

OpenEye
SCIENTIFIC

ROCS

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 ROCS **3**
- 2.1 Overview . . . 3
- 2.2 Input Files . . . 4
- 2.3 Example Commands . . . 6
- 2.4 Report File . . . 7
- 2.5 Command Line Help . . . 9
- 2.6 Required Parameters . . . 9
- 2.7 Optional Parameters . . . 10

3 vROCS **17**
- 3.1 Overview . . . 17
- 3.2 Setup a simple run and a validation run . . . 18
- 3.3 Results visualization and analysis . . . 24
- 3.4 Statistics metrics . . . 32
- 3.5 Saving ROCS data . . . 38
- 3.6 ROCS shape query sources . . . 42
- 3.7 Editing ROCS queries in vROCS . . . 43
- 3.8 vROCS menus . . . 68

4 Utilities **75**
- 4.1 CheckCff . . . 75
- 4.2 Chunker . . . 78
- 4.3 HLMerge . . . 79
- 4.4 MakeRocsDB . . . 81
- 4.5 ROCSReport . . . 82

5 ROCS Tutorials **93**
- 5.1 Introduction . . . 93
- 5.2 Build/edit a query using the Wizard . . . 93
- 5.3 Build/edit a query manually . . . 95
- 5.4 Perform a ROCS validation run . . . 99
- 5.5 Perform a simple ROCS run . . . 102
- 5.6 Perform a ROCS run from the command line . . . 104

6 Theory **107**
- 6.1 Shape Theory . . . 107

6.2 Color Features . . . 110
6.3 Color Force Field . . . 111
6.4 Similarity Measures . . . 113

7 Release History **115**
7.1 ROCS 3.10.0 . . . 115
7.2 ROCS 3.9.0 . . . 116
7.3 ROCS 3.8.0 . . . 116
7.4 ROCS 3.7.0 . . . 117
7.5 Shape TK 3.7.0 . . . 117
7.6 ROCS 3.6.2 . . . 117
7.7 Shape TK 3.6.2 . . . 118
7.8 ROCS 3.6.1 . . . 118
7.9 Shape TK 3.6.1 . . . 118
7.10 ROCS 3.6.0 . . . 118
7.11 Shape TK 3.6.0 . . . 119
7.12 ROCS 3.5.1 . . . 119
7.13 Shape TK 3.5.1 . . . 119
7.14 ROCS 3.4.3 . . . 120
7.15 Shape TK 3.4.3 . . . 120
7.16 ROCS 3.4.2 . . . 121
7.17 Shape TK 3.4.2 . . . 121
7.18 ROCS 3.4.1.0 . . . 121
7.19 Shape TK 3.4.1 . . . 121
7.20 ROCS 3.4.0 . . . 121
7.21 ROCS 3.3.2 . . . 122
7.22 ROCS 3.3.1 . . . 122
7.23 ROCS 3.3.0 . . . 123
7.24 ROCS 3.2.2 . . . 123
7.25 ROCS 3.2.1 . . . 124
7.26 ROCS 3.2.0 . . . 125
7.27 ROCS 3.1.2 . . . 126
7.28 ROCS 3.1.1 . . . 126
7.29 ROCS 3.1.0 . . . 126
7.30 ROCS 3.0.0 . . . 127
7.31 ROCS 2.4.2 . . . 128
7.32 ROCS 2.4.1 . . . 128
7.33 ROCS 2.3.1 . . . 128
7.34 ROCS 2.3.0 . . . 129
7.35 ROCS 2.2.0 . . . 129

8 Selected ROCS Publications **133**

9 Citation **137**

10 Copyright and Trademarks **139**

11 Sample Code **141**

12 Citation **143**
12.1 Orion® . . . 143
12.2 Orion Floes . . . 143
12.3 Toolkits and Applications . . . 143
12.4 Publications for Bibliographies . . . 145
12.5 OpenEye MMDS Web Service . . . 146

13 Technology Licensing **147**
13.1 GCC 219

Index **235**

CHAPTER

ONE

INTRODUCTION

1.1 Overview

ROCS ® is a ligand-based virtual screening tool for aligning and scoring a database of molecules to a query or template molecule. The alignments can be used for a variety of purposes. The scores are used to rank molecules based on the probability that they share relevant (biological) properties with the query molecule.

ROCS aligns molecules based on shape similarity and their distributions of color or chemical features. The minimal inputs into **ROCS** are a query molecule in a single 3D conformation and a search database of molecules in multiple 3D conformations. The minimal output is a file of the best alignment and scores for each of the database molecules to the query.

1.2 Applications

The **ROCS** distribution comprises 2 applications:

ROCS

- Aligns and scores molecules in a database file to a query molecule.

vROCS

- A GUI application for **ROCS**.
- Interactive generation, editing, and validation of queries for **ROCS**.

1.3 Utility Programs

The following utility programs are also included in this distribution:

- *MakeRocsDB*: Generates a database in an optimized format for searching with **ROCS** from a database of molecules in 3D conformations from OMEGA.
- *Chunker*: Divides an input database into a specified number of pieces (chunks) of similar size. Used to generate sub-databases suitable for running large calculations in a `divide and conquer` fashion.
- *CheckCff*: Applies color features to input molecules that are set based on the input color force field file (CFF). Used to visually check that the CFF is functioning correctly.
- *HLMerge*: Merges multiple output files from **ROCS** into single file. (Re-)ranks **ROCS** hits based on a specified score. Used to combine results from `divide and conquer` calculations on sub-databases made using chunker.
- *ROCSReport*: Creates a multi-page PDF report document with the hit molecule file generated by ROCS.

CHAPTER

TWO

ROCS

2.1 Overview

The application of chemical similarity analysis in drug design is a commonly used and useful technique. Numerous topological (2D) and superposition (3D) methods exist for the measurement of chemical similarity. Methods that work in three dimensions have traditionally been much slower than 2D methods. This is due in large part to the fact that 3D methods must have some notion of the energetically accessible conformational ensembles available to a molecule, while 2D methods only work with a single structure. 3D methods, however, have the advantage of being able to find chemically less intuitive structures that have approximately the same shape and chemical properties. **ROCS** is designed to perform large scale 3D database searches by using a superposition method that finds the similar but non-intuitive compounds that are so valuable in the drug discovery process.

ROCS is a shape-based superposition method. Molecules are aligned by a solid-body optimization process that maximizes the overlap volume between them (See *Shape based alignment method in ROCS*). Volume overlap in this context is not the hard-sphere overlap volume, but rather a Gaussian-based overlap parameterized to reproduce hard-sphere volumes (see *Shape Characteristics and the use of Gaussians*). ROCS uses only the heavy atoms of a ligand, hydrogens are ignored. Since shape and volume in this context are so closely related, a volume overlap maximization procedure is an excellent method for gaining insights into similar shapes. Although **ROCS** is primarily a shape-based method, user specified definitions of chemistry can be included into the superposition and similarity analysis process which facilitates the identification of those compounds which are similar both in shape and chemistry.

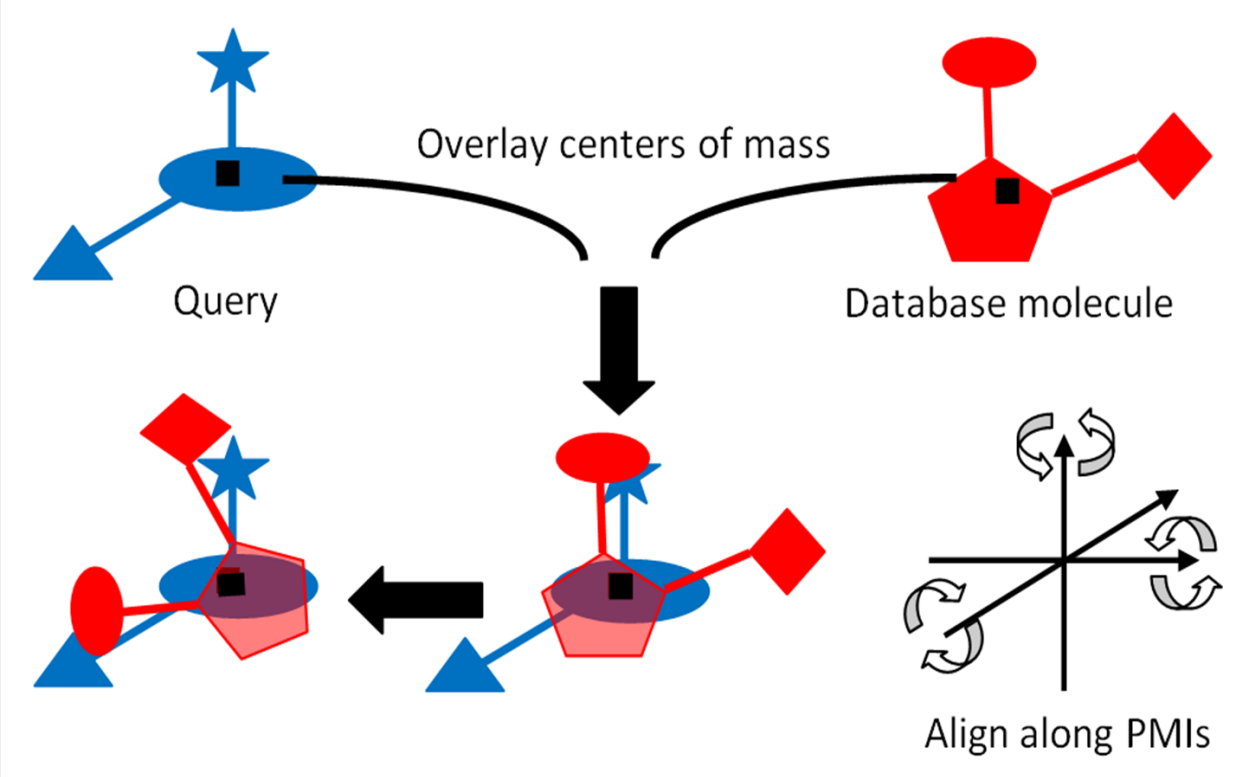


Fig. 1: **Shape based alignment method in ROCS**

Molecular superposition has had limited impact in 3D database searching because of the slow speed (1-2 molecules/second) of previously reported superposition methods. **ROCS** can routinely perform global shape and color alignments at the rate of 600-800 conformers per second. Medium-sized database searching (10's of millions of conformers) becomes tractable but slow at this rate of superposition. Distributed computing makes the entire process much more facile for screening larger numbers of compounds and conformers. **ROCS** can automatically split up similarity searches over entire networks of computers in an efficient and manner taking full advantage of parallel virtual machines. The coupling of shape and chemistry screening with a distributed architecture makes **ROCS** an incredibly powerful tool for searching large 3D databases.

2.2 Input Files

2.2.1 The Database File

The most common use of **ROCS** is overlaying a large collection of molecules onto a query (reference) molecule. For the purposes of this document, we'll call this large file the dbase (fit) file. The most common format for the dbase file is a multi-conformer OEBinary file created by OpenEye's OMEGA program, however, this file can be one of several 3D formats. These formats include SDF, MOL2 and PDB. ROCS determines the input file format from the file extension, `.sdf` or `.mol` for SDF, `.mol2` for MOL2, `.pdb` or `.ent` for PDB. Gzip compressed files of these same formats are allowed as well. **ROCS** will interpret `infile.sdf.gz` as a gzip'ed SDF file.

> **Note**
>
> Note that even though all these formats are supported, using SDF or MOL2 can result in a loss of speed due to the huge I/O penalty of these formats.

ROCS has no provision for conversion of 1D/2D molecules to 3D. The input file must already be 3D. More importantly, ROCS will interpret conformers in the input file as part of a single multi-conformer molecule as long as they:

- Are contiguous in the input file.
- Have the same numbers of atoms and bonds in the same order
- Have identical atom and bond properties with their order correspondent in the subsequent connection table
- Have the same atom and bond stereochemistry

While this may appear to be a restrictive list, many programs write multi-conformer molecules into SDF or MOL2 files such that the above rules will be satisfied. If the conformers are named differently, (*i.e.* they have a conformer number appended to the base name like acetsali_1, acetsali_2), **ROCS** will still consider them part of a single multi-conformer molecule if the criteria above are met. For file formats that are not inherently multi-conformer, this behavior can be turned off with the `-scdbase` command-line switch. With the `-scdbase` switch on, **ROCS** will *not* attempt to combine multiple conformers into a single multi-conformer molecule.

A new molecule file format, specifically for ROCS on large clusters is the `.rocsdb` format. See the *MakeRocsDB* section for when to use this file and how to create it.

One other file type is allowed as the dbase file. A file name ending in `.list` or `.lst` is assumed to be a list of actual molecule files, one per line. **ROCS** will then open each in turn and treat the entire collection as a single dbase file. Note that the conformer detection/concatenation code above will not span the gaps between these separate files.

Here is an example list file:

```
part1.oeb.gz
part2.oeb.gz
part3.oeb.gz
hits.mol2
```

2.2.2 The Query File

The second required input for a **ROCS** run is a file containing one or more molecules to be used as the query. **ROCS** will loop over molecules read in from the dbase file and attempt to overlay each of them against the query. In order to be consistent with other OpenEye software, this query molecule can also be referred to as the *reference molecule*.

Normally, **ROCS** treats each molecule in the query file as a single conformer molecule. For each molecule in the query file, **ROCS** will run a complete loop over the dbase molecules and write out a hits structure file and a report file, depending on the values of other command line switches described below.

Alternatively, **ROCS** can read queries as multi-conformer molecules by adding the `-mcquery` command line switch. In this mode, **ROCS** uses the same rules as described in the *The Database File* section to determine if two consecutive molecules are actually conformers of the same molecule. For each multi-conformer molecule in the query file, **ROCS** will loop over the dbase molecules' conformers comparing them to all query conformers. By default, **ROCS** will only return the single best overlay of this NxM set of comparisons. More than one can be returned by using the `-maxconfs` command line switch.

2.2.3 Shape Queries

Version 3.0 of **ROCS** introduced a new type of query called a shape query. It is a format that encompasses multiple elements of shape, including molecules, color features and grids. It can be generated from *vROCS* and saved in a shape query file with the extension `.sq`.

2.2.4 Grid Queries

ROCS can also use a grid instead of a molecule as a query ([Virtanen-2010]). These grids must be in GRASP, OpenEye, OpenEye ASCII Grid (.agd), CCP4, or XPLOR grid format and can be created with the OpenEye Grid toolkit or with a graphical application like GRASP. Certain **ROCS** features are not available when using a grid query. For example, the color force field features are not available with a grid query.

2.3 Example Commands

The example commands in this section can be run with files found under the appropriate version directory in `examples/rocs` under the top level installation directory.

ROCS always requires at the very least a file containing the query molecule(s) and a file containing the database molecule(s). The query file follows the *-query* command line flag and database file follows the *-dbase* flag. When **ROCS** is given no other arguments besides a query file and a database file, it will attempt to read the first query molecule, fit all database molecules to the query molecule, and write out the top 500 structures that have a Tanimoto Combo score above a given cutoff (default cutoff = -1.0). It is important to note that a matching structure, or *hi*, is the best fitting conformer of a database molecule. Only the best fitting conformer of any molecule will be written out. Even if multiple conformers of a molecule pass the cutoff, only the conformer which fits the best will be written out by default.

ROCS writes a structure file and a report file for each query molecule. The *-prefix* command line switch is used to name these files. The default prefix is **rocs**. The output structure file is by default `sdf` so that Shape Tanimoto and other calculated values can be included as tagged data, but the format can be changed by using the *-oformat* flag or by giving a specific filename using *-hitsfile*.

Note that as of **ROCS** 2.4, the defaults include using a color force field (ImplicitMillsDean), optimization against chemistry (-optchem true) and ranking the hitlist via TanimotoCombo (-rankby TanimotoCombo).

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf
```

will cause structures in the file `database.oeb.gz` that match query molecule in `4cox.sdf` to be written to a file called `rocs_hits_1.sdf`. A tab-delimited report file containing the scores will be written to `rocs_1.rpt`. If `rocs_hits_1.sdf` is viewed in **VIDA**, hits can be visually compared with the query and the numerical scores will appear in the spreadsheet. Molecules in the hits file and the report file will be ranked by their TanimotoCombo score.

To prevent continually over-writing output files, the *-prefix* flag allows you to give unique names to the files.

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -prefix FOO
```

will write the hit structures into a file named `FOO_hits_1.sdf` and the overlay values will be in a file called `FOO_1.rpt`. As you follow the rest of the examples in this section, you may wish to use different prefixes each time so that you can compare how the output files differ.

The *-cutoff* flag is used to control which database molecules are considered hits. By default this is set at -1.0. The following demonstrates changing the cutoff from the default value

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -cutoff 1.0
```

The difficulty in choosing a cutoff value is that the number of hits at a given value is not usually known *a priori*, so setting too high of a cutoff could result in no hits. The *-besthits* and *-maxhits* flags can be used in conjunction with specifying a cutoff value to coax **ROCS** into giving output of a manageable size. Quick searches can be done to assess an appropriate cutoff values for a particular query molecule. The following demonstrates a search that will give a quick answer:

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -cutoff 1.0 -maxhits 20
```

After 20 hits are found above a combo score of 1.0 in `database.oeb.gz` for the query molecule(s) in `4cox.sdf` the search terminates and the results are written. This option prevents the entire database file from being searched if a sufficient number of hits are found before the end of the database file. Finding the best N hits above a threshold tends to be a more common exercise. If the top N hits of a database up to a maximum of 100 and above a value of 1.0 are desired, the following search can be done:

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -cutoff 1.0 -besthits 100
```

If you just want the best N hits regardless of the cutoff, then using the default cutoff of -1.0 along with *-besthits* generates the N best:

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -besthits 100
```

If a report file alone is desired, the output of matching structures can be suppressed with the *-nostructs* option. For example:

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -cutoff 1.0 -nostructs
```

will only generate a report file for matching structures but the matches will not be written to a structure file.

By default, **ROCS** uses an inertial frame alignment to generate 4 separate starting positions, optimizes all 4 overlays and selects the best match of the 4. By default, this inertial frame alignment aligns the centers-of-mass of the two structures being aligned. If either molecule is substantially smaller than the other, this may not be the best starting position, so the choice to use random starting positions is offered. The command:

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -randomstarts 20
```

will use 20 random starting positions and keep the best score. Runtime is proportional to the number of starting positions, so using a large number for randomstarts can significantly slow down a ROCS job.

ROCS also calculates the Tversky coefficient based either on the fit (database) molecule (FitTversky) or on the reference (query) molecule (RefTversky). These scores will appear in the report file and in SD tags if the structure are written to an SD or OEB file. ROCS can use these other scores as the ranking score for the hitlist by using the *-rankby* switch.

To search a database and find the best 300 hits, scored by the FitTverskyCombo coefficient weighted to each database molecule:

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -rankby FitTverskyCombo -besthits 300
```

A chemical force field is used by default (ImplicitMillsDean) but a different one can be specified. Please refer to the chemical force field (CFF) section for a description of how to define a chemical force field. To simply calculate the CFF score after finding the best alignment based on shape use the *-chemff* option. For example:

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -chemff ExplicitMillsDean
```

To turn off all color and run ROCS as shape overlap only, you can use the *-shapeonly* flag:

```
prompt> rocs -dbase database.oeb.gz -query 4cox.sdf -shapeonly
```

2.4 Report File

The **ROCS** report file format appears as a tab-delimited file with the following fields. Since the names of the query and the hits are of indeterminate length, fixed size fields for these names could result in loss of information. Unfortunately this gives a file that is hard to read in a terminal session, but it can easily be read into a spreadsheet program or into the spreadsheet in **VIDA**.

Name
This is the name of the database molecule. If the database contains multi-conformer molecules, the specific conformer index is appended to the molecule name with an underscore if `-conflabel` is used.

ShapeQuery
This is the name of the query molecule. If the query is a multi-conformer molecule, then the specific conformer index is appended to the molecule name with an underscore.

Rank
The numerical ranking in the hitlist, based on the chosen score to sort by. Can be altered by using `-rankby` command line switches. If the `-stats` command line switch is used with **best** or **all**, data is written into the report file in the order that the search is performed. If no hitlist was used in the calculation, this field will be 0 (zero).

TanimotoCombo
To provide a score that includes both shape fit and color, the Shape Tanimoto is added to the Color Tanimoto, resulting in the TanimotoCombo score. This has a value between 0 and 2 and is the score used for ranking the hitlist when the `-rankby` *TanimotoCombo* command line switch is used.

ShapeTanimoto
This column gives the Shape Tanimoto, a value between 0 and 1 as calculated by the Tanimoto equation (see the *Theory* section).

ColorTanimoto
This column gives the Color Tanimoto, a value between 0 and 1 as calculated by the Tanimoto equation (see the *Theory* section).

FitTverskyCombo
To provide a Tversky score that includes both shape and color, the FitTversky is added to the FitColorTversky, resulting in the FitTverskyCombo score. This has a nominal value between 0 and 2, although do due the field nature of shape matching, the value can be higher than 2.

FitTversky
Shape Tversky is calculated using the Tversky equation (see the *Theory* section) with the fit (database) molecule as the main self-overlap with beta = 0.95. This was previously called Tversky(d).

FitColorTversky
Color Tversky is calculated using the Tversky equation (see the *Theory* section) with the database molecule as the main self-overlap with beta = 0.95.

RefTverskyCombo
To provide a Tversky score that includes both shape and color, the RefTversky is added to the RefColorTversky, resulting in the RefTverskyCombo score. This has a nominal value between 0 and 2, although do due the field nature of shape matching, the value can be higher than 2.

RefTversky
Shape Tversky is calculated using Tversky equation with the reference (query) molecule as the main self-overlap term with alpha = 0.95. This was previously called Tversky(q).

RefColorTversky
Color Tversky is calculated using the Tversky equation with the reference (query) molecule as the main self-overlap term with alpha = 0.95.

ColorScore
This column provides the actual color score. Since differing color force field files can use different strengths for the color forces and since each molecule may have a different number of color atoms, there is no upper bound on this score. By default, the color score is calculated by looping over all the color atoms in the query molecule and summing the single best color interaction with the hit molecule. This leads to scores that mirror the one-to-one correspondence of features sometimes seen in pharmacophore matching programs.

SubTan
One additional score can be calculated by giving the *-subtan* commandline argument. Since there is an additional time cost for this calculation, it is not included by default. Subtan is defined by taking the positions of the query and dbase molecule at the final overlay and removing all dbase atoms greater than 1.5 Angstroms from any query atom. Then a shape Tanimoto calculation is performed using these 2 structures and this Tanimoto coefficient is recorded as SubTan. Note that this has the effect of raising scores for small queries against much larger dbase molecules. In some respects, this is similar to Tversky for a sub-shape match, but does result in different rankings than Tversky. It is recommended that for searches involving a small query against a dbase of large molecules that both Tversky and SubTan be considered.

Overlap
This is the absolute value of volume overlap between the query and the dbase molecule. The value is in arbitrary units, and is most useful when using a grid as query.

DBConformerIndex
By default, the actual conformer index of each hit is appended to the molecule title. If the *-conflabel* `sdtag` or *-conflabel* `both` is specified, the conformer index will appear in this column.

2.5 Command Line Help

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

```
prompt> rocs --help
```

will generate the following output:

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

2.6 Required Parameters

-query <filename>
File containing molecules or a single grid to use a shape query.

For molecule queries, available formats (and file extensions) include:

File type	Extension
OEBinary	.oeb .oeb.gz
SDF	.sdf .mol .sdf.gz .mol.gz
MOL2	.mol2 .mol2.gz
PDB	.pdb .ent .pdb.gz .ent.gz
MacroModel	.mmod .mmod.gz

For grid queries, available formats (and file extensions) include:

Grid File type	Extension
OpenEye	.grd
Grasp	.phi
CCP4	.map .ccp4
XPLOR	.xplor .xplmap
ASCII Grid	.agd

For shape queries, only available format and file extension is:

Shape Query type	Extension
OpenEye	.sq

[Aliases=**-ref**,**-q**]

-dbase <filename>

File containing one or more 3D molecules to overlay against query from above. This flag supports all the same molecule file formats (**not** grids or shape queries) as *-query* plus the ROCS DB format `.rocsdb` and the list file (`.lst` or `.list`) as described in the *The Database File* section. [Aliases=**-fit**,**-db**]

2.7 Optional Parameters

2.7.1 Execute Options

-param

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

-mpi_np <n>

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

-mpi_hostfile <filename>

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

2.7.2 Input Options

-mcquery

Combine contiguous conformers in *-query* file into a multi-conformer query molecule, following the same rules for combining sequential conformers in the *-dbase* file. By default, this is false and each connection table in the *-query* file is treated as a separate query. Labelling the conformer by adding a wart to the name can be set using the *-qconflabel* parameter.

[Default=false]

-scdbase

Don't combine contiguous conformers in *-dbase* file into a multi-conformer molecule.

> **Note**
>
> For `.oeb` files that store a multi-conformer molecule directly, this switch has no effect.

[Default=false]

2.7.3 General Output Options

-prefix <name>

Prefix used to name output files. Using `-prefix FOO` will create a hits structure file named like `FOO_hits_1.sdf` and a report file, `FOO_1.rpt`, where `_1` will be replaced by a sequential number corresponding to the index of the query in the *-query* file. Additionally, a parameter file containing all options for the current run will be written to `FOO.parm`. This parameter file can be used with the *-param* switch.

[Default=rocs]

-outputdir <dirname>

Output directory for output files. The directory specified by this parameter must exist otherwise it will be ignored.

-besthits <N>

Search entire dbase file and keep a hitlist, sorted by score given by *-rankby* switch. Size of hitlist is determined by integer value N. Note that all members of the hitlist must pass the *-cutoff*, if given, so the final size can be smaller than the N requested. This switch is ignored if *-maxhits* is given. Note, that if this is set to zero (0) and *-maxhits* is also zero (0), then no hitlist will be maintained and all results will be streamed directly to the respective output files.

[Default=500]

-cutoff <F>

Cutoff (F) to determine whether a specific overlay should be considered good enough for hitlist inclusion. This is a floating point value and the actual parameter used for the scoring is as defined by the *-rankby* switch.

[Default=-1.0]

-rankby <score>

Score to use for ranking the hitlist. Legal values include:

- TanimotoCombo
- ShapeTanimoto (tanimoto)
- ColorTanimoto
- RefTverskyCombo
- RefTversky (tverskyq)
- RefColorTversky
- FitTverskyCombo
- FitTversky (tverskyd)
- FitColorTversky
- Overlap (overlap)

[Default=TanimotoCombo]

-maxconfs <N>

Maximum number of overlays returned for each comparison of a dbase molecule with a query molecule. This defaults to 1. As an example, if the query has `n` conformers and a given dbase molecule has `m` conformers, then a total of `nxm` overlays will be performed. By default, the single best one (1) will be returned (if it passes any *-cutoff* given). Choosing an alternate value for *-maxconfs* will cause up to the top `N` of these overlays to be returned and merged into the hitlist. In the hitlist, these conformers will not be associated with each other. Throughout a run, some can drop off the hitlist while others remain.

[Default=1]

-maxhits <N>

Maximum number of hits to return. This option causes ROCS to finish as soon as N molecules are in the hitlist. Useful for a quick check of a query to see what hits it is finding; this option overrides any value for *-besthits*.

[Default=0]

2.7.4 Hits Output Options

-conflabel

Controls where the conformer index from a database molecule gets labeled on output molecules. The allowed values are none, title, sdtag, and both.

[Default=title]

-qconflabel

Controls whether the conformer index from a query molecule gets labeled on output molecules. The allowed values are none and title.

[Default=title]

-outputquery

Put the query structures at the top of the output structure file. This is very useful for keeping the query structure in the same file as the hits, so that for instance, you only need to load one file into VIDA to browse the results. For a grid query, a copy of the grid will be written to PREFIX_ref.grd, where PREFIX is defined by the *-prefix* commandline option.

[Default=true]

-nostructs

Don't write a structure file. There are times when all you really want are the numerical results from ROCS. If you don't want or need an output structure file, you can prevent its creation with this switch.

[Default=false]

-hitsfile <filename>

Instead of writing to `PREFIX_hits_n.sdf` (for example) where PREFIX is provided by the *-prefix* commandline flag, write all hit structures to the file provided with this flag. Can be a filename or full/relative path. Also, if the name provided is actually a molecule file format extension (i.e. .sdf, .mol2.gz, .oeb, etc.), ROCS will write to `stdout` using the format derived from the file extension. For example if the following is used:

```
-hitsfile .sdf
```

then ROCS will write all the hits out to `stdout` in SDF format.

Note that this option will only work for a single query. If more than one query is provided along with the *-hitsfile* option, ROCS will issue an error and stop.

-oformat <extension>

Format for the output structure file(s). This option gives a file extension to be used for all output structure files. The format for the file is determined from the extension. Valid values include all the molecule file formats listed in the table above for *-query* files.

[Default=oeb.gz]

-sdTags

This parameter controls whether to attach score information to output molecules as SD data.

2.7.5 Report Output Options

-reportfile <filename>

Instead of writing to `PREFIX_n.rpt` where PREFIX is provided by the *-prefix* commandline flag, write all report information (stats) to the file provided with this flag. Can be filename or full/relative path. Note that if more than one query molecule is provided, this flag will not work unless the *-report* flag is also set to `one` to put all report info into one report file.

-report

Controls report file generation. The default, `each`, writes a separate report file for each query in the *-query* file. If `one` is chosen, stats for multiple queries in the same *-query* file will be placed in a single report file. This is useful for computing a NxN comparison of a file as both the dbase and query. Finally, to prevent ROCS from writing report files, use `none`.

[Default=each]

-stats

Determines which stats get placed into report files. Values include `hits` (the default), `best` and `all`. The default is to include just the stats for the compounds in the hitlist. If `best` is chosen, the report file will include stats for the best overlay(s) for every dbase molecule. The number of best overlay score is determined by the value of *-maxconfs*. Finally, if `all` is given, stats for every single overlay will be placed in the report file. Be careful. For a multi-conformer query against a large dbase file, `all` can generate a HUGE amount of data.

[Default=hits]

2.7.6 Status Output Options

-statusfile <filename>

Instead of writing to `PREFIX_n.status` where PREFIX is provided by the *-prefix* commandline flag, write all status information to the file provided with this flag. Can be filename or full/relative path. Note that if more than one query molecule is provided, the status written to this file is only for the most recently processed query.

-status

Controls status file generation. The default, `each`, writes a separate status file for each query in the *-query* file. If `one` is chosen, the status for multiple queries in the same *-query* file will be placed in a single status file. Note that if more than one query molecule is provided, the status written to this file is only for the most recently processed query. Finally, to prevent ROCS from writing status files, use `none`.

[Default=each]

2.7.7 Log Output Options

-logfile <filename>

Filename for log file. Overrides log filename created from *-prefix*.

-progress <method>

Method for showing job progress on the command line. Choices include:

- percent - show a percent complete progress bar (DEFAULT)
- log - echo the log message for each molecule
- logoutput - write each log line to screen
- dots - show dots
- none - print nothing to console

-verbose

Add extra verbosity to log file.

2.7.8 Shape Options

[Aliases=**-v**]

-tanimoto_cutoff <F>

Flag that can be used to limit output hits to only those with some minimum shape score. This can be used regardless of which score is chosen (*-rankby*) for ranking the hitlist. For example, using:

```
-rankby TanimotoCombo -cutoff 1.1 -tanimoto_cutoff 0.6
```

any molecule with Shape Tanimoto <= 0.6 will not be retained. Additionally, the constraint on TanimotoCombo to be at least 1.1 implies that Color Tanimoto must also be > 0.5 so that the sum can be greater than 1.1.

[Default=0.0]

-randomstarts <N>

Specifies number (N) of random starting positions to try instead of inertial frame overlay as described in the theory section. Since inertial frame alignment involves 4 (or 8 in the case of highly symmetric molecules) starting positions, setting *-randomstarts* to a value much larger will result in much slower run times.

-subtan

Also calculate sub-Tanimoto score. See the *Report File* section for a complete description of how sub-Tanimoto is calculated.

-subrocs

Specifies starting the search at all heavy atoms of the larger molecule as well as the default inertial starts. The larger molecule is chosen by comparing the self shape-overlap terms of the query and database molecule. The *-subrocs* option is especially useful when the query and database molecules have a large difference in size.

-shapeonly

A color force field is used by default and optimization against shape and color is the default overlap method. It is incompatible with a shape file. As an easy way to run ROCS with just shape overlap, this flag is the equivalent of setting:

```
-chemff none
-optchem false
-rankby tanimoto
```

-scoreonly

Perform scoring calculation only on input molecules. Sets the following flags:

```
-opt false
-optchem false
-besthits 0
-maxhits 0
-scdbase
```

-opt

Turn optimizer on if true, off if false. Not normally used by itself, but with other flags via the *-scoreonly* flag.

[Default=true]

2.7.9 Color Options

-chemff <cffname>

Color-force-field name. Either the name of one of the built-in color force fields (ImplicitMillsDean or ExplicitMillsDean) or the name of a user-defined color force field file. The format of this file is given in the *Color Force Field* section.

[Default=ImplicitMillsDean]

-optchem

Use color force field forces and gradients as part of overlay optimization. Ignored when *-opt* is `false`.

[Default=true]

-coloronly

Enables overlay optimization against only the color components, i.e., color atoms for a molecule or color gaussians for a shape query, of the query. This provides a color equivalent option of the corresponding `-shapeonly` option.

[Default=false]

CHAPTER

THREE

VROCS

3.1 Overview

vROCS provides a single user interface from which the user can build/edit **ROCS** queries, set up **ROCS** runs and analyze/visualize the results. It also includes rigorous statistics tools for validating a query, facilitating the comparison of different queries and selection of the most appropriate query for the project.

There are four primary workflows (tasks) in **vROCS** available from an initial *Welcome page* with a button for accessing each task. Each workflow provides the tools required to guide the user through the task. These workflows are:

1. Perform a simple **ROCS** run
2. Create a query with a wizard
3. Create or edit a query manually
4. Perform a **ROCS** validation

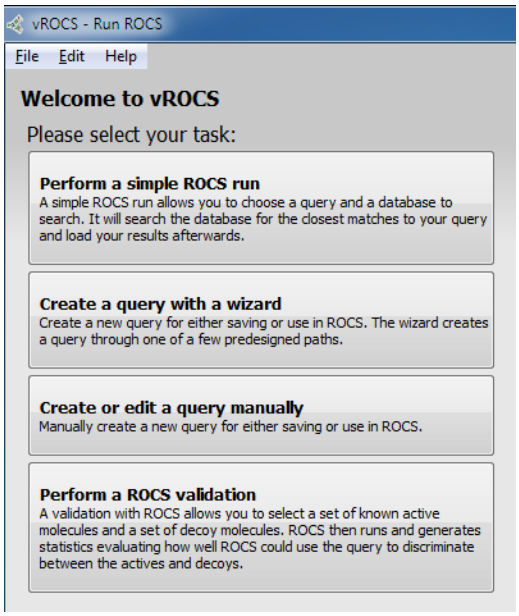


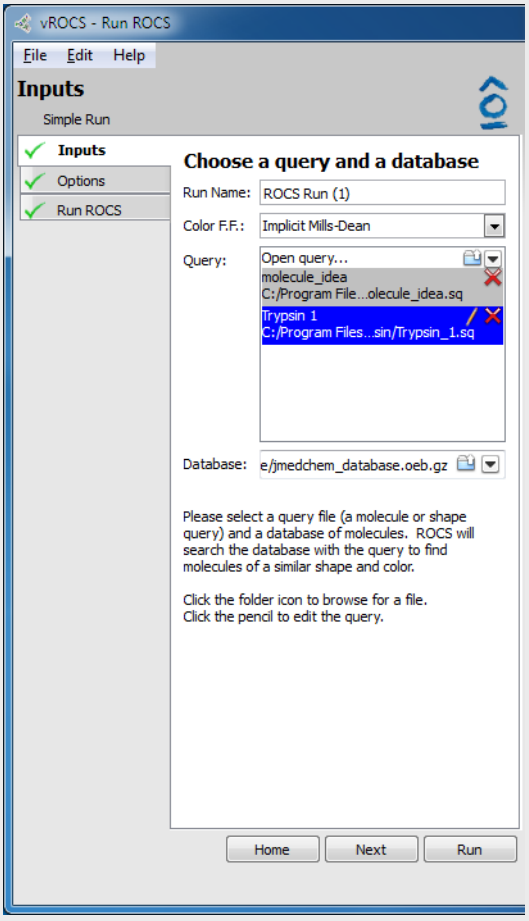
Fig. 1: **vROCS Welcome page**

3.2 Setup a simple run and a validation run

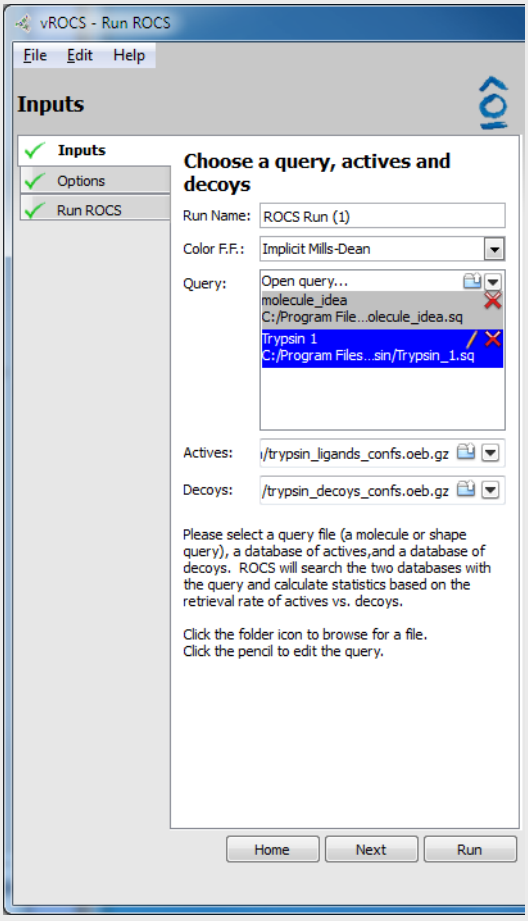
vROCS guides the user through all the steps of setting up and performing a **ROCS** run and visualizing and analyzing the results. There are two main run types for which a user would wish to employ **ROCS**.

1. Simple run
2. Validation run

From the Welcome page click on the button to `Perform a simple ROCS run` or `Perform a ROCS validation` to bring up the *Run set-up dialog*.



Simple run setup



Validation run setup

Run Name
Editable name that will be used for the **ROCS** run and displaying the results in **vROCS**.

Color F.F
A dropdown menu that allows selection of the current *color force field*. Options are:

- Implicit Mills Dean (default unless changed in User Preferences),

- Explicit Mills Dean

If a custom color force field was selected using `ROCS > Preferences` then this will also be available here. The current force field cannot be changed for a specific active query. Changing the current force field in the dropdown will filter the active query list to show only queries which use that color force field. Opening a saved query file (*.sq, *.sq.gz) will use the color force field previously associated with that file and the active force field in the Color F.F. dropdown will change to reflect this.

Query
List of queries that can be selected for the **ROCS** run. Click on `Open queries...` or the folder icon to browse to saved **ROCS** query files (molecules/grids/queries). Click on the black down arrow icon to select from a list of recently used queries. The source file path is shown below the opened query name. Queries built in the **vROCS** query editor are automatically added to this list for the current **vROCS** session. The query highlighted in blue is the selected (active) query. The query name can be edited (pencil) or the query can be deleted from the list (red X). Corresponds to the `-query` command line flag.

Database
The source of ligands which ROCS is to align to the query file during a simple ROCS run. Click on `Open database...` or the folder icon to browse to database files. Click on the black down arrow icon to select from a list of recently used databases. Corresponds to the `-dbase` command line flag.

Actives
The source of 'active' ligands which **ROCS** is to align to the query file during a **ROCS** validation run. Click on `Open database...` or the folder icon to browse to database files. Click on the black down arrow icon to select from a list of recently used databases.

Decoys
The source of 'decoy' ligands which **ROCS** is to align to the query file during a **ROCS** validation run. Click on `Open database...` or the folder icon to browse to database files. Click on the black down arrow icon to select from a list of recently used databases. The number of decoy ligands cannot exceed 100,000.

Home
Return to the Welcome screen or the **vROCS** query editor.

Next
Proceed to the Run set-up details Inputs tab. This button only becomes active once the query and database (or actives and decoys) files are selected.

Run
Run ROCS using the selected run name, query and database (or actives and decoys) and default parameters. Only becomes active once query and database (or actives and decoys) are selected.

The input for **ROCS** is a shape-based query with optional color atoms and one (or more) databases of molecules to search. The query shape is most frequently derived from a ligand of interest although other sources are possible, such as a variety of grids built in AFITT, Spicoli, OEDocking, OEChem and third party tools (see *ROCS Shape Query Sources*). **vROCS** allows the user to load a pre-saved query for **ROCS**, having 3D coordinates, or to build or modify one *in situ*. The list of available queries for a specific run is filtered based upon the type of color force field shown in the Color F.F. drop down. The database(s) are required to be prepared externally with 3D coordinates generated and conformers enumerated, usually by OMEGA. (See *Simple run setup and Validation run setup*)

3.2.1 Simple Run

A simple run aligns a database of pre-computed molecular conformers against a query. For each molecule in the database it overlays every conformer based on molecular shape with the option to employ color force fields. For a full description of the shape and Gaussian theory employed by **ROCS**, see *Shape Theory*. The conformers are scored based upon the Gaussian overlap to the query and the best scoring conformer is reported. The most common scores are ShapeTanimoto (shape alone), or TanimotoCombo (shape + color) which is the default for **ROCS** and **vROCS**.

The molecules in the database are finally ranked by the scores for their best aligned conformers. This type of simple **ROCS** run is commonly used when lead-hopping i.e. looking for structurally dissimilar molecules which have a higher probability of biological activity at the same target as the query while also overcoming issues such as ADME/Tox or patent coverage. Numerous literature examples of this application exist and some representative examples are given here.

Note

See *List of selected ROCS publications* for a list of selected **ROCS** publications. Lead-hopping examples include:

- *A Shape-Based 3-D Scaffold Hopping Method and its Application to a Bacterial Protein-Protein Interaction*
- *Scaffold hopping, synthesis and structure-activity relationships of 5,6-diaryl-pyrazine-2-amide derivatives: A novel series of CB1 receptor antagonists*
- *Novel Approach for Chemotype Hopping Based on Annotated Databases of Chemically Feasible Fragments and a Prospective Case Study: New Melanin Concentrating Hormone Antagonists*

3.2.2 Validation Run

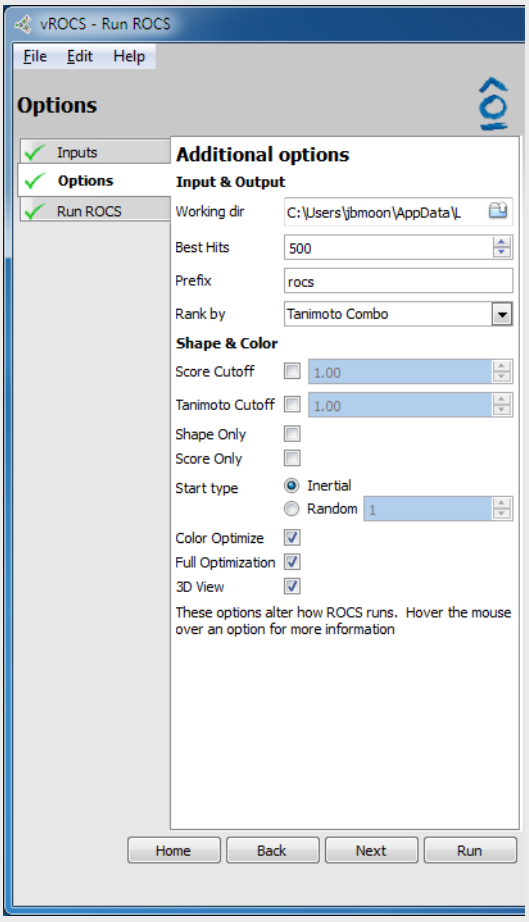
Before running a simple **ROCS** run on a large database, e.g. a corporate database of thousands or potentially millions of compounds (and even more conformers!), one should have confidence that the query is indeed able to distinguish true actives from inactives. For this purpose the validation **ROCS** run is employed. The major difference when setting up a validation run is that the validation run searches two sets of compounds, whereas the simple run searches only a single database. These two datasets are:

1. A set of molecules known to possess the desired biological activity. These are the actives.
2. A set of molecules known (or presumed) not to possess the desired biological activity. These are the decoys. The decoys can be a random set of molecules from a database or could be property matched (e.g. DUD [Huang-2006]) for a more stringent validation.

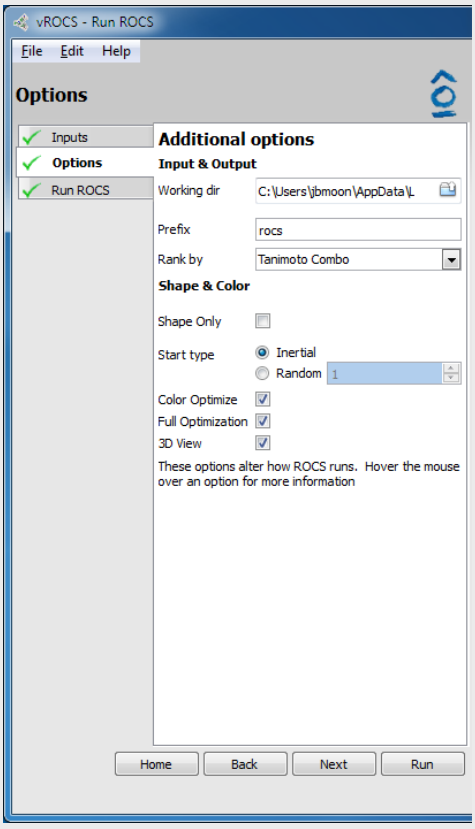
The method of alignment of compounds is the same for both run types (simple and validation). In the case of the validation run the desired result is that molecules from the set of actives are generally scored more highly than the set of decoys i.e. they have a greater shape similarity. Measurement of the degree of selectivity between these two datasets provides the user with confidence that the query is, indeed, selective and suitable for use in a simple **ROCS** run on a larger dataset.

A good validation experiment is vital to the success of future research. It needs to be carefully planned and set up e.g. selection of active and decoy datasets as well as query design (see *Editing ROCS Queries in vROCS*). For example, is a modification to a query really beneficial to the selectivity of that query? The rigorous use of validated statistical methods and parameters to analyze and, more importantly, compare runs is essential and frequently overlooked. For that reason statistical analysis tools are included in **vROCS** when visualizing the results of a **ROCS** validation run. These are described below in *Statistics Metrics*.

