

OpenEye
SCIENTIFIC

OEMedChem TK – C#

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Introduction **1**

2 Theory **3**
2.1 Molecule Fragmentation . . . 3
2.2 Matched Molecule Pair Analysis . . . 8

3 Examples **13**
3.1 OEMedChem TK Examples . . . 13

4 API **63**
4.1 OEMedChem API . . . 63

5 Release History **137**
5.1 OEMedChem TK 1.2.6 . . . 137
5.2 OEMedChem TK 1.2.5 . . . 137
5.3 OEMedChem TK 1.2.4 . . . 137
5.4 OEMedChem TK 1.2.3 . . . 137
5.5 OEMedChem TK 1.2.2 . . . 137
5.6 OEMedChem TK 1.2.1 . . . 137
5.7 OEMedChem TK 1.2.0 . . . 137
5.8 OEMedChem TK 1.1.7 . . . 138
5.9 OEMedChem TK 1.1.6 . . . 138
5.10 OEMedChem TK 1.1.5 . . . 138
5.11 OEMedChem TK 1.1.4 . . . 138
5.12 OEMedChem TK 1.1.3 . . . 138
5.13 OEMedChem TK 1.1.2 . . . 138
5.14 OEMedChem TK 1.1.1 . . . 139
5.15 OEMedChem TK 1.1.0 . . . 139
5.16 OEMedChem TK 1.0.7 . . . 139
5.17 OEMedChem TK 1.0.6 . . . 141
5.18 OEMedChem TK 1.0.5 . . . 141
5.19 OEMedChem TK 1.0.4 . . . 142
5.20 OEMedChem TK 1.0.3 . . . 143
5.21 OEMedChem TK 1.0.2 . . . 144
5.22 OEMedChem TK 1.0.1 . . . 144
5.23 OEMedChem TK 1.0.0 . . . 144
5.24 OEMedChem TK 0.9.6 . . . 146
5.25 OEMedChem TK 0.9.5 . . . 147
5.26 OEMedChem TK 0.9.4 . . . 148
5.27 OEMedChem TK 0.9.3 . . . 150

5.28 OEMedChem TK 0.9.2 . . . 151
5.29 OEMedChem TK 0.9.1 . . . 151
5.30 OEMedChem TK 0.9.0 . . . 151

6 Copyright and Trademarks **153**

7 Sample Code **155**

8 Citation **157**
8.1 Orion® . . . 157
8.2 Orion Floes . . . 157
8.3 Toolkits and Applications . . . 157
8.4 Publications for Bibliographies . . . 159
8.5 OpenEye MMDS Web Service . . . 160

9 Technology Licensing **161**
9.1 GCC . . . 233

Index **249**

CHAPTER

ONE

INTRODUCTION

This manual is to familiarize the user with **OEMedChem TK** functionality.

CHAPTER

TWO

THEORY

2.1 Molecule Fragmentation

The *OEMedChem TK* currently provides four ways to partition a molecule into fragments:

- *OEGetRingChainFragments* - fragments a molecule into ring and chain components.
- *OEGetRingLinkerSideChainFragments* - fragments a molecule into ring, linker and side-chain components as defined in [Bemis-1996] .
- *OEGetFuncGroupFragments* - fragments a molecule into ring and functional group components.
- *OEGetBemisMurcko* - fragments a molecule into ring, linker, framework and functional group components as in [Bemis-1996] .

These functions return an iterator over OEAtomBondSet objects that store the atoms and the bonds of the fragments.

The following examples (*Listing 1*, *Listing 2*) show how to fragment a molecule into ring and chain components. The code loops over the OEAtomBondSet objects returned by the *OEGetRingChainFragments*. Each OEAtomBondSet object is used to initialize an atom and a bond predicates. These predicates specify which atoms and bonds have to be considered when creating a subset of the molecule, *i.e* the fragment, when calling the `OESubsetMol` function. See the depiction of the input molecule and the generated fragments in *Figure: Example of fragmentation*.

Listing 1: Example of molecule fragmentation

```
using System;
using OpenEye.OEMedChem;
using OpenEye.OEChem;

public class FragRingChain
{
    public static int Main(string[] args)
    {
        OEGraphMol mol = new OEGraphMol();
        OEChem.OESmilesToMol(mol, "COc1ccc(cc1)CC(=O)N");

        foreach (OEAtomBondSet absset in OEMedChem.OEGetRingChainFragments(mol))
        {
            OEIsAtomMember fragatompred = new OEIsAtomMember(absset.GetAtoms());
            OEIsBondMember fragbondpred = new OEIsBondMember(absset.GetBonds());

            OEGraphMol fragment = new OEGraphMol();
            bool adjustHCount = true;
            OEChem.OESubsetMol(fragment, mol, fragatompred, fragbondpred, adjustHCount);
```

(continues on next page)

(continued from previous page)

```
            Console.WriteLine(OEChem.OEMolToSmiles(fragment));
        }
        return 0;
    }
}
```

The output of *Listing 1* is the following:

```
CO
c1ccccc1
CC(=O)N
```

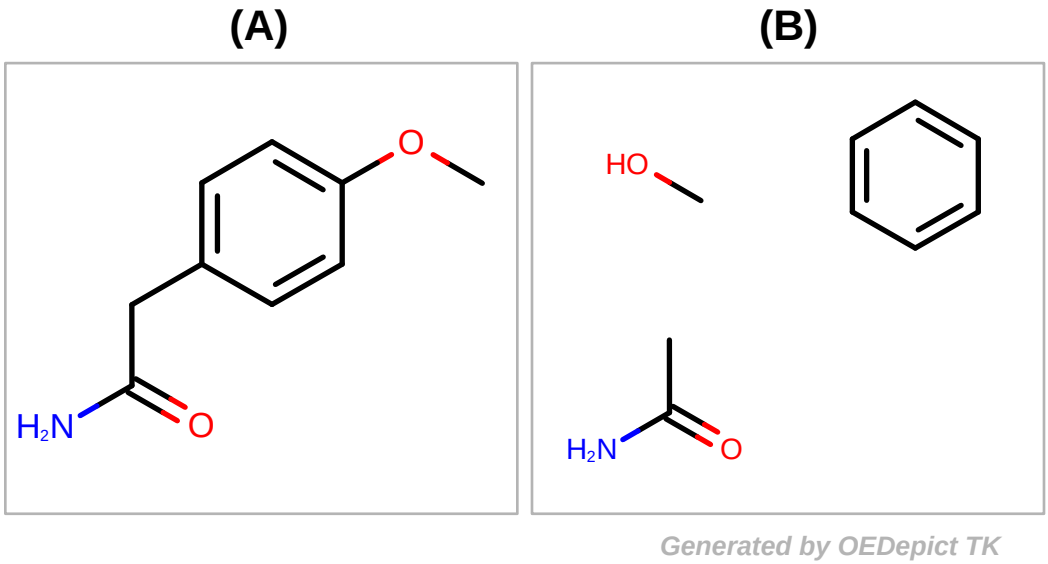


Fig. 1: **Example of fragmentation** (A) input molecule (B) fragments returned by the OEGetRingChainFragments function

See also

- OEAtomBondSet class and `OESubsetMol` function in the *OEChem TK* manual
- Predicate Functors chapter in the *OEChem TK* manual.

The following example (*Listing 2*) shows how to fragment a molecule into ring and chain components with annotations. See the depiction of the input molecule and the generated fragments in *Figure: Example of Bemis Murcko fragmentation*.

Listing 2: Example of molecule fragmentation with annotations

```
using System;
using OpenEye.OEMedChem;
using OpenEye.OEChem;
```

(continues on next page)

(continued from previous page)

```
public class OEGetBemisMurcko
{
    public static int Main(string[] args)
    {
        OEGraphMol mol = new OEGraphMol();
        OEChem.OESmilesToMol(mol, "CCOc1ccc(cc1)CC(OC)c2ccccc2CC(=O)N");

        foreach (OEAtomBondSet abset in OEMedChem.OEGetBemisMurcko(mol))
        {
            OEIsAtomMember fragatompred = new OEIsAtomMember(abset.GetAtoms());
            OEIsBondMember fragbondpred = new OEIsBondMember(abset.GetBonds());

            OEGraphMol fragment = new OEGraphMol();
            bool adjustHCount = true;
            OEChem.OESubsetMol(fragment, mol, fragatompred, fragbondpred, adjustHCount);
            foreach (OERole role in abset.GetRoles())
            {
              Console.WriteLine("{0} {1}", role.GetName(), OEChem.
↪OEMolToSmiles(fragment));
            }
        }
        return 0;
    }
}
```

The output of *Listing 2* is the following:

```
Framework c1ccc(cc1)CCc2ccccc2
Ring c1ccccc1.c1ccccc1
Linker CC
Sidechain CCO.CC(=O)N.CO
```

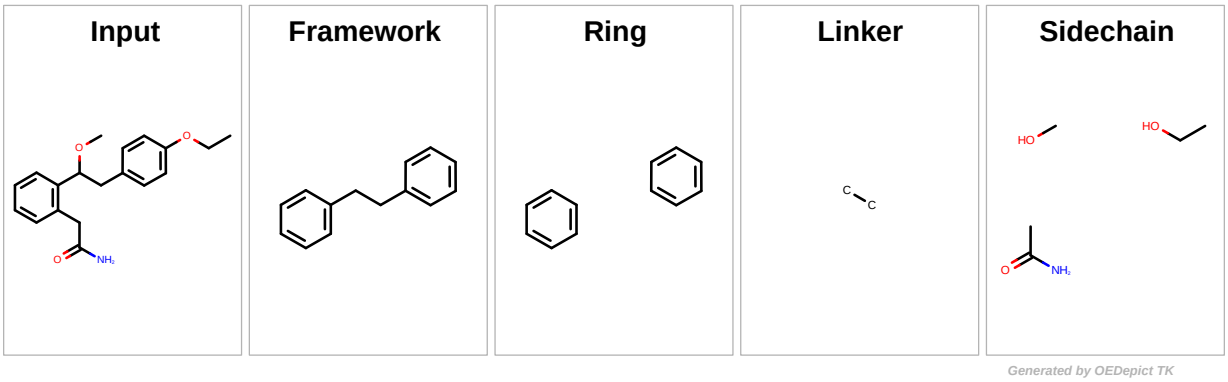


Fig. 2: **Example of Bemis Murcko fragmentation**

The following example (*Listing 3*) shows how to fragment a molecule and include unsaturated hetero bonds on the main framework. See the depiction of the input molecule and the generated fragments in *Figure: Example of custom Bemis Murcko fragmentation including heteroatoms*.

Listing 3: Example of custom molecule fragmentation

```
using System;
using OpenEye.OEMedChem;
using OpenEye.OEChem;

public class OEGetBemisMurckoUnsatHeteroBonds
{
    public static int Main(string[] args)
    {
        OEGraphMol mol = new OEGraphMol();
        OEChem.OESmilesToMol(mol, "CCc1nc(nc(n1)OC)NC(=O)NS(=O)(=O)c2ccccc2OC");
        OEBemisMurckoOptions options = new OEBemisMurckoOptions();
        options.SetUnsaturatedHeteroBonds(true);

        foreach (OEAtomBondSet abset in OEMedChem.OEGetBemisMurcko(mol, options))
        {
            OEIsAtomMember fragatompred = new OEIsAtomMember(abset.GetAtoms());
            OEIsBondMember fragbondpred = new OEIsBondMember(abset.GetBonds());

            OEGraphMol fragment = new OEGraphMol();
            bool adjustHCount = true;
            OEChem.OESubsetMol(fragment, mol, fragatompred, fragbondpred, adjustHCount);
            foreach (OERole role in abset.GetRoles())
            {
              Console.WriteLine("{0} {1}", role.GetName(), OEChem.
↪OEMolToSmiles(fragment));
            }
        }
        return 0;
    }
}
```

The output of *Listing 3* is the following:

```
Framework c1ccc(cc1)S(=O)(=O)NC(=O)Nc2ncncn2
Ring c1ccccc1.c1ncncn1
Linker C(=O)(N)NS(=O)=O
Sidechain CC.CO.CO
```

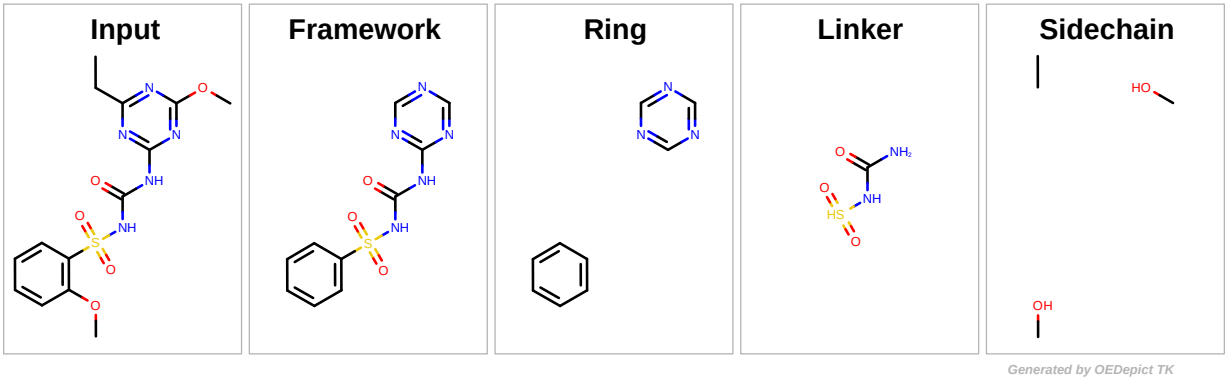


Fig. 3: **Example of Bemis Murcko fragmentation including unsaturated hetero bonds on main framework.**

The following example (*Listing 4*) shows how to fragment a molecule and include custom sidechains on the main framework. See the depiction of the input molecule and the generated fragments in *Figure: Example of custom Bemis Murcko fragmentation including custom substituents*.

Listing 4: Example of custom molecule fragmentation

```
using System;
using OpenEye.OEMedChem;
using OpenEye.OEChem;

public class OEGetBemisMurckoCustomSubstituents
{
    public static int Main(string[] args)
    {
        OEGraphMol mol = new OEGraphMol();
        OEChem.OESmilesToMol(mol, "CCc1nc(nc(n1)OC)NC(CC(=O)N)NS(=O)(=O)c2ccccc2CC(=O)N
↪");
        OESubSearch subsearch = new OESubSearch();
        subsearch.Init("[#6]-CC(=O)N");
        OEBemisMurckoOptions options = new OEBemisMurckoOptions();
        options.SetSubstituentSearch(subsearch);

        foreach (OEAtomBondSet abset in OEMedChem.OEGetBemisMurcko(mol, options))
        {
            OEIsAtomMember fragatompred = new OEIsAtomMember(abset.GetAtoms());
            OEIsBondMember fragbondpred = new OEIsBondMember(abset.GetBonds());

            OEGraphMol fragment = new OEGraphMol();
            bool adjustHCount = true;
            OEChem.OESubsetMol(fragment, mol, fragatompred, fragbondpred, adjustHCount);
            foreach (OERole role in abset.GetRoles())
            {
              Console.WriteLine("{0} {1}", role.GetName(), OEChem.
↪OEMolToSmiles(fragment));
            }
        }
        return 0;
```

(continues on next page)

(continued from previous page)

```
    }
}
```

The output of *Listing 4* is the following:

```
Framework c1ccc(c(c1)CC(=O)N)[SH4]NC(CC(=O)N)Nc2ncncn2
Ring c1ccc(cc1)CC(=O)N.c1ncncn1
Linker C(C(N)N[SH5])C(=O)N
Sidechain CC.CO.O.O
```

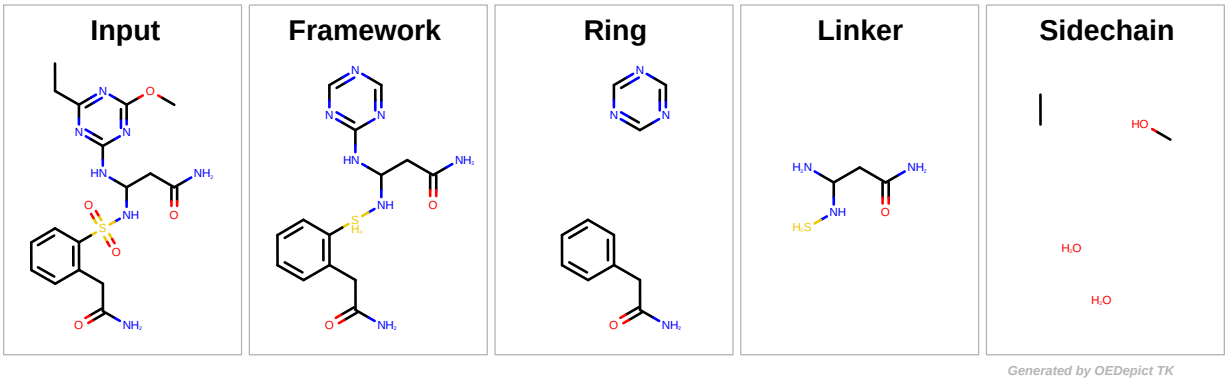


Fig. 4: **Example of Bemis Murcko fragmentation including custom substituents on framework**

2.2 Matched Molecule Pair Analysis

This overview is intended for users which have a working knowledge of matched pair analysis in the context of medicinal chemistry workflows. If the concept of matched pair analysis is unfamiliar, the following references are highly recommended as background to this powerful strategy ([Griffen-2011], [Kramer-2014], [Papadatos-2010], [Warner-2010]).

The *OEMedChem TK* provides the ability to index a set of input structures and identify matched molecular pairs. Matched Molecular Pair Analysis (MMPA) is becoming recognized as a powerful tool for the extraction of the effects of chemical changes on properties of interest in large data sets. One of the powers of the MMPA approach stems from the assumption that it is easier to predict differences in an activity or a property than to predict an actual value. A matched molecular pair is considered to be a pair of compounds that differ by a small localized change in a chemical substituent. The magnitude of the localized change that is considered acceptable is highly specific to the chemist or analyst. For a given matched molecular pair, one can consider the chemical difference between two compounds to embody a virtual chemical transformation with an accompanying change in a property of interest. For a homologous series of input structures, a set of matched molecular pairs for each transformation can be identified and analyzed for statistical relevance, sorting and/or ranking. Similarly, large input data sets can be mined for interesting or desired property changes as a function of structure.

A common method of identifying matched molecular pairs is an `NxM` pairwise maximum common subgraph (MCS) analysis of an input data set. Owing to the complexity and performance of MCS algorithms generally, *OpenEye* instead chose to use a molecular fragmentation approach (cf [Hussain-2010]) to identify matched molecular pairs and to deliver robust performance characteristics over a wider variety of input data set sizes. Additionally, this approach is entirely data-driven and does not require *a priori* defined cores. A small set of fragmentation strategies are currently provided, and future releases will expose additional fragmentation types and capabilities. MMPA as a technique is relatively new and there have been spirited discussions about the applicability of the method when applied to various measured

properties. *OpenEye* encourages users to critically examine the results available from such an analysis and to evaluate the applicability or non-applicability of property predictions with sound statistical analyses.

