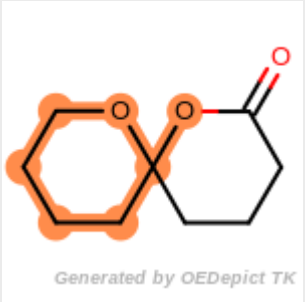




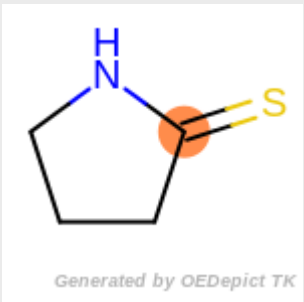
tetrahydropyran_THP

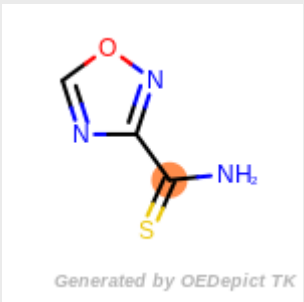


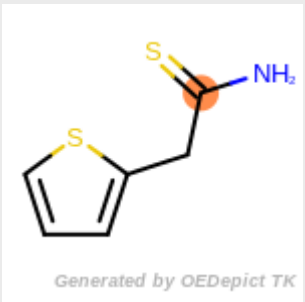




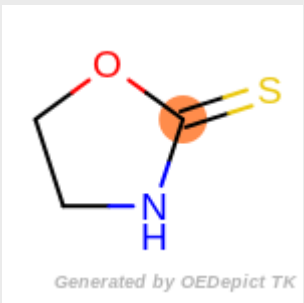
thioamide

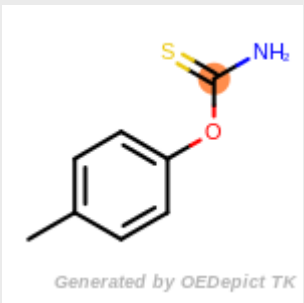


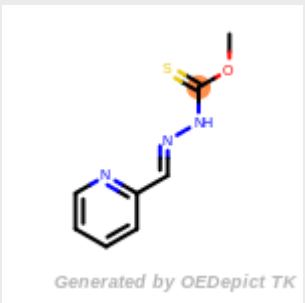




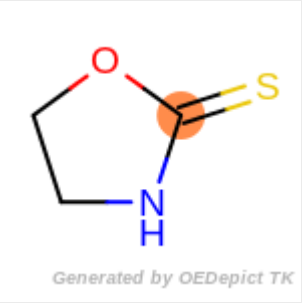
thiocarbamate

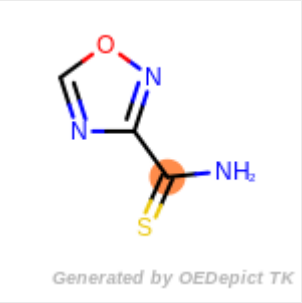


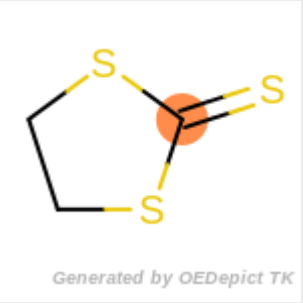




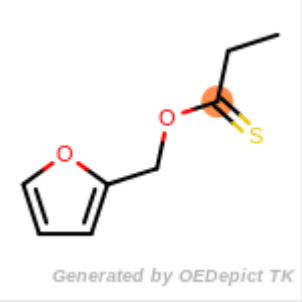
thiocarbonyl

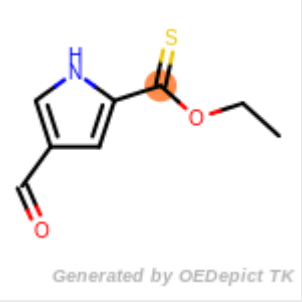


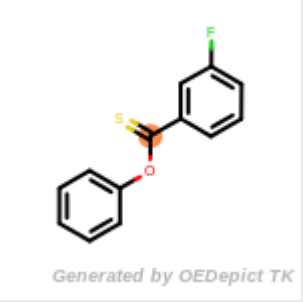




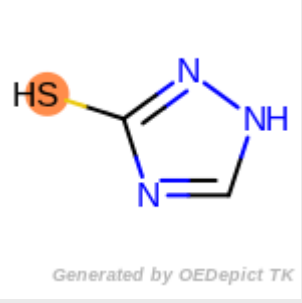
thioester

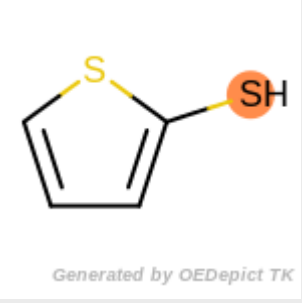


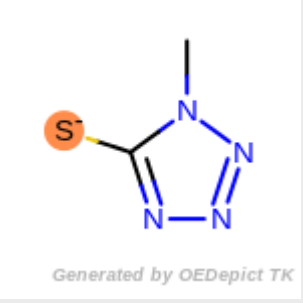




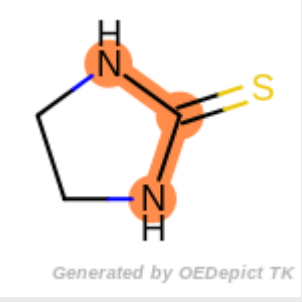
thiol

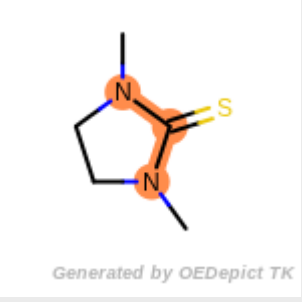


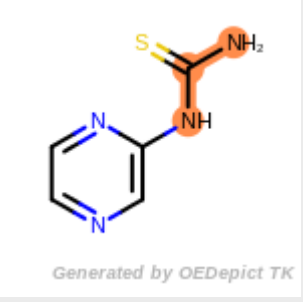




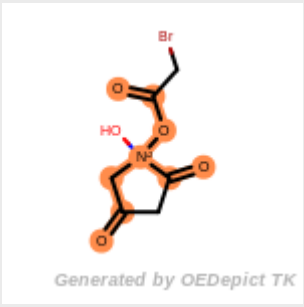
thiourea

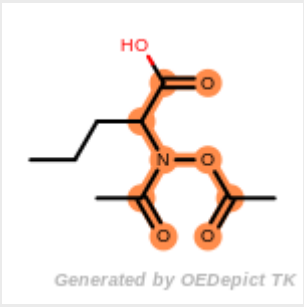


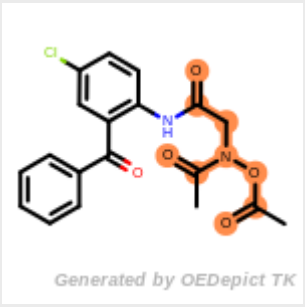




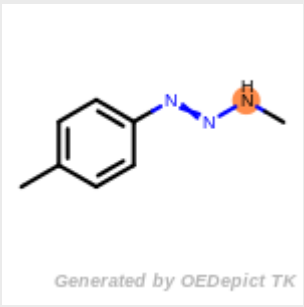
triacyloxime

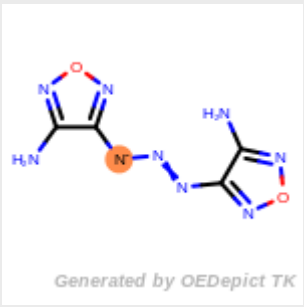


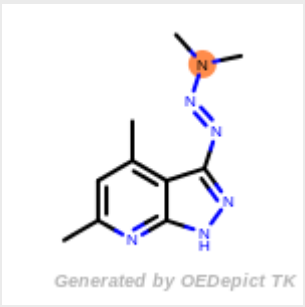




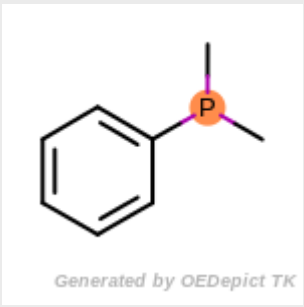
triazine

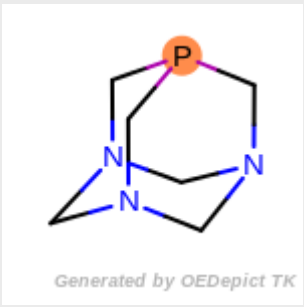


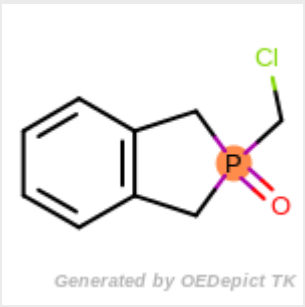




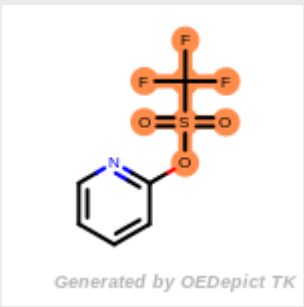
tricarbo_phosphene

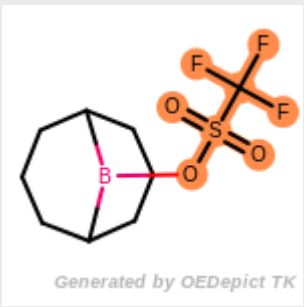


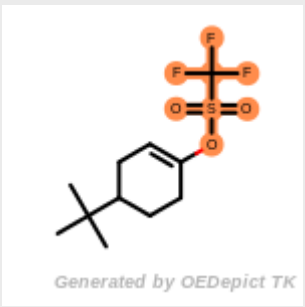




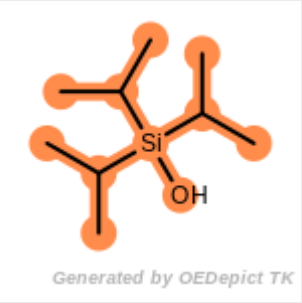
triflates

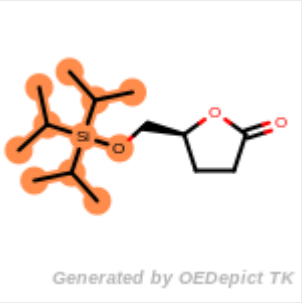


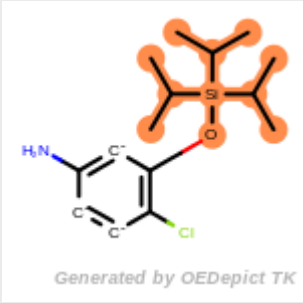




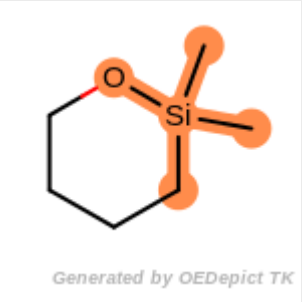
triisopropylsilyl_TIPS

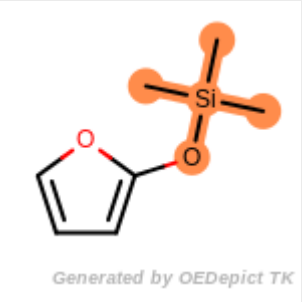


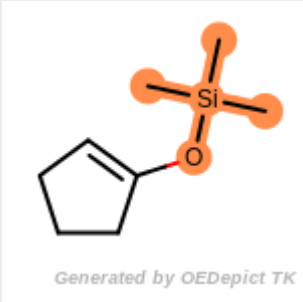




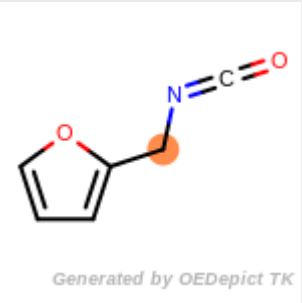
trimethylsilyl_TMS

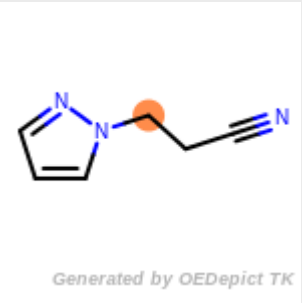


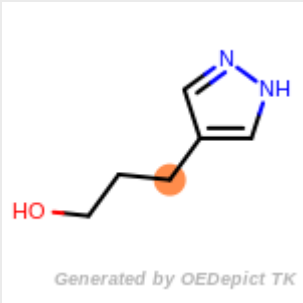




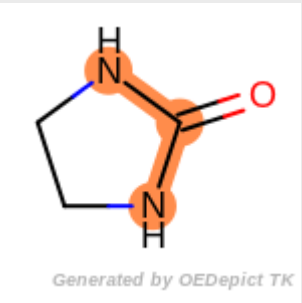
unbranched_chain

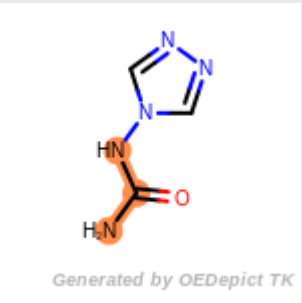


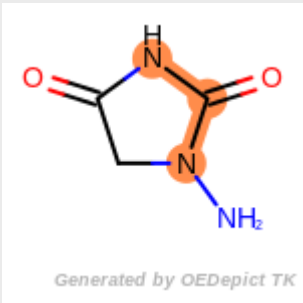




urea







2.6.3 New Rules

New rules specify additional functional groups or substructures that may be used. They must specify a substructure definition in the form of a SMARTS pattern, in addition to the substructure name and maximum limit. For example:

```
NEWRULE norbornane 1 C1CC2CCC1C2
```

The first field is the NEWRULE keyword. The second field defines the name associated with the substructure (primarily for logging purposes). The third field indicates the maximum number of the substructure that can be allowed. The fourth field is the SMARTS string for the substructure, norbornane in this case. This example rule indicates that molecules with a single norbornane substructure would be allowable, but those with two or more norbornanes would be eliminated.

New rules that have a name that is identical with one of the original rules take precedence over the original rule.

2.6.4 Selection Statements

The selection statement allows a filter file to specify the required number of substructures in order to be able to pass the filter. These statements are similar to new rules except that they list a required range for passing the filter rather than the range for failing to pass the filter. For example:

```
SELECT amine 1 1 [N;!$(*-*[!#6;!#1]);!$(*-a);!$(*=,#*)]
```

The first field is the SELECT keyword. The second field indicates the name for the selection (again for logging purposes). The third field is the minimum number of substructures required to be in the molecule. The fourth field is the maximum number of substructures allowed in the molecule. The fifth field is the substructure defined by a SMARTS pattern. The example requires that molecules contain exactly one amine. Only a single SELECT statement is allowed in the filter file. Any complex boolean substructure statements can be incorporated directly into the SMARTS. If multiple SELECT statements occur in a filter file, only the final one will be applied.

CHAPTER

THREE

OMEGA

3.1 Overview

OMEGA is a tool for generating 3D conformations of molecules. It provides two different methods for conformation generation: torsion driving and distance geometry. The torsion driving method works best on molecules that do NOT contain large, flexible rings, while the distance geometry method works for all molecules but has been designed for molecules that do contain large, flexible rings (macrocycles).

OMEGA can be used with various preset modes that generate conformers suitable for different downstream applications.

- `classic`: The `classic` mode is the original customizable **OMEGA** interface.
- `rocs`: The `rocs` mode generates conformers for **ROCS**.
- `fastrocs`: The `fastrocs` mode generates conformers for **FastROCS**.
- `pose`: The `pose` mode generates conformers for **OEDocking**.
- `dense`: The `dense` mode generates conformers for **Freeform**.
- `macrocycle`: The `macrocycle` mode uses a different algorithm than the other modes and is designed for conformation generation of molecules that contain large, flexible rings (macrocycles).

3.2 GPU-Omega

GPU-Omega refers to the CUDA-enabled GPU implementation of **OMEGA**. **GPU-Omega** takes advantage of a GPU during torsion driving for accelerated conformer generation. To make use of **GPU-Omega** conformer generation, please follow the guidelines below.

3.2.1 Prerequisites

GPU-Related Requirements

The following features are required in order to use GPU-accelerated OpenEye toolkits and applications.

3.2.2 Supported Platforms

CUDA-enabled OpenEye software is only available on supported Linux platforms. For supported Linux platforms see the Platform Support Page.

3.2.3 Supported GPUs

An NVIDIA Tesla, Quadro, or GeForce GPU with a **compute capability of 3.5 or higher** is required on your system. For a comprehensive table of compute capability categories for each GPU, please refer to the CUDA Wikipedia page.

3.2.4 NVIDIA Drivers

- **Minimum NVIDIA Driver version: 450.x**.
- CUDA is **not** required to be installed.

We recommend driver 450.80.02, and we strongly advise manually downloading and installing the appropriate NVIDIA driver for your system as opposed to using a package manager.

To install, root privilege is required. Follow these steps:

1. Download the driver to the machine you are installing it on.
2. `chmod +x` the driver package to make it executable.
3. Ensure you have disabled X-server by killing any running sessions. Reboot may be required if X-server is still running after this step.

> **Warning**
>
> Disabling X-server requires different processes to be killed depending on your Linux distribution. See the NVIDIA installation guide for more details.

> **Warning**
>
> The NVIDIA kernel module can often conflict with the open source Nouveau display drivers, depending on your specific Linux distribution. The NVIDIA documentation is a much more complete and up-to-date source for information on how to work around this issue. See Disabling Nouveau on the NVIDIA website.

4. Install the driver by `sudo ./NVIDIA-Linux-x86_64-450.80.02.run` and follow the step-by-step installation instructions.

For more details on driver installations, see the CUDA Installation Guide

> **Note**
>
> The output of the `nvidia-smi` command is extremely useful when debugging GPU issues. Please include the output from `nvidia-smi` in any request to support@eyesopen.com.

3.2.5 Performance Tuning

To get the most performance out of an NVIDIA graphics card, use the persistence daemon to switch persistence mode across all cards on the system (root privilege required):

```
sudo nvidia-persistenced --user foo
```

This will automatically enable persistence mode after reboot.

For full instructions on the persistence daemon, see the Persistence daemon section of the NVIDIA docs.

3.2.6 Usage

- **GPU-Omega** accelerates the torsion driving component of conformer generation in **OMEGA** and is available through the *-useGPU* flag.

```
prompt> oeomega classic -in drugs.mol2 -out drugs.oeb.gz -useGPU true
```

  The *-useGPU* true flag will engage **GPU-Omega** for accelerated torsion driving using a GPU if one is present on the system.

- By default, if a GPU is detected at runtime, torsion driving will be carried out on the GPU. To turn this feature off and run on the CPU, set the flag *-useGPU* to false.

- **GPU-Omega** uses the default mmff94smod_NoEstat force field. If attempting to use an alternative force field, conformer generation will fall back to the CPU. As a result of this, **GPU-Omega** is not compatible with the default settings of the **OMEGA** `dense` sampling mode. To take advantage of a GPU with `dense` sampling mode, set *-searchff* to *mmff94smod_NoEstat*.

- **GPU-Omega** does *not* support the distance geometry method of torsion driving, and therefore is incompatible with the `macrocycle` mode.

3.3 Example Commands

This section has a series of example **OMEGA** command line executions. Each example is followed by a brief description of its behavior. Sample data files can be found in the data directory.