The run set-up options pages (See *Simple run options and Validation run options*) in **vROCS** are pre-populated with the default **ROCS** options e.g. how compounds are initially oriented and aligned and how alignments are scored and ranked. These default values are calculated to give a good starting point in the majority of examples. However, these are also some of the most common options that a **ROCS** user might want to modify. For example, changing the start type from inertial to random can be particularly useful for a grid-based query (as opposed to a shape-based query) because it is more difficult to identify and set the 4 true inertial points for a grid. The disadvantage of using random starts and setting this number to be high is that it will significantly increase the run time. Deselecting the 3D view option will speed up runs, particularly on computers with limited compute resources. By default an Open GL 3D alignment for each compound is shown as the run progresses and, since this can be somewhat CPU intensive, switching the display off can be beneficial.



Simple run options



Validation run options

Working Directory

Directory in which the files for the **ROCS** run are to be saved. Default location is the **vROCS** installation directory, if it is user writable, otherwise, a temporary directory is used. Click on the folder icon to browse and select alternative directories. Corresponds to the `-outputdir` command line flag.

Best Hits

Number of top ranking hits to be saved after searching the entire database. Use the arrows to increase/decrease or type the desired number in the field. Corresponds to the `-besthits` command line flag. Only available for simple **ROCS** run.

Prefix

Naming prefix for the current **ROCS** run. All the output **ROCS** files will contain this name. If no name is specified the default is "rocs". Corresponds to the `-prefix` command line flag. The output files using the prefix are: Parameter file (prefix.parm), Log file (prefix.log), Report file (prefix_1.rpt), Status file (prefix_1.status), Structure file (prefix_hits_1.oeb.gz)

Rank by
Dropdown allows selection of one of the many score types available in vROCS. The results will be ranked by the selected score for selection of Best N hits (above). Default is TanimotoCombo. Corresponds to the `-rankby` command line flag. Available scores are: TanimotoCombo, ShapeTanimoto, ColorTanimoto, Combo Reference Tversky, Shape Reference Tversky, Color Reference Tversky, Combo Fit Tversky, Shape Fit Tversky, Color Fit Tversky, Overlap

Score Cutoff
Check the check box to exclude any hit with a score less than the specified value from the hitlist. The score used is the one specified by the Rank by field. Change the cutoff value by using the arrows to increase/decrease or type the desired number in the field. The allowed score range varies according to the score selected in the Rank by field. Corresponds to the `-cutoff` command line flag. Only available for simple **ROCS** run.

Tanimoto Cutoff
Check the check box to exclude any hit with a ShapeTanimoto score less than the specified value from the hitlist. Change the cutoff value by using the arrows to increase/decrease or type the desired number in the field. The allowed score range is 0-1 (min-max). Corresponds to the `-tanimoto_cutoff` command line flag. Only available for simple **ROCS** run.

Shape Only
Check the check box to perform a shape only overlay, turning off the color force field. Corresponds to the `-shape_only` command line flag.

Score Only
Check the check box to score the incoming poses against the query in their current 3D coordinate frame, turning off alignment and hitlist. This is useful for scoring a pre-aligned dataset. Corresponds to the `-score_only` command line flag. Only available for simple **ROCS** run.

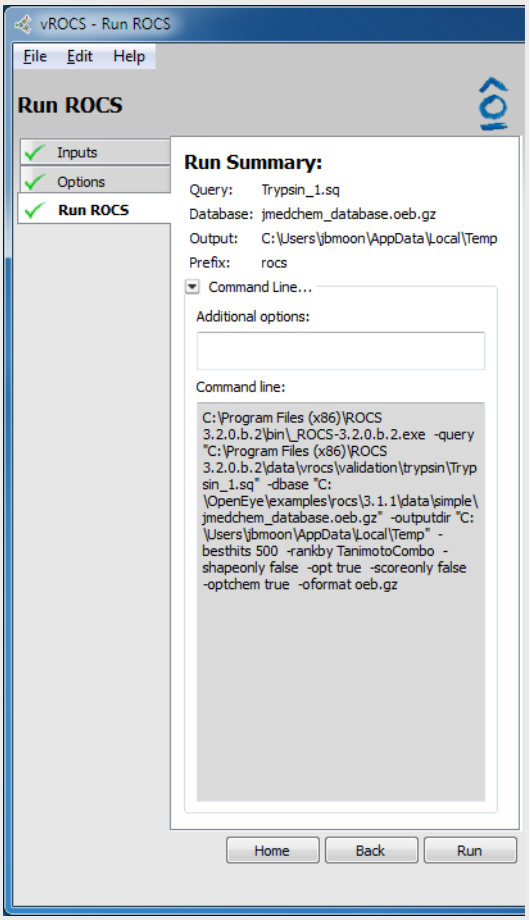
Start Type
Use the radio buttons to specify how **ROCS** places the initial alignment. Inertial is the default option and uses 4 initial starts. Random specifies using random starts for the initial overlay and corresponds to the `-randomstarts` command line flag. Specify the number of random starting configurations by using the arrows to increase/decrease or type the desired number in the field.

Color Optimize
Check the check box to use the color force field in the optimization of the alignments. Default is checked on. Corresponds to the `-optchem` command line flag.

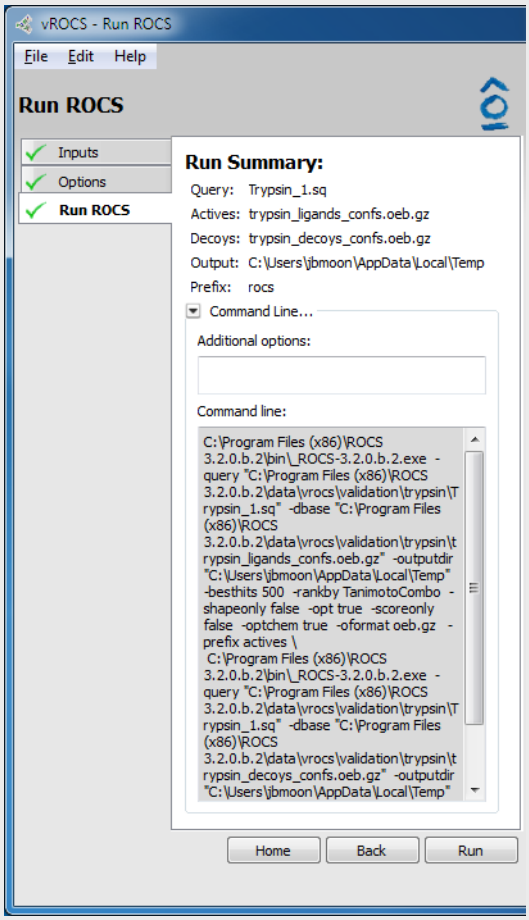
Full Optimization
Check the check box to perform full best overlay optimization. Default is checked on. If off (false) then score only. Corresponds to the `-opt` command line flag.

3D View
Check the check box to select whether a 3D view of the query and database molecules aligning is displayed as the run progresses. Default is on. If checked off a text-based progress screen is displayed. This will increase ROCS' run speed on low powered computers.

The final page of set-up is the Run Summary on the *Run Rocs* page. The summary gives a quick rundown of the query file and database used, as well as the **ROCS** version. It also contains a collapsible panel to display the full set of command line options that will be fed to **ROCS** and will be saved as the ROCS parameter file (.parm). This can be useful when setting up and validating runs in **vROCS** that will later be run on the command line across a remote cluster. The **Additional Options** prompt allows entry of a command not listed in the command line such as a new parameter not yet available in the released version of **ROCS**.



Simple run summary



Validation run summary

Query
: Query file as specified on the Inputs tab

Database
: Database file as specified on the Inputs tab in a simple run

Actives
: Actives database file as specified on the Inputs tab in a validation run

Decoys
: Decoys database file as specified on the Inputs tab in a validation run

Output
: Working directory where all files will be written.

Prefix
: Naming prefix for the output files that will be written to the Working Directory, as specified in the Options tab. This is defined by the Prefix field on the Options tab.

Command Line...
: Click to display/hide the full command line that will be sent to the ROCS executable. This can be copied to export and use in command line ROCS installations. A field is available for typing additional ROCS parameters that will be included in the command line not exposed via the vROCS interface. Note that the command line may use temporary files in some instances.

3.3 Results visualization and analysis

The **vROCS** interface provides multiple tools for results visualization and analysis. The *3D visualization window* shows the query where the molecule structure is displayed as green sticks with associated shape and color atoms. All three portions (molecule, shape and color) can be made visible or hidden using controls in the window. The aligned hit molecules are shown as sticks colored by atom type. Buttons at the bottom of the 3D window allow the shape grid, shape atoms, color atoms and color atom labels to be toggled on or off. The color of the shape contour can be changed and the contour level displayed for the shape grid can also be modified using a slider. This is particularly useful when adding color atoms to a grid-based query, for example.

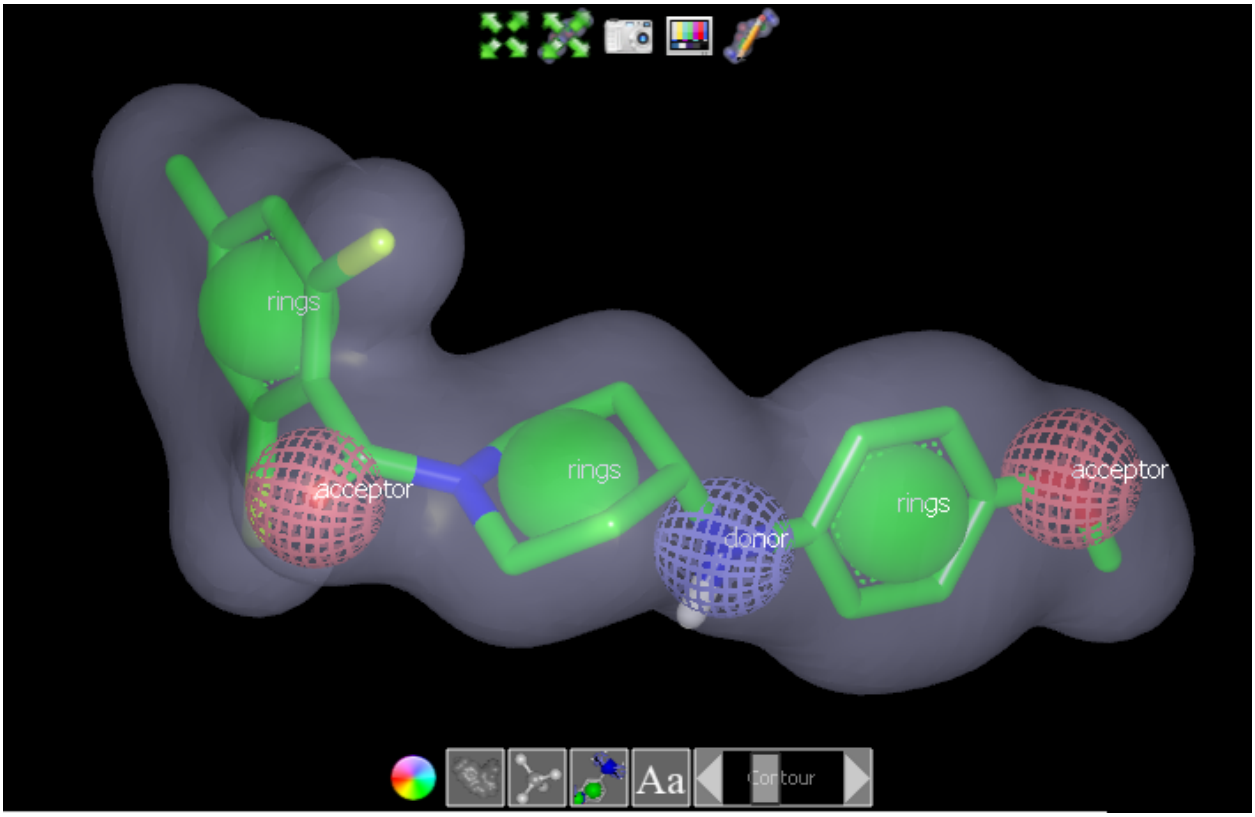
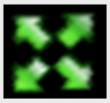
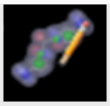
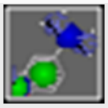


Fig. 2: **3D visualization window**

Icon	Description
	Fit scene to screen
	Fit query to screen
	Take screenshot of 3D Window (excludes the gray query information panel)
	Show/hide the *3D parameters control window*
	Edit query: Open the Edit Query panel and add the query editing icons (See *Editing ROCS Queries in vROCS*). This icon is replaced by a Done Editing icon while in editing mode.
	Change color of the contour
	Toggle display of the shape contour on/off
	Toggle display of shape atoms on/off
	Toggle display of color atoms on/off
	Toggle display of color atom labels on/off
	Slider to adjust display level of the shape contour from 0-3 (default 1). This only changes the contour display and not the query itself

The *3D parameters control window* provides user control for graphics rendering of the image in the 3D window.

The font size for text labels in the 3D display can be altered, as can the stereo visualization type and settings. Not all stereo settings are available on all machines and therefore some stereo options may be grayed out. See *the 3D parameters table* below for details.

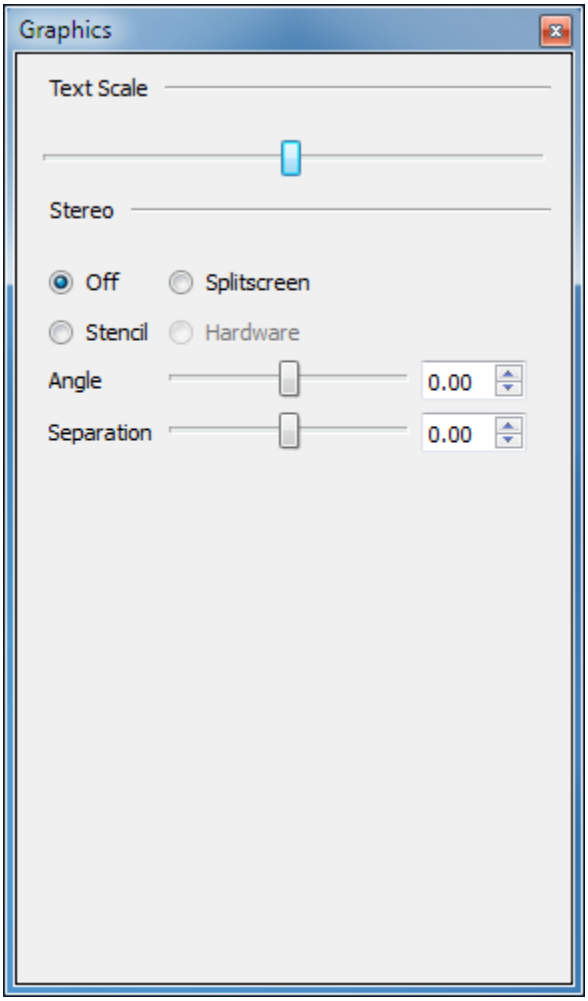


Fig. 3: **3D Parameters control window**

Text Scale
Slider to adjust the size of the font for the color atom labels.

Stereo Off
Disable stereo graphics.

Splitscreen
Display the image in the 3D window in splitscreen stereo mode for unassisted 3D viewing.

Stencil
Display the image in the 3D window in a format suitable for viewing with a Zalman Trimon LCD 3D monitor (or similar hardware).

Hardware
Enabled only on machines which are capable of performing 3D hardware stereo-in-a-window. Hardware stereo requires a graphics card that supports "stereo in a window" display as well as the appropriate stereo glasses.

Angle
Slider to adjust the angle between the images for splitscreen, stencil or hardware stereo modes.

Separation

Slider to adjust the separation between the images for splitscreen, stencil or hardware stereo modes.

A *results spreadsheet* below the main 3D window lists results for each molecule (with best-fitting conformer number) and its associated scores. The data is displayed for the run associated with the highlighted Run Name tab. Individual or multiple molecules can be observed overlaid with the query in the 3D window. Only the top 20 scoring molecules are displayed in the spreadsheet, based on the `Rank by` score selected in the Run Set-up Options tab. The spreadsheet can be resorted by clicking on other column headers, and the top (or bottom) results for that column will be displayed. Note: this can be a DIFFERENT set of 20 molecules than were displayed originally. To see ALL results users are encouraged to use the spreadsheet tools in **VIDA**. This can be done by right-clicking on the Run Name tab in the results panel and following the option to "Open 'Run Name' in **VIDA**".

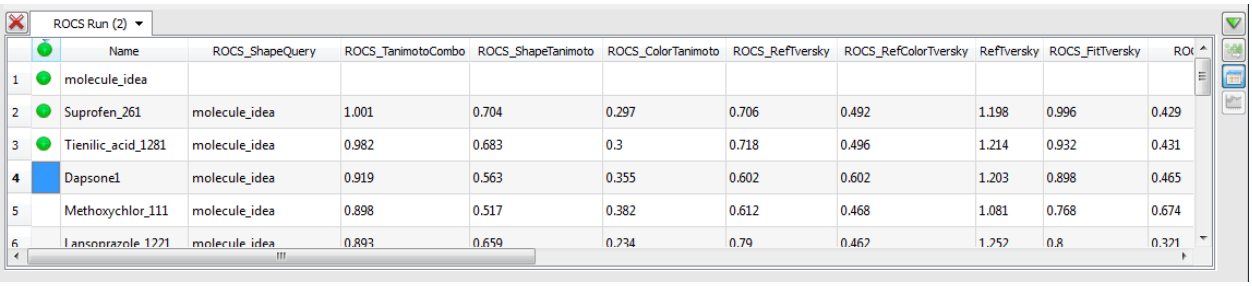


Fig. 4: **Results Spreadsheet**

Icon	Description
	Display/hide the results panel.
	Show the ROCS output. This is the information that would be displayed in the terminal window during a command line ROCS run.
	Show the results spreadsheet.
	Show the statistics panel. Only available for ROCS validation run.
	Make this compound visible in the 3D window and keep it visible while scrolling through other results.
	Delete the results for the highlighted Run Name tab

The spreadsheet columns include the name of the database compound, the name of the query, 14 different scores (see *section Report File* for full definitions) and a rank column (based on the score type chosen when setting up the run). The available scores are:

1. TanimotoCombo
2. ShapeTanimoto
3. ColorTanimoto

4. Ref Tversky
5. RefColorTversky
6. RefTverskyCombo
7. FitTversky
8. FitColorTversky
9. FitTverskyCombo
10. ColorScore - score type from older **ROCS** versions not available as a Rank by… choice but can be used to sort the spreadsheet.
11. SubTan - score type from older **ROCS** versions not available as a Rank by… choice but can be used to sort the spreadsheet.
12. Overlap

The spreadsheet for a validation run has an additional three columns:

13. Active - indicates whether the compound was in the set of actives (1) or decoys (0)
14. Rocs_db_index - identifies the placement of each compound in the database ROCS formed by combining the active and decoy sets prior to search. This is required in case a compound in the actives and decoys happens to have the same name.
15. Lingos similarity - the 2D fingerprint similarity to the query, if the query is a molecule

The spreadsheet can be sorted by any field. If an alternative score is chosen for sorting then the best 20 molecules by that score will be displayed. This may be a different set of molecules from the original 20 displayed because **vROCS** sorts and retrieves data from the saved structure hitlist and report files. Additionally, the spreadsheet includes controls to show/hide or mark each molecule. This allows the user to compare overlays between compounds and against the query in the 3D visualization window as well as control the data that is saved out.

The most common scores used are ShapeTanimoto (shape only) or the default score, TanimotoCombo (shape + color). Tanimoto scores should be used when the query and database molecules are a similar size. Tversky scores include a weighting factor to deal with size differences and are therefore useful when the query is small and the database molecules are large, or *vice versa*. The RefTversky score is weighted for a small query e.g. to find all instances of a known active scaffold fragment in a database. The FitTversky score has the opposite weighting.

Additionally the validation runs have a *statistics panel* available. It provides several statistical metrics for analysis of the quality of the results. The metrics reported in **vROCS** consist of the following and are described below:

1. ROC (receiver operating characteristic) curve together with its AUC (area under the curve) 95% confidence limits
2. Score histogram to examine the distribution of scores obtained for the active and decoy datasets
3. Early enrichment at 0.5%, 1% and 2% of decoys retrieved 95% confidence limits
4. When comparing multiple runs p-values are calculated for each enrichment level & AUC

These metrics and the rationale behind their inclusion are fully described in the section *Statistics Metrics*.

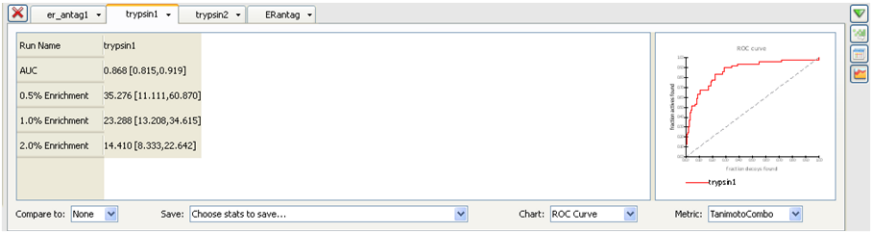


Fig. 5: **Statistics Panel**

Compare to
: Add the results from another run to the statistics spreadsheet and calculate the p-values between the two runs. The additional run will also be plotted in the ROC curve and score histogram. The dropdown lists None, Lingos and all other validation runs available from that vROCS session. Lingos is the 2D similarity and is always available as a comparison choice with molecular queries. Default is None.

Save
: Chose score, plot or spreadsheet data to save as `.csv` format for the active run. If another run(s) is selected in the `Compare to` field that data will also be saved. Select Plot data to save an image file of the ROC plot or score histogram.

Chart
: Select from a dropdown whether to display the ROC curve or the score histogram

Metric
: Select one of the scores (metrics) to be used for the ROC plot or score histogram. These correspond to the score columns in the results spreadsheet

The statistics panel includes a spreadsheet listing the values for the statistics metrics, together with score histograms and an ROC plot from which is calculated the AUC (See section *Statistics metrics*. The ROC plot graphs actives *vs* decoys and a higher AUC represents greater selectivity in favor of the actives. The ROC curve can be plotted for any of the 14 scores available (See *ROC plot*). Note that changes to the score used for the ROC plot will probably cause changes to the AUC and enrichment values.

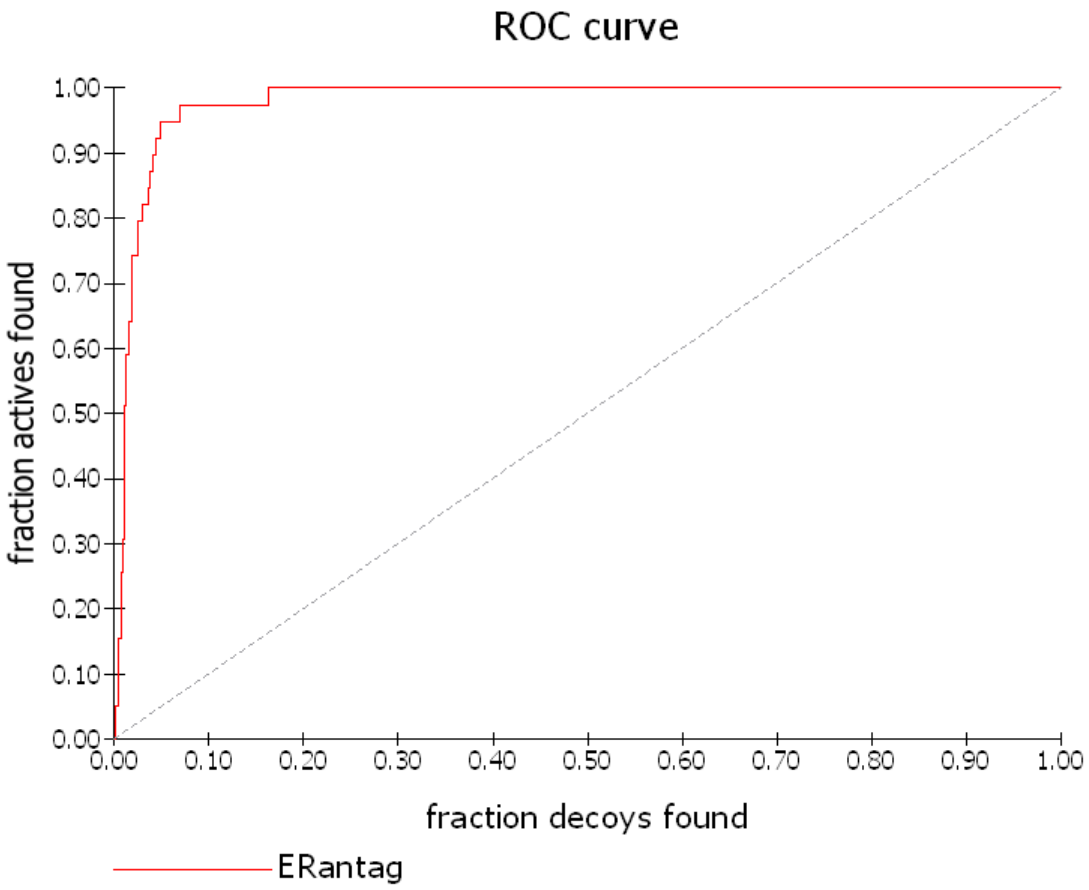


Fig. 6: **ROC Curve**

Instead of the ROC plot a score histogram can be plotted. The score histogram compares the distribution of scores for the actives and the decoys (See *Score Histogram*). The better the AUC (closer to 1.0), the greater the separation will be, in general, between the two histograms, with the actives scoring higher and further to the right than the decoys.

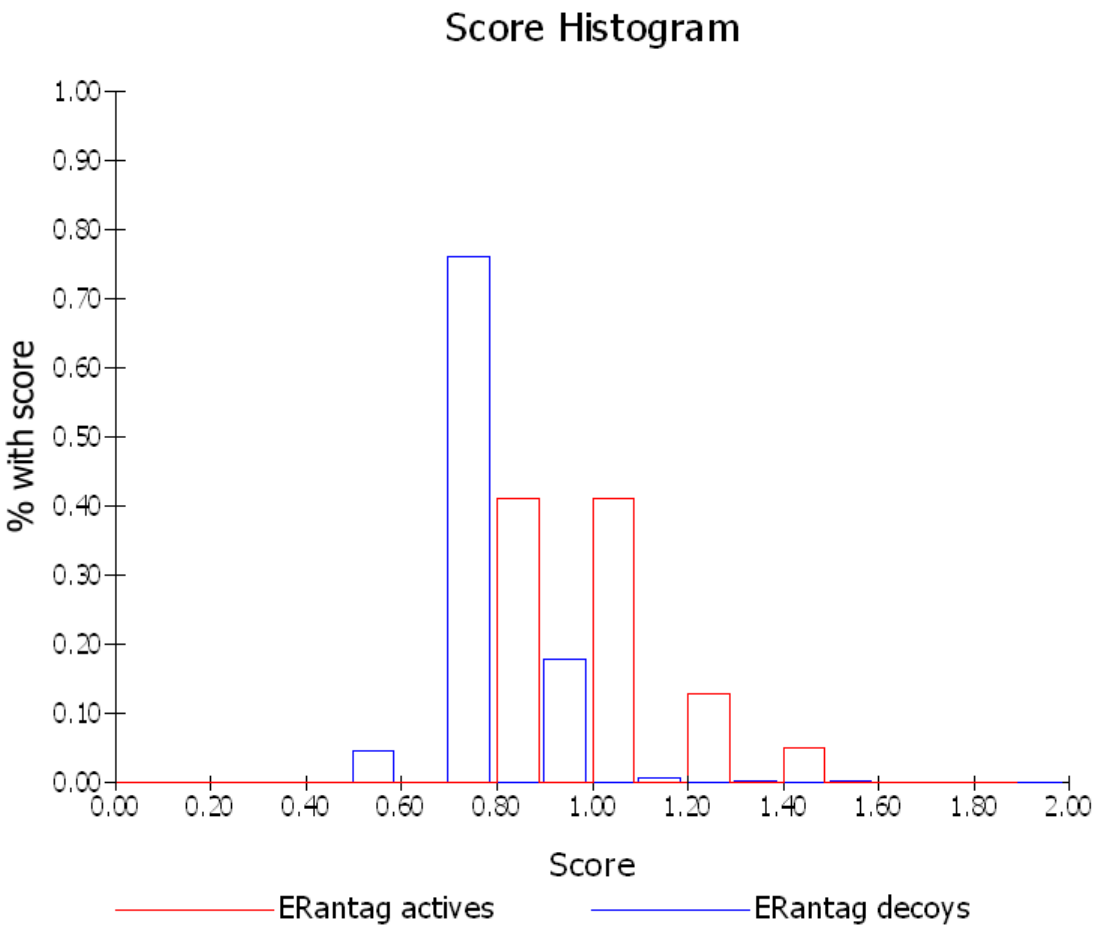


Fig. 7: **Score Histogram**

To better visualize the plots the plot area can be resized by dragging the divider between the plot and spreadsheet. The statistics panel can also be resized by moving the divider between the panel and the 3D window.

Multiple runs can be compared in the spreadsheet. The statistics panel enables the comparison of multiple validation runs using the `Compare to` dropdown and the data and plots can be exported to a CSV (`.csv`) file for import into other applications or statistics packages. The statistics for multiple runs will be displayed side by side in a spreadsheet and these runs will be plotted together on the ROC plot and score histogram for a direct comparison. This could help to answer the following questions:

- Is one query more selective than another on the same database?
- Is the query selectivity the same for multiple training databases? Was a representative validation database selected?

When comparing two runs it is useful to gauge whether one is giving statistically better results than another. For this reason p-values are displayed in the comparison (see *Statistics for comparison of ROCS runs*). A low p-value suggests that the base run is statistically better than the run selected in the `Compare to` dropdown. For a description of p-values see section *Statistics Metrics*. If comparing particularly large data sets it is wise to pay attention the memory foot print – save and close any unneeded runs.

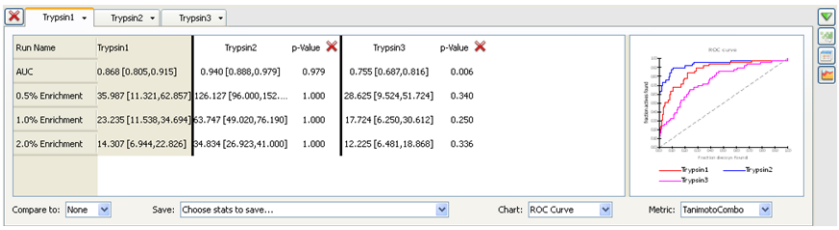


Fig. 8: **Statistics for comparison of ROCS runs**

While it is more open to individual interpretation, inspection of the overlays in the 3D window should not be overlooked as a valuable tool for results interpretation. Can additional knowledge of the receptor be applied that validates the ROCS alignments?

3.4 Statistics metrics

To facilitate accurate understanding, interpretation and comparison of virtual screening results from multiple (independent) experiments when publishing or presenting research it is important to use consistent and industry standard metrics. To date no official industry standard has been set. However, steps and recommendations were made in this direction at the "Evaluation of Computational Methods" symposium at the 234th American Chemical Society in August 2007 and the follow-up Journal of Computer Aided Molecular Design issue 22 in March 2008. Measures that have become standard in other fields tend to possess the following short list of characteristics:

1. Independence to extensive variables
2. Robustness
3. Straightforward assessment of error bounds
4. No free parameters
5. Easily understood and interpretable

The widespread and habitual use of good reporting practice is something that OpenEye is keen to encourage and therefore **vROCS** implements statistics metrics discussed in these recommendations ([Jain-2008], [Nicholls-2008]).

The metrics reported in **vROCS** consist of the following and are described below:

1. ROC (receiver operating characteristic) curve together with its AUC (area under the curve) 95% confidence limits
2. Early enrichment at 0.5%, 1% and 2% of decoys retrieved 95% confidence limits
3. When comparing multiple runs p-values are calculated for each enrichment level

3.4.1 ROC Curve

A ROC curve ([Hanley-1982]) in **vROCS** plots the percentage (or fraction) of actives found on the Y-axis versus the percentage of decoys on the X-axis as the scores decrease. The top scoring compounds are plotted closest to the origin. It gives an indication of how the actives and inactives are ranked as a result of the ROCS run. An ideal ROC plot for a perfectly selective query would show all of the actives being identified first because they score most highly. The plot would shoot up the Y axis at X=0. Then the lower scoring decoys would be plotted and the curve would follow the X axis at Y=100 %. *ROC Plot 2* illustrates an ROC plot for an almost perfectly selective query where most of the actives rank more highly than most of the decoy molecules.

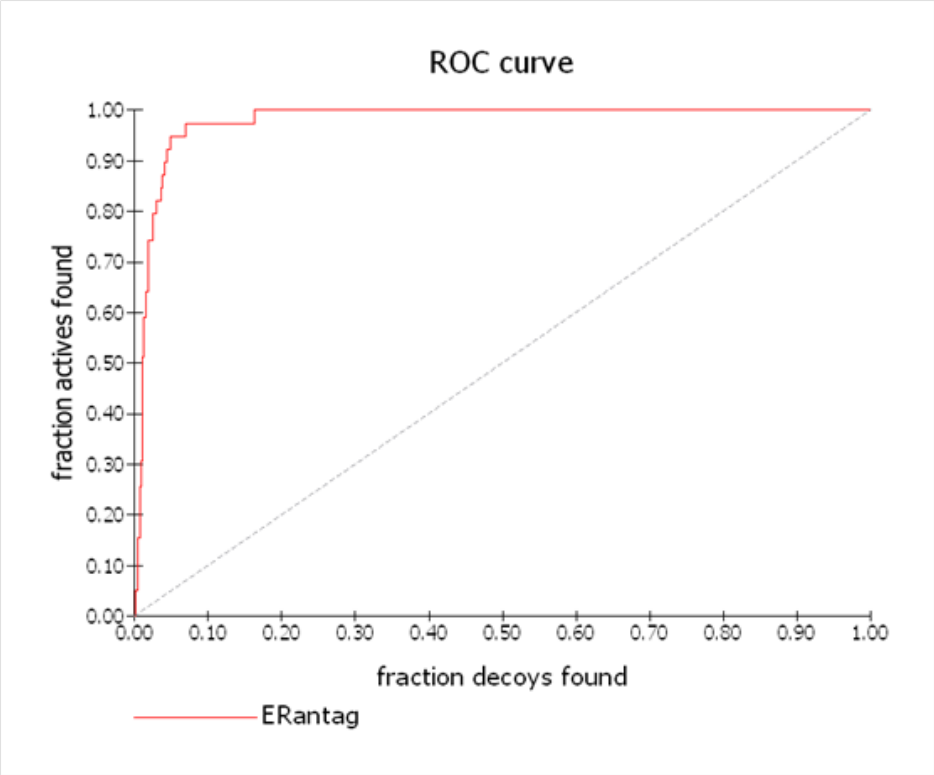


Fig. 9: **ROC Plot 2**
An almost perfectly selective ROC curve with AUC = 0.979 where most of the actives rank more highly than most of the decoy molecules. The dashed diagonal line represents random.

3.4.2 AUC

The AUC (area under the curve of an ROC plot) is simply the probability that a randomly chosen active has a higher score than a randomly chosen inactive. A useless query, one with no better chance of identifying an active from an inactive, would give an AUC of exactly 0.5, as shown by the dotted line in *ROC Plot 2*. A perfect query is one which ranks all the actives above all the inactives. In this case the AUC would be 1.0. In most cases the observed AUC will be somewhere between these two extremes, and for a highly selective query it will often be in the 0.8-1.0 range. Sometimes an AUC of < 0.5 is observed. This occurs when the query is scoring the decoys more highly that the actives i.e. it is selective for the inactives.

Note

AUC has, for a long time, been a standard metric for other fields. The main complaint against the AUC is that is does not directly answer the questions some want posed, i.e. the performance of a method in the top few percent. It is a global measure and therefore it reflects the performance throughout a ranked list. Thus, the notion of "early enrichment" may not be well characterized by just AUC, particularly when virtual screening methods yield AUC values short of the 0.8-1.0 range. For this reason we include early enrichment values in the **ROCS** output for a validation run. Early enrichment, while certainly more reflective of the common usage of virtual screening methods, is a property of the experiment conducted, not the methods being studied in that experiment and thus should be used with care.

AUC is quoted in **vROCS** as a mean value 95% confidence limits. Bootstrapping the data produces a set of samples from which the mean and confidence levels are obtained.

3.4.3 Enrichment

Consider the example in *Early Enrichment Comparison* below taken from the Nicholls paper ([Nicholls-2008]) which illustrates how AUC provides no information on early enrichment. Both the Early (pink) and Late (blue) curves have an AUC of exactly 0.5. Clearly both examples are equally likely to score an active higher than an inactive (or *vice versa*) overall. However, the solid (pink) plot also shows that some fraction of the actives is scoring significantly higher than the inactives, while another fraction of the actives scores worse. In a virtual screen it is desirable not to screen the entire database but to select only the top scoring fraction of the compounds. Only the average behavior across the whole database, not the early enrichment of actives in the solid pink plot, is reflected in the AUC. Thus, it is beneficial to report early enrichment in addition to AUC.

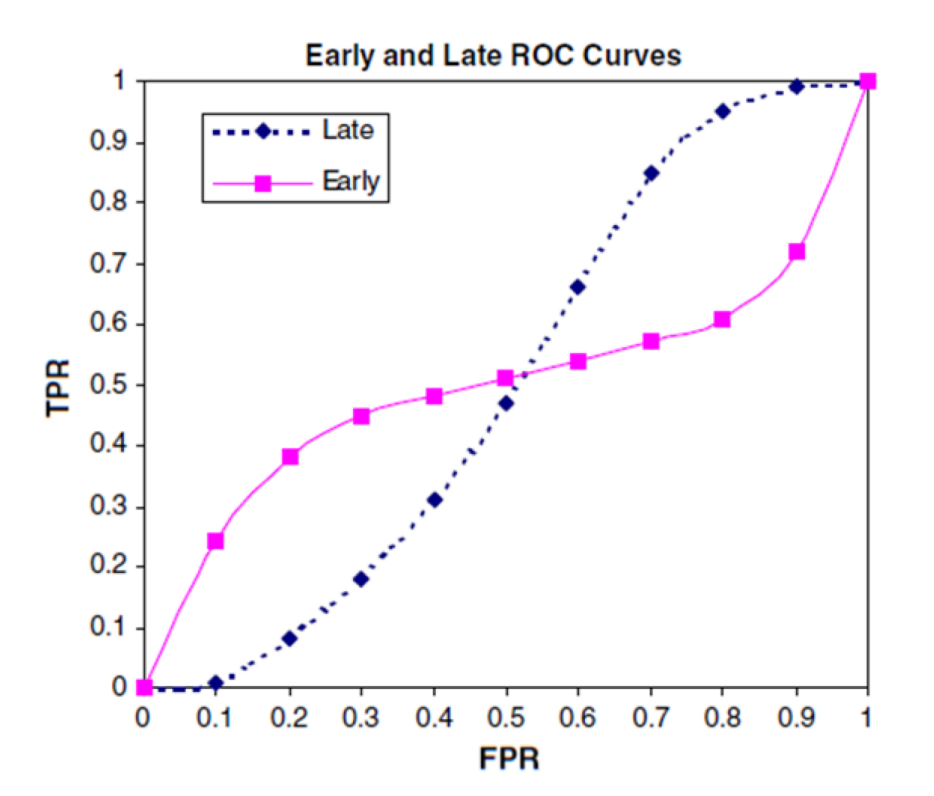


Fig. 10: **Early Enrichment Comparison**

Use of early enrichment values overcomes this deficit in AUC. **vROCS** reports enrichment percentages at the following values: 0.5%, 1% and 2%. The formulation of enrichment that is used in vROCS reports the ratio of true positive rates (the Y axis in an ROC plot) to the false positive rates of 0.5%, 1% and 2% (found on the X axis in an ROC plot). Thus "enrichment at 1%" is the fraction of actives seen along with the top 1% of known decoys (multiplied by 100). This removes the dependence on the ratio of actives and inactives and directly quantifies early enrichment. It also makes standard statistical analysis of error bars much simpler.

Enrichment values are quoted as a mean value 95% confidence limits. Bootstrapping the data produces a set of samples from which the mean and confidence levels are obtained. Repeating a run within a single **ROCS** session will always result in identical enrichments. However, enrichments may vary slightly between **ROCS** sessions because a new random number is supplied to the bootstrapping algorithm for each **ROCS** session.

3.4.4 p-Value

In statistical hypothesis testing, the p-value is the probability of obtaining a result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. The fact that p-values are based on this assumption is crucial to their correct interpretation ([Wikipedia-pValue], [Dallal-2001]).

In the **vROCS** analysis there are two runs being compared, a Base run (A) and a 'Compare to' run (B). These two runs use two different queries or methods (e.g. color force fields) to search the same active and decoy databases. We have a statistic, AUC (or % enrichment), one for each distribution (A & B). We would like to know whether AUC-A is statistically better than AUC-B otherwise we cannot say anything about the comparison of the methods. AUC-A and AUC-B alone are not enough to generate anything of statistical significance. To circumvent this we use a bootstrapping method which randomly selects a statistical sampling of the input molecules to repeatedly generate many AUCs.

Traditionally, the null hypothesis is that while the perceived results may be different (e.g. between AUC or % enrichment), the underlying processes are indistinguishable. However, since null-hypothesis testing predicts the likelihood of obtaining a given result if the null hypothesis is true, use of this null hypothesis wouldn't give any indication of whether method A or method B is better. To avoid this confusion OpenEye has used a modified null hypothesis. The null hypothesis, as implemented in vROCS, is that making a change to the query/method results in a better result (AUC or % enrichment) for run B than run A. Therefore we utilize a one-sided statistical test, not the usual two-sided test, based on the prior assumption that method B is superior to method A. The p-value is the probability that AUC-B > AUC-A and that this difference is due to differences between the methods/queries and not due to random chance alone.

If the null hypothesis holds true then we observe that AUC-B > AUC-A and the p-value tends towards 1.0. If the null hypothesis is incorrect then the p value tends towards 0.0 and the query/method used in run A (Base run) is statistically better than that used in run B (the 'Compare to' run). If the results for the two runs are indistinguishable and the result could be due to random chance then the p-value = 0.5.

The *plot below* illustrates these three p-value extremes. Each curve in the plot represents an example of comparing two ROCS runs. For each run bootstrapping produced a statistical sampling of the data from which the mean and 95% confidence limit values were calculated for AUC and % enrichments. The distribution of differences in AUC (or % enrichment) between the bootstrapped samples for the two runs can also be calculated and is plotted below.

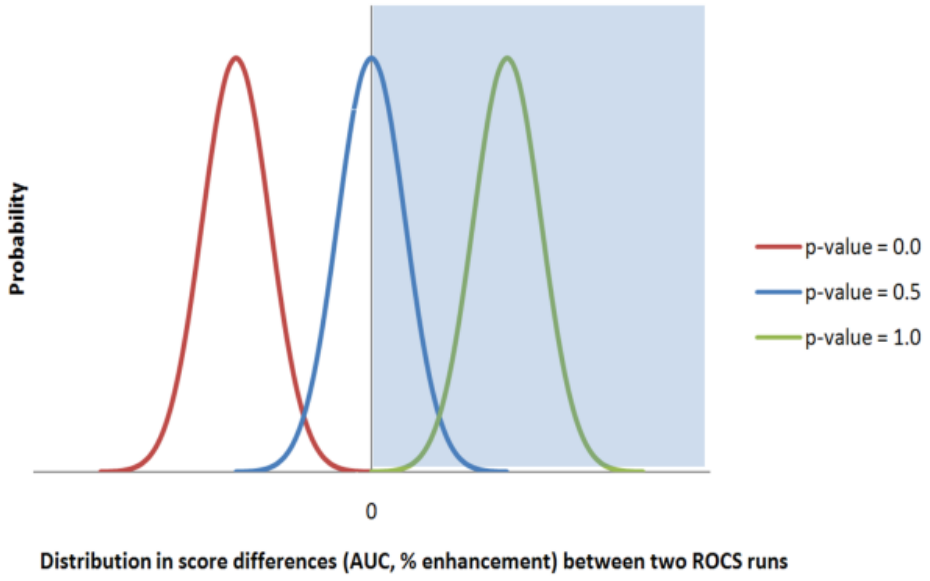


Fig. 11: **Plot to illustrate calculation of p-values**

In the case of p-value = 0.5 half of the distribution is positive and half of the distribution is negative. The p-value is calculated from the integral of the area under the curve from 0 to infinity (the part of the curve that falls within the

shaded area). In the case of p-value = 1.0 the difference between run B and run A is always positive and the entire curve is above 0 on the X-axis. The entire curve falls within the shaded area and so the integral is 1.0. In the case where the p-value = 0.0 the difference between run B and run A is always negative. None of the curve falls within the shaded area and so the integral is 0.0

When considering the results from two **ROCS** runs the p-values should be interpreted as follows. If the p-value tends towards 0.0 then the results for the Base run are better than the 'Compare to…' run (run A > run B). If the p-value = 0.5 then the results for the two runs are statistically indistinguishable. If the p-value tends towards 1.0 then the Base run is not better than the 'Compare to…' run or, in other words, the 'Compare to…' run is giving results better than the Base run (run B > run A).

Consider the example below for three different trypsin queries run against the same active and decoy databases. From the *ROC plot for the three trypsin queries*, we observe that run Trypsin1 has an AUC intermediate between those of Trypsin2 and Trypsin3.