The general steps involved in an MMP analysis are,

- Prepare the input structures
- Prune input structure data to the property or properties of interest
- Select indexing filters to identify the desired "localized change" in structures
- Index the input structures
- Extract transformations and/or matched molecular pairs and data
- Optionally save the index for subsequent analyses

The internal index captures the common cores for the identified matched molecular pairs and the set of substituent changes associated with each core. Once the index has been generated, it can be used to,

- Extract the virtual chemical transformations and set of delta properties for the MMPs
- Extract the MCS cores of MMPs for clique or binning analysis
- Extract statistics for property changes from a series of substituents on a common core
- Identify or remove singletons or unrelated chemical frameworks from the input structures

Matched molecular pair transformations inherently have the concept of a chemical context for any given set of matched pairs. At the site (or sites) of the substituent differences in the two compounds one can consider the accompanying effects of the adjacent common core environment for the pair. To this end, the extraction of the matched pair transformations supports the ability to tune the amount of the adjacent common core to be included as part of the transformation. As more chemical context is included, more differentiation among the set of transformations occurs, generally resulting in fewer numbers of matched molecular pairs associated with each transformation. Thus, requesting a *OEMatchedPairContext.Bond0* context when extracting transformations from an index is likely to produce the largest set of matched pairs associated with each transform, but at the expense of coalescing possibly unrelated chemical environments into the same transformation bin. That is, a `Cl>>Br` transformation on a ring or aromatic ring is likely quite different compared to the same transformation in a non-ring portion of the structure with regards to the properties of interest for an MMP analysis.

The *OEMatchedPairAnalyzer* is the utility class the provides the bulk of the Matched Pair Analysis capabilities in *OEMedChem TK*, see docs for additional details.

2.2.1 Introductory Usage of Matched Molecular Pair Analysis

As a trivial example, suppose a chemist is interested in extracting all the "simple" substituent changes in a large input set. One possible way to accomplish this is to define some limits to what constitutes a "simple" substituent change and run an MMP analysis of the input structures. Here we define a "simple" substituent change to be singly attached groups only, and where the size of the group cannot exceed 20% of the input structure heavy atom count.

```
       // create options class with defaults
       OEMatchedPairAnalyzerOptions mmpOpts = new␣
↪OEMatchedPairAnalyzerOptions();

       // for 'simple' pairs, alter default indexing options
       // - single cuts only, heavy atom substituents only (HMember indexing␣
↪off)
       mmpOpts.SetOptions(OEMatchedPairOptions.SingleCuts |
                          OEMatchedPairOptions.ComboCuts |
                          OEMatchedPairOptions.UniquesOnly);
```

(continues on next page)

(continued from previous page)

```
        // - limit substituent size to no more than 20% of input structure
        mmpOpts.SetIndexableFragmentRange(80.0F, 100.0F);

        // create analyzer class with nondefault options
        OEMatchedPairAnalyzer mmpAnalyzer = new OEMatchedPairAnalyzer(mmpOpts);

        // ignore common index status returns
        String sIgnoreStatus = "FragmentRangeFilter,DuplicateStructure,
↪FragmentationLimitFilter,HeavyAtomFilter";

        // index the input structures
        int recindex = 0;
        OEGraphMol mol = new OEGraphMol();
        while (OEChem.OEReadMolecule(ifs, mol))
        {
            ++recindex;

            // consider only the largest input fragment
            OEChem.OEDeleteEverythingExceptTheFirstLargestComponent(mol);
            // ignore stereochemistry
            OEChem.OEUncolorMol(mol, (OEUncolorStrategy.RemoveAtomStereo |
                                      OEUncolorStrategy.RemoveBondStereo));

            // explicitly provide a 1-based index to refer to indexed␣
↪structures
            //   - to allow references back to external data elsewhere
            int status = mmpAnalyzer.AddMol(mol, recindex);
            if (status != recindex)
                if (sIgnoreStatus.IndexOf(OEMedChem.
↪OEMatchedPairIndexStatusName(status)) < 0)
                    OEChem.OEThrow.Warning(String.Format("{0}: molecule␣
↪indexing error, status={1}",
                                                         recindex,
                                                         OEMedChem.
↪OEMatchedPairIndexStatusName(status)));

            // if limiting input, quit after limit
            if (maxrecs != 0 && recindex >= maxrecs)
                break;
        }

        Console.WriteLine("Index complete, matched pairs = {0}", mmpAnalyzer.
↪NumMatchedPairs());

        OEMatchedPairTransformExtractOptions extractOptions = new␣
↪OEMatchedPairTransformExtractOptions();
        // specify amount of chemical context at the site of the substituent␣
↪change
        //   in the transform
        extractOptions.SetContext(OEMatchedPairContext.Bond0);
        // specify how transforms are extracted (direction and allowed␣
↪properties)
```

(continues on next page)

(continued from previous page)

```
        extractOptions.SetOptions(OEMatchedPairTransformExtractMode.Sorted |
                                  OEMatchedPairTransformExtractMode.NoSMARTS |
                                  OEMatchedPairTransformExtractMode.
↪AddMCSCorrespondence);

        // walk the transforms and print the matched pairs
        int xfmidx = 0;
        foreach (OEMatchedPairTransform mmpxform in OEMedChem.
↪OEMatchedPairGetTransforms(mmpAnalyzer, extractOptions))
        {
            ++xfmidx;
            Console.WriteLine("{0:D} {1}", xfmidx, mmpxform.GetTransform());
            // dump matched molecular pairs and index identifiers
            //   (recindex from indexing loop above)
            foreach (OEMatchedPair mmppair in mmpxform.GetMatchedPairs())
            {
                Console.WriteLine("\tmatched pair molecule indices=({0},{1})",
                                  mmppair.FromIndex(),
                                  mmppair.ToIndex());

            }
        }
```

Sample output of the example above is the following:

```
Index complete, matched pairs = 146
 1 [c:1]Br>>[c:1]F
  matched pair molecule indices=(4793,1225)
  matched pair molecule indices=(1304,9796)
  matched pair molecule indices=(9515,6129)
  matched pair molecule indices=(4201,2611)
 2 [c:1]Br>>[c:1]OCC
  matched pair molecule indices=(7846,9727)
  matched pair molecule indices=(1547,2939)
  matched pair molecule indices=(5128,8928)
 3 [c:1]Cl>>[c:1]F
  matched pair molecule indices=(8116,7661)
  matched pair molecule indices=(239,6631)
  matched pair molecule indices=(9225,8434)
 4 [c:1]F>>[c:1]OC
  matched pair molecule indices=(3862,8684)
  matched pair molecule indices=(1349,1434)
  matched pair molecule indices=(416,3900)
 5 [c:1]Br>>[c:1]SC
  matched pair molecule indices=(6364,8527)
  matched pair molecule indices=(1999,6845)
 6 [c:1]C#N>>[c:1]F
  matched pair molecule indices=(2760,5231)
  matched pair molecule indices=(6677,241)
 7 [c:1]C>>[c:1]Cl
  matched pair molecule indices=(1191,1605)
  matched pair molecule indices=(9591,4139)
```

(continues on next page)

(continued from previous page)

```
 8 [c:1]F>>[c:1]N(C)C
  matched pair molecule indices=(4012,853)
  matched pair molecule indices=(1779,1233)
 9 [C:1]C>>[C:1]O
  matched pair molecule indices=(2072,3429)
10 [C:1]C>>[C:1]OCC
  matched pair molecule indices=(9803,9538)
...
```

CHAPTER

THREE

EXAMPLES

3.1 OEMedChem TK Examples

The following table lists the currently available **OEMedChem TK** examples:

Program	Description
OEMedChemInfo	Print library version information
BemisMurckoPerception	Bemis-Murcko region perception
ChEMBLsolubility	Apply solubility transformations found from ChEMBL structures
CreateMMPIndex	Generation of an MMP index from a Matched Pair analysis
CreateMMPIndexThreaded	Multi-threaded generation of an MMP index from a Matched Pair analysis
MatchedPairTransform	Load a Matched Pair index and apply transformations
MatchedPairTransformList	Load a Matched Pair index and list of transformations
MatchedPairTransformProbe	Load a Matched Pair index and list of transformations related to a probe structure
CreateMCSFragDatabase	Generation of an MCS fragment index from an MCS fragmentation analysis
MCSFragDatabase	Load an MCS fragment index and report similarity scores for probe structures
MCSFragOccurrence	Load an MCS fragment index and report occurrence counts for common fragments

Examples:

3.1.1 Bemis Murcko perception

A program that perceives the [Bemis-1996] regions of the input structures and outputs role information and annotations as SD data in the output. Uncoloring of the fragment regions to remove atom types and properties can optionally be requested for the output SMILES string annotations.

See also

- *OEGetBemisMurcko* function

Command Line Interface

```
Usage: ./BemisMurckoPerception input.sdf output.sdf [ -uncolor ] [-
↪unsaturatedHeteroBonds] [ -smartsSubstituents substituentsString ]
[-regionType regionTypeString]
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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.  OPENEYE 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.
*/
//****************************************************************************
//* Utility to fragment the input structures by Bemis-Murcko rules
//* ---------------------------------------------------------------------------
//* BemisMurckoPerception.py [ -uncolor ] [-i] <input_mols> [-o] <output_mols>
//* [-regionType ] <region_type> [-smartsSubstituents] <sub_smarts>
//* [-unsaturatedHeteroBonds]
//* input_mols: filename of molecules to fragment and uncolor
//* output_mols: filename of input structure with SDData of perceived regions
//* [ -uncolor ]: optional arg to request uncoloring of output fragment info
//* [ -unsatHero]: optional flag to include unsaturated bonds to hetero atoms
//* [ -smartsSubstituents]: SMARTS of substituents to include on the main Bemis-Murcko␣
↪framework
//****************************************************************************
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;

public class BemisMurckoPerception
{
    public static void Main(string[] args)
    {
        OEInterface itf = new OEInterface(interfaceData, "BemisMurckoPerception", args);
```

(continues on next page)

(continued from previous page)

```
        // flag on command line indicates uncoloring option or not;
        bool bUncolor = itf.GetBool("-uncolor");

        // optional unsaturated hetero bonds flag
        bool bUnsaturatedHeteroBonds = itf.GetBool("-unsatHetero");

        // optional user-specified SMARTS strings for substituents
        string smartsSubstituentsString = itf.GetString("-smartsSubstituents");

        // optional user-specified SMARTS strings for substituents
        string regionTypeString = itf.GetString("-regionType");

        OEBemisMurckoOptions options = new OEBemisMurckoOptions();
        if (bUnsaturatedHeteroBonds)
          options.SetUnsaturatedHeteroBonds(true);

        if (!String.Equals(smartsSubstituentsString,"None")) {
          OESubSearch ss = new OESubSearch();
          if (!ss.Init(smartsSubstituentsString))
            OEChem.OEThrow.Fatal("Invalid SMARTS for subsearch.");
          options.SetSubstituentSearch(ss);
        }

        if (regionTypeString.Length > 0)
          options.SetRegionType(regionTypeString);

        // input structure(s) to transform;
        oemolistream ifsmols = new oemolistream();
        if (!ifsmols.open(itf.GetString("-i")))
            OEChem.OEThrow.Fatal("Unable to open file for reading: " + itf.GetString("-i
↪"));

        // save output structure(s) to this file;
        oemolostream ofs = new oemolostream();
        if (!ofs.open(itf.GetString("-o")))
            OEChem.OEThrow.Fatal("Unable to open file for writing: " + itf.GetString("-o
↪"));
        if (!OEChem.OEIsSDDataFormat(ofs.GetFormat()))
            OEChem.OEThrow.Fatal("Output file format does not support SD data: " + itf.
↪GetString("-o"));

        int irec = 0;
        int ototal = 0;
        OEGraphMol frag = new OEGraphMol();
        foreach (OEGraphMol mol in ifsmols.GetOEGraphMols())
        {
            ++irec;
            OEChem.OEDeleteEverythingExceptTheFirstLargestComponent(mol);

            int regions = 0;
            foreach (OEAtomBondSet abset in OEMedChem.OEGetBemisMurcko(mol, options))
```

(continues on next page)

(continued from previous page)

```
            {
                ++regions;
                // create a fragment from the perceived region;
                OEChem.OESubsetMol(frag, mol, abset, true);
                if (bUncolor)
                {
                    // ignore 3D stereo parities
                    if (frag.GetDimension() == 3) {
                        frag.SetDimension(0);
                    }
                    // uncolor the fragment;
                    OEChem.OEUncolorMol(frag);
                }
                string smi = OEChem.OEMolToSmiles(frag);
                // annotate the input molecule with the role information;
                foreach (OERole role in abset.GetRoles())
                    OEChem.OEAddSDData(mol, role.GetName(), smi);

            }
            if (regions == 0)
            {
                string name = mol.GetTitle();
                if (name.Length == 0)
                    name = "Record " + irec.ToString();
                OEChem.OEThrow.Warning(name + ": no perceived regions");
            }
            else
            {
                ++ototal;
                OEChem.OEWriteMolecule(ofs, mol);
            }
        }
        if (irec == 0)
            OEChem.OEThrow.Fatal("No records in input structure file to perceive");

        if (ototal == 0)
            OEChem.OEThrow.Warning("No annotated structures generated");

        Console.WriteLine("Input molecules={0:D}, output annotated {1:S}molecules={2:D}",
                          irec, ((bUncolor) ? "(uncolored) " : ""), ototal);

    }
    private static string interfaceData = @"
!BRIEF [ -uncolor ] [-i] <infile1> [-o] <infile2> [-uhb] [-r] <regionType> [-sub]
↪<smartsSubs>
!PARAMETER -i
  !ALIAS -in
  !ALIAS -input
  !TYPE string
  !REQUIRED true
  !BRIEF Input structure file name
  !KEYLESS 1
```

(continues on next page)

(continued from previous page)

```
!END
!PARAMETER -o
  !ALIAS -out
  !ALIAS -output
  !TYPE string
  !REQUIRED true
  !BRIEF Output SD file name
  !KEYLESS 2
!END
!PARAMETER -uncolor
  !ALIAS -u
  !TYPE bool
  !DEFAULT false
  !BRIEF Uncolor output molecules
!END
!PARAMETER -unsatHetero
  !ALIAS -uhb
  !TYPE bool
  !DEFAULT false
  !BRIEF Include sidechains on main framework if connected by unsaturated bonds to␣
↪hetero atoms.
!END
!PARAMETER -regionType
  !ALIAS -r
  !TYPE string
  !DEFAULT All
  !BRIEF Region type of fragments to include. Valid inputs are All, Framework, Ring,␣
↪Linker, and Sidechain.
!END
!PARAMETER -smartsSubstituents
  !ALIAS -sub
  !TYPE string
  !DEFAULT None
  !BRIEF SMARTS string for custom substituents to be included in framework.
!END
";
}
```

3.1.2 Apply ChEMBL solubility transformations

A utility that reads an input set of structures and outputs transformed structures along with annotations containing of the original data based on the matched pairs discovered from indexing solubility data from the [ChEMBL24-2018] data set.

See also

- *OEApplyChEMBL24SolubilityTransforms* function

Command Line Interface

```
prompt> ChEMBLsolubility input.sdf output.sdf
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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.  OPENEYE 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.
*/
//****************************************************************************
//* Utility to apply ChEMBL24 solubility transforms to an input set of structures
//* ---------------------------------------------------------------------------
//* ChEMBLsolubility [-i] input_mols [-o] output_mols [ -verbose ] [ -context [0|2] ] [ -
↪minpairs # ]
//*
//* input_mols: filename of molecules to transform based on analysis
//* output_mols: filename to collect transformed molecules
//* [-verbose]: optional flag to request verbose progress
//* [-context #]: optional flag to request a specific chemistry context
//* [-minpairs #]: optional flag to request a minimum number of pairs to apply transforms
//****************************************************************************
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;

public class ChEMBLsolubility
{
    public static void Main(String[] args)
    {
        OEInterface itf = new OEInterface(interfaceData, "ChEMBLsolubility", args);

        bool verbose = itf.GetBool("-verbose");
```

(continues on next page)

(continued from previous page)

```
        // input structure(s) to transform;
        oemolistream ifsmols = new oemolistream();
        if (!ifsmols.open(itf.GetString("-i")))
            OEChem.OEThrow.Fatal("Unable to open file for reading: " + itf.GetString("-i
↪"));

        // save output structure(s) to this file;
        oemolostream ofs = new oemolostream();
        if (!ofs.open(itf.GetString("-o")))
            OEChem.OEThrow.Fatal("Unable to open file for writing: " + itf.GetString("-o
↪"));

        // request a specific context for the transform activity, here 0-bonds;
        int chemctxt = OEMatchedPairContext.Bond0;
        String askcontext = itf.GetString("-context");
        char ctxt = askcontext[0];
        switch (ctxt)
        {
            case '0':
                chemctxt = OEMatchedPairContext.Bond0;
                break;
            case '2':
                chemctxt = OEMatchedPairContext.Bond2;
                break;
            default:
                OEChem.OEThrow.Fatal("Invalid context specified: " + askcontext + ",␣
↪only 0|2 allowed");
                break;
        }

        uint minpairs = (uint)itf.GetInt("-minpairs");
        if (minpairs > 1 && verbose)
          Console.WriteLine("Requiring at least {0:D} matched pairs to apply␣
↪transformations", minpairs);

        int irec = 0;
        int ocnt = 0;
        int ototal = 0;
        foreach (OEGraphMol mol in ifsmols.GetOEGraphMols())
        {
            ++irec;
            OEChem.OEDeleteEverythingExceptTheFirstLargestComponent(mol);

            ocnt = 0;
            foreach (OEMolBase outmol in OEMedChem.
↪OEApplyChEMBL24SolubilityTransforms(mol, chemctxt, minpairs))
            {
                ++ocnt;
                OEChem.OEWriteMolecule(ofs, outmol);
            }
            if (ocnt == 0)
            {
```

(continues on next page)

(continued from previous page)

```
                String name = mol.GetTitle();
                if (name.Length == 0)
                    name = "Record " + irec.ToString();
                Console.WriteLine(name + ": did not produce any output");
                Console.WriteLine(OEChem.OEMolToSmiles(mol));
            }
            else
            {
                ototal += ocnt;
                if (verbose)
                    Console.WriteLine("Record: {0:D} transformation count={1:D} total␣
↪mols={2:D}",
                                      irec, ocnt, ototal);
            }
        }

        if (irec == 0)
            OEChem.OEThrow.Fatal("No records in input structure file to transform");

        if (ototal == 0)
            OEChem.OEThrow.Warning("No transformed structures generated");
        else
            Console.WriteLine("Input molecules={0:D} output molecules={1:D}",
                              irec, ototal);
    }
    private static String interfaceData = @"
!BRIEF [-i] <infile1> [-o] <infile2> [ -verbose ] [ -context [0|2]]
!PARAMETER -i
  !ALIAS -in
  !ALIAS -input
  !TYPE string
  !REQUIRED true
  !BRIEF Input file name
  !KEYLESS 1
!END
!PARAMETER -o
  !ALIAS -out
  !ALIAS -output
  !TYPE string
  !REQUIRED true
  !BRIEF Output file name
  !KEYLESS 2
!END
!PARAMETER -verbose
  !ALIAS -v
  !TYPE bool
  !DEFAULT false
  !BRIEF Verbose output
!END
!PARAMETER -context
  !ALIAS -c
  !TYPE string
```

(continues on next page)

(continued from previous page)

```
  !DEFAULT 0
  !BRIEF Chemistry context for output
!END
!PARAMETER -minpairs 2
   !TYPE int
   !DEFAULT 0
   !BRIEF require at least -minpairs to apply the transformations (default: all)
!END";
}
```

3.1.3 Matched Pair analysis generation of a MMP index

A program that performs a matched pair analysis of a set of structures for indexing and saves the generated index file for subsequent loading and querying.