3.3.1 Basic Commands

```
prompt> oeomega classic drugs.smi drugs.oeb.gz
prompt> oeomega classic -in drugs.smi -out drugs.oeb.gz
```

These two commands will yield identical results. These execute **OMEGA** `classic` with the default parameters. The file *drugs.smi* is opened in SMILES format for input, and the output is written to the file *drugs.oeb.gz* in gzipped OEBinary format.

```
prompt> oeomega macrocycle -in drugs.smi
prompt> oeomega macrocycle -in pdb_10_examples.oeb -prefix pdbexamples -mpi_np 8
```

The first command executes **OMEGA** `macrocycle` mode with the default parameters. By default, the conformers are written to the file *oeomega_macrocycle_conformers.oeb.gz*. The macrocycle mode execution will take a longer time than the classic mode and usually result in fewer conformers for the same input molecule. The second command executes `macrocycle` mode using an input file with macrocyclic molecules. The conformers are written to the file *pdbexamples_conformers.oeb.gz* and the execution time will be even longer because of the macrocyclic molecules. It is preferable to run the `macrocycle` mode under Open MPI using the maximum number of CPU cores available.

3.3.2 Using Parameters and Param Files

```
prompt> oeomega classic -in drugs.smi -out drugs.oeb.gz -maxconfs 1
```

The parameter *-maxconfs* can be used to specify the maximum number of conformers for each output molecule. This command will generate a single low energy conformer for every molecule in drugs.smi.

```
prompt> oeomega classic -in drugs.smi -out drugs.oeb.gz -param myparameters
```

This command is the same as the basic commands above except for the *-param* flag. It executes **OMEGA** `classic` with the parameters found in the *myparameters* file. The file *drugs.smi* is opened in SMILES format for input, and the output is written to the file *drugs.oeb.gz* in gzipped OEBinary format.

```
prompt> oeomega classic -param myparameters drugs.smi drugs.oeb.gz
prompt> oeomega classic drugs.smi drugs.oeb.gz -param myparameters
```

The first of these two commands will yield exactly the same results as the example above. *drugs.smi* will be mapped to the *-in* flag and *drugs.oeb.gz* will be mapped to the *-out* flag to begin the second to last and last command line arguments, respectively. Unfortunately, the second of these two commands will fail to parse because the implicit input and output arguments are not the final two arguments in the list.

```
prompt> oeomega classic -in drugs.smi -out drugs_maxconfs600.oeb.gz
               -param myparameters -maxconfs 600
```

Again, this is a very similar command. It executes **OMEGA** *classic* using the parameters in the *myparameters* file, except the *-maxconfs* parameter is overridden with the 600 from the command line. The command line *-maxconfs* parameter would take precedence over the value in the parameter file independent of the order of flags on the command line.

3.3.3 Advanced Commands

```
prompt> oeomega classic -in drugs.mol2 -out drugs.oeb.gz -fromCT true
```

The *-fromCT* true flag will cause **OMEGA** *classic* to ignore the input conformations in the *drugs.mol2* file. An initial conformation will be generated by a distance-bounds algorithm from the connection table of the molecules in the input file. Be aware that the default value of *-fromCT* is true, however it is listed explicitly here for emphasis.

```
prompt> oeomega classic -in drugs.smi -out drugs.oeb.gz -log null
```

The *-log* flag normally allows specification of the log file's name. However, *nul* and *null* are reserved names which indicate that no log file will be written (this includes failure modes).

3.3.4 CSV Report File for `macrocycle` Mode

For each molecule in the input file, a row of data with the following columns is recorded:

- ***Molecule***
  Title of the molecule.
- ***SMILES***
  SMILES string corresponding to the molecule.
- ***Conformers***
  Number of conformers in the output file. Its maximum number is always smaller than or equal to the value of the *-maxconfs*.
- ***LargestRing***
  Size of the largest ring system in the molecule.

> **Note**
>
> **All fused rings are considered to be a part of the macrocycle ring. Within this convention, decalin is assumed to be a 10-membered ring system.**

- ***Iterations***
  This number is equal to the parameter *macrocycle -max_iterations*.
- ***Elapsed Time***
  Time used to generate conformers (in seconds).
- ***Status***
  Failure reason in case conformers are not generated.

> **Note**
>
> **Message: "No conformer left after post processing" is usually caused by a chirality mismatch between the chirality of the input molecule and the chirality of each generated conformation. It means that the force field refinement was not able to reproduce input chirality. It is recommended therefore to check the correctness of chirality input molecule. This situation is common when the input is taken from the 3D PDB file. A good example is the PDB structure of the rifampicin antibiotic extracted from the 1SKX entry. This molecule has nine atomic chiral centers and two chiral bonds. Its chirality assignment based on the 3D PDB structure differs from the actual chirality of this antibiotic.**

3.4 Command Line Help

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

```
prompt> oeomega --help
```

will generate the following output:

```
Available Modes:
  classic - The original customizable **OMEGA** interface
  macrocycle - Conformer generation for macrocycles
  fastrocs - Optimal conformer generation for FastROCS
  rocs - Optimal conformer generation for ROCS
  pose - Optimal conformer generation for molecular alignment
         and pose prediction by docking
  dense - Optimal conformer generation for Freeform

For more help type:
  oeomega classic --help
  oeomega macrocycle --help
  oeomega fastrocs --help
  oeomega rocs --help
  oeomega pose --help
  oeomega dense --help
```

There are four help functions available for each mode:

```
Help functions:
  oeomega <mode> --help simple      : Get a list of simple parameters
  oeomega <mode> --help all         : Get a complete list of parameters
  oeomega <mode> --help <parameter> : Get detailed help on a parameter
  oeomega <mode> --help html        : Create an html help file for this program
```

3.5 Required Parameters

-in

File containing one or more molecular connection tables to be processed by **OMEGA**. Multiple input files of the same format can be merged together into a single input file using the *cat* command. A pipe can be used to avoid writing the merged input file to disk.

```
cat filename1.ism filename2.ism | oeomega classic -in .ism -out output.oeb.gz
```

Supported file formats: cxsmiles, ism, mol2, oeb, oez, sdf, smi, and the corresponding compressed formats (i.e., xx.gz). `lst` and `list` formats containing lists of supported files are also applicable.

3.6 Optional Parameters

3.6.1 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 *'_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.

3.6.2 File Options

-out

This file writes conformers generated by **OMEGA**. Gzipped OEBinary is the recommended output format.

Supported file formats: mol2, oeb, oez, sdf, and the corresponding compressed formats (i.e., xx.gz).

-failed

This file writes molecules for which **OMEGA** fails to generate conformers. A failed file is only generated when specifically requested by users.

Supported file formats: cxsmiles, ism, mol2, oeb, oez, sdf, smi, and the corresponding compressed formats (i.e., xx.gz).

-commentEnergy

This flag causes the conformer energy in kcal/mol to be written in the comment field for each conformer. This is particularly useful when writing SD or MOL2 files that will be passed to software that anticipates strain energies written in the comment field. [Default=false]

-includeInput

When true, this boolean flag will include the input conformer in the output file. This requires that the input file format is in a 3D format (e.g., SDF, MOL2, OEB). [Default=false]

`-log`

The argument for this flag specifies the name of the log file. The level of detail for log file information can be altered using the *-verbose* flag. Output can be directed to the terminal as well with the *-progress* log option. 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 by **OMEGA**. Most important among these is the *oeomega_<mode>_parm.txt* file which includes a copy of all the parameters used in the **OMEGA** run. The prefix is also used to generate a default log file name if it is not explicitly specified with the *-log* flag. [Default=oeomega_<mode>].

`-progress`

This flag shows the progress on screen. Options are *none*, *dots*, *log*, and *percent*. The *dots* options will display dots on-screen to show molecules completed. The *log* option will duplicate the log file on-screen. The *percent* option will track progress through the input file. [Default=none]

`-rotorOffsetCompress`

This flag controls the behavior of writing the rotor offset compressed flavor of the OEBinary format. Rotor offset compressed files contain the same information as standard Cartesian OEBinary files but require a small fraction of the storage space. Optimal compression of **OMEGA** output can be obtained by writing gzip compressed OEBinary files with the *-rotorOffsetCompress* flag turned on. [Default=true]

Note

Rotor offset compression cannot be performed and is automatically disabled when any of the methods to fix part of the molecule geometry, such as -fixmol, -fixfile, -fixsmarts, or -fixmcs, are used during conformer generation.

`-sdEnergy`

This flag controls the behavior of writing strain energies as SD tags. This flag may be used with either OEBinary or MDL SD files. [Default=false]

`-verbose`

This is a boolean flag that controls the level of detail written to the log file. By default, **OMEGA** will only write minimal information to the log file. Molecule titles and warning messages constitute the bulk of logging at the default level. Verbose logging will cause more information to be written to the log file in order to follow behavior during program execution. In order to have the log shown on the screen, use *-progress log*. [Default=false]

`-warts`

This boolean flag is used to generate unique titles for conformers that reflect their position in an ensemble of conformations produced by **OMEGA**. The title given to each conformer will begin with the molecule title taken from the input file and appended with an underscore and the integer corresponding to the rank order number of the conformer in the final ensemble. [Default=false]

3.6.3 3D Construction Parameters

`-addfraglib`

[alias: -setfraglib]

The argument that follows this flag is one fragment file, normally generated by MakeFraglib. These fragments will supplement the built-in fragment library.

For use in a Windows platform, an enhanced fragment library, `omega_fragment_lib_2023.oeb.gz`, that improves the performance of conformer generation is available upon request from the database downloads page. The enhanced library is the default on both Linux and OSX.

`-buildff`

This flag sets the force field used for constructing fragments that are assembled to build an initial model of the input structure. Consult the description of force fields (see *Force Fields*) for an explanation of appropriate arguments for this flag. [Default=mmff94smod_NoEstat]

`-canonOrder`

This flag can be used to disable the automatic reordering of input molecules to a canonical atom and bond order. In order to obtain consistent results from different file formats and different connection table orders, **OMEGA** reorders the input connection table. This behavior can be turned off using the *-canonOrder* flag; however, the resultant ensembles will likely be inconsistent in their composition. [Default=true]

`-deleteFixHydrogens`

When a fixfile fragment is specified, all possible mappings (matches) of the fragment and the input molecule are considered for positioning the input molecule. This process begins with a substructure search of the fragment in the input molecule. If hydrogens are included on the fixfile fragment, the number of possible matches will grow exponentially with the number of equivalences. Each geminal hydrogen pair on the fixfile produces a nonproductive match multiplied by all other nonproductive matches. This flag can be used to prevent the explosion of nonproductive matches by deleting hydrogens from the fixfile fragment prior to the substructure search. [Default=true]

`-fixfile`

The argument that follows this flag is a molecule file used to specify the coordinates for a substructure of the input molecules. An initial structure is generated for every input molecule, then a substructure search is performed using the molecule or fragment provided in the fixfile as a query molecule. Every instance of the fixed substructure found in the input molecule up to a predetermined limit (see *maxmatch* and *umatch*) is used to replace the coordinates of the atoms that match the substructure. The input molecule coordinates are aligned relative to the substructure prior to fragment replacement, then the coordinates are taken from the fixed fragment and assigned to the corresponding atoms of the input molecule. A separate alignment, replacement, and conformer search is carried out for every matching substructure in the input molecule. Molecules that do not match the fixfile will fail and will be reported as such in the log file.

`-fixmcs`

This flag fixes a portion of a molecule based on the maximum common subgraph against the *-fixfile* molecule. Molecules that do not have any common subgraph will fail and be reported as such in the log file. [default = false]

`-fixrms`

Fixfile fragments taken from crystallographic sources may differ in their geometry relative to optimal MMFF94 geometries. **OMEGA** attempts to superimpose built structures onto fixfile fragments, and if the geometry differs too greatly, **OMEGA** considers the superposition a poor match and will fail to build a structure using the fixfile. This flag can be used to loosen the default RMS superposition criteria to allow suboptimal superpositions to succeed in spite of the poor geometric complementarity. [Default=0.15]

`-fixsmarts`

This is another way to specify a fixed portion of a molecule. The SMARTS pattern is used to fix a portion of the *-fixfile* primarily, or to fix a portion of the input molecule if *-fromCT* is set to *false* secondarily. Molecules that do not match will fail and be reported as such in the log file.

`-fixmcs`

This flag fixes a portion of a molecule based on the maximum common subgraph against the *-fixfile* molecule. Molecules that do not have any common subgraph will fail and be reported as such in the log file. [Default=false]

`-fromCT`

This boolean flag determines whether **OMEGA** should generate an initial set of 3D coordinates using only the connection table of the input molecule. Initial model generation is always necessary for molecule file formats devoid of coordinates (i.e., SMILES). Bond lengths and angles taken from molecule files containing coordinates may be retained by setting this flag to false. *-enumNitrogen off* must also be set for identical stereochemistry to be guaranteed in the output. [Default=true]

`-maxmatch`

This flag is used to limit the number of fixfile substructure matches in the input molecule. Each match will result in replacement of the matching substructure with coordinates taken from the fixfile fragment. The number of matches may need to be limited using this parameter for a substructure where many matches are possible. [Default=1]

`-mcsMinAtoms`

This parameter is used to define the limit of MCS-based fixes using *-fixmcs*. This is the minimum number of atoms that is required for a MCS-based fix. [Default=1]

`-strictfrags`

This flag sets how **OMEGA** generates fragments that are not in the fragment library. The default is a faster and less rigorous fragment generation than the **MakeFragLib** program, whereas a *true* value will require **OMEGA** to be identical to the **MakeFragLib** program. The most noticeable difference is the time limit for fragment generation, which is increased from 30 seconds to 300 seconds with *-strictfrags* set to `true`. [Default=false]

`-strictatomtyping`

This flag uses strict atom typing for MMFF94 or allows "close enough" atom typing. A *true* setting will fail any molecule that contains an atom type that does not have specified parameters. A *false* setting will allow parameters from a similar atom type to be used. [Default=true]

`-umatch`

The *-umatch* boolean flag determines whether only the unique substructure matches of the fixfile are used for coordinate replacement. A unique substructure match is defined as a match that does not cover the identical set of target atoms as any other substructure match in a set. For example, a benzene substructure will match a benzene ring 12 times. Only one substructure match constitutes a unique match, while the other 11 matches are duplicates. If the flag is set to *false*, then all possible substructure matches may be used for coordinate replacement. This behavior is usually unnecessary as nonunique matches will frequently lead to duplication. [Default=true]

3.6.4 Structure Enumeration

`-enumNitrogen`

The *-enumNitrogen* string flag controls the behavior of **OMEGA** with respect to enumeration of nonterminal nitrogens. Any nitrogen with pyramidal geometry in the initial model of the input molecule and having no more than two ring bonds is considered by OMEGA to be *invertible*. OMEGA will enumerate all possible puckers if the *-enumNitrogen* flag is set to `all`. OMEGA will only enumerate unspecified invertible nitrogens if the flag is set to `unspecified`. Nitrogen enumeration can be turned off by setting the flag to `off`. [Default=unspecified]

`-enumRing`

The *-enumRing* boolean flag controls the behavior of **OMEGA** with respect to ring conformations. If this flag is set to *true*, **OMEGA** will generate all possible combinations of all ring conformations in a molecule. Ring systems with only a single conformation will be replaced with a conformation taken from a fragment file or generated on the fly by **OMEGA**. If this flag is set to *false*, then no ring conformer enumeration or replacement will occur. Initial geometries provided by an input file (see *-fromCT*) may therefore be preserved by setting *-enumRing* to *false* as well. [Default=true]

`-sampleHydrogens`

This flag sets whether hydrogens will be sampled. This option enables sampling of hydrogen locations for -OH, -SH, and amines. [Default=false]

3.6.5 Torsion Driving Parameters

-addtorlib

This flag takes a filename as a parameter. Torsion rules in the file are placed above any previous torsion rules and therefore are matched first (see *Torsion Library Format*).

-erange

The *-erange* flag sets the energy cutoff used as an accept or reject criteria for conformers depending on the number of rotatable bonds in the structure. Any conformer that has a calculated strain energy less than the sum of the energy window and the energy of the global minimum conformer will be accepted. Conformers with strain energies above this threshold are rejected. The energy range is given as a list of values that correspond to the *-rangeIncrement* parameter. For example, *-erange 5.0 10.0 15.0 20.0* used with *-rangeIncrement 3* sets the energy window to 5.0 kcal/mol for structures with zero to two rotatable bonds, 10.0 kcal/mol for structures with three to five rotatable bonds, and so on. The energy window for structures with more rotors than the highest *-erange* value specified will be taken as the highest specified value. [default = 10.0 15.0 15.0 20.0 (classic)]

-ewindow

The *-ewindow* flag sets the energy window in kcal/mol used as an accept or reject criteria for conformers. Any conformer that has a calculated strain energy less than the sum of the energy window and the energy of the global minimum conformer will be accepted. Conformers with strain energies above this threshold are rejected. [Default=10.0 (classic) or 20.0 (macrocycle)]

-maxConfRange

The *-maxConfRange* flag allows the user to specify the maximum number of conformers to be output for a structure based on the number of rotatable bonds in the structure. The maximum conformer range is given as a list of values that correspond to the *-rangeIncrement* parameter. For example, *-maxConfRange 100 150 200 250* used with *-rangeIncrement 3* sets the maximum number of conformers to 100 for structures with zero to two rotatable bonds, 150 for structures with three to five rotatable bonds, and so on. The number of conformers for structures with more rotors than the highest *-maxConfRange* value specified will be taken as the highest specified value.

-maxconfs

The *-maxconfs* flag sets the maximum number of conformations to be included in the output ensemble for each molecule. For molecules with few conformational degrees of freedom, the number of output conformers might be lower than the value set for this flag but will never exceed it. Out of all the generated conformers that satisfy the other ensemble slicing criteria (like *-ewindow* , *-rms*, etc.), the output includes only up to the *-maxconfs* number conformers that are the lowest in energy according force field set in *-searchff* flag. As a special case, setting *-maxconfs 0* along with *-rms 0* will result in **OMEGA** skipping the duplicate removal step and will write all generated conformers that satisfy all other criteria to the output file. [Default=200]

-maxrot

The *-maxrot* flag sets the cutoff for the maximum number of rotatable bonds. Molecules that have rotors less than or equal to the *-maxrot* cutoff will be processed by **OMEGA**. **OMEGA** will not search for conformers of molecules that have more rotors than the *-maxrot* cutoff. By default, **OMEGA** does not apply a cutoff for the number of rotatable bonds. Instead, a desired cutoff must be supplied by the user. [Default=9999]

-maxtime

This flag limits the amount of time (in seconds) spent for the torsion driving search during conformer generation for each molecule. The total time used for each molecule may exceed this limit due to all other steps of conformer generation. [Default=120.0]

-rangeIncrement

The *-rangeIncrement* is used to control the range for the number of rotatable bonds used with the *-maxConfRange*, *-rmsrange*, and *-erange* flags. The preceding flags are used to control the maximum number of conformers, RMS cutoff, and energy windows that are dependent on the number of rotors in a given structure. [Default=5]

-rms

The *-rms* flag sets the minimum Root Mean Square (RMS) Cartesian distance below which two conformers are duplicates. The RMS calculation is performed after superposition such that the true minimum distance between conformers is calculated. Lowering the *-rms* value may cause **OMEGA** to generate ensembles that contain more representative conformers of a similar shape. Higher *-rms* values may result in smaller, yet possibly more, shape-diverse ensembles. [Default=0.5]

-rmsrange

The *-rmsrange* flag allows the user to specify the RMS cutoff used for duplicate conformer removal based on the number of rotatable bonds in the structure. The RMS range is given as a list of values that correspond to the *-rangeIncrement* parameter. For example, *-rmsrange 0.8 1.0 1.2 1.4* used with *-rangeIncrement 3* sets the RMS cutoff value to 0.8 for structures with zero to two rotatable bonds, 1.0 for structures with three to five rotatable bonds, and so on. The RMS cutoff value for structures with more rotors than the highest *-rmsrange* value specified will be taken as the highest specified value.