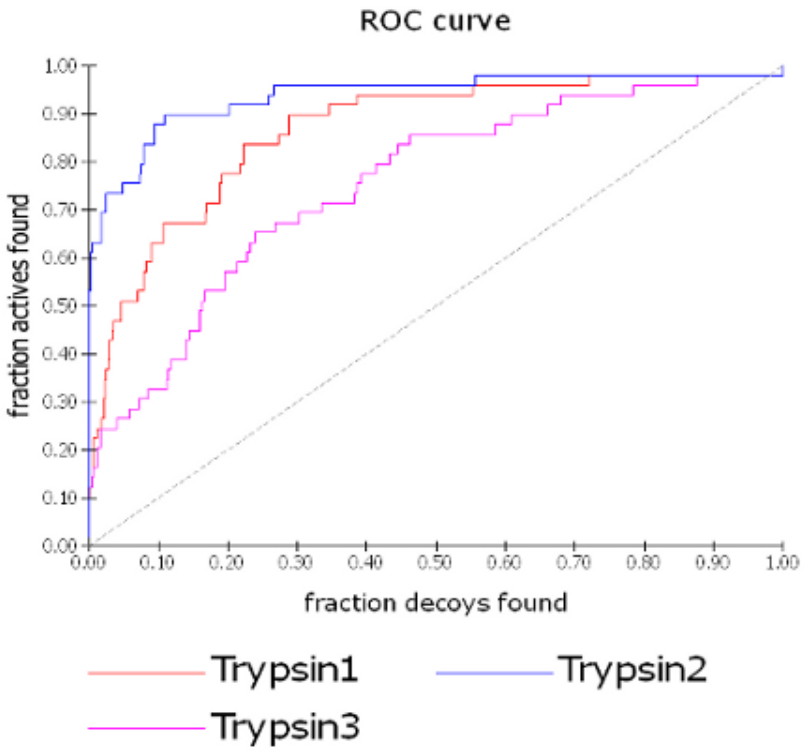
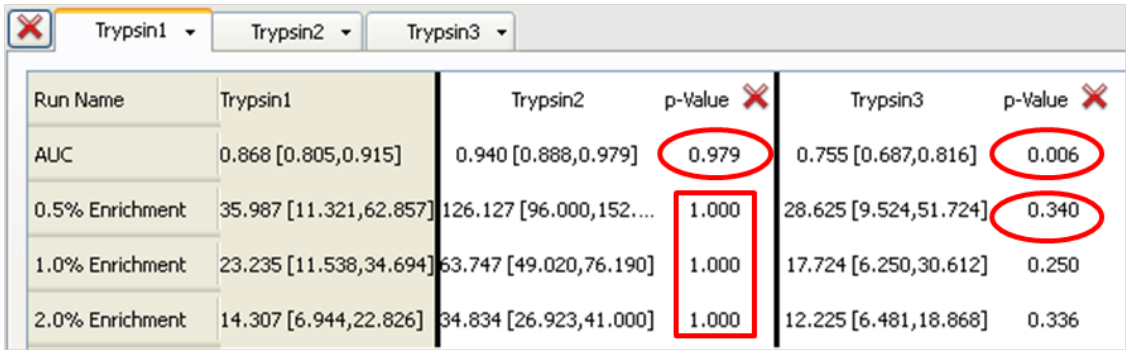


Fig. 12: **ROC plot for three trypsin queries**

Looking at *Table 1*, where Trypsin1 is the Base run (run A) and is compared to Trypsin2 and Trypsin3 (run B), we see that there is a p-value of 0.979 for Trypsin2. The Trypsin2 AUC of 0.940 mean value with 95% confidence limits of 0.888 and 0.979 is has very little overlap with Trypsin1 at 0.868 with 95% confidence limits of 0.805 and 0.915. The p-value = 0.979 suggests that Trypsin2 is producing superior results and these are due to differences between the queries, not to chance alone. The null hypothesis (run B > run A) holds true in this case. Note that in *Table 2*, where Trypsin2 is now the Base run, the p-value is reversed. In this case p-value = 0.021 suggests that Trypsin1 is producing inferior results and these are due to differences between the queries, not to chance alone. The null hypothesis (that the 'Compare to' run produces superior results to the Base run) can be rejected. Similarly, when comparing Trypsin3 to Trypsin1 in *Table 3*, the p-value of 0.006 suggests that Trypsin3 is producing inferior results and these are due to

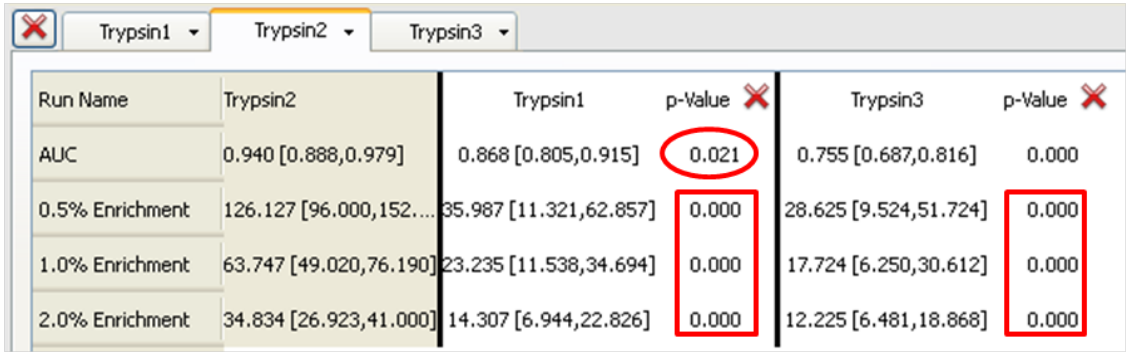
differences between the queries, not to chance alone. This is supported by our observations in the ROC plot (see *ROC plot for three trypsin queries*) where Trypsin3 clearly has the lowest AUC.



Trypsin1 | Trypsin2 | Trypsin3

Run Name	Trypsin1	Trypsin2	p-Value	Trypsin3	p-Value
AUC	0.868 [0.805,0.915]	0.940 [0.888,0.979]	0.979	0.755 [0.687,0.816]	0.006
0.5% Enrichment	35.987 [11.321,62.857]	126.127 [96.000,152....	1.000	28.625 [9.524,51.724]	0.340
1.0% Enrichment	23.235 [11.538,34.694]	63.747 [49.020,76.190]	1.000	17.724 [6.250,30.612]	0.250
2.0% Enrichment	14.307 [6.944,22.826]	34.834 [26.923,41.000]	1.000	12.225 [6.481,18.868]	0.336

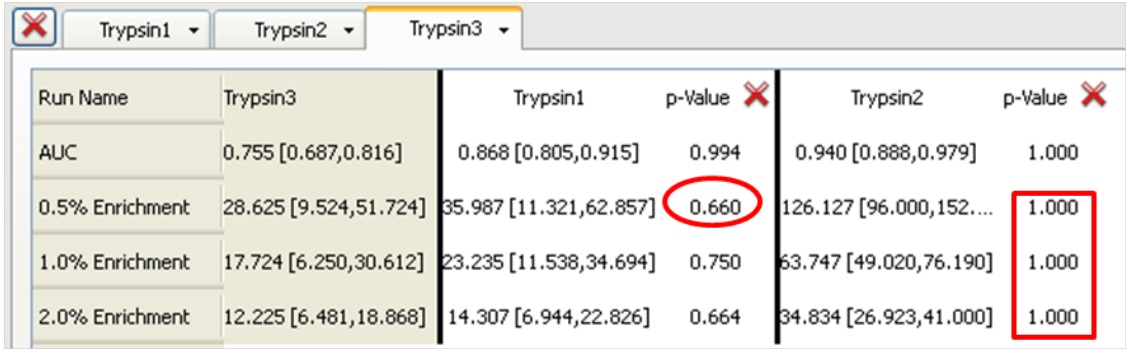
Fig. 13: **Table 1: Trypsin1 (Base) compared to Trypsin2 and Trypsin3**



Trypsin1 | Trypsin2 | Trypsin3

Run Name	Trypsin2	Trypsin1	p-Value	Trypsin3	p-Value
AUC	0.940 [0.888,0.979]	0.868 [0.805,0.915]	0.021	0.755 [0.687,0.816]	0.000
0.5% Enrichment	126.127 [96.000,152....	35.987 [11.321,62.857]	0.000	28.625 [9.524,51.724]	0.000
1.0% Enrichment	63.747 [49.020,76.190]	23.235 [11.538,34.694]	0.000	17.724 [6.250,30.612]	0.000
2.0% Enrichment	34.834 [26.923,41.000]	14.307 [6.944,22.826]	0.000	12.225 [6.481,18.868]	0.000

Fig. 14: **Table 2: Trypsin2 (Base) compared to Trypsin1 and Trypsin3**



Trypsin1 | Trypsin2 | Trypsin3

Run Name	Trypsin3	Trypsin1	p-Value	Trypsin2	p-Value
AUC	0.755 [0.687,0.816]	0.868 [0.805,0.915]	0.994	0.940 [0.888,0.979]	1.000
0.5% Enrichment	28.625 [9.524,51.724]	35.987 [11.321,62.857]	0.660	126.127 [96.000,152....	1.000
1.0% Enrichment	17.724 [6.250,30.612]	23.235 [11.538,34.694]	0.750	63.747 [49.020,76.190]	1.000
2.0% Enrichment	12.225 [6.481,18.868]	14.307 [6.944,22.826]	0.664	34.834 [26.923,41.000]	1.000

Fig. 15: **Table 3: Trypsin3 (Base) compared to Trypsin1 and Trypsin2**

Now consider the p-values for the 0.5%, 1% and 2% enrichments. Trypsin1 and Trypsin3 have similar enrichment levels. For example, at 0.5% enrichment Trypsin1 is 35.987 with 95% confidence levels of 11.321 and 62.857 while Trypsin3 is 28.625 with 95% confidence levels of 9.524 and 51.724. Each has an average enrichment that is well within the 95% confidence limits of the other. This is supported by p-values tending towards 0.5 i.e. 0.340 when comparing Trypsin3 to Trypsin1 (in *Table 1*) and 0.660 when comparing Trypsin1 to Trypsin3 (in *Table 2*) (the inverse around 0.5). From this we can conclude that the query Trypsin1 gives a slightly better 0.5% enrichment than does Trypsin3 (p-value 0.660, from *Table 1*) but that the differences may not be entirely statistically significant.

In the case of Trypsin2 the enrichments at all levels are much higher than Trypsin1 or Trypsin3 (e.g. 126.127 with 95% confidence limits of 96.000 and 152.381 for the 0.5% enrichment) with 95% confidence limits that do not overlap at all with those for Trypsin1 or Trypsin3. This results in p-values of 1.000 when either Trypsin1 or Typsin3 is the Base run (Tables 1 or 3) (i.e. Trypsin2 is the superior query and the null hypothesis holds true) or 0.000 when Trypsin2 is the Base run (in Table 2) (i.e. Trypsin1 and Trypsin3 are clearly inferior to Trypsin2 and the null hypothesis is rejected). These conclusions are also clearly visible in the *ROC plot*.

Repeating a run within a single ROCS session will always result in identical p-values for enrichments. However, since enrichments may vary slightly between ROCS sessions, when a new random number is supplied to the bootstrapping algorithm, there may be small differences in p-value for the same combination of runs if repeated in different ROCS sessions.

A typical cutoff for statistical significance of p-values is applied at the 5% (or 0.05) level. Thus, a p-value of 0.05 corresponds to a 5% chance of obtaining a result that extreme, given that the null hypothesis holds. A p-value of less than 0.05 (or greater than 0.95) would give good confidence that the selectivity you observe in your ROC plot is derived exclusively from differences between the two queries or methods and not a result of chance alone.

3.5 Saving ROCS data

The **vROCS** interface provides multiple tools for saving data. Data that can be saved includes:

1. The query file
2. The entire set of results obtained from a simple or validation **ROCS** run
3. Data and statistics from a validation run in `.csv` delimited file format
4. Screenshots of the 3D window illustrating the query and/or aligned hit molecules
5. Screenshots of the ROC or Score Histogram plots

Query file: A query that is built or modified in **vROCS** can be saved for future use. The default file type for saving a query is a **ROCS** Saved Query file with extension `.sq` or `.sq.gz`. This file type is not compatible with older versions of **ROCS**. Additionally, it contains information about the color force field used and therefore cannot be used with an alternative color force field.

There are multiple ways to save a query file from the vROCS interface.

- Firstly, using the File menu click on `File > Save Query...` or use the `Ctrl+S` shortcut keys. This is a 'Save As…' action and will always prompt for a filename and a directory in which to save it. This action is performed on the query currently selected in the Query list of the Run Set-up Inputs dialog.
- There is also a right-click option. When viewing the results panel right-clicking on the run name tab opens a *right-click menu* in which the second option is Save Query from 'Run Name'. This is a 'Save As…' action and operates specifically on the query associated with that run. Therefore, it allows the user to save an older version of a query that may have been subsequently modified by going back to activate the Results tab for the earlier run.

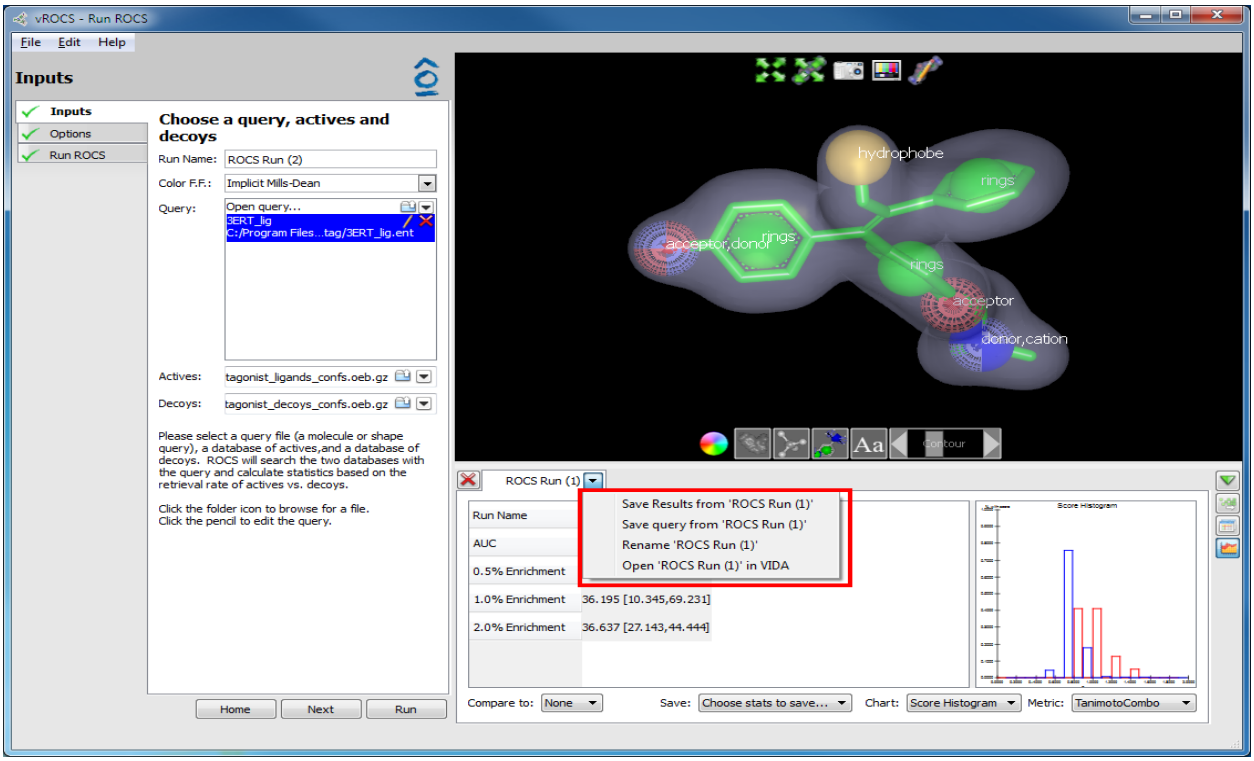


Fig. 16: **Right-click menu on results spreadsheet Run Name tab**

Option	Description
Save results from 'Run name'	Save the results for the active results set in a 3D structure and data file. The default file type is OE Binary (*.oeb, *.oeb.gz).
Save query from 'Run name'	Save the query for the active results set in a 3D shape query file (*.sq, *.sq.gz). The file contains information about both shape and color, as well as the color force field used to apply the color atoms. This is a Save As... action and will always prompt for a filename.
Rename 'Run name'	Rename the Results Name tab
Open 'Run name' in VIDA	Exports the structures and data for the active Run Name tab into VIDA. If VIDA is not already open a new session is opened. If VIDA is already in use the dataset is appended to the list of molecules already in the VIDA List Window. All the data is available to view in the VIDA spreadsheet.

Results: From the results spreadsheet for either the simple or validation run the user can right-click on the run name tab to open a *right-click menu* in which the first option is Save Results from 'Run Name'. This is a Save As... action and operates specifically on the data for the run associated with that spreadsheet. Having multiple spreadsheets available allows the user to save results from either the current run or an older run by selecting the appropriate run name tab. All the data points for the run are saved, not just the top 20 results visible in the spreadsheet. The sort order from the spreadsheet is not retained. The compounds in the saved file are sorted by the Rank By score selected during runs set-up. The results are saved in a variety of possible molecule file types suitable for opening in the **VIDA** spreadsheet

or other third party applications. The default is the OpenEye OE Binary file type with `.oeb` or `.oeb.gz` file extension. Since only the top 20 results are visible in the **vROCS** spreadsheet users are encouraged to save the results and use the **VIDA** spreadsheet, not the **vROCS** interface as the primary tool for analyzing results.

Statistics data: From the statistics panel in a validation run three different types of data - score data, plot data and spreadsheet - can be exported and saved in a comma delimited format (`.csv`), suitable for loading in text-based applications and other statistics packages. These options are selected from the `Choose stats to save...` drop down menu in the *statistics panel*.

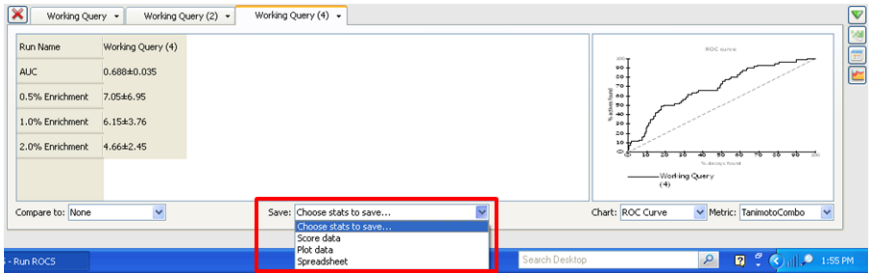


Fig. 17: **Choose stats to save drop-down in the statistics panel**

Option	Description
Score Data	Save a file containing the raw scores for the top scoring aligned conformer of each compound searched using each of the scoring functions of a validation run displayed. Data for **all** database compounds are exported.
Plot Data	Export the data points from either the ROC Curve or the Score Histogram. These are (x,y) datapoints from the curve or histogram created in **vROCS**, not raw data. Note that this export is context sensitive upon the plot that is currently displayed (i.e. when the ROC curve is displayed the plot data output will have (x,y) values to recreate the ROC curve and when the Score Histogram is displayed the data output will have (x,y) data for both actives and decoys to recreate a histogram).
Spreadsheet	Save the data displayed in the **vROCS** statistics panel spreadsheet (AUC and enrichment values with error bars). If a second run has been selected to compare against the currently active run then data for both runs and the associated p-values are exported in a single file.

ROC plot/Score Histogram: Exporting the data to recreate the ROC plot (or Score Histogram) to a `.csv` delimited file (described above) provides the opportunity to rebuild the plots in a third party graphing application and combine plots from different **ROCS** sessions. However, this can prove somewhat cumbersome and it is frequently useful to take a screenshot of the current plot (either of a single run or a comparison of multiple runs) for inclusion in a report, presentation or publication. To do this, right click on the plot and choose "Save Image...".

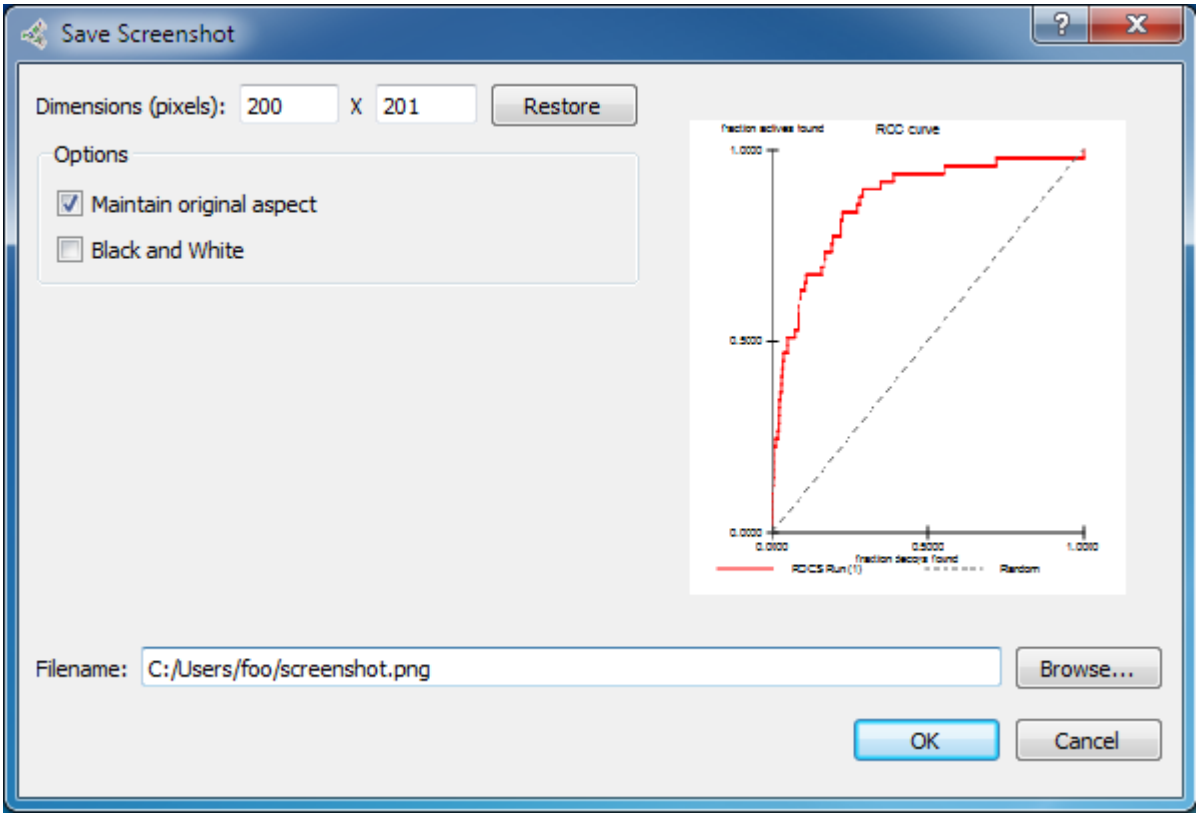


Fig. 18: **Save an image file of the ROC plot or score histogram**

3D window screenshot: A screenshot can be useful for insertion into presentations and publications. A *camera* icon at the top of the 3D window allows for taking a single click screenshot of the view in the 3D window. It is a WYSIWYG (what you see is what you get) screenshot of the 3D window with the exception that the surrounding buttons are not included. See the *figure* below.

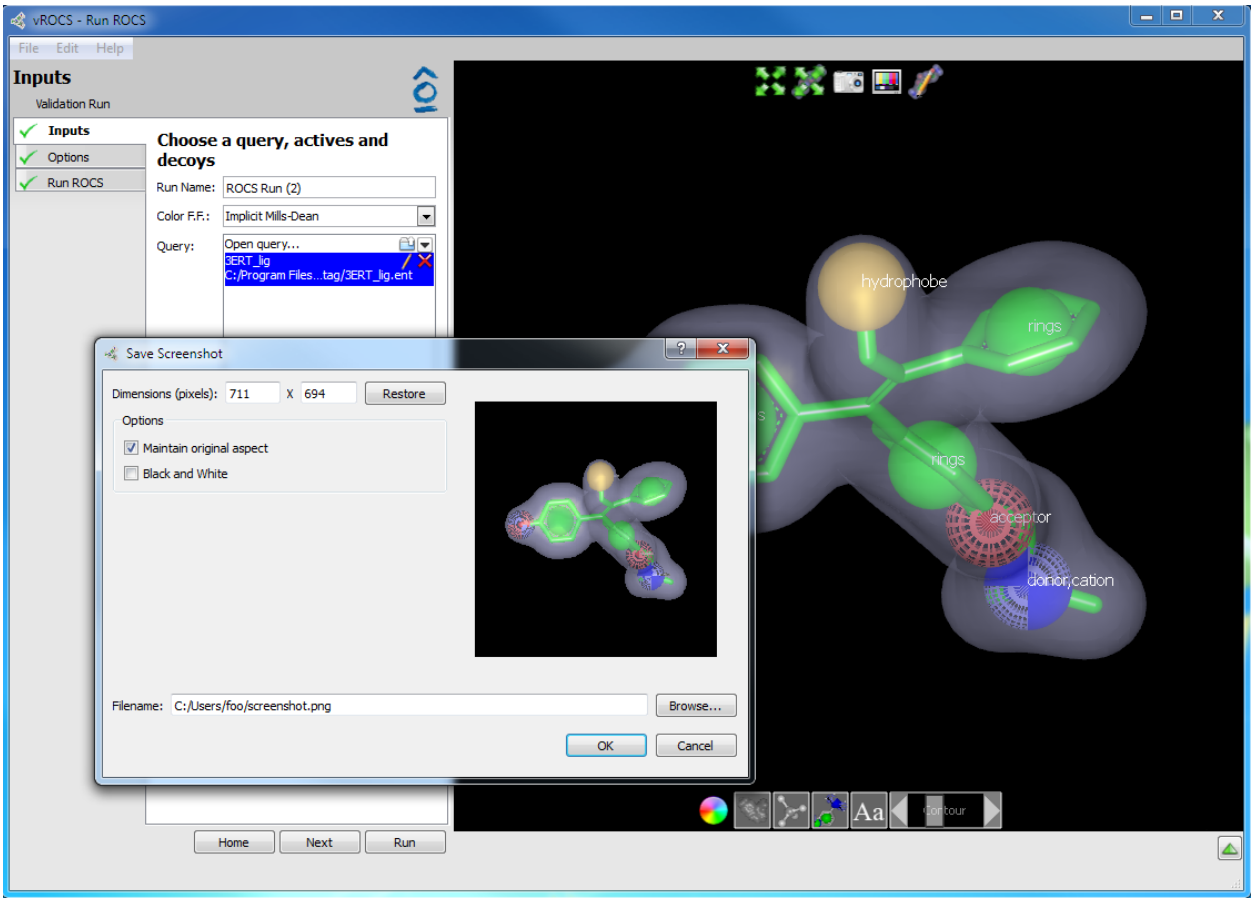


Fig. 19: **3D Window screenshot option**

3.6 ROCS shape query sources

ROCS is most commonly used to compare alignments of molecular shapes. However, a range of other shapes, e.g. molecular grids, form equally valid and useful alignment target queries, with the following provisos.

Grids are built without color atoms. The absence of color atoms in a query usually causes **ROCS** performance to be lower. For ligand shape queries adding color atoms has been shown to enhance **ROCS** performance with twice as much signal over random when color atoms are used, compared to shape alone. Without color the **ROCS** TanimotoCombo scores will also generally be lower (TanimotoCombo 0-1 instead of 0-2). Therefore, one should either add color atoms manually to a grid-based query (see section *Editing ROCS queries in vROCS*) or compare with the ShapeTanimoto score obtained from a ligand-shape query.

> **Note**
>
> Using DUD 1.0 with **ROCS** shape only the average AUC across the 38 cases is approximately 0.6. With shape + color the average AUC is around 0.73. Therefore, the delta over random for shape is 0.1 and for shape + color is 0.23. Hence, by this rather odd way of looking at it there is twice as much signal. - *P. Hawkins, OpenEye*

It is possible to add color points to grid shapes using the editing tools available in **vROCS**, described in the tutorial *Building and editing a query manually* and this can usefully guide the alignments. However, ligand shape with color generally provides superior results to using a grid-based query. Grids can be useful in cases where no suitable ligand

query exists.

There are several potential sources of grids:

- AFITT can produce a grid of electron density from crystallographic data. It is also possible to back-compute a grid of density for a crystallographic or docked ligand. This allows heavier atoms to contribute more to the grid than light ones, whereas shape grids are uniform.
- Spicoli will make grids from surfaces.
- **OEDocking** also produces a shape grid.
- Using the OEGrid toolkit you can read in any grid format and, using an ASCII interchange format, write it to an OE format that could be used by **vROCS**. This capability allows access to grids produced by third party applications. For example, DOCK ([DOCK]) uses scoring grids, GRID ([GRID]) makes grids and so on. All of these could be used to makes queries for **vROCS**, but their application and usefulness has not been thoroughly validated.

Recent research has been carried out at OpenEye to validate some tools currently under development ([Nicholls-2010]). These produce shapes that describe a protein binding pocket (using the same technology as Spicoli) for use as **ROCS** queries. Initial results show that shapes from sources other than pure ligands can be successfully used as useful **ROCS** queries and that adding color atoms is often useful to increase selectivity (and is never detrimental, to date), just as for ligand-based shape and color queries.

3.7 Editing ROCS queries in vROCS

In earlier versions of **ROCS** it was difficult to edit a query using the various command line utilities. The input to **ROCS** was generally required to be either a whole molecule query or a grid or shape query, although it was possible to load one or more molecules into a 3D builder and then modify or merge them into a super molecule. Having the **vROCS** graphical editor for **ROCS** provides the ability to move from a simple molecule with automatic color atom assignment or a grid (with no color atoms at all) to a position where the user can decide how the query is built. The **vROCS** graphical editor will facilitate this process by:

1. Reducing the time required.
2. Reducing the risk of errors.
3. Increasing the flexibility of the editing process by allowing a greater range of editing tasks to be accomplished.

This will have a knock-on effect that more complex and/or selective queries can be employed in **ROCS** and, in some cases, it is possible that higher quality results could be obtained. Additionally, it will facilitate the use of queries that are not directly molecule shape-based e.g. multiple fragments or grid-based queries.

There is a danger associated with the ability to edit the query and that is over-editing i.e. editing a query until it does not work. For this reason the validation run and its associated statistical analysis tools were included in **vROCS** (see section *Statistics metrics*). By providing the validation the user has the tools necessary to decide whether a complex new query is really better than simply using e.g. the x-ray ligand. This caveat should constantly be uppermost in the user's mind.

There are two methods for editing queries in **vROCS**. An automated wizard guides the user through one of a few predesigned paths for building a new query. Manual query building and editing is also available. Both functions are available from the Welcome interface.

3.7.1 Query Building Wizard

The *query building wizard* is designed to walk the user through building a query through one of the paths below:

1. SMILES
2. Ligand Model Builder

These are typically paths for which manual query building is less straightforward.

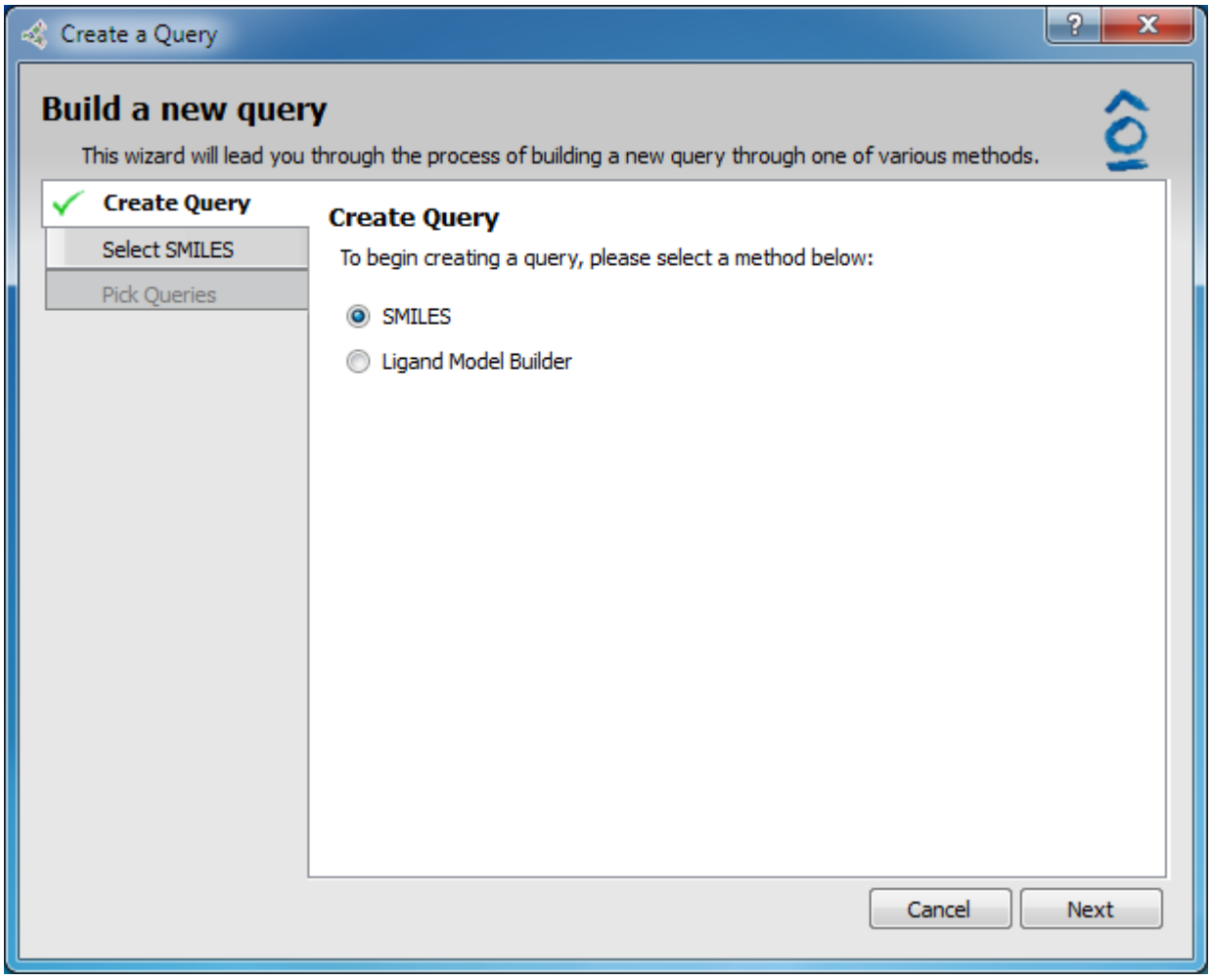


Fig. 20: **Query building wizard interface**

There are two ways to access the Wizard. Either select the Create a Query With a Wizard button on the **vROCS** Welcome page or select File > New Query… from the menu at any time during a session.

SMILES

The SMILES option produces up to 5 queries from an input SMILES string, calculating a reasonable 3D structure and conformations.

In the Create Query tab of the Query Wizard select the radio button SMILES and then click Next (as seen *above*).

The Select SMILES tab becomes active. This gives the option to type in a SMILES string or molecule name (systematic IUPAC or molecule name, i.e. aspirin). The molecule structure will be incrementally displayed as the SMILES string is entered.

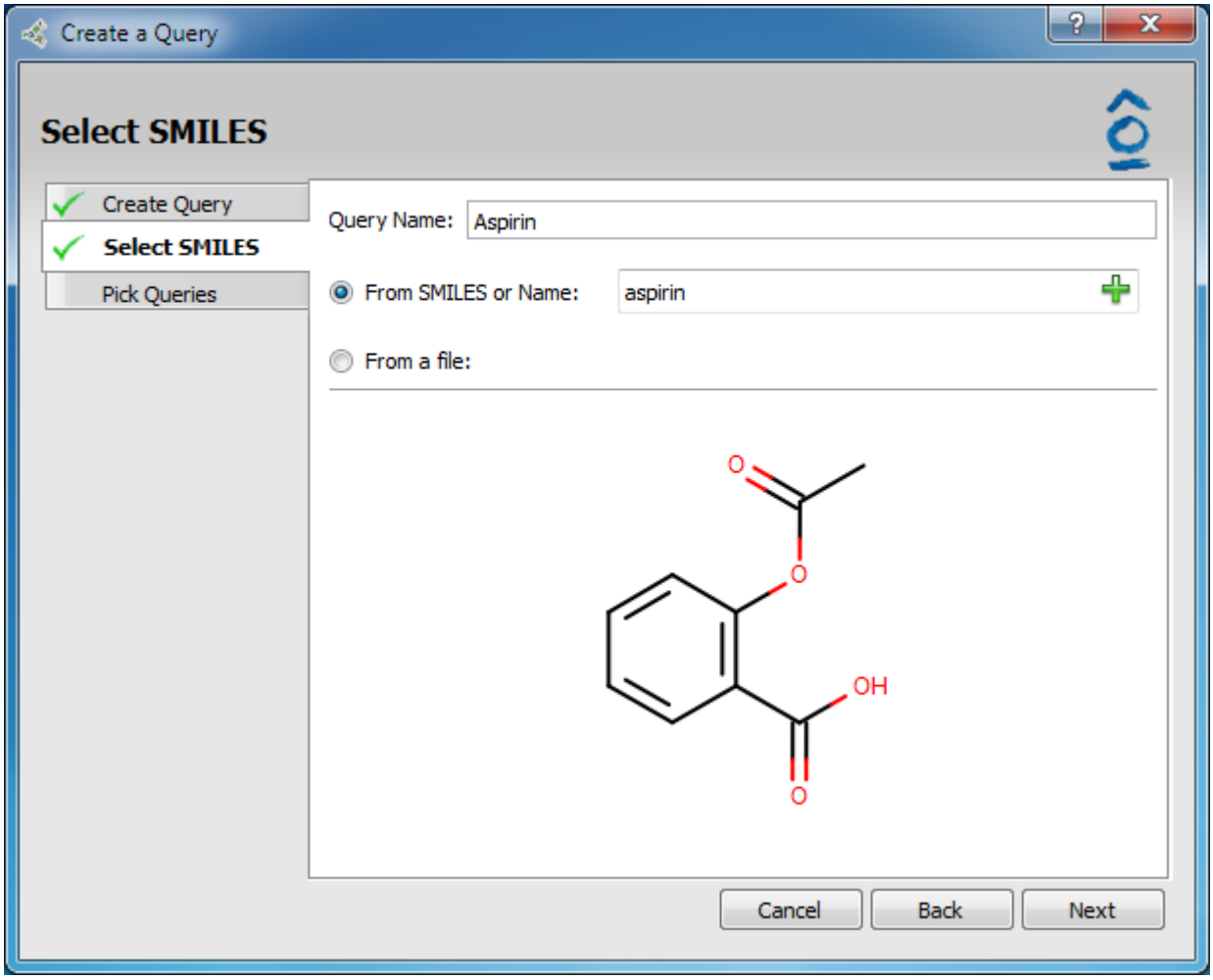


Fig. 21: **Select SMILES page**

Clicking on the green "+" icon in the SMILES entry field pops up a *Sketcher* in which the molecule can be sketched or the SMILES string or molecule name can be entered in the input field. When sketching is complete, clicking OK will close the Sketcher and update the structure displayed in the Select SMILES tab.

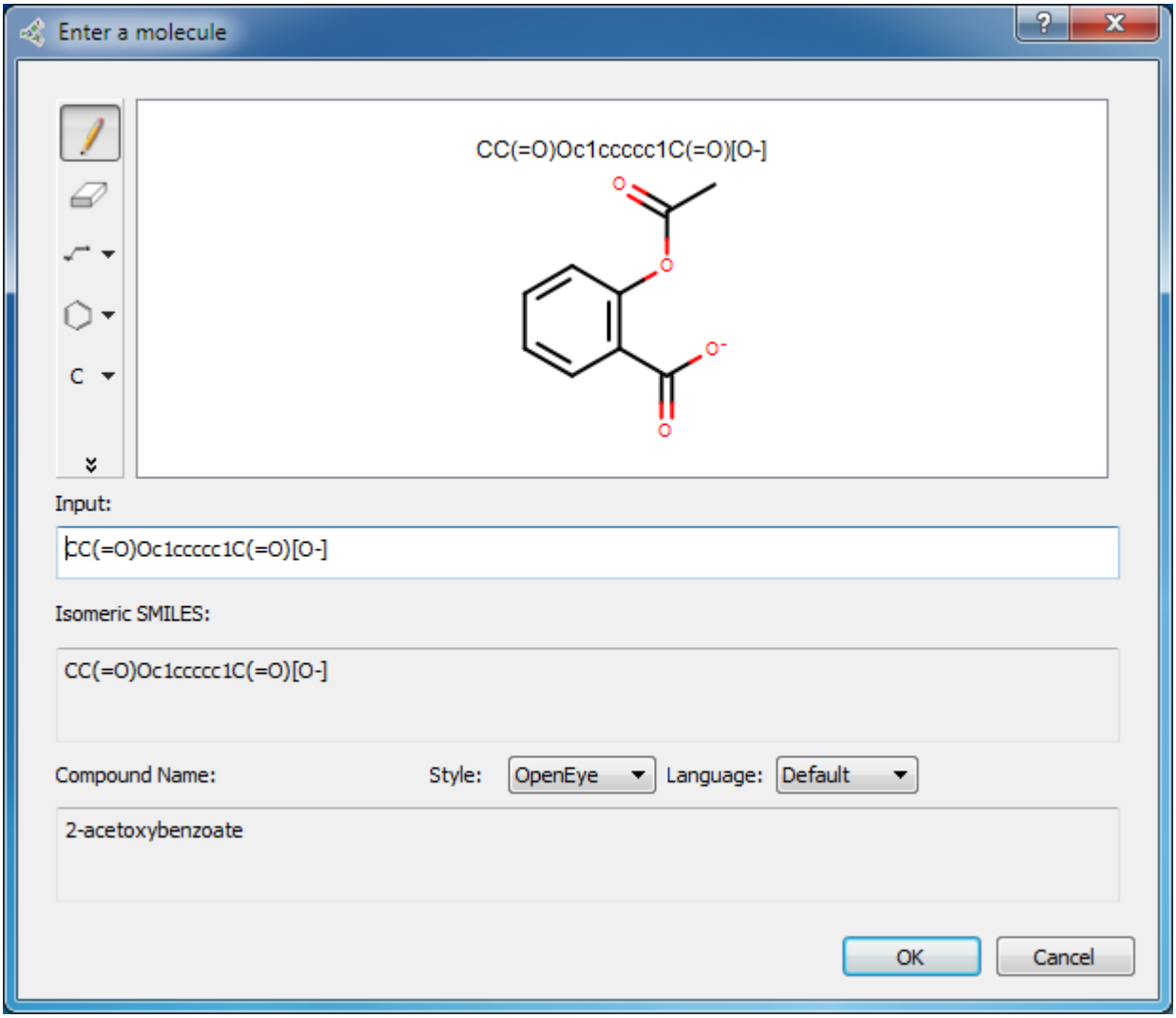


Fig. 22: **Picto sketcher**

Alternatively, a file of one or more molecules can be loaded and a SMILES string is displayed for each molecule in the file. Scrolling through the list of molecules will change the structure displayed. The highlighted structure in the list is the one that will be selected for the next step.

Because the query will be run through **OMEGA** to generate conformers, and **OMEGA** requires stereochemistry to be defined, chiral molecules with undefined stereo centers will require an additional step to specify the proper configuration at each stereo center. Atoms and bonds with undefined stereochemistry will appear highlighted in red (see *figure below*). Clicking on the highlighted atoms will cycle through the possible configurations.

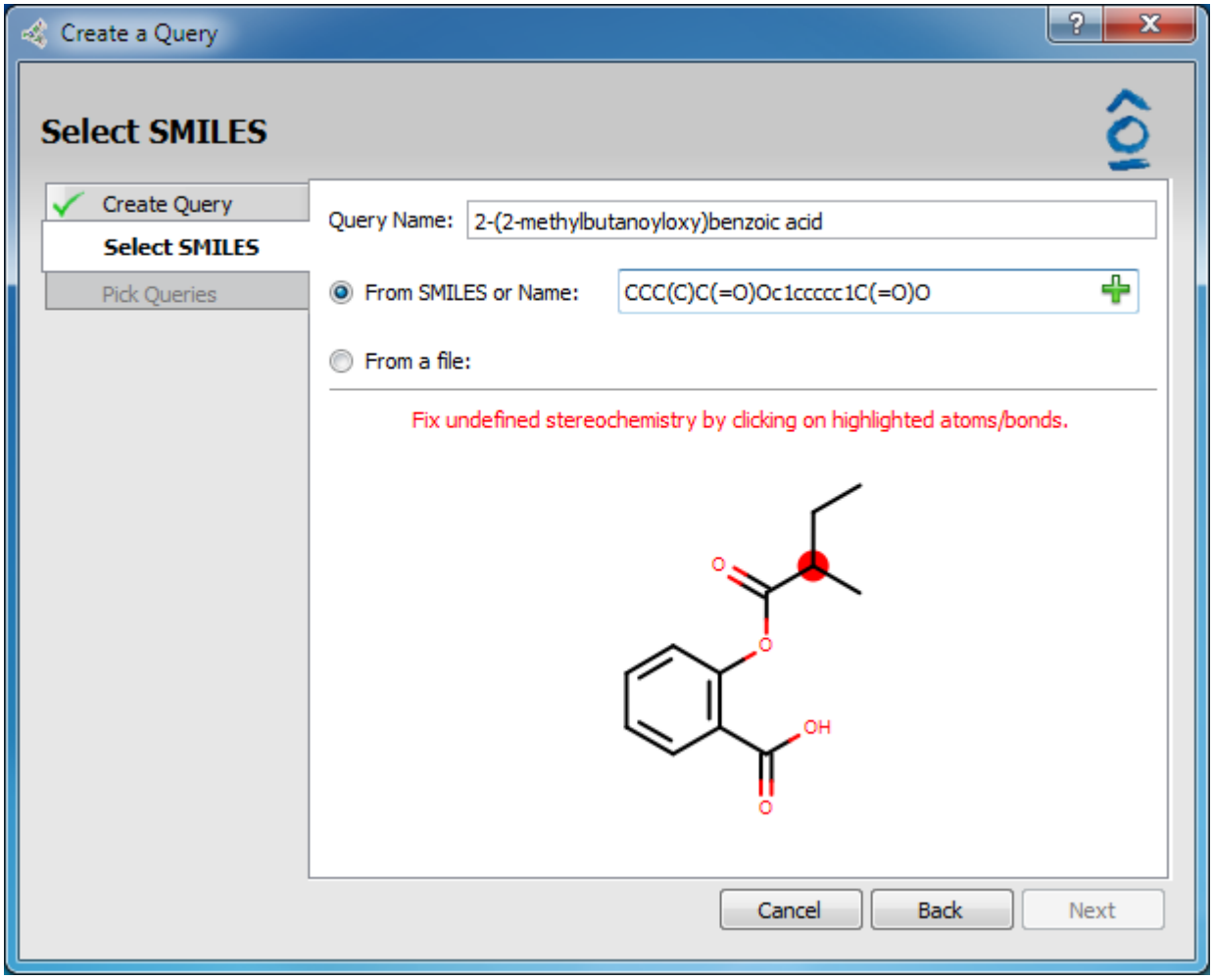


Fig. 23: **Molecule with Undefined Stereochemistry**

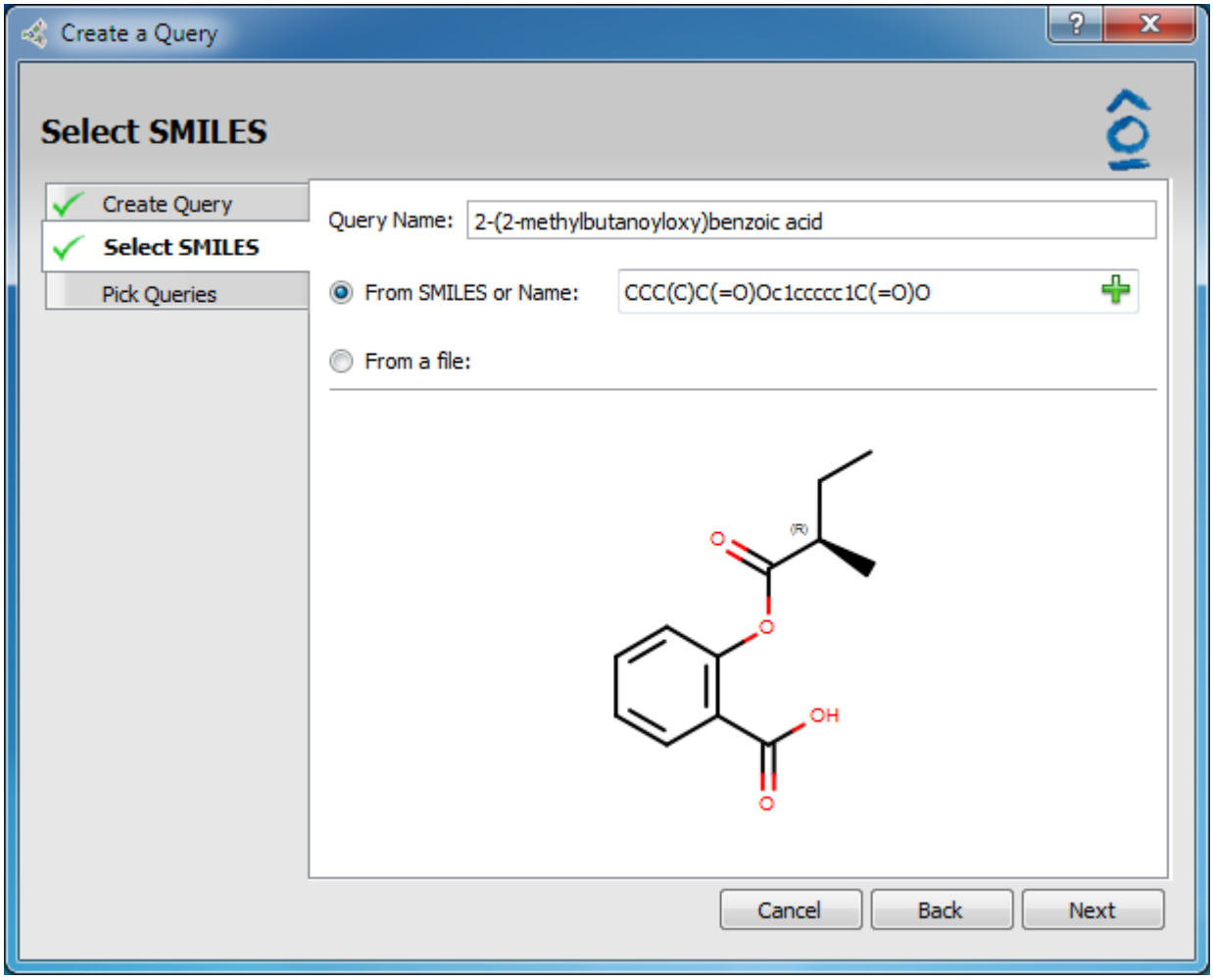


Fig. 24: **Molecule with Stereochemistry Defined**

Clicking Next activates the *Pick Queries page*. For the previously highlighted structure five (5) **OMEGA** lowest energy conformers are generated and listed. Conformer 1 is the lowest energy conformer. Scrolling through the list of conformer names using the up/down arrow keys or clicking on a specific conformer displays that structure in the 3D window above where it can be rotated or zoomed using the mouse. Multiple conformers can be chosen for import as **ROCS** queries into the main **vROCS** interface. The desired conformers are marked with a red check mark by double-clicking on their entries in the list.