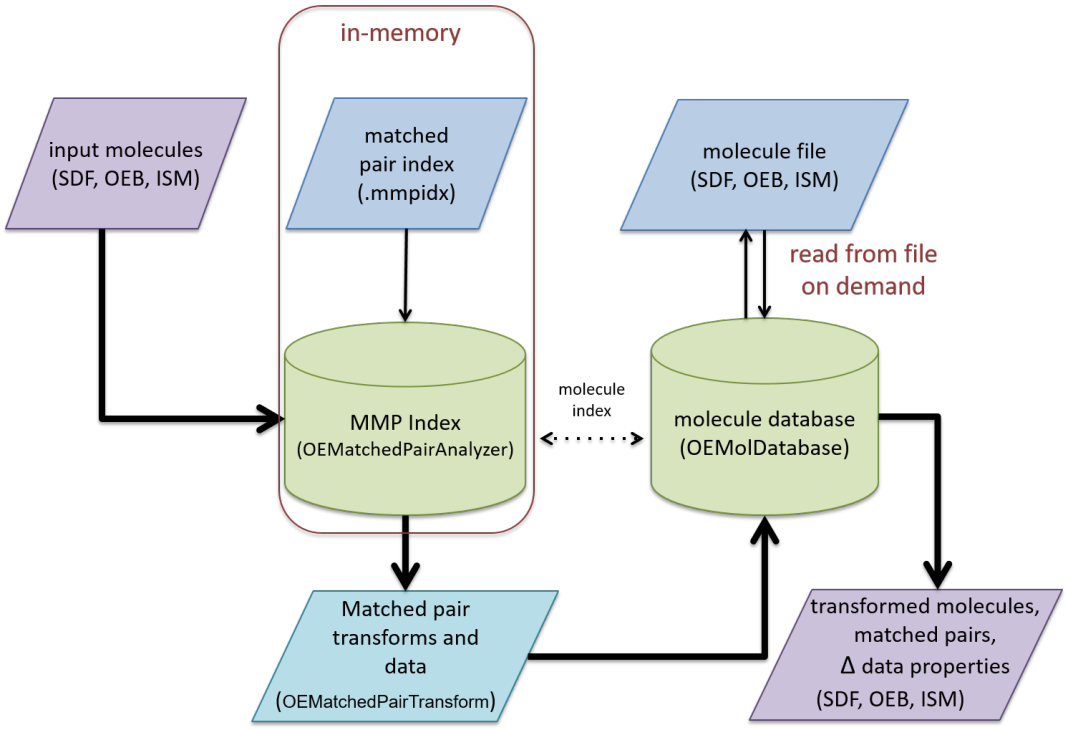


Fig. 1: **Schematic representation of the Matched Pair Analysis process**

See also

- *OEMatchedPairAnalyzer* class
- *OEMatchedPairAnalyzerOptions* class
- *OEMatchedPairApplyTransforms* function

Command Line Interface

```
prompt> CreateMMPIndex index.sdf output.mmpidx
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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.  OPENEYE 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.
*/
//**************************************************************************
//* Utility to perform a matched pair analysis on a set of structures
//*  and save the index for subsequent analysis
//* ------------------------------------------------------------------------
//* CreateMMPIndex index_mols output_index
//*
//* index_mols: filename of input molecules to analyze
//* output_index: filename of generated MMP index
//**************************************************************************/
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;
using System.Collections.Generic;

public class CreateMMPIndex
{
    private static void MMPIndex(OEInterface itf)
    {
        // output index file
        String mmpindexfile = itf.GetString("-output");
        if (!OEMedChem.OEIsMatchedPairAnalyzerFileType(mmpindexfile))
```

(continues on next page)

(continued from previous page)

```
            OEChem.OEThrow.Fatal("Output file is not a matched pair index type - needs .
↪mmpidx extension: " +
                                 mmpindexfile);

        // create options class with defaults
        OEMatchedPairAnalyzerOptions mmpopts = new OEMatchedPairAnalyzerOptions();
        // set up options from command line
        if (!OEMedChem.OESetupMatchedPairIndexOptions(mmpopts, itf))
            OEChem.OEThrow.Fatal("Error setting matched pair indexing options!");

        // checking input structures
        oemolistream ifsindex = new oemolistream();
        if (!ifsindex.open(itf.GetString("-input")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-input") + " for␣
↪reading");

        // get requested verbosity setting
        bool verbose = itf.GetBool("-verbose");
        bool vverbose = itf.GetBool("-vverbose");
        if (vverbose)
            verbose = vverbose;

        int maxrec = Math.Max(itf.GetInt("-maxrec"), 0);
        int statusrec = itf.GetInt("-status");

        if (itf.GetBool("-exportcompress"))
        {
            if (!mmpopts.SetOptions(mmpopts.GetOptions() | OEMatchedPairOptions.
↪ExportCompression))
                OEChem.OEThrow.Warning("Error enabling export compression!");
        }

        bool stripstereo = itf.GetBool("-stripstereo");
        bool stripsalts = itf.GetBool("-stripsalts");

        bool alldata = itf.GetBool("-allSD");
        bool cleardata = itf.GetBool("-clearSD");

        IList<string> keepFields = new List<string>();
        if (itf.HasString("-keepSD"))
        {
            foreach (String field in itf.GetStringList("-keepSD"))
                keepFields.Add(field);
            if (verbose && (alldata || cleardata)) {
                OEChem.OEThrow.Info("Option -keepSD overriding -allSD, -clearSD");
            }
            alldata = false;
            cleardata = false;
        }
        else if (cleardata) {
            alldata = false;
            keepFields.Add("-CLEARSD");
```

(continues on next page)

(continued from previous page)

```
        }
        else {
            keepFields.Add("-ALLSD");
            if (verbose && !alldata)
                OEChem.OEThrow.Info("No SD data handling option specified, -allSD assumed
↪");
            alldata = true;
            cleardata = false;
        }

        if (verbose)
        {
            if (!mmpopts.HasIndexableFragmentHeavyAtomRange())
                OEChem.OEThrow.Info("Indexing all fragments");
            else
                Console.WriteLine("Limiting fragment cores to {0:F}-{1:F}% of input␣
↪molecules",
                                  mmpopts.GetIndexableFragmentRangeMin(), mmpopts.
↪GetIndexableFragmentRangeMax());

            if (statusrec != 0)
                OEChem.OEThrow.Info("Status output after every " + statusrec + " records
↪");

            if (maxrec != 0)
                OEChem.OEThrow.Info("Indexing a maximum of " + maxrec + " records");

            if (itf.GetBool("-exportcompress"))
                OEChem.OEThrow.Info("Removing singleton index nodes from index");

            if (stripstereo)
                OEChem.OEThrow.Info("Stripping stereo");

            if (stripsalts)
                OEChem.OEThrow.Info("Stripping salts");

            if (cleardata)
                OEChem.OEThrow.Info("Clearing all input SD data");
            else if (alldata)
                OEChem.OEThrow.Info("Retaining all input SD data");
            else if (keepFields.Count != 0)
            {
                String allfields = "";
                foreach (String field in itf.GetStringList("-i"))
                    allfields = allfields + field;
                OEChem.OEThrow.Info("Retaining floating point SD data fields: " +␣
↪allfields);
            }
        }

        // create indexing engine
        OEMatchedPairAnalyzer mmp = new OEMatchedPairAnalyzer(mmpopts);
```

(continues on next page)

(continued from previous page)

```
        // interpret SD fields as floating point data
        FilterSDData validdata = new FilterSDData(keepFields, true);

        // add molecules to be indexed
        int record = 0;
        int unindexed = 0;
        OEGraphMol mol = new OEGraphMol();
        while (OEChem.OEReadMolecule(ifsindex, mol))
        {
            if (!alldata)
            {
                // filter the input molecule SD data based on allowed fields
                validdata.FilterMolData(mol);
            }

            if (stripsalts)
                OEChem.OEDeleteEverythingExceptTheFirstLargestComponent(mol);

            if (stripstereo)
                OEChem.OEUncolorMol(mol,
                                    (OEUncolorStrategy.RemoveAtomStereo |
                                     OEUncolorStrategy.RemoveBondStereo |
                                     OEUncolorStrategy.RemoveGroupStereo));

            int status = mmp.AddMol(mol, record);
            if (status != record)
            {
                ++unindexed;
                if (vverbose)
                    OEChem.OEThrow.Info("Input structure not added to index, record=" +␣
↪record + " status=" + OEMedChem.OEMatchedPairIndexStatusName(status));
            }
            ++record;

            if (maxrec != 0 && record >= maxrec)
                break;

            if (statusrec != 0 && (record % statusrec) == 0)
                OEChem.OEThrow.Info("Records: " + record + " Indexed: " + (record -␣
↪unindexed) + " Unindexed: " + unindexed);
        }

        if (mmp.NumMols() == 0)
            OEChem.OEThrow.Fatal("No records in index structure file");

        if (mmp.NumMatchedPairs() == 0)
            OEChem.OEThrow.Fatal("No matched pairs found from indexing, use -fragGe,-
↪fragLe options to extend indexing range");

        if (!OEMedChem.OEWriteMatchedPairAnalyzer(mmpindexfile, mmp))
            OEChem.OEThrow.Fatal("Error serializing MMP index: " + mmpindexfile);
```

(continues on next page)

(continued from previous page)

```
        // return some status information
        OEChem.OEThrow.Info("Records: " + record + ", Indexed: " + mmp.NumMols() + ",␣
↪matched pairs: " + mmp.NumMatchedPairs());
    }

    public static void Main(String[] args)
    {
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, InterfaceData);
        OEMedChem.OEConfigureMatchedPairIndexOptions(itf);

        if (OEChem.OEParseCommandLine (itf, args, "CreateMMPIndex"))
            MMPIndex(itf);
    }

    // simple SD data filter class
    private class FilterSDData
    {
        private IList<string> fields;
        private bool asFloating;
        private bool allfields;
        private bool clearfields;

        private FilterSDData() {}

        public FilterSDData(IList<string> fieldlist, bool asfloating)
        {
            fields = fieldlist;
            asFloating = asfloating;
            allfields = false;
            clearfields = false;
            if (fieldlist.Count == 1)
            {
                String opt = fieldlist[0].ToUpper();
                if (opt == "-ALLSD")
                    allfields = true;
                else if (opt == "-CLEARSD")
                    clearfields = true;
            }
        }

        public int FilterMolData(OEMolBase mol)
        {
            if (!OEChem.OEHasSDData(mol))
                return 0;

            if (allfields)
                return -1;

            if (clearfields)
            {
```

(continues on next page)

(continued from previous page)

```
                OEChem.OEClearSDData(mol);
                return 0;
            }

            if (fields.Count == 0)
                return -1;

            int validdata = 0;
            IList<string> deletefields = new List<string>();

            foreach (OESDDataPair dp in OEChem.OEGetSDDataPairs(mol))
            {
                String tag  = dp.GetTag();
                if (!fields.Contains(tag))
                {
                    deletefields.Add(tag);
                    continue;
                }
                if (asFloating)
                {
                    bool asfloat = false;
                    try
                    {
                        float.Parse(dp.GetValue());
                        asfloat = true;
                    }
                    catch (FormatException)
                    {
                        OEChem.OEThrow.Warning("Failed to convert " + tag +
                                               " to numeric value (" + OEChem.
↪OEGetSDData(mol, tag) +
                                               ") in " + mol.GetTitle());
                    }
                    if (!asfloat)
                    {
                        deletefields.Add(tag);
                        continue;
                    }
                }
                ++validdata;
            }

            if (validdata == 0)
                OEChem.OEClearSDData(mol);
            else
            {
                foreach (String nuke in deletefields)
                    OEChem.OEDeleteSDData(mol, nuke);
            }

            return validdata;
        }
```

(continues on next page)

(continued from previous page)

```
    }

    private static String InterfaceData = @"
!BRIEF [ -input ] index_mols [ -output ] output_index
# createmmpindex interface file
!CATEGORY CreateMMPIndex
    !CATEGORY I/O
        !PARAMETER -input 1
          !ALIAS -in
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of structures to index
          !KEYLESS 1
        !END
        !PARAMETER -output 2
          !ALIAS -out
          !TYPE string
          !REQUIRED true
          !BRIEF Output filename for serialized MMP index
          !KEYLESS 2
        !END
    !END
    !CATEGORY indexing_options
        !PARAMETER -maxrec 1
           !TYPE int
           !DEFAULT 0
           !LEGAL_RANGE 0 inf
           !BRIEF process at most -maxrec records from -input (0: all)
        !END
        !PARAMETER -status 2
           !TYPE int
           !DEFAULT 0
           !LEGAL_RANGE 0 inf
           !BRIEF emit progress information every -status records (0: off)
        !END
        !PARAMETER -exportcompress 3
          !TYPE bool
          !DEFAULT 0
          !BRIEF Whether to remove singleton nodes on export of the MMP index (i.e., no␣
↪additional structures will be added to the index)
        !END
        !PARAMETER -verbose 4
           !TYPE bool
           !DEFAULT 0
           !BRIEF generate verbose output
        !END
        !PARAMETER -vverbose 5
           !TYPE bool
           !DEFAULT 0
           !BRIEF generate very verbose output
        !END
    !END
```

(continues on next page)

(continued from previous page)

```
    !CATEGORY molecule_SDData
        !PARAMETER -allSD 1
           !TYPE bool
           !DEFAULT 0
           !BRIEF retain all input SD data
        !END
        !PARAMETER -clearSD 2
           !TYPE bool
           !DEFAULT 0
           !BRIEF clear all input SD data
        !END
        !PARAMETER -keepSD 3
           !TYPE string
           !LIST true
           !BRIEF list of SD data tags of floating point data to *retain* for indexing␣
↪(all other SD data is removed)
        !END
    !END
    !CATEGORY molecule_processing
        !PARAMETER -stripstereo 1
          !TYPE bool
          !DEFAULT 0
          !BRIEF Whether to strip stereo from -input structures
        !END
        !PARAMETER -stripsalts 2
          !TYPE bool
          !DEFAULT 0
          !BRIEF Whether to strip salt fragments from -input structures
        !END
    !END
!END
";
}
```

3.1.4 Matched Pair analysis using a multi-process generation of an MMP index

A program that performs a multi-process matched pair analysis of a set of structures for indexing is available as a python-only utility.

Command Line Interface

```
prompt> CreateMMPIndexParallel.py [ -pool num_processes ] index.sdf output.mmpidx
```

Code

3.1.5 Matched Pair analysis using a multi-threaded generation of a MMP index

A program that performs a multi-threaded matched pair analysis of a set of structures for indexing and saves the generated index file for subsequent loading and querying.

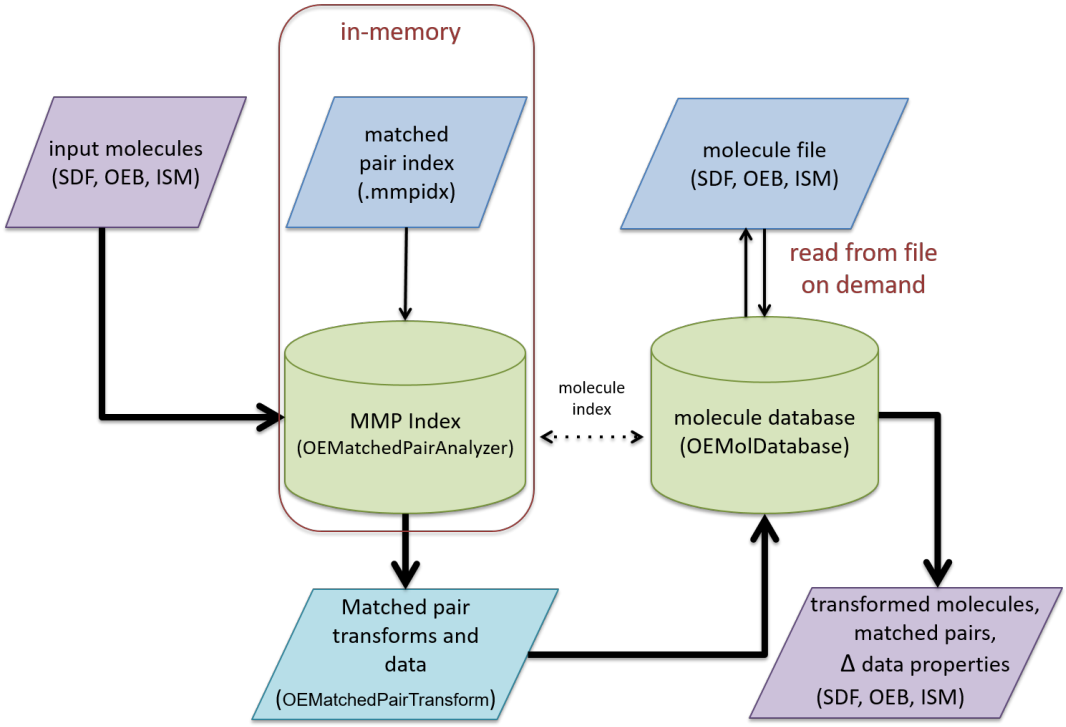


Fig. 2: **Schematic representation of the Matched Pair Analysis process**

See also

- *OEMatchedPairAnalyzer* class
- *OEMatchedPairAnalyzerOptions* class
- `OEMatchedPairApplyTransforms` function

Command Line Interface

```
prompt> CreateMMPIndexThreaded [ -threads num_threads ] index.sdf output.mmpidx
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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,
```

(continues on next page)

(continued from previous page)

```
EXPRESS OR IMPLIED.  OPENEYE 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.
*/
///**************************************************************************
//* Utility to perform a matched pair analysis on a set of structures
//*  using a multithreaded API and save the index for subsequent analysis
//* ------------------------------------------------------------------------
//* CreateMMPIndexThreaded index_mols output_index
//*
//* index_mols: filename of input molecules to analyze
//* output_index: filename of generated MMP index
//**************************************************************************
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;

public class CreateMMPIndexThreaded
{
    private static void MMPIndex(OEInterface itf)
    {
        // output index file
        String mmpindexfile = itf.GetString("-output");
        if (!OEMedChem.OEIsMatchedPairAnalyzerFileType(mmpindexfile))
            OEChem.OEThrow.Fatal("Output file is not a matched pair index type - needs .
↪mmpidx extension: " +
                                 mmpindexfile);

        // create options class with defaults
        OEMatchedPairAnalyzerOptions mmpopts = new OEMatchedPairAnalyzerOptions();
        // set up options from command line
        if (!OEMedChem.OESetupMatchedPairIndexOptions(mmpopts, itf))
            OEChem.OEThrow.Fatal("Error setting matched pair indexing options!");

        // checking input structures
        oemolistream ifsindex = new oemolistream();
        if (!ifsindex.open(itf.GetString("-input")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-input") + " for␣
↪reading");
        ifsindex.close();

        // get requested verbosity setting
        bool verbose = itf.GetBool("-verbose");
        bool vverbose = itf.GetBool("-vverbose");
        if (vverbose)
            verbose = vverbose;

        int maxrec = Math.Max(itf.GetInt("-maxrec"), 0);

        if (verbose)
```

(continues on next page)