-searchff

This flag sets the force field used to calculate strain energies of conformers generated during a torsion search. Consult the description of force fields (see *Force Fields*) for an explanation of appropriate arguments for this flag. [Default=mmff94smod_NoEstat]

-settorlib

The *-settorlib* flag is used to specify the file name of the file containing rules used in resolution control of the torsion driving part of conformer generation. Refer to the *Torsion Library Format* section describing the format of the torsion library.

-thompson

The *-thompson* flag sets whether to use Thompson sampling for exploring the conformer space. **OMEGA** will use this by default. If switched off, exhaustive sampling will be used. [default = true]

-torlibtype

The *-torlibtype* flag is used to specify the type of built-in torsion library. There are two types: *original*, and *guba*. The *original* is the default OMEGA torsion library and the *guba* [Guba-2016] is available in OMEGA 3.0 and above. [Default=original]

-useGPU

The *-useGPU* flag is used to engage GPU-Omega to accelerate torsion driving on a GPU if one is available on the system. See *GPU-Omega* for details. [Default=true]

-thompson

The *-thompson* flag sets whether to use Thompson sampling for exploring the conformer space. OMEGA will use exhaustive sampling by default. If switched Thompson sampling is used, the parameters can be controlled by the Thompson Sampling Options class. [Default=false]

3.6.6 Thompson Sampling Parameters

-maxTrials

This parameter sets a limit on the number of Thompson sampling trials. OMEGA will stop exploration of the conformer space once this limit is reached, irrespective of how many conformers have been found so far. The default (and recommended) way of controlling this quantity, however, is through the *-maxTrialsRange* parameter. [Default=0]

-maxTrialsRange

This flag sets the maximum number of trials in Thompson sampling depending on the number of rotatable bonds in a given molecule. The range is determined by the *-trialsRangeIncrement* parameter. For example, *-maxTrialsRange 1000 2000 3000 5000* used with *-trialsRangeIncrement 3* sets the maximum number of trials

to 1000 for molecules with 0–2 rotors, 2000 for 3–5 rotors, 3000 for 6–8 rotors, and 5000 for all molecules with 9 or more rotors.

`-trialsRangeIncrement`

This parameter is used to control the dependence of maximum number of Thompson sampling trials on the number of rotors in the molecule along with the SetMaxTrialsRange parameter. [Default=1].

`-batchMaxRotors`

This parameter controls the maximum allowed number of rotors for which batching is done; that is, two samples are drawn from the current distributions to form all these rotors. The resulting $2^{BatchMaxRotors}$ conformers are processed in a batch and the distributions associated with the rotors are then updated. Higher *-batchMaxRotors* values result in a lower runtime, but for large enough batches, accuracy can be compromised as the learning is not reinforced frequently enough to influence many samples. [Default=2]

`-priorTemperature`

This parameter is used to set the initial probabilities (prior distributions) associated with each angle in the torsion rule according to values of respective strain associated with the angles. If a rule does not provide these strain values, all angles are assumed to be at zero strain and equally likely. For nonzero strains, this parameter acts as a Boltzmann temperature and gives the probability as $= exp(-1 \times \frac{strain}{PriorTemperature})$. The lower the parameter, the stronger the initial preference for the angles with low strain.

For instance, a torsion angle with strain of 5 kcal/mol will be initialized to a probability of 0.7% for *-priorTemperature* of 1, 60.6% for *-priorTemperature* = 10 and 95% for *-priorTemperature* = 100. [Default=100.0]

3.6.7 Stereo Parameters

`-flipper`

This parameter creates an ensemble of stereoisomers before passing the molecules for conformer generation. A value of *true* will flip all unspecified stereo centers and a value of *force* will flip all stereo centers. [Default=false]

`-flipper_enhstereo`

This flag considers enhanced stereo information during stereo-enumeration. [default = false]

`-flipper_maxcenters`

This flag sets the maximum number of stereocenters that can be flipped exhaustively. The number of molecules generated by enumerating the stereocenters is 2^N, where *N* is the number of stereocenters. In some instances, this may be larger than is desired. The *-flipper_maxcenters* parameter indicates the maximum number of stereocenters which will be fully enumerated. If a molecule has more than *-flipper_maxcenters* stereocenters, **OMEGA** will randomly enumerate *2^(flipper_maxcenters)* instances from the full set of potential isomers. [Default=12]

`-flipper_warts`

This flag adds a wart to the title of each stereoisomer. [Default=false]

`-flipper_enhstereo`

This flag considers enhanced stereo information during stereo-enumeration. [Default=false]

`-strictstereo`

This parameter requires all stereo centers to be specified. Molecules sent to conformer generation with unspecified stereo will fail. [Default=true]

3.6.8 General

`-strict`

This is a convenience flag for setting *-strictstereo*, *-strictatomtyping*, and *-strictfrags* all at once. This setting will override any individual settings for these flags.

3.6.9 Options Specific to the Macrocycle Mode

Execute Options

`-mpi_hostfile`

This is the path to the host file to be used for launching MPI processes.

`-mpi_np`

This is the number of MPI processors to launch. If this is not used, the job will run on a single processor.

> **Note**
>
> **The number of processors should be set to a reasonable number, preferably the maximum number of CPU cores available.**

`-param`

This is a parameter file name.

File Options

`-in`

This is the input molecular file name. Contrary to oeomega classic, pipe streaming cannot be used.

`-molnames`

This parameter specifies 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) (parameter `-in`) with names that match those in the text files will be read.

The general purpose of this flag is to provide an easy mechanism for reading a few specific molecules that are contained in a large database, without having to extract those molecules by hand from the database.

`-out`

This is the output molecular file name. The following SD tags will be attached to the output: "Macrocycle Energy," "Relative Energy," "Macrocycle Iteration," "Macrocycle Duplicate Minima Count," and "RMSD from Input Structure." Important note: the macrocycle mode does *not* support the *-rotorOffsetCompress* option.

`-prefix`

This is the prefix used to name output files.

`-progress`

This shows progress on the screen. Options are *none*, *dots*, *log*, and *percent*. The *dots* options will display dots on the screen to show molecules that are completed. The *log* option will duplicate the log file on screen. The *percent* option will track progress through the input file. [Default=none]

Calculation Options

`-ewindow`

This is the energy window for output conformers. All conformations with energy higher than the energy window will be skipped from the molecular output file.

`-iteration_cycle_size`

This is the number of iterations to run before checking if a new minimum was found (the run will finish if no new minimum is found). [Default=100] [Aliases=-cycle_size]

`-max_iterations`

This is the maximum number of iterations (the calculation may converge before reaching this number). Each iteration is a single conformation generation which originates from a unique random atom (or fragment) placement. [Default=2000] [Aliases=-max_iter]

-ref_tolerance

This is the RMS gradient tolerance for force field refinement. [Default=0.001]

-rms

This is the deduplication threshold. [Default=0.5 Å].

-maxconfs

The *-maxconfs* flag sets the maximum number of conformations to be included in the output ensemble for each molecule. For molecules with few conformational degrees of freedom, the number of output conformers might be lower than the value set for this flag but will never exceed it. [Default=400]

3.7 Additional Specifications

3.7.1 OMEGA Modes and Parameters

OMEGA provides the facility for users to select one of a number of pre-defined `modes`. Except for the `macrocycle` mode, all of the other modes share the same set of parameters. However, the default parameters vary between the various modes.

Default parameters for each mode are described below. While the default parameters for each mode may vary, the user has the ability to modify any parameter in any mode.

classic:

The `classic` mode is the original customizable OMEGA interface. All **OMEGA** default parameters mentioned in the *Optional Parameters* refer to the `classic` mode.

```
-maxConfRange = 200
-rangeIncrement = 4
-eRange = "10, 15, 15, 20"
-rms = 0.5
-searchff = mmff94smod_NoEstat
```

fastrocs:

For `fastrocs` mode, all default parameters are identical to the default parameters in `classic` mode except for the following changes:

```
-maxconfs = 10
-eWindow = 10
```

rocs:

For `rocs` mode, all default parameters are identical to the default parameters in `classic` mode except for the following changes:

```
-maxconfs = 50
-eWindow = 10
```

pose:

For `pose` mode, all default parameters are identical to the default parameters in `classic` mode except for the following changes:

```
-maxConfRange = "200,200,800"
-eRange = "10, 15, 25, 30"
```

dense:

For `dense` mode, all default parameters are identical to the default parameters in `classic` mode except for the following changes:

```
-searchff = mmff94s_sheff
-strictstereo = false
-ewindow = 15
-maxtime = 3600
-rms = 0.3
-maxrot = 20
-maxconfs = 20000
-thompson = false
```

macrocycle:

The `macrocycle` mode has its own set of default parameters as described in the *Options Specific to the Macrocycle Mode*.

Note

All molecules containing 10-membered or larger rings are considered by **OMEGA to be macrocycles except those which are made entirely by fused small rings. For example, cyclodecane and porphyrin are macrocycles but decaline, naphthalene, and all polycyclic aromatic hydrocarbons are not.**

3.7.2 Force Fields

OMEGA provides the facility for users to select one of a number of predefined force fields. The force field selected may be different for model construction and torsion search. The ability to select a force field provides a mechanism for task specificity. Some force fields may be more appropriate for solution phase ensemble generation, while others may excel for bioactive conformer reproduction. The following predefined force fields may be used as arguments to the *-buildff* and *-searchff* flags.

- *mmff* This is an exact reproduction of the published Merck Molecular Force Field (MMFF94) with additional atom types parameterized using the same algorithm.
- *mmff_NoEstat* This force field variant includes all MMFF94 terms except Coulomb interactions.
- *mmff_Trunc* This force field variant excludes both Coulomb interactions and the attractive part of van der Waals interactions. All other components of the MMFF94 force field are calculated according to literature specifications.
- *mmff_Sheff* This force field variant includes all MMFF94 terms, and the Sheffield solvation model [Grant-2007] is used during the calculation of energies for the conformers generated.
- *mmff94s* This is an exact reproduction of the 94s variant of the Merck Molecular force Field (MMFF94s) with additional atom types parameterized using the same algorithm.
- *mmff94s_NoEstat* This force field variant includes all MMFF94s terms except Coulomb interactions.
- *mmff94s_Trunc* This force field variant excludes both Coulomb interactions and the attractive part of van der Waals interactions. All other components of the MMFF94s force field are calculated according to literature specifications.
- *mmff94s_Sheff* This force field variant includes all MMFF94s terms, and the Sheffield solvation model [Grant-2007] is used during the calculation of energies for the conformers generated.

- *mmff94smod* This is an exact reproduction of the 94s variant of the Merck Molecular force Field (MMFF94s) with additional atom types parameterized using the same algorithm and some torsion interaction parameters modified to produce desired equatorial conformers for monosubstituted cyclohexanes.
- *mmff94smod_NoEstat* This force field variant includes all MMFF94s-Mod terms except Coulomb interactions.
- *mmff94smod_Trunc* This force field variant excludes both Coulomb interactions and the attractive part of van der Waals interactions. All other components of the MMFF94s-Mod force field are calculated according to literature specifications.
- *mmff94smod_Sheff* This force field variant includes all MMFF94s-Mod terms, and the Sheffield solvation model [Grant-2007] is used during the calculation of energies for the conformers generated.
- *sage* This is an exact reproduction of the published SAGE force field, OpenFF-2.2.0, from the Open Force Field Initiative.
- *sage_NoEstat* This force field variant includes all SAGE terms except Coulomb interactions.
- *sage_Sheff* This force field variant includes all SAGE terms and the Sheffield solvation model [Grant-2007] is used during the calculation of energies for the conformers generated.

3.7.3 Torsion Library Format

A file of alternate torsion rules may be specified with the *-settorlib* command. **OMEGA** will match only the first rule found for a torsion angle and will discontinue matching alternate possible rules. Thus, a correctly ordered torsion file will be arranged with the most specific patterns appearing at the top of the file, and more general patterns appearing toward the end. Simple torsion rules are composed of a single SMARTS pattern containing at least four atom expressions, followed by a listing of the torsion angles that **OMEGA** will sample. Each reference atom in the SMARTS pattern that is used to define the torsion angle being sampled must have a map index (numbered 1 through 4) specified that indicates the ordering of the atoms in the torsion angle. The pattern must appear all on a single line, with a carriage return separating one rule from the next. Comments in the file must be preceded with a # character. The following is an example of a simple torsion rule.

```
#methyl ester pattern
[O:1]=[C:2]-[O:3][CH3:4] 0
```

More advanced rules may be included that alter the energy calculation for particular torsion angles. In these types of torsion rules, a SMARTS pattern with associated map indices is still used to define the molecular environment in which the rule is to be applied, but the sampled values appear on subsequent lines with one torsion angle per line. The first number per line indicates the torsion angle, in degrees, that **OMEGA** must sample. If a second number follows a torsion angle on the same line, the value is added to the total energy computed for that conformer. The following is an example of an advanced torsion rule.

```
#experimental structure test
O=[C:1][NX3H:2][c:3]([cH,nH0])[nH:4]
0
180 10.0
<end>
```

CHAPTER

FOUR

UTILITIES

4.1 Flipper

4.1.1 Overview

The **OMEGA** distribution includes **Flipper**, a utility program for enumerating stereocenters in a molecule. We have taken the philosophy that conformer generation and stereochemistry enumeration are related but distinct problems. Thus we provide **Flipper** as part of the **OMEGA** product, but not as part of the same executable.

The current version of **Flipper** can handle enumeration of R/S stereochemistry (including N and S stereochemistry) as well as cis/trans stereochemistry. **Flipper** uses graph algorithms to determine which atoms are stereocenters. If a stereocenter has a designated stereochemistry, by default **Flipper** does not change the stereochemistry. However, if a stereocenter does not have a specified stereochemistry, **Flipper** will enumerate both stereochemistry states of the stereocenter.

Many common file formats are limited in their ability to fully express stereochemistry without resorting to Cartesian coordinates. This can lead to ambiguity in whether a particular conformer is supposed to represent a racemic mixture or an isolated stereoisomer. SD file format performs better than most, as it allows the designation of R, S, or racemic on each tetrahedral stereocenter independent of the particular conformation. However, SD format is not as robust for cis/trans stereochemistry. Isomeric SMILES is the only common file format which allows complete specification of R/S and cis/trans stereochemistry. In fact, although **Flipper** does not support higher orders of stereochemistry, the isomeric SMILES format is capable of handling them. For this reason, **Flipper** output is limited to SD format (.sdf or .mol) and isomeric SMILES format (.ism). **OpenEye** OEBinary format, like isomeric SMILES, is capable of fully specifying stereochemistry; however, isomeric SMILES is more efficient for compounds without any data beyond the title and connection table.

The **Flipper** process can also be done with **OMEGA** using the *-flipper* flag.

Input Dimensionality

Flipper is designed for use prior to 3D coordinate generation. So 2D or 1D (SMILES) input is normally expected. Input files containing 3D coordinates are handled, but the 3D coordinates are ignored and not transferred to the output.

4.1.2 Example Commands

Below are three example *flipper* executions. Each example is followed by a brief description of the parameter's effects. Sample data files can be found in the data directory.

```
flipper -in drugs.smi -out enumerated.ism
```

This execution will examine each stereocenter in *drugs.smi*. If the stereocenter does not have a specified stereochemistry, **Flipper** will generate molecules with each of the stereochemistry states at each stereocenter.

```
flipper -in drugs.smi -out enumerated_forceflip.ism -forceflip true
```

This execution will differ from the first in that **all** stereocenters will be enumerated, regardless of whether the stereochemistry is specified in the input file.

```
flipper -in drugs.smi -out enumerated_maxcenters6.ism -maxcenters 6
```

In this execution, **Flipper** will exhaustively enumerate all unspecified stereocenters in molecules with up to six stereocenters. For molecules with more than six stereocenters, a random set of 64 isomers will be generated from the larger set of potential stereoisomers.

4.1.3 Command Line Help

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

```
flipper --help
```

will generate the following output:

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

4.1.4 Required Parameters

-in

File containing one or more molecules for which stereoisomers will be generated.

4.1.5 Optional Parameters

-out

Output molecule file name. The output file format is limited to OEB, isomeric SMILES (*.ism* and *.smi*), or SD formats since these are the only formats which fully specify stereocenters without the need for three-dimensional coordinates.

-enhstereo

The *-enhstereo* string flag controls the behavior of **Flipper** with respect to enumeration of stereocenters in the presence of enhanced stereogroups. When true, stereocenters in enhanced stereogroups are enumerated to retain relative stereo relationships between centers within a given stereogroup. [Default=false]

-enumAtoms

Enumerate atom stereocenters in a molecule according to the atom qualifier flags. [Default=true]

-enumAtomSpecifiedStereo

Modify atom stereocenters in a molecule. If *false*, only atoms which do not already have a specified stereochemistry would be enumerated. [Default=false]

-enumBonds

Enumerate bond stereocenters in a molecule according to the bond qualifier flags. [Default=true]

-enumBondSpecifiedStereo

Modify cis/trans bonds in a molecule. If *false*, only bonds which do not already have a specified parity would be enumerated. [Default=false]

-enumBridgehead

Enumerate bridgehead stereocenters. [Default=false]

-enumEZ

Enumerate E/Z stereocenters only. [Default=false]

-enumNitrogen

The *-enumNitrogen* string flag controls the behavior of **Flipper** with respect to enumeration of nonterminal nitrogens. Any nitrogen with pyramidal geometry in the initial model of the input molecule, and having no more than two ring bonds is considered by **OMEGA** to be *invertible*. **Flipper** will enumerate all possible puckers if the *-enumNitrogen* flag is set to *True*. [Default = False] [Aliases: -en]

-enumRS

Enumerate R/S stereocenters only. [Default=false]

-enumRelativeAtomStereo

Enumerate relative atom stereocenters in a molecule. [Default=false]

-enumSpecifiedStereo

This parameter forces **Flipper** to modify all of the stereocenters in a molecule. If *-enumSpecifiedStereo* is false (the default),

FLIPPER only enumerates stereocenters which do not already have a specified stereochemistry. [Default=false] [Aliases=-forceflip]

-maxCenters

Obviously, the number of molecules generated by enumerating the stereocenters is 2^N, where N is the number of stereocenters. In some instances, this may be larger than is desired. The *-maxCenters* parameter indicates the maximum number of stereocenters which will be fully enumerated. If a molecule has more than *-maxCenters* stereocenters, **Flipper** will randomly enumerate *2^(maxCenters)* instances from the full set of potential isomers. [Default=12] [Aliases=-maxcenters]

-molNames

Molecule names file corresponding to the input file.

-prefix

Prefix to use to name output files.

-progress

Type of progress to show on screen during run.

-warts

Add wart to title of each stereoisomer. [Default=false]

4.2 MakeFragLib

4.2.1 Overview

MakeFragLib identifies, extracts, and generates conformers of molecular fragments. The model builder in **OMEGA** fragments input molecules, retrieves corresponding fragment conformers, and assembles them into a three dimensional structure. Pregenerating fragment conformers using **MakeFragLib** accelerates the model building process.

The model builder in **OMEGA** attempts to capture all of the relevant bond lengths, bond angles, and ring conformations for an input molecule. Fragment assembly allows for a "divide and conquer" approach to model building. Much of the relevant information for building a molecular model is contained in carefully chosen fragments. *MakeFragLib* uses distance constraints and geometry optimization to generate fragment conformations. Only a single conformation is stored for acyclic fragments, while all possible unique ring conformers are retained within user defined limits. Libraries created with **MakeFragLib** can then be provided to **OMEGA** using the *-setfraglib* or *-addfraglib* flags.