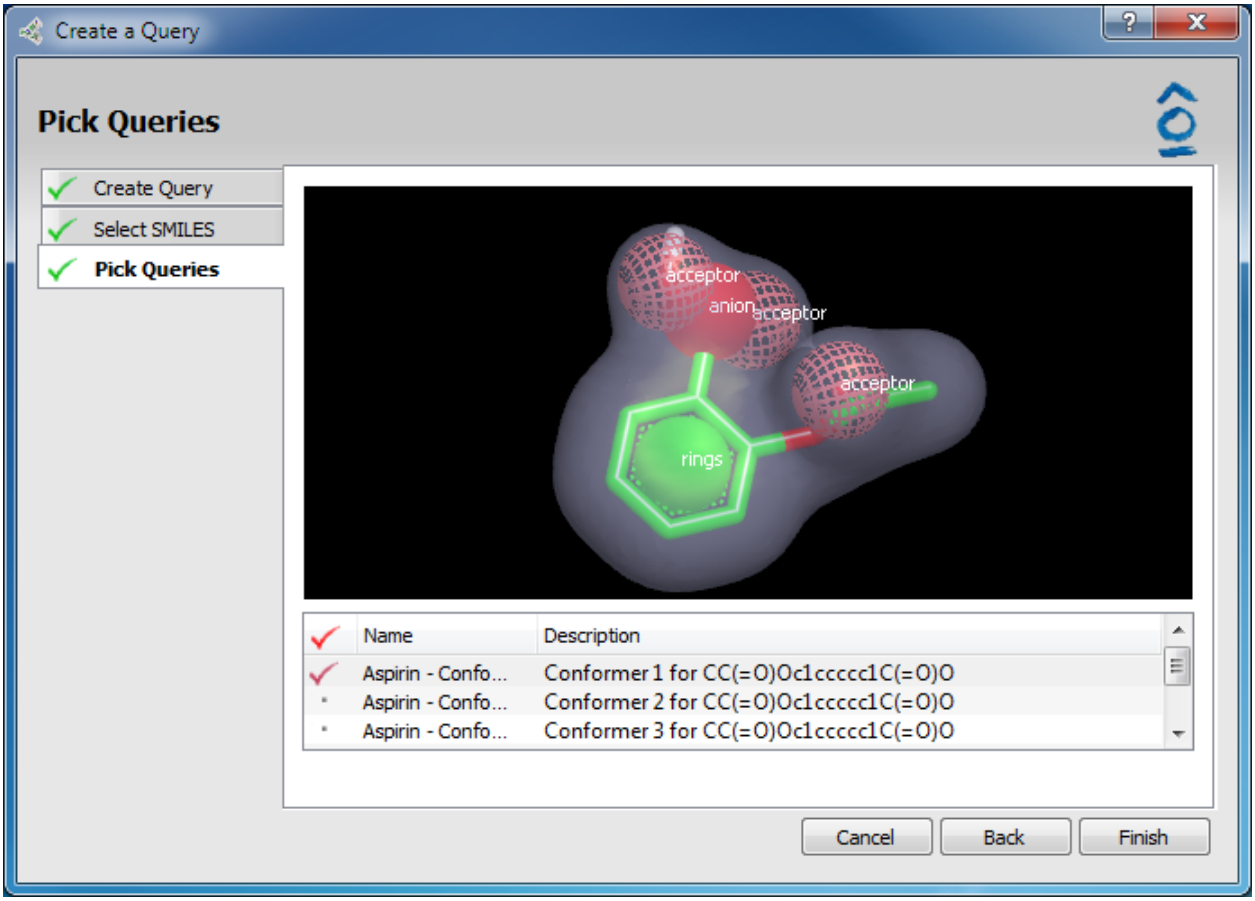


Fig. 25: **Pick queries page**

Click Finish to close the Wizard and import the selected conformers into **vROCS**. They will be listed in the Query list for simple or validation runs and the lowest energy conformer will be displayed in the 3D window.

Ligand Model Builder

If there are several known active ligands for a given project it can be desirable in **ROCS** to use a hypothesis query which is an alignment of more than one of the active ligands. This avoids losing potentially important information from one ligand that may not be present in others. The ligand model builder takes a set of pre-aligned ligands in the same Cartesian coordinate frame and carries out a rigid alignment in the same frame of reference. It produces hypothesis models for 1,2...n molecules (where n is selected by the user as the maximum number of molecules per model). The top scoring model(s), based on TanimotoCombo score, are returned. These are the model(s) that best represent the set of ligands as a whole. Since this is a rigid alignment no **OMEGA** conformers are generated and it is therefore important to use a 'reasonable' structure for each compound that represents a putative binding mode e.g. a set of docked ligands or a set of x-ray crystal structures.

Consider the following example. A set of 19 trypsin protein crystal structures are sequence aligned and the ligands extracted to give a set of 19 Cartesian aligned ligands. The user is interested in building 2 models, each containing up to 3 molecules. The ligand model builder builds hypothesis alignment models containing 1, 2 and 3 of these 19 ligands. It scores all the models against the set of 19 compounds and returns the top scoring 2 models. These may contain 1, 2 or 3 of the ligands and a 3-ligand model does not necessarily contain any of the ligands used in a 1- or 2-ligand model.

From the *Create Query dialog* of the Wizard select the radio-button option for Ligand Model Builder and click Next. This will activate the *Load Aligned Ligands page*.

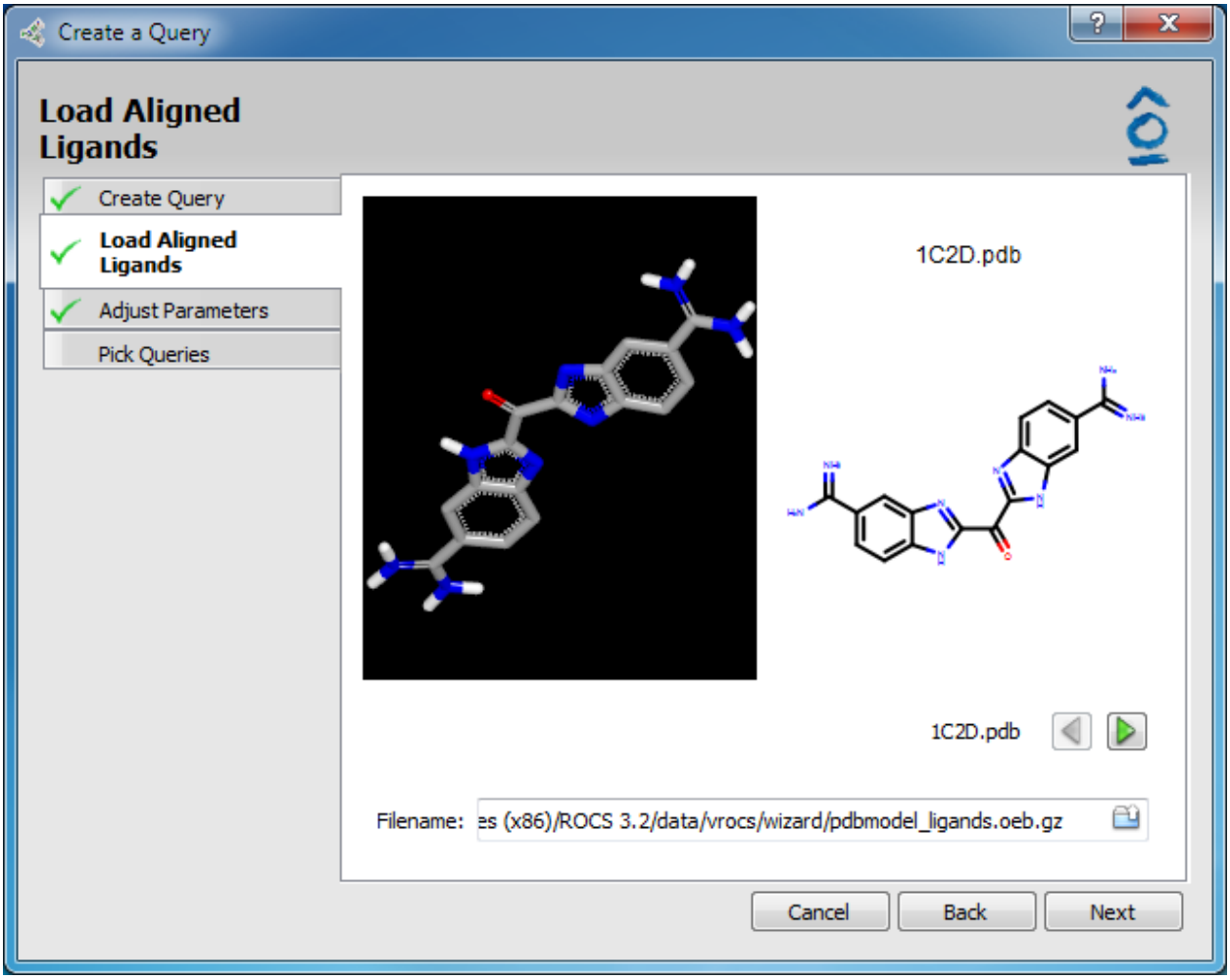


Fig. 26: **Load Aligned Ligands page**

The user is required to browse and select a file containing the aligned ligands (single conformer only). The input file type can be any format with 3D coordinates but must contain molecules. Other potential ROCS inputs (e.g. shape grids) cannot be supported in this workflow. Both 3D and 2D preview windows allow scrolling through the individual members of the file. The 3D window is interactive for zooming or rotating with the mouse. Click the Next button to proceed to the Adjust Parameters page.

All the required parameters are set by default so it is optional to make any additions or changes to fields in the *Adjust Parameters page*.

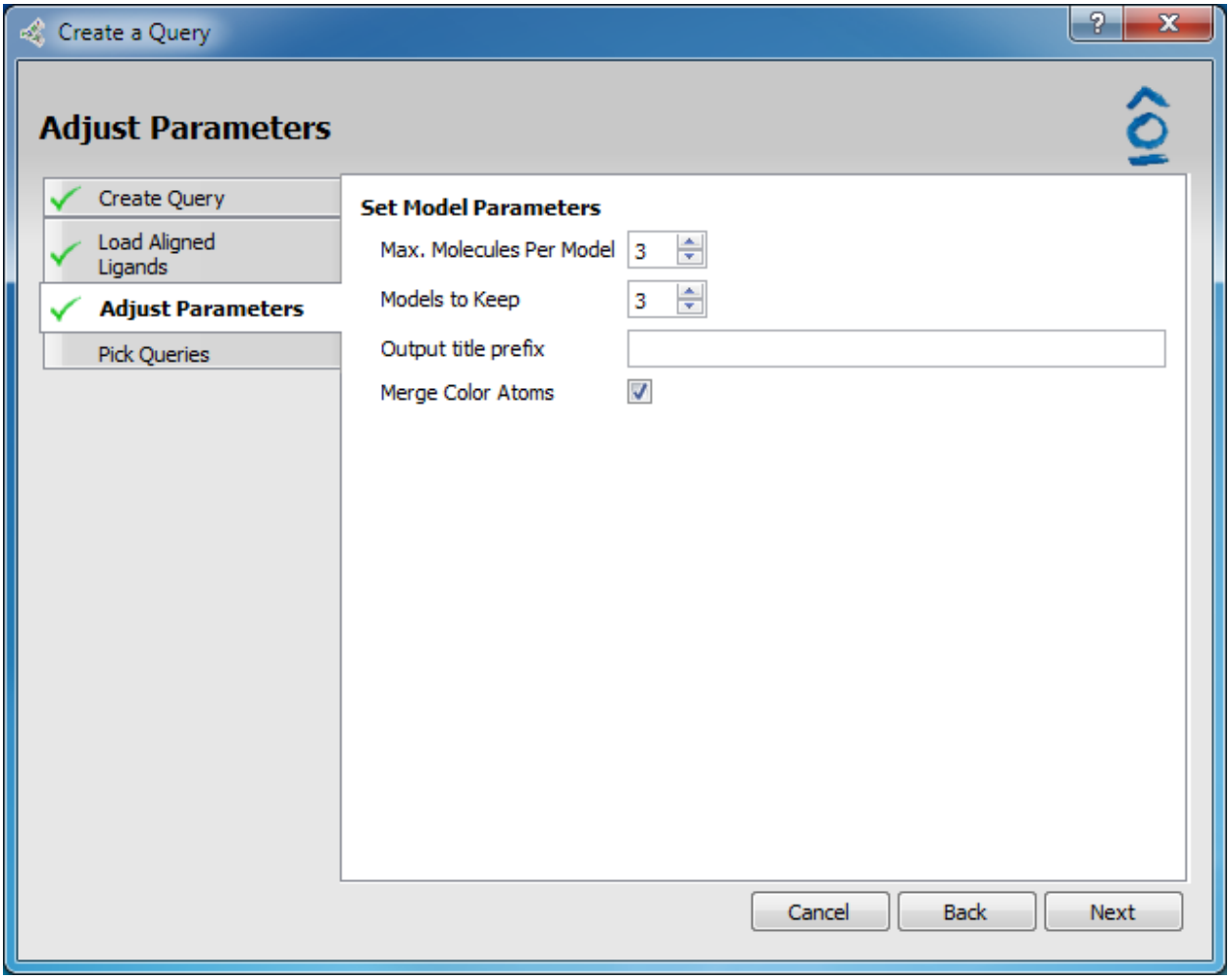


Fig. 27: **Adjust Parameters page**

Option	Description
Max. molecules per model	Models containing 1,2,...n molecules will be considered for an input value of n.
Models to keep	The number of best models to output, based on TanimotoCombo score.
Output title prefix	Optional naming prefix for the output models.
Merge color atoms	Optional merging of close color atoms of the same type in multi-molecule models.

The following options are available:

Max. molecules per model: This is the value n described above. For n=4 models containing 1, 2, 3 & 4 molecules will be considered. The input value for n cannot exceed the number of ligands in the input file. As n is increased the number of models considered will increase at a rate of the sum of the binomial coefficients. It is the sum of the binomial coefficient elements of the nth row of Pascal's triangle from 1 to n. (https://en.wikipedia.org/wiki/Pascal%27s_triangle.)

$k = \sum_{i=1}^{n} \frac{h!}{(h-n)!n!}$

k = total number of models considered;

h = no. of molecules in the pool;

n = max. molecules per model

Consider the trypsin example above where k=19. The binomial coefficients (for 1 to 19) are:

$$19 + 171 + 969 + 3,876 + 11,628 + 27,132 + 50,388 + 75,582 + 92,378 + \\ 92,378 + 75,582 + 50,388 + 27,132 + 11,628 + 3,876 + 969 + 171 + 19 + 1$$

If only models containing 1 molecule are considered then 19 models will need to be evaluated. If models containing up to 3 molecules are considered then 1159 models will be built and evaluated:

$$19 + 171 + 969 = 1159$$

This represents 19 one-molecule models, 171 two-molecule models and 969 three-molecule models. However, if models containing all 19 molecules were to be considered then:

$$k = 2^{19} - 1 = 524,287$$

k = total number of models considered

Thus, 524,287 models would be built and evaluated. Clearly this can become a cpu intensive and time consuming process so it is recommended to keep n low (<5) when the pool of molecules is large. This also avoids building overly complex models.

Models to keep: This is the number of top ranking models to output. By default this is set to 1 and therefore a single model will be produced. The output model is the one with the highest mean TanimotoCombo score across all the ligands. If this input value is set higher then more models will be output for visual evaluation and use as possible hypotheses. Increasing the number of models to keep has no effect on the run time; the same number of models are created and evaluated. It only changes the number of models retained in the output set.

Output title prefix: This optional field names the models produced. For a prefix 'model' the resulting models will be named 'model 1', 'model 2', etc. If no prefix is provided then the model name will be formulated by the names of the ligands which comprise the model. For example, '1GJ6_1QBO' is a model made from two ligands named '1GJ6' and '1QBO' in the input file.

Merge color atoms: This is an optional field. If two color atoms are overlaid then they will automatically be merged. However, color atoms are often close but not perfectly overlaid, for example, the two donor atoms highlighted *below*. Checking the Merge color atoms box will attempt to produce a single color atom which describes both. This simplifies the resulting model. Color atoms can also be manually deleted or merged later, if desired, as described in *Manual query building*.

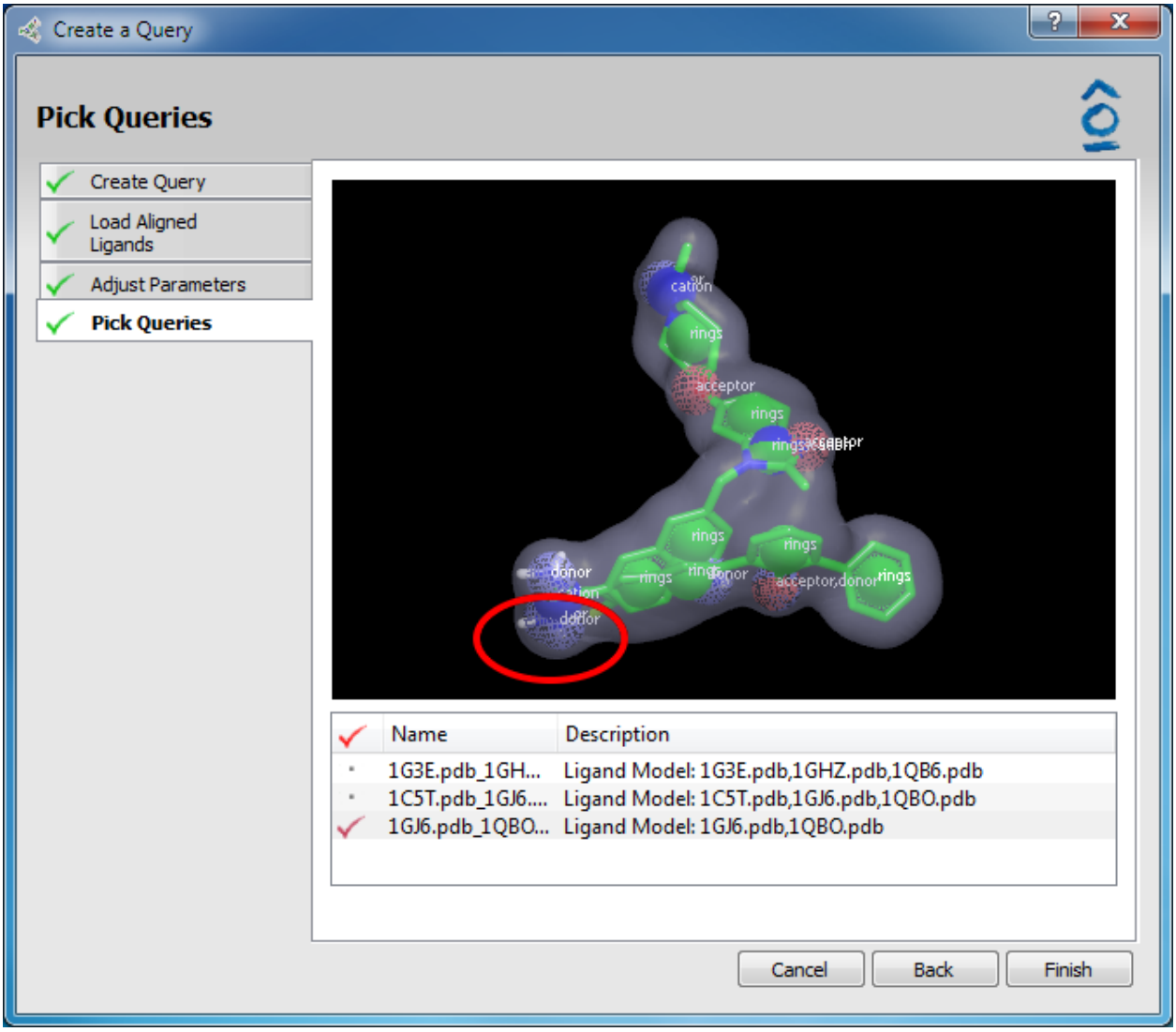


Fig. 28: **A two-ligand model produced by the model query builder**

When the desired parameters are set in the Adjust Parameters dialog click Next to begin the calculation. A progress tab will provide information on the progress of the model building. Models are built using the color force field currently selected in the user preferences (`Edit > Preferences > vROCS`, see *vROCS Preferences*).

When the calculation and evaluation is complete the models best fitting the parameters, based upon TanimotoCombo score, will be displayed in the *Pick Queries tab*. A 3D preview window shows the model which can be rotated and zoomed. The model currently on display is highlighted in the list below the 3D display. The list gives each model's name and a description made up of the names of the ligand(s) taken from the input ligand pool to build the model.

Clicking the Back button from the Pick Queries tab and changing any of the parameters will prompt with a warning that the previous results will be lost. To avoid losing potential queries it is advisable to first import any models of interest to the main **vROCS** interface before re-running the Wizard.

To import one or more models as queries to the main **vROCS** interface click to add a check mark in the column next to the model description. At least one check mark is required to activate the Finish button. The checked models will be imported into the Query list for a simple or validation **ROCS** run and each can be highlighted and viewed in the 3D window.

Queries built using the Ligand Model Builder can include many color features, some derived from each of the ligands used to build the query. This will have a couple of ramifications.

- **ROCS** search speeds are typically slower with more color atoms
- Unmatched color atoms give rise to scoring penalties and so scores using color might be lower than one might expect (e.g. TanimotoCombo score <1) compared to using a single ligand as a query. However, AUC and enrichment values will not be affected.

For these reasons it can frequently be useful to further edit the queries produced by the Ligand Model Builder and simplify them by removing some of the color features. Example candidates might be those that were not quite close enough to be merged by the Merge color atoms algorithm.

3.7.2 Manual Query Building

Manual query editing and building operations can be carried out on a new query (e.g. an imported ligand), a saved query file or the output from the Ligand Model Builder. To carry out the editing operations **vROCS** is required to be in the Edit Query mode.

Edit Query Mode: The Edit Query mode can be accessed in one of the following ways. Either click on the `Create or Edit a Query Manually` button in the Welcome screen or, at any time during an **vROCS** session, click on the `Edit Query` icon at the top of the 3D window. While in the editing mode the `Edit Query` button will be replaced by the `Done Editing` button and an *editing toolbar* will appear at the left of the 3D Window to select atoms, add color atoms, delete atoms or color atoms and merge color atoms.

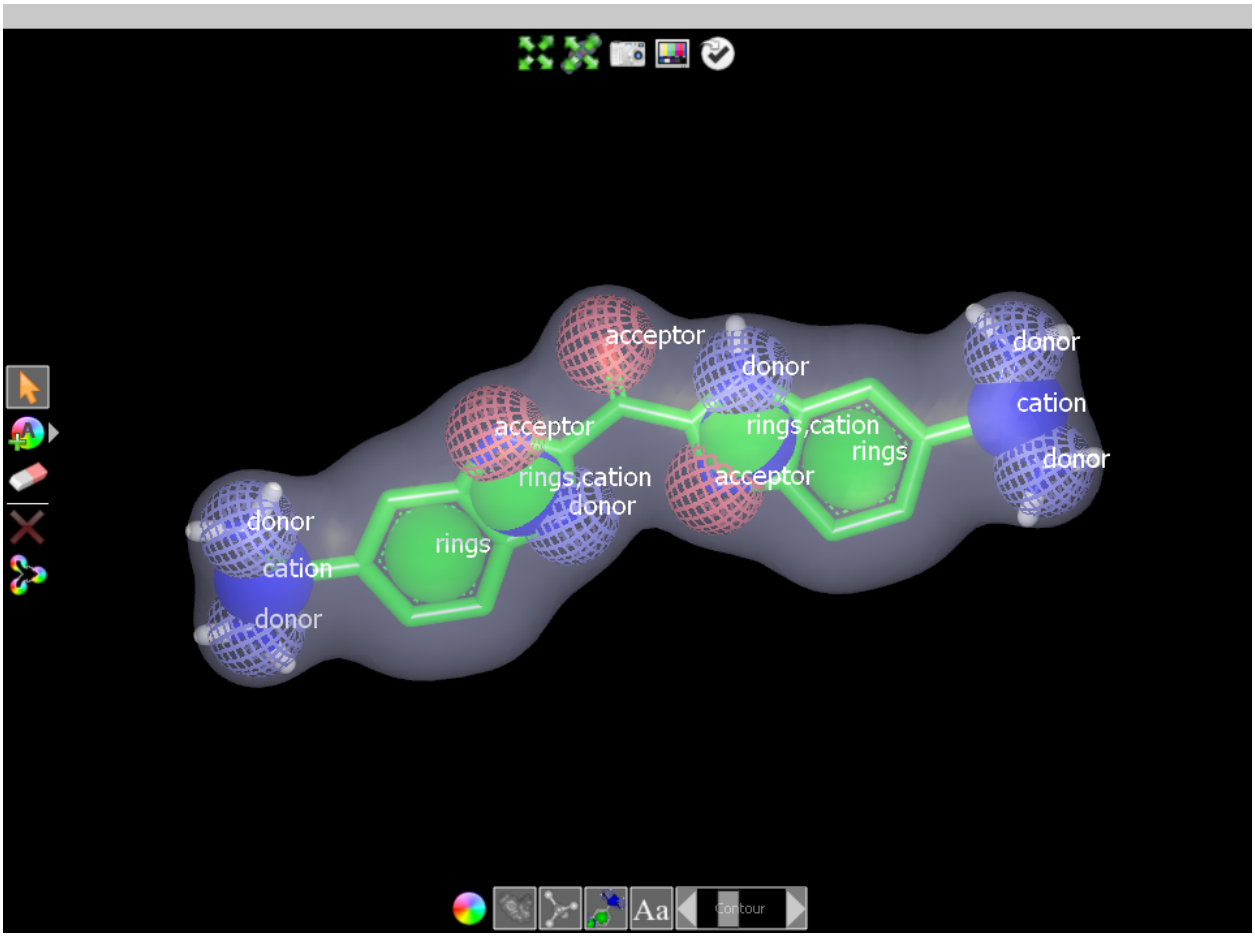


Fig. 29: **Editing a query**

Icon	Description
	Edit Query icon. Displayed in the *3D Visualization Window*. Open the Edit Query panel and add the query editing icons. This icon is replaced by a Done Editing icon while in editing mode.
	Done Editing icon. This returns the user to the 3D visualization window and hides the editing icons once query editing is complete. This icon is only visible in editing mode.
	Selection mode. Click on a shape or color atom to select it. Selected atoms will be highlighted in orange. CTRL click to select multiple shape atoms or color atoms. Only all color atoms or all shape atoms can be selected using CTRL click. Right-click and drag a box to select a portion of the query including both shape and color atoms.
	Add Color Atom mode. Click and select the desired color atom type from the pop-out menu. Choices for built-in color force fields are: 1) Acceptor (A), 2) Anion (An), 3) Cation (C), 4) Donor (D), 5) Hydrophobe (H) and 6) Rings (R). The letter on the icon indicates the currently active atom type. For other force field types, the list will be populated accordingly.
	Delete Atoms mode. Click on shape or color atoms to delete them from the query. Right-click and drag a box to delete all shape and color atoms currently visible within the box.
	Delete Selected Atoms action. Delete all currently selected atoms and/or color atoms. If there are no selected atoms, this button is disabled.
	Merge Color Atoms action. Merge multiple molecular fragments into a single query and merges color atoms of the same type to a single average representation. If color atoms are selected, those will be merged. If no color atoms are selected, all color atoms of the same type within 0.75 Angstroms of each other will be merged.

An *Edit Query panel* will display on the left hand side of the screen. This panel contains two areas. At the bottom is a Shape Inventory area. This lists all the open shape files that could be used in the query. These can be opened ligand or grid files, hits from an earlier **ROCS** run or other active queries. Items can be displayed in the 3D window by clicking their name. Molecules will be displayed as atom-colored sticks. At this stage they have no associated **ROCS** shape or color elements. Multiple items can be displayed together for comparison by clicking on the green visibility icon to the right of the item name on or off. Hovering over an item's name will display a 2D depiction of the structure, if it is a molecule file.

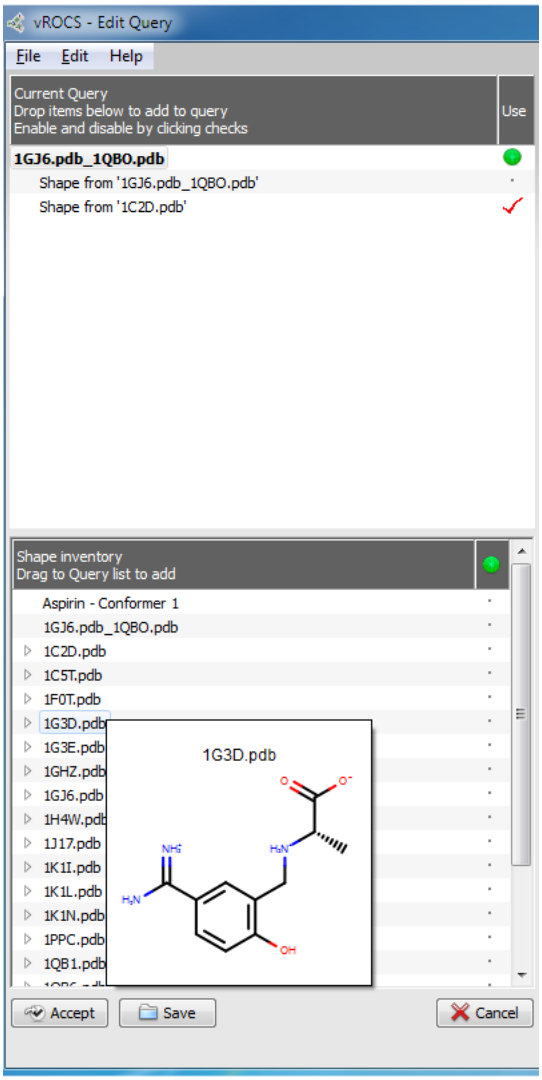


Fig. 30: **Edit Query panel**

At the top of the Edit Query panel is the Current Query area where components (e.g. atom components, color components, shape components) of the current working query are listed and can be selected or deselected. A selected item has a red check mark next to its name and will be used in the current query. Queries derived from molecules are displayed in the 3D Window as green colored sticks with atom-type colored heteroatoms. Color atoms are shown as colored spheres with labels. Grid shapes are shown as a gray, transparent surface (see *Editing a query*).

Any item can be moved from the Shape Inventory to the Current Query by dragging it from the bottom panel to the top panel or by right-clicking its name and selecting Add to Query. The other right-click options available in the Shape Inventory are to Delete the current item from the list or to Rename the item. A query can be made up of multiple query elements. To remove any element from the working query click on the red check mark to undisplay it or right-click on its name and select the option to Disable in Query. Other right-click options for the Current Query elements are Delete and Rename.

At any point the current working query can be saved by clicking on the `Accept` button below the Shape Inventory or by `File > Save Query...` These are both save as... actions and will always prompt for a filename and directory for the file to be saved. Note that the color force field parameters used to apply color atoms to a query are saved with the query and cannot be changed on re-opening a saved query. The color force field is set in the `Edit > Preferences` dialog (see *vROCS Preferences*) and can only be changed before any molecules or shapes are loaded into vROCS. When

editing is complete either click on the `Done Editing` icon at the top of the 3D Window or the `Use in ROCS` button below the Shape Inventory. Both will return to the display from which the editing mode was accessed (e.g. Welcome screen or Run set-up dialog).

The manual query editing tasks available via the vROCS interface are:

1. Merge two or more molecules/grids.
2. Delete color atom(s).
3. Delete shape atom(s).
4. Add color atom(s) from one or more selected atom(s).
5. Alter shape atom or color atom weighting.
6. Merge color atom(s).
7. Load grid.
8. Add color atom(s) to grids.

Each is described in more detail below.

Merge two or more molecules/grids: A query can be made up of molecules and/or grids from multiple sources/files. For example, two molecular fragments that describe ligand-protein interactions at different positions in the binding pocket, as shown below.

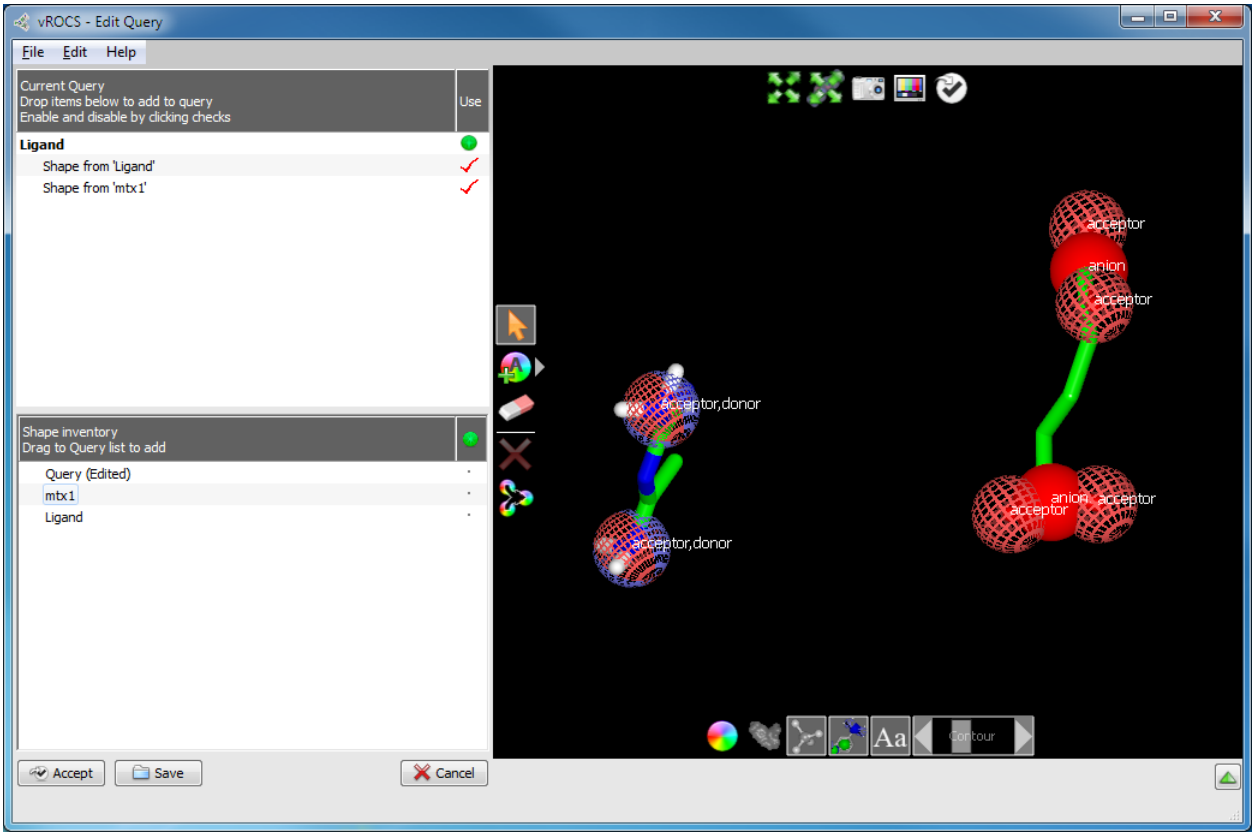


Fig. 31: **Query built from two fragments**

Open the files into the Shape Inventory using `File > Open` in the Edit Query mode and then drag them to the Current Query area of the Edit Query Panel. Each of these molecules/grids will be added to the current working query. They

should have a similar 3D coordinate frame so all portions of the query can be viewed in the 3D window at the same time. Saving the current query will save all constituent parts of the query together unless the red check mark indicating `Use in Current Query` is checked off. The combined query can be further edited using the options below.

Delete color atom(s): A ligand-based query automatically has color atoms assigned by **vROCS**. They are assigned using the currently selected color force field, as applied in the checkcff (command line) utility. The default color force field for **vROCS** is ImplicitMillsDean (see section *Color Force Field*) but this can be changed using the `Edit > Preferences` dialog (see *vROCS Preferences* for more details). Color atoms can also be manually placed on atoms or grids. It can be desirable to delete a color atom. For example, a hydroxyl oxygen atom would be assigned as both an H-bond donor and an H-bond acceptor by **vROCS** as highlighted in *Deleting a color atom*. However, knowledge of your active compounds and/or receptor cavity may lead you to believe that an acceptor is required at that position. The donor color atom can be deleted.

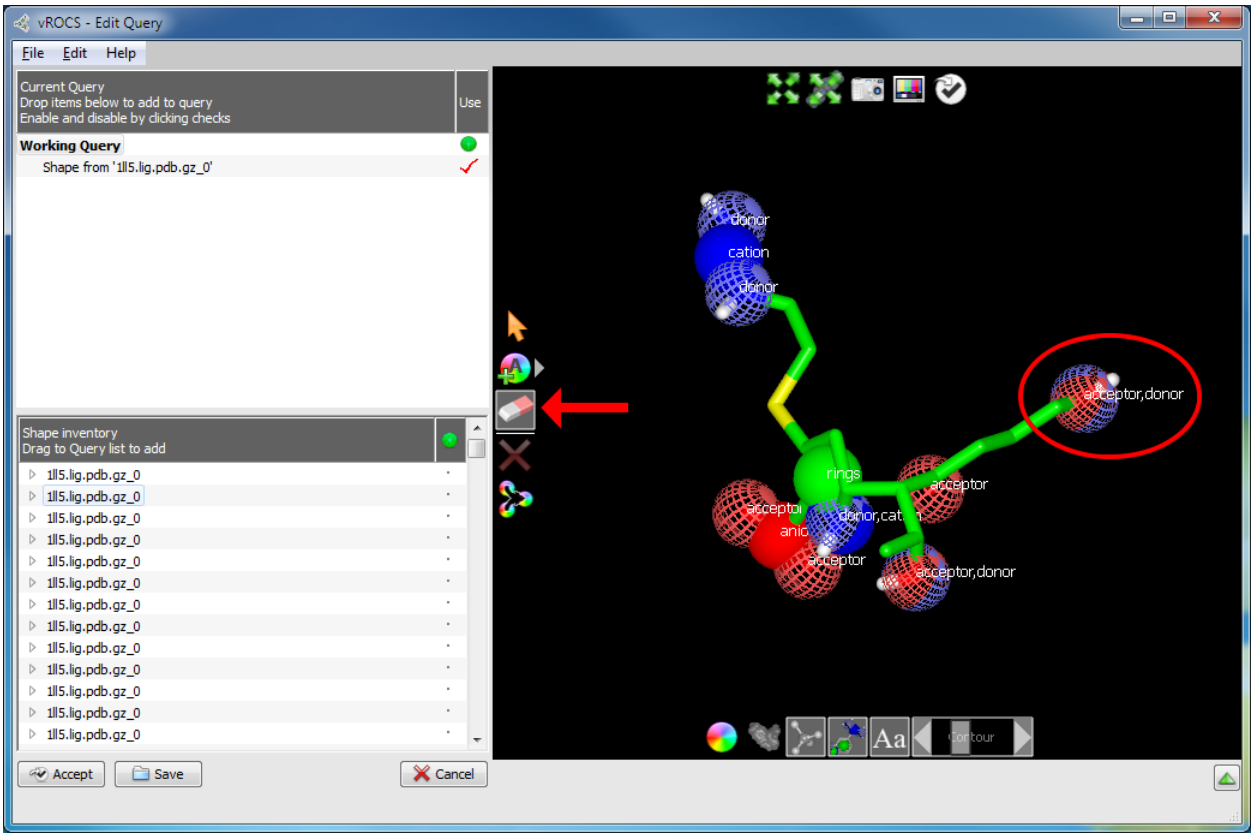


Fig. 32: **Deleting a color atom**

Click on the eraser icon highlighted in *Deleting a color atom*, above, to activate the `Delete Atoms` mode. A gray background to the button shows it has been selected. A single left click on the color atom (or atoms) that you wish to delete will remove that feature from the query. In the case of the combined donor/acceptor example clicking on the blue quadrants of the color atom will remove the donor feature and clicking on the red quadrants will remove the acceptor. Multiple color atoms can be deleted using sequential clicks.

At any point the `Edit > Undo` menu item can be used to replace a color atom (or shape atom) that is accidentally deleted. Since the Delete Atoms mode operates on both color and shape atoms it is often useful to hide (undisplay) the shape atoms and surface contour using the buttons at the bottom of the Edit Query 3D window.

To delete multiple adjacent color atoms right click and drag to draw a rectangular box around the features to be deleted with the Delete Atoms button highlighted. In this case it is desirable to display only the color atoms (hide the shape atoms and contour). This prevents other parts of the query from being deleted. The delete function only operates on

the part of the query that is visible.

An alternative method is to click on the lightning bolt icon to activate the `Selection` mode and select the color atom(s) to be deleted. The grid contour may have to be hidden before the color atom(s) can be selected. With the `Selection` mode active (icon highlighted gray) select the color atom by clicking on it. Once selected the color atom is highlighted in orange. To select multiple color atoms CTRL-click on each or right-click and draw a box around the group. Either click on the `Delete Selected Atoms` button (red 'X') or right-click on the highlighted color atom and select the `Delete` option. Using the `Selection` mode is useful when multiple color atoms are to be deleted.

Delete shape atom(s): It can be useful to delete part of the query molecule (shape atoms) if, for example, you are starting from a large query molecule but are carrying out vHTS to identify small ligands that are a good shape fit to only part of that query. Alternatively, some parts of the known active molecule may be important for binding and another portion requires less stringent alignment so a query built only from those fragments would be useful.

Click on the `Delete Atoms` (eraser) button to make it the active mode. A gray background to the button shows it has been selected. A single left click on the shape atom (or atoms) that you wish to delete will remove that feature from the query. Multiple shape atoms can be deleted using sequential clicks. The portion of the shape contour associated with the deleted shape atom(s) will also be deleted. To delete multiple adjacent shape atoms right click and drag to draw a rectangular box around the features to be deleted with the Delete Atoms mode button highlighted.

The `Delete Atoms` mode operates on both color and shape atoms and will delete a color atom preferentially over a shape atom. The delete function only operates on the part of the query that is visible. Therefore it is possible to delete shape atoms but leave behind color atoms with no associated shape in the query (as seen *below*) if the color atoms are hidden during the delete operation. The utility of queries of this nature has not been evaluated at OpenEye. It is possible that if a conformer is able to align with the color atom outside the shape during a **ROCS** alignment it may score higher than another conformer with an equally good alignment to the shape part of the query. In that case the conformer would receive a higher ColorTanimoto score although the ShapeTanimoto would be unchanged. However, this is probably only the case for color atoms close to the shape and these color atoms will not help to drive the optimization of the shape-based alignment. Therefore, for most queries the user should manually delete the color atoms as well as the shape atoms.