(continued from previous page)

```
        {
            if (!mmpopts.HasIndexableFragmentHeavyAtomRange())
                OEChem.OEThrow.Info("Indexing all fragments");
            else
                Console.WriteLine("Limiting fragment cores to {0:F}-{1:F}% of input␣
↪molecules",
                                  mmpopts.GetIndexableFragmentRangeMin(), mmpopts.
↪GetIndexableFragmentRangeMax());

            if (maxrec != 0)
                OEChem.OEThrow.Info("Indexing a maximum of " + maxrec + " records");

            if (itf.GetBool("-exportcompress"))
                OEChem.OEThrow.Info("Removing singleton index nodes from index");
        }

        if (itf.GetBool("-exportcompress"))
        {
            if (!mmpopts.SetOptions(mmpopts.GetOptions() | OEMatchedPairOptions.
↪ExportCompression))
                OEChem.OEThrow.Warning("Error enabling export compression!");
        }

        // set indexing options
        OECreateMMPIndexOptions indexopts = new OECreateMMPIndexOptions(mmpopts);

        // set requested verbosity setting
        if (vverbose)
            indexopts.SetVerbose(2);
        else if (verbose)
            indexopts.SetVerbose(1);

        // limit number of records to process
        indexopts.SetMaxRecord((uint)maxrec);

        // set number of threads to use
        indexopts.SetNumThreads((uint)itf.GetInt("-threads"));
        if (verbose)
        {
            if (indexopts.GetNumThreads() == 0)
                OEChem.OEThrow.Info("Using the maximum number of threads available");
            else
                OEChem.OEThrow.Info("Limiting indexing to " + indexopts.GetNumThreads()␣
↪+ " thread(s)");
        }

        if (verbose)
            OEChem.OEThrow.Info("Threaded indexing of " + itf.GetString("-input") + ",␣
↪all SD data will be preserved");

        // create index
        OECreateMMPIndexStatus indexstatus = OEMedChem.OECreateMMPIndexFile(mmpindexfile,
```

(continues on next page)

(continued from previous page)

```
                                                                    itf.
↪GetString("-input"),
                                                                    indexopts);
        if (!indexstatus.IsValid())
            OEChem.OEThrow.Fatal("Invalid status returned from indexing!");

        if (indexstatus.GetTotalMols() == 0)
            OEChem.OEThrow.Fatal("No records in index structure file: " + itf.GetString(
↪"-input"));

        if (indexstatus.GetNumMatchedPairs() == 0)
            OEChem.OEThrow.Fatal("No matched pairs found from indexing, " +
                                 "use -fragGe,-fragLe options to extend indexing range");

        // return some status information
        OEChem.OEThrow.Info("Records: " + indexstatus.GetTotalMols() +
                            ", Indexed: " + indexstatus.GetNumMols() +
                            ", matched pairs: " +indexstatus.GetNumMatchedPairs());
    }

    public static void Main(String[] args)
    {
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, InterfaceData);
        OEMedChem.OEConfigureMatchedPairIndexOptions(itf);

        if (OEChem.OEParseCommandLine (itf, args, "CreateMMPIndexThreaded"))
            MMPIndex(itf);
    }

    private static String InterfaceData = @"
!CATEGORY CreateMMPIndexThreaded
    !CATEGORY I/O
        !PARAMETER -input 1
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of structures to index
          !KEYLESS 1
        !END
        !PARAMETER -output 2
          !TYPE string
          !REQUIRED true
          !BRIEF Output filename for serialized MMP index
          !KEYLESS 2
        !END
    !END
    !CATEGORY options
        !PARAMETER -maxrec 1
           !TYPE int
           !DEFAULT 0
           !LEGAL_RANGE 0 inf
           !BRIEF process at most -maxrec records from -input (0: all)
```

(continues on next page)

(continued from previous page)

```
        !END
        !PARAMETER -verbose 2
           !TYPE bool
           !DEFAULT 0
           !BRIEF generate verbose output
        !END
        !PARAMETER -threads 3
           !TYPE int
           !DEFAULT 0
           !LEGAL_RANGE 0 inf
           !BRIEF limit number of indexing threads to -threads (0:default)
        !END
        !PARAMETER -exportcompress 4
          !TYPE bool
          !DEFAULT 0
          !BRIEF Whether to remove singleton nodes on export of the MMP index
        !END
        !PARAMETER -vverbose 5
           !TYPE bool
           !DEFAULT 0
           !BRIEF generate very verbose output
        !END
    !END
!END";
}
```

3.1.6 Matched Pair analysis and transformations

A program that loads a previously generated Matched Pair index, then reads an input set of structures and outputs transformed structures based on the matched pairs discovered during indexing.

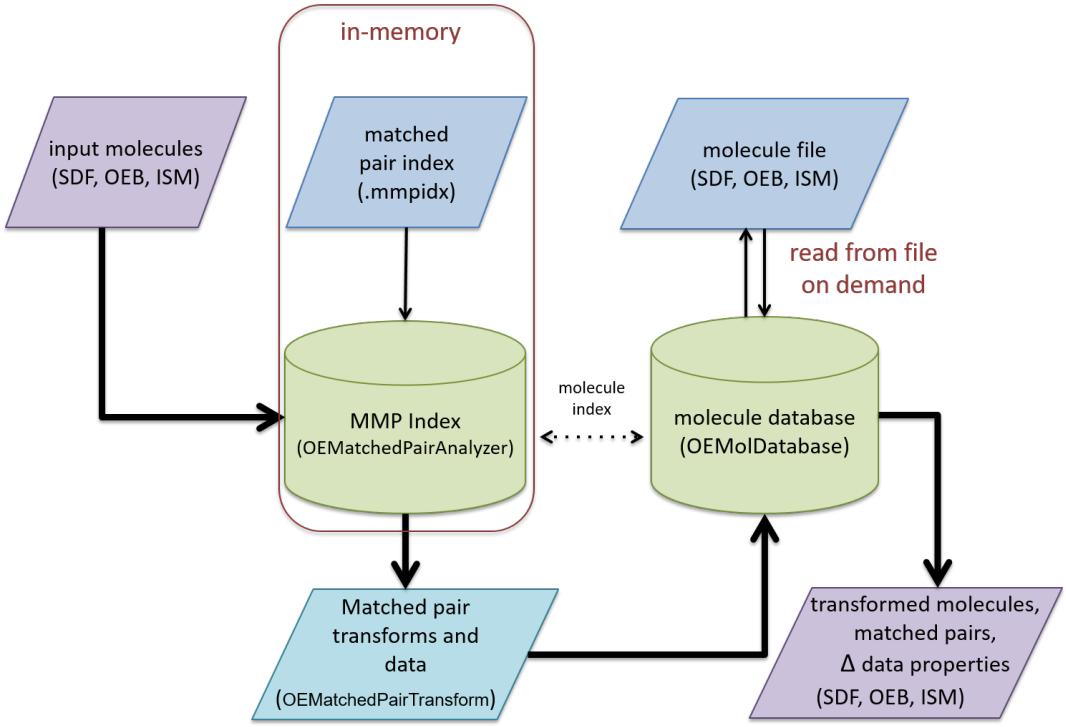


Fig. 3: **Schematic representation of the Matched Pair Analysis process**

See also

- *OEMatchedPairAnalyzer* class
- *OEMatchedPairAnalyzerOptions* class
- `OEMatchedPairApplyTransforms` function

Command Line Interface

```
prompt> MatchedPairTransform -mmpidx index.mmpidx -input input.sdf -output output.sdf
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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,
```

(continues on next page)

(continued from previous page)

```
EXPRESS OR IMPLIED.  OPENEYE 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.
*/
//****************************************************************************
//* Utility to load a previously generated MMP index
//*  and use the transformations discovered to alter a second set of structures
//* -------------------------------------------------------------------------
//* MatchedPairTransform mmp_index input_mols output_mols
//*
//* mmp_index: filename of matched pair index
//* input_mols: filename of molecules to transform based on analysis
//* output_mols: filename to collect transformed molecules
//****************************************************************************
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;

public class MatchedPairTransform
{
    private static void MMPTransform(OEInterface itf)
    {
        // input structure(s) to process
        oemolistream ifsmols = new oemolistream();
        if (!ifsmols.open(itf.GetString("-input")))
            OEChem.OEThrow.Fatal("Unable to open input file for reading: " + itf.
↪GetString("-input"));

        // check MMP index
        String mmpimport = itf.GetString("-mmpindex");
        if (!OEMedChem.OEIsMatchedPairAnalyzerFileType(mmpimport)) {
            OEChem.OEThrow.Fatal("Not a valid matched pair index input file, " +
                                 mmpimport);
        }

        // load MMP index
        OEMatchedPairAnalyzer mmp = new OEMatchedPairAnalyzer();
        if (!OEMedChem.OEReadMatchedPairAnalyzer(mmpimport, mmp)) {
            OEChem.OEThrow.Fatal("Unable to load index " + mmpimport);
        }

        if (mmp.NumMols() == 0) {
            OEChem.OEThrow.Fatal("No records in loaded MMP index file: " + mmpimport);
        }

        if (mmp.NumMatchedPairs() == 0) {
            OEChem.OEThrow.Fatal("No matched pairs found from indexing, " +
                                 "use -fragGe,-fragLe options to extend index range");
        }
```

(continues on next page)

(continued from previous page)

```
        // output (transformed) structure(s)
        oemolostream ofs = new oemolostream();
        if (!ofs.open(itf.GetString("-output")))
            OEChem.OEThrow.Fatal("Unable to open output file for writing: " + itf.
↪GetString("-output"));

        // request a specific context for the transform activity, here 0-bonds;
        int chemctxt = OEMatchedPairContext.Bond0;
        String askcontext = itf.GetString("-context");
        char ctxt = askcontext[0];
        switch (ctxt)
        {
            case '0':
                chemctxt = OEMatchedPairContext.Bond0;
                break;
            case '1':
                chemctxt = OEMatchedPairContext.Bond1;
                break;
            case '2':
                chemctxt = OEMatchedPairContext.Bond2;
                break;
            case '3':
                chemctxt = OEMatchedPairContext.Bond3;
                break;
            case 'a':
            case 'A':
                chemctxt = OEMatchedPairContext.AllBonds;
                break;
            default:
                OEChem.OEThrow.Fatal("Invalid context specified: " + askcontext + ",␣
↪only 0|1|2|3|A allowed");
                break;
        }

        bool verbose = itf.GetBool("-verbose");

        if (verbose) {
            // return some status information;
            OEChem.OEThrow.Info(mmpimport+ ": molecules: " + mmp.NumMols() + " matched␣
↪pairs: " + mmp.NumMatchedPairs());
        }

        uint minpairs = (uint)itf.GetInt("-minpairs");
        if (minpairs > 1 && verbose)
          Console.WriteLine("Requiring at least {0:D} matched pairs to apply␣
↪transformations", minpairs);

        int orec = 0;
        int ocnt = 0;
        int ototal = 0;
        foreach (OEGraphMol mol in ifsmols.GetOEGraphMols())
        {
```

(continues on next page)

(continued from previous page)

```
            ++orec;
            ocnt = 0;
            foreach (OEMolBase outmol in OEMedChem.OEMatchedPairApplyTransforms(mol, mmp,
↪ chemctxt, minpairs))
            {
                ++ocnt;
                OEChem.OEWriteMolecule(ofs, outmol);
            }
            ototal += ocnt;
            if (verbose && ocnt == 0)
            {
                String name = mol.GetTitle();
                if (name.Length == 0)
                    name = "Record " + orec.ToString();
                OEChem.OEThrow.Info(name + ": did not produce any output");
            }
        }
        if (orec == 0)
            OEChem.OEThrow.Fatal("No records in input structure file to transform");

        if (ototal == 0)
            OEChem.OEThrow.Fatal("No transformed structures generated");

        Console.WriteLine("Input molecules={0:D} Output molecules={1:D}", orec, ocnt);
    }

    public static void Main(String[] args)
    {
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, InterfaceData);

        if (OEChem.OEParseCommandLine(itf, args, "MatchedPairTransform"))
            MMPTransform(itf);
    }

    private static String InterfaceData = @"
!CATEGORY MatchedPairTransform
    !CATEGORY I/O
        !PARAMETER -mmpindex 1
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of serialized matched pair index to load
          !KEYLESS 1
        !END
        !PARAMETER -input 2
          !ALIAS -i
          !ALIAS -in
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of structures to process using matched pairs from -
↪mmpindex
          !KEYLESS 2
```

(continues on next page)

(continued from previous page)

```
        !END
        !PARAMETER -output 3
          !ALIAS -o
          !ALIAS -out
          !TYPE string
          !REQUIRED true
          !BRIEF Output filename
          !KEYLESS 3
        !END
    !END
    !CATEGORY options
        !PARAMETER -context 1
           !ALIAS -c
           !TYPE string
           !DEFAULT 0
           !BRIEF chemistry context to use for the transformation [0|1|2|3|A]
        !END
        !PARAMETER -minpairs 2
           !TYPE int
           !DEFAULT 0
           !LEGAL_RANGE 0 inf
           !BRIEF require at least -minpairs to apply the transformations (default: all)
        !END
        !PARAMETER -verbose 3
           !TYPE bool
           !DEFAULT 0
           !BRIEF generate verbose output
        !END
        !PARAMETER -nowarnings 4
           !TYPE bool
           !DEFAULT 1
           !BRIEF suppress warning messages from applying transformations
        !END
    !END
!END";
}
```

3.1.7 Matched Pair analysis and listing of transformations

A program that loads a previously generated Matched Pair index, then prints the transformations and matched pairs found from the analysis. If an (optional, numeric) data field is provided, the returned transformations can also be ranked by the largest increase in the delta property found from the matched pairs that define it.

See also

- *OEMatchedPairAnalyzer* class
- *OEMatchedPairAnalyzerOptions* class
- *OEMatchedPairGetTransforms* function

Command Line Interface

```
prompt> MatchedPairTransformList -mmpindex index.mmpidx [ -datafield sdDataField ]
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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.  OPENEYE 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.
*/
//****************************************************************************
//* Utility to load a previously generated MMP index
//*  and dump a listing of the transformations derived from matched pairs found
//* -------------------------------------------------------------------------
//* MatchedPairTransformList index_file
//*
//* index_file: filename of MMP index
//****************************************************************************
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;
using System.Collections.Generic;

public class MatchedPairTransformList
{
    // simple internal class to rank transform output
    public class MMPXform : IComparable<MMPXform>
    {
        public OEMatchedPairTransform xform;
        public float avg;
        public float std;
```

(continues on next page)

(continued from previous page)

```
        public int num;

        public MMPXform(OEMatchedPairTransform xfm,
                        float average, float stddev, int count)
        {
            xform = xfm;
            avg = average;
            std = stddev;
            num = count;
        }

        public int CompareTo(MMPXform another)
        {
            // sort descending by absolute value of the average
            double lAvg = Math.Abs(this.avg);
            double rAvg = Math.Abs(another.avg);

            int retval = 0;
            if (lAvg > rAvg)
                retval = -1;
            else if (lAvg < rAvg)
                retval = 1;
            return retval;
        }
    }

    private static float getAverage(List<float> floatList)
    {
        if (floatList.Count == 0)
            return 0;

        float sum = 0;
        foreach (float value in floatList)
        {
            sum += value;
        }
        return sum / floatList.Count;
    }

    private static float getStdDev(List<float> floatList)
    {
        if (floatList.Count <= 1)
            return 0;

        float average = getAverage(floatList);
        float sumOfDerivation = 0;
        foreach (float value in floatList)
        {
            sumOfDerivation += (value) * (value);
        }
        double sumOfDerivationAverage = sumOfDerivation / floatList.Count; // population,
↪ not sample
```

(continues on next page)

(continued from previous page)

```
        return (float)Math.Sqrt(sumOfDerivationAverage - (average * average));
    }

    private static void MMPTransformList(OEInterface itf)
    {
        // check MMP index
        String mmpimport = itf.GetString("-mmpindex");
        if (!OEMedChem.OEIsMatchedPairAnalyzerFileType(mmpimport)) {
            OEChem.OEThrow.Fatal("Not a valid matched pair index input file, " +
                                 mmpimport);
        }

        // load MMP index
        OEMatchedPairAnalyzer mmp = new OEMatchedPairAnalyzer();
        if (!OEMedChem.OEReadMatchedPairAnalyzer(mmpimport, mmp)) {
            OEChem.OEThrow.Fatal("Unable to load index " + mmpimport);
        }

        if (mmp.NumMols() == 0) {
            OEChem.OEThrow.Fatal("No records in loaded MMP index file: " + mmpimport);
        }

        if (mmp.NumMatchedPairs() == 0) {
            OEChem.OEThrow.Fatal("No matched pairs found from indexing, use -fragGe,-
↪fragLe options to extend index range");
        }

        // request a specific context for the transform activity, here 0-bonds;
        int chemctxt = OEMatchedPairContext.Bond0;
        String askcontext = itf.GetString("-context");
        char ctxt = askcontext[0];
        switch (ctxt)
        {
            case '0':
                chemctxt = OEMatchedPairContext.Bond0;
                break;
            case '1':
                chemctxt = OEMatchedPairContext.Bond1;
                break;
            case '2':
                chemctxt = OEMatchedPairContext.Bond2;
                break;
            case '3':
                chemctxt = OEMatchedPairContext.Bond3;
                break;
            case 'a':
            case 'A':
                chemctxt = OEMatchedPairContext.AllBonds;
                break;
            default:
                OEChem.OEThrow.Fatal("Invalid context specified: " +
                                     askcontext + ", only 0|1|2|3|A allowed");
```

(continues on next page)

(continued from previous page)

```
                break;
        }

        bool bPrintTransforms = itf.GetBool("-printlist");
        // if a data field was specified, retreive the SD data field name
        String field = null;
        if (itf.HasString("-datafield"))
            field = itf.GetString("-datafield");

        OEMatchedPairTransformExtractOptions extractOptions = new␣
↪OEMatchedPairTransformExtractOptions();
        // specify amount of chemical context at the site of the substituent change
        extractOptions.SetContext(chemctxt);
        // controls how transforms are extracted (direction and allowed properties)
        extractOptions.SetOptions(OEMatchedPairTransformExtractMode.Sorted |
                                  OEMatchedPairTransformExtractMode.NoSMARTS);

        // now walk the transforms from the indexed matched pairs
        List<MMPXform> xforms = new List<MMPXform>();

        int xfmidx = 0;
        foreach (OEMatchedPairTransform mmpxform in OEMedChem.
↪OEMatchedPairGetTransforms(mmp,

↪extractOptions))
        {
            ++xfmidx;
            if (bPrintTransforms)
                Console.WriteLine(" {0:D} {1}", xfmidx, mmpxform.GetTransform());

            int mmpidx = 0;
            List<float> prop = new List<float>();
            foreach (OEMatchedPair mmppair in mmpxform.GetMatchedPairs())
            {
                ++mmpidx;
                String mmpinfo = String.Format("\t {0:D}: ( {1:D}, {2:D})",
                                               mmpidx, mmppair.FromIndex(), mmppair.
↪ToIndex());

                foreach (String tag in mmppair.GetDataTags())
                {
                    mmpinfo = mmpinfo + String.Format(" {0}=({1},{2})",
                                                      tag,
                                                      mmppair.GetFromSDData(tag),
                                                      mmppair.GetToSDData(tag));
                    if (tag.Equals(field))
                    {
                        float fromValue = float.Parse(mmppair.GetFromSDData(tag));
                        float toValue = float.Parse(mmppair.GetToSDData(tag));
                        prop.Add(toValue - fromValue);
                    }
```

(continues on next page)

(continued from previous page)

```
                }
                if (bPrintTransforms)
                    Console.WriteLine(mmpinfo);
            }
            // skip if property not found
            if (prop.Count != 0)
            {
                // add
                MMPXform item = new MMPXform(mmpxform,
                        MatchedPairTransformList.getAverage(prop),
                        (float)MatchedPairTransformList.getStdDev(prop),
                        prop.Count);
                xforms.Add(item);
            }
        }
        if (field == null)
            return;

        if (xforms.Count == 0)
            OEChem.OEThrow.Error("No matched pairs found with " + field + " data");

        // sort the transforms by largest absolute delta property value
        xforms.Sort();

        Console.WriteLine("\n*** Transforms sorted by delta ({0})", field);

        int idx = 0;
        foreach (MMPXform xfm in xforms)
        {
            ++idx;
            if ((extractOptions.GetOptions() & OEMatchedPairTransformExtractMode.
↪NoSMARTS) != 0)
            {
                // not 'invertable' if SMARTS qualifiers were applied
                if (xfm.avg < 0.0F)
                {
                    xfm.avg = -1.0F * xfm.avg;
                    xfm.xform.Invert();
                }
            }
            Console.WriteLine("{0,2:D} {1}=(avg={2:F2},stdev={3:F2},num={4:D}) {5}", idx,
                              field,
                              xfm.avg,
                              xfm.std,
                              xfm.num,
                              xfm.xform.GetTransform());
        }

    }

    public static void Main(String[] args)
    {
```

(continues on next page)

(continued from previous page)

```
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, InterfaceData);

        if (OEChem.OEParseCommandLine(itf, args, "MatchedPairTransformList"))
            MMPTransformList(itf);
    }

    private static String InterfaceData = @"
# matchedpairtransformlist interface file
!BRIEF -mmpindex index.mmpidx [ -datafield sdDataField ]
!CATEGORY MatchedPairTransformList
    !CATEGORY I/O
        !PARAMETER -mmpindex 1
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of serialized matched pair index to load
        !END
    !END
    !CATEGORY options
        !PARAMETER -context 1
           !ALIAS -c
           !TYPE string
           !DEFAULT 0
           !BRIEF chemistry context to use for the transformation [0|1|2|3|A]
        !END
        !PARAMETER -printlist 2
           !ALIAS -p
           !TYPE bool
           !DEFAULT 1
           !BRIEF print all transforms and matched pairs
        !END
        !PARAMETER -datafield 3
           !ALIAS -d
           !TYPE string
           !BRIEF sort transforms based on delta change in this property
        !END
    !END
!END";
}
```

3.1.8 Matched Pair analysis and listing of transformations related to probe structures

A program that loads a previously generated Matched Pair index, then prints the transformations and matched pairs found from the analysis that are related to the provided probe structure(s). If an (optional, numeric) data field is provided, the returned transformations can also be ranked by the largest increase in the delta property found from the matched pairs that define it.