MakeFragLib can be used from the command line to generate ring templates for use with **OMEGA** starting from only a collection of molecules. The output file from **MakeFragLib** can then be passed directly into **OMEGA** using the *-setfraglib* or *-addfraglib* flag. Although fragment libraries may be concatenated using the Unix 'cat' command, this is not strictly necessary as multiple fragment libraries may be specified as one argument to the flag.

4.2.2 Example Commands

The following section shows several common command line executions of *MakeFragLib*. Each example is followed by an explanation of what the program will do. Sample data files can be found in the data directory.

```
prompt> makefraglib -in drugs.smi -out drugs_fraglib.oeb
```

All of the molecules in *drugs.smi* will be processed. The set of all unique fragments (according to **OMEGA** fragmentation rules) from all molecules in *drugs.smi* will be collected.

```
prompt> makefraglib -in vendor.smi -out new_drugs_fraglib.oeb -ewindow 5.0
```

Conformers will be generated for all of the unique fragments found in the user-supplied file *vendor.smi*. An energy window of 5 kcal above the global minimum conformer will be used to accept or reject conformations of flexible ring systems.

4.2.3 Command Line Help

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

```
> makefraglib --help
```

will generate the following output:

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

4.2.4 Required Parameters

-in

File containing one or more molecules from which fragments will be generated.

-out

File to write fragments generated by **MakeFragLib**. OEBinary format is required and gzipped OEBinary is recommended.

4.2.5 Optional Parameters

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.

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].

-progress

Show progress on screen. Options are 'none', 'dots', 'log' and 'percent'. The 'dots' options will displays dots on screen to show molecules completed. The 'log' option will duplicate the log file on screen. The 'percent' option will track progress through the input file. [default = none]

3D Construction Parameters

-buildff

This flag sets the force field used for constructing fragments that are assembled to build an initial model of the input structure. Consult the description of Force Fields (see *Force Fields*) for an explanation of appropriate arguments for this flag. [Default=mmff94smod_NoEstat] [Aliases=**-bldff**,**-bldFF**]

-fromCT

This boolean flag determines whether *makefraglib* should generate an initial set of 3D coordinates using only the connection table of the input molecule. Initial model generation is always necessary for molecule file formats devoid of coordinates (i.e., SMILES). Bond lengths and angles taken from molecule files containing coordinates may be retained by setting this flag to false. [Default=true] [Aliases=**-fromct**]

-ewindow

-ewindowRing

These flags set the energy window in accepting or rejecting fragment conformers. Any fragment conformer that has a calculated strain energy less than the sum of the energy window and the energy of the global minimum fragment conformer will be accepted. Conformers with strain energies above this threshold are rejected. [Default=4.0]

-rms

-rmsRing

These flags set the minimum Root Mean Square (RMS) Cartesian distance below which two fragment conformers are duplicates. The RMS calculation is performed after superposition such that the true minimum distance between fragment conformers is calculated. Lowering the value may cause **MakeFragLib** to generate ensembles that contain more fragment conformers of a similar shape. Higher values may result in smaller, yet possibly more shape diverse ensembles. [Default=0.1]

-startfact

-startfactRing

Factor for determining number of random starting geometries when generating fragments. Legal values are from 1 to 100. [Default=2, 20]

-fragGen

-fragGenRing

Maximum Number of conformers to be generated with distance geometry for each fragment. [Default=40, 50000]

-fragKeep

-fragKeepRing

Maximum Number of conformers to be kept for each fragment. [Default=1, 1024]

-timeLimit

-timeLimitRing

Maximum time in sec allowed for each fragment generation. [Default=300]

-maxtime

Maximum time (s) allowed for fragment conformer search. [Default=300.0] [Aliases=**-maxsearchtime**]

4.3 Oeb2SdConf

Oeb2SdConf is a utility that allows **OMEGA**-generated conformers to be used with the MOE program. Please contact Chemical Computing Group support if you require assistance with this utility.

CHAPTER

FIVE

THEORY

5.1 OMEGA Overview

OMEGA is a fast, systematic conformer generator that produces high-quality 3D conformations of drug-like molecules. Conformation generation is a foundational step in many molecular modeling workflows because the biologically relevant properties of a molecule and the interactions with its environment depend on the accessible three-dimensional shapes it can adopt. **OMEGA** generates conformer ensembles rapidly while maintaining accuracy, affordability, robustness across diverse chemical space, and utility for downstream applications.

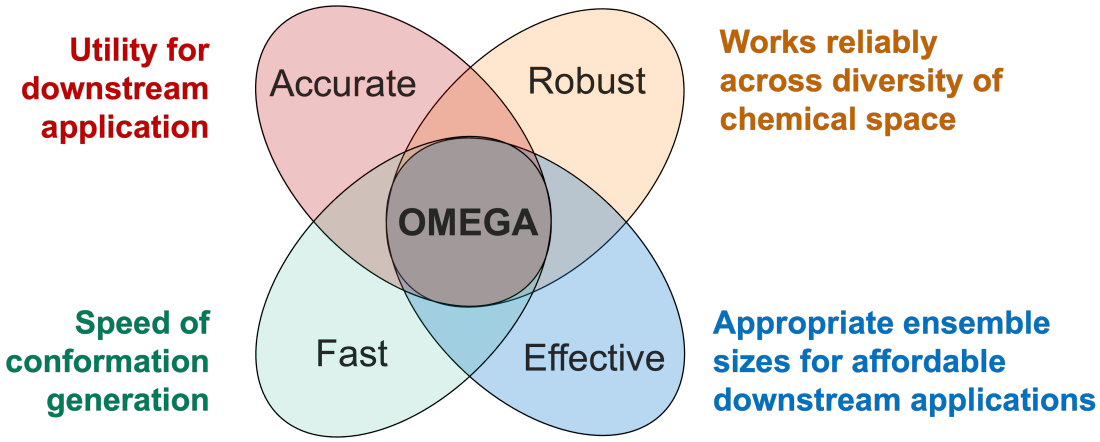


Fig. 1: **Figure 1.** Key objectives of the **OMEGA** conformer generator: speed, accuracy, and broad coverage of chemical space.

5.1.1 Two Approaches to Conformer Generation

OMEGA uses two main approaches to generate 3D conformers based on the topology of the input molecule.

- **Torsion Driving (TD):** Used for linear molecules with only small rings. This method samples the torsional degrees of freedom along rotatable bonds, making it fast and effective at covering the conformational space of typical drug-like molecules.
- **Distance Geometry (DG):** Used for macrocycles, which are molecules with large rings. In this approach, atoms and rigid fragments are placed randomly in Cartesian space, then an error function is minimized to satisfy distance constraints derived from MMFF94 force field parameters, followed by force field refinement. This approach is general and provides thorough conformational coverage, making it especially suited for macrocycles where torsion driving cannot be applied.

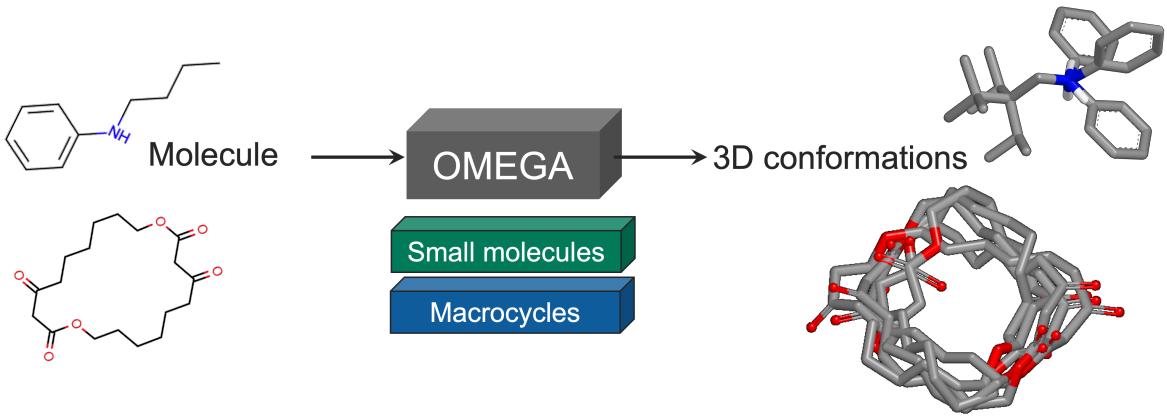


Fig. 2: **Figure 2.** Overview of the two conformer generation approaches in **OMEGA**: distance geometry for macrocycles and torsion driving for linear molecules.

5.1.2 Torsion Driving (TD) in OMEGA

The torsion driving (TD) algorithm of **OMEGA** is composed of two main components; building the molecular scaffold (model) from non-rotor fragments and sampling the torsional degrees of freedom. Model generation may be bypassed by importing fragment structures from external sources ([Hawkins-2010]; [Hawkins-2012]).

OMEGA builds initial structures for the molecule by assembling fragment templates along sigma bonds. Input molecules' graphs are fragmented at exocyclic sigma, and carbon to heteroatom acyclic (but not exocyclic) sigma bonds. Conformations for the fragments are either retrieved from pregenerated libraries built with *MakeFraglib* or constructed on-the-fly using the same approach that *MakeFraglib* uses. Molecule assembly is accomplished by simple vector alignment since all interfragment joints are along sigma bonds.

Once an initial structure is constructed or given as input, **OMEGA** generates additional ones enumerating ring conformations and invertible nitrogen atoms. Ring conformations are taken from the same fragment library used to build the initial model. **OMEGA** detaches all exocyclic substituents from a ring system, then aligns and attaches them relative to the new ring conformation. **OMEGA** attempts to generate every possible combination of ring conformations possible for a given structure.

The next step in model generation is to detect and enumerate invertible nitrogens. Nitrogens that have pyramidal geometry, no specified stereochemistry, no more than one hydrogen, are three valent, and have no more than three ring bonds are considered by **OMEGA** to be invertible. Invertible in this context simply means that at room temperature a pyramidal nitrogen is likely to be able to rapidly (on an NMR timescale) interconvert between two puckered forms. All multiconformer ring models are further expanded by enumerating all possible nitrogen puckers. The resulting model set is the starting point for conformer search by torsion driving.

OMEGA begins the torsion search process by examining the molecular graph and determining the bonds that may freely rotate. By default, **OMEGA** selects acyclic sigma bonds that have at least one non-hydrogen atom attached to each end of the bond. By default, hydrogen rotors (e.g., hydroxyl groups) are not altered during the torsion search; however, this can be enabled. The final ensemble selection is based on RMS distance of heavy atoms and sampled hydrogen atoms; unsampled hydrogen atoms do not affect this RMS distance. A list of possible dihedral angles are then assigned to each rotatable bond. The current mechanism for assignment is based on SMARTS matching, although alternate strategies for assigning angles based on experimental (i.e., X-ray) or theoretical (i.e., fragment optimization studies) are possible. The molecular graph is then subjected to pattern and geometric symmetry detection. Common patterns such as para-disubstituted benzene are used to reduce the number of symmetry equivalent dihedral angles that need to be searched. All torsions are altered by 120 and 180 degrees, and an RMS calculation is performed, taking into account symmetry-equivalent atoms in order to detect two- and three-fold symmetries. Once a set of nonredundant dihedral angles is assigned to each rotatable bond, **OMEGA** samples the torsion space through a systematic enumeration of

combinations of these angles. Depending on the flexibility of the molecule, the maximum number of conformers to be generated, and the mode in which **OMEGA** is run, the algorithm to sample the torsion space is decided. By default, Thompson sampling, a Bayesian reinforcement learning algorithm, is used. The best conformers identified in the torsion search are rank-ordered by energy. A final ensemble is selected by sequentially testing the conformers using the RMS distance cutoff. To be accepted into the final ensemble, a conformer must have an RMS distance to every other member of the ensemble that exceeds the user-defined cutoff value. The final ensemble is populated up to the user-defined maximum ensemble size limit, or until the list of low energy conformers is exhausted.

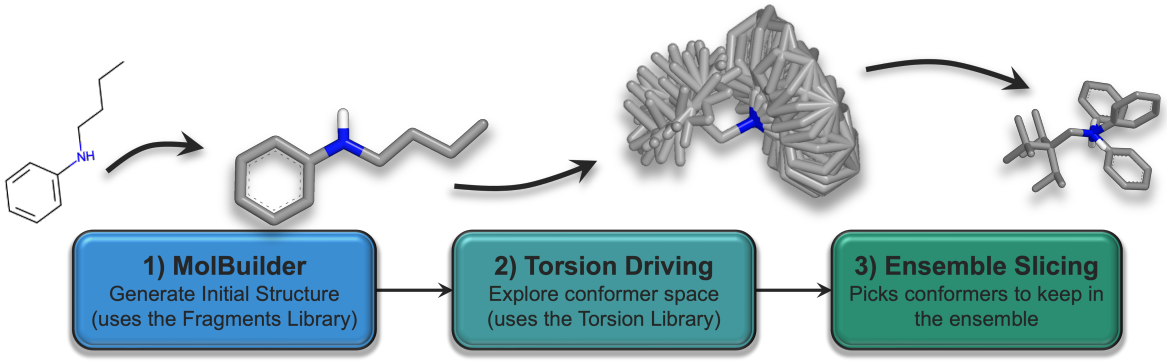


Fig. 3: **Figure 1.** Illustration of steps in torsion driving **OMEGA**.

5.1.3 Thompson Sampling in OMEGA

The torsion library provides a set of likely dihedral angles for each rotatable bond. In Thompson sampling for torsion driving, we use these angles from the torsion rules as priors for a Bayesian process. Instead of enumerating every possible combination of angles and searching the combinatorially expanding space of conformers, conformers are instead generated based on these priors. For every angle for a given rotatable bond, a distribution of beliefs is maintained for modeling the likelihood of that angle to be part of a new low strain conformer. We sample these distributions to pick an angle for a given rotor. This process is repeated for every rotatable bond to pick a combination of angles that defines a conformer. After a conformer is generated, its energy is evaluated and the beliefs are updated based on the result. As we generate more conformers while continuously updating the beliefs, the sampling process is steered toward the low-strain structures of interest.

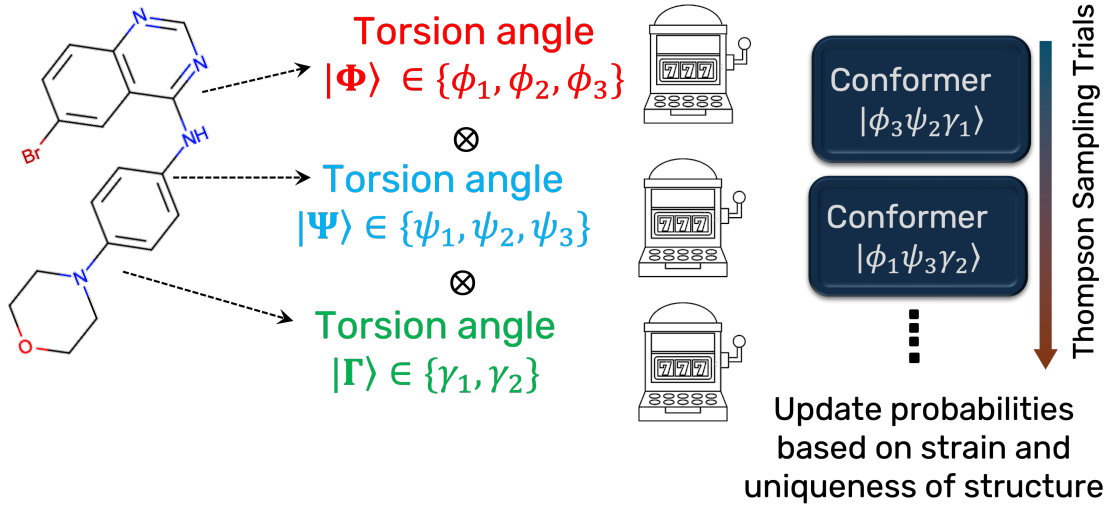


Fig. 4: **Figure 2.** Illustration of Thompson sampling algorithm in torsion driving **OMEGA**.

> **Note**
>
> Conformers generated from **OMEGA** with default options do not maintain input relative stereochemistry, that is, cis/trans configurations for cyclohexane or other cyclic molecules.

5.1.4 OMEGA Sampling Modes

OMEGA provides a set of predefined sampling modes, each tuned for a specific downstream application. Selecting the right mode ensures that the output ensemble is appropriately sized and shaped for the intended use case: not too sparse to miss binding-relevant geometries, but not so dense as to be computationally unaffordable in the downstream step. All modes except `macrocycle` (see the *Macrocycle Conformations section*) share the same underlying torsion driving (TD) algorithm described in *Torsion Driving in Omega*, but they differ in the default values of key parameters such as the maximum number of conformers, energy window, and RMS pruning threshold.

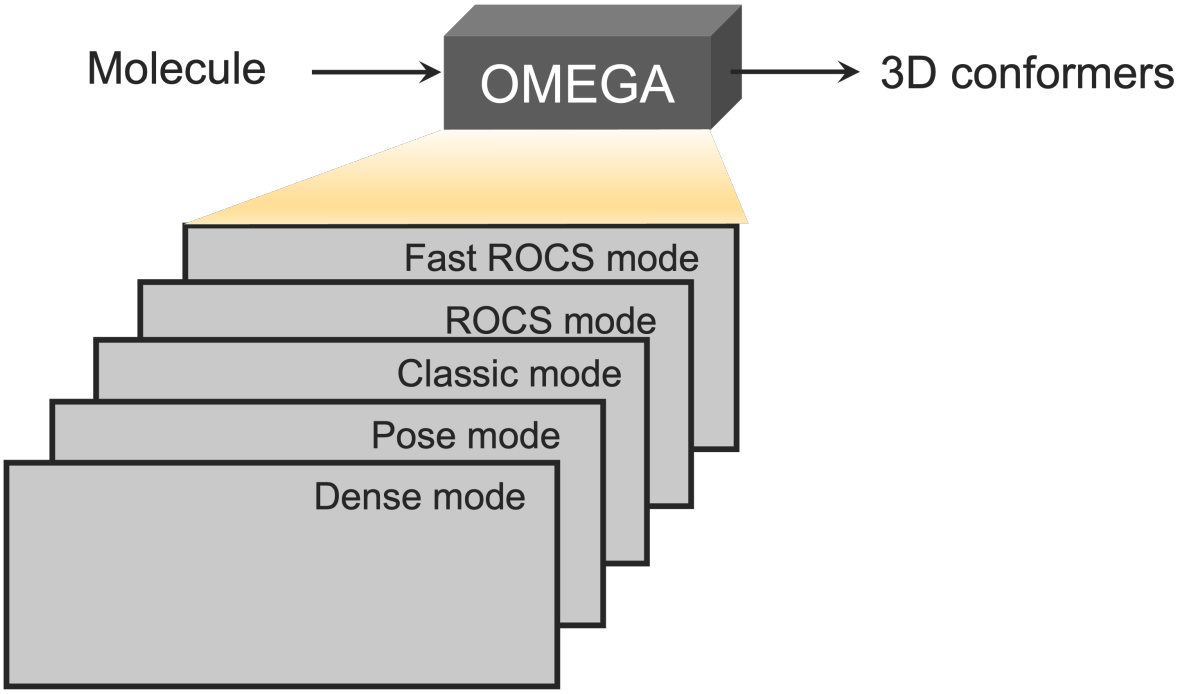


Fig. 5: **Figure 1.** The torsion driving sampling modes in **OMEGA**.