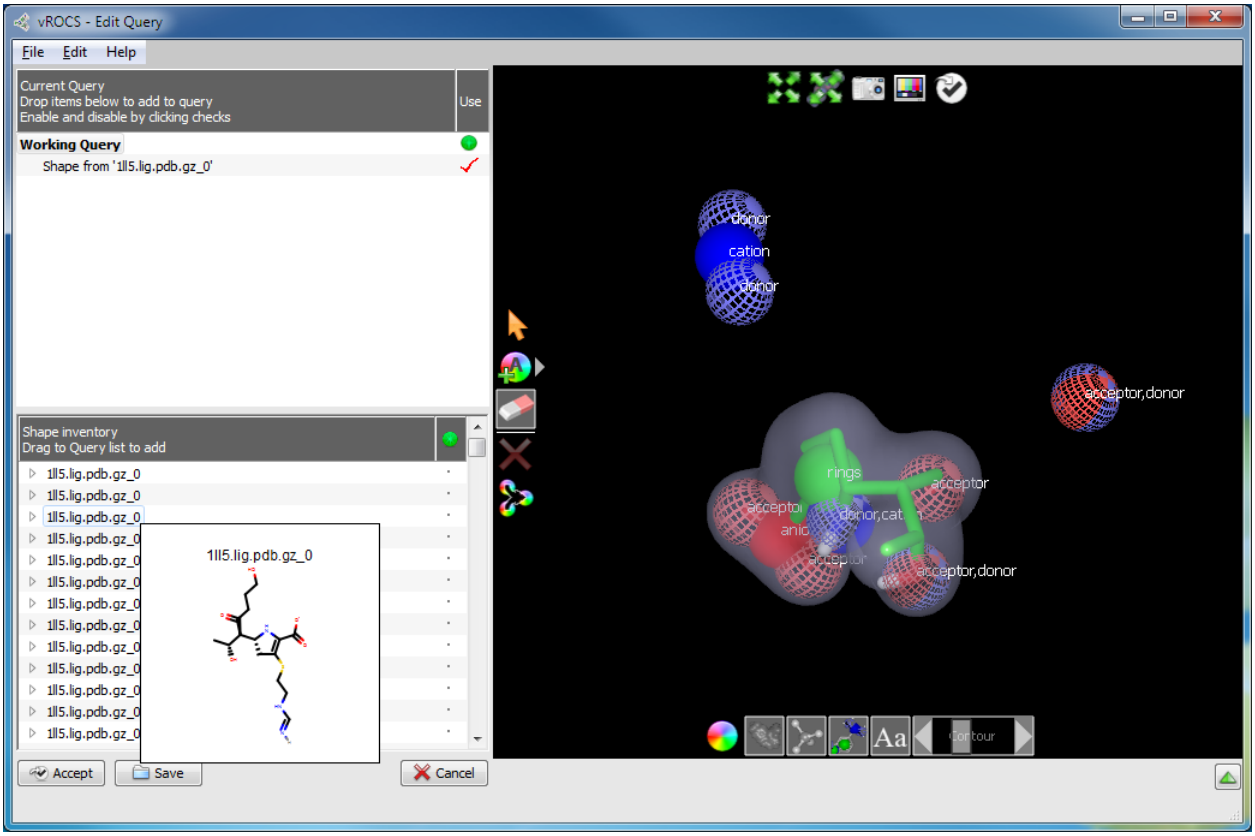


Fig. 33: **Color atoms remain after the shape atoms (and contour) have been deleted**

An alternative is to select the shape atom to be deleted using the `Selection` tool (lightning bolt icon). The grid contour and color atoms may have to be hidden before the shape atom can be selected. Once selected, the shape atom is highlighted in orange. Either click on the `Delete Selected Atoms` button (red 'X') or right-click on the highlighted color atom and select the `Delete` option.

Add color atom(s) to atoms: Color atoms can be added to shape atoms in a query molecule using the *Add Color Atoms tool*. This button contains a pop-out menu for selection of any color atom type for the current color force field. In the two built-in color force fields, these are acceptor (A), anion (An), cation (C), donor (D), hydrophobe (H) or rings (R). Click on any color atom type to create atoms for that color type. The button change will change the letter on the icon, as indicated in the list above, to indicate the active color atom type. In the Edit Query mode position the color atom by clicking on any atom of the ligand. Since color atoms can also be added to grids and surfaces the surface contour should be hidden before adding the color atoms. This is achieved using the Toggle Surface Contour button at the bottom of the 3D window.

If a color atom is added in error then the `File > Undo` option will remove it again. Alternatively, follow the instructions to Delete Color Atoms above.

An example of a situation where this might be useful is if the query ligand is a basic amine but you believe the N-atom is protonated at physiological pH to better interact with the protein. Deleting the Donor color atom from the N followed by adding a Cation color atom would effect this change. The overall shape of the query would be unchanged.

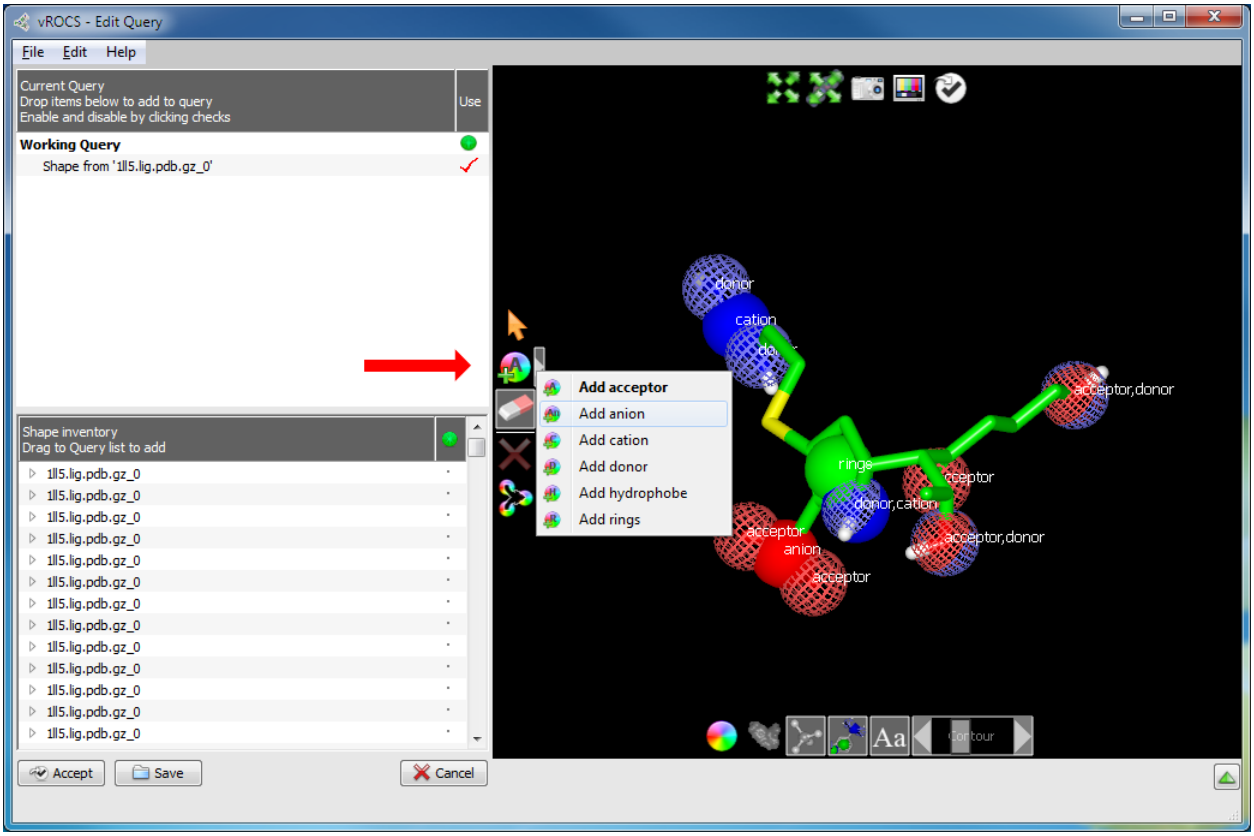


Fig. 34: **The Add Color Atoms button and sub-menu**

An alternative method is to select the shape atom where the color atom is to be added using the Selection mode. The grid contour may have to be hidden before the shape atom can be selected. Once selected the shape atom is highlighted in orange. Right-click on the highlighted atom. A pop-up menu provides the option to `Create color atom...` with a drop-down for selection of color atom type.

Move a color atom: Immediately after adding a color atom, `shift + click` in a new location to move the added color atom to the geometric center of its initial position and the new location. This allows a color atom to be positioned within a ring.

Alter shape atom or color atom weighting: Some molecular features or color interactions between a ligand and a protein are more important than others. This knowledge can be incorporated into the query to help drive the alignment and rank the hits. One way to weight only the important interactions is to delete those considered less important. However, this can cause loss of valuable information from the query. More preferred would be to increase the weighting of the important shape or color atoms while retaining the other features in the query.

Color atom weighting is achieved in a similar fashion to adding color atoms, described above. In the *image below*, the highlighted acceptor has been given a double weighting (acceptor x2) by adding a second acceptor feature to the hydroxyl. Select the Add Acceptor tool and click on the atom/color atom to be weighted. Similarly, use the Delete tool (eraser icon) to remove additional weighting from color atoms, exactly as previously described for deleting color atoms. The added color atoms will be listed as 'Shape from "User Added Features"' in the Current Query panel and can therefore also be deleted, hidden or selected for use in the current query (red check mark) in that panel. There is no limit to the increase in weighting that can be employed.

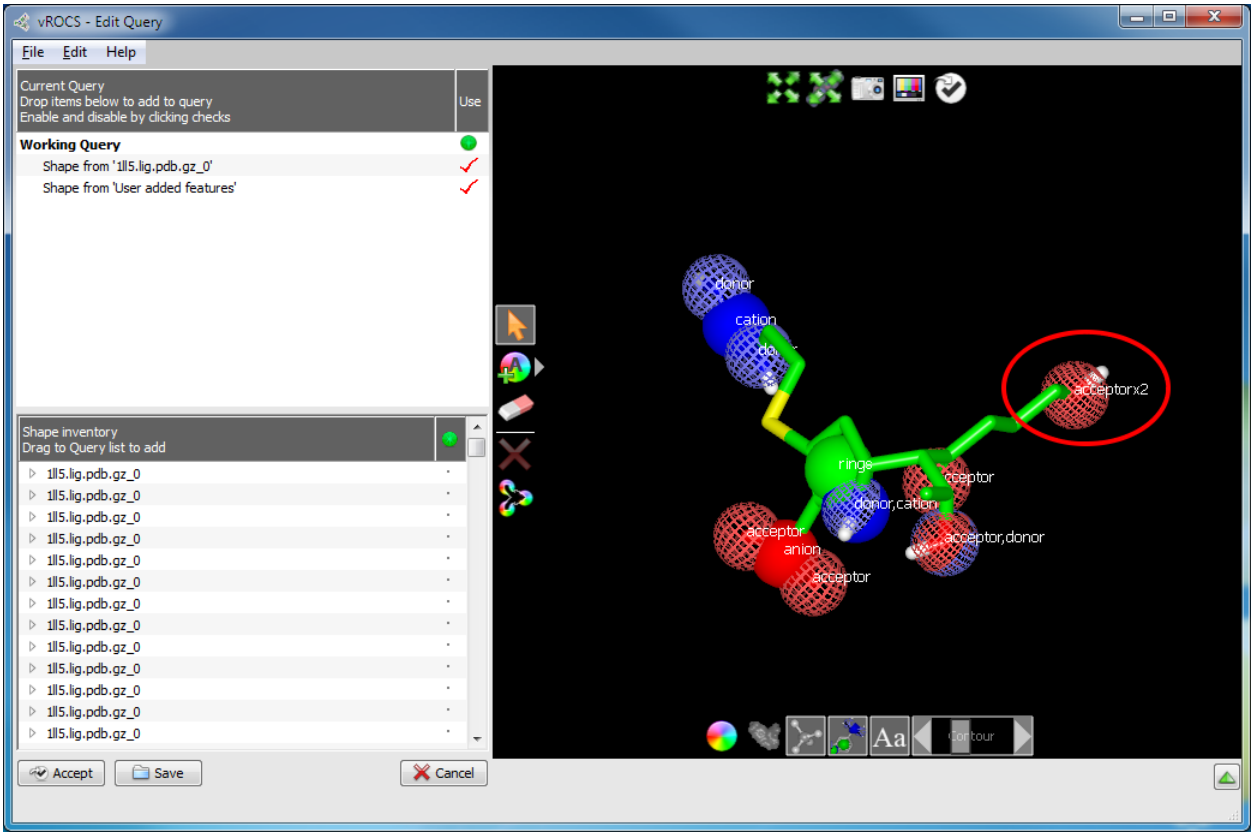
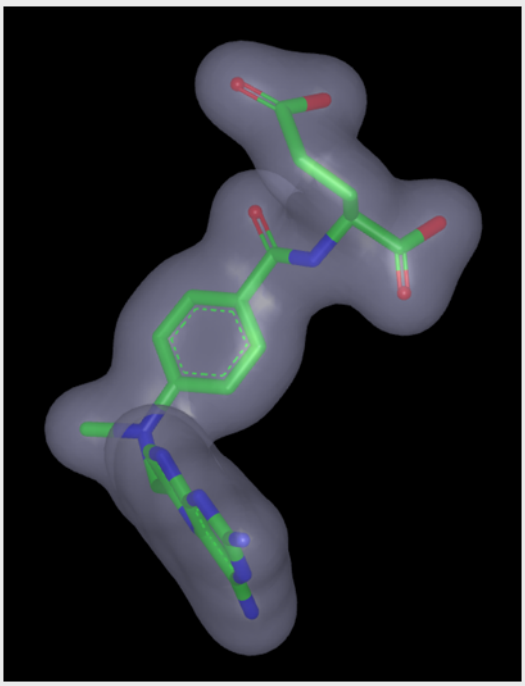
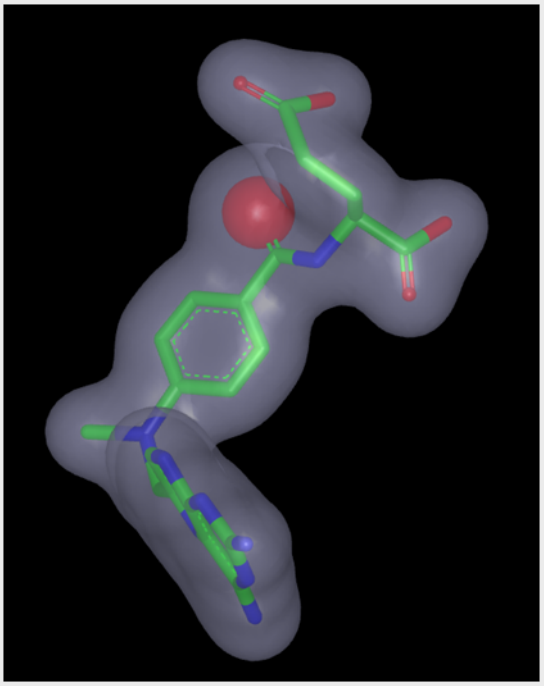


Fig. 35: **Weighting a color atom**

Shape atoms can be weighted by selecting the desired shape atom with the Selection tool so that it is highlighted in orange. Right-click on the highlighted atom and from the pop-up menu select the option to `Set shape atom strength...` The strength can be weighted from 1 (normal) to 5. This has the effect of placing up to 5 of those atoms at that position in space. The atom will be displayed larger to indicate its weighting and the shape contour will also be expanded, as seen for the carbonyl oxygen *below*.



Methotrexate query with all shape atom weightings set to 1 (default)



Methotrexate query with the carbonyl shape atom weighting set to 5

Merge color atoms: There are two cases where it may be useful to merge multiple color atoms together into a single color atom. The first occurs when a query is composed from multiple aligned molecules, resulting in color atoms of the same type lying very close to one another in space. The second case occurs when a molecule gives rise to multiple color atoms where a single color atom might better represent what the user would like to match. For example, a carboxylate group will by default be represented by two acceptor color atoms, but in some cases a single acceptor color atom, located midway between the two oxygen atoms, might be preferable. Likewise, a bicyclic ring system could be represented by a single ring color atom instead of the default two. Color atoms of the same type may be merged into a single color atom, located at the geometric centroid of the original color atoms, using the Merge Color Atoms tool.

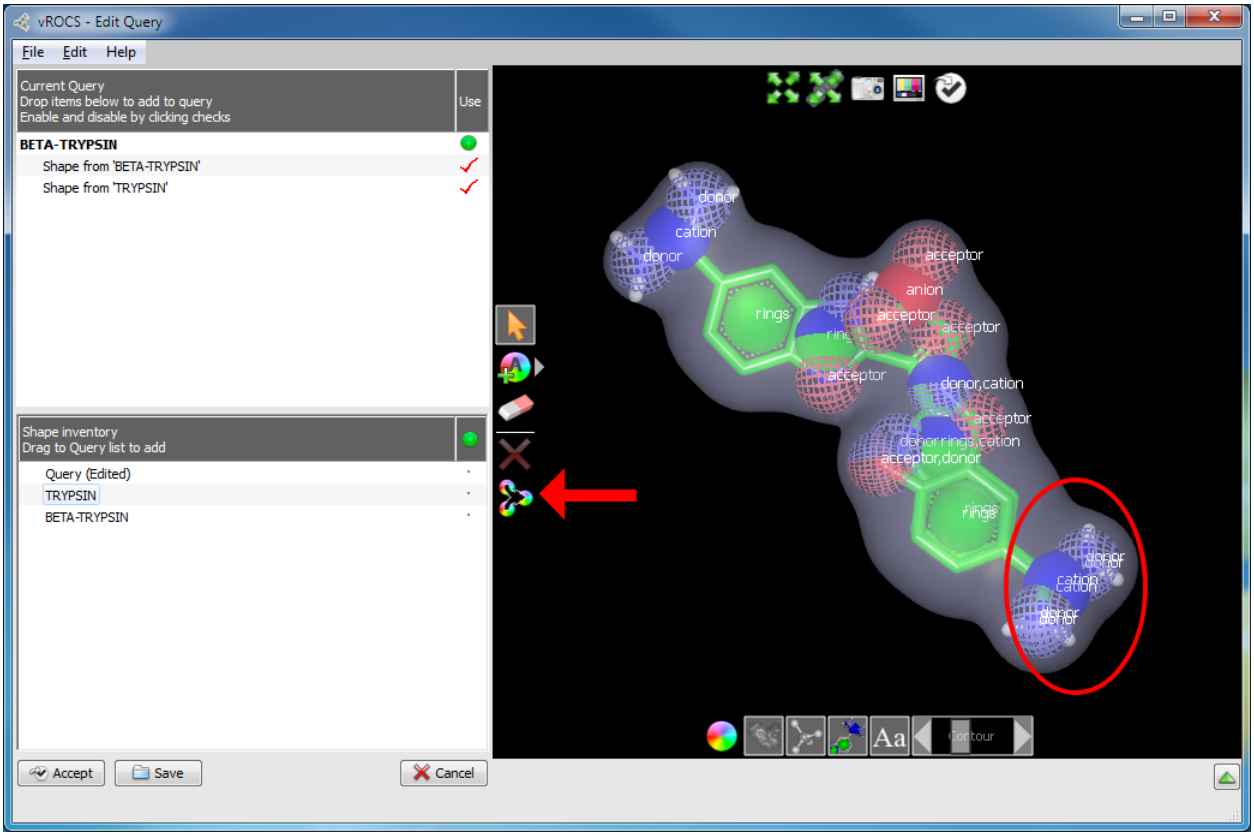


Fig. 36: **Merging color atoms**
Aligned ligands from 1C2D.pdb and 1G3D.pdb in a single query. The highlighted area illustrates where color atom merging is useful. The arrow indicates the `Merge Color Atoms` button

In the *Merging color atoms* figure, the ligands from 1C2D.pdb and 1G3D.pdb *above* both possess a terminal benzamidine group, located at nearly identical positions.

Selecting the Merge Color Atoms tool when no color atoms are selected, will merge similar color atoms within 0.75 Angstroms of each other. (See the figure *Merged ligands and color atoms*.) To merge specific color atoms together, simply select two or more color atoms of the same type, and then use the Merge Color Atoms tool.

Note that if the current query is composed from multiple molecules, it will be collapsed into a single super molecule when color atoms are merged. This means that each individual ligand can no longer be separately selected, hidden, or deleted from the current query.

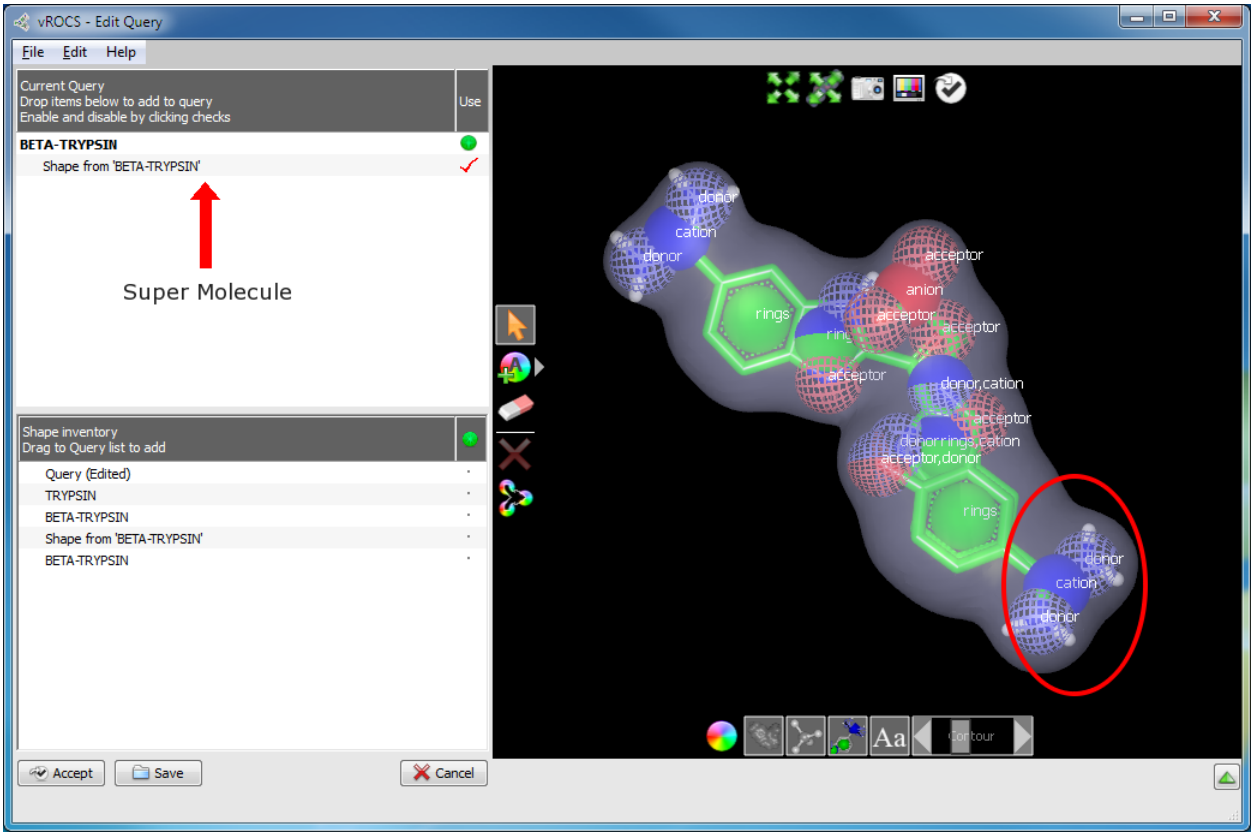


Fig. 37: **Merged ligands and color atoms**
Aligned ligands from 1C2D.pdb and 1G3D.pdb in a single query ("super molecule"). The highlighted area illustrates where color atom merging simplified the query.

Load grid: The majority of **ROCS** queries tend to be ligand-based. However, queries from other sources can also be useful, particularly if no active ligand is known. This is discussed in *ROCS Shape Query Sources*. Grids are loaded into the **vROCS** interface in the same way as ligands, the major difference being that no color features are automatically added to a grid. In the Edit Query mode `Use File > Open` and browse to the desired grid file. The grid will be loaded into the Shape Inventory area of the Build Query panel from where it can be dragged up to the Current Query.

Add color atom(s) to grids: When a grid is first loaded into **vROCS** it has no associated color atoms because the color atoms can only be automatically assigned to a ligand. However, adding color atoms to grid-based queries can enhance **ROCS** search selectivity, just as for ligand-based queries.

Color atoms can be added to grid shapes in the same manner as adding color atoms to ligands. Select the desired Add Color Atom tool (acceptor, donor, etc.) and click on the grid contour surface to place a color atom at that position on the contour surface. Clicking subsequent times on the color atom will increase the weighting for that color atom. Color atoms that one might expect to place on/near a grid surface are those that would make strong protein-ligand interactions, e.g. H-bonding.

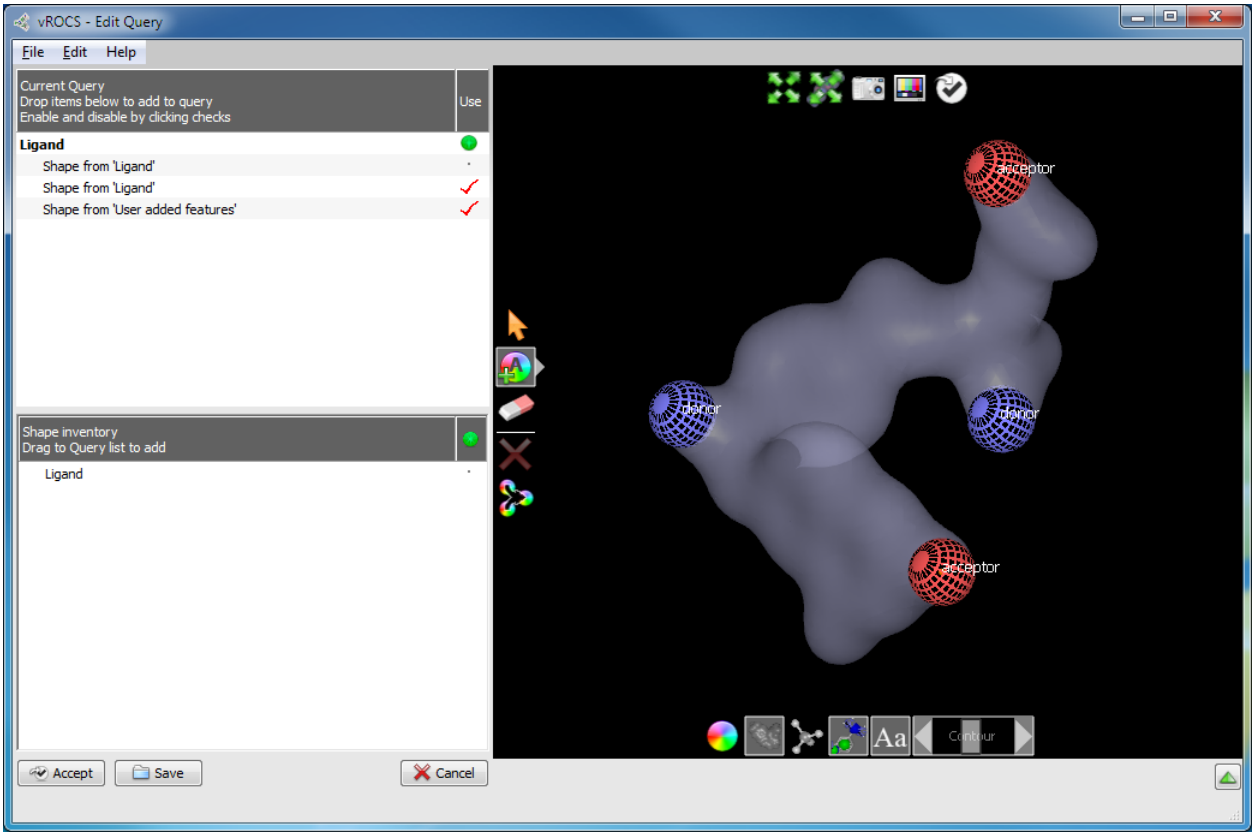
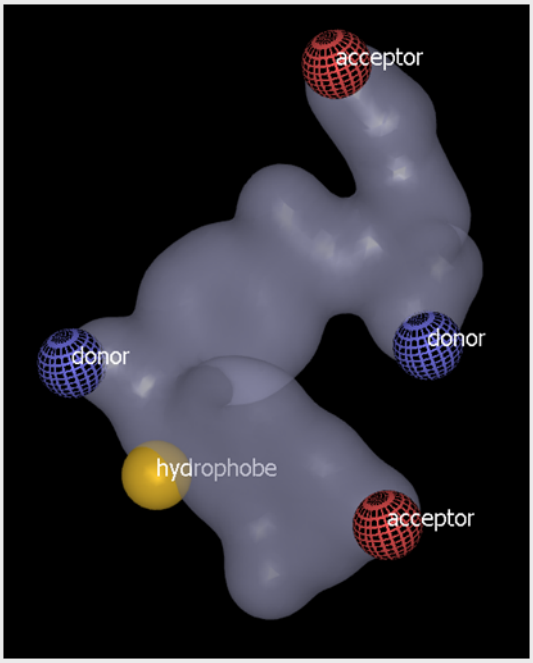
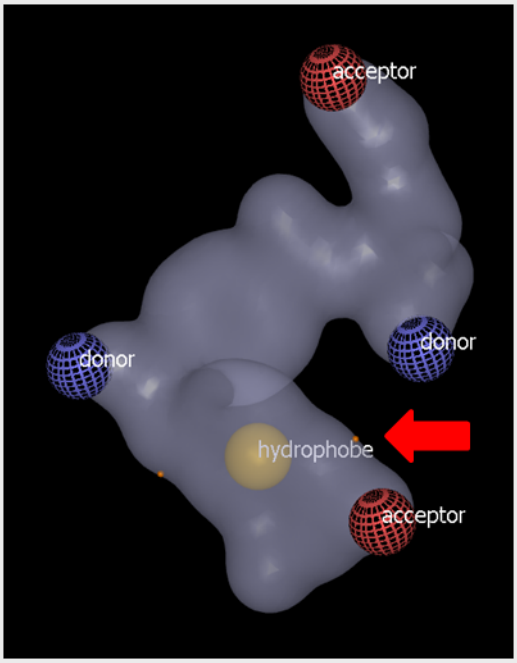


Fig. 38: **Grid surface color features**

Hydrophobic and ring color atom types would normally be buried within the shape contour. These can be placed by initially clicking on the grid surface (see *Grid surface color features* above) and then CTRL-click elsewhere on the surface to move the color feature to the mid-point between the two surface points (see *below*). Additional surface points can be used to position the color atom at the desired position.

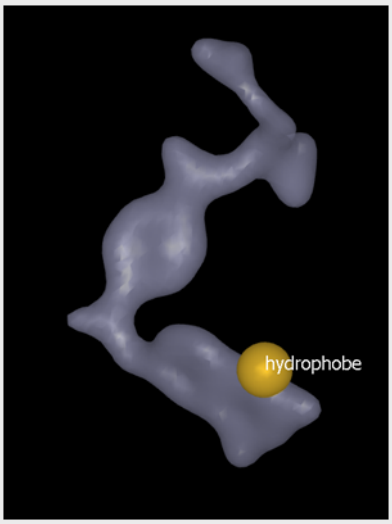


Click on the grid surface to place an initial hydrophobe feature, indicated by the yellow sphere

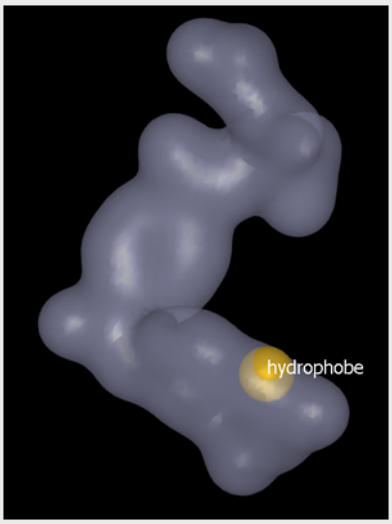


CTRL-click (red arrow) to bury the feature mid-way between the two surface points (indicated by orange dots)

An alternative method of placing color atoms within the grid is to use the Contour level slider. The default contour level is set at 1. Increasing the contour level (up to a maximum of 3) has the effect of displaying a smaller surface shape. Place the color feature on the new contour surface (*Hydrophobe feature placed on contour surface…*) using the Add color atoms tool described above and then use the slider to change contour level back up to 1.0 (*Contour level returned to 1.0…*).



Hydrophobe feature placed on contour surface at contour level 3.0



Contour level returned to 1.0 results in a partially buried color atom

3.8 vROCS menus

3.8.1 File

Menu Item	Description
New Query…	Starts the Query Building Wizard. The resulting query (or queries) will be listed in the query input for a simple or validation run and displayed in the 3D window. They are also placed in the Shape Inventory list. This option is only available in view mode.
Open Query…	Opens a file browser for opening a saved **ROCS** query file. Acceptable saved query sources are molecules of a variety of 3D molecule types, grids or shape queries (*.sq). The opened file is listed as a potential query for simple or validation ROCS runs and displayed in the 3D window. It is also placed in the Shape Inventory list. This option is only available in view mode.
Add to Shape Inventory…	Starts the Query Building Wizard. The resulting query (or queries) will be placed in the Shape Inventory list. This option is only available in edit mode
Open…	Opens a file browser for opening a shape query source. Acceptable shape sources are molecules of a variety of 3D file types, grids or shape queries (*.sq). The opened file will be placed in the Shape Inventory list. This option is only available in edit mode
Save Query…	Save the current query in **ROCS** query format (*.sq). This is a save as… operation. The user will be prompted to enter a filename and directory location for saving the new *.sq file. The current file will never be overwritten. Keyboard shortcut is CTRL+S.
Save Color Force Field…	Save as… operation to save a copy of the current color force field file and prompts for a filename and directory. This can be useful as a starting point to edit a new custom color force field file.
Clear…	Clears all objects from the current **vROCS** session. Keyboard shortcut is CTRL+N.
Recent Queries.	Select from a list of recently opened queries - molecules, grids and shape queries. The selected query will be added to the list of queries available for simple or validation ROCS runs and will be displayed in the 3D window. The list persists across vROCS sessions. This option is only available in view mode.
Recent Database	Select from a list of recently opened databases. The selected database will be used to populate the database field for simple **ROCS** runs. The list persists across **vROCS** sessions. This option is only available in view mode for the simple run set-up and sometimes from the Welcome page.
Recent Actives…	Select from a list of recently opened databases. The selected database will be used to populate the actives field for validation **ROCS** runs. The list persists across vROCS sessions. This option is only available in view mode for the validation run set-up and sometimes from the Welcome page.
Recent Decoys…	Select from a list of recently opened databases. The selected database will be used to populate the decoys field for validation **ROCS** runs. The list persists across **vROCS** sessions. This option is only available in view mode for the validation run set-up and sometimes from the Welcome page.
Recents…	Select from a list of recently opened shape sources - molecules, grids and shape queries. The list persists across **vROCS** sessions. This option is only available in edit mode.
Exit…	Close the **vROCS** session. Displayed in the Application menu on Mac as `Quit...`

3.8.2 Edit

Menu Item	Description
Undo…	Undo the last action. This can be done repeatedly or see Undo History… Keyboard shortcut is CTRL+Z
Undo History…	Shows the last 10 operations that can be selected to undo. The Undo history list can be much greater than 10 items so revisit this list to see additional items.
Redo…	Redo the last action that was just undone. This can be done repeatedly or see Redo History… Keyboard shortcut is CTRL+Y
Redo History…	Shows the last 10 operations that have been undone and can be selected to redo. The Redo history list can be much longer than 10 items so revisit this list to see additional items.
Preferences…	Opens the Preferences dialog to set user preferences that will persist across **vROCS** sessions. There are two pages: vROCS and Display. See below for full details. Displayed in the Application menu on Mac.

Preferences

Every user has his or her own individual preferences with regards to how molecules, grids, and surfaces should look and how applications should behave. For this reason, a Preferences dialog is available which allows customization of the application to the user's preference. The first time **vROCS** is launched the Preferences will open automatically, to enable the user to set his or her own options. A snapshot of the Preferences dialog can be seen *below*. On the left-hand side of the dialog is column containing preference categories. These categories include: **vROCS** and Display. Clicking on any of these categories will update the right-hand side of the dialog to display the options corresponding to the selected category. The vROCS category includes preferences for **ROCS** and the color force fields. The Display category contains preferences for the Open GL display of molecules and shapes.

vROCS Preferences

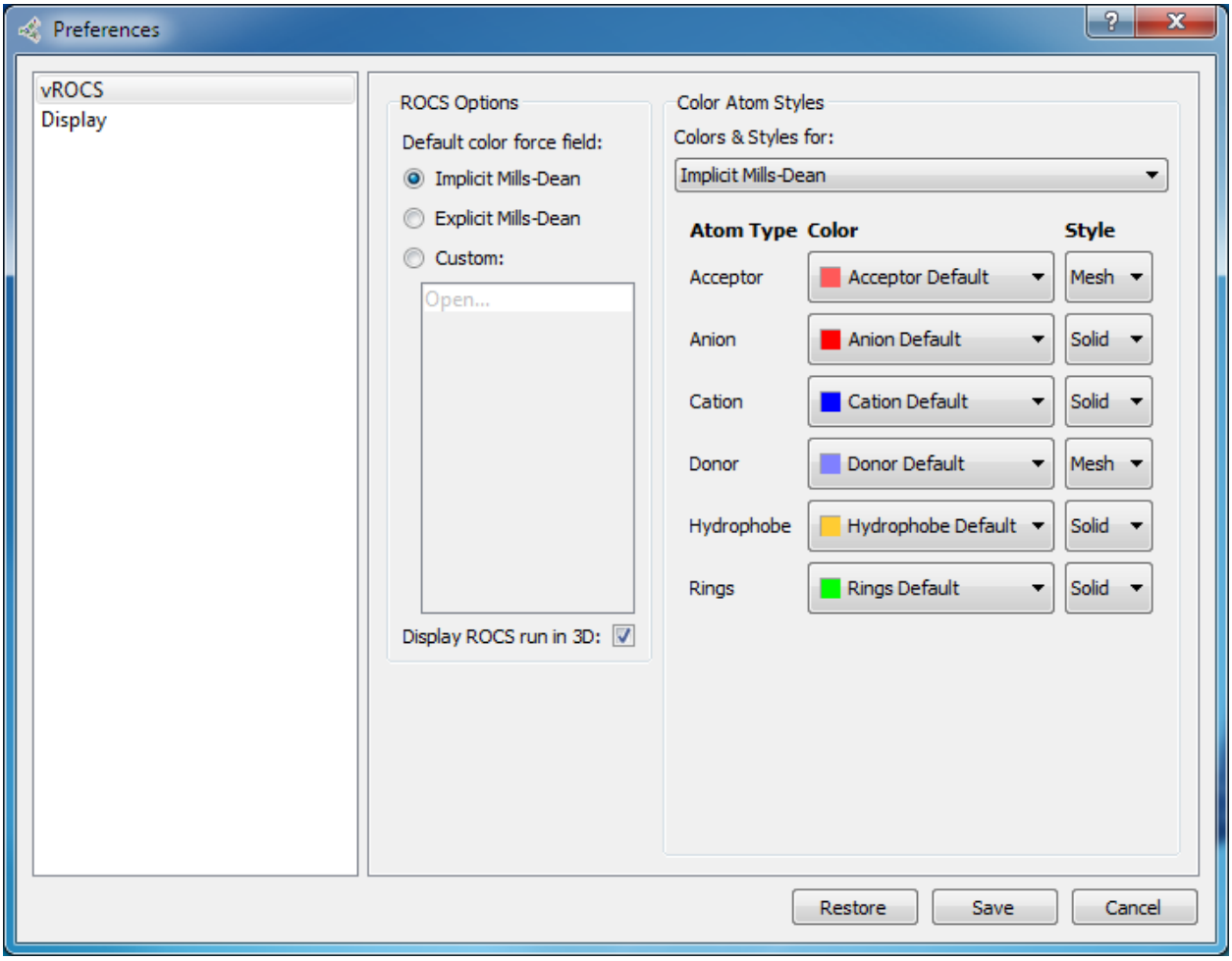


Fig. 39: **Preferences: vROCS**

Menu Item	Description
Default color force field	Click the radio buttons to select the default color force field which will be used to apply color atoms to molecules that are loaded into **vROCS** or to define the nature and interaction of color atoms that are added during manual query editing. This color force field information will be saved in any saved query file. Options are: Implicit Mills-Dean Explicit Mills-Dean Custom To use a custom color force field define the path to the color force field (*.cff) file that contains definitions of the color atoms. Multiple custom color force fields can be loaded into **vROCS** and are listed in the box. The selected (active) custom color force field will be highlighted in blue. Custom color force fields can be deleted from the list by clicking on the red X beside their name in the list. Changing the default color force field during an **vROCS** session will not result in changing the force field for any currently opened molecules or queries. Changes will take place only if no molecules or queries are opened or on closing and reopening **vROCS**.
Display ROCS run in 3D	Check on to display a 3D Open GL rendering of the query and database molecules aligning as the run progresses together with 2D structures for the current 5 best hits. Click off to display a text-based run progress summary, saving compute resources on lower performance computers. This is equivalent to checking on/off the 3D View option in the simple or validation Run Set-Up Options dialog but persists across all runs and sessions, not a single run.
Color Atom Styles	Change the color and style for the different color atom types. Select the force field from the drop-down menu. For each color atom type select a color from the drop-down color list and a style (solid or mesh) for the color atom display. Restore default color atom colors and styles by clicking the Restore button below.
Restore	Click to restore the current preferences to the default ones.
Save	Save the current preferences for this session and for future sessions of **vROCS**.
Cancel	Click to close this dialog without applying any of the changed preferences.

OpenGL Preferences

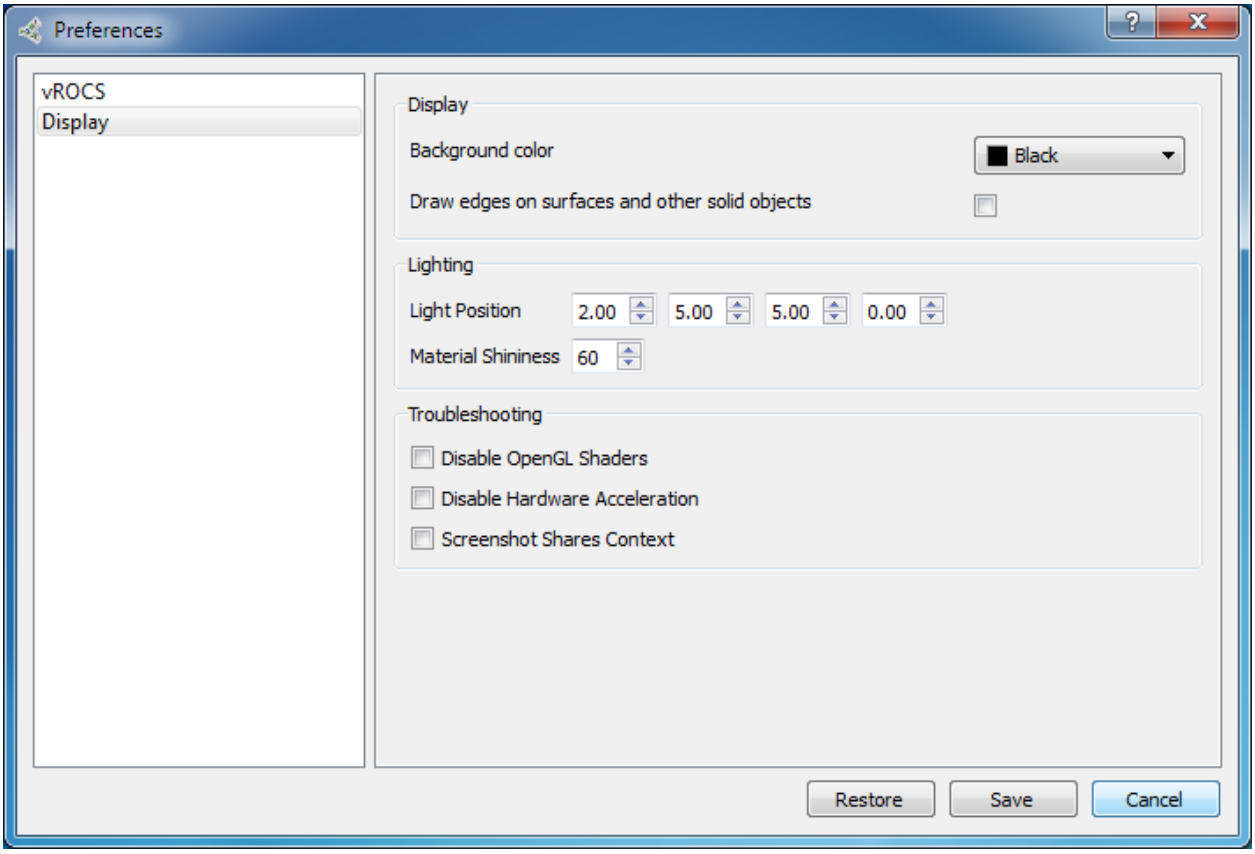


Fig. 40: **Preferences: Display**

Menu Item	Description
Background Color	Change the background color by selecting from a drop-down color list.
Lighting Position	Sets the position of the lighting used in the 3D view.
Material Shininess	Sets the shininess of solid-rendered objects in the 3D view.
Disable OpenGL Shaders	Turns off hardware shading functions. May be useful if the 3D scenes are not rendering properly.
Disable Hardware Acceleration	Turns off all hardware rendering. May be useful if scenes are not rendering properly due to video driver problems.
Screenshot Shares Context	May be useful if screenshots are not being saved properly on some systems.
Restore	Click to restore the current preferences to the default ones.
Save	Save the current preferences for this session and for future sessions of **vROCS**.
Cancel	Click to close this dialog without applying any of the changed preferences.

There are three buttons at the bottom of the dialog. Clicking on the Restore button will restore the current preferences to the default ones. Clicking on the Save button will save the current preferences for this run and for future runs of

vROCS. Clicking on the Cancel button will close this dialog and will not apply any of the changed preferences.

Preferences are stored in a binary file (preferences.oeb) in a user-specific local directory on the computer currently running the application. The preferences file can be found in:

- `C:\Documents_and_Settings\USERNAME\AppData\Local\OpenEye\vROCS\<version>` on Microsoft Windows Vista.
- `C:\Users\USERNAME\AppData\Local\OpenEye\vROCS\<version>` on Microsoft Windows 7.
- `~USERNAME/.OpenEye/vROCS/<version>` on all other platforms.

While the preference file shares the same file extension as the OpenEye's binary database file, it cannot be read into **vROCS** using the `File > Open` menu item. The preferences file is loaded automatically when the application starts and is saved back to disk when the application exits. Deleting this file is equivalent to clicking on the Restore button in the dialog box.

There is also a file in this same directory which called vROCS.ini which contains machine specific settings like the list of recent files, preferred layouts, and hardware stereo options. Deleting this file will restore these settings to the defaults as well.

This directory can be opened from within vROCS by selecting the `Open User Directory` menu item in the Help menu.

3.8.3 Help

Menu Item	Description
About…	Provides version and release information. This should be provided when requesting technical support. Displayed in the Application menu on Mac
Documentation…	Access this user documentation manual in either PDF or HTML format.
License…	Open and display the current license file or access a browser to set the license file.
Open Data Directory…	Opens a file browser in a platform-specific location for access to the tutorial datasets.
Open User Directory…	Open a file browser in the user directory containing the preferences.oeb and vROCS.ini files described above.
File Bug Report…	Opens a new email prefilled with "support@eyesopen.com" in the To: field. Write an informative title and add details of the bug you wish to report. Don't forget to include details of your platform and OS.