See also

- *OEMatchedPairAnalyzer* class

- *OEMatchedPairAnalyzerOptions* class
- *OEMatchedPairGetTransforms* function

Command Line Interface

```
prompt> MatchedPairTransformProbe [ -datafield sdDataField ] index.mmpidx probe_
↪molecules.sdf
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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.  OPENEYE 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.
*/
//****************************************************************************
//* Utility to load a matched pair index
//*  and dump a listing of transformations related to probe molecule(s)
//* -------------------------------------------------------------------------
//* MatchedPairTransformProbe mmp_index probe_mols
//*
//* mmp_index: filename of MMP index file
//* probe_mols: filename of input molecules to use as probes for related transforms
//****************************************************************************
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;
using System.Collections.Generic;

public class MatchedPairTransformProbe
{

    // simple internal class to rank transform output
    public class MMPXform : IComparable<MMPXform>
    {
        public OEMatchedPairTransform xform;
        public float avg;
        public float std;
```

(continues on next page)

(continued from previous page)

```
        public int num;

        public MMPXform(OEMatchedPairTransform xfm,
                        float average, float stddev, int count)
        {
            xform = xfm;
            avg = average;
            std = stddev;
            num = count;
        }

        public int CompareTo(MMPXform another)
        {
            // sort descending by absolute value of the average
            double lAvg = Math.Abs(this.avg);
            double rAvg = Math.Abs(another.avg);

            int retval = 0;
            if (lAvg > rAvg)
                retval = -1;
            else if (lAvg < rAvg)
                retval = 1;
            return retval;
        }
    }

    private static float getAverage(List<float> floatList)
    {
        if (floatList.Count == 0)
            return 0;

        float sum = 0;
        foreach (float value in floatList)
        {
            sum += value;
        }
        return sum / floatList.Count;
    }

    private static float getStdDev(List<float> floatList)
    {
        if (floatList.Count <= 1)
            return 0;

        float average = getAverage(floatList);
        float sumOfDerivation = 0;
        foreach (float value in floatList)
        {
            sumOfDerivation += (value) * (value);
        }
        double sumOfDerivationAverage = sumOfDerivation / floatList.Count; // population,
↪ not sample
```

(continues on next page)

(continued from previous page)

```
        return (float)Math.Sqrt(sumOfDerivationAverage - (average * average));
    }

    private static void MMPAnalyzeProbe(OEInterface itf)
    {
        // check MMP index
        String mmpimport = itf.GetString("-mmpindex");
        if (!OEMedChem.OEIsMatchedPairAnalyzerFileType(mmpimport)) {
            OEChem.OEThrow.Fatal("Not a valid matched pair index input file, " +␣
↪mmpimport);
        }

        // load MMP index
        OEMatchedPairAnalyzer mmp = new OEMatchedPairAnalyzer();
        if (!OEMedChem.OEReadMatchedPairAnalyzer(mmpimport, mmp)) {
            OEChem.OEThrow.Fatal("Unable to load index " + mmpimport);
        }

        if (mmp.NumMols() == 0)
            OEChem.OEThrow.Fatal("No records in input structure file for indexing");

        if (mmp.NumMatchedPairs() == 0)
            OEChem.OEThrow.Fatal("No matched pairs found from index file, " +
                                 "use -fragGe,-fragLe options to extend index range");

        // input probe structure(s)
        oemolistream ifs = new oemolistream();
        if (!ifs.open(itf.GetString("-input")))
            OEChem.OEThrow.Fatal("Unable to open input probe molecule file: " + itf.
↪GetString("-input"));

        // request a specific context for the transform activity, here 0-bonds;
        int chemctxt = OEMatchedPairContext.Bond0;
        String askcontext = itf.GetString("-context");
        char ctxt = askcontext[0];
        switch (ctxt)
        {
            case '0':
                chemctxt = OEMatchedPairContext.Bond0;
                break;
            case '1':
                chemctxt = OEMatchedPairContext.Bond1;
                break;
            case '2':
                chemctxt = OEMatchedPairContext.Bond2;
                break;
            case '3':
                chemctxt = OEMatchedPairContext.Bond3;
                break;
            case 'a':
            case 'A':
                chemctxt = OEMatchedPairContext.AllBonds;
```

(continues on next page)

(continued from previous page)

```
                break;
            default:
                OEChem.OEThrow.Fatal("Invalid context specified: " +
                                     askcontext + ", only 0|1|2|3|A allowed");
                break;
        }

        bool bPrintTransforms = itf.GetBool("-printlist");
        // if a data field was specified, retreive the SD data field name
        String field = null;
        if (itf.HasString("-datafield"))
            field = itf.GetString("-datafield");

        OEMatchedPairTransformExtractOptions extractOptions = new␣
↪OEMatchedPairTransformExtractOptions();
        // specify amount of chemical context at the site of the substituent change
        extractOptions.SetContext(chemctxt);
        // controls how transforms are extracted (direction and allowed properties)
        extractOptions.SetOptions(OEMatchedPairTransformExtractMode.Sorted |
                                  OEMatchedPairTransformExtractMode.NoSMARTS);

        uint record = 0;
        foreach (OEGraphMol mol in ifs.GetOEGraphMols())
        {
            ++record;
            Console.WriteLine("*** Probe molecule, {0} record={1:D}", OEChem.
↪OEMolToSmiles(mol), record);

            // walk the transforms from the indexed matched pairs that are related to␣
↪this probe molecule
            List<MMPXform> xforms = new List<MMPXform>();

            int xfmidx = 0;
            foreach (OEMatchedPairTransform mmpxform in OEMedChem.
↪OEMatchedPairGetTransforms(mmp,
                                                                                    ␣
↪    mol,
                                                                                    ␣
↪    extractOptions))
            {
                ++xfmidx;
                if (bPrintTransforms)
                    Console.WriteLine(" {0:D} {1}", xfmidx, mmpxform.GetTransform());

                // compute delta property
                int mmpidx = 0;
                List<float> prop = new List<float>();

                foreach (OEMatchedPair mmppair in mmpxform.GetMatchedPairs())
                {
                    ++mmpidx;
                    String mmpinfo = String.Format("\t{0:D}: ({1:D},{2:D})",
```

(continues on next page)

(continued from previous page)

```
                                                    mmpidx, mmppair.FromIndex(), mmppair.
↪ToIndex());

                    foreach (String tag in mmppair.GetDataTags())
                    {
                        mmpinfo = mmpinfo + String.Format(" {0}=({1},{2})",
                                                          tag,
                                                          mmppair.GetFromSDData(tag),
                                                          mmppair.GetToSDData(tag));
                        if (tag.Equals(field))
                        {
                            float fromValue = float.Parse(mmppair.GetFromSDData(tag));
                            float toValue = float.Parse(mmppair.GetToSDData(tag));
                            prop.Add(toValue - fromValue);
                        }

                    }
                    if (bPrintTransforms)
                        Console.WriteLine(mmpinfo);
                }
                // skip if property not found
                if (prop.Count != 0)
                {
                    xforms.Add(new MMPXform(mmpxform,
                                            MatchedPairTransformProbe.getAverage(prop),
                                            (float)MatchedPairTransformProbe.
↪getStdDev(prop),
                                            prop.Count));
                }
                else if (field == null)
                {
                    xforms.Add(new MMPXform(mmpxform,
                                            0.0F, 0.0F, mmpidx));
                }
            }
            if (xforms.Count == 0)
            {
                if (field != null)
                    Console.WriteLine("*** No related transforms found with {0}", field);
                else
                    Console.WriteLine("*** No related transforms");
            }
            else
            {
                if (field != null)
                {
                    Console.WriteLine("*** Related transforms sorted by delta ({0})",␣
↪field);
                    xforms.Sort();
                }
                else
                {
```

(continues on next page)

(continued from previous page)

```
                    Console.WriteLine("*** Related transforms");
                }

                int idx = 0;
                foreach (MMPXform xfm in xforms)
                {
                    ++idx;
                    if (field == null)
                    {
                        Console.WriteLine("{0,2:D} {1}", idx,
                                          xfm.xform.GetTransform());
                    }
                    else
                    {
                        if ((extractOptions.GetOptions() &␣
↪OEMatchedPairTransformExtractMode.NoSMARTS) != 0)
                        {
                            // not 'invertable' if SMARTS qualifiers were applied
                            if (xfm.avg < 0.0F)
                            {
                                xfm.avg = -1.0F * xfm.avg;
                                xfm.xform.Invert();
                            }
                        }
                        Console.WriteLine("{0,2:D} {1}=(avg={2:F2},stdev={3:F2},num={4:D}
↪) {5}", idx,
                                          field,
                                          xfm.avg,
                                          xfm.std,
                                          xfm.num,
                                          xfm.xform.GetTransform());
                    }
                }
            }
            Console.WriteLine("");
        }
    }

    public static void Main(String[] args)
    {
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, InterfaceData);

        if (OEChem.OEParseCommandLine(itf, args, "MatchedPairTransformProbe"))
            MMPAnalyzeProbe(itf);
    }

    private static String InterfaceData = @"
# matchedpairtransformprobe interface file
!CATEGORY MatchedPairTransformProbe
    !CATEGORY I/O
        !PARAMETER -mmpindex 1
```

(continues on next page)

(continued from previous page)

```
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of serialized matched pair index to load
          !KEYLESS 1
        !END
        !PARAMETER -input 2
          !ALIAS -in
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of probe molecule(s) to use to retrieve related␣
↪transforms
          !KEYLESS 2
        !END
    !END
    !CATEGORY options
        !PARAMETER -context 1
           !ALIAS -c
           !TYPE string
           !DEFAULT 0
           !BRIEF chemistry context to use for the transformation [0|1|2|3|A]
        !END
        !PARAMETER -printlist 2
           !ALIAS -p
           !ALIAS -pri
           !TYPE bool
           !DEFAULT 1
           !BRIEF print all transforms and matched pairs
        !END
        !PARAMETER -datafield 3
           !ALIAS -d
           !TYPE string
           !BRIEF sort transforms based on delta change in this property
        !END
    !END
!END";
}
```

3.1.9 MCS Fragment database generation

A program that performs an MCS fragmentation analysis of a set of structures and saves the generated index file for subsequent loading and querying.

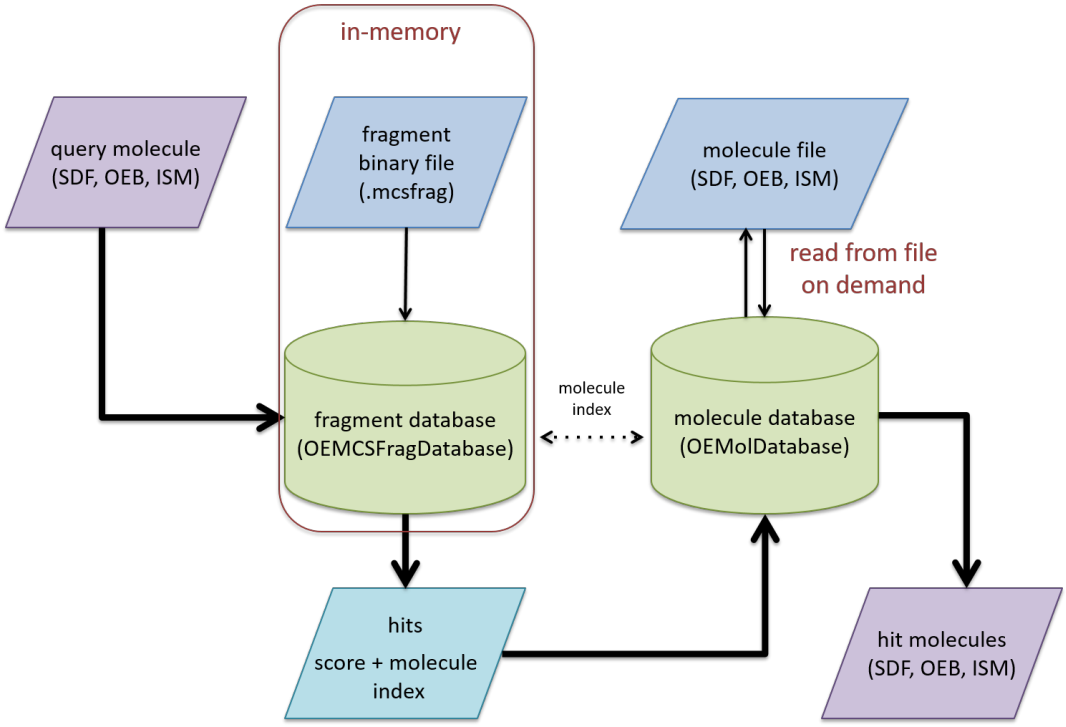


Fig. 4: **Schematic representation of the MCS fragment search process**

See also

- *OEMCSFragDatabase* class
- *OEMCSFragDatabaseOptions* class

Command Line Interface

```
prompt> CreateMCSFragDatabase index.sdf output.mcsfrag
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
```

(continues on next page)

(continued from previous page)

```
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.
*/
//*****************************************************************************
//* Utility to perform an MCS fragmentation on an input set of structures
//*  and save the index for subsequent analysis
//* ---------------------------------------------------------------------------
//* CreateMCSFragDatabase index_mols output_mcsindex
//*                       [ -verbose 1 ]  // optional verbosity
//*
//* index_mols: filename of input molecules to analyze
//* output_index: filename of MCS fragment index
//*****************************************************************************
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;

public class CreateMCSFragDatabase
{
    private static void MCSFragIndex(OEInterface itf)
    {
        // output index file
        String mcsindexfile = itf.GetString("-output");
        if (!OEMedChem.OEIsMCSFragDatabaseFileType(mcsindexfile))
            OEChem.OEThrow.Fatal("Output file is not a matched pair index type - needs .
↪mcsfrag extension: " +
                                 mcsindexfile);

        // create options class with defaults
        OEMCSFragDatabaseOptions mcsopts = new OEMCSFragDatabaseOptions();
        // set up options from command line
        if (!OEMedChem.OESetupMCSFragDatabaseOptions(mcsopts, itf))
            OEChem.OEThrow.Fatal("Error setting matched pair indexing options!");

        // input structures to index
        oemolistream ifsindex = new oemolistream();
        if (!ifsindex.open(itf.GetString("-input")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-input") + " for␣
↪reading");

        // get requested verbosity setting
        bool verbose = itf.GetBool("-verbose");

        int maxrec = Math.Max(itf.GetInt("-maxrec"), 0);
        int statusrec = itf.GetInt("-status");

        if (verbose)
        {
            if (!mcsopts.HasIndexableFragmentHeavyAtomRange())
                OEChem.OEThrow.Info("Indexing all fragments");
```

(continues on next page)

(continued from previous page)

```
            else
                Console.WriteLine("Limiting fragment cores to {0:F}-{1:F}% of input␣
↪molecules",
                                  mcsopts.GetIndexableFragmentRangeMin(), mcsopts.
↪GetIndexableFragmentRangeMax());

            if (statusrec != 0)
                OEChem.OEThrow.Info("Status output after every " + statusrec + " records
↪");

            if (maxrec != 0)
                OEChem.OEThrow.Info("Indexing a maximum of " + maxrec + " records");
        }

        // create indexing engine
        OEMCSFragDatabase mcsdb = new OEMCSFragDatabase(mcsopts);

        // add molecules to be indexed
        int record = 0;
        int unindexed = 0;
        OEGraphMol mol = new OEGraphMol();
        while (OEChem.OEReadMolecule(ifsindex, mol)) {
            int status = mcsdb.AddMol(mol, record);
            if (status != record) {
                ++unindexed;
                if (verbose)
                    OEChem.OEThrow.Info("Input structure not added to index, record=" +␣
↪record + " status=" + OEMedChem.OEMatchedPairIndexStatusName(status));
            }
            ++record;

            if (maxrec != 0 && record >= maxrec)
                break;

            if (statusrec != 0 && (record % statusrec) == 0)
                OEChem.OEThrow.Info("Records: " + record + " Indexed: " + (record -␣
↪unindexed) + " Unindexed: " + unindexed);
        }

        if (record == 0)
            OEChem.OEThrow.Fatal("No records in index structure file");

        if (mcsdb.NumFragments() == 0)
            OEChem.OEThrow.Fatal("No fragments found from indexing, use -fragGe,-fragLe␣
↪options to extend indexing range");

        if (!OEMedChem.OEWriteMCSFragDatabase(mcsindexfile, mcsdb))
            OEChem.OEThrow.Fatal("Error serializing MCS fragment database: " +␣
↪mcsindexfile);

        // return some status information
        OEChem.OEThrow.Info("Records: " + record + ", Indexed: " + mcsdb.NumMols() + ",␣
```

(continues on next page)

(continued from previous page)

```
↪fragments: " + mcsdb.NumFragments());
    }

    public static void Main(String[] args)
    {
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, InterfaceData);
        OEMedChem.OEConfigureMCSFragDatabaseOptions(itf);

        if (OEChem.OEParseCommandLine (itf, args, "CreateMCSFragDatabase"))
            MCSFragIndex(itf);
    }

    private static String InterfaceData = @"
!BRIEF [ -input ] index_mols [ -output ] output_mcsindex
#createmcsfragdatabase interface file
!CATEGORY CreateMCSFragDatabase
    !CATEGORY I/O
        !PARAMETER -input 1
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of structure(s) to index
          !KEYLESS 1
        !END
        !PARAMETER -output 2
          !TYPE string
          !REQUIRED true
          !BRIEF Output filename of MCS fragment serialized index
          !KEYLESS 2
        !END
    !END
    !CATEGORY options
        !PARAMETER -verbose 1
           !TYPE bool
           !DEFAULT 0
           !BRIEF generate verbose output
        !END
        !PARAMETER -maxrec 2
           !TYPE int
           !DEFAULT 0
           !BRIEF limit indexing to -maxrec records from the -input structures
        !END
        !PARAMETER -status 3
           !TYPE int
           !DEFAULT 0
           !BRIEF print indexing status every -status records
        !END
    !END
!END";
}
```

3.1.10 MCS Similarity score reporting

A program that loads a previously generated MCS fragment database, then returns the highest Maximum Common Substructure (Tanimoto) scores from the provided input query structures.