The `classic` mode is the default mode. This allows a maximum of 200 conformations for a molecule. These conformers represent a gas-phase ensemble. The `fastrocs` and `rocs` modes are tailored toward effective use in shape-based virtual screening applications: **FastROCS** and **ROCS**, with a maximum of 10 and 50 conformations, respectively. The `pose` mode is designed for docking with **OEDocking** (FRED/HYBRID) and allows up to 800 conformations for molecules with eight or more rotors. The `dense` mode is designed for free-energy or solvation calculations with **Freeform** and generates large ensembles (up to 20,000 conformations) with a tight RMS pruning threshold and a wide energy window to provide thorough coverage of accessible conformational space.

Each of these modes not only has its own set of default parameters for ensemble slicing that are optimized for the intended downstream application, but the Thompson sampling strategy is also tailored for each mode, based on the typical flexibility of the molecule and the maximum number of conformers generated in that mode.

Choosing a Mode

The table below summarizes the recommended mode for common downstream applications:

Mode	Downstream Application	Key Characteristic
`classic`	General purpose	Gas-phase ensemble with balanced defaults
`fastrocs`	FastROCS GPU screening	Compact ensemble (up to 10 confs)
`rocs`	ROCS shape screening	Moderately sized, representative ensemble (up to 50 confs)
`pose`	OEDocking (FRED/HYBRID)	Wide energy window for bioactive conformers, up to 800 confs for flexible molecules
`dense`	Freeform free energy	Very dense ensemble with solvation scoring

5.1.5 Macrocycle Conformations

Torsion driving conformational sampling methods such as described in the *Torsion Driving (TD) in OMEGA* section often perform poorly for macrocyclic molecules due to the problem of ring closure after the torsion driving step. Conformational sampling of macrocycles therefore requires a different approach; we address this issue with **OMEGA** using distance geometry (DG) [Hawkins-2020]. Other methods have been applied to this problem, such as distance geometry (DG) [Crippen-1988], molecular dynamics (MD/LLMOD) [Watts-2014], MD with perturbation along low energy eigenvectors [Labute-2010], and inverse kinematics [Coutsias-2016]. It is important to note that **OMEGA** DG also works well for very flexible linear molecules which have such a large conformational space that torsion driving may not be able to adequately sample it.

OMEGA's method of conformational sampling of macrocycles is an adaptation of the distance geometry method of Spellmeyer et al. [Spellmeyer-1997]. In this method, a traditional embedding DG algorithm is replaced with a direct error function minimization of the random atomic coordinates, followed by force field refinement. Each initial Cartesian atomic coordinate *x* is assigned by choosing a random number *r* between *-1* and *1*, multiplied by a factor $f\sqrt{N}$ which determines the box of maximum extent for a molecule with *N* atoms:

$$x = f\sqrt{N}r \tag{5.1}$$

Alternatively, instead of randomly placing atoms in Cartesian space, the method allows for random placement of rigid fragments such as aromatic rings, nitro groups, and so on. After randomly placing molecules, atoms, or rigid fragments, an error function of the form:

$$F = \sum_{i,j}(d_{ij} - c_{i,j})^2 + \sum_k V_k \tag{5.2}$$

is optimized. In the above equation, the first sum runs over all pairs of atoms in the molecule, where d_{ij} are the interatomic distances and c_{ij} are elements of the constraint matrix; they are obtained from the MMFF94 force field parameters. When atoms *(i,j)* are bonded or are the first and last atoms in a bond angle, the upper and lower bounds are the same and are taken as the corresponding equilibrium force field distances. When atoms *(i,j)* are the first and last atoms of a torsion angle, the lower bound corresponds to the cis configuration and the upper bound to the trans configuration of those atoms. Finally, when atoms *(i,j)* are separated by more than three bonds, the lower bound is taken as the sum of the vdW radii and the upper bound as the sum of bond lengths which separate the pair. The second summation in equation (5.2) is over the tetrahedral constraints which result from:

1. planarity
2. chirality
3. cis-trans isomerism

Optimization of the error function (5.2) leads to a rough conformation. No hydrogen atoms except those which are bonded to chiral atoms are included in the error function minimization. Each rough conformation is checked for chirality correctness before refinement. If the rough conformation passes the chirality checks, it is refined against a forcefield,

MMFF94 ([Halgren-I-1996], [Halgren-II-1996], [Halgren-III-1996], [Halgren-IV-1996], [Halgren-V-1996]). Solvent forces can be included in the refinement step using a simple continuum solvation model (the Sheffield model [Grant-2007]). When the sequence of random placement followed by error function minimization and force field refinement is repeated for a large enough number of times for a given macrocycle, its conformational space is reasonably well covered.

Special measures are taken for zwitterionic molecules (containing both positively and negatively charged groups, such as CO_2^- and NH_3^+). In order to prevent possible Coulombic collapse in the absence of a stabilizing receptor, zwitterionic molecules are neutralized before performing the distance geometry calculation and the refined structures are recharged before writing them out. This approach meaningfully improves **OMEGA**'s ability to reproduce solid-state conformations of zwitterionic ligands.

This method of conformation sampling is completely general and can be used to generate conformations for any molecule, whether or not it contains a large ring. However the `macrocycle` mode in **OMEGA** has been specifically developed and parameterized to perform well on macrocycles, therefore its performance on linear and small ring molecules is worse than with `classic` **OMEGA**.

5.2 Filtering Theory

5.2.1 Introduction

Filtering attempts to eliminate inappropriate or undesirable compounds from a large set before beginning to use them in modeling studies. The goal is to remove all of the compounds that should not be suggested to a medicinal chemist as a potential hit. This exercise is obviously case dependent, depending on ease of the assay, intended target, needs of the user, and so on.

To match this need, the default filter encapsulates many of the standard filtering principles, such as removal of unstable, reactive, and toxic moieties. In addition, FILTER allows the customization of the filtering criteria to fit specific needs.

The criteria for passing or failing a given molecule fall into three categories.

- Physical properties
  - Molecular weight
  - Topological polar surface area (TPSA)
  - logP
  - Bioavailability
- Atomic and functional group content
  - Absolute and relative content of heteroatoms
  - Limits on a very wide variety of functional groups
- Molecular graph topology
  - Number and size of ring systems
  - Flexibility of the molecule
  - Size and shape of non-ring chains

All of the data generated in filtering molecules can be written to a tab-separated file for easy import into a spreadsheet. This functionality allows for combining the values dynamically for a variety of purposes, including, but not limited to, determining which filter values best fit each project's needs.

History

When OpenEye's work on filtering technology began in 2000, it was designed simply to remove compounds with reactive or otherwise undesirable functional groups. Over the years, the understanding of lead-like and drug-like compound selection has advanced. In addition, with the publication of Lipinski's "Rule of 5" [Lipinski-1997], more and more pharmacokinetic properties have appeared earlier in the virtual screening process.

In addition to providing basic functional group selection, the technology is a one-stop database preparation tool aimed at generating databases suitable for high-throughput virtual screening. The following are some components of this tool.

- Cheminformatics quality control
  - Valence-state validation
  - Aromaticity perception
  - Implicit hydrogen perception
  - Bond-order perception
- Database preparation
  - Setting pKa states
  - Applying normalizations (tautomers and dative or hypervalent states)
- Compound selection
  - Physical properties (see *above*)
  - Assay counter-indicators (aggregators and dyes)
  - Promiscuous actives [Baell-2010]
  - PK ([Martin-2005], [Veber-2002], [Egan-2000], [Lipinski-1997])

Finally, it should be pointed out that in the virtual screening world, time is of the essence. Algorithms for preliminary database preparation should not take large amounts of time. Because of this, all the calculations included are 2D or graph-based algorithms. While this does occasionally limit the technology, it allows for the delivery of a product that is appropriate for the task of virtual-screening database preparation.

A Detailed Examination of False Positives and False Negatives

Nearly every computational tool used in early drug discovery yields statistically predictive, rather than absolutely definitive results. In nearly every case, prudence demands that one consider the causes of false positives and false negatives and make an attempt to optimize the area under the receiver-operator curve (ROC) for the computational tool. However, there are well-known methods for improving statistical predictions of this nature that are independent of the absolute false-positive and false-negative rates. These methods include filtering the population to which a test will be applied. By applying a test to smaller populations that only contain molecules appropriate for the specific application at hand, the negative impact of the false-positive rate on the predictive results can be dramatically improved.

A familiar example from the medical world will serve to illustrate this principle. Assume we have a test for the presence of the new *foo virus* which has an exceptional ROC curve with false-positive and false-negative values (1/1,000 and 1/1,000 respectively). Let us assume that the *foo syndrome*, caused by the *foo virus*, affects 1 person in 20,000. If we gave this test to 100,000 people from the general population, we would expect 5 to actually have the *foo syndrome*. With this test, there is only a 0.05% percent chance that any of them would not be detected (that is, be a false negative). However, we would expect there to be 100 false-positive test results. Thus of the 105 total positive test results, only 4.8% would actually have the *foo syndrome* (positive predictive value = 4.8%).

Table 1: Confusion table for the **unfiltered** *foo virus* test (prevalence 1 in 20,000)

	Actual Positive	Actual Negative	
Predicted Positive	True Positives = 5	False Positives = 100	Positive Predictive Value = 4.8%
Predicted Negative	False Negatives = 0	True Negatives = 99,895	

Alternatively, we could start by using very simple screening before applying the test. We first eliminate people who do not have any risk factors for contracting the *foo virus*. Next, we may eliminate people whose blood is incompatible with the test for the *foo virus*. Further, we may want to eliminate people who acknowledge that they will refuse treatment for the *foo virus* even if we determine that they do have it. By these admittedly simple screens, we apply the test for the *foo virus* to a much smaller group with a decidedly higher prevalence of the virus. For instance, after the filtering, we may be left with a group of only 1,000 people who have a 1 in 200 chance of having the syndrome. Now, we still have the same five people who actually have the disease, but we only expect one false-positive test. Now there are six total positive tests, and 83% of them actually have the syndrome! This is reflected in a much more reasonable (83%) positive predictive value.

Table 2: Confusion table for the **filtered** *foo virus* test (prevalence 1 in 200)

	Actual Positive	Actual Negative	
Predicted Positive	True Positives = 5	False Positives = 1	Positive Predictive Value = 83%
Predicted Negative	False Negatives = 0	True Negatives = 994	

Returning to drug design, in a ligand-based design tool such as **ROCS**, we can posit that the receiver-operator curve may have a false positive rate as low as 1 in 10,000. For this exercise, assume no false negatives. When using that to identify 50 inhibitors from a database of 2.5 million available compounds, we would identify 300 potential inhibitors, and 5 out of every 6 of these would be a false positive (positive predictive value of 17%). However, running `filter` first would eliminate 65% of the 2.5 million compounds, leaving 875,000 compounds to run through **ROCS**. There will be approximately 88 false positives along with the 50 true positives increasing the positive predictive value more than two-fold with relatively little work.

Table 3: Confusion table for the **unfiltered** ROCS virtual screen

	Actual Positive	Actual Negative	
Predicted Positive	True Positives = 50	False Positives = 250	Positive Predictive Value = 17%
Predicted Negative	False Negatives = 0	True Negatives = 2,499,700	

Table 4: Confusion table for the **filtered** ROCS virtual screen

	Actual Positive	Actual Negative	
Predicted Positive	True Positives = 50	False Positives = 88	Positive Predictive Value = 36%
Predicted Negative	False Negatives = 0	True Negatives = 874,862	

Filtering Principles

The same principle of increasing the positive predictive value by removing obvious true negatives applies to screening for lead candidates, regardless of whether it is virtual screening or high-throughput screening (HTS). While both are reasonable screens, each can be plagued by very low positive predictive values (despite low false-positive rates), particularly when applied to all available compounds or large virtual libraries. Simple filtering techniques focus the set of compounds passed on to more computationally intensive screening methods.

The first approach is to filter based on *functional groups*. Generally speaking, there are toxic and reactive functional groups that should be removed from consideration, such as alkyl bromides or metals. There are also functional groups that are not strictly forbidden, but are undesirable in large quantities. For instance, parafluorobenzene or trifluoromethyl have specific purposes, but heavily fluorinated molecules can be eliminated.

Beyond simple functional group filtering, both simple and complex physical properties can be used to characterize the kinds of compounds to keep or to eliminate. These properties attempt to consider drug-like qualities, such as bioavailability, solubility, toxicity, and synthetic accessibility, even before the primary high-throughput or virtual screening, which are primarily geared toward detecting potency alone. The best known physical property filters is *Lipinski's "rule-of-five"*, which focuses on bioavailability [Lipinski-1997]. However, many other physical properties, such as *solubility*, *atomic content*, *ring structures*, and *surface area ratios* can also be considered. FILTER provides algorithms for calculating many of these properties and applying them with filters based on literature studies.

Finally, types of compounds that can be troublesome at later stages should be eliminated. For instance, Shoichet's *aggregating compounds* often produce false positives that can waste enormous resources if they are identified by virtual or HTS ([McGovern-2003]; [Seidler-2003]). Similarly, *dyes* can appear to be inhibitors by interfering with colorimetric or fluorometric assays or by nonspecific binding to the target protein.

Variations of Filters

Different types of filters are appropriate under different circumstances. Very early in a project, when little or no information on the structure-activity relationship (SAR) is available, very strict drug-like filters can be applied. This avoids spending chemistry resources to pursue difficult compounds that may not be modifiable for appropriate properties. However, when considering compounds to purchase for HTS, different filters can be applied. Oprea et al. pointed out that the best molecules for initial HTS are smaller and have less functionality than drugs but show some activity [Oprea-2000]. Therefore, strict lead-like filters can be applied to ensure that hits identified from HTS can be investigated for potential leads, especially those that may be larger or more highly functionalized. However, when SAR suggests that particular compounds or series may yield valuable information, filtering criteria can be loosened, because secondary screens can be effective in detecting useful compounds. Circling back to the medical analogy, this is a case where an improved primary screen with a dramatically improved false-positive rate (e.g., 1 in 100,000) can be applied to a larger population without detrimental effects on the positive-predictive value.

FILTER provides the following filters:

- **BlockBuster**: The BlockBuster filter is based on 141 best-selling, non-antibiotic prescription drugs. We designed the physical property portion of the filter so that it passes all of the compounds. The physical property values in this filter are quite good. However, the functional group filters in this filter may be too restrictive because 141 compounds will not span all acceptable functionality.
- Drug: The original drug filter is provided as well [Oprea-2000], but has been shown to be very restrictive. The BlockBuster filter was developed as an alternative.
- Fragment: The fragment filter is designed specifically to filter molecules with attachment points. This filter is used inside **CHOMP** to restrict the fragments generated for the **BROOD** database.
- Lead: The Lead filter corresponds to the Oprea lead-like filters that are useful for preparing HTS screening databases.
- PAINS: The PAINS filter is based on work by Baell which describes identification and filtering of promiscuous (nonspecific) actives across a number of screening types and targets [Baell-2010]. Unlike the other filter types,

the PAINS filter consists of only functional group filters; no physical property filters are included. In practice, combination of the PAINS filter with a user-defined set of physical property filters may be desirable.

Hint

We recommend the default BlockBuster filter for most purposes. If you have specific filtering needs or are unsatisfied with the results, please see the *filter file*.

If you modify the *filter file*, the depictions found in the *Functional Group Rules* section can be particularly helpful in determining which functional groups are indicated by each name in the file.

Accumulation of Rules

FILTER contains rules to judge the quality of molecules on various properties. Individually, each of these rules seems quite reasonable and even profitable. However, when each molecule is tested against hundreds of individual filters, the fraction of molecules that pass all the filters can be surprisingly small. Sometimes less than 50% of vendor databases pass the filters. If this is unacceptable, we recommend you examine the predicted aggregator, solubility, and Veber filters. In our experience, these are the most common failures. The best method of investigating failures is looking at the filter log.

The BlockBuster filter addresses these challenges. Sometimes the range needs to be expanded in order to allow more compounds to pass. For each value, its properties extend from the 2.5th percentile to the 97.5th percentile. The differences between the original BlockBuster filter and the adjusted filter are both in reasonable ranges. For instance, the full range of molecular weights for the BlockBuster filter spans 130 to 781, while the 2.5th percentile is 145 and the 97.5th percentile is 570. Surprisingly, when the reduced filter is used, only 75 of the 141 original molecules pass the filter! This demonstrates how slight changes to many filters can lead to a significant reduction in the number of compounds that pass all filters.

The opposite approach of allowing everything to pass can be equally challenging. For example, a filter was designed around the "small molecule drug" file available from the DrugBank website. In order to pass all of these molecules, including some that would no longer be acceptable, many of the individual filters must be set to unreasonable values, such as a molecular weight range from 30 to 1500 and a heteroatom count range from 1 to 60.

Hint

Because every project is different, we recommend that you inspect each *filter file*. The depictions found in the *Functional Group Rules* section can be particularly helpful in discerning what functional groups are desirable and undesirable.

5.2.2 Filter Preprocessing

Before the application of any *molecular property filters*, a preprocessing step may be used to alter the molecule significantly to fit the criteria needed for most modeling applications. This filter preprocessing step is applied to each molecule in a series of stages in the following order:

1. *Metal Removal*
2. *Salt Removal*
3. *Canonicalization*
4. *pKa Normalization*
5. *Normalization*
6. *Reagent Selection*

7. *Type Checking*
8. *MMFF94 Atom Type Checking*

Metal Removal

Metal removal is the first stage of element-based filtering. This stage removes specified metal complexes from the molecule. It will not reject a molecule with a metal complex. This allows the filter to treat atoms in the counterion portion of a molecule separately from the atoms in the primary portion of the molecular record.

For instance, organic molecules that are complexed with silver can be eliminated based on their metal chelate even though they themselves are acceptable, while at the same time eliminating a sulfate counterion from another molecule before it leads to elimination of the acceptable cationic molecule.

See also

Elemental Filters

Salt Removal

This step deletes all atoms that are not part of the largest connected component of a compound. This effectively eliminates all non-covalently bound portions of the compound.

Canonicalization

This step canonicalizes the atom and bond order of the portion of the molecule that remains after the previous removal steps. This is necessary to avoid different atom orderings producing slightly different normalizations in the following normalization steps.

pKa Normalization

pKa normalization uses a rule-based system to set the ionization state of input molecules. If pKa normalization is turned on, the molecule is set to its most energetically favorable ionization state for pH=7.4. The rule-based nature of this calculation allows it to be very fast. Furthermore, despite being rule-based, this approach takes into account many secondary charge interactions.

While more advanced levels of theory can be found for predicting ionization states, this method is very well suited to virtual-screening database preparation. However, this may not be appropriate for hit-to-lead or lead optimization.

Normalization

In addition to *pKa normalization*, FILTER allows any number of additional molecular normalizations. Since normalizations are usually specific to a particular company or site, FILTER provides the ability for users to input normalizations, such as the nitro tautomer state, but does not provide default implementations.

Reagent Selection

Reagent selection for small linear library synthesis or large combinatorial library synthesis is usually a necessary task. A user hoping to identify a set of acyl-halide reagents could specify a selection parameter to require each compound to have exactly one acyl-halide. In addition, the filter could be modified to exclude functional groups (such as primary amines) that may be acceptable for typical lead-like molecules, but are not acceptable for the specific reagent the user has in mind.

Therefore, the selection parameter is the reverse of a filtering parameter. The molecule *must include* the given substructure in order to pass the filter.

See also

The *select parameter* in filter files.