CHAPTER

FOUR

UTILITIES

4.1 CheckCff

4.1.1 Overview

This is a simple utility that will simply apply a color force field to one or more input molecules and output a report of color atoms added, the TYPE of the added atom and the corresponding input molecule atoms that matched the SMARTS defining that TYPE. Additionally, **checkcff** will output an OEB file containing molecules with the color atoms added. This file can be loaded into VIDA and the atoms labeled with *Name* to visually inspect which color atoms are being added.

4.1.2 Example Commands

By default, **checkcff** will use the ImplicitMillsDean color force field. This can be over-ridden by using the *-chemff* option and providing either a name of one of the built-in color force fields (ImplicitMillsDean, ExplicitMillsDean, ImplicitMillsDeanNoRings, ExplicitMillsDeanNoRings) or the name of a user-defined color force field file.

So to see which atoms are considered color atoms using the ImplicitMillsDean force field:

```
prompt> checkcff -in mymolecules.sdf
```

To use the ExplicitMillsDean force field to see which atoms are considered to be color atoms:

```
prompt> checkcff -in molecules.sdf -chemff ExplicitMillsDean
```

The run always generates a report CSV file and an OEB file that can be viewed in VIDA:

```
prompt> checkcff -in 4cox-neutral.sdf
```

And the view in **VIDA** might look a bit like:

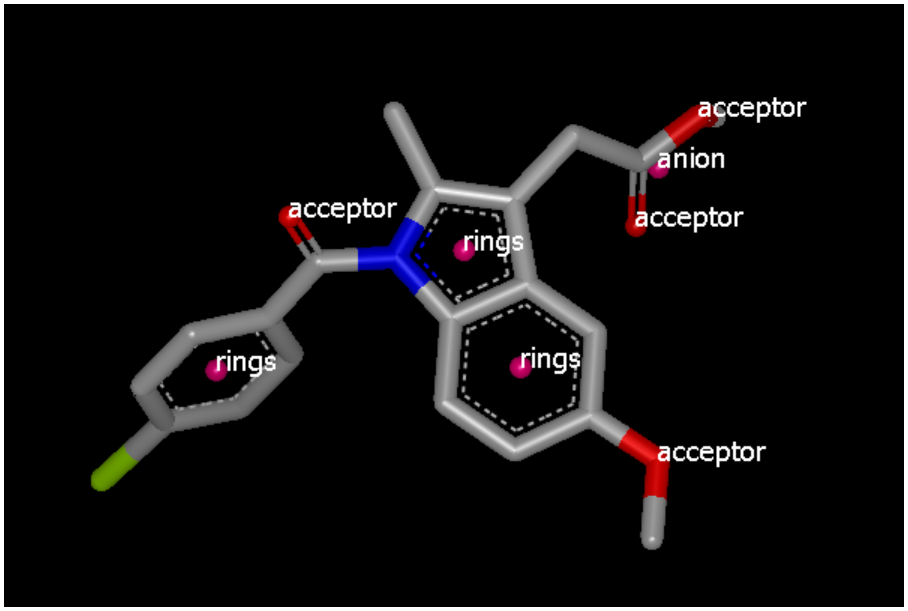


Fig. 1: **VIDA** view of 4COX with color atoms attached and labeled.

4.1.3 Command Line Help

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

```
> checkcff --help
```

will generate the following output:

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

4.1.4 Required Parameters

checkcff only has 1 required commandline parameter.

-in <filename>

Input molecule file to be colored. Can be any one of the molecule file formats described in Section *The Query File*.

4.1.5 Optional Parameters

[Aliases=**-i**]

-out <oebfile>

Output OEB file name for 3D structures with color atoms named by the TYPE from the color force field.

[Default=checkcff.oeb] [Aliases=**-o**]

-report <csvfile>

Override prefix used to name report file.

[Default=checkcff_rpt.csv]

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]

Overlap Prep Option Parameters

There are 6 overlap prep option parameters.

-chemff <cff_file>

Color-force-field name. Either the name of one of the built-in color force fields (ImplicitMillsDean or ExplicitMillsDean or ImplicitMillsDeanNoRings or ExplicitMillsDeanNoRings) or the name of a user-defined color force field file. The format of this file is given in section *Color Force Field*.

[Default=ImplicitMillsDean]

-assignColor

Determines the state of the assign color flag. If true, color atoms would be assigned during the Prep.

[Default=true]

-useHydrogens

Determines the state of the use hydrogens flag. If true, explicit hydrogens atoms would be created during the Prep. If false, all hydrogen atoms would be removed.

[Default=false]

-assignBondiRadii

Determines the state of the assign Bondi radii flag. If true, Bondi radii would be assigned to all the atom during the Prep.

[Default=false]

-removeDups

Determines if close color atoms with the distance less than *-minDistance* should be removed.

[Default=false]

-minDistance

Threshold for duplicate removal. A value in the interval (0,1) that determines the minimum allowed distance between two color atoms.

[Default=0.001]

4.2 Chunker

4.2.1 Overview

This is a simple commandline utility to take an input database file and divide it into similar-sized smaller pieces. Then each piece can be used as a dbase file in a separate **ROCS** run. This divide-and-conquer approach is an alternative way to run a single dbase over multiple CPUs but without the use of MPI.

4.2.2 Example Commands

To break `input.oeb.gz` into 5 chunks, each with the same number of molecules.

```
prompt> chunker -in input.oeb.gz -base bar -nchunks 5
```

would create

```
bar0000001.oeb.gz
bar0000002.oeb.gz
bar0000003.oeb.gz
bar0000004.oeb.gz
bar0000005.oeb.gz
```

To break input.oeb.gz into chunks, each with 1000 multi-conformer molecules:

```
prompt> chunker -in input.oeb.gz -base foo -chunksize 1000
```

4.2.3 Command Line Help

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

```
> chunker --help
```

will generate the following output:

```
Help functions:
  chunker --help simple      : Get a list of simple parameters
  chunker --help all         : Get a complete list of parameters
  chunker --help defaults    : List the defaults for all parameters
```

(continues on next page)

(continued from previous page)

```
chunker --help <parameter> : Get detailed help on a parameter
chunker --help html        : Create an html help file for this program
chunker --help versions    : List the toolkits and versions used in the application
```

4.2.4 Required Parameters

-in <filename>

Name of input file to chunk.

-base <NAME>

Base name of output files. Output files will be sequentially numbered.

And one of the following two options must be used:

-nchunks N

Create N new files of equal number of conformers or molecules. Chunker will read through the entire file once to count the number of conformers/molecules, then will create the new files. The switch is the -countConfs flag. N must be a positive integer.

-chunksize M

Create new files, each containing M molecules. M must be a positive integer.

> **Note**
>
> Only one of -nchunks or -chunksize can be used.

4.2.5 Optional Parameters

-countConfs

If the flag -countConfs is set to true then the file will be switched to give approximately equal numbers of conformers in each chunk. The split always occurs at the end of each molecule so all the conformers for each molecule are kept together. If the flag -countConfs is set to false then the file will be switched to give equal numbers of molecules in each chunk.

[Default=true]

-pad_zeros

This option will pad the front of the output filenames with zeroes, which helps keep files in order when doing a sort of filenames.

[Default=true]

4.3 HLMerge

4.3.1 Overview

This utility takes as input a single molecule file (SD or OEB) or list file containing one or more molecules files. It will re-rank these input files by a user-defined SD tag. The most common use is to create a single hitlist after running **ROCS** on the separate files created by chunker.

4.3.2 Example Commands

Assuming a set of ROCS results for the previous chunker example are in a file called `bar.list`:

```
bar0000001_hits_01.sdf
bar0000002_hits_01.sdf
bar0000003_hits_01.sdf
bar0000004_hits_01.sdf
bar0000005_hits_01.sdf
```

To merge these, rank by Tanimoto Combo and then keep the top 200.

```
prompt> hlmerge -in bar.list -out bar_hits.sdf -rankby ROCS_TanimotoCombo -besthits 200
```

4.3.3 Command Line Help

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

```
> hlmerge --help
```

will generate the following output:

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

4.3.4 Required Parameters

-in <filename>

Name of input file to rank, or the name of a list file (`.lst` or `.list`) containing a collection of files to rank. [Aliases=**-i**]

-out <filename>

Output structure file for sorted hits. Formats other than SDF or OEB will lose the SD tag data. [Aliases=**-o**]

-rankby <SD tag>

SD tag containing a score (that can be converted to a floating point value) to use as the score to rank with.

4.3.5 Optional Parameters

-besthits N

Number of hits to keep. [Default=500]

-biggerisbetter

By default, hits are ranked with higher scores meaning better results. To rank by a score where the smaller value is better, use -biggerisbetter false. [Default=true]

4.4 MakeRocsDB

4.4.1 Overview

> **Note**
>
> The format for `.rocsdb` is specific for **ROCS**. The format may change in the future, following more work in optimizing for large clusters. Or the format may ultimately be retired if no longer needed.

In order to provide maximum throughput for scalability, this version of ROCS comes with a utility to convert existing **OMEGA** dbase files into a `.rocsdb` file. This is **not** required for normal **ROCS** usage, only for scaling to 64 CPUs and beyond under MPI.

4.4.2 Example Commands

So to convert an OMEGA database file, `omega_confs.oeb.gz` into **ROCS** DB format:

```
prompt> makerocsdb -in omega_confs.oeb.gz -out rocsinput
```

will create a new file, `rocsinput.rocsdb`

By default, **makerocsdb** will apply the same contiguous conformer test that **ROCS** does and will attempt to merge contiguous conformers into a single multi-conformer molecule on output. If this behavior is not desired, then the `-scdbase` flag can be set to true and each input molecule will be written into the rocsdb file as a separate molecule.

4.4.3 Command Line Help

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

```
> makerocsdb --help
```

will generate the following output:

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

4.4.4 Required Parameters

makerocsdb has only 2 required command-line parameters:

-in <filename>
: Input dbase file to be converted. Can be any one of the molecule file formats described in Section *The Database File*. [Aliases=**-i**]

-out <prefix>
: Prefix for output `.rocsdb` file. [Aliases=**-o**]

-scdbase
: Read input as single conformer. Do not combine contiguous conformers. [Default=false]

-chemff <cff_file>

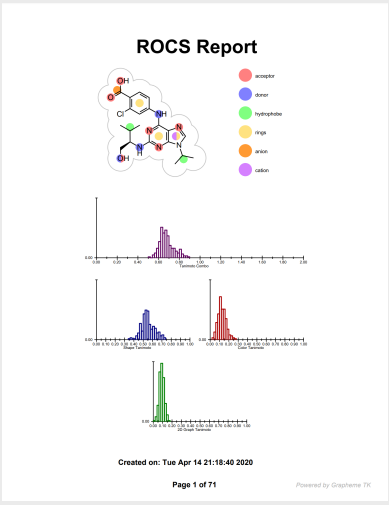
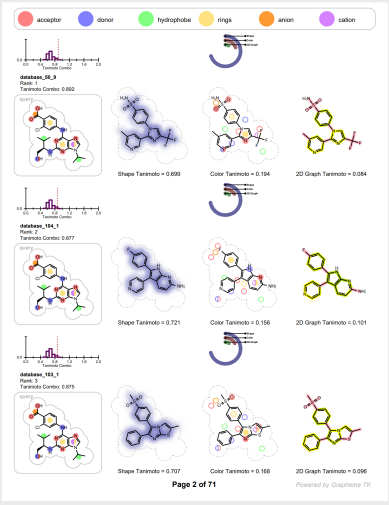
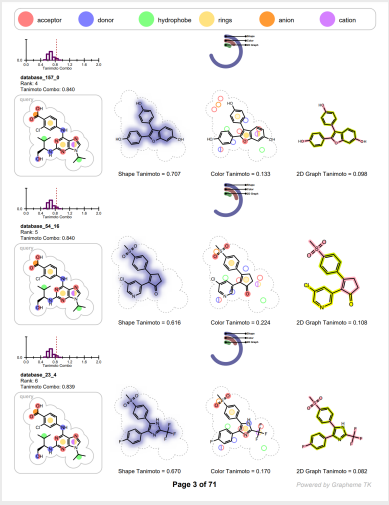
Chemical force field file to use. [Default=ImplicitMillsDean]

4.5 ROCSReport

4.5.1 Overview

RocsReport is a utility program that takes the hit molecule file generated by ROCS and creates a multi-page PDF document that visualizes the hit structures along with their score information. The layout of the generated report is depicted in *Table: Example of a multi-page document generated by the rocs_report program*

Table 1: **Example of a multi-page document generated by rocs_report program** (The pages are reduced here for visualization convenience)

page 1 (front page)	page 2	page 3
		

4.5.2 Report File

Front Page

The front page of document summarizes information about the ROCS search:

It shows the 2D representation of the query molecule. (See *Figure: Query in 2D*). The 2D coordinates of the query displayed in the report are driven by 3D coordinates read from the input file. The atom colors of the 3D query are projected into the 2D molecular graph and visualized by filled circles. Each atom color type is associated with a color and the legend of the color atoms is display along the 2D query structure. The arcs around the query represents the 2D surface of the molecule. The 3D representation of same query along with its color atoms is shown in *Figure: Query in 3D*.

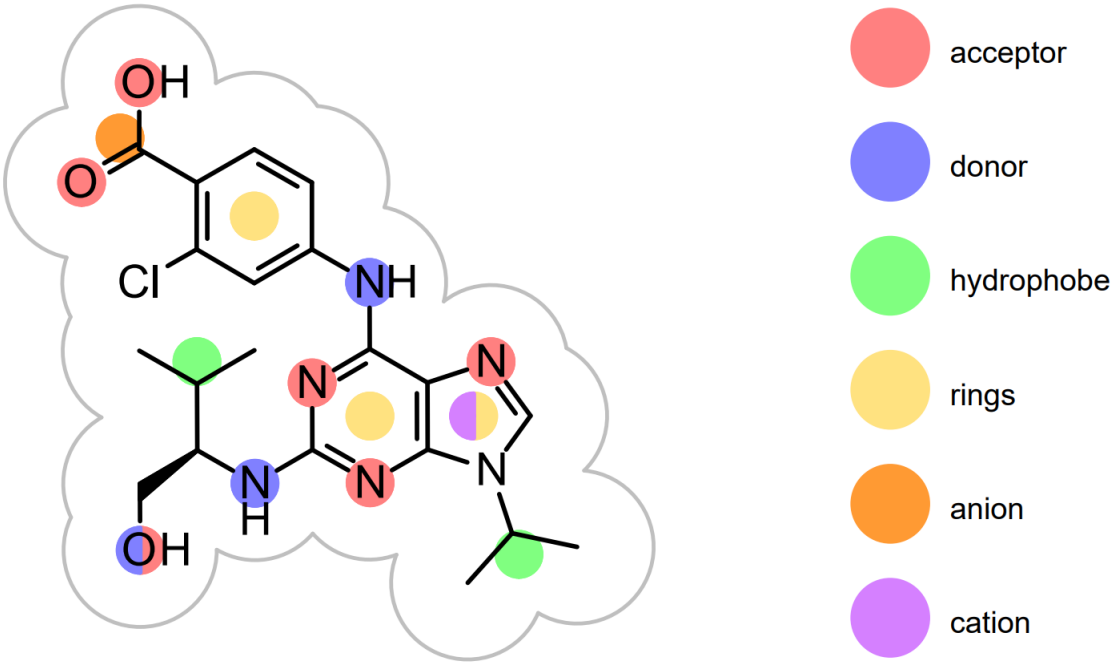


Fig. 2: **Query in 2D**
Example of the 2D visualization of the query structure with color atoms on the front page of a ROCS report

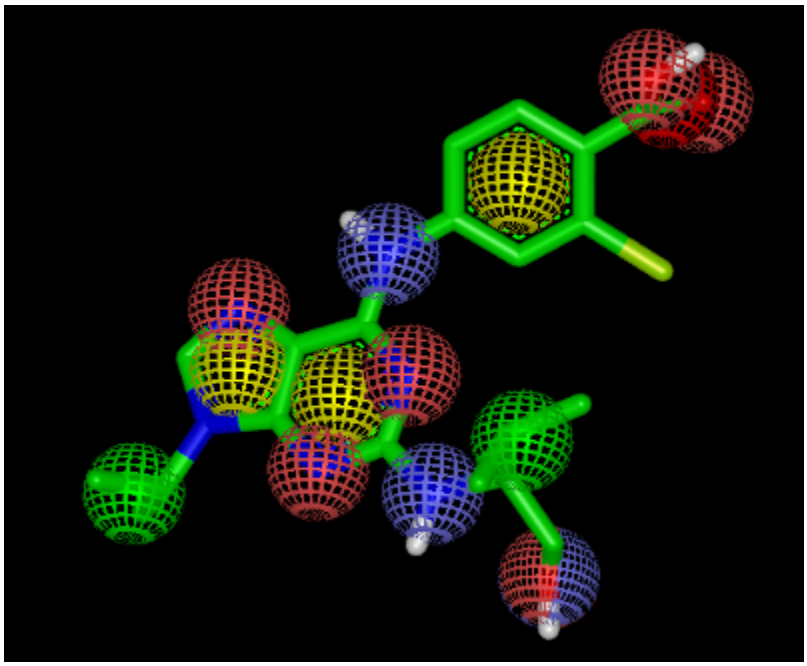


Fig. 3: **Query in 3D**
Example of the 3D visualization of the query structure with color atoms in vROCS

See also

- *Color Features* section
- OERenderShapeQuery function in the **Grapheme TK** manual.

Warning

`rocs_report` currently cannot visualize ROCS outputs for multi-conformer queries.

The front page also displays the histogram of the score distribution of the input hit molecules for the following scores:

- Tanimoto Combo score
- Shape Tanimoto score
- Color Tanimoto score
- 2D Similarity Tanimoto score

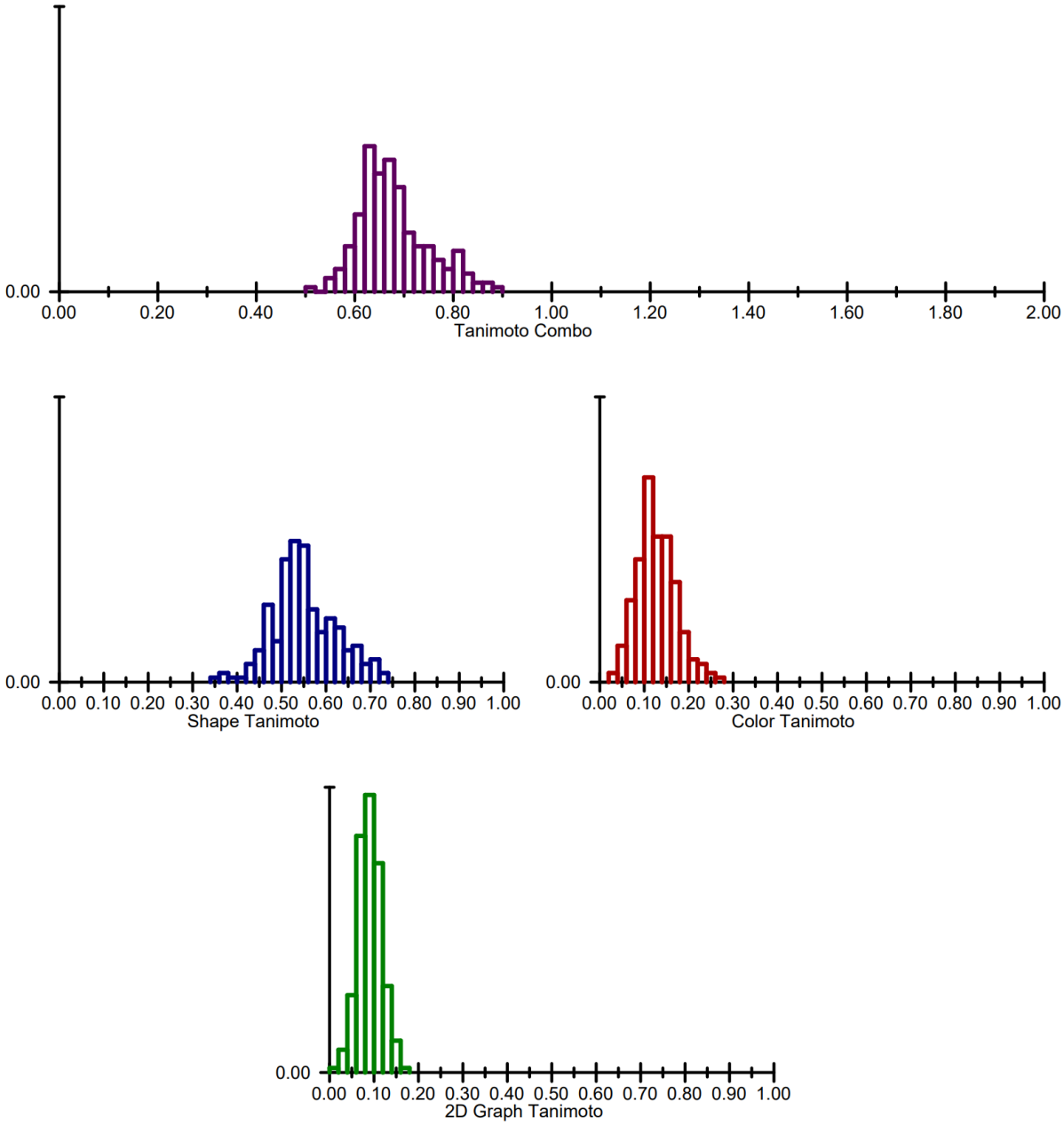


Fig. 4: **The histograms of the score distribution**

The first three scores are calculated by ROCS and read from the input file. The 2D similarity scores are calculated on-the-fly by the **`rocs_report`** program using the tree fingerprints of the **OEGraphSim TK** ([GraphSim]).

Visualizing Shape and Color Overlays

Each row on the following pages represents information of an individual hit structure read from the input file. See example in *Figure: Example of the 2D visualization of a hit structure*. The 2D coordinates and layout of the hit molecule are calculated based on the 3D alignment of the hit and the query structures. The corresponding 3D overlay between of the hit molecule and the query is show in *Figure: Example of 3D overlay between the query and hit structure*.

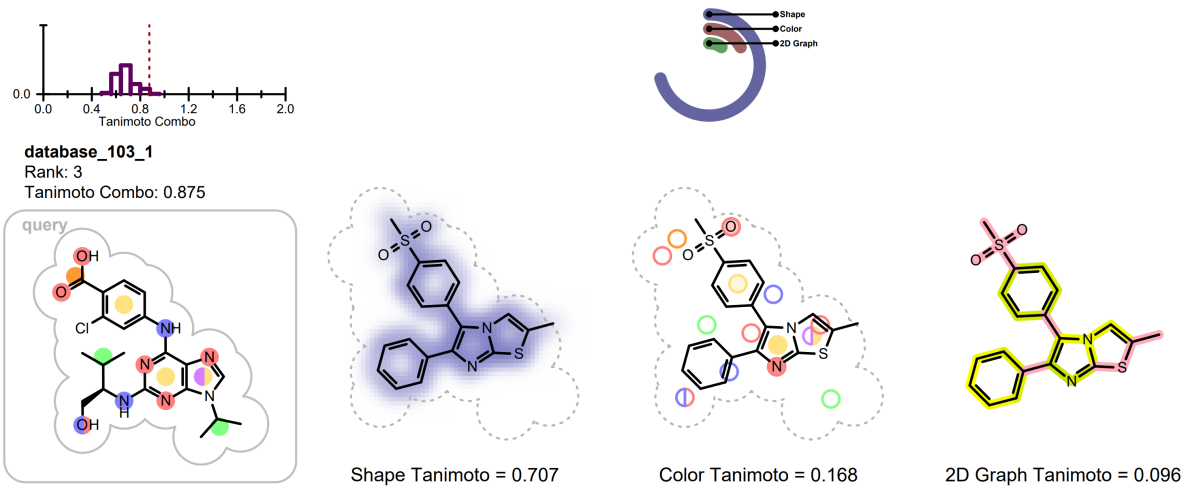


Fig. 5: **Example of the 2D visualization of a hit structure**

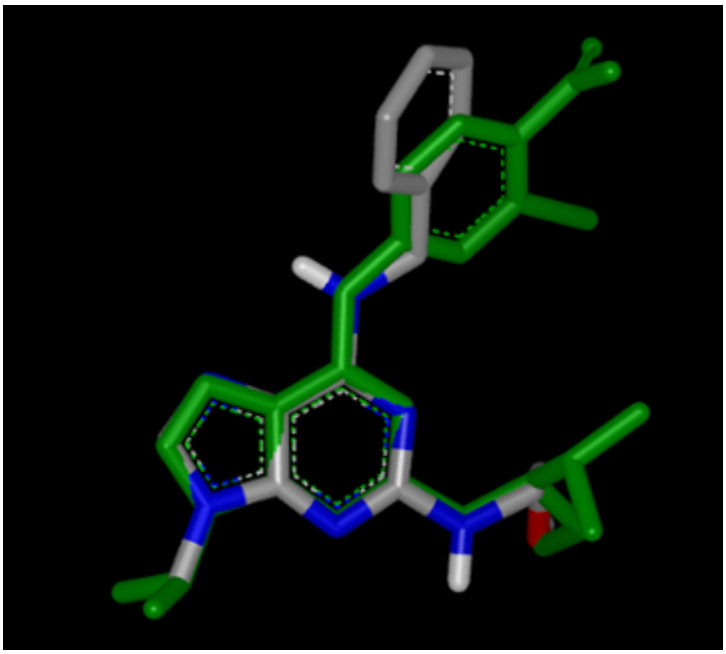


Fig. 6: **Example of 3D overlay between the query (colored by green) and hit structure (colored by atomic number)**

In each row the query molecule among with distribution of the TanimotoCombo scores is depicted. On the histogram, the position of the score of the given hit is marked with a dotted red line (see *Figure: Example of the 2D visualization of a hit structure*).

The hit structure is depicted three times visualizing the following information:

- **shape overlap** between the hit and the query
- **color atom overlap** between the hit and the query
- **2D graph similarity** between the hit and the query

The **shape overlap** between the hit and the query is visualized by using a property map, *i.e.* a 2D grid, laid underneath the molecule structure, where the cells of the grid that are colored blue indicate good 3D shape overlap between the query and the hit structure. Additionally, a clash between the hit structure and 2D molecule surface of the query structure indicates shape mismatch in 3D. (See *Figure: Example of visualization of shape overlay*).

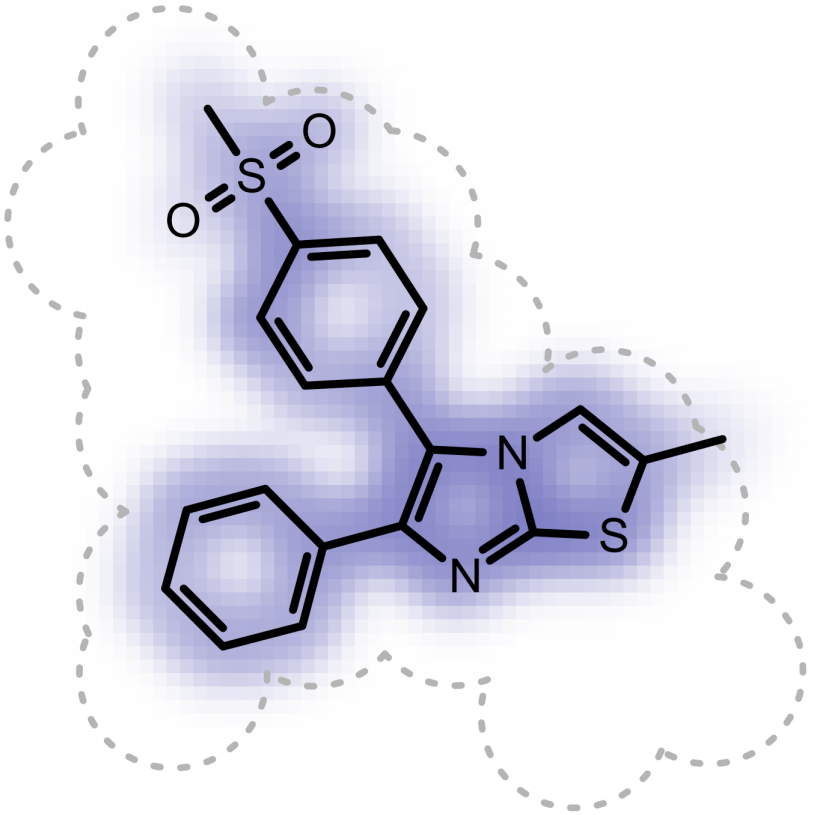


Fig. 7: **Example of visualization of shape overlay**

See also

- *Shape Theory* section
- OERenderShapeQuery function in the **Grapheme TK** manual.

The **color atom overlap** between the hit and query are visualized with circles. Each circle corresponds to a color atom in the query molecule. The color of the circle indicates the fitness of the color atom match in 3D. The lighter the color, the smaller the overlap between the query and hit color atoms in 3D. Unfilled circles represent unmatched query color atoms. If there is a good color atom match exist for a query color atom in 3D, then the circle representing the color atom is positioned to the matching fit color atom in 2D.

(See *Figure: Example of visualization of color atom match*).

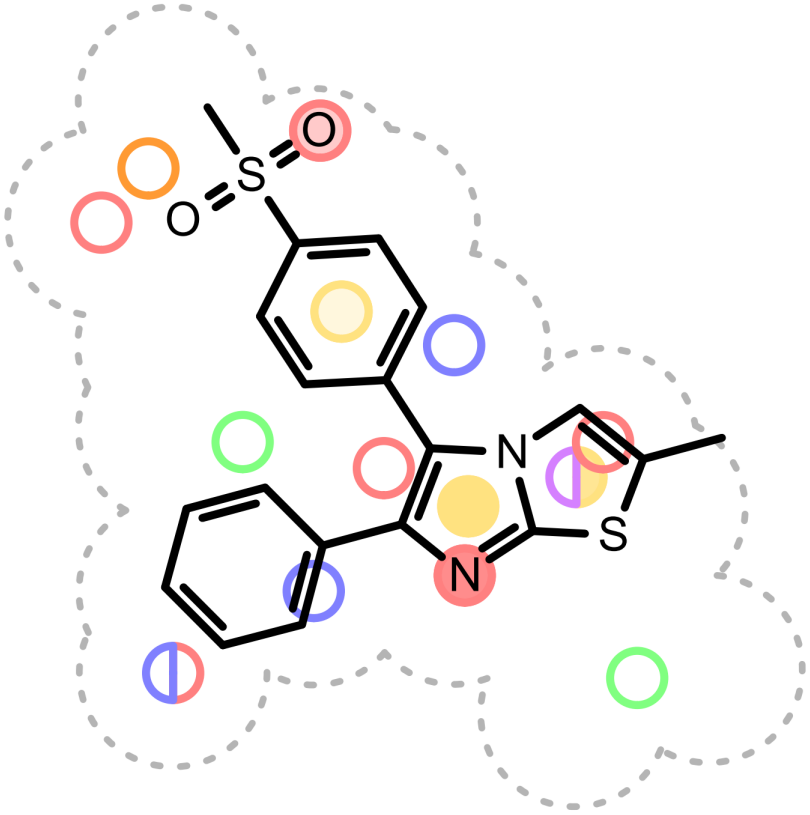


Fig. 8: **Example of visualization of color atom match**

See also

- *Color Features* section
- OERenderShapeOverlap function in the **Grapheme TK** manual.

Visualizing 2D Graph Similarity

The **2D graph similarity** is visualized by using a linear color gradient to highlight molecular similarity or dissimilarity between the hit and query structures. After calculating the 2D molecule similarity score, the bonds of the hit molecule are colored based on how frequently they occur in molecular fragments that can be also detected in the query structure. The color pink is used to highlight parts of the hit molecule that are 2D dissimilar to the query structure. Where there is 2D similarity detected between the hit and the query, the "yellow to dark green" color gradient is used to highlight the bonds and the color gets greener and darker with increasing similarity. (See *Figure: Example of visualization of 2D fingerprint similarity*). The calculation of 2D similarity score and similarity visualization use the tree fingerprint type implemented in the OEGraphSim toolkit.

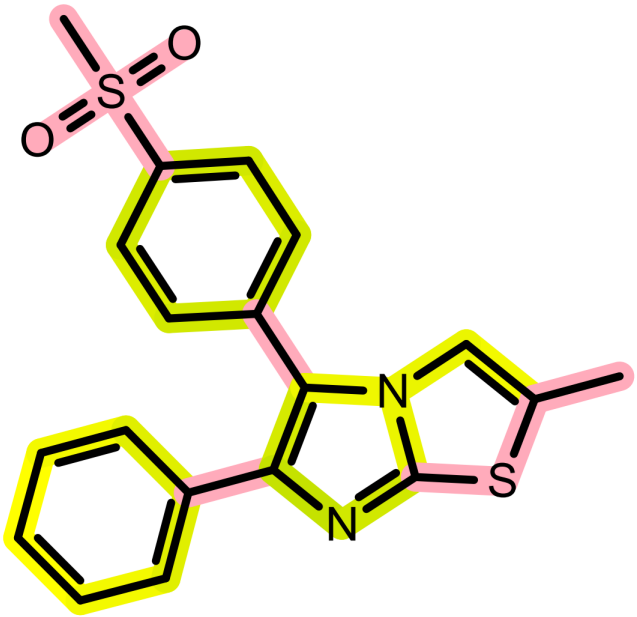


Fig. 9: **Example of visualization of 2D fingerprint similarity**

The relevant score of the given hit structure is printed below the 2D molecular structure while the histogram of the score distribution is displayed above the 2D molecular structure. (see *Figure: Example of the 2D visualization of a hit structure*).

See also

The Python script that visualizes molecule similarity based on fingerprints can be downloaded from the OpenEye Python Cookbook

4.5.3 Command Line Help

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

```
prompt> rocs_report --help
```

will generate the following output:

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

4.5.4 Required Parameters

-in <filename>

-i <filename>

[keyless parameter 1]

OEBinary input file (*.oeb* or *.oeb.gz*) or *.sdf* with results from ROCS. [Aliases=**-i**]

-out <filename>

-o <filename>

[keyless parameter 2]

Output .pdf file.

4.5.5 Optional Parameters

Input/Output Options

[Aliases=**-o**]

-refmol <filename>

Reference i.e. query molecule. If omitted, the first molecule in "-in" is used as the reference. This parameter is not usually required since the **ROCS** hitlist includes the query molecule by default.

General Options

[Aliases=**-ref**]

-maxpages

Maximum number of results pages to output, including the title page. A value of 1 will only return the title page. Use 0 for no maximum.

[Default=0]

-verbose

Triggers more messages

[Default=true]

Molecule Display Options

-aromstyle <style>

-astl <style>

Aromatic ring display style: *Kekule, Circle, Dash*.

[Default=Kekule]

Report Options

-pagesize <size>

-psize <size>

Output file page size: ISO_A4, US_Letter, US_Legal.

[Default=US_Letter] [Aliases=**-psize**]

CHAPTER

FIVE

ROCS TUTORIALS

5.1 Introduction

Four tutorials are included which guide the user through examples of each of the main tasks available in **vROCS**. A fifth tutorial illustrates setting up a simple run via the **ROCS** command line interface. The aim of providing these tutorials is to familiarize the user with the steps required to complete each task and give an understanding of what the task involves. More detailed background information on any of the options or dialogs is available in the Usage sections of this manual. Each tutorial is designed to stand alone so the user can choose which bets fits his/her current research needs. We encourage the user to run the tutorials initially and thereafter they can be used as a guide for your own experiments.

Data files for these tutorials are located in the directory `OPENEYE_DIR/data/vrocs` where `OPENEYE_DIR` refers to the top level OpenEye installation directory. A versioned ROCS directory in `C:\Program Files (x86)\` is default on Windows. The data and documentation directories are easily accessible in OSX distributions as standalone folders in the package. This directory contains four sub-directories:

- The files for *Build/edit a query using the Wizard* are in the sub-directory `wizard`
- The files for *Build/edit a query manually* are in the sub-directory `edit`
- The files for *Perform a ROCS validation run* are in the sub-directory `validation`
- The files for *Perform a simple ROCS run* and *Perform a ROCS run from the command line* are in the sub-directory `simple`

5.2 Build/edit a query using the Wizard

5.2.1 Background information

This tutorial teaches the user to create a new query to either save or to use in ROCS. The wizard creates a query through one of a few predesigned paths.

You have recently been assigned to a trypsin inhibitor project. You are interested in building a query from known ligands in their binding modes, suitable for use in **ROCS** for vHTS. A set of 19 trypsin inhibitors are available as co-crystal structures from the PDB (see the IDs for trypsin inhibitors in the [PDB-IDs] citation). The dataset has been prepared by aligning the 19 protein crystal structures. The ligands were then extracted to give a set of 19 ligands, aligned in the protein binding pocket frame of reference. The query building wizard will be used to construct a **ROCS** query to employ only the ligands that are most representative of the set as a whole.

The tutorial will require approximately 10 minutes to complete.

5.2.2 Build models using the wizard

Open a new **ROCS** session. At the Welcome screen select the option to `Create a query with a wizard`. This will open up the `Create a query` dialog.

In the top tab, Create Query, select the radio button for Ligand Model Builder and then click Next.

In the Load Aligned Ligands tab click on the Filename area and browse to the file containing the 19 aligned trypsin ligands. This file is located in the `OPENEYE_DIR/data/vrocs/wizard` directory and is called `pdbmodel_ligands.oeb.gz`. Click OK in the file browser to accept the choice of file. The file will be loaded and the first ligand is displayed in the preview. Examine the ligand by rotating (left mouse button) or zooming (mouse wheel) the structure image. Scroll through the list of ligands using the green arrows. When satisfied click Next to continue.

It is not required to change any of the options in the Adjust Parameters tab, as indicated by the green check mark next to the tab name. However, for the purposes of this tutorial we will keep the Max Molecules Per Model as 3 (consider only models containing 1,2 or 3 molecules) but increase the Models to Keep to 3 (keep the best 3 models for further review). Enter Trypsin in the `Prefix` field and check the box to `Merge Color Atoms`.

Click Next to begin building the models. A progress dialog will provide information on the model building. When all the 1159 models containing 1, 2 or 3 molecules from the dataset of 19 trypsin inhibitors have been built and compared the top 3 models are listed in the Pick Queries dialog.

5.2.3 Visualize the results

The first model in the list, 'Trypsin 1', is displayed in the preview window and is made from three of the ligands: 1G3E.pdb, 1GHZ.pdb and 1QB6.pdb. This is the model that is most representative of the dataset of 19 inhibitors as a whole.

Note that the three ligands align at one end of the model whilst the other end is described by only the single, larger molecule. Where the three ligands align closely the donor/cation/donor triad of color atoms have been merged to a single representation, instead of close color atoms from each ligand (e.g. three partially overlaid cation color atoms). Retaining multiple instances of the color atoms would serve to stress the importance of these features in the model over the other features.

Click on the second model, 'Trypsin 2'. Its name will be highlighted in blue and it will be displayed in the preview window. This is the second most preferred model and is also made of three ligands: 1H4W.pdb, 1K1I.pdb and 1QB9.pdb. Note that this is a different set of three ligands than Trypsin 1 (although it is possible for some ligands to be used in multiple high ranking models).

The third model, 'Trypsin 3' is made of 1C5T.pdb, 1GJ6.pdb and 1QBO.pdb. All three models contain one of the larger, hinged ligands and at least one of the set of smaller ligands.

5.2.4 Save the results

Place a check mark next to the name of each model to select it for export to the main vROCS interface. In this tutorial we will export all three models and save them so they can be used in further validation experiments. However, in your own work you may choose only some of the models for export.

Having checked all three models click `Finish` to close the `Build a new query` dialog. Model 'Trypsin 1' will be displayed in the main vROCS 3D window and the Welcome panel is to the left of the screen. In the Welcome panel click on `Perform a simple ROCS run`. The three models are listed as potential queries in the Inputs dialog, with 'Trypsin 1' as the active query. To save 'Trypsin 1' as a ROCS saved query (*.sq) file select `File > Save Query As...` from the main File menu. Navigate to your preferred working directory and enter `Trypisn_1.sq` as the file name. Click `OK` to save the file. In the query list select model 'Trypsin 2' and use `File > Save Query As...` to save this query as `Trypsin_2.sq`. Repeat with model 'Trypsin 3' to save a query with filename `Trypsin_3.sq`.

5.2.5 Conclusions

This concludes the tutorial "Build/edit a query using the Query Wizard". In this tutorial we used a set of 19 aligned known trypsin ligands and built three potential ROCS queries, using up to three of the ligands from the dataset. These queries best describe the dataset as a whole, based on TanimotoCombo scores of all the potential models. These models can be further validated before use, as described in the tutorial *Perform a **ROCS** validation run*.

If you have time some suggestions for further study are:

- Try changing some of the model building parameters. (Note, it is recommended to keep the maximum number of molecules in a model low (less than or equal to five), to avoid extended run times for model building and overly complex output models/ROCS queries.
- Modify the current force field in the Preferences dialog (Edit > Preferences, vROCS tab) and see whether this changes the models
- Re-run the tutorial steps with your own set of aligned ligands

5.3 Build/edit a query manually

5.3.1 Background information

In this tutorial the user will learn to manually create a new query either for saving or for use in **ROCS**. It also covers the steps required to edit or modify a saved query.

A grid shape query is available that describes a protein binding pocket. To improve the quality of results obtained for alignment to the query you wish to add some color features that are known to be important for protein-ligand binding. This tutorial will guide the user through the basics of adding color atoms and saving the resulting **ROCS** query file.

The tutorial will require approximately 15 minutes of personal time to complete.

5.3.2 Build and edit a new query

Open a new ROCS session. At the Welcome screen select the option to `Create or Edit a Query Manually`. Click `File > Open` and browse to the file `OPENEYE_DIR/data/vrocs/edit/erantag_shape.grd`. This is a grid-based shape file that represents the shape of the binding pocket for the estrogen antagonist receptor, 3ERT, that can be downloaded from the PDB (see ID for the estrogen antagonist receptor in the [PDB-IDs] citation). The file will be listed in the Shape Inventory list as "unnamed". The grid shape will display in the 3D window as an opaque shape. Click and drag the file up to the Current Query list. It is renamed as "Shape from 'erantag'". The representation in the 3D window becomes transparent.

> **Note**
>
> To use this query as-is in **vROCS** click the Use in **ROCS** button to return to the Welcome screen. The grid shape will be imported into the query list as the active query for either a simple or validation ROCS run

Several active estrogen receptor antagonists have a similar color pattern, consisting of two phenol moieties, which can act as either donor or acceptor, and a cation on a flexible chain, as shown in *Annotated OEDepict TK depiction of generalized estrogen receptor active ligand*. These are the color atoms that will be added to the query.

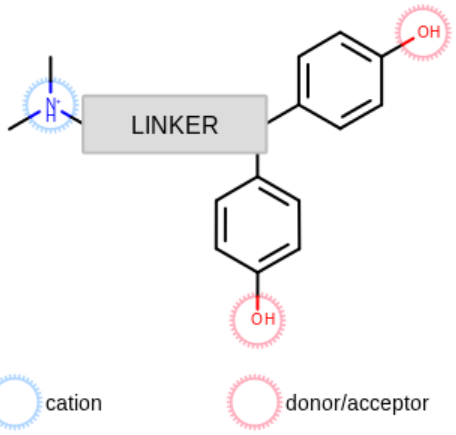


Fig. 1: **Annotated OEDepict TK depiction of generalized estrogen receptor active ligand**

Rotate the shape in the 3D window until it resembles the orientation in *Profile orientation*.

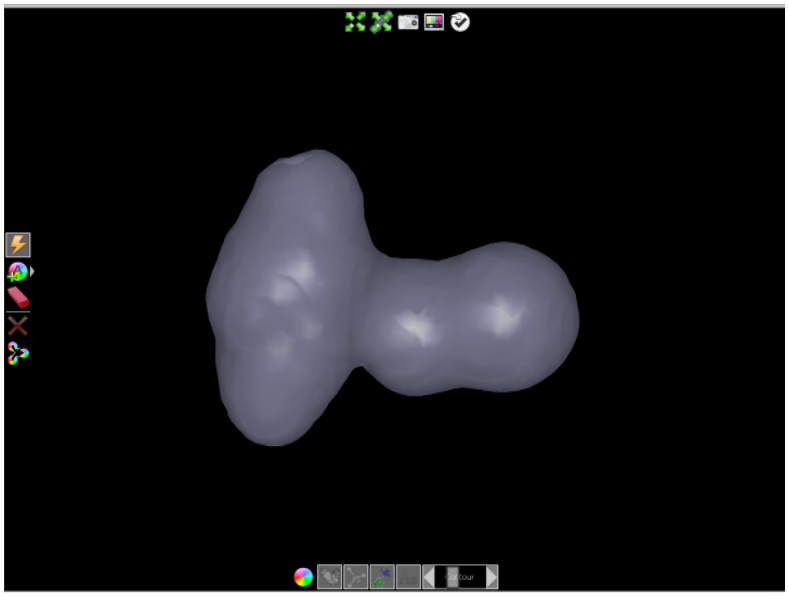


Fig. 2: **Profile orientation**

Use the Contour slider at the bottom of the 3D window to raise the contour display threshold to 1.25. Click on the `Add color atom` button at the left of the 3D Window (See *Add acceptor atoms*) and select Acceptor from the dropdown.

Click on the contour surface to add two acceptor atoms, as shown in *Add acceptor atoms*.



Fig. 3: **Add acceptor atoms**

Click on the Add color atom tool at the left of the 3D Window and select Donor from the dropdown. Click on each of the acceptor features to add two donor atoms at the same point, as shown in *Add donor atoms*.

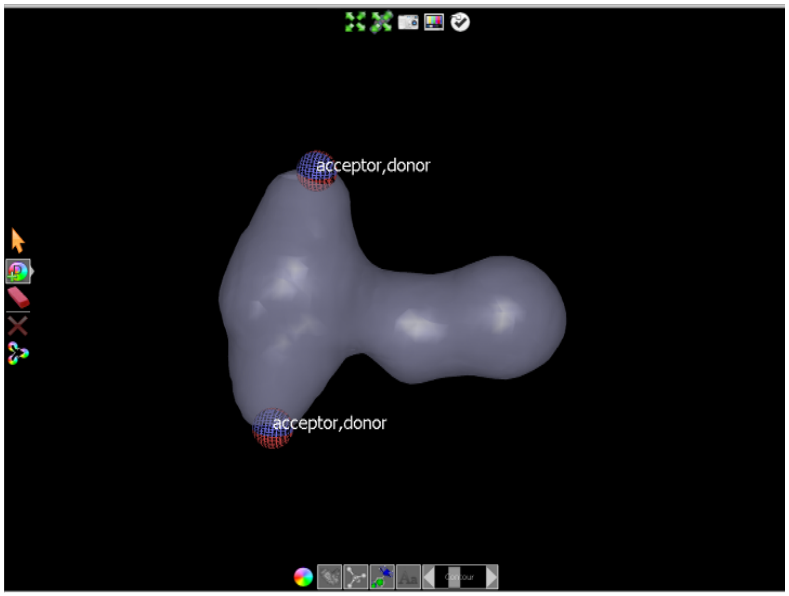


Fig. 4: **Add donor atoms**

Adjust the Contour slider back to a contour level of 1.0. Click on the Add color atom button at the left of the 3D Window and select `Cation` from the dropdown. Click on the contour surface to add a cation atom as shown in *Initial cation placement*.