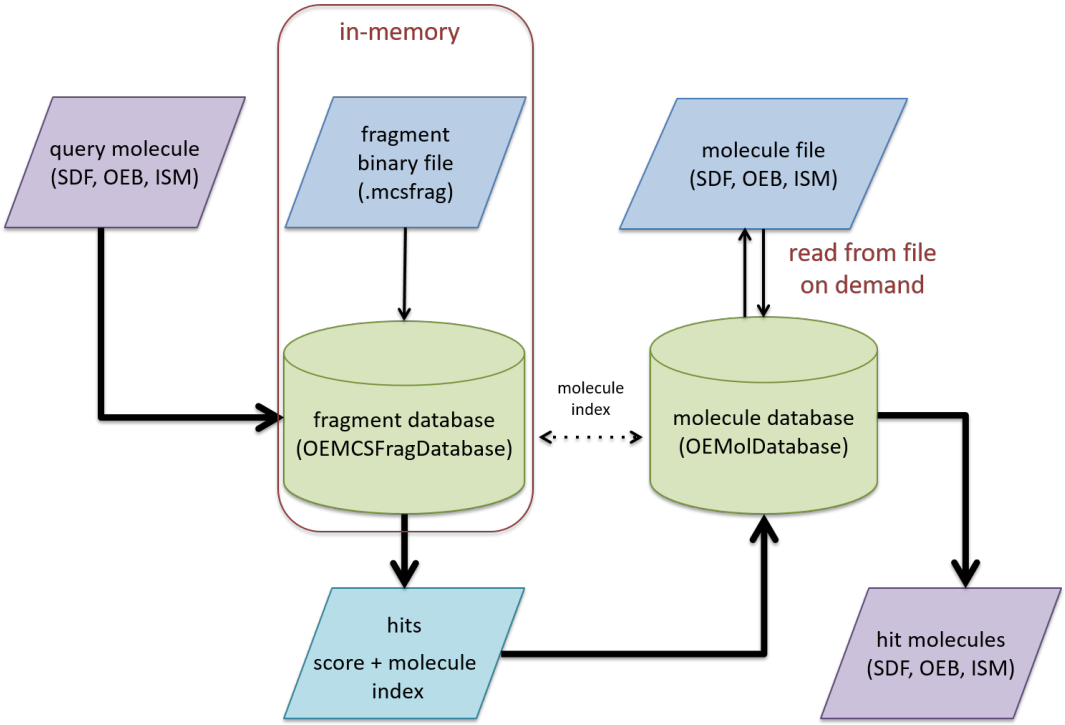


Fig. 5: **Schematic representation of the MCS fragment index and search process**

See also

- *OEMCSFragDatabase* class
- *OEMCSFragDatabaseOptions* class

Command Line Interface

```
prompt> MCSFragDatabase -indb index.mcsfrag -query queries.sdf
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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
```

(continues on next page)

(continued from previous page)

```
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.  OPENEYE 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.
*/
//**************************************************************************
//* Utility to load a pregenerated index and return the highest MCS similarity
//*  structures from an query structure(s)
//* ------------------------------------------------------------------------
//* MCSFragDatabase [ -indb index.mcsfrag ]
//*                 [ -query query_file ]
//*                 [ -type bond -limit 10 ]    // 10 Tanimoto bond similarity scores
//*                 [ -verbose 1 ]              // optional verbosity
//*                 [ -outdb outindex.mcsfrag ] // optional save index file
//*
//* index.mcsfrag: filename of index file to load
//* query_file: filename of query to report MCS similarity results
//**************************************************************************
using OpenEye.OEChem;
using OpenEye.OEMedChem;
using System;

public class MCSFragDatabase {

    private static void MCSFragQuery(OEInterface itf)
    {
        bool verbose = itf.GetBool("-verbose");

        String indb = itf.GetString("-indb");
        if (!OEMedChem.OEIsMCSFragDatabaseFileType(indb))
            OEChem.OEThrow.Fatal("Option -indb must specify a .mcsfrag filename: " +␣
↪indb);
        if (verbose) {
            OEChem.OEThrow.Info("Loading index from " + indb);
        }

        OEMCSFragDatabase mcsdb = new OEMCSFragDatabase();
        if (!OEMedChem.OEReadMCSFragDatabase(indb, mcsdb))
            OEChem.OEThrow.Fatal("Error deserializing MCS fragment database: " + indb);

        if (mcsdb.NumMols() == 0)
            OEChem.OEThrow.Fatal("Loaded empty index");

        if (mcsdb.NumFragments() == 0) {
            OEChem.OEThrow.Fatal("No fragments loaded from index, " +
                                 "use -fragGe,-fragLe options to " +
                                 "extend indexable range");
        }
```

(continues on next page)

(continued from previous page)

```
        oemolistream ifsquery = new oemolistream();
        String queryfile = itf.GetString("-query");
        if (!ifsquery.open(queryfile))
            OEChem.OEThrow.Fatal("Unable to open query file: " + queryfile);

        // process the query options
        int qmaxrec = 1;
        if (itf.HasInt("-qlim"))
            qmaxrec = itf.GetInt("-qlim");

        uint numscores = 10;
        if (itf.HasInt("-limit"))
            numscores = (uint)itf.GetInt("-limit");

        uint cnttype = OEMCSScoreType.BondCount;
        if (itf.HasString("-type"))
        {
            if (itf.GetString("-type").StartsWith("atom"))
                cnttype = OEMCSScoreType.AtomCount;
            else if (itf.GetString("-type").StartsWith("bond"))
                cnttype = OEMCSScoreType.BondCount;
            else
                OEChem.OEThrow.Warning("Ignoring unrecognized -type option, expecting␣
↪atom or bond: " + itf.GetString("-type"));
        }

        // process the queries
        int qnum = 0;
        foreach (OEGraphMol qmol in ifsquery.GetOEGraphMols())
        {
            ++qnum;

            int numhits = 0;
            foreach (OEMCSMolSimScore score in mcsdb.GetSortedScores(qmol, numscores, 0,␣
↪0, true, cnttype))
            {
                if (numhits++ == 0)
                    Console.WriteLine("Query: {0}: {1}", qnum, qmol.GetTitle());

                Console.WriteLine("\trecord: {0,6}\ttanimoto_{1}_score: {2}\tmcs_core:
↪{3}",
                                  score.GetIdx(),
                                  ((cnttype == OEMCSScoreType.BondCount) ? "bond" : "atom
↪" ),
                                  string.Format("{0:0.00}", score.GetScore()),
                                  score.GetMCSCore());
            }
            if (numhits == 0)
                OEChem.OEThrow.Warning("Query: " + qnum + ": " + qmol.GetTitle() + " -␣
↪no hits");
```

(continues on next page)

(continued from previous page)

```
            if (qmaxrec > 0 && qnum >= qmaxrec)
                break;
        }
        ifsquery.close();

        if (qnum == 0)
            OEChem.OEThrow.Fatal("Error reading query structure(s): " + itf.GetString("-
↪query"));
    }

    public static void Main (String[] args)
    {
        OEInterface itf = new OEInterface ();
        OEChem.OEConfigure(itf, interfaceData);

        if (OEChem.OEParseCommandLine (itf, args, "MCSFragDatabase")) {
            MCSFragQuery(itf);
        }
    }

    private static String interfaceData = @"
#MCSFragmentDatabase interface file
!CATEGORY MCSFragmentDatabase
    !CATEGORY I/O
        !PARAMETER -indb 1
          !TYPE string
          !REQUIRED true
          !BRIEF Input filename of index file to load
          !KEYLESS 1
        !END
        !PARAMETER -query 2
          !ALIAS -q
          !TYPE string
          !REQUIRED true
          !BRIEF query structure file to report similarity results against
          !KEYLESS 2
        !END
    !END
    !CATEGORY options
        !PARAMETER -type 1
           !TYPE string
           !DEFAULT bond
           !BRIEF MCS core counts to use for reported scores: atom or bond
        !END
        !PARAMETER -limit 2
           !TYPE int
           !DEFAULT 10
           !BRIEF report -limit scores for the query structure
        !END
        !PARAMETER -verbose 3
           !TYPE bool
           !DEFAULT 0
```

(continues on next page)

(continued from previous page)

```
            !BRIEF generate verbose output
         !END
         !PARAMETER -qlim 4
            !TYPE int
            !DEFAULT 0
            !BRIEF limit query processing to -qlim records from the -query structures
         !END
      !END
!END";
}
```

3.1.11 MCS Fragmentation API

A program that loads a previously generated MCS fragment database, then reports the most common fragments identified by fragmentation from the input structure(s).

See also

- *OEMCSFragDatabase* class
- *OEMCSFragDatabase.MoleculeToCores*
- *OEMCSFragDatabase.CoreToMolecules*
- *OEMCSFragDatabase.CoreToMoleculeCount*
- *OEMCSFragDatabaseOptions* class

Command Line Interface

```
prompt> MCSFragOccurrence -indb index.mcsfrag -query queries.sdf
```

Code

3.1.12 Print OEMedChem version information

A utility to print library version information

See also

- *OEMedChemGetRelease* function
- *OEMedChemGetVersion* function
- *OEMedChemGetArch* function
- *OEMedChemGetPlatform* function

Command Line Interface

This simple example reports library version information

```
prompt> OEMedChemInfo
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
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.  OPENEYE 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.
*/
/*****************************************************************************
 * Print toolkit release date and build information.
 *****************************************************************************/
using System;
using OpenEye.OEMedChem;

public class OEMedChemInfo
{
    public static void Main()
    {
        Console.WriteLine("OEMedChem version: {0} built: {1} arch: {2} platform: {3}",
                          OEMedChem.OEMedChemGetRelease(),
                          OEMedChem.OEMedChemGetVersion(),
                          OEMedChem.OEMedChemGetArch(),
                          OEMedChem.OEMedChemGetPlatform());
    }
}
```

CHAPTER

FOUR

API

4.1 OEMedChem API

4.1.1 OEMedChem Classes

OEBemisMurckoOptions

```
class OEBemisMurckoOptions : public OESystem::OEOptions
```

The functions *OEGetRingLinkerSideChainFragments* and *OEGetBemisMurcko* both return sets of Bemis-Murcko fragments. The *OEBemisMurckoOptions* class can be used to specify both the *OERegionType* of these fragments and to add sidechains that match user-specified criteria to the main Bemis-Murcko framework.

See also

- *OEGetBemisMurcko* function
- *OEGetRingLinkerSideChainFragments* function

Constructors

```
OEBemisMurckoOptions()
```

Default constructor, which sets the perceived fragment type to All, as defined in the *OERegionType* namespace. No additional SMARTS substituents are set.

CreateCopy

```
OEBemisMurckoOptions* CreateCopy() const
```

Returns a pointer to a copy of the options object.

GetRegionType

```
std::string GetRegionType() const
```

Returns the name of the *OERegionType* set in the options class.

GetSubstituentSearch

```
OEChem::OESubSearch GetSubstituentSearch() const
```

Returns the subsearch used to add custom substituents to the Bemis-Murcko framework.

GetUnsaturatedHeteroBonds

```
bool GetUnsaturatedHeteroBonds() const
```

Gets the boolean indicating whether sidechains beginning with unsaturated bonds to hetero atoms will be included in the main Bemis-Murcko framework.

SetRegionType

```
bool SetRegionType(std::string regionType)
```

Sets the name of the *OERegionType* for fragments perceived.

SetSubstituentSearch

```
bool SetSubstituentSearch(const OEChem::OESubSearch value)
```

Sets an OESubSearch for custom substituents to be added to the Bemis-Murcko framework.

SetUnsaturatedHeteroBonds

```
bool SetUnsaturatedHeteroBonds(bool includeUnsaturatedHeteroBonds)
```

Sets whether sidechains beginning with unsaturated bonds to hetero atoms will be included in the main Bemis-Murcko framework. To be included in the main framework when this option is true, the beginning atom of the sidechain must satisfy the OEIsHetero predicate and the bond from the framework to this sidechain must have order greater than one.

Validate

```
bool Validate() const
```

Returns true is the current settings in the options class are valid.

OECreateMMPIndexOptions

```
class OECreateMMPIndexOptions
```

Encapsulates the properties that control a matched pair indexing analysis.

See also

- *OEMatchedPairAnalyzer* class
- *OEMatchedPairAnalyzerOptions* class

Constructors

```
OECreateMMPIndexOptions()
OECreateMMPIndexOptions(const OEMatchedPairAnalyzerOptions& mmpopts)
```

Constructors that initialize an *OECreateMMPIndexOptions* object. The default constructor provides the following properties:

Table 1: **Default parameters of OEMatchedPairAnalyzerOptions**

Property	Default value
processor type	*OEMatchedPairProcessor.Default*
options	*OEMatchedPairOptions.Default*
bond fragmentation limit	20
heavy atom filter	100
indexable fragment filter based on min,max heavy atom percentage of the input structure	85-100%

GetMaxRecord

```
unsigned int GetMaxRecord() const
```

Return the maximum number of records to index from the input structure file or `0` to indicate all structures should be indexed.

GetMMPOptions

```
const OEMatchedPairAnalyzerOptions& GetMMPOptions() const
```

Returns the options controlling the Matched Pair indexing information.

GetNumThreads

```
unsigned int GetNumThreads() const
```

Returns the number of threads to use during the indexing phase. If the number of threads specified is `0`, `OEGetNumProcessors` will be used to interrogate the maximum threads allowed.

GetVerbose

```
unsigned int GetVerbose() const
```

Return the level of output verbosity during the indexing activity.

SetMaxRecord

```
void SetMaxRecord(unsigned int maxrec)
```

Sets the maximum number of records to index from the input structure file. `0` indicates all structures should be indexed.

SetNumThreads

```
void SetNumThreads(unsigned int nthreads)
```

Sets the maximum number of threads to use during the indexing phase. If `nthreads` is set to `0`, `OEGetNumProcessors` will be used to interrogate the maximum threads allowed.

SetTracer

```
void SetTracer(const OESystem::OEDots& dots)
```

Sets a tracer that can be used to show the progress of indexing.

See also

- OEDots class in the **OEChem TK** manual

SetVerbose

```
void SetVerbose(unsigned int verbose)
```

Sets the level of output verbosity during the indexing activity. `0 (off)`, `1 (verbose)`, or `2 (very verbose)` are sufficient values for output verbosity.

OECreateMMPIndexStatus

```
class OECreateMMPIndexStatus
```

Encapsulates the status information returned from a matched pair indexing analysis.

See also

- *OECreateMMPIndexFile* function

Constructors

```
OECreateMMPIndexStatus(bool valid = true)
OECreateMMPIndexStatus(const OECreateMMPIndexStatus& rhs)
```

AddIndexedMol

```
unsigned int AddIndexedMol()
```

Increment the internal count of indexed molecule records by one.

AddUnindexedMol

```
unsigned int AddUnindexedMol()
```

Increment the internal count of unindexed molecule records by one.

GetNumMatchedPairs

```
unsigned int GetNumMatchedPairs() const
```

Return the number of Matched Pairs generated from indexing.

GetNumMols

```
unsigned int GetNumMols() const
```

Return the internal count of indexed molecule records.

GetTotalMols

```
unsigned int GetTotalMols() const
```

Return the internal count of indexed plus unindexed molecule, (i.e., total) records.

GetUnindexedMols

```
unsigned int GetUnindexedMols() const
```

Return the internal count of unindexed molecule records.

SetNumMatchedPairs

```
void SetNumMatchedPairs(unsigned int mmps)
```

Sets the internal count for the number of Matched Pairs generated from indexing.

SetNumMols

```
void SetNumMols(unsigned int indexed)
```

Sets the internal count of indexed molecule records to the specified value.

SetUnindexedMols

```
void SetUnindexedMols(unsigned int unindexed)
```

Sets the internal count of unindexed molecule records to the specified value.

SetValid

```
void SetValid(bool valid)
```

Sets the boolean validity of this status class.

OEMCSFragDatabase

```
class OEMCSFragDatabase
```

Performs a fragmentation indexing on an input set of structures to enable MCS-based similarity searching for compounds with common cores.

See also

- *OEMCSFragDatabaseOptions* class

Constructors

```
OEMCSFragDatabase(const OEMCSFragDatabaseOptions& opts = OEMCSFragDatabaseOptions())
```

Default constructor that initializes the fragment database with the default options defined by *OEMCSFragDatabaseOptions*.

```
OEMCSFragDatabase(const OEMCSFragDatabaseOptions& opts = OEMCSFragDatabaseOptions())
```

Constructor that initializes the fragment database with the options defined by the *OEMCSFragDatabaseOptions* argument.

AddConstMol

```
int AddConstMol(const OEChem::OEMolBase& inmol, int recordID = (-1))
```

Adds the molecule to the index and returns the 0-based recordID if the structure was successfully added, or a negative index if the structure was unable to be indexed. Negative values can be used to retrieve status information for indexing failures. The optional user-defined recordID is provided to allow indexed fragments to be referenced to externally maintained data structures. If the provided recordID is less than zero, an autogenerated index is returned which is an index greater than all ids seen so far.

See also

- *OEMatchedPairIndexStatus*
- *OEMatchedPairIndexStatusName*

AddMol

```
int AddMol(OEChem::OEMolBase& inmol, int recordID = (-1))
```

A high performance version of *OEMCSFragDatabase.AddConstMol* that eliminates molecule copy activities and modifies the passed molecule directly. The state of the returned structure after indexing activities is *undefined* and should generally be discarded or reinstantiated from the original source for any subsequent activities.

ClearIndexedMols

```
void ClearIndexedMols()
```

This method discards the current index contents and frees internal memory in preparation for subsequent reindexing.

CoreToMolecules

```
OESystem::OEIterBase<unsigned int>*CoreToMolecules(const std::string& core) const
```

Given a fragmentation core, return the molecule ids that have this core from the index.

CoreToMoleculeCount

```
unsigned int CoreToMoleculeCount(const std::string& core) const
```

Given a fragmentation core, return a count of the number of molecules that have this core in the index.

GetMaxCoreMolecule

```
std::string GetMaxCoreMolecule(const char delim = '.') const
```

Return the most common core fragment(s) recorded in the index. If there are multiple cores with the same occurrence count, the specified `delim` character is used to delimit the strings.

GetMaxCoreMoleculeCount

```
unsigned int GetMaxCoreMoleculeCount() const
```

Return the maximum core occurrence count recorded in the index.

GetMaxMolIdx

```
unsigned int GetMaxMolIdx() const
```

Returns the highest molecule id currently present in the index.

GetOptions

```
const OEMCSFragDatabaseOptions& GetOptions() const
```

Returns the currently active options for the instance. Options cannot be changed, and require the desired options to be provided when the index is instantiated.