Type Checking

This checks the valence state and formal charge of the entire molecule. The check identifies molecules that are poorly specified, or represent nonsensical chemical states, often from corrupt input data. For example, an oxygen with eight hydrogens attached or a carbon with a +9 formal charge would be rejected.

MMFF94 Atom Type Checking

This checks that all atoms in the molecule have valid MMFF94 atom type assignments. The check identifies molecules that will fail downstream processing that depends on MMFF94 atom types (e.g., **Omega**).

5.2.3 Molecular Properties and Predictors

FILTER provides a range of properties to predictors to be used as molecular filters. Molecular properties are distinct physical properties that can be measured, including the following:

- *Structural and Chemical Features*
- *Functional Groups*

There have been several attempts to develop fast-approximate QSAR models for bioavailability. The first of these was *Lipinski's work* [Lipinski-1997], followed by work at Pharmacopia [Egan-2000], Abbott [Martin-2005], and GSK [Veber-2002]. The simplest and probably most trusted work includes studies of *LogP* and *PSA* used by Egan. Martin's Abbott Bioavailability Score (ABS) appears to be a refinement of the first generation models and is designed specifically to categorize a molecule's probability of having a bioavailability >10% in rats.

See also

The *Pharmacokinetic Predictors* section in the Filter Files chapter.

Structural and Chemical Features

A number of important structural and chemical features of molecules should be limited for virtual screening. The following simple measures are provided as filterable properties:

- Molecular weight
- Ring count
- Ring-system size
- Size of non-ring structures
- Length of unbranched chains
- Heteroatom fraction
- Halide fraction
- Formal charges
- Rotatable bonds

MolProp TK also includes slightly more complex algorithms for hydrogen bond donors and acceptors as well as chiral centers.

See also

The *Basic Properties* section in the Filter Files chapter.

Functional Groups

Functional group removal remains at the heart of the **FILTER** algorithm. Functional groups fall into several categories, including:

- Reactive or labile groups
- Undesirable groups
- Generally acceptable groups
- Protecting groups
- User-derived groups

Reasonable defaults are provided for each functional group in the above categories, with the filter files being designed as basic guides. They can be tailored for each user's particular needs. Examining the filter file can provide useful information. Developing a custom filter for a specific task is strongly encouraged.

See also

The filter files contain only functional group names. For clarification of the definitions, please refer to the *Functional Group Rules* section for complete descriptions.

Dyes

In the past, colored molecules interfered with many high-throughput assays. While assay technology continues to advance and this problem has decreased, most dyes are not commonly carried forward in lead development projects. Therefore, a pattern-based filter for dye molecules is included. While these patterns occasionally identify molecules that are considered acceptable by some users, in general, they identify molecules that the majority of chemists would rather not see at the top of their virtual screening hit lists.

LogP

The XLOGP algorithm ([Wang-R-1997]) is provided because its atom type contribution allows calculation of the XLogP contribution of any fragment in a molecule and allows minimal corrections in a simple additive form to calculate the LogP of any molecule made from combinations of fragments. Further, although the method contains many free parameters, its simple linear form allows for ready interpretation of the model, and most of the parameters in the model make rational sense.

Unfortunately, the original algorithm is difficult to implement as published. First, the internal hydrogen bond term was calculated using a single 3D conformation, which was found to be both arbitrary and unnecessary. This arbitrary 3D calculation has been replaced with a 2D approach to recognize common internal hydrogen bonds. In tests, the 2D method worked comparably to the published 3D algorithm. Next, the training set had a few subtle atom-type inconsistencies.

Both of these problems were corrected and refit to the original XLOGP training data. This implementation gives results that are quite similar to the original XLOGP algorithm, so it is called OEXLogP to distinguish it from the original method.

See also

- The *XLogP parameter* in the Filter Files chapter.

LogS

The work of Samuel Yalkowsky [Yalkowsky-1980] at the University of Arizona has resulted in what is now called the "generalized solvation equation." It states that the solubility of a compound can be broken into two steps, first the melting from the pure solid to pure liquid, and second, the phase transfer from pure liquid into water. For many small organic molecules, this second step is somewhat related to LogP.

Because of this relationship, we explored the use of the XLogP atom types in solubility prediction, with the goal of providing an approximate though robust and fast method for calculating solubility. We fit the XLogP atom types to a training set of nearly 1000 public solubility values, then derived a linear model for solubility. The model is extremely fast and can classify compounds as insoluble, poorly soluble, slightly soluble, moderately, soluble, or very soluble. However, it has difficulty in predicting the solubility of compounds with ionizable groups. Furthermore, it is not suitable for the PK predictions that come late in a project. Nevertheless, it can be utilized to eliminate compounds with severe solubility problems early in the virtual screening process.

See also

The *solubility parameter* in the Filter Files chapter.

Polar Surface Area

Topological polar surface area (TPSA) is based on the algorithm developed by Ertl et al [Ertl-2000]. In Ertl's publication, use of TPSA both with and without accounting for phosphorus and sulfur surface area is reported. However, evidence shows that in most PK applications, it is better to avoid counting the contributions of phosphorus and sulfur atoms toward the total TPSA for a molecule. This implementation of TPSA allows either inclusion or exclusion of phosphorus and sulfur surface area, with the default being to exclude it.

See also

Including phosphorus and sulfur surface area:

- The `PSA_USE_SandP` parameter in the *filter file*.

Warning

TPSA values are mildly sensitive to the protonation state of a molecule. If the pKaNorm parameter is false, the TPSA value is calculated using the input structure, and if pKaNorm is true, the TPSA is calculated using the pKa normalized molecular structure.

See also

- The *pKa normalization* section of the Filter Preprocessing chapter.

Lipinski and Hydrogen Bonds

The work of Lipinski [Lipinski-1997] introduced the application of simple filter-like rules to roughly predict late-stage PK properties, in particular oral bioavailability. Unfortunately, Lipinski's rule of five has come into the common vernacular to such a large degree that some of the specific details are often lost. There are two critical examples of this. First, Lipinski used "violation of two rules" to categorize compounds. In the subsequent analysis, a significant difference in the two populations of molecules was detected, however, little analysis of the importance of a single violation was done. However, many now consider a single violation to be bad and two violations to be worse. Second, Lipinski used well-codified and well-understood yet imprecise definitions of "hydrogen bond donors" and "hydrogen bond acceptors" in his classification model. While this makes the algorithm quite understandable and easy to implement, it sometimes causes confusion with those who prefer more refined definitions of hydrogen bond donors and acceptors.

To address the first problem, users are allowed to set the number of Lipinski failures required to reject a molecule. In keeping with the original publication, the default value is 2. To address the second problem, two kinds of hydrogen bond donors and acceptors are calculated. In the Lipinski calculation, the published definitions are used (donor count is the number of nitrogen or oxygen atoms with at least one hydrogen attached and acceptors are the number of nitrogens and oxygens). For calculation of the number of donors and acceptors in the molecule for the sake of chemical properties, a more complex algorithmic approach is taken. This approach identifies the donors and acceptors outlined in the work of Mills and Dean [Mills-Dean-1996] and also in the book by Jeffrey [Jeffrey-1997].

See also

The *Lipinski violations* and *hydrogen-bond acceptors* sections in the Filter Files chapter.

Aggregators

The Shoichet lab ([McGovern-2003]; [Seidler-2003]) has demonstrated the importance of small-molecule aggregation in medium and high-throughput assays. Because these small-molecule aggregates can sequester some proteins, they give the appearance of being active inhibitors. There are now several hundred published aggregators, in addition to a published QSAR model for predicting aggregation propensity. The user can eliminate any of the known aggregators and also eliminate compounds that are predicted to be aggregators using the QSAR model. The published QSAR model for predicting aggregators may be considered quite aggressive, as it occasionally identifies compounds that are known to be genuine small-molecule inhibitors of a specific protein. While in most cases this model can be useful, it is important to gain experience with the model and judge the performance for specific needs.

In some cases, aggregation properties are specific to specific experimental conditions, thus caution should be exercised in the interpretation of these predictions. Nevertheless, aggregation remains an important issue in HTS hit follow-up and, short of experimental validation, this flag may be the best available.

See also

The *aggregators parameter* in the Filter Files chapter.

PAINS

Pan-assay interference compounds (PAINS) filters are a set of functional group filters created to identify and filter promiscuous (nonspecific) actives across a number of screening types and targets [Baell-2010]. The original PAINS paper includes a set of functional group prefilters, created via two iterations of their work, and three sets of structure class filters, called filter sets A, B, and C, which were used to identify and remove promiscuous actives. The OpenEye PAINS filter set is a combination of all four filter sets (the functional group prefilter set plus the three structural class sets) into a single filter set.

There are two distinct use cases for the PAINS filters. The first is the typical prefiltering of structures prior to screening or compound selection to remove undesirable molecules. One can simply eliminate any molecules which fail the PAINS filtering.

The second use case, which may ultimately be more interesting, is to use the PAINS filters retrospectively on a set of data which has been filtered (and tested) using another methodology. In this scenario, the PAINS filters can be used to identify potentially problematic reactive functionality in otherwise reasonable, drug-like structures. By using the table output mode and identifying matches for the A, B, and C filter sets, one can identify fairly specific chemical functionalities that have been known to result in nonspecific activity. This can be useful information for either prioritizing further lead development or for modifying lead structures to mitigate adverse reactivity.

There are approximately 650 SMARTS rules which have been converted from the SLN query strings provided in the supplemental information of the original paper. For the functional group filters, the authors do not provide identifying names for each individual filter, so our PAINS rules are named based on the comments used to describe each functional group class, prefixed with the string `pains_fg_[classname]`. For the structural class filters, the authors provided unique RegID values for each filter. For those cases, our rules are named based on the PAINS group and RegID as `pains_[group]_[RegID]`.

Note that in certain cases, the SLN queries cannot be represented with a single SMARTS, so in those cases the rules have been split out into multiple SMARTS patterns and are named `pains_[group]_[RegID]1`, `pains_[group]_[RegID]2`, etc.

Hint

Unlike the other filter types, the PAINS filter consists only of functional group filters; no physical property filters are included. It may be desirable to combine the PAINS filter with a separate user-defined set of physical property filters. The *filter file* section describes the creation of additional filter rules.

CHAPTER

SIX

RELEASE HISTORY

6.1 OMEGA 7.0.0

Minor internal improvements have been made.

6.2 Related Toolkit Information

6.2.1 Omega TK 7.0.0

New Features

- The following changes have been made to extend Thompson sampling usage for torsion driving in OEOmega:
  - The Thompson sampling algorithm for torsion driving has been improved for robustness and speed.
  - The default value for SetUseThompson in OETorDriveOptions has been changed to `True`.
  - The SetUseThompson is honored even when SetUseGPU is set to `True`.
  - The preliminary API class `OEThompsonOptions` has been removed, as it is no longer relevant.
- The default energy window for torsion driving in Classic mode has been changed to a range-based value to better suit **OMEGA** with Thompson sampling. Similarly, the energy range for the Pose mode has also been adjusted.

Minor Bug Fixes

- OEMolBuilder now respects the MaxEnumConfs provided in the OEMolBuilderOptions even when an input molecule has more conformers than specified.
- Certain edge cases where the torsion driving algorithm would fail with a FailedTorDrive error message have been addressed.

6.2.2 MolProp TK 2.7.0

Minor Bug Fixes

- The SMARTS pattern of the *halo_amine* functional group has been fixed.

6.3 OMEGA 6.1.1

Minor internal improvements have been made.

6.3.1 Related Toolkit Information

Omega TK 6.1.1

Minor Bug Fixes

- The default value and the lower range for MaxTrials in OEThompsonOptions have been changed from `0` to `1`.

MolProp TK 2.6.7

Minor internal improvements have been made.

6.4 OMEGA 6.1.0

Minor internal improvements have been made.

6.4.1 Related Toolkit Information

Omega TK 6.1.0

New Features

- The new methods SetOverlayRMS and GetOverlayRMS have been added to OESliceEnsembleOptions, enabling conformer slicing with in-place RMSD calculation between conformers, with OEConfSlicer and OESliceEnsemble.

Minor Bug Fixes

- OEOmega now properly places carboxylic acids in cis configuration when hydrogen sampling is not performed.
- All conformers generated with OEMacrocycleOmega now properly set the dimensionality of the conformers to 3.

MolProp TK 2.6.6

Minor internal improvements have been made.

6.5 OMEGA 6.0.0

6.5.1 New Features

- Supported input and output file formats for **OMEGA** have been revised to improve clarity.

6.5.2 Minor Bug Fixes

- **OMEGA** now properly propagates `SD data` of molecules when running jobs using `MPI`.
- **OMEGA** no longer reports time used per molecule in the log output, but they are properly reported in the `CSV` report.

6.6 Omega TK 6.0.0

6.6.1 New features

- The following new methods have been added to OETorDriveOptions to enable use of Thompson sampling, instead of exhaustive sampling, of torsion angles during torsion driving. Use of Thompson sampling is especially

useful, being significantly faster without losing quality, in generating smaller ensembles as obtained with the FastROCS or ROCS modes.

  - GetUseThompson
  - SetUseThompson
  - GetThompsonOptions
  - SetThompsonOptions

- A new class, OEThompsonOptions, has been added that enables controlling Thompson sampling during torsion driving.
- Generation of a single conformer with OEConformerBuilder has been modified to use Thompson sampling.

6.6.2 Minor bug fixes

- User-defined values passed through SetEnergyWindow, SetMaxConfs, and SetRMSThreshold are now properly accounted for, even when the corresponding defaults are range-based.
- Molecule titles are retained when Build fails in OEOmega with `fails to build structure from CT`.
- A verbose log is provided for fragments that fail when Build fails in OEOmega with `fails to build structure from CT`.
- The OEOmegaIsGPUReady function has been safeguarded from a rare crash. The function has also been updated to check not only for device 0, but for devices from 0 to 100.

6.7 MolProp TK 2.6.5

Minor internal improvements have been made.

6.8 OMEGA 5.1.0

6.8.1 New Features

- **MakeFragLib** now generates outputs that are consistent with outputs generated from **OMEGA**. The following outputs are generated by default.
  - makefraglib_frags.oeb.gz: Output fragments library
  - makefraglib_log.txt: Output run log
  - makefraglib_param.txt: Parameter file for rerunning the job
- These new command line parameters have been introduced in **MakeFragLib**.
  - *-ewindowRing*: Energy window for ring fragment conformer generation
  - *-rmsRing*: RMS threshold for ring fragment conformer generation
  - *-startfactRing*: Multiplier for number of starts during ring fragment conformer generation
  - *-timeLimit*: Maximum time in seconds allowed for non-ring fragment generation
  - *-fragGen*: Maximum number of conformers to generate for non-ring fragments
  - *-fragKeep*: Maximum number of fragments to keep for non-ring fragments
  - *-timeLimitRing*: Maximum time in seconds allowed for ring fragment generation
  - *-fragGenRing*: Maximum number of conformers to generate for ring fragments

  - *-fragKeepRing*: Maximum number of fragments to keep for ring fragments

6.8.2 Major bug fixes

- Running **OMEGA** with the `-useGPU false` option on a GPU-configured machine previously caused a memory leak. This issue has now been resolved.
- **MakeFragLib** now generates multiple conformers of fragments as expected.
- **OMEGA** can now properly handle input `CIF` files.

6.9 Omega TK 5.1.0

6.9.1 New features

- A new **Flipper** option has been added to enumerate internally perceived relative stereocenters (e.g., 1,4-disubstituted cyclohexanes). See SetEnumRelativeAtomStereo.

6.9.2 Minor bug fixes

- Stereochemistry of input molecules is now respected more consistently in OEMacrocycleBuilder.
- For SetMacroCycOptions in OEFragBuilderOptions, defaults for SetFragGen, SetFragKeep, and SetStartFactor have been changed to more appropriate values of 50000, 1024, and 20, respectively.
- An issue with the order of loading rules to an existing torsion library with AddTorsionLibrary has been fixed.
- A duplicate entry from the `GUBA` torsion library has been removed.

6.10 MolProp TK 2.6.4

Minor internal improvements have been made.

6.11 OMEGA 5.0.0

6.11.1 New Features

- The default fragment library in **OMEGA** has been updated in `Linux` and `OSX` platforms. The new fragment library consists of over 580 K fragments, replacing the previous library of ~80 K fragments. **OMEGA** in Windows still uses the previous library as default, but the modified library, `omega_fragment_lib_2023.oeb.gz`, is available through the `-addfraglib` on the database downloads page.
- A new parameter, `-failed`, has been added to **OMEGA** to allow reporting the failed molecules into a separate file.

6.11.2 Major bug fixes

- Running **OMEGA** with `-fixfile` and `-mpi_np` now appropriately picks up the file for fixing part of molecule.
- Output `rpt.csv` file from **OMEGA** now contains SMILES strings for most molecules. The entry is still blank for extremely large molecules that are beyond 400 atoms.

6.11.3 Minor bug fixes

- Running **OMEGA** with one molecule in the `classic/pose/dense/rocs/fastrocs` modes and using `-progress percent` no longer causes the input molecule to be read twice.

6.12 Omega TK 5.0.0

6.12.1 New features

- The default fragment library for OEOmega and OEMolBuilder has been updated. The new fragment library consists of over 580 K fragments, replacing the previous library with ~80 K fragments.
- Three new force fields have been added and can be used with OEMolBuilder and OETorDriver in OEOmega:
  - SAGE
  - SAGE_NOESTAT
  - SAGE_SHEFF
- The behavior of OEFlipperOptions has changed. New flags have been added to OEFlipperOptions, facilitating more complete decoupling of individual flags and more selective enumeration of specific stereocenter classes and environments than was previously possible.
- A new option class, OEFlipperClassicOptions, has been added to allow a detailed comparison of the new flag behaviors with the previous OEFlipper enumeration behavior. This class will be deprecated in a future toolkit release.
- The internal algorithm for storage and usage of torsion rules in OETorLib has been modified to improve performance when working with a library containing a large set of rules.
- A new return code FailedCharges has been added to OEOmegaReturnCode to indicate failures due to partial charge assignment.

6.12.2 Minor bug fixes

- The order of torsion rules in the default OETorLib has been changed to ensure that a specific rule is not ignored due to the presence of a more generic rule.

6.13 MolProp TK 2.6.3

Minor internal improvements have been made.

6.14 OMEGA 4.2.2

6.14.1 Minor bug fixes

- Defaults for `-warts` and `-flipper_warts` have been changed to `False` in **OMEGA**.
- **OMEGA** now fails robustly when incorrectly attempting to use `-fixsmarts` without providing `-fixfile` or turning `-fromCT` to `false`.

6.15 Omega TK 4.2.2

6.15.1 Minor bug fixes

- The OEOmega method Build will now fail when `MCS matching` is requested but a corresponding `FixMol` has not been set.

6.16 MolProp TK 2.6.2

Minor internal improvements have been made.

6.17 OMEGA 4.2.1

Fall 2022

6.17.1 New features

- A new command line parameter, *-fixmcs*, has been added to provide enhanced support to fix part of a molecule during conformer generation based on maximum common substructures (MCS).

6.17.2 Minor bug fixes

- *OMEGA* now generates report files with appropriate content.
- Log files from *OMEGA* now contain newly generated titles of molecules for untitled molecules.