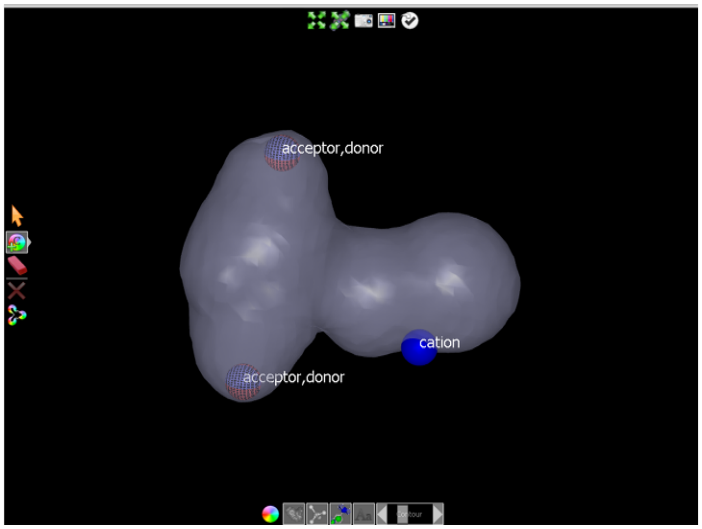


Fig. 5: **Initial cation placement**

Then CTRL-click on the surface twice more, as indicated by the orange dots in *Final cation placement*, to move the cation color atom mid-way between the three surface points.

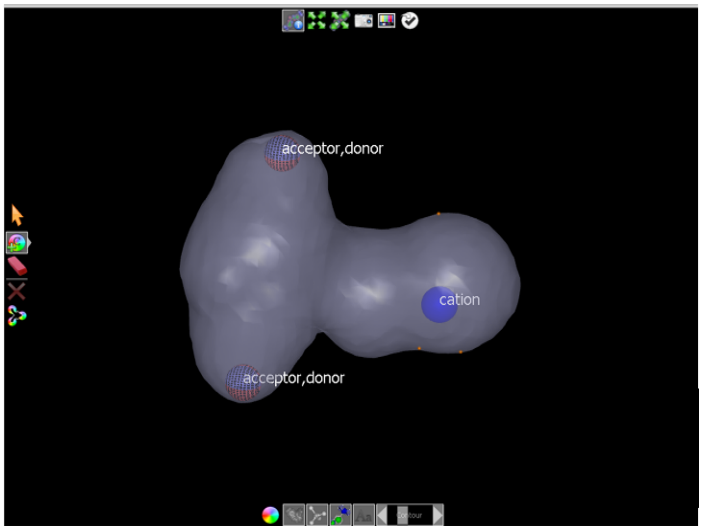


Fig. 6: **Final cation placement**

This completes adding the color points to the query. If you make any errors use the Delete tool (eraser icon) at the left of the 3D window to delete a color atom and try again.

5.3.3 Save query

In the Current Query area right–click on the query name and rename the query as "erantag_shape_color". This is the query name that will be displayed in the Query list for any ROCS runs. To save the edited query click the Save Query button or use the menu item `File > Save Query...` Choose a directory in which to save the query and give it the name `erantag_shape_color.sq`. To use this query in ROCS click on the `Done editing` icon in the 3D window or the `Accept` button to return to the Welcome screen.

5.3.4 Conclusions

This concludes the tutorial "Create or edit a query manually". In this tutorial we modified a shape grid by adding color atoms to build a more complex **ROCS** query containing information about known binding interactions. This query can be further validated before use, as described in Tutorial 3: Perform a ROCS validation run.

5.4 Perform a ROCS validation run

5.4.1 Background information

A validation run with **ROCS** allows you to select a set of active molecules and a set of decoy molecules against which to run your query. **ROCS** is run against both datasets and generates statistics evaluating how well the query discriminated between the actives and the decoys. This becomes particularly important when building a complex query. It suggests confidence levels for this query in future **ROCS** runs against databases of compounds of unknown activity.

You have just been assigned to a new research project looking for trypsin antagonists. There are no in-house lead molecules or SAR (structure activity relationship) yet. However, you have built several potential queries from published trypsin antagonists using the **vROCS** Ligand Model Builder. You would like to screen the corporate database to identify some compounds for screening in an in-house biological assay that has just come on-line. You plan to use **vROCS** to validate your queries on a sample database and identify the most selective query before running it on the larger corporate database.

After completing this tutorial the user will be aware of the steps required to set up and run a **ROCS** query validation in **vROCS** and analyze the resulting data. The tutorial will require approximately 10 minutes of personal time and 30 minutes of computer time to complete.

5.4.2 Setup ROCS run

Open a new **vROCS** session. At the Welcome screen select the option to `Perform a ROCS validation`.

In the Inputs dialog you will need to select a query, set of active molecules and set of decoy molecules. The fields that require data input are highlighted in red. Note: if you did not open a new **ROCS** session you may have some fields populated with previously used queries and datasets.

Check that the Color F.F. dropdown has Implicit Mills Dean selected as the current color force field. This is the force field that was used to build the query we will use. (Note: opening a query that was built with a force field other than that listed in the Color F.F. dropdown will result in a warning pop-up. If this should occur then click Yes to accept the change to the active color force field. This will update the Color F.F. dropdown and the list of available queries but will not change your default selection for future vROCS sessions in `Edit > Preferences`.)

In the Query input field click on `Open...` Use the file browser to navigate to either the directory where you saved the queries built in Tutorial 1 or to `OPENEYE_DIR/data/vrocs/validation/trypsin/` and open the file `Trypsin_1.sq`. This is the first of three queries selected by the Ligand Model Builder and saved, as described in the tutorial *Build/edit a query using the Query Wizard*. The files are provided for you and completion of that tutorial is not a prerequisite to this tutorial. It will be used as one of the queries for ROCS.

The Trypsin_1.sq query is shown in the 3D window as green sticks. A molecular shape (transparent gray) and color atoms are automatically assigned. For the purposes of this tutorial we will use the ligand query as-is. Tutorial 2: Build/edit a query manually covers the details of editing a query.

Repeat the steps to open queries Trypsin_2.sq and Trypsin_3.sq into the query list.

Click on the first query in the list (Trypsin_1.sq) so that it is highlighted in blue and becomes the active query for ROCS. Click on the text ROCS Run 1 in the Run Name field of the Inputs dialog and edit it to read Trypsin1.

For a validation run two databases are required. The first contains the 'active' molecules. This is a dataset that you would like to score highly against the query. The second dataset of 'decoys' should contain molecules that you predict will align poorly to the query. In a typical pharmaceutical industry setting the 'actives' might be compounds from your current SAR and the 'decoys' might be a sub-set of the corporate database. In this tutorial we will use the Trypsin actives

and decoys sets from the DUD (Directory of Useful Decoys) database (See [Huang-2006]). The decoy set is property matched to the actives (e.g. similar molecular weight, calculated LogP) but molecules have dissimilar topology in order to provide a challenging validation experiment. The more similar the actives and decoys the more confidence you can have that your query is truly selective.

The databases have been pre-prepared for this tutorial in the following manner. 1. The ligands (actives) and decoys datasets were downloaded from DUD in mol2 file format. 2. Conformers were generated using OMEGA2.2 and default settings (up to 200 conformers for each molecule).

No effort was made to clean up the dataset and remove any duplicates, filter for molecular properties (in theory this was done by DUD) and enumerate stereoisomers. These are all data preparation steps you should consider for your own dataset. However, the purpose of this tutorial is to illustrate the vROCS validation tools, not dataset preparation.

In the Inputs dialog click on the Actives option. Browse to the database of 49 known active trypsin ligands at `OPENEYE_DIR/data/vrocs/validation/trypsin/trypsin_ligands_confs.oeb.gz`. Similarly, populate the Decoys option with the database of 1664 property matched trypsin decoy ligands at `OPENEYE_DIR/data/vrocs/validation/trypsin/trypsin_decoys_confs.oeb.gz`.

Click Next to set the run options.

The Options dialog provides access to modify the main options for ROCS. It also displays the ROCS command line, should you wish to repeat this run outside the vROCS interface. Full descriptions of all the options are given in the *Validation Run options*.

The Working Directory is set by default to your vROCS installation directory. It is good practice to set a unique working directory for each run to avoid the risk of overwriting output files from old runs. Alternatively, changing the Run Prefix option would have a similar outcome. Create a working directory named `trypsin_validation` in a location of your choice and assign the prefix `trypsin1`.

Leave all the fields in the Options dialog with their default values except if you are using a computer with low memory. In that situation you may want to toggle Off the 3D View option. This will speed up the ROCS runs because the CPU is not being used for an Open GL 3D display of the aligned query and current database molecule.

Click Next to see the run summary. This lists the run name, database, working directory, etc.

5.4.3 Run ROCS in validation mode

Click Run **ROCS** to start the run.

As the run progresses the 3D window fills the screen. The query is shown and the database molecules scroll by in their alignment with the query. On the right hand side of the screen the five current top scoring molecules are shown in 2D depiction, together with their TanimotoCombo score (or whichever score type was chosen in the Run Set-up Options dialog). A progress bar at the bottom of the screen indicates how far the run has proceeded. If the 3D view was selected Off then text-based progress information will be displayed.

This run requires about 10 minutes. The relatively slow run speed is due to the large number of color features in the query used. Manually editing the query to remove some of the color features does result in some run time speed up. See the section *Editing ROCS queries in vROCS* and the tutorial *Build/edit a query manually* for details on how to accomplish this.

Repeat the steps to set-up and run query Trypsin_2.sq with run name Trypsin2 and prefix `trypsin2` and Trypsin_3.sq with run name Trypsin3 and prefix `trypsin3`.

5.4.4 Visualize results

Once the three runs are complete you will see results listed for each run on a separate tab in the results spreadsheet at the bottom of the screen. Click on the tab name for the Trypsin1 run to view the results of the first run. On the left side of the screen navigate to the Run Set-up Inputs dialog and select/highlight the query Trypsin_1.sq in the Query list. This action displays the query in the 3D window.

> **Note**
>
> In the case that 3D View was checked off during the runs click Done in the run progress informational panel to restart the 3D window and display the query.

The spreadsheet lists the query and the top 20 scoring conformers ranked based on TanimotoCombo score (or whichever score type was chosen in the Run Set-up Options dialog). The query is also listed as the first entry as an aid to comparison. Make the query visible by clicking in the visibility column (green circle) next to its entry. Select individual results by clicking on their name and scroll up/down the list with the arrow keys to check the alignments look reasonable.

Click on the Show the statistics icon at the far right of the spreadsheet to display the statistics panel for Trypsin1. All the database molecules used in the search are included in these calculations, not just the top 20 that were listed in the spreadsheet for visual inspection. AUC for the ROC curve and enrichment at 0.5%, 1% and 2% are listed for the run, together with their upper and lower 95% confidence limits. The ROC curve is also displayed. An AUC of 0.855 indicates a query that is predictive and well able to separate the actives from the decoys. Change the score used in the calculations from TanimotoCombo to ShapeTanimoto in the Metric dropdown. This will update the ROC plot and statistics to reflect that scoring metric. The AUC is now 0.680, indicating that shape alone is a less selective metric for identifying trypsin actives from decoys and that color is an important addition to the query.

Inspect the score histogram plot by selecting Score Histogram in the Chart dropdown. This displays color coded histograms for the score distribution within the active and decoy databases. A more selective query will have a score distribution with higher frequency of obtaining a higher score (i.e. further to the right of the plot). As before, changing the score metric will update this chart.

5.4.5 Compare results from multiple runs

With the Trypsin1 results tab selected, making Trypsin1 the Base run, select Trypsin2 and Trypsin3 from the Compare to dropdown. Choose ROC plot from the Chart dropdown and TanimotoCombo from the Metric dropdown. The display should look similar to the image below:

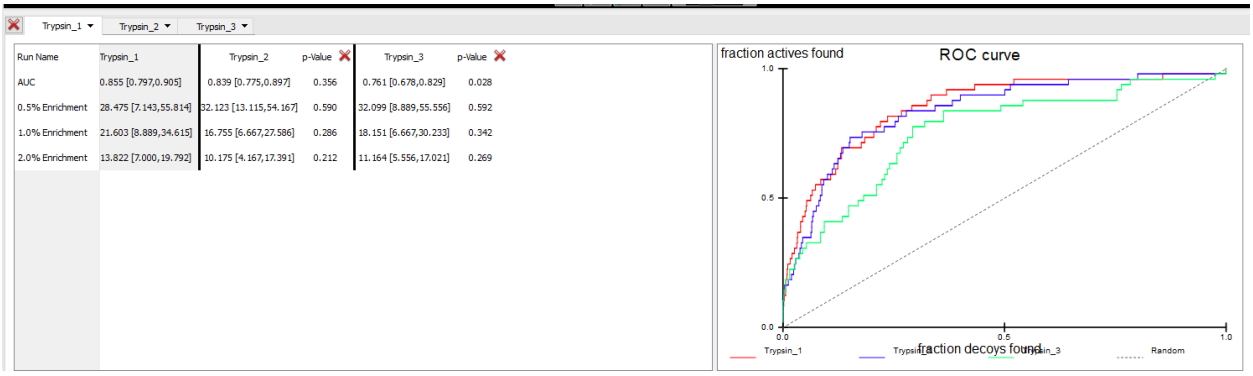


Fig. 7: **Comparison of runs Trypsin1 (Base) with Trypsin2 and Trypsin3**

At a glance the ROC plot shows that Trypsin1 has the highest AUC, suggesting it is the most selective query over the entire database search. Trypsin2 has a similar AUC to that of Trypsin1, suggesting that it is also an equally selective query. However, this is not completely supported by the p-values. Trypsin2 has a p-value of 0.370 when compared to the Trypsin1 base run, indicating that the Trypsin1 query is probably the most selective of the two. Run Trypsin3 has the lowest AUC and a p-value of 0.024 indicating that there is a very low probability of query Trypsin_3.sq being more selective than Trypsin1. The early enrichments for all three runs are comparable, especially when the ±95% confidence limits for each run are compared. However, Trypsin1 has a slightly higher early enrichment (except at 5%), and can probably be considered superior at ranking most of the actives very highly compared to both Trypsin2 and Trypsin3 as the early enrichment p-values for those are mediocre.

5.4.6 Save results

An image of the ROC plot can be saved and used in presentations. Click on the Trypsin1 run-name tab to make it highlighted as the active view. The ROC plot currently shows three curves, one each for runs Trypsin1 as well as Trypsin2 and Trypsin3. In the `Choose stats to save...` dropdown select the option to Plot data. In the dialog increase the dimensions (resolution) to 500 in the first box. Since Maintain original aspect is checked on the second dimension will update automatically. Use the Browse option to select a directory of your choice. Change the name of the file form screenshot.png to trypsin_ROC_plot.png. Click OK to save the image file.

It can also be useful to save a copy of the statistical comparison in the spreadsheet. Click on the `Choose stats to save...` dropdown and select the option for Spreadsheet. In the dialog navigate to a directory of your choice and name the file trypsin_spreadsheet.csv. A file will be saved containing the AUC and enrichment values, 95% confidence limits and p-values for the three runs currently being compared in the Trypsin1 tab.

5.4.7 Conclusions

From the comparison we can conclude that Trypsin_1.sq is the most selective query of the three in this validation. When the query models were built Trypsin_1.sq was also ranked highest by the Ligand Model Builder. However, that was on a rigid dataset of the 19 single conformer candidates for the model building. This validation was carried out on a multi-conformer set of active and decoy ligands which is closer to the real life scenario under which the query will be used i.e. the ROCS run of the corporate database to identify potential screening candidates.

This concludes the tutorial "Perform a ROCS validation".

If you have time some suggestions for further study are:

- Try comparing the runs above to the Lingos 2D similarity metric. Lingos is a useful metric and often provides good selectivity
- Run a validation of the estrogen receptor antagonist shape + color query built in the tutorial *Create or edit a query manually*. Compare it against the shape alone and the native ligand. The following files are provided in the `OPENEYE_DIR/data/vrocs/validation/erantag/` directory.
  - erantag_grid.sq – a ROCS saved query file of the grid shape
  - erantag_color.sq – a ROCS saved query file of the grid shape with added color points
  - 3ERT_lig.ent – the ligand from the 3ERT estrogen receptor crystal structure
  - er_antagonist_ligands_confs.oeb.gb – the dataset of active ligands
  - er_antagonist_decoys_confs.oeb.gb – the dataset of decoy molecules

5.5 Perform a simple ROCS run

5.5.1 Background information

A simple **ROCS** run allows you to choose a query and a database to search. It will search the database for the closest matches to your query and load your results afterwards. The majority of runs with **ROCS** are simple runs using a molecular shape-based query, with or without color atoms. Other query types can also be used, in a similar fashion, as described in the section *ROCS shape query sources*.

You are working on a project for which the current lead series has a problem with CYP2C9 metabolism. You are hoping to suggest some synthesis candidates that have a low risk for CYP2C9 metabolism. You recently saw a paper by Sykes et. al. ([Sykes-2008]) in which they use **ROCS** to validate a database of 70 known Cytochrome P450 2C9 substrates. You plan to use **ROCS** similarity to compounds in this database to predict whether your ideas will be metabolized by CYP2C9. One of your ideas is shown in the depiction below. You will use **ROCS** through the vROCS interface.

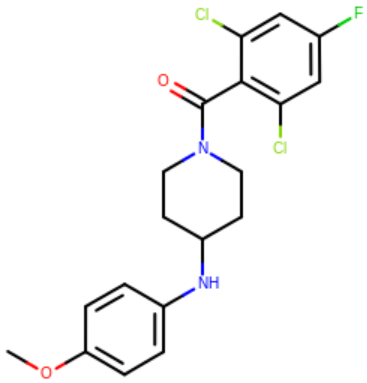


Fig. 8: **OEDepict TK depiction of the synthesis candidate molecule and ROCS query**

This tutorial will provide the user with the background required to set up a simple **ROCS** run in the **vROCS** interface from a saved query and analyze the results. The tutorial will require approximately 10 minutes of personal time and 1 minute of computer time to complete.

5.5.2 Setup ROCS run

Open a new **ROCS** session. At the Welcome screen select the option to `Start a simple ROCS run`.

In the Inputs dialog you will need to select a query and a database to search. The fields that require data input are highlighted in red. Note: if you did not open a new **ROCS** session you may have some fields populated with previously used queries and datasets.

In the Query input field click on `Open query...` Use the file browser to navigate to `OPENEYE_DIR/data/vrocs/simple` and open the file `molecule_idea.sq`. The query was prepared by generating **OMEGA** conformers from a SMILES file of the molecule. Default OMEGA parameters were used and the lowest energy conformer was selected as the query and saved as a **ROCS** saved query file named `molecule_idea.sq`.

The molecule is shown in the 3D window as green sticks. A molecular shape (transparent gray) and color atoms are automatically assigned. For the purposes of this tutorial we will use the ligand query as-is. The tutorial *Create or edit a query* manually covers the details of editing a query.

Click on the text `ROCS Run (1)` in the Run Name field and edit it to read `Molecule_idea`.

In the Inputs dialog click on the `Database` option. Browse to the database at `OPENEYE_DIR/data/vrocs/simple` and open the database file called `jmedchem_database.oeb.gz`. The database was prepared from a SMILES file of the 70 CYP2C9 substrates, for which OMEGA conformers were generated with default parameters.

Click Next to set the run options.

In a location of your choice create a working directory called `simple`.

In the literature, the authors obtained best results for their validation using the parameter `Random Starts` = 50 instead of the default Inertial start. This allows more conformational space to be sampled but does increase the time taken by the run.

> **Note**
>
> Inertial starts will give identical results from run to run. Using Random starts may result in slight variation in score (e.g. 0.750 vs 0.749) if a run is repeated.

Click on the `Random` radio button for Start type and type 50 in the input box. Leave all other options as default.

Click `Next` to see the run summary. This lists the run name, database, working directory, etc. Click on the `Command Line...` option to view the full command line for the run. This can be useful to set up future ROCS runs outside the vROCS interface e.g. to make use of distributed computing.

5.5.3 Run a simple ROCS run

Click Run ROCS to start the run.

As the run progresses the 3D window fills the screen. The query is shown and the 70 database molecules scroll by in their alignment with the query. On the right hand side of the screen the five current top scoring molecules are shown in 2D depiction, together with their score (TanimotoCombo by default or whichever score type was chosen in the Options dialog). A progress bar at the bottom of the screen indicates how far the run has proceeded.

This run requires around 40 seconds.

5.5.4 Visualize and save results

When the run is complete the query will display in the 3D window and the 20 top scoring hits, based on the score chosen in the Score by field during the run set-up (default is TanimotoCombo), are listed in the results spreadsheet. This is intended as a quick summary view. To examine all the hits then the full hitlist, named `rocs_hits_1.oeb.gz` (default filename) is automatically saved in the working directory for viewing with tools such as **VIDA**. The query is also listed as an aid to comparison. Make the query visible by clicking in the visibility column (green circle) next to its entry. Click on a molecule name in the spreadsheet to display its best overlay with the query and scroll through the molecules to view each one.

Scores for all the available scoring functions are shown in the spreadsheet. Clicking on any one of the column headings will sort by that score and clicking multiple times will switch between ascending and descending. The 20 hits listed in the spreadsheet will change to reflect the sort preference (score and ascending (bottom 20) or descending (top 20 hits)).

With descending TanimotoCombo selected as the sort order look at the TanimotoCombo scores for the top scoring ligand. One of the Suprofen conformer has a TanimotoCombo score of approximately 1.0. (Because Random starts were used your score might be slightly different.) The authors found that 0.99 was a reasonable threshold for a "reliability" cutoff for alignments that reproduced known binding poses in CYP2C9. This molecule satisfies that requirement.

5.5.5 Conclusions

The query molecule has a relatively high shape similarity to a Suprofen conformer. The Shape Tanimoto score of that conformer is one of the highest for the entire dataset. However, low color similarity for the Suprofen conformer results in an overall TanimotoCombo score around 1.0. Given the low TanimotoCombo score to the known CYP2C9 ligands one might conclude that this molecule has a low risk of being metabolized by CYP2C9 and is worthy of further investigation.

This concludes the tutorial "Perform a simple ROCS run" which guides the user through the basics of running **ROCS** through the **vROCS** interface and visualizing the results.

5.6 Perform a ROCS run from the command line

5.6.1 Background information

A simple **ROCS** run allows you to choose a query and a database to search. It will search the database for the closest matches to your query and load your results afterwards. The majority of runs with **ROCS** are simple runs using a molecular shape-based query, with or without color atoms. Other query types can also be used, in a similar fashion, as described in section *ROCS shape query sources*.

You are working on a project for which the current lead series has a problem with CYP2C9 metabolism. You are hoping to suggest some synthesis candidates that have a low risk for CYP2C9 metabolism. You recently saw a paper by Sykes

et. al ([Sykes-2008]) in which they use ROCS to validate a database of 70 known Cytochrome P450 2C9 substrates. You plan to use ROCS similarity to compounds in this database to predict whether your ideas will be metabolized by CYP2C9. One of your ideas is shown in the *OEDepict TK depiction below*. You plan to use ROCS command line.

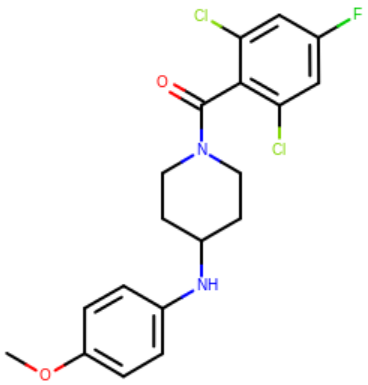


Fig. 9: **OEDepict TK depiction of the synthesis candidate molecule and ROCS query**

This tutorial will provide the user with the background required to set up a simple **ROCS** run in the command line ROCS interface from a saved query and analyze the results. The tutorial will require approximately 10 minutes of personal time and 1 minute of computer time to complete. This is a repeat of the simple run the vROCS tutorial *Perform a simple ROCS run*.

5.6.2 Setup ROCS run

Open a command prompt and navigate to a working directory of your choice. At the very minimum **ROCS** requires an input query file and an input database file. There are numerous additional parameters (e.g. specifying a custom color force field file) that can be added to the command, as well.

In the **vROCS** tutorial *Perform a simple ROCS run*, the command line provided to **ROCS** (on Windows) is:

```
rocs.bat \
-query molecule_idea.sq \
-dbase jmedchem_database.oeb.gz \
-prefix rocs \
-outputdir C:/Users/username \
-besthits 500 \
-rankby TanimotoCombo \
-shapeonly false \
-randomstarts 50 \
-opt true \
-scoreonly false \
-optchem true \
```

This is found in the `Command line...` information field of the run set-up dialog. Most of these parameters are default. The only one that was changed was –randomstarts 50, instead of the default Inertial start.

5.6.3 Run a simple ROCS run

At the command prompt type the following command to run ROCS.

> **Note**
>
> If the query and database files are in your current working directory you do not need to specify a full directory path to the files. If not, specify the full path to the query and database files.

```
rocs \
-query \
OPENEYE_DIR/data/vrocs/simple/molecule_idea.sq \
-dbase \
OPENEYE_DIR/data/vrocs/simple/jmedchem_database.oeb.gz \
-prefix tutorial5 \
-randomstarts 50
```

This run requires about 40 seconds.

5.6.4 Visualize and save results

ROCS automatically saves the following files using the prefix given above in the naming convention. If no prefix is given then the default prefix is rocs.

- prefix_parm
- prefix.log
- prefix_ref.sq
- prefix_1.rpt
- prefix_1.status
- prefix_hits_1.sdf

The .rpt file contains all the score data in a tab delimited format. The .sdf contains all the database hits in their aligned conformation, together with scores. This sd file is suitable for opening in VIDA for further examination.

5.6.5 Conclusions

This concludes the tutorial "Perform a **ROCS** run from the command line" which guides the user through the basics of running a simple **ROCS** run through the command line interface and visualizing the results.

CHAPTER

SIX

THEORY

6.1 Shape Theory

What do we mean by shape? The word is often used without consideration of precise meaning but in this document we shall be very clear as to the definition of shape. Two entities will have the same shape if their volumes exactly correspond. The more the volumes differ, the more the shapes will differ. We will give a precise mathematical exposition below, but it is worth noting even at this most basic level shape is defined as a relative quantity, depending on references to other shapes. In this we differ from approaches that attempt to provide absolute, canonical, *shapes* by which to categorize molecules.

What do we mean by *volume*? A volume is any scalar field. This means a function that has a single number, or *scalar*, value at each point in space. The *special case* for the common understanding of volume is a specific scalar field that has a value of one inside an object and zero outside. The volume of a scalar field is:

$$V(\text{volume}) = \int f(x,y,z)dv$$

The volume function, **f**, is also referred to as the *characteristic* function. When the characteristic function corresponds to the common definition of a volume field this integral corresponds to what is commonly expected by volume. However, we are not restricted to such simple functions and can still calculate a **V**. In general the volume of a scalar field is a *contraction* of the information represented by that characteristic function. It is more precisely referred to as the zeroth-order contraction, or *moment*. We will discuss other moments and their uses later, but one immediate observation is that two objects cannot have the same shape if their volumes are not the same. The converse is obviously not true. Rather, two objects can have the same volume and not have the same shape. Volume is typical, therefore, of most contractions of information.

We can now write down a precise definition of shape similarity. Consider the integral:

$$S_1 = \int |f(x,y,z) - g(x,y,z)|dV$$

where **f** and **g** are different characteristic functions. If this integral is zero then **f** and **g** are actually the same function and therefore correspond to the same shape. The larger the integral, the more different the shapes defined by **f** and **g**. It defines a metric quantity between the two fields **f** and **g**. The word *metric* is used loosely to mean *shape*, but here we mean the precise mathematical definition: *i.e.* a distance that is 1) always positive, 2) zero if and only if two entities are identical and 3) that obeys the triangle inequality. The triangle inequality states that if entity A is distance x from entity B and B is distance y from entity C then the distance between A and C is bounded by **|x-y|** and **|x+y|**. The type of comparison shown in S_1 is referred to as an L_1 metric. Another metric is the S_2 metric:

$$S_2 = \sqrt{\int [f(x,y,z) - g(x,y,z)]^2 dV}$$

Multiplying the terms in the integral out gives:

$$S_2^2 = \int f(x,y,z)^2 dV + \int g(x,y,z)^2 dV - 2\int f(x,y,z)g(x,y,z)dV$$

This is the fundamental equation for shape comparison. We rewrite it as:

$$S_{f,g} = I_f + I_g - 2O_{f,g}$$

The *I* terms are the self-volume overlaps of each entity (for our purposes - molecule), while the *O* term is the overlap between the two functions. They constitute the three terms we need to compare the shapes of two fields. The *I* terms are independent of orientation but not *O*. Finding the orientation that maximizes *O*, and hence minimizes *S_{f,g}*, is equivalent to finding the best overlay between the two objects (a quantity that has its own, distinct metric properties). We also note here that the quantity referred to as a Tanimoto coefficient may be derived by recombining *I*'s and *O* so:

$$Tanimoto_{f,g} = \frac{O_{f,g}}{I_f + I_g - O_{f,g}}$$

Tanimoto coefficients will be familiar to those who use them for bitvector fingerprint comparison. An alternative measure is the Tversky coefficient, also mostly used for similarity between bitvector fingerprints. Similarly to the Tanimoto coefficient above, we can define a shape Tversky measure. The base equation for the Tversky coefficient is:

$$Tversky_{f,g} = \frac{O_{f,g}}{\alpha I_f + \beta I_g}$$

Normally, *alpha + beta = 1*, and for our current use, *alpha* is chosen to be 0.95. Since this introduces an asymmetry, the Tversky calculation depends on which molecule's self-overlap has the *alpha* pre-factor. ROCS calculates two Tversky values, one with the query molecule with *alpha* as the pre-factor and a second with the database molecule with *alpha* as the pre-factor. Also, note that since shape is a field property, instead of a simple scalar like a bitvector, shape Tversky can be larger than 1.0 since the overlap *O_{f,g}* can be larger than a molecule's self-overlap, *I_f*.

The OpenEye Shape Toolkit is a set of calculational objects designed to facilitate the calculation of these field-metric quantities. ROCS is an application built on top of the Shape toolkit.

6.1.1 Shape Characteristics and the Use of Gaussians

Molecules are traditionally viewed as a set of fused spheres, sometimes referred to as the CPK model. The common view of molecular volume is then of a characteristic function that is one (1) inside at least one sphere and zero (0) outside. How do we calculate the volume of such a seemingly simple function? The volume of a single sphere is *(4pi r^3)/3* but the complication for two fused spheres is that we have to account for the shared volume and not count it twice. For more than two atoms, there are triple intersections that must be added back in if we have removed the three pairs of intersections. The general formula for N spheres that explicitly calculates the volume of every level of overlap and its correct contribution is:

$$V = 1 - \int \prod_i^N (1 - f_i) dv$$

This is easy to write, not so easy to solve because the analytic formulae for overlaps of increasing order are highly non-trivial (although they have been derived to arbitrary order). It is fair to say that this has hindered the development of shape comparison in many ways. Attempts to use analytic formulae led to very slow programs and approximate methods, for instance using grids of points that are turned *in* or *out* by each sphere, do not give smooth gradients required for minimization. Brian Masek (AstraZeneca) was the first to attempt to optimize overlaps of molecules using the analytic approach [Masek-1993]. His program would take several minutes per minimization. In addition it would often suffer from a common problem when using functions that vary sharply (such as solid spheres): it would often get stuck in local minima. Nevertheless, Brian did have encouraging success using this method to find similarities not obvious from chemical structure.

The conceptual breakthrough in shape comparison came in 1995 in a paper by Andrew Grant (AstraZeneca) and Barry Pickup (University of Sheffield) ([Grant-1995], [Grant-1996], [Grant-1997]). They showed that if one let go of the concept of the characteristic function being binary, and instead use a sum of continuous functions, *i.e.* a Gaussian, that the solid-sphere volume, could be recovered to high accuracy (typically ~0.1%). A sphere has one defining parameter, its radius, whereas a Gaussian has two defining parameters, its prefactor, *p*, and its width, *w*:

$$pe^{-wx^2}$$

Grant and Pickup found that by fixing *p* to 2.7 and setting *w* for each atom such that the volume integral for each atom agreed with its solid-sphere volume, they achieved remarkable precision. In addition, the overlap terms between any two atoms, and hence any higher-order overlaps, are all Gaussian functions themselves because of the Gaussian Contraction formula (shown here for one spatial variable):

$$\int e^{a(x-x_i)^2} e^{b(x-x_i)^2} = \int e^{(a+b)(x-x_i)^2}$$

i.e. two atomic-Gaussians overlap to produce another Gaussian. Likewise, a three atomic-Gaussian overlap is that of an overlap-Gaussian with an atomic-Gaussian, hence another Gaussian. The simplicity of these formulae and the formula for the volume of each individual Gaussian leads to very efficient algorithms for the calculation of the volume of a molecule so represented (the OpenEye method calculates several thousand volumes per second while calculating intersections up to sixth order).

In addition to simple calculation of molecular volume, which is the zeroth-order moment of the characteristic function, the ease of evaluation of intersections allows for accurate calculation of high-order moments: called the *steric multipoles*. For instance, if the product formulae for atomic and intersection Gaussians yields *n* Gaussians, the first order moments are:

$$M_{1,x} = \sum_{i=1}^{n} \int x e^{a_i|(x-x_i)^2+(y-y_i)^2+(z-z_i)^2|}$$

$$M_{1,y} = \sum_{i=1}^{n} \int y e^{a_i|(x-x_i)^2+(y-y_i)^2+(z-z_i)^2|}$$

$$M_{1,z} = \sum_{i=1}^{n} \int z e^{a_i|(x-x_i)^2+(y-y_i)^2+(z-z_i)^2|}$$

These integrals are easy to solve and their sum can be set to zero by an appropriate choice of origin: the center of *mass* for the sum of Gaussians. Second-order moments are found from integrals of the type:

$$M_{2,PQ} = \sum_{i=1}^{n} \int PQ e^{a_i|(x-x_i)^2+(y-y_i)^2+(z-z_i)^2|}$$

where P and Q are chosen from (x,y,z), *e.g.* x^2, xy etc.

These moments can be thought of as a symmetric 3*3 matrix which we refer to as the *mass matrix*. Rotating or translating the molecule will change the moments and the transform that sets the first-order moments to zero and diagonalizes the mass-matrix puts the molecule into its *inertial frame*. By convention we assign the x-axis to the largest eigenvector of the mass matrix, y-axis to the median eigenvector, z-axis to the smallest. Note that this orientation is still not uniquely defined: 180 degree rotations around any axis also diagonalize the mass-matrix. The eight (2^3) possible transforms that can be generated by combinations of such rotations actually lead to four unique inertial orientations.

If a molecule is aligned to its inertial frame, all higher-order steric multipoles become invariant, ignoring certain sign-changes from the four-fold degeneracy of the inertial frame. As such they, as well as the second-order moments, are shape *descriptors*. They are still contractions of the information contained in the characteristic field, *i.e.* two molecules can have the same steric moments and yet have different shapes. (Moments are *complete* in that if we calculate them to infinite order they do exactly define volume but this is seldom a practical approach!) Nevertheless, they do contain useful information and can be used as a rapid, approximate, filter for shape similarity.

The same advantages that allow for the calculation of molecular volume carry over to the calculation of molecular volume overlap. The overlap of volumes are Gaussian contractions, easily tabulated and efficiently retrieved. Andy re-wrote Brian's program and obtained an order-of-magnitude improvement in performance as well as another remarkable observation: if the starting orientation of each molecule is that given from the inertial frame then very few "false" minima are produced. The smoothness of the Gaussian characteristic function is enough to overcome the problems with convergence in Brian's program. The four possible "inertial" starting points were enough to find the best, global, overlay between two molecules. This observation and the Gaussian approach are the basis of the OpenEye Shape Toolkit and ROCS program for rapid shape overlay.

But note, despite the algorithmic advantages, a correlation with common perception has been lost. Because the pre-factor of each atomic Gaussian is not unity, the characteristic function does not correspond to the inside/outside description with which we are most comfortable. In the Gaussian model all points in space are to some degree inside and to some degree outside. That is, the Gaussian model typically shows about 0.1% error with respect to the solid sphere model due to the fact that is includes a portion of all points in space inside the volume.

6.2 Color Features

In addition to shape-alignments **ROCS**, optionally, considers chemistry alignment, known as 'color'. User specified definitions of chemistry can be included in the superposition and similarity analysis process to facilitate the identification of those compounds that are similar both in shape and chemistry.

Color atoms are described as Gaussians and displayed in **vROCS** as colored spheres. The Gaussian for a color atom is relatively hard with a steep gradient. *Figure: Hard vs. Soft Gaussians* illustrates hard vs fuzzy Gaussians. Both Gaussians in the figure represent the same volume as the sphere. However, the hard Gaussian, with the steep gradient, reaches a probability of zero (0) within the radius of the sphere. The color features are either matched, if they fall within the sphere radius, or not matched. In the case of the fuzzy Gaussian there are areas outside the volume of the sphere (the area under the curve indicated by the two arrows) where the Gaussian probability is greater than zero. This would allow color features to match even when they align well outside the sphere representing the color atom. That situation would lead to less precise alignments and, for that reason, the 'hard' Gaussian is employed.

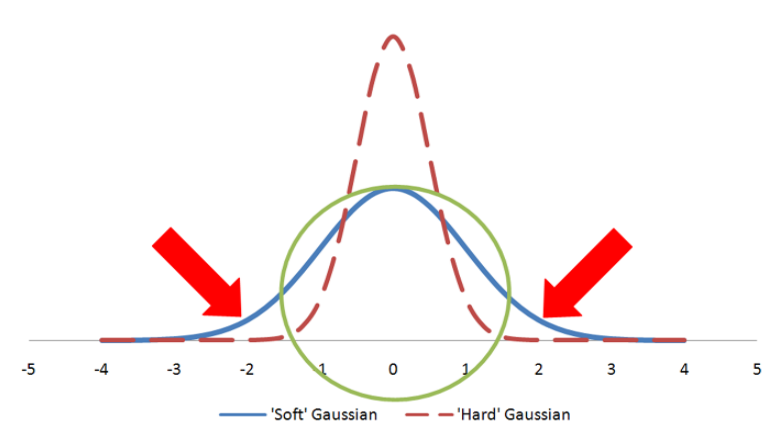


Fig. 1: **Hard vs. Soft Gaussians**
A sphere described by two different Gaussian functions. The 'hard' Gaussian (dashed) is the one employed by ROCS to approximate a color atom sphere.

ROCS comes pre-loaded with two color force fields, Implicit Mills Dean (default) and Explicit Mills Dean. These are described in color force field files (*.cff) located in the **ROCS** data directory. A sample color force field file, `sample.cff`, is also provided as a template for a user's own custom force field. The desired force field file is supplied to **ROCS** either at the command line using the `-chemff` command or in the **vROCS** Preferences menu. Further information on editing color force field files is given in the section *Color Force Field*.

The color force field is used to measure chemical similarity between the query and the database molecule and to refine shape-based overlays. The color force field file describes:

- Color atom types
- Which functional groups the color atoms should be applied to. **ROCS** uses only the heavy atoms of molecules, hydrogens are ignored.
- Whether the interaction between color atoms is attractive or repulsive. Interactions between color atoms of the

same type are always attractive. The weight term describes the strength of the interaction, relative to the shape gradients and the range term affects the range of the interaction.

The color features described in the Implicit and Explicit Mills Dean color force field files include:

Donor
Functional groups that can act as H-bond donors e.g. acid-OH

Acceptor
Functional groups that can act as H-bond acceptors e.g. carboxylate

Anion
Functional groups with either localized or delocalized negative charge e.g. tetrazole

Cation
Functional groups with either localized or delocalized positive charge e.g. guanidinium

Hydrophobe
Terminal or nonterminal aliphatic groups, including Br and I

Rings
Rings of defined size e.g. 4-7 atoms

A custom force field file can include other features that you define e.g. positive, negative, carbonyl_linker, metal_binder. For each color atom type a set of SMARTS is used to define the specific functional groups to which the color atom will be applied. The Implicit and Explicit Mills Dean force fields differ in these functional group definitions. For example, the Explicit Mills Dean force field allows a primary amine to be an acceptor as well as a donor whereas it is a donor only in the Implicit Mills Dean force field.

The color force field can also be used for post-shape scoring either alone, e.g. ColorTanimoto and Color Tversky, or in combination with shape scores, e.g. TanimotoCombo and TverskyCombo. Some additional scores are available with color:

- ScaledColor
- ComboScore
- ColorScore

These scores are defined in the section Report File.

6.3 Color Force Field

The chemical force field can be used to measure chemical complementarity, and to refine shape based superpositions based on chemical similarity. The CFF is composed of SMARTS rules that determine chemical centers, plus rules to determine how such centers interact.

6.3.1 Default Color Files

Two color force fields, ImplicitMillsDean and ExplicitMillsDean, are built into ROCS. Both these force fields define six similar TYPE color force-fields. The types are hydrogen-bond donors, hydrogen-bond acceptors, hydrophobes, anions, cations, and rings. The ImplicitMillsDean force field is recommended.

ImplicitMillsDean includes a simple pKa model that assumes pH=7. It defines cations, anions, donors and acceptors in such a way that they will be assigned the appropriate value independent of the protonation state in the dbase or query file. For example, if a molecule contains a carboxylate, ImplicitMillsDean will consider it an anionic center independent of whether it is protonated or deprotonated in the dbase file. This is convenient for searching databases which have not had careful curation of their protonation states. The ExplicitMillsDean file has a similar overall interaction model, however, it does not include a pKa model. It interprets the protonation and charge state of each molecule exactly as it is in the database. Thus, if a sulfate is protonated and neutral, it will not consider it an anion.

The hydrogen-bond models in both ImplicitMillsDean and ExplicitMillsDean are extensions of the original model presented by Mills and Dean ([Mills-Dean-1996]). They both have donors and acceptors segregated into strong, moderate and weak categories.

6.3.2 Color File Format

As an alternative to the built-in force fields, the user can define a new color force field using the format described in this section. The following is a simplified example of a color force field specification.

```
DEFINE hetero [#7,#8,#15,#16]
DEFINE notNearHetero [!#1;!$($hetero);!$(*[$hetero])]
#
#
TYPE donor
TYPE acceptor
TYPE rings
TYPE positive
TYPE negative
TYPE structural
#
#
PATTERN donor [$hetero;H]
PATTERN acceptor [#8&!$(*~N~[OD1]),#7&H0;!$([D4]);!$([D3]-*=,:[$hetero])]
PATTERN rings [R]~1~[R]~[R]~[R]1
PATTERN rings [R]~1~[R]~[R]~[R]~[R]1
PATTERN rings [R]~1~[R]~[R]~[R]~[R]~[R]1
PATTERN rings [R]~1~[R]~[R]~[R]~[R]~[R]~[R]1
PATTERN positive [+,$([N;!$(*-*=O)])]
PATTERN negative [-]
PATTERN negative [OD1+0]-[!#7D3]~[OD1+0]
PATTERN negative [OD1+0]-[!#7D4](~[OD1+0])~[OD1+0]
PATTERN structural [$notNearHetero]
#
#
INTERACTION donor donor attractive gaussian weight=1.0 radius=1.0
INTERACTION acceptor acceptor attractive gaussian weight=1.0 radius=1.0
INTERACTION rings rings attractive gaussian weight=1.0 radius=1.0
INTERACTION positive positive attractive gaussian weight=1.0 radius=1.0
INTERACTION negative negative attractive gaussian weight=1.0 radius=1.0
INTERACTION structural structural attractive gaussian weight=1.0 radius=1.0
```

There are four basic keywords in a cff file: DEFINE, TYPE, PATTERN, and INTERACTION. The TYPE field can be any user-defined term. TYPES can be any user-specified string such as “donor”, “acceptor”, “lipophilic anion” etc. The PATTERN keyword is used to associate SMARTS patterns with these types. There is no restriction on the number of patterns that can be associated with a user defined type. The position in Cartesian space of the PATTERN is taken as the average of the coordinates of the atoms that match the SMARTS pattern. If the desired location of the PATTERN is on a single atom of a larger SMARTS pattern recursive SMARTS (written as ‘[$(SMARTS)]’ can be used to this effect. Only the first atom in a recursive SMARTS pattern ‘matches’ the molecule, and the rest of the SMARTS pattern defines an environment. By writing a SMARTS pattern in recursive notation the location of the PATTERN will be taken as the atomic position of the first matching atom in the pattern. In order to simplify both reading and writing SMARTS, intermediate SMARTS can be associated with words using the DEFINE keyword. Once defined, these words can then be used as atom primitives in subsequent SMARTS patterns with the $ prefix (see “DEFINE hetero” and “PATTERN donor” above).

Interactions between types are associated with the INTERACTION keyword. Two user-defined types must be listed,

and whether their interaction is attractive or repulsive. The height and radius can be modified by keywords WEIGHT and RADIUS. At present, the only alternative to a Gaussian decay is invoked by the DISCRETE keyword. A discrete interaction contributes all of WEIGHT if the inter-type distance is less than RADIUS, or zero. Since it is not differentiable it makes no contribution to optimization (i.e. because the gradient of a DISCRETE function is 0 or infinite).

6.4 Similarity Measures

Measuring molecular similarity or dissimilarity has two basic components: the representation of molecular characteristics (such as shape and color) and the similarity coefficient that is used to quantify the degree of resemblance between two such representations. Different similarity coefficients quantify different types of structural resemblance.

The table below defines the basic terms that are used in shape-based similarity calculations.

Table 1: **Basic components of similarity calculation**

Symbol	Description
$selfA$	Self-overlap or self-color score for molecule A
$selfB$	Self-overlap or self-color score for molecule B
$overlapAB$	Overlap or color score between molecules A and B

6.4.1 Tanimoto

Formula:

$$Tanimoto_{A,B} = \frac{overlapAB}{selfA+selfB-overlapAB}$$

The Tanimoto similarity measure is symmetric and always has a value between 0.0 and 1.0 for both shape and color.

6.4.2 Tversky

Formula:

$$Tversky_{A,B} = \frac{overlapAB}{\alpha*selfA+\beta*selfB}$$

The Tversky similarity measure is asymmetric. Setting the parameters $\alpha = \beta = 0.5$ makes it symmetric and somewhat identical to using the *Tanimoto* measure.