GetScores

```
OESystem::OEIterBase<OEMedChem::OEMCSMolSimScore>* GetScores(const OEChem::OEMolBase&␣
↪query,
                                                             unsigned int bgn = 0,
                                                             unsigned int end = 0,
                                                             unsigned int scorecounts =␣
↪OEMCSScoreType::Default,
                                                             const OEMCSSimFuncBase&␣
↪scorefunc = OEMCSTanimotoSim()) const
```

Performs similarity calculations between a query molecule and the fragment cores stored in the *OEMCSFragDatabase* object. It returns an iterator over the calculated similarity scores (*OEMCSMolSimScore*). Each *OEMCSMolSimScore* object holds a similarity score, an index of database molecule and the fragment core in common between the hit and the query.

bgn, end
: The `bgn` and `end` arguments define the segment of the database on which the similarity calculation will take place. If both of these parameters are omitted (or set to zero), then the similarity calculation is performed on the entire fragment database.

scorecounts
: The value that defines the type of similarity scores (atom or bond) returned by the *OEMCSFragDatabase.GetScores* method. The default type is *OEMCSScoreType.AtomCount*.

scorefunc
: A class that computes the similarity score. By default this is *OEMCSTanimotoSim* but an implementation of *OEMCSTverskySim* is also provided. By providing a class derived from *OEMCSSimFuncBase* user-defined similarity measures other than the provided versions can be generated.

Note

- By default, the *OEMCSFragDatabase.GetScores* method calculates `Tanimoto` atom similarity scores.

GetSortedScores

```
OESystem::OEIterBase<OEMedChem::OEMCSMolSimScore>* GetSortedScores(const␣
↪OEChem::OEMolBase& query,
                                                                   unsigned int limit =␣
↪0,
                                                                   unsigned int bgn = 0,
                                                                   unsigned int end = 0,
                                                                   bool descending =␣
↪true,
                                                                   unsigned int␣
↪scorecounts = OEMCSScoreType::Default,
                                                                   const␣
↪OEMCSSimFuncBase& scorefunc = OEMCSTanimotoSim()) const
```

Performs similarity calculations between a query molecule and the fragment cores stored in the *OEMCSFragDatabase* object. It returns an iterator over the calculated similarity scores (*OEMCSMolSimScore*) in sorted order. Each *OEMCSMolSimScore* object holds a similarity score an index of database molecule and the fragment core in common between the hit and query.

limit
The value that defines the number of similarity scores returned by the *OEMCSFragDatabase.GetSortedScores* method. If it is omitted (or set to zero) then all of the similarity scores are returned.

bgn, end
The `bgn` and `end` arguments define the segment of the database on which the similarity calculation will take place. If both of these parameters are omitted (or set to zero), then the similarity calculation is performed on the entire fragment database.

descending
A boolean value that indicates the direction of the sort values where `true` requests a descending sort and `false` requests ascending.

scorecounts
The value that defines the type of similarity scores (atom or bond) returned by the *OEMCSFragDatabase.GetSortedScores* method. The default type is *OEMCSScoreType.AtomCount*.

scorefunc
A class that computes the similarity score. By default this is *OEMCSTanimotoSim* but an implementation of *OEMCSTverskySim* is also provided. By providing a class derived from *OEMCSSimFuncBase* user-defined similarity measures other than the provided versions can be used.

Note

- By default, the *OEMCSFragDatabase.GetSortedScores* method calculates `Tanimoto` atom similarity scores and returns the highest scores first.

IsIndexed

```
bool IsIndexed(unsigned int recordID) const
```

Returns whether this record id is in the index.

MoleculeToCores

```
OESystem::OEIterBase<const std::string>*MoleculeToCores(const OEChem::OEMolBase& mol,
                                                         bool permuteFragments = true)
```

Given a molecule, return the fragmentation cores using either the provided fragmentation options, or using the fragmentation options from the *OEMCSFragDatabase* instance. If the `permuteFragments` argument is `true`, all combinations of the generated fragmentation cores are generated, otherwise a unique set of multi-fragment cores is returned representing all combinations of bond fragmentations between the min and max cut limits.

Shown below are versions for two type of examples - one that uses an *OEMCSFragDatabase* instance so the database index options are used to control the fragmentation behavior and the other uses the free function and custom options.

```
            // create an MCS fragment database with defaults
            OEMCSFragDatabase fragdb = new OEMCSFragDatabase();
            // use the options from the frag database to fragment an arbitrary input␣
↪molecule
            Console.WriteLine("MoleculeToCores using default fragment database options:
↪");
            List<string> cores = new List<string>();
            foreach (String c in fragdb.MoleculeToCores(mol))
```

(continues on next page)

(continued from previous page)

```
            cores.Add(c);
        cores.Sort();
        int corenum = 0;
        foreach (String core in cores)
        {
            Console.WriteLine(corenum + ": " + core);
            ++corenum;
        }
```

```
        // set the MCS fragment database options from the command-line arguments
        OEMCSFragDatabaseOptions fragopts = new OEMCSFragDatabaseOptions();
        if (!OEMedChem.OEConfigureMCSFragDatabaseOptions(itf))
            OEChem.OEThrow.Fatal("Error configuring options");
        if (!OEMedChem.OESetupMCSFragDatabaseOptions(fragopts, itf)) {
            OEChem.OEThrow.Fatal("Error setting options");
        }

        // use the custom options to fragment an arbitrary input molecule
        Console.WriteLine("MoleculeToCores using command-line options:");
        List<string> cores = new List<string>();
        foreach (String c in OEMedChem.OEMoleculeToCores(mol, fragopts))
            cores.Add(c);
        cores.Sort();
        int corenum = 0;
        foreach (String core in cores)
        {
            Console.WriteLine(corenum + ": " + core);
            ++corenum;
        }
```

See also

- *OEMCSFragDatabaseOptions* class
- *OEMoleculeToCores* function

NumFragments

```
unsigned int NumFragments() const
```

Returns the total number of fragment cores present in the index.

NumMols

```
unsigned int NumMols() const
```

Returns the total number of indexed molecules present in the index.

ProcessMol

```
int ProcessMol(OEChem::OEMolBase& inmol)
```

Since the fragmentation engine used internally to generate the index *modifies* the input structures (eg discarding all but the largest fragment of the input), this method is provided to force the application of the indexing modifications on the input structure. This is generally useful for depiction or reporting of the returned similarity results.

OEMCSFragDatabaseOptions

```
class OEMCSFragDatabaseOptions
```

Encapsulates the properties that control MCS fragmentation indexing.

See also

- *OEMCSFragDatabase* class

Constructors

```
OEMCSFragDatabaseOptions()
```

Default constructor that initializes an *OEMCSFragDatabaseOptions* object with the following properties:

Table 2: **Default parameters of OEMCSFragDatabaseOptions**

Property	Default value
options	*OEMatchedPairOptions.DefaultFragIndex*
maximum allowed fragmentation bonds for indexed molecules	20
maximum allowed heavy atoms for indexed molecules	100
indexable fragment filter based on min,max heavy atom percentage of the input structure	85-100%

```
OEMCSFragDatabaseOptions(const OEMCSFragDatabaseOptions& options)
```

Copy constructor.

ClearIndexableFragmentRange

```
void ClearIndexableFragmentRange()
```

Clears the min-max range, which removes the fragment range filter and results in indexing fragments of all sizes.

GetFragmentationLimit

```
unsigned int GetFragmentationLimit() const
```

Return the bond fragmentation limit value.

GetIndexableFragmentRangeMax

```
float GetIndexableFragmentRangeMax() const
```

Return the maximum range of the indexable fragment filter.

GetIndexableFragmentRangeMin

```
float GetIndexableFragmentRangeMin() const
```

Return the minimum range of the indexable fragment filter.

GetMaxAtomFilter

```
unsigned int GetMaxAtomFilter() const
```

Return the maximum heavy atom filter value. Input structures with more than this number of heavy atoms will be rejected.

GetMaxFragLimit

```
unsigned int GetMaxFragLimit() const
```

Return the maximum number of fragmentation bonds that are allowed to be cut. Structures with more than this limit are not fragmented.

GetMinFragLimit

```
unsigned int GetMinFragLimit() const
```

Return the minimum number of fragmentation bonds that are allowed to be cut. Structures with less than this limit are not fragmented.

GetOptions

```
unsigned int GetOptions() const
```

Return the options setting value. The default is *OEMatchedPairOptions.DefaultFragIndex*.

GetRingFragLimit

```
unsigned int GetRingFragLimit() const
```

Return the maximum number of ring fragmentation bonds that are allowed to be cut. Structures with more possible ring bonds than this limit will not have ring bonds fragmented.

HasIndexableFragmentHeavyAtomRange

```
bool HasIndexableFragmentHeavyAtomRange() const
```

Returns whether a min-max range has been specified.

SetFragLimits

```
bool SetFragLimits(unsigned int minval, unsigned int maxval)
```

Sets the min and max range of fragmentation bond cuts.

> **Warning**
>
> Very high limits for the allowed number of fragmentation bonds will have severe performance implications for indexing due to the combinatorial explosion in the number of fragmentation cores generated.

SetFragmentationLimit

```
bool SetFragmentationLimit(unsigned int limit)
```

Set the bond fragmentation limit. Any structures that generate more fragmentation bonds than this limit are rejected.

SetIndexableFragmentRange

```
bool SetIndexableFragmentRange(float minHeavyAtomPercent,
                               float maxHeavyAtomPercent)
```

Sets the min-max heavy atom percentage range to request which fragment sizes should be indexed. The default is 85-100% of the input structures.

SetMaxAtomFilter

```
bool SetMaxAtomFilter(unsigned int numAtoms)
```

Sets the maximum heavy atom filter value. Input structures with more than this number of heavy atoms will be rejected for indexing.

SetOptions

```
bool SetOptions(unsigned int opts)
```

Set the options value. The default is *OEMatchedPairOptions.DefaultFragIndex*.

SetRingFragLimit

```
bool SetRingFragLimit(unsigned int maxringcuts)
```

Sets maximum number of ring cuts allowed. Structures with more possible ring bonds than this limit will not have ring bonds fragmented.

> **Warning**
>
> Very high limits for the allowed number of ring fragmentation bonds will have severe performance implications for indexing due to the combinatorial explosion in the number of fragmentation cores generated.

OEMCSMolSimScore

```
class OEMCSMolSimScore
```

This class represents *OEMCSMolSimScore* that holds a similarity value, an index and a fragment core that identifies the molecule in the *OEMCSFragDatabase* index from which the score is calculated.

> **See also**
>
> - *OEMCSFragDatabase.GetScores*
> - *OEMCSFragDatabase.GetSortedScores*

Constructors

```
OEMCSMolSimScore()
OEMCSMolSimScore(const OEMCSMolSimScore& ref)
OEMCSMolSimScore(unsigned int i, double s,
                 const std::string& coresmi)
```

Initializes an *OEMCSMolSimScore* from an index `i` and a score `s` and a core `coresmi`, or from another *OEMCSMolSimScore* instance, or with default (null) values.

GetIdx

```
unsigned int GetIdx() const
```

Returns the index of the indexed molecule corresponding to the similarity score.

GetMCSCore

```
std::string GetMCSCore() const
```

Returns the SMILES core relating the query and the indexed molecule corresponding to the similarity score.

GetScore

```
double GetScore() const
```

Returns the score that relates the query and the indexed molecule.

Set

```
void Set(unsigned int i, double s, const std::string& coresmi)
```

Initializes an *OEMCSMolSimScore* from an index `i` and a score `s` and a core `coresmi`.

OEMCSSimFuncBase

```
class OEMCSSimFuncBase
```

OEMCSSimFuncBase is an abstract base class which defines the interface necessary to perform similarity calculations that populate *OEMCSMolSimScore* objects.

The following classes derive from this class:

- *OEMCSTanimotoSim*
- *OEMCSTverskySim*

See also

- *OEMCSFragDatabase* class
- *OEMCSFragDatabase.GetScores* class
- *OEMCSFragDatabase.GetSortedScores* class

Constructors

```
OEMCSSimFuncBase()
```

Default constructor.

operator()

```
double operator() (unsigned int commonAB, unsigned int totalA, unsigned int totalB) const
```

Virtual *const* operator that takes three count values and computes the similarity value. Depending on the method, the counts provided may be either the atom or bond counts for the MCS core, query molecule and indexed molecule.

CreateCopy

```
OEMCSSimFuncBase* CreateCopy() const
```

OEMCSSimFuncBase.CreateCopy is a virtual constructor which allows copying of concrete derived objects using a reference to this base class.

GetMCSSimTypeString

```
std::string GetMCSSimTypeString() const
```

Returns a string representation of the concrete derived class.

OEMCSTanimotoSim

```
class OEMCSTanimotoSim : public OEMedChem::OEMCSSimFuncBase
```

This class represents *OEMCSTanimotoSim* that calculates the `Tanimoto` similarity score.

Formula:

$$Sim_{Tanimoto}(A,B) = \frac{bothAB}{|A|+|B|-bothAB} = \frac{bothAB}{onlyA+onlyB+bothAB}$$

The following methods are publicly inherited from *OEMCSSimFuncBase*:

operator()	*CreateCopy*	*GetMCSSimTypeString*

Constructors

```
OEMCSTanimotoSim()
```

Default constructor.

operator()

```
double operator() (unsigned int commonAB, unsigned int totalA,  unsigned totalB) const
```

Returns the `Tanimoto` similarity coefficient based on the fragment core counts for the fragment core, query molecule and indexed molecule.

CreateCopy

```
OEMedChem::OEMCSSimFuncBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEMCSTanimotoSim* object is dynamically allocated and owned by the caller.

GetMCSSimTypeString

```
std::string GetMCSSimTypeString() const
```

Returns a string representation of the *OEMCSTanimotoSim* class.

OEMCSTverskySim

```
class OEMCSTverskySim : public OEMedChem::OEMCSSimFuncBase
```

This class represents *OEMCSTverskySim* that calculates the `Tversky` similarity score.

Formula:

$$Sim_{Tversky}(A,B) = \frac{bothAB}{\alpha*onlyA+\beta*onlyB+bothAB}$$

The following methods are publicly inherited from *OEMCSSimFuncBase*:

operator()	*CreateCopy*	*GetMCSSimTypeString*

Constructors

```
OEMCSTverskySim()
OEMCSTverskySim(const OEMCSTverskySim& ref)
```

Default constructor ($\alpha = 0.95$, $\beta = 0.05$)

```
OEMCSTverskySim(double tvAlpha = 0.95, double tvBeta = 0.05)
```

Constructor with arbitrary α and β parameters.

operator()

```
double operator() (unsigned int commonAB, unsigned int totalA,  unsigned totalB) const
```

Returns the `Tversky` similarity coefficient based on the fragment core counts for the fragment core, query molecule and indexed molecule.

CreateCopy

```
OEMedChem::OEMCSSimFuncBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEMCSTverskySim* object is dynamically allocated and owned by the caller.

GetAlpha

```
double GetAlpha() const
```

Returns the α parameter from the *OEMCSTverskySim* class.

GetBeta

```
double GetBeta() const
```

Returns the β parameter from the *OEMCSTverskySim* class.

GetMCSSimTypeString

```
std::string GetMCSSimTypeString() const
```

Returns a string representation of the *OEMCSTverskySim* class.

OEMatchedPairAnalyzer

```
class OEMatchedPairAnalyzer
```

Performs a matched pair indexing analysis on an input set of structures.

See also

- *OEMatchedPairAnalyzerOptions* class

Constructors

```
OEMatchedPairAnalyzer(const OEMatchedPairAnalyzerOptions& opts =␣
↪OEMatchedPairAnalyzerOptions())
```

Default constructor that initializes the analysis engine with the default options defined by *OEMatchedPairAnalyzerOptions*.

```
OEMatchedPairAnalyzer(const OEMatchedPairAnalyzerOptions& opts =␣
↪OEMatchedPairAnalyzerOptions())
```

Constructor that initializes the analysis engine with the options defined by the *OEMatchedPairAnalyzerOptions* argument.

AddMol

```
int AddMol(const OEChem::OEMolBase& inmol, int recordID = (-1))
```

Add the molecule to the index and return the 0-based recordID if the structure was successfully added, or a negative index if the structure was unable to be indexed. Negative values can be used to retrieve status information detail for indexing failures. The optional user-defined recordID is provided to allow indexed matched pairs to be referenced to externally maintained data structures. If the provided recordID is less than zero, an autogenerated index is returned which is an index greater than all ids seen so far.

See also

- *OEMatchedPairIndexStatus*
- *OEMatchedPairIndexStatusName*

GetMaxMolIdx

```
unsigned int GetMaxMolIdx() const
```

Returns one more than the maximum molecule index this Matched Pair index currently contains.

GetMolecule

```
bool GetMolecule(OEChem::OEMolBase& mol, unsigned int molID,
                 bool includeData = true) const
```

Returns the molecule referenced by `molID` in the index. An optional *includeData* argument controls whether generic data present on the molecule at index time is also returned.

GetOptions

```
const OEMatchedPairAnalyzerOptions& GetOptions() const
```

See also

- *OEMatchedPairAnalyzerOptions* class

GetProcessorID

```
const char* GetProcessorID() const
```

Returns a character string name of the processor type.

See also

- *OEMatchedPairProcessorName* class

GetProcessorType

```
unsigned int GetProcessorType() const
```

Returns a constant representing the processor type in use.

See also

- *OEMatchedPairProcessor*

IsIndexed

```
bool IsIndexed(unsigned int recordID) const
```

Returns whether the specified record id has been indexed.

NumIndexNodes

```
unsigned int NumIndexNodes() const
```

Return a count of the number of index nodes currently present in the index.

NumMatchedPairs

```
unsigned int NumMatchedPairs() const
```

Return a count of the number of matched molecular pairs currently present in the index.

NumMols

```
unsigned int NumMols() const
```

Return a count of the number of molecules that have currently been added to the index.

ModifyOptions

```
bool ModifyOptions(unsigned int addopts, unsigned int delopts)
```

An active analyzer has a limited set of options that can be altered once it has been instantiated as many of the options affect the internal indexing contents. This method can be used to alter any options that do not affect the internal index state.

See also

- *OEMatchedPairOptions*

OEMatchedPairAnalyzerOptions

```
class OEMatchedPairAnalyzerOptions
```

Encapsulates the properties that control a matched pair analysis.

See also

- *OEMatchedPairAnalyzer* class

Constructors

```
OEMatchedPairAnalyzerOptions(unsigned int processorType =␣
↪OEMatchedPairProcessor::Default)
OEMatchedPairAnalyzerOptions(const OEMatchedPairAnalyzerOptions& rhs)
```

Default constructor that initializes an *OEMatchedPairAnalyzerOptions* object with the following properties:

Table 3: **Default parameters of OEMatchedPairAnalyzerOptions**

Property	Default value
processor type	*OEMatchedPairProcessor.Default*
options	*OEMatchedPairOptions.Default*
bond fragmentation limit	20
heavy atom filter	100
indexable fragment filter based on min,max heavy atom percentage of the input structure	85-100%

```
OEMatchedPairAnalyzerOptions(const OEMatchedPairAnalyzerOptions& rhs)
```

Copy constructor.

ClearIndexableFragmentRange

```
void ClearIndexableFragmentRange()
```

Clears the min-max range, which removes the fragment range filter and results in indexing fragments of all sizes.

GetFragmentationLimit

```
unsigned int GetFragmentationLimit() const
```

Return the bond fragmentation limit value.

GetIndexableFragmentRangeMax

```
float GetIndexableFragmentRangeMax() const
```

Return the maximum range of the indexable fragment filter.