Omega TK 4.2.1

6.17.3 New features

- The ability to fix a part of a molecule during conformer generation based on maximum common substructure (MCS) has been added. The following new methods have been added to the OEConfFixOptions to enable this functionality:
  - SetFixMCS
  - GetFixMCS
  - SetMCSFunc
  - GetMCSFunc
  - SetMCSType
  - GetMCSType
  - SetMCSMinAtoms
  - GetMCSMinAtoms
- New methods SetIgnoreStereo and GetIgnoreStereo have been added to OEMolBuilderOptions to allow continuation of conformer generation when specified stereo signatures in a molecule could not be honored.
- New methods SetMaxEnumConfs and GetMaxEnumConfs have been added to OEMolBuilderOptions to allow specifying the maximum number of conformers to be generated from ring enumeration, nitrogen enumeration, and/or hydrogen sampling.
- Verbose mode logging now reports torsion angles sampled, along with the rules, during torsion driving.

6.17.4 Minor bug fixes

- Conformer generation with OEOmega.Build or OEMolBuilder.Build now fails correctly when specified stereo signatures on a molecule cannot be honored. The existing behavior, to carry on even if stereo signatures could not be honored, can be obtained by setting IgnoreStereo to `true`.
- Ring fragment building during OEOmega.Build, OEFragBuilder.Build, OEMolBuilder.Build, or GenerateMissingFrags now properly honors the SetFragKeep parameter regarding the maximum number of generated fragment conformers to be kept.

MolProp TK 2.6.1

6.17.5 New features

- A new free function, OECheckXLogXType, that returns `true` if an atom has a valid XLogPType has been added.
- A new free function, OECheckXLogXTypes, that returns `true` if all atoms excluding hydrogen in a molecule have valid XLogP types has been added.

6.17.6 Minor bug fixes

- An issue with filter output failure reporting, where failure due to the rules *ALLOWED_ELEMENTS* and *ELIMINATE_METALS* was not included in the output table, has been fixed.

6.18 OMEGA 4.2.0

Spring 2022

6.18.1 New features

- **OMEGA** torsion driving (that is, in the `classic`, `rocs`, `fastrocs`, `dense`, and `pose` modes) now generates outputs that are consistent with outputs generated in the `macrocycle` mode. The following outputs are generated by default:
  - oeomega_mode_conformers.oeb.gz: Output molecule conformers
  - oeomega_mode_log.txt: Output run log
  - oeomega_mode_param.txt: Parameters file for re-running the job
  - oeomega_mode_rpt.csv: Output Report
- **OMEGA** now supports `cxsmiles` file format with enhanced stereo information as input.
- A new command line parameter, `-flipper_enhstereo`, has been added to provide enhanced stereo support for stereo enumeration when `-flipper` is set to `true`.
- **OMEGA** no longer supports writing outputs in non-3D file formats (e.g. `smi`).
- **OMEGA** now provides a detailed description of available modes when an invalid mode is presented in the command line.
- **OMEGA** torsion driving (i.e., in the `classic`, `rocs`, `fastrocs`, `dense`, and `pose` modes) now reports correct defaults for parameters in the command line help.

6.18.2 Minor bug fixes

- `oeomega classic --help` now returns a list of `simple` command line parameters. A list of all parameters can be viewed by typing `oeomega classic --help all`

- Inputs for **OMEGA** command line range parameters (i.e., `-erange`, `-maxConfRange`, and `-rmsrange`) have been simplified and now accept list-like parameters (e.g., `-maxConfRange 10 20 30` is a valid input).
- An issue has been fixed to enable **OMEGA** to generate an appropriate set of conformers when run with `-fixfile` and a `-maxmatch` value greater than 1.
- An issue has been fixed related to **GPU-OMEGA** occasionally writing to standard out stream.
- An issue has been fixed to enable **Flipper** to support piped input.

6.19 Omega TK 4.2.0

6.19.1 New features

- Support for consideration of MDL enhanced stereo collection information during isomer generation has been added. The OEFlipperOptions class adds a new `SetEnhancedStereo` option to control consideration of explicit enhanced stereo groups defined by OEGroupBase information that may have been read from `V3000` the OEFormat values indicating `SDF` structures or `CXSMILES` files.
- A new method, SetFragBuilderMode, has been added that allows setting fragment building mode in the corresponding OEFragBuilderOptions.

6.19.2 Minor bug fixes

- Internal generic data tags applied to the `flip-ee` used for performance during the flipping activity are now stripped from the structures returned by OEFlipper as this information is no longer relevant.

6.20 MolProp TK 2.6.0

6.20.1 New features

- The ability to filter based on aromatic ring count and fraction of CSP3 carbons has been added.

6.21 OMEGA 4.1.2

Fall 2021

- Minor internal improvements have been made.

6.22 Omega TK 4.1.2

Fall 2021

6.22.1 Minor bug fixes

- An issue with overwriting partial charges of atoms in input molecules has been fixed.
- The Generating Stereoisomers example has been modified to use OEFlipperOptions.
- An issue running OEOmega when both SetUseGPU and SetSampleHydrogens are set to *True* was fixed by disabling SetUseGPU in these cases.

6.23 MolProp TK 2.5.7

Fall 2021

6.23.1 New Features

- A method, GetMessage, was added to the OEFilter class to be able to retrieve information on whether the filter passed a molecule, or why it failed a specific molecule.
- A method, SetErrorLevel, was added to the OEFilter class to add explicit control of warnings output by the filter class and the filter application.
- A BlockBusterPAINS filter was added in the OEFilterType namespace, which combines filtering criteria for the BlockBuster and PAINS filters .
- Excess warning output that was causing large log files is now suppressed by default, and can be switched on using the new SetErrorLevel method. See OEFilterType.

6.24 OMEGA 4.1.1

July 2021

6.24.1 Minor Bug Fixes

- `Flipper` now has an improved default output file name.

6.25 Omega TK 4.1.1

July 2021

6.25.1 New features

- Function OEOmegaSystemHasGPU has been added to test for availability of a GPU.

6.25.2 Major bug fixes

- A bug that caused a segmentation fault when zero was passed to the SetMaxConfs method has been fixed.
- And issue that prevented oeomega from using the GPU in child threads when OEOmegaIsGPUReady was called from the main thread has been fixed.

6.25.3 Minor bug fixes

- An issue with stereo specificity of the imino group (=NH) not being retained in a macrocycle has been fixed.
- An issue with loss of relative stereo marked atoms in macrocycle mode has been fixed.

6.26 MolProp TK 2.5.6

July 2021

6.26.1 Minor bug fixes

- An issue with OEGetRotatableBondCount counting carbamates and ureas as rotatable has been fixed.

6.27 OMEGA 4.1.0.0

Fall 2020

6.27.1 New Features

- The defaults for the `-warts` and `-flipper_warts` in **OMEGA** have been changed from `false` to `true`.
- The Boolean flag `-mmff` has been added to the **Filter** application to allow filtering of atoms that cannot be atom-typed with `MMFF`. It is `false` by default.
- Six new flags have been added to **Flipper** to allow more control when selecting stereo centers as well as improving usability of the program:
  - `-enumEZ`
  - `-enumRS`
  - `-enumBridgehead`
  - `-molNames`
  - `-prefix`
  - `-progress`
- The built-in default filter for the **Filter** application is now in sync with the provided `filter_frag.txt` file.
- **GPU-OMEGA** now requires a minimum NVIDIA Driver version of 450.x and a minimum GPU compute capability of 3.5. See NVIDIA Drivers and Supported GPUs for more details.
- Conformer generation using `dense` mode will now default to running on the CPU instead of the GPU. The other torsion driving sampling modes (`classic`, `rocs`, `fastrocs` and `pose`) will still run on a GPU by default if one is available on a Linux platform. For details on how to run with `dense` mode sampling on a GPU see the *Usage* section of the **GPU-OMEGA** docs.

6.27.2 Major Bug Fixes

- **OMEGA** in the `macrocycle` mode now works with pipes.
- A memory leak in **GPU-OMEGA** has been fixed.

6.27.3 Documentation Changes

- Documentation for the `-erange`, `-maxConfRange`, and `-rmsrange` for **OMEGA** has been improved for clarification.

6.28 Omega TK 4.1.0

Fall 2020

6.28.1 New features

- **GPU-Omega** now requires a minimum NVIDIA Driver version of 450.x and a minimum GPU compute capability of 3.5. See NVIDIA Drivers and Supported GPUs for more details.
- Conformer generation using `dense` mode now defaults to running on the CPU instead of the GPU. The other torsion-driving sampling modes (`sparse`, `rocs`, `fastrocs`, and `pose`) will still run on a GPU by default if one is available on a Linux platform. For details on how to run with `dense` mode sampling on a GPU, see the Usage section of the **GPU-Omega** docs.
- A new class, OEFlipperOptions, has been added that defines parameters for specifying stereo centers and other options.
- OEFlipper has been updated to take a molecule and an OEFlipperOptions value. The old version of OEFlipper that takes five parameters has been deprecated.
- A new function, OEFlipperStereoCenters, has been added that returns a number of undefined stereo centers in a molecule as interpreted by OEFlipper.

6.28.2 Major bug fixes

- A memory leak in **GPU-Omega** has been fixed.
- An issue with generating conformers on the CPU using child processes with the same OEOmega object has been fixed.
- **GPU-Omega** no longer throws an error when attempting to generate conformers with the same OEOmega object in child processes. Instead, it falls back to the CPU. To use **GPU-Omega** with child processes, each child must construct a new OEOmega object.

6.28.3 Minor bug fixes

- OEConfGen now clears **all** fragment libraries. Previous releases did not clear the internal shared fragment library (a shared memory library of fragments accrued while generating conformers with the same OEOmega object).

6.28.4 Documentation changes

- OEOmegaReturnCode has been updated to reflect the current possible values.

6.29 MolProp TK 2.5.5

Fall 2020

- Minor internal improvements have been made.

6.30 OMEGA 4.0.0.6

- The **Filter** application no longer counts invertible-nitrogens as chiral centers.
- **OMEGA** no longer enumerates specified nitrogen when `-enumNitrogen` is `unspecified`.

6.31 OMEGA 4.0.0

Spring 2020

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

6.31.1 New Features

- **GPU-Omega**, a GPU-accelerated implementation of the torsion driving algorithm, is now available. **GPU-Omega** can make use of any suitable Nvidia GPU on supported Linux platforms by default for classic mode runs. Users are advised to read the *GPU-Omega* section of the documentation to ensure that their systems meet all the prerequisites and that the availability for the different sampling modes is understood. **GPU-Omega** will be disabled and fall back to the CPU on systems not meeting the requirements.
- **OMEGA** in the `macrocycle` mode has been rewritten to take advantage of the newly extended corresponding functionality in the 2020.0 release of **Omega TK**. With this update, some of the previous application-specific functionality, including duplicate removal and overlaying of the generated conformers, are now directly taken care of by the underlying toolkit functionality. From a performance point of view, **OMEGA** `macrocycle` is now more efficient for finding new conformers during conformer generation.
- The `-dielectric_constant` is no longer available as an option for torsion driving based **OMEGA** modes. This flag has been removed since it had little effect on conformers generated using torsion driving. The `-dielectric_constant` flag is still available as an option for **OMEGA** in `macrocycle` mode.

6.32 OMEGA 3.1.2

Nov 2019

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

6.32.1 Minor bug fixes

- An issue that caused *MakeFraglib* to not properly handle fullerene-like molecules has been fixed.
- *MakeFraglib* no longer hangs when `-progress` is set to `percent`.
- An occasional segmentation fault in *MakeFraglib* to fail to generate conformers for a fragment due to lack of atom typing, has been fixed.
- A memory issue that sometimes caused *MakeFraglib* to take too much memory has been fixed.
- **FILTER** now properly handles filter files with singular MIN or MAX values.

6.33 OMEGA 3.1.1

May 2019

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

6.33.1 New Features

Users can now use `oez` as the output format for **OMEGA**.

6.33.2 Major bug fixes

- The default for `-rotorOffsetCompress` is set to `true` for all modes.
- An issue that caused **OMEGA** to generate conformers with an occasional Coulombic collapse in the `macrocycle mode` for conformations of zwitterionic molecules has been fixed.

6.33.3 Minor bug fixes

- The default for `-enumNitrogen` has been changed from `true` to `unspecified` so that any user-specified stereo information is not lost by default.
- The flag `-skip` has been removed from *MakeFraglib*.
- **OMEGA** calculations no longer repeat an error message.

6.34 OMEGA 3.1.0

November 2018

- This version of **OMEGA** has been built using **OEToolkits 2018.Oct**. The previous version was built using **OEToolkits 2017.Feb**.

6.34.1 New features

- A new `fastrocs` mode has been added to generate conformers for use in **FastROCS**.
- All available options in **OMEGA** `classic` mode are now also available in the `pose`, `rocs`, `fastrocs`, and `dense` modes.
- Four new options, `mmff94smod`, `mmff94smod_NoEstat`, `mmff94smod_Trunc`, and `mmff94smod_Sheff`, have been added to the `-buildff` flag of *MakeFraglib*. `mmff94smod` modifies the torsion interaction parameters of `mmff94s` to produce the desired equatorial conformers for monosubstituted cyclohexanes. `mmff94smod_NoEstat` includes all `mmff94smod` terms except Coulomb interactions. `mmff94smod_Trunc` excludes both Coulomb interactions and the attractive part of Van der Waals interactions. `mmff94smod_Sheff` includes all `mmff94smod terms` along with the Sheffield solvation model for calculation of energies.

6.34.2 Major bug fixes

- Except for `macrocycle`, all **OMEGA** modes are now driven by the same executable. Previously, each mode had its own executable.

6.34.3 Minor bug fixes

- An issue that caused the `-maxtime` option to be not honored during torsion search has been fixed.
- An issue that caused the `-strictatomtyping` option to fail to return `False` has been fixed.
- An issue related to how the number of newly generated conformers per cycle is calculated for the `macrocycle` mode has been fixed.

6.35 OMEGA 3.0.1

(August 2018)

6.35.1 Major bug fixes

- A bug that caused memory leaks in **OMEGA** `classic`, `dense`, `pose`, and `rocs` modes, has been fixed. Previously, a substantial memory leak was experienced when running **OMEGA** on a large database.
- **OMEGA** `macrocycle` no longer uses excessive memory for molecules with terminal heavy atoms.

6.35.2 Minor bug fixes

- **OMEGA** `classic` no longer omits conformer titles when `-includeInput` is set to true. Previously, the resulting conformers had no titles.

6.35.3 Other changes

- Default values of `-max_iterations` and `-maxconfs` of **OMEGA** `macrocycle` have been changed to 2000 and 400 respectively.

6.36 OMEGA 3.0.0

(February 2018)

6.36.1 New features

- A method for sampling macrocyclic conformational space (see `macrocycle` mode below) has been added to **OMEGA**. This method uses a distance geometry approach to generate conformer ensembles for macrocyclic molecules. It can also be applied to linear and small ring-containing molecules but is substantially slower than **OMEGA** `classic`.
- **OMEGA** now includes five modes:
  - `classic`: the original customizable **OMEGA** interface
  - `macrocycle`: the conformer generator for macrocycles
  - `rocs`: the optimal conformer generator for **ROCS**
  - `pose`: the optimal conformer generator for pose prediction or docking
  - `dense`: the optimal conformer generator for **FREEFORM**

  The classic mode is the original customizable **OMEGA** interface and the `macrocycle` mode uses an adaptation of the distance geometry method of Spellmeyer et al. [Spellmeyer-1997]. All other modes use fixed optimal parameters of the original **OMEGA** interface for various downstream applications:

```
rocs:  -maxconfs = 50
pose:  -maxConfRange = "200,800", -rangeIncrement = 8
dense: -strictstereo = false, -ewindow = 15, -maxtime = 3600, -rms = 0.3,
       -maxrot = 20, -maxconfs = 20000
```

- A new command, `oeomega`, has been added to invoke these modes. In addition, the original `omega2` command has been retained for backward compatibility. In all modes except `macrocycle`, **OMEGA** automatically detects macrocycles and sends them to the fail file. The following are examples of `oeomega` commands to run these modes:

```
oeomega classic -in drugs.smi -out drugs_classic.oeb.gz
oeomega macrocycle -in drugs.smi -out drugs_macrocycle.oeb.gz
oeomega rocs -in drugs.smi -out drugs_rocs.oeb.gz
oeomega pose -in drugs.smi -out drugs_pose.oeb.gz
oeomega dense -in drugs.smi -out drugs_dense.oeb.gz
```

- There are now two built-in torsion libraries in **OMEGA** in `classic` mode: the original PDB-derived torsion library and a new CSD-derived library, "Guba," based on work from Roche using the frequency of particular angles found in the CSD for common torsion types [Guba-2016]. Introduced in the 2017.Oct Toolkits Release, this torsion library contains more than 500 new entries and complements the existing library, which is primarily

based on an analysis of structures found in the Protein Data Bank (PDB). A new flag, `-torlibtype`, has been added to the **OMEGA** `classic` mode that allows users to specify which of the two libraries should be used.

- Two new options, `macrocycle` and `nonmacrocycle`, have been added to the `-filter` flag of **FILTER**. `macrocycle` is used to remove non-macrocycle molecules in the input file; `nonmacrocycle` is used to remove macrocycle molecules.
- Four new options, `mmff94smod`, `mmff94smod_NoEstat`, `mmff94smod_Trunc`, and `mmff94smod_Sheff`, have been added to both the `-buildff` and `-searchff` flags of **OMEGA** `classic`. `mmff94smod` modifies the torsion interaction parameters of `mmff94s` to produce the desired equatorial conformers for monosubstituted cyclohexanes. `mmff94smod_NoEstat` includes all `mmff94smod` terms except Coulomb interactions. `mmff94smod_Trunc` excludes both Coulomb interactions and the attractive part of Van der Waals interactions. `mmff94smod_Sheff` includes all `mmff94smod terms`; the Sheffield solvation model is used during the calculation of energies for the conformers generated. `mmff94smod_NoEstat` is now the default for both `-buildff` and `-searchff`.
- A Pan Assay Interference Compounds (`PAINS`) filter has been added to the `-filter` flag of **FILTER** by adapting the supplementary material from the original [Baell-2010] paper. The `PAINS` filter can be used as a prefilter set for screening or, more interestingly, as a post-processing filter to identify possible protein-reactive or other assay-confounding functionalities within hits. Any hits so identified will require more care to determine if the measured activity is valid.

6.36.2 Major bug fixes

- `GZ` files are now read correctly when used with `-fixfile`. Previously, `-fixfile` would not be set and the command would proceed as a default **OMEGA** `classic` run.
- Both **FLIPPER** and **OMEGA** `classic` no longer flip 3D molecules if stereo is defined when using `-flipper false`.

6.36.3 Minor bug fixes

- `-maxConfRange` is now limited to 1 million to avoid out-of-memory crashes.
- **FLIPPER** now writes out to SMI files. Previously, stereo information was not supported in SMI format.
- **OMEGA** `classic` now fails if an empty torsion library is set to `-settorlib`.
- **OMEGA** `classic` no longer removes zeros at the end of a prefix. For example, the prefix `XXX.00100` previously would be expressed as `XXX.001`.
- For **FILTER**, `-SF5` groups no longer fail due to the `-typecheck` check. They may still fail for other reasons.

6.36.4 Platform support

- Support for 64 bit Windows has been added.
- 32-bit Windows is no longer supported.
- Support for Ubuntu 14 and 16 has been added.
- Ubuntu 12 is no longer supported.
- Support for RedHat v7 has been added.
- RedHat v5 is no longer supported.
- Support for macOS 10.11, Sierra (10.12), and High Sierra (10.13) has been added.