The factor α weights the contribution of the first *reference* molecule. The larger α becomes, the more weight is put on the self-overlap of the reference molecule.

Unlike Tanimoto similarity, Tversky similarity may not always have a value between 0.0 and 1.0. This is true for both shape and color. Depending on the number of atoms between molecules A and B, it is possible to have $|overlapAB| > |selfA|$, and that along with certain values of α can sometimes lead to $Tversky_{A,B} > 1.0$.

Note

The default settings for `RefTversky` use $\alpha = 0.95$ and $\beta = 0.05$, and the default settings for `FitTversky` use $\alpha = 0.05$ and $\beta = 0.95$.

CHAPTER

SEVEN

RELEASE HISTORY

7.1 ROCS 3.10.0

Minor internal improvements have been made.

7.1.1 Related Toolkit Information

Shape TK 3.10.0

New Features

- When using the Overlay method in OEOverlay for overlay optimization, any of the inertial start methods now automatically trigger an additional as-is start if the optimization results in similarity measures less than the original similarity. This improves the default behavior of Overlay for some calculations.
- The following modifications have been made to OEOverlapPrepOptions to provide more flexibility in assigning color atoms during the Prep process in OEOverlapPrep:
  - Two new methods, SetRemoveColor and GetRemoveColor, have been added to enable removal or reassignment of color atoms.
  - The new methods AddDuplicateSmarts, GetDuplicateSmarts, and ClearDuplicateSmarts have been added to enable duplicate color atom removal based on provided SMARTS patterns.
  - Duplicate color atom removal is now more robust in keeping the central atoms instead of the first atom that it encounters and in placing the remaining atoms appropriately (at the center of corresponding clusters).
  - Defaults for SetRemoveDuplicate and SetDuplicateDistance have been changed from `False` and *0.001* to `True` and *0.0*, respectively. This improves robustness but keeps the default behavior unchanged.
- A new method, SetupFrame, has been added to OEStarts that simplifies setting up a reference frame for calculating the starts.

Minor Bug Fixes

- The overlay optimization in OEHermiteOverlay with the OEAsIsStarts start option has been updated to use Hermite expansions around the center of mass to prevent inconsistent overlaps when objects are far from the origin.
- Overlay optimization with the Overlay method in OEOverlay now captures the fit conformer self-overlap and color scores more accurately during the optimization.

7.2 ROCS 3.9.0

7.2.1 New Features

- A new option, *-coloronly*, has been added that enables overlay optimization against only the color components of a query.
- Molecules containing just color atoms or OEShapeQuery containing just color Gaussians are now allowed as queries through *-query*.

7.2.2 Related Toolkit Information

Shape TK 3.9.0

New Features

- The SetupRef method in OEOverlay now accepts molecules containing only color atoms. Accordingly, any molecule containing only color atoms or an OEShapeQuery containing only color Gaussians can now be used as a reference for overlay optimization.
- The SetupRef method in OEOverlapFunc now accepts molecules and OEShapeQuery containing just color components.
- A new class, OEHermiteOverlay, and a corresponding options class, OEHermiteOverlayOptions, are introduced, enabling overlays using the Hermite expansion approach. Three types of reference and database molecule representations are supported: Hermite vector, molecule, and scalar grid. Consequently, grid/grid overlays are enabled by using these classes.
- The grid-to-Gaussians fit algorithm, to add Gaussians in an OEShapeQuery using AddShapeGaussians and AddColorGaussians, has been improved for better representation of the grid by the generated Gaussians. With the introduction of the modified algorithm, some of the associated options in OEShapeQueryOptions are no longer relevant and have been removed from the class API. The following functions have been removed:
  - `GetDuplicateDistance` and `SetDuplicateDistance`
  - `GetMaxNumGaussians` and `SetMaxNumGaussians`
- A new overload has been added for Transform that enables transformation of a scalar grid by the rigid transformation in OEBestOverlayScore.
- A new overload has been added for Setup that enables setting up OEStarts with known fit symmetry.
- The following new methods have been added to OEHermite to enhance usability:
  - NumCoefficients
  - A new overload for GetCoefficients

Minor Bug Fixes

- An issue related to Init was causing intermittent slowdown in some API calls, especially in CUDA-enabled builds. This has been fixed.

7.3 ROCS 3.8.0

Minor internal improvements have been made.

7.3.1 Related Toolkit Information

Shape TK 3.8.0

New Features

- A new method, AddShapeGaussians, has been added that allows adding shape Gaussians to the OEShapeQuery based on the intensity of values on a grid.
- A new class, OEShapeQueryOptions, has been added that allows modifying options that would be used in adding Gaussians using the AddColorGaussians and AddShapeGaussians. Subsequently, a new constructor for OEShapeQuery has been added that takes the OEShapeQueryOptions as an argument.

7.4 ROCS 3.7.0

7.4.1 New Features

- The utility application *CheckCff* has been restructured to provide the same underlying functionality for color assignment as that of the Prep method from the OEOverlapPrep class. The following new command line parameters have been added.
  - `-removeDups`: Flag if color atoms within a certain distance from each other should be removed.
  - `-minDistance`: Minimum distance between color atoms for them to be treated as duplicates.
- The modified *CheckCff* always generates a report file in `.csv` format and `-report` can be used to provide a nondefault name to the report file.

7.4.2 Minor Bug Fixes

- `--help -chemff` for the *CheckCff* now properly displays all of the available options.

7.5 Shape TK 3.7.0

7.5.1 New features

- A new method, AddColorGaussians, has been added that allows adding color Gaussians to the OEShapeQuery based on intensity of values on a grid.
- A new class, OEOverlapPrepOptions, has been added that allows modifying options that would be used during the Prep. Subsequently, a new constructor for OEOverlapPrep has been added that takes the OEOverlapPrepOptions as an argument.

7.5.2 Minor bug fixes

- Color force field names in OEColorFFParameter are now treated as case-insensitive.

7.6 ROCS 3.6.2

7.6.1 New Features

- Command line flags related to the **EON** input generation category have been removed. This is a consequence of new behavior of **EON** which no longer requires pre-alignment of input molecules. In particular, `-eon_input`, `-eon_maxconfs`, `-eon_input_size`, and `-eon_input_file` are no longer command line flags of this version of the **ROCS** Application.

- Double-clicking a shape query *.sq* file or selecting `Open with vROCS` will now start **vROCS** with the query loaded and available for editing or use in a run.

7.6.2 Major bug fixes

- Color force field files `ImplicitMillsDean.cff` and `ExplicitMillsDean.cff` shipped with `Applications data` are now consistent with the internal implementation of the corresponding force fields. This update specifically fixes an issue with the definition of `tertiaryAmine`.
- 2D hitlist depictions during a run in **vROCS** have been hidden on macOS, due to an incorrect graphical display.

7.6.3 Minor bug fixes

- During query editing in **vROCS**, the behavior of right-clicking on an atom and `Create a Color Atom` has more appropriate restrictions and no longer causes a crash.

7.7 Shape TK 3.6.2

7.7.1 New features

- The following preliminary API classes have been made permanent.
  - OEColorFFParameter
  - OEFlexiOverlapFunc
  - OEFlexiOverlapOptions
  - OEFlexiOverlapResults
  - OEFlexiOverlay
  - OEFlexiOverlayOptions
- New methods, SetAtomStrength and GetAtomStrength, have been added that allow multiplying the contribution of shape overlap for any atoms, without adding an additional atom or Gaussian.

7.8 ROCS 3.6.1

7.8.1 Minor bug fixes

- Multiple successive runs of **vROCS** now correctly provide default job names.

Minor internal improvements have been made.

7.9 Shape TK 3.6.1

7.9.1 Major bug fixes

- Memory consumption when initiating an OEColorForceField has been reduced by moving instantiation of internal OESubSearch-based *colorer* objects to a lazy instantiation mode.

7.10 ROCS 3.6.0

Minor internal improvements have been made.

7.11 Shape TK 3.6.0

7.11.1 New features

- OEFlexiOverlapFunc and OEFlexiOverlay have been updated to allow users to use a Shape Query as the reference system. This allows OEFlexiOverlapFunc SetupRef and OEFlexiOverlay SetupRef to accept a Shape Query.
- A new method, SetupFlex, has been added. This method can be used to set up overlap calculation functions when molecules are expected to be flexible and their self overlaps are not fixed to the initial value.
- A new method, SortedOverlay, has been added as an alternative to Overlay. This method provides a desired number of top hit results instead of all of the hits and is especially suitable when working with molecules with excessive numbers of conformers.
- Two additional control flags, GetRigidOverlay and SetRigidOverlay, have been added to the OEFlexiOverlayOptions class. These flags indicate if rigid overlay optimization between the fit molecule conformers and the reference system should be performed prior to the flexible optimization.

7.11.2 Major bug fixes

- An issue in OEFlexiOverlay that caused a TanimotoCombo value greater than 2.0 when a molecule was flexibly fitted against itself has been fixed.

7.11.3 Minor bug fixes

- The default value for the `color` function in OEFlexiOverlapOptions during OEFlexiOverlay has been changed from `none` to `exact`. This ensures that color overlaps are reported when flexible optimization is performed with default parameters.
- An issue has been fixed with OEFlexiOverlay that was causing shape Tanimoto to be occasionally reported as larger than 1.0 after a flexible optimization.
- The legal range of values for the color score multiplier has been added to the OEFlexiOverlapOptions and OEOverlayOptions option classes.
- OEColorFFParameter now allows users to select *explicitmillsdeannorings* properly as the color force field.
- The acceptable ranges for the number of optimization iteration steps have been defined for OEFlexiOverlayOptions and OEOverlayOptions.
- Newer implementation of Hermite code had limits on lambda parameters of the expansion. In rare cases, these were affecting the lambda parameter search (Newton optimization stepping outside the limits). Removing the limits fixed the previous problems with finding lambda parameters.

7.12 ROCS 3.5.1

Fall 2022

Minor internal improvements have been made.

7.13 Shape TK 3.5.1

7.13.1 New features

- The following new methods have been added to OEOverlayOptions to improve flexibility in defining parameters:
  - SetShapeFuncType
  - GetShapeFuncType

  - SetColorFuncType
  - GetColorFuncType
  - SetShapeOptions
  - GetShapeOptions
  - SetColorOptions
  - GetColorOptions

- A new overload for the method CreateGrid has been added that takes a transform object as an additional argument that is applied after the grid is generated.

7.13.2 Major bug fixes

- Default forcefields in OEFlexiOverlapOptions now ignore electrostatic interactions, as electrostatics appear to have negligible effects on optimized overlaid structures. Ignoring the electrostatic interactions also improves efficiency of optimization for the corresponding OEFlexiOverlapFunc.
- The default for OEOverlapFunc in OEFlexiOverlayOptions now consists of `Shape` only and no `Color`. The default `Shape score` used is Overlap.

7.13.3 Minor bug fixes

- A mismatch between the Hermite grid and the input molecule has been fixed.
- The maximum value on the Hermite resolution parameter has been increased from 30 to 100.

7.13.4 Documentation changes

The following C++ and Python examples have been added to demonstrate how to use new preliminary APIs:

- Two new C++ and Python examples have been added to the Flexible Overlay with Shape, Color, and Forcefield section that show flexible overlay optimization of the fit molecules against a single reference molecule conformer.
- When using BestOverlay, only one conformer that has the highest Tanimoto-combo similarity to the reference molecule is returned.

7.14 ROCS 3.4.3

Fall 2021

- Minor internal improvements have been made.

7.15 Shape TK 3.4.3

Fall 2021

7.15.1 New features

- A new method, OEMol2Query, has been added that creates an OEShapeQuery from an input OEMolBase.
- A new preliminary API class, OEColorFFParameter, has been added as part of the extended set of OEParameter classes. The new parameter class works with predefined color force fields using an OEColorForceField object.

7.15.2 Minor bug fixes

- OESetColorParents now errors out properly if the color atom and the parent atoms specified do not have the same parent molecule.

7.16 ROCS 3.4.2

July 2021

- Minor internal improvements have been made.

7.17 Shape TK 3.4.2

July 2021

7.17.1 Major bug fixes

- A bug that caused some molecules to get a perfect Shape Tanimoto score of 1.0, even though their overlap was not optimal, has been fixed.

7.18 ROCS 3.4.1.0

Fall 2020

- Minor internal improvements have been made.

7.19 Shape TK 3.4.1

Fall 2020

7.19.1 New features

- A new function, OEIsFastROCSShapeQuery, has been added that determines if an OEShapeQuery object is compatible with **FastROCS TK**.

7.19.2 Major bug fixes

- An issue that caused performance slowdown in OESelfShape has been fixed.
- An issue that caused OEGridColorFunc to give incorrect results when used with a custom color force field containing a range value other than 1.0 has been fixed.

7.19.3 Minor bug fixes

- The UserColor example has been updated to more robustly check return statuses. See Calculating Overlap.

7.20 ROCS 3.4.0

Spring 2020

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

7.20.1 Major bug fixes

- Shape queries with the file extension SQ no longer give incorrect shape Tanimoto scores.

7.20.2 Minor bug fixes

- `ScaledColor` and `ComboScore` are no longer available as options for the `-rankby` parameter. Accordingly, those quantities are also no longer reported in the output results files. Users are encouraged to use `ColorTanimoto` and `TanimotoCombo` instead.
- Images in the **RocsReport** documentation have been updated.

7.21 ROCS 3.3.2

Nov 2019

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

7.21.1 Minor bug fixes

- **vROCS** now properly accepts a custom force field file.
- An issue that caused **ROCS** to fail to properly reporting molecule read failures when running with `-mpi_np` has been fixed.
- An issue with misaligned molecules when using a multi-conformer query has been fixed in **ROCS**.
- **ROCS** now appends a conformation number to the molecule even for single conformer molecules in an output molecules.
- Hardset limitation of 100,000 decoys for **vROCS** validation runs have been removed.
- The **ROCS** output file for **EON** now contains proper title numbering (wart).
- The `Custom` option for the `-pagesize` parameter in **RocsReport** has been removed.

7.22 ROCS 3.3.1

May 2019

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

7.22.1 New Features

Users can now specify **oez** files as input to **ROCS**.

7.22.2 Major bug fixes

- **ROCS** now properly handles empty query objects and queries consisting of only color atoms.
- **ROCS** now properly handles molecules without coordinates.

7.22.3 Minor bug fixes

- `ROCS_ShapeQuery` field now contains conformer ID when using `-mcquery true`.
- **ROCS** no longer throws warning when used with a shape query that contains no molecule.

7.23 ROCS 3.3.0

November 2018

- This version of **ROCS** has been built using **OEToolkits 2018.Oct**. The previous version was built using **OEToolkits 2016.Oct**.
- The engine behind **ROCS**, **Shape TK**, has gone through a major re-engineering in the **OpenEye Toolkits v2017.Oct** release. With the introduction of **Shape TK 2.0**, overlay optimization in ROCS is now approximately 10% faster than in the previous release.

7.23.1 Major bug fixes

- An issue that caused **ROCS** to crash when a shape query from **Shape TK** was used as a **ROCS** query has been fixed.
- The `Ligand Model Builder` in **vROCS** now properly handles queries that contain no heavy atoms.

7.23.2 Minor bug fixes

- The `Documentation` sub-menu of **vROCS** now correctly points to the online documentation.
- Negative values and values larger than 1.0 for color Tanimoto are no longer reported.
- The Rank by drop-down menu of the Options tab in **vROCS** validation has been modified to use the same score names as those used in the **ROCS** application.

7.24 ROCS 3.2.2

June 2017

7.24.1 Platform Support

- Support has been added for Ubuntu 16, OSX 10.10, OSX 10.11, and macOS Sierra.
- OSX 10.7, OSX 10.8, OSX 10.9, and Redhat 5 x86 are no longer supported.
- This is the last release to support Redhat 5 x64.
- SUSE Linux is no longer supported.

7.24.2 Bug Fixes

- **ROCS** will no longer crash if a query contains only hydrogens or dummy atoms. If the query file contains multiple queries, **ROCS** will skip the bad structure and continue the search; if the query file contains only a bad structure, **ROCS** will halt the search.
- A bug that caused **ROCS** to ignore the `-eon_input_file` command line flag has been fixed.
- A bug that caused ROCS to ignore the `-shapeonly` option if a shape query (.sq) was used has been fixed. **ROCS** now issues a warning and terminates the search.
- **ROCS** will no longer terminate a search if a molecule stream is used as the database file.

- **ROCS** will no longer terminate a search if a named pipe is used as the database file.
- A bug that caused **CHUNKER** to fail when splitting large conformer databases has been fixed. **CHUNKER** will now split large databases into the sizes requested by the user.
- **CHUNKER** will no longer accept negative numbers for `-chunksize` and `-nchunks` command line options.
- A bug that caused **CHUNKER** to crash if the input file was empty or did not exist has been fixed.
- Query information reported by **ROCS** has been enhanced. For example, for a multi-conformer query, **ROCS** will output conformer information as follows: Query (#1): acetsali conformer 1 of 15, Query (#2): acetsali conformer 2 of 15 . . . and so on.
- **CHECKCFF** will no longer accept non-OEB format as output file format.
- **HLMERGE** will no longer accept negative numbers for `-besthits` command line option.
- An issue causing subsequent runs in **vROCS** to have odd naming has been fixed.
- Obsolete *Rank By* run options in **vROCS** have been removed.

7.25 ROCS 3.2.1

September 2015

7.25.1 New Features

- The default **ROCS** output format is now gzipped OEBinary, `oeb.gz`, in order to save disk space.
- **ROCS** will treat database list file entries that do not include a path as though they are relative to the list file location, rather than relative to the working directory. The *hlmerge* application will do the same with entries in lists of hit lists.

7.25.2 Bug Fixes

- **ROCS** will no longer crash if an empty shape query is given.
- **ROCS** now writes a warning and skips any database molecule that have either missing xyz coordinates or are clearly 2 dimensional (z coordinates are all zero and dimension set to 2).
- **ROCS** now issues a warning and halts if *-subrocs* is true and the query is a grid. Grids are not supported for SubROCS.
- **ROCS** now issues a warning and halts if a query grid contains negative values. Grids with negative values are not supported in **ROCS**.
- Rocs_report will halt if it does not detect a query. Previously, an incorrect query mol was shown in the report when there was no query in the **ROCS** output file and *-refmol* was not specified.
- **ROCS** now writes files ending in `_ref.grd`, `_ref.sq`, `_errors.sdf`, `_eon_input.oeb.gz` to the directory defined when the *-outputdir* parameter is specified, along with all the other output files.
- ROCS now works with gzipped grid queries.
- ROCS now issues a warning to clarify that the *-optchem* parameter is ignored when *-opt* is false and *-scoreonly* is false.
- Setting the **ROCS** parameter *-opt* to false now turns off optimization directly. Previously, the parameter *-scoreonly* did this when set to true.
- The **ROCS** parameter *-eon_input_size* can now be set to 0, to indicate that no hit list is to be maintained and results are to be streamed, as previous documentation had indicated. Setting the parameter to 0 is required if *-besthits* and *-maxhits* are both 0.

- **ROCS** will terminate if a color force field specified by *-chemff* is not identical to that defined in a shape query file.
- The **ROCS** parameter *-tanimoto_cutoff* is now restricted to the range 0.0 to 1.0.
- Subtan SD tags are removed if *-subtan* is false.
- The *rocs_report* application parameter *-maxpages* now properly counts the title page.
- The input file to the *checkcff* and *makerocsdb* applications is now checked to ensure it is 3-dimensional.

Note

On Windows it is advisable to use the 64-bit ROCS executable during memory-intensive tasks (for example, combining *-mcquery* with *-subrocs*) to avoid a crash due to insufficient memory.

7.26 ROCS 3.2.0

September 2013

7.26.1 New Features

- The default hitlist format has been changed from sdf to oeb for increased functionality and decreased filesize. The output format is adjustable with the *-oformat* parameter.
- The *-subrocs* option has been added. The *-subrocs* starting points include 4 positions at every heavy atom in addition to the default inertial alignments.
- The rocs_report application has been added for making pdf reports from a hitlist.
- Now SD tags are prefixed with *ROCS_*. The tags are now optional with the *sdTags* parameter.
- **vROCS** now supports the manual merging of selected color atoms, to create a new color atom at their centroid.
- An improved molecule sketcher in **vROCS** now highlights unspecified stereochemistry in atoms and bonds when they occur in query structures, and requires the user to correct any unspecified stereochemistry before proceeding with a query.
- Numerous minor enhancements have been made to the user interface.

7.26.2 Bug Fixes

- Multiconformer molecules from **OMEGA** could be split into multiple molecule when the OMEGA output was written to .sdf instead of .oeb and had invertible stereocenters. Now ROCS will keep these .sdf multiconformer molecules together.
- Color atoms are now removed from **ROCS** hitlists for all output formats. Color atoms can be added to hitlists using the *checkcff* utility.
- Fixing bug for when eon_input_size is larger than -best, now it increase -best when needed.
- Now when **ROCS** hits are being streamed with -besthits 0, then eon_input molecules will be as well.
- Fixed a bug where sometimes .sdf files would get warts applied twice.
- Cleaned up missing entries or zeroes for Subtan and the conf index in the report file.
- Fixed a bug where report file entries could be duplicated with the *-scoreonly* flag.
- Fixed a bug where *-report one* could cause problems writing status files.

- Failure cases have been cleaned up. Incorrect and incompatible command line flags will cause ROCS to exit quickly and cleanly.
- Fixed a bug in **vROCS** in which files saved with a filename ending in ".sq.gz" would have an additional ".sq" appended to the end of the name.
- Fixed a problem where **ROCS** database files (.rocsdb) were not being read properly by vROCS.

7.26.3 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 *ROCS* in MPI mode. These new flags replace the *oempirun* script.
- This will be the last release to support SuSe 10.

7.27 ROCS 3.1.2

June 2011

7.27.1 Bug Fixes

- Fixed a bug where **EON** input files were not generated correctly when using multiple queries.
- Fixed a bug where the query was written to the **EON** input file too many times.
- Fixed a bug concerning validation runs failing in **vROCS**.
- Fixed a bug where application could crash on exit when using the `-maxhits` option
- Fixed a bug concerning output of color atoms to non-OEB format. Now precalculated color will only pass through if the output format is OEB.
- Now **ROCS** will not write EON_input if the **ROCS** input is a shape query. Only real molecules can be used as **ROCS** input when writing EON_input.

7.28 ROCS 3.1.1

February 2011

7.28.1 New Features

- Added new platform: RedHat 6 x64

7.28.2 Bug Fixes

- Fixed a bug running user-provided command line options in **vROCS**.
- Fixed a bug of reporting erroneous error messages.
- Fixed a problem concerning DLL dependencies on Windows.

7.29 ROCS 3.1.0

November 2010

7.29.1 New Features

- Added new platforms: Ubuntu 10 x64 and Sled 11 x64.
- **ROCS** now works with named pipes created using mkfifo on non-Windows platforms.
- The hlmerge application was not opening lists unless the file had an extension of ".list". Now hlmerge will attempt to open any non-molecule file as a list.

7.29.2 Bug Fixes

- Fixed a bug related to saving screenshots on Linux.
- Fixed a bug related to adding color atoms to a query.
- Fixed a bug related to the contour slider losing focus.
- Fixed a bug related to two contours overlapped by default causing display artifacts.
- Removed spurious warnings for "-auto" and "-rqedbmols".
- Fixed a bug in chunker that was causing the output file to be empty.
- Fixed a bug where the "-reportfile" option was being ignored.
- Fixed a bug where the "-maxconfs" option was being ignored.
- Fixed a bug in the coloring of aldehydes.
- Fixed a bug where multiple processor jobs would print "Slave started on host Host". The message now prints the proper hostname for the process.
- Fixed a bug where using "-stats best" was causing an empty report file.

7.30 ROCS 3.0.0

November 2009

7.30.1 New Features

- **ROCS** now comes with a new graphical application, **vROCS**, for setting up and testing queries. There are new sections in the manual that describe **vROCS** and new tutorials that walk you through using it. Please note that while **vROCS** makes creating complex queries much more manageable, it is **not** required. The *rocs* command line app will continue to be the core of **ROCS**. **vROCS** is not available on AIX or Solaris
- **vROCS** saves queries as Shape Query (.sq) files. Running **ROCS** from the command line will not accept Shape Query files as well as all the previously supported file formats.
- **ROCS** now supports MPI as well as PVM for distributed job management. This new feature is built on top of Open MPI and doesn't require any additional installs on cluster machines.

7.30.2 Bug Fixes

- Fixed a bug that could cause a crash when piping molecules into **ROCS** via stdin.
- Updated documentation to show that XYZ (.xyz) files are also suitable as **ROCS** queries.
- Fixed a bug when using .list files, the progress bar was not reset.

7.31 ROCS 2.4.2

September 2009

7.31.1 Bug Fixes

- Fixed a bug that caused a crash on Windows when trying to load an external CFF file. (Case 3413)
- Fixed the additional utilities so that they only require a "rocs" license and not an "oechem" license. (Cases 3350, 3435, 3440)
- Removed some extraneous command line arguments from "checkcff" (Case 3392)

7.32 ROCS 2.4.1

May 2009

7.32.1 New Features

- As of this release, the new default score for ranking is the TanimotoCombo, the sum of Shape Tanimoto and the new Color Tanimoto. Version 1.7 of the Shape Toolkit introduced new color scores based on Tanimoto. This version of **ROCS** incorporates all the new scores including Color Tanimoto and the resulting Color Tversky scores. Additionally all the Tversky scores are now named relative to either the Ref(erence) or Fit molecule, instead of the former query and dbase labels. This move is designed to standardize the names used in **ROCS** and the Shape Toolkit.
- In order to facilitate all the new scores, the Report file columns have been renamed and re-ordered.
- **ROCS** no longer counts the input file prior to starting the calculation. Progress is now based on file size not number of molecules processed. This alleviates a problem with very long start-up times for extremely large database files.
- This release adds **ROCS** to the set of applications that now use a common script in `openeye/bin` to determine the appropriate architecture and run the appropriate binary.
- On Windows, there is now an installer that installs the documentation and sets up a command prompt to facilitate running the ROCS command line.

7.32.2 Bug Fixes

- Fixed a bug that caused a crash when trying to write an empty hitlist.

7.33 ROCS 2.3.1

August 2007

7.33.1 Bug fixes

- Fixed a bug reading .rocsdb files.
- Fixed a subtle bug that caused `-report one` to only work if `-reportfile` is also used. Using `-report one` will now create a single report file named based on the `-prefix` given.
- Removed an annoying but benign warning about the now deprecated and removed `-allcolor` flag.
- Fixed a bug where the `-eon_maxconfs` flag expected a value greater than 1. Now any value greater than zero is valid.

7.34 ROCS 2.3.0

August 2007

7.34.1 Enhancements

- Use of the color force field is now on by default. With this, the default score to rank the hitlist has been changed to `combo`. The following new default flag values:

```
-chemff ImplicitMillsDean
-optchem
-rankby combo
```

- **ROCS** now has multiple methods of showing job progress. In addition to the dots of previous versions, ROCS now has a progress bar by default that show percent completion of the dbase file. Note that means that when **ROCS** starts, it will count the number of molecules in the dbase and then use that value to measure percent completion. This also allows log file messages to be tagged to the index into the dbase file for each molecule as it is scored. See the description for the new flag *-progress* for info on the other options.
- Since color is now on by default, a new flag *-shapeonly* has been added to turn on the shape only features of previous versions for **ROCS**. This is only a convenience flag that just sets the following:

```
-chemff none
-optchem false
-rankby tanimoto
```

- Since **ROCS** is routinely used to preparation of input files for **EON**, a new set of flags have been added that aid in creation of a specific **EON** input file, different from the standard **ROCS** hits file. Please see the section about describing these options:

```
-eon_input
-eon_maxconfs
-eon_input_size
-eon_input_file
```

- Added a new status file (`rocs.pvmlog` by default), that shows PVM master and slave statistics, during PVM runs. The actual filename can be modified with the new `-pvmlog` switch.
- Deprecated in the last release, the `-allcolor` flag has now been removed.

7.35 ROCS 2.2.0

March 2006

7.35.1 Enhancements

- Added ability to read rotor-offset-compressed OEB files as written from **OMEGA** 2.0 and later.
- Added new *-tanimoto_cutoff* flag that can be used to limit output hits to only those with some minimum shape score. This can be used regardless of which score is chosen (-rankby) for ranking the hitlist. For example, using

```
-rankby combo -cutoff 1.2 -tanimoto_cutoff 0.6
```

  any molecule with Shape Tanimoto <= 0.6 will not be retained. Additionally, the constraint on combo to be at least 1.2 implies that scaledcolor must also be > 0.6 so that the sum can be greater than 1.2.

- *-scoreonly* flag to allow scoring of pre-aligned positions. Using this one flag is the equivalent of the following settings:

```
-opt false
-besthits 0
-maxhits 0
-scdbase
```

- New *-conflabel* flag to control where conformer indices are placed in output files. Previous versions added the conformer index at the end of the molecule title following an underscore. The new flag has 4 possible values:
  - `title` just like previous behavior, added index to end of title
  - `sdtag` Conf idx is placed in SD tag <ConfIndex>
  - `both` combination of previous two places
  - `none` conformer indices are not marked on output.

> **Note**
>
> `sdtag` is only useful if the output file is SDF or OEB.

- Several new flags have been added to allow more complete control of the different files **ROCS** can output. The added flags include:
- *-logfile* Instead of writing to stderr, all logging information goes into a file provided by this switch can be filename or full/relative path
- *-reportfile* Instead of writing to `PREFIX_n.rpt` where PREFIX is provided by the *-prefix* commandline flag, write all report information (stats) to the file provided with this flag. Can be filename or full/relative path. Note that if more than one query molecule is provided, this flag will not work unless the `-report one` flag is also provided to put all report info into one report file.
- *-hitsfile* Instead of writing to `PREFIX_hits_n.sdf` (for example) where PREFIX is provided by the *-prefix* commandline flag, write all hit structures to the file provided with this flag. Can be a filename or full/relative path. Also, if the name provided is actually a molecule file format extension (i.e. .sdf, .mol2.gz, .oeb, etc.), **ROCS** will write to stdout using the format derived from the file extension. For example if the following is used

```
-hitsfile .sdf
```

  then ROCS will write all the hits out to stdout in SDF format.

  Note that this option will only work for a single molecule query. If more than one query is provided along with the *-hitsfile* option, **ROCS** will issue an error and stop.

- In order to facilitate using **ROCS** as part of a larger workflow, the *-dbase* flag can now be used to force **ROCS** to read its dbase molecules from stdin. If a file extension (i.e. .sdf, .mol2.gz, etc.), is used as the -dbase filename, ROCS will attempt to read dbase molecules from stdin in the format described by that file extension. For example, if the following is used:

```
-dbase .sdf
```

  then ROCS will read the dbase molecules from stdin in SDF format.

  Note that this option will only work for a single molecule query. If more than one query is provided along with a -dbase flag attempting to read from stdin, **ROCS** will issue an error and stop.

- Changed the way untitled ref and fit molecules are named so that in hits files, they can be distinguished and so that untitled names are different than conformer index names.
- `-allcolor` flag has been deprecated and will be removed from a future version of **ROCS**.
- Added new *-out* flag to checkcff to output molecules to an OEB file with color atoms attached. This file can be loaded into VIDA and the atoms labeled with "Name" to show the location and type of all added color atoms.

7.35.2 Bug fixes

- Added check for untitled reference (query) molecules. If no title is found, a title similar to "untitled-ref-N" is applied. This address a bug in EON caused by blank titles in **ROCS** queries, resulting in broken SD tags in the input to **EON**.
- Fixed a bug that caused molecules of high-symmetry to be skipped. If a molecule's moments of inertia were all equal, it would be skipped. If the molecule was the reference, no hits would be found.
- Fixed a bug in random starts.

CHAPTER

EIGHT

SELECTED ROCS PUBLICATIONS

Numerous journal articles have been published about ROCS through the years. Included here are some of the more impactful papers. They are arranged in chronological order, beginning with the seminal work on Gaussians by Grant, Gallardo, and Pickup.

- Grant, J.A.; Gallardo, M.A.; Pickup, B.T. **A Fast Method of Molecular Shape Comparison. A Simple Application of a Gaussian Description of Molecular Shape.** *J. Comput. Chem.* **1996**, *17* (14), 1653–1666. DOI: 10.1002/(sici)1096-987x(19961115)17:14<1653::aid-jcc7>3.0.co;2-k
- Nicholls, A.; MacCuish, N.E.; MacCuish, J.D. **Variable Selection and Model Validation of 2D and 3D Molecular Descriptors.** *J. Comput.-Aided Mol. Des.* **2004**, *18* (7), 451–474. DOI: 10.1007/s10822-004-5202-8
- Rush, T.S.; Grant, J.A.; Mosyak, L.; Nicholls, A. **A Shape-Based 3D Scaffold Hopping Method and Its Application to a Bacterial Protein–Protein Interaction.** *J. Med. Chem.* **2005**, *48* (5), 1489–1495. DOI: 10.1021/jm040163o
- Haigh, J.A.; Pickup, B.T.; Grant, J.A.; Nicholls, A. **Small Molecule Shape-Fingerprints.** *J. Chem. Inf. Model.* **2005**, *45* (3), 673-684. DOI: 10.1021/ci049651v
- Chen, H.; Lyne, P.D.; Giordanetto, F.; Lovell, T.; Li, J. **On Evaluating Molecular-Docking Methods for Pose Prediction and Enrichment Factors.** *J. Chem. Inf. Model.* **2006**, *46* (1), 401–415. DOI: 10.1021/ci0503255
- Muchmore, S.W.; Souers, A.J.; Akritopoulou-Zanze, I. **The Use of Three-Dimensional Shape and Electrostatic Similarity Searching in the Identification of a Melanin-Concentrating Hormone Receptor 1 Antagonist.** *Chem. Biol. Drug Des.* **2006**, *67* (2), 174–176. DOI: 10.1111/j.1747-0285.2006.00341.x
- 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
- McGaughey, G.B.; Sheridan, R.P.; Bayly, C.I.; Culberson, J.C.; Kreatsoulas, C.; Lindsley, S.; Maiorov, V.; Truchon, J.-F.; Cornell, W.D. **Comparison of Topological, Shape, and Docking Methods in Virtual Screening.** *J. Chem. Inf. Model.* **2007**, *47* (4), 1504–1519. DOI: 10.1021/ci700052x
- Boström, J.; Berggren, K., Elebring, T.; Greasley, P.J.; Wilstermann, M. **Scaffold Hopping, Synthesis and Structure–Activity Relationships of 5,6-diaryl-pyrazine-2-amide Derivatives: A Novel Series of CB1 Receptor Antagonists.** *Bioorg. Med. Chem.* **2007**, *15* (12), 4077–4084. DOI: 10.1016/j.bmc.2007.03.075
- Sutherland, J.J.; Nandigam, R.K.; Erickson, J.A.; Vieth, M. **Lessons in Molecular Recognition. 2. Assessing and Improving Cross-Docking Accuracy.** *J. Chem. Inf. Model.* **2007**, *47* (6), 2293–2302. DOI: 10.1021/ci700253h
- Freitas, R.F.; Oprea, T.I.; Montanari, C.A. **2D QSAR and Similarity Studies on Cruzain Inhibitors Aimed at Improving Selectivity over cathepsin L.** *Bioorg. Med. Chem.* **2008**, *16* (2), 838–853. DOI: 10.1016/j.bmc.2007.10.048
- Sheridan, R.P.; McGaughey, G.B.; Cornell, W.D. **Multiple Protein Structures and Multiple Ligands: Effects on the Apparent Goodness of Virtual Screening Results.** *J. Comput.-Aided Mol. Des.* **2008**, *22*, 257–265. DOI: 10.1007/s10822-008-9168-9

- Venhorst, J.; Núñez, S.; Terpstra, J.W.; Kruse, C.G. **Assessment of Scaffold Hopping Efficiency by Use of Molecular Interaction Fingerprints.** *J. Med. Chem.* **2008**, *51* (11), 3222–3229. DOI: 10.1021/jm8001058
- Nandigam, R.K.; Evans, D.A.; Erickson, J.A.; Kim, S.; Sutherland, J.J. **Predicting the Accuracy of Ligand Overlay Methods with Random Forest Models.** *J. Chem. Inf. Model.* **2008**, *48* (12), 2386–2394. DOI: 10.1021/ci800216f
- Sheridan, R.P. **Alternative Global Goodness Metrics and Sensitivity Analysis: Heuristics to Check the Robustness of Conclusions from Studies Comparing Virtual Screening Methods.** *J. Chem. Inf. Model.* **2008**, *48* (2), 426–433. DOI: 10.1021/ci700380x
- Lee, H.S.; Choi, J.; Kufareva, I.; Abagyan, R.; Filikov, A.; Yang, Y.; Yoon, S. **Optimization of High Throughput Virtual Screening by Combining Shape-Matching and Docking Methods.** *J. Chem. Inf. Model.* **2008**, *48* (3), 489–497. DOI: 10.1021/ci700376c
- Pérez-Nueno, V.I.; Ritchie, D.W.; Rabal, O.; Pascual, R.; Borrell, J.I.; Teixidó, J. **Comparison of Ligand-Based and Receptor-Based Virtual Screening of HIV Entry Inhibitors for the CXCR4 and CCR5 Receptors Using 3D Ligand Shape Matching and Ligand-Receptor Docking**, *J. Chem. Inf. Model.* **2008**, *48* (3), 509–533. DOI: 10.1021/ci700415g
- Moffat, K.; Gillet, V.J.; Whittle, M.; Bravi, G.; Leach, A.R. **A Comparison of Field-Based Similarity Searching Methods: CatShape, FBSS, and ROCS.** *J. Chem. Inf. Model.* **2008**, *48* (4), 719–729. DOI: 10.1021/ci700130j
- Muchmore, S.W.; Debe, D.A.; Metz, J.T.; Brown, S.P.; Martin, Y.C.; Hajduk, P.J. **Application of Belief Theory to Similarity Data Fusion for Use in Analog Searching and Lead Hopping.** *J. Chem. Inf. Model.* **2008**, *48* (5), 941–948. DOI: 10.1021/ci7004498
- Naylor, E.; Arredouani, A.; Vasudevan, S.R.; et al. **Identification of a Chemical Probe for NAADP by Virtual Screening.** *Nat. Chem. Biol.* **2009**, *5*, 220–226. DOI: 10.1038/nchembio.150
- Oyarzabal, J.; Howe, T.; Alcazar, J.; Andres, J.I.; Alvarez, R.M.; Dautzenberg, F.; Iturrino, L.; Martınez, S.; Van der Linden, I. **Novel Approach for Chemotype Hopping Based on Annotated Databases of Chemically Feasible Fragments and a Prospective Case Study: New Melanin Concentrating Hormone Antagonists.** *J. Med. Chem.* **2009**, *52* (7), 2076–2089. DOI: 10.1021/jm8016199
- Tresadern, G.; Bemporad, D.; Howe, T. **A Comparison of Ligand Based Virtual Screening Methods and Application to Corticotropin Releasing Factor 1 Receptor.** *J. Mol. Graphics Modell.* **2009**, *27* (8), 860–870. DOI: 10.1016/j.jmgm.2009.01.003
- Tuccinardi, T.; Ortors, G.; Santos, M.A.; Marques, S.M.; Nuti, E.; Rossello, A.; Martinelli, A. **Multitemplate Alignment Method for the Development of a Reliable 3D-QSAR Model for the Analysis of MMP3 Inhibitors.** *J. Chem. Inf. Model.* **2009**, *49* (7), 1715–1724. DOI: 10.1021/ci900118v
- Nicholls, A.; McGaughey, G.B.; Sheridan, R.P.; Good, A.C.; Warren, G.; Mathieu, M.; Muchmore, S.W.; Brown, S.P.; Grant, J.A.; Haigh, J.A.; Nevins, N.; Jain, A.N.; Kelley, B. **Molecular Shape and Medicinal Chemistry: A Perspective.** *J. Med. Chem.* **2010**, *53* (10), 3862–3886. DOI: 10.1021/jm900818s
- Krüger, D.M.; Evers, A. **Comparison of Structure- and Ligand-Based Virtual Screening Protocols Considering Hitlist Complementarity and Enrichment Factors.** *ChemMedChem* **2010**, *5* (1), 148–158. DOI: 10.1002/cmdc.200900314
- Swann, S.L.; Brown, S.P.; Muchmore, S.W.; Patel, H.; Merta, P.; Locklear, J.; Hajduk, P.J. **A Unified Probabilistic Framework for Structure- and Ligand-Based Virtual Screening.** *J. Med. Chem.* **2011**, *54* (5), 1223–1232. DOI: 10.1021/jm1013677
- Wollenhaupt, J.; Metz, A.; Barthel, T.; Lima, G.M.A.; Heine, A.; Mueller, U.; Klebe, G.; Weiss, M.S. **F2X-Universal and F2X-Entry: Structurally Diverse Compound Libraries for Crystallographic Fragment Screening.** *Structure* **2020**, *28* (6), 694–706.e5. DOI: 10.1016/j.str.2020.04.019

- Taylor, A.I.P.; Xu, Y.; Wilkinson, M.; Chakraborty, P.; Brinkworth, A.; Willis, L.F.; Zhuravleva, A.; Ranson, N.A.; Foster, R.; Radford, S.E. **Kinetic Steering of Amyloid Formation and Polymorphism by Canagliflozin, a Type-2 Diabetes Drug.** *J. Am. Chem. Soc.* **2025**, *147* (14), 11859–11878. DOI: 10.1021/jacs.4c16743
- Rossen, L.; Sirockin, F.; Schneider, N.; Grisoni, F. **Scaffold Hopping with Generative Reinforcement Learning.** *J. Chem. Inf. Model.* **2025**, *65* (13), 6513–6525. DOI: 10.1021/acs.jcim.5c00029

CHAPTER

NINE

CITATION

Note

To cite ROCS please use the following:

ROCS 3.10.0.0. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.eyesopen.com.

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

CHAPTER

TEN

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

ELEVEN

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

TWELVE

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.

12.1 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.

12.2 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.

12.3 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.

12.4 Publications for Bibliographies

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

12.4.1 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

12.4.2 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

12.4.3 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

12.4.4 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

12.4.5 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

12.5 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

THIRTEEN

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

13.1 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.

13.1.1 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

13.1.2 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

-aromstyle
    rocs_report command line option, 91
-assignBondiRadii
    checkcff command line option, 77
-assignColor
    checkcff command line option, 77
-astl
    rocs_report command line option, 91
-base
    command line option, 79
-besthits
    command line option, 80
    rocs command line option, 11
-biggerisbetter
    command line option, 80
-chemff
    checkcff command line option, 77
    makerocsdb command line option, 81
    rocs command line option, 15
-chunksize
    command line option, 79
-coloronly
    rocs command line option, 15
-conflabel
    rocs command line option, 12
-countConfs
    command line option, 79
-cutoff
    rocs command line option, 11
-dbase
    rocs command line option, 10
-hitsfile
    rocs command line option, 12
-in
    checkcff command line option, 76
    command line option, 79, 80
    makerocsdb command line option, 81
    rocs_report command line option, 90
-log
    checkcff command line option, 77
-logfile
    rocs command line option, 13
-maxconfs
    rocs command line option, 11
-maxhits
    rocs command line option, 12
-maxpages
    rocs_report command line option, 90
-mcquery
    rocs command line option, 10
-minDistance
    checkcff command line option, 78
-molNames
    checkcff command line option, 77
-mpi_hostfile
    rocs command line option, 10
-mpi_np
    rocs command line option, 10
-nchunks
    command line option, 79
-nostructs
    rocs command line option, 12
-oformat
    rocs command line option, 12
-opt
    rocs command line option, 15
-optchem
    rocs command line option, 15
-out
    checkcff command line option, 76
    command line option, 80
    makerocsdb command line option, 81
    rocs_report command line option, 90
-outputdir
    rocs command line option, 11
-outputquery
    rocs command line option, 12
-pad_zeros
    command line option, 79
-pagesize
    rocs_report command line option, 91
-param
    rocs command line option, 10

-prefix
    checkcff command line option, 77
    rocs command line option, 11
-progress
    checkcff command line option, 77
    rocs command line option, 13
-qconflabel
    rocs command line option, 12
-query
    rocs command line option, 9
-randomstarts
    rocs command line option, 14
-rankby
    command line option, 80
    rocs command line option, 11
-refmol
    rocs_report command line option, 90
-removeDups
    checkcff command line option, 78
-report
    checkcff command line option, 76
    rocs command line option, 13
-reportfile
    rocs command line option, 13
-scdbase
    makerocsdb command line option, 81
    rocs command line option, 10
-scoreonly
    rocs command line option, 14
-sdTags
    rocs command line option, 13
-shapeonly
    rocs command line option, 14
-stats
    rocs command line option, 13
-status
    rocs command line option, 13
-statusfile
    rocs command line option, 13
-subrocs
    rocs command line option, 14
-subtan
    rocs command line option, 14
-tanimoto_cutoff
    rocs command line option, 14
-useHydrogens
    checkcff command line option, 77
-verbose
    checkcff command line option, 77
    rocs command line option, 14
    rocs_report command line option, 90

C

checkcff command line option
    -assignBondiRadii, 77
    -assignColor, 77
    -chemff, 77
    -in, 76
    -log, 77
    -minDistance, 78
    -molNames, 77
    -out, 76
    -prefix, 77
    -progress, 77
    -removeDups, 78
    -report, 76
    -useHydrogens, 77
    -verbose, 77
command line option
    -base, 79
    -besthits, 80
    -biggerisbetter, 80
    -chunksize, 79
    -countConfs, 79
    -in, 79, 80
    -nchunks, 79
    -out, 80
    -pad_zeros, 79
    -rankby, 80

M

makerocsdb command line option
    -chemff, 81
    -in, 81
    -out, 81
    -scdbase, 81

R

rocs command line option
    -besthits, 11
    -chemff, 15
    -coloronly, 15
    -conflabel, 12
    -cutoff, 11
    -dbase, 10
    -hitsfile, 12
    -logfile, 13
    -maxconfs, 11
    -maxhits, 12
    -mcquery, 10
    -mpi_hostfile, 10
    -mpi_np, 10
    -nostructs, 12
    -oformat, 12
    -opt, 15
    -optchem, 15
    -outputdir, 11
    -outputquery, 12

-param, 10
-prefix, 11
-progress, 13
-qconflabel, 12
-query, 9
-randomstarts, 14
-rankby, 11
-report, 13
-reportfile, 13
-scdbase, 10
-scoreonly, 14
-sdTags, 13
-shapeonly, 14
-stats, 13
-status, 13
-statusfile, 13
-subrocs, 14
-subtan, 14
-tanimoto_cutoff, 14
-verbose, 14
rocs_report command line option
-aromstyle, 91
-astl, 91
-in, 90
-maxpages, 90
-out, 90
-pagesize, 91
-refmol, 90
-verbose, 90