6.37 OMEGA 2.5.1

(May 2013)

6.37.1 New features

- **FILTER** is now completely bundled into the **OMEGA** package. The manuals have been consolidated into this single manual and **FILTER** has taken on the **OMEGA** version number.
- An option has been added to allow hydrogen atoms in -OH, -SH, and amines to take part in conformational sampling. This new option can be enabled via the *-sampleHydrogens* parameter. By default, hydrogen atoms are not sampled.
- A *-warts* option has been added to *flipper*.

6.37.2 Major bug fixes

- Now using *-fixsmarts* without *-fixfile* will rematch for every input structure. Previously, this would only match the first input structure and reuse that match for the rest of the calculation. Using both *-fixsmarts* and *-fixfile* will continue to match against the fixfile and use that match for the entire calculation.
- Partial ring matching with *-fixsmarts* or *-fixfile* could cause a crash. Now only full ring systems may be specified and any attempt at matching a partial ring system will fail gracefully.
- Planar molecules could have problems sprouting hydrogens when using *-fromCT false* because **OMEGA** was treating them as 2D. Now **OMEGA** will assume that planar molecules are properly defined 3D coordinates in this situation.
- Stereochemistry at the attachment points of rings with four or less atoms could be lost. Now stereochemistry is saved for rings of all sizes.
- A bug has been fixed where exceptionally long torsion rules could get cutoff early.
- A bug has been fixed where older TK licenses in the license file could cause the OMEGA application to fail. [Case #13466]

6.37.3 Minor bug fixes

- Only 3D coordinates were being cleared in **FLIPPER**. Now 2D coordinates are cleared as well.
- Fragment generation during an **OMEGA** calculation with *strictfrags* parameter enabled could yield slightly different fragments than the *MakeFraglib* app. This has been corrected for consistency.
- Tags for rotor compression are no longer attached to the output molecules when rotoroffsetcompress is false. It was discovered that these tags confuse VIDA when they are attached but not used.
- The torsion library has been updated with better angles for a biphenyl torsion. The entry for 90 degrees has been removed and +/-30 degrees has been added.
- FILTER has been fixed to no longer have an atom order dependence when setting a pH model.

6.37.4 Other changes

- PVM (parallel virtual machine) is no longer supported. OpenMPI version 1.6 is supported on all platforms. The *-mpi_np* and *-mpi_hostfile* flags are now used to run **OMEGA** and *makefraglib* in MPI mode. These new flags replace the *oempirun* script.
- This will be the last release to support SuSe 10.

6.38 OMEGA 2.4.6

(February 2012)

6.38.1 New features

- Now uses an extension of the MMFF94 force field for tricoordinate boron compounds. Most compounds containing B-X bonds where X=C,N,O,S, and H are covered with the following exceptions: X=N(imine),N(sulfonamide), N(pyridinium) and N(quaternary). Also not supported are compounds in which boron is bonded to X=F,Cl,Br,I,B and Si, or makes a bond angle BYX. Compounds in which boron is a part of four-membered rings of B1CCC1 type are also not available in the current parameterization because their existence is questionable: Ab initio calculations at the MP2/6-31G** level failed to identify stable structures for them (highly polar structures in which boron is four-coordinated are formed).

6.38.2 Bug fixes

- Fixed a bug where **OMEGA** wasn't checking if 3D construction of fragment generation was successful.
- Fixed a bug where output from flipper could still fail in **OMEGA** for missing stereo chemistry. Now the rules for required stereo chemistry are consistent.
- Fixed a bug where setting stereo chemistry in too small of a ring could cause a crash. Now only rings greater than 7 members and only one double bond will be set with flipper.
- Fixed a bug where **OMEGA** wasn't checking if user generated fragment libraries were in OEB format. Fragment libraries must be in OEB format.
- Fixed bugs related to nonsense values for command line parameters. User-supplied parameters must be a legal value. The '–help <parameter>' command will print out the legal values.
- Fixed a bug where SD tagged data wasn't passed along to output molecules.

6.39 OMEGA 2.4.3

(August 2010)

6.39.1 Bug fixes

- A bug was fixed where slave nodes required a license. This has been fixed and now only the master requires a license.
- The -warts flag did not work when writing to formats other than .oeb or .oeb.gz. This has been corrected in version 2.4.3.

6.40 OMEGA 2.4.1

(June 2010)

6.40.1 New features

- FILTER is now bundled with the **OMEGA** distribution and will run with a valid **OMEGA** license.
- **OMEGA** and *MakeFraglib* may now be run with MPI using the *oempirun* script. (Not available on all platforms)
- The default for **OMEGA** has been changed to require all stereocenters to be specified. Molecules with unspecified stereo will fail, except for unspecified invertible nitrogens which are allowed and will not cause the molecule

to fail. The previous behavior was for **OMEGA** to choose a random stereoisomer. This default can be changed back to the previous behavior by setting *-strictstereo* to `false`.

- By default, **FLIPPER** will no longer flip invertible nitrogens. The default path is for **FLIPPER** to leave them unspecified and allow **OMEGA** to enumerate them. Flipping invertible nitrogens in both **FLIPPER** and then **OMEGA** will lead to duplication. These options can be changed with the *-enumNitrogen* flag available in both applications.
- The default for **OMEGA** has been changed to no longer use 'close enough' atom typing. If a proper MMFF atom type cannot be found for an atom then the molecule will fail. Previous versions would use a different atom type of the same element if it was available. The previous behavior for atom type substitution can be enabled by setting *-strictatomtyping* to `false`.
- **OMEGA** now has the option to require on-the-fly generation of fragments to be the same as the *MakeFraglib* utility by setting *-strictfrags* to `true`. This fragment generation is more rigorous but also more time consuming.
- The convenience flag *-strict* has been added to turn on or off all of the strict options. These include *-strictstereo*, *-strictatomtyping* and *-strictfrags*.
- On-screen progress has been enabled for OMEGA and *MakeFraglib* using the *-progress* flag. The options are *none*, *dots*, *log*, and *percent*.
- The *-maxconfgen* and *-maxpoolsize* flags have been removed. These options are set internally and adjusted automatically subject to the user defined variable of *-maxconfs*. Additionally, hard limits have been removed from these variables and they are now only limited by available memory.
- Improper formatting of the *-erange*, *-maxConfRange*, and *-rmsrange* parameters will cause OMEGA to exit immediately. Previously the application would throw a warning but continue the calculation without using the parameters.
- The flag *-fixsmarts* has been created to allow a smarts pattern to be used to fix a portion of the molecule. This requires 3D coordinates to be available from either a 3D input file with *-fromCT* set to `false` or from a molecule set using *-fixfile*.

6.40.2 Bug fixes

- A bug was fixed in *MakeFraglib* where the setting *-fromCT false* could give nonsense fragments.
- Crashes have been fixed that could occur when the *-fixfile* could not find a valid match or did not have enough atoms to match against in a ring system.
- Program no longer crashes if there is not enough memory for duplicate removal. If there is not enough memory available for allocation then the application will exit immediately.

6.41 OMEGA 2.4.0

(November 2009) (Toolkit only)

This release was for the toolkit only. For more details please refer to the OmegaTK documentation.

6.42 OMEGA 2.3.3

(March 2009) (Toolkit only)

This release was for the toolkit only. For more details please refer to the OmegaTK documentation.

6.43 OMEGA 2.3.2

6.43.1 Bug fixes

- Fixed a bug where molecules containing boron or selenium could cause a crash. These molecules now are written to the *'.fail'* file due to missing force field parameters and the program will proceed to the next molecule.

6.44 OMEGA 2.3.1

6.44.1 New features

- The **OMEGA** application now includes a utility named oeb2sdconf that allows **OMEGA** generated conformers to be used with the MOE program. Please contact Chemical Computing Group support if you require assistance with this utility.

6.44.2 Bug fixes

- Fixed a bug where the omega2 PVM features would not work on some platforms.
- Fixed a bug where the omega2 script would not function properly with filenames that had spaces in them.

6.45 OMEGA 2.3.0

6.45.1 New features

- The distribution and installation of **OMEGA** has been modified. The Windows distribution is now a standard .exe installer, and the macOS distribution is a dmg containing a standard pkg installer. The executables are now scripts that chose the correct version of the program at runtime. Please see the Installation and Platform Notes for details.
- The defaults for MaxConfs and RMSThreshold are changed in this version. The default for MaxConfs is reduced from 400 to 200, while the value for RMSThreshold goes from 0.8 down to 0.5. These changes are based on maintaining good reproduction of the crystal structure test set, virtual screening tests with **ROCS** and **FRED**, and pose prediction tests with **FRED**.
- The method `OEConfGen::OEOmega::ClearFixFile` is added to the **OMEGA** library. This allows an `OEConfGen::OEOmega` instance to be reused for an unrestrained search after it has previously been used for a fixed search.
- Untitled molecules are given unique titles of omega_1, omega_2, and so on. Titles allow molecules to be located in the log files.

6.45.2 Bug fixes

- Fixed a major bug where molecules that have missing torsion rules caused **OMEGA** to crash. Molecules with missing torsion rules are now sent to the *omega2.fail* file and are not processed. Missing torsion rules are typically caused by highly unusual molecules for which OEChem cannot properly assign valence to all atoms.
- Fixed a bug in duplicate conformer removal that caused some conformers that should have been removed to pass through as unique. This bug also caused occurrences where a reduction in maxconfs led to an increase in the number of conformers returned.
- Fixed a bug that caused SD data on single conformer molecules to be removed.
- Fixed a bug in the *flipper* library that left 3D coordinates intact, which caused stereo chemistry to be ambiguous. The *flipper* library now replaces 3D coordinates with zeros.

- Fixed a bug where the `SyM` tag from previous **OMEGA** calculations caused problems when those molecules where passed back in. Also, the `O2MolId` tag is now removed from molecules before they are sent to the output.
- When the number of stereo centers exceeds the number of -maxcenters M, *flipper* now generates 2^M random non-repeating flips. Previous versions generated 2^M random flips, leading to molecules with the same stereochemistry being returned multiple times.

6.46 OMEGA 2.2.2

6.46.1 Bug fixes

- Fixed a major bug that would cause highly-symmetric, highly-fluorinated compounds to crash the program or hang a slave in PVM mode.
- Fixed a bug in atom typing when building fragments from scratch that, in some rare cases, could result in differing fragment geometries depending on the order of input of the molecules.
- Added more places that check for time exceeding max time. This is more just to prevent overly large molecules from appearing to hang.
- If OMEGA is called with a fixed predicate, this now implies that FromCT is false. In other words, if you want to use a predicate to fix atoms, you need to use a 3D structure as input.

6.47 OMEGA 2.2.1

6.47.1 Bug fixes

- The new build algorithm introduced in v2.2.0 could, on rare occasions, generate a structure with heavy atom clashes. This release is mainly a bug fix for this problem.

CHAPTER

SEVEN

CITATION

Note

To cite **OMEGA**, please use the following:

OMEGA 7.0.0.0. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.eyesopen.com.

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; 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

CHAPTER

EIGHT

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

NINE

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

TEN

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.

10.1 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.

10.2 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.

10.3 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.

10.4 Publications for Bibliographies

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

10.4.1 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

10.4.2 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

10.4.3 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

10.4.4 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

10.4.5 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

10.5 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

ELEVEN

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

11.1 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.

11.1.1 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

11.1.2 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

-addfraglib
    omega command line option, 71
-addtorlib
    omega command line option, 74
-batchMaxRotors
    omega command line option, 76
-buildff
    makefraglib command line option, 85
    omega command line option, 72
-canonOrder
    omega command line option, 72
-commentEnergy
    omega command line option, 70
-deleteFixHydrogens
    omega command line option, 72
-dots
    command line option, 7
-enhstereo
    flipper command line option, 82
-enumAtomSpecifiedStereo
    flipper command line option, 82
-enumAtoms
    flipper command line option, 82
-enumBondSpecifiedStereo
    flipper command line option, 82
-enumBonds
    flipper command line option, 82
-enumBridgehead
    flipper command line option, 83
-enumEZ
    flipper command line option, 83
-enumNitrogen
    flipper command line option, 83
    omega command line option, 73
-enumRS
    flipper command line option, 83
-enumRelativeAtomStereo
    flipper command line option, 83
-enumRing
    omega command line option, 73
-enumSpecifiedStereo
    flipper command line option, 83
-erange
    omega command line option, 74
-ewindow
    macrocycle command line option, 77
    makefraglib command line option, 86
    omega command line option, 74
-ewindowRing
    makefraglib command line option, 86
-fail
    command line option, 5
-failed
    omega command line option, 70
-filter
    command line option, 5
-fixfile
    omega command line option, 72
-fixmcs
    omega command line option, 72
-fixrms
    omega command line option, 72
-fixsmarts
    omega command line option, 72
-flagTableFailures
    command line option, 6
-flipper
    omega command line option, 76
-flipper_enhstereo
    omega command line option, 76
-flipper_maxcenters
    omega command line option, 76
-flipper_warts
    omega command line option, 76
-fragGen
    makefraglib command line option, 86
-fragGenRing
    makefraglib command line option, 86
-fragKeep
    makefraglib command line option, 86
-fragKeepRing
    makefraglib command line option, 86
-fromCT

    makefraglib command line option, 86
    omega command line option, 72
-in
    command line option, 4, 70
    flipper command line option, 82
    macrocycle command line option, 77
    makefraglib command line option, 84
-includeInput
    omega command line option, 70
-info
    command line option, 5
-interval
    command line option, 5, 6
-iteration_cycle_size
    macrocycle command line option, 77
-log
    command line option, 6
    makefraglib command line option, 85
    omega command line option, 70
-maxCenters
    flipper command line option, 83
-maxConfRange
    omega command line option, 74
-maxTrials
    omega command line option, 75
-maxTrialsRange
    omega command line option, 75
-max_iterations
    macrocycle command line option, 77
-maxconfs
    macrocycle command line option, 78
    omega command line option, 74
-maxmatch
    omega command line option, 73
-maxrot
    omega command line option, 74
-maxtime
    makefraglib command line option, 86
    omega command line option, 74
-mcsMinAtoms
    omega command line option, 73
-mmff
    command line option, 6, 7
-molNames
    flipper command line option, 83
    makefraglib command line option, 85
-molnames
    macrocycle command line option, 77
-mpi_hostfile
    command line option, 70
    macrocycle command line option, 77
    makefraglib command line option, 85
-mpi_np
    command line option, 70
    macrocycle command line option, 77
    makefraglib command line option, 85
-newrule
    command line option, 6
-normalize
    command line option, 7
-out
    command line option, 4
    flipper command line option, 82
    macrocycle command line option, 77
    makefraglib command line option, 84
    omega command line option, 70
-param
    command line option, 5, 70
    macrocycle command line option, 77
    makefraglib command line option, 85
-pkanorm
    command line option, 6
-prefix
    command line option, 5
    flipper command line option, 83
    macrocycle command line option, 77
    makefraglib command line option, 85
    omega command line option, 71
-priorTemperature
    omega command line option, 76
-progress
    flipper command line option, 83
    macrocycle command line option, 77
    makefraglib command line option, 85
    omega command line option, 71
-rangeIncrement
    omega command line option, 74
-ref_tolerance
    macrocycle command line option, 78
-rms
    macrocycle command line option, 78
    makefraglib command line option, 86
    omega command line option, 74
-rmsRing
    makefraglib command line option, 86
-rmsrange
    omega command line option, 75
-rotorOffsetCompress
    omega command line option, 71
-salt
    command line option, 7
-sampleHydrogens
    omega command line option, 73
-sdEnergy
    omega command line option, 71
-sdtag
    command line option, 7
-searchff

    omega command line option, 75
-select
    command line option, 6
-settorlib
    omega command line option, 75
-startfact
    makefraglib command line option, 86
-startfactRing
    makefraglib command line option, 86
-strict
    omega command line option, 76
-strictatomtyping
    omega command line option, 73
-strictfrags
    omega command line option, 73
-strictstereo
    omega command line option, 76
-table
    command line option, 6
-thompson
    omega command line option, 75
-timeLimit
    makefraglib command line option, 86
-timeLimitRing
    makefraglib command line option, 86
-torlibtype
    omega command line option, 75
-trialsRangeIncrement
    omega command line option, 76
-typecheck
    command line option, 6
-umatch
    omega command line option, 73
-unique
    command line option, 7
-useGPU
    omega command line option, 75
-verbose
    makefraglib command line option, 85
    omega command line option, 71
-warts
    flipper command line option, 83
    omega command line option, 71

C

command line option
    -dots, 7
    -fail, 5
    -filter, 5
    -flagTableFailures, 6
    -in, 4, 70
    -info, 5
    -interval, 5, 6
    -log, 6
    -mmff, 6, 7
    -mpi_hostfile, 70
    -mpi_np, 70
    -newrule, 6
    -normalize, 7
    -out, 4
    -param, 5, 70
    -pkanorm, 6
    -prefix, 5
    -salt, 7
    -sdtag, 7
    -select, 6
    -table, 6
    -typecheck, 6
    -unique, 7

F

flipper command line option
    -enhstereo, 82
    -enumAtomSpecifiedStereo, 82
    -enumAtoms, 82
    -enumBondSpecifiedStereo, 82
    -enumBonds, 82
    -enumBridgehead, 83
    -enumEZ, 83
    -enumNitrogen, 83
    -enumRS, 83
    -enumRelativeAtomStereo, 83
    -enumSpecifiedStereo, 83
    -in, 82
    -maxCenters, 83
    -molNames, 83
    -out, 82
    -prefix, 83
    -progress, 83
    -warts, 83

M

macrocycle command line option
    -ewindow, 77
    -in, 77
    -iteration_cycle_size, 77
    -max_iterations, 77
    -maxconfs, 78
    -molnames, 77
    -mpi_hostfile, 77
    -mpi_np, 77
    -out, 77
    -param, 77
    -prefix, 77
    -progress, 77
    -ref_tolerance, 78
    -rms, 78
makefraglib command line option

-buildff, 85
-ewindow, 86
-ewindowRing, 86
-fragGen, 86
-fragGenRing, 86
-fragKeep, 86
-fragKeepRing, 86
-fromCT, 86
-in, 84
-log, 85
-maxtime, 86
-molNames, 85
-mpi_hostfile, 85
-mpi_np, 85
-out, 84
-param, 85
-prefix, 85
-progress, 85
-rms, 86
-rmsRing, 86
-startfact, 86
-startfactRing, 86
-timeLimit, 86
-timeLimitRing, 86
-verbose, 85

O

omega command line option
-addfraglib, 71
-addtorlib, 74
-batchMaxRotors, 76
-buildff, 72
-canonOrder, 72
-commentEnergy, 70
-deleteFixHydrogens, 72
-enumNitrogen, 73
-enumRing, 73
-erange, 74
-ewindow, 74
-failed, 70
-fixfile, 72
-fixmcs, 72
-fixrms, 72
-fixsmarts, 72
-flipper, 76
-flipper_enhstereo, 76
-flipper_maxcenters, 76
-flipper_warts, 76
-fromCT, 72
-includeInput, 70
-log, 70
-maxConfRange, 74
-maxTrials, 75
-maxTrialsRange, 75
-maxconfs, 74
-maxmatch, 73
-maxrot, 74
-maxtime, 74
-mcsMinAtoms, 73
-out, 70
-prefix, 71
-priorTemperature, 76
-progress, 71
-rangeIncrement, 74
-rms, 74
-rmsrange, 75
-rotorOffsetCompress, 71
-sampleHydrogens, 73
-sdEnergy, 71
-searchff, 75
-settorlib, 75
-strict, 76
-strictatomtyping, 73
-strictfrags, 73
-strictstereo, 76
-thompson, 75
-torlibtype, 75
-trialsRangeIncrement, 76
-umatch, 73
-useGPU, 75
-verbose, 71
-warts, 71