GetIndexableFragmentRangeMin

```
float GetIndexableFragmentRangeMin() const
```

Return the minimum range of the indexable fragment filter.

GetMaxAtomFilter

```
unsigned int GetMaxAtomFilter() const
```

Return the maximum heavy atom filter value. Input structures with more than this number of heavy atoms will be rejected.

GetOptions

```
unsigned int GetOptions() const
```

Return the options setting value. The default is *OEMatchedPairOptions.Default*.

GetProcessorType

```
unsigned int GetProcessorType() const
```

Return the processor type. The type using the default constructor is *OEMatchedPairProcessor.Default*

HasIndexableFragmentHeavyAtomRange

```
bool HasIndexableFragmentHeavyAtomRange() const
```

Returns whether a min-max range has been specified.

ModifyOptions

```
bool ModifyOptions(unsigned int addopts, unsigned int delopts)
```

An active analyzer has a limited set of options that can be altered once it has been instantiated as many of the options affect the internal indexing contents. This method can be used to alter any options that do not affect the internal index state.

SetFragmentationLimit

```
bool SetFragmentationLimit(unsigned int limit)
```

Set the bond fragmentation limit. Any structures that generate more fragmentation bonds than this limit are rejected.

SetIndexableFragmentRange

```
bool SetIndexableFragmentRange(float minHeavyAtomPercent,
                               float maxHeavyAtomPercent)
```

Sets the min-max heavy atom percentage range to request which fragment sizes should be indexed. The default is 85-100% of the input structures.

SetMaxAtomFilter

```
bool SetMaxAtomFilter(unsigned int numAtoms)
```

Sets the maximum heavy atom filter value. Input structures with more than this number of heavy atoms will be rejected for indexing.

SetOptions

```
bool SetOptions(unsigned int opts)
```

Set the options value. The default is *OEMatchedPairOptions.Default*.

SetProcessorType

```
bool SetProcessorType(unsigned int processorType = OEMatchedPairProcessor::Default)
```

Sets the index processor type to the requested value.

See also

- *OEMatchedPairProcessor* constants

OEMatchedPair

```
class OEMatchedPair
```

This class identifies a specific matched molecular pair.

Constructors

```
OEMatchedPair(const OEMatchedPair& rhs)
OEMatchedPair(unsigned int fromIndex = 0u,
              unsigned int toIndex = 0u,
              const char* fromSmiles = 0, const char* toSmiles = 0)
```

Constructs an object from the passed arguments.

AddSDData

```
bool AddSDData(const std::string& tag,
               const std::string& fromValue,
               const std::string& toValue)
```

Add SD data information to the matched pair. The tag is the field name of the SD data and the from and to arguments indicate the respective data for the molecular pair.

ClearSDData

```
void ClearSDData()
```

Deletes all SD data recorded for the molecular pair.

CreateCopy

```
OEMatchedPair* CreateCopy() const
```

FromIndex

```
unsigned int FromIndex() const
```

This is the from index of the matched molecular pair.

FromSmiles

```
const std::string& FromSmiles() const
```

This is the from SMILES string of the matched molecular pair.

GetDataTags

```
OESystem::OEIterBase<const std::string>* GetDataTags() const
```

This method returns an iterator of over the SD data field tags recorded for the matched molecular pair.

GetFromSDData

```
std::string GetFromSDData(const std::string& tag) const
```

This method returns SD data with the specified 'tag' for the from structure of the matched molecular pair.

GetSDData

```
bool GetSDData(const std::string &tag, std::string &fromValue,
               std::string &toValue) const
```

This method returns SD data for the specific data field requested for the matched molecular pair. If either of the from or to structures have no data present, an empty string is returned.

GetToSDData

```
std::string GetToSDData(const std::string& tag) const
```

This method returns SD data with the specified 'tag' for the to structure of the matched molecular pair.

HasSDData

```
bool HasSDData(const std::string& tag) const
```

This method returns a boolean indicating whether SD data of field name 'tag' is present for either of the from or to structures.

NumDataTags

```
unsigned int NumDataTags() const
```

This method returns the number of SD data tags present for the molecular pair.

ToIndex

```
unsigned int ToIndex() const
```

This is the to index of the matched molecular pair.

ToSmiles

```
const std::string& ToSmiles() const
```

This is the to SMILES string of the matched molecular pair.

OEMatchedPairTransform

```
class OEMatchedPairTransform
```

This class embodies a unique transformation that relates one or more matched molecular pairs.

Constructors

```
OEMatchedPairTransform()
OEMatchedPairTransform(const OEMatchedPairTransform& rhs)
OEMatchedPairTransform(const std::string& xformString,
                       int context = OEMatchedPairContext::Default)
OEMatchedPairTransform(const std::string& ForwardXForm,
                       const std::string& ReverseXForm,
                       int context = OEMatchedPairContext::Default)
```

Constructs an object from the passed arguments.

AddMatchedPair

```
bool AddMatchedPair(const OEMatchedPair& mmp)
bool AddMatchedPair(unsigned int fromIndex,
                    const std::string& fromSmiles,
                    unsigned int toIndex,
                    const std::string& toSmiles)
```

This method links a specific from and to structure to the transform.

CreateCopy

```
OEMatchedPairTransform* CreateCopy() const
```

GetMatchedPairContext

```
int GetMatchedPairContext() const
```

Returns the chemistry context type for the transformation.

See also

- *OEMatchedPairContext*

GetMatchedPairs

```
OESystem::OEIterBase<OEMatchedPair>* GetMatchedPairs() const
OESystem::OEIterBase<OEMatchedPair>* GetMatchedPairs(unsigned int fromORtoIndex) const
OESystem::OEIterBase<OEMatchedPair>* GetMatchedPairs(unsigned int fromIndex,
                                                     unsigned int toIndex) const
```

Returns an iterator of the all the matched pairs associated with this transformation, or restricts the returned matched pairs that meeting the from or to index constraint.

See also

- *OEMatchedPair*

GetMolecule

```
bool GetMolecule(unsigned int index, std::string& molsmi) const
std::string GetMolecule(unsigned int index) const
```

This method retrieves the molecule of the specified molecule id from the transform.

GetMoleculeTitle

```
bool GetMoleculeTitle(unsigned int molindex, std::string& name) const
std::string GetMoleculeTitle(unsigned int molindex) const
```

This method returns a molecule title for the passed molecule index.

GetTransform

```
const std::string& GetTransform() const
```

Returns the reaction transformation string that can be used to instantiate a OEUniMolecularRxn object.

HasMatchedPair

```
bool HasMatchedPair(unsigned int FromOrToIndex) const
bool HasMatchedPair(unsigned int fromIndex, unsigned int toIndex) const
```

Returns a boolean to indicate if the specific molecule pair is registered with this transformation.

Invert

```
bool Invert()
```

Reverses the direction of the transform and the matched pair information for the transformation.

> **Warning**
>
> If the transform was extracted with SMART query qualifiers for use in molecule transformation activities, the inversion of the transform will result in the query features moved to the product side of the reaction. Inverting transformations is only meaningful when using the *OEMatchedPairTransformExtractMode.NoSMARTS* extraction option.

NumMatchedPairs

```
unsigned int NumMatchedPairs() const
```

Returns a count of the number of matched molecular pairs registered with this transformation.

SetMoleculeTitle

```
bool SetMoleculeTitle(unsigned int index, const std::string& name)
```

This method set the molecule title for the passed molecule index.

SetTransform

```
bool SetTransform(const std::string& xform,
                  int context = OEMatchedPairContext::Default)
bool SetTransform(const std::string& ForwardXForm,
```

(continues on next page)

(continued from previous page)

```
                    const std::string& ReverseXForm,
                    int context = OEMatchedPairContext::Default)
```

Sets the internal reaction transformation string.

OEMatchedPairTransformExtractOptions

```
class OEMatchedPairTransformExtractOptions
```

Encapsulates the properties that control extraction of matched molecular pair transformations.

See also

- *OEMatchedPairAnalyzer* class
- *OEMatchedPairGetTransforms* function

Constructors

```
OEMatchedPairTransformExtractOptions(int context = OEMatchedPairContext::Default,
                                     unsigned int options =␣
↪OEMatchedPairTransformExtractMode::Default)
```

Default constructor that initializes an *OEMatchedPairTransformExtractOptions* object with the following properties:

Table 4: **Default parameters of OEMatchedPairTransformExtractOptions**

Property	Default value
transformation context	*OEMatchedPairContext.Default*
extraction options	*OEMatchedPairTransformExtractMode.Default*
maximum matched pair core structure substituents	200
skip maximum matched pair core index nodes exceeding substituent limit	false

```
OEMatchedPairTransformExtractOptions(const OEMatchedPairTransformExtractOptions& rhs)
```

Copy constructor.

GetAttachmentPoints

```
bool GetAttachmentPoints() const
```

Returns the setting that requests the addition of attachment points on the matched pair transformation.

GetContext

```
int GetContext() const
```

Returns the setting that controls the amount of core chemistry adjacent to the substituent attachment site for the matched pairs.

See also

- *OEMatchedPairContext.Default*

GetDirection

```
unsigned int GetDirection() const
```

Returns the setting that controls the direction for the returned matched pair transformation. One of *OEMatchedPairTransformExtractMode.Forward*, *OEMatchedPairTransformExtractMode.Backward*, (*OEMatchedPairTransformExtractMode.Forward* | *OEMatchedPairTransformExtractMode.Backward*)`` (both) or *OEMatchedPairTransformExtractMode.Sorted* are supported. For *OEMatchedPairTransformExtractMode.Sorted*, the transformation is lexically sorted based on the reactants and products and may need to be inverted to have the transformation represent driving an arbitrary data property associated with the matched pair in a desired direction.

See also

- *OEMatchedPairTransformExtractMode.Default*
- *OEMatchedPairTransform.Invert*

GetIsotopes

```
bool GetIsotopes() const
```

Returns the setting that controls the generalization of the returned transformations for isotopic substitutions.

GetMCSCorrespondence

```
bool GetMCSCorrespondence() const
```

Returns the setting that requests the addition of atom-atom mapping correspondence to the core of the matched pair transformation.

GetMatchedPairData

```
bool GetMatchedPairData() const
```

Returns whether the matched pair data is included in the matched pair transformation. If only the transformation itself is required, there may not be a need for extraction of related matched pair indexed data.

GetMatchedPairs

```
bool GetMatchedPairs() const
```

Returns whether the matched pair data ids are returned for the matched pair transformation. If only the transformation itself is required, there may not be a need for extraction of related matched pair indices.

GetMaxMMPLimit

```
unsigned int GetMaxMMPLimit() const
```

Returns the requested upper limit of matched pairs for the extracted transformations.

GetMaxSubstituentLimit

```
unsigned int GetMaxSubstituentLimit() const
```

Returns the setting for the maximum number of indexed core substituents that should be processed into matched pair transformations.

> **Warning**
>
> Setting this limit to `0` will suppress any core structure truncation or skipping activities, but may result in extremely poor transformation extraction performance owing to the combinatorial explosion of generating all `NxM` combinations of matched pairs.

GetMaxSubstituentLimitSkip

```
bool GetMaxSubstituentLimitSkip() const
```

Returns whether the matched pair core structures in the index that have more than the *OEMatchedPairTransformExtractOptions.GetMaxSubstituentLimit* number of substituents are processed into matched pair transformation entries. If the setting is `true`, cores that exceed the limit are skipped, and if `false`, the substituent list is truncated to the first *OEMatchedPairTransformExtractOptions.GetMaxSubstituentLimit* substituents, if non-zero.

GetMinMMPLimit

```
unsigned int GetMinMMPLimit() const
```

Returns the requested lower limit of matched pairs for the extracted transformations.

GetMinSubstituentLimit

```
unsigned int GetMinSubstituentLimit() const
```

Returns the setting for the minimum number of core structure substituents for index nodes that should be processed into matched pair transformations.

GetOptions

```
unsigned int GetOptions() const
```

Returns the bit option settings for the extraction options.

See also

- *OEMatchedPairTransformExtractMode*

GetSMARTS

```
bool GetSMARTS() const
```

Returns whether the matched pair transformations are annotated with additional SMARTS qualifiers to make the transformations more chemically relevant when used for generating chemical transformations from extracted matched pairs.

SetAttachmentPoints

```
bool SetAttachmentPoints(bool val)
```

Set the option that controls the addition of attachment points on the matched pair transformation.

SetContext

```
bool SetContext(int context)
```

Sets the option that that controls the amount of core chemistry adjacent to the substituent attachment site in then matched pair transformation.

See also

- *OEMatchedPairContext.Default*

SetDirection

```
bool SetDirection(unsigned int direction)
```

Sets the specified transformation direction for the returned matched pair transformation. One of `Forward`, `Backward`, `(Forward | Backward)` (both) or `Sorted` are supported. For `Sorted`, the transformation is lexically sorted based on the reactants and products and may need to be inverted to have the transformation represent driving an arbitrary data property associated with the matched pair in a desired direction.

See also

- *OEMatchedPairTransformExtractMode.Default*
- *OEMatchedPairTransform.Invert*

SetIsotopes

```
bool SetIsotopes(bool require)
```

Sets the option that controls whether the transformation is allowed to be generalized for isotopic substitutions. If `true`, isotopic information will be included in the transformations for the chemical modifications.

SetMCSCorrespondence

```
bool SetMCSCorrespondence(bool add)
```

Sets the option that requests the addition of atom-atom mapping correspondence to the core of the matched pair transformation.

SetMatchedPairData

```
bool SetMatchedPairData(bool enable)
```

Sets the option that requests matched pair data be included in the matched pair transformation. If only the transformation itself is required, there may not be a need for extraction of related matched pair indexed data, so the data extraction can be disabled.

SetMatchedPairs

```
bool SetMatchedPairs(bool enable)
```

Sets the option that requests that matched pair structure ids are returned for the matched pair transformation. If only the transformation itself is required, there may not be a need for extraction of related matched pair structure indices.

SetMaxSubstituentLimitSkip

```
bool SetMaxSubstituentLimitSkip(bool skip)
```

Sets the option that controls whether the matched pair core structures in the index that have more than the *OEMatchedPairTransformExtractOptions.GetMaxSubstituentLimit* number of substituents are processed into matched pair transformation entries. If the setting is `true`, cores that exceed the limit are skipped, and if `false`, the substituent list is truncated to the first *OEMatchedPairTransformExtractOptions.GetMaxSubstituentLimit* substituents, if non-zero.

SetMMPLimit

```
bool SetMMPLimit(unsigned int atLeast, unsigned int atMost = 0u)
```

Sets the requested range limits for the number matched pairs in the extracted transformations.

SetSubstituentLimit

```
bool SetSubstituentLimit(unsigned int atLeast,
                         unsigned int atMost = 0u)
```

Sets the requested range limits for the core structure index nodes to constrain processing to core index nodes matching the specified range.

> **Warning**
>
> Setting the maximum limit to `0` will suppress any core structure truncation or skipping activities, but may result in extremely poor transformation extraction performance owing to the combinatorial explosion of generating all `NxM` combinations of matched pairs and their associated transformations and data.

SetOptions

```
bool SetOptions(unsigned int options)
```

Sets the bit option settings that control the transformation extraction activities.

See also

- *OEMatchedPairTransformExtractMode*

SetSMARTS

```
bool SetSMARTS(bool enable)
```

Sets the option that indicates that matched pair transformations should be annotated with additional SMARTS qualifiers to make the transformations more chemically relevant when used for generating chemical transformations from extracted matched pairs.

4.1.2 OEMedChem Constants

OEMCSFragDatabaseSetup

This namespace encodes symbolic constants used as bit-masks to indicate which MCS fragmentation indexing parameters are being created when invoking the *OEConfigureMCSFragDatabaseOptions* function.

See also

- *OEMCSFragDatabaseOptions* class

All

The combination of following constants:

- *OEMCSFragDatabaseSetup.AllFrags*
- *OEMCSFragDatabaseSetup.BondFragLimit*
- *OEMCSFragDatabaseSetup.FragCutsRange*
- *OEMCSFragDatabaseSetup.FragMinMaxRange*
- *OEMCSFragDatabaseSetup.FragmentationCuts*
- *OEMCSFragDatabaseSetup.HeavyAtomLimit*
- *OEMCSFragDatabaseSetup.IndexHydrogenSites*
- *OEMCSFragDatabaseSetup.RingCuts*

AllFrags

Passing this constant to the *OEConfigureMCSFragDatabaseOptions* function results in generating the following default interface to configure all fragment indexing.

```
Contents of parameter -allfrags
    Aliases : -all
    Type : bool
    Allow list : false
    Default : false
    Simple : true
    Required : false
    Brief : Index all fragment sizes, ignore fragment heavy atom percentage
```

See also

- *OEMCSFragDatabaseOptions.HasIndexableFragmentHeavyAtomRange* method
- *OEMCSFragDatabaseOptions.ClearIndexableFragmentRange* method
- *OESetupMCSFragDatabaseOptions*

BondFragLimit

Passing this constant to the *OEConfigureMCSFragDatabaseOptions* function results in generating the following default interface to configure maximum number of fragmentation bonds allowed.

```
Contents of parameter -bondfraglimit
    Aliases : -blim
    Type : int
    Allow list : false
    Default : 20
    Simple : true
    Required : false
    Brief : Bond fragmentation limit
```

See also

- *OEMCSFragDatabaseOptions.SetFragmentationLimit* method
- *OESetupMCSFragDatabaseOptions*

FragCutsRange

Passing this constant to the *OEConfigureMCSFragDatabaseOptions* function results in generating the following default interface to configure the minimum and maximum fragmentation bond cuts allowed.

```
Contents of parameter -minfragcuts
    Aliases : -mincuts
```

(continues on next page)

(continued from previous page)

```
    Type : int
    Allow list : false
    Default : 1
    Simple : true
    Required : false
    Legal ranges :
        1 to 100
    Brief : Minimum number of fragmentation cuts allowed

Contents of parameter -maxfragcuts
    Aliases : -maxcuts
    Type : int
    Allow list : false
    Default : 3
    Simple : true
    Required : false
    Legal ranges :
        1 to 100
    Brief : Maximum number of fragmentation cuts allowed
```

See also

- *OEMCSFragDatabaseOptions.SetFragLimits* method
- *OESetupMCSFragDatabaseOptions*

FragMinMaxRange

Passing this constant to the *OEConfigureMCSFragDatabaseOptions* function results in generating the following default interface to configure the min and max allowed fragment sizes to be indexed.

```
Contents of parameter -fragGe
    Aliases : -ge
    Type : float
    Allow list : false
    Default : 85.0
    Simple : true
    Required : false
    Legal ranges :
        0.0 to 100.0
    Brief : fragments greater than or equal to this heavy atom percentage

Contents of parameter -fragLe
    Aliases : -le
    Type : float
    Allow list : false
    Default : 100.0
    Simple : true
    Required : false
    Legal ranges :
```

(continues on next page)

(continued from previous page)

```
        0.0 to 100.0
    Brief : fragments less than or equal to this heavy atom percentage
```

See also

- *OEMCSFragDatabaseOptions.SetIndexableFragmentRange* method
- *OESetupMCSFragDatabaseOptions*

FragmentationCuts

Passing this constant to the *OEConfigureMCSFragDatabaseOptions* function results in generating the following default interface to configure the fragmentation bond cut combinations to be performed.

```
Contents of parameter -fragcuts
    Aliases : -cuts
    Type : string
    Allow list : true
    Default : All
    Simple : true
    Required : false
    Legal values : Single Double Triple All
    Brief : Fragmentation bond combinations
    Detail
```

See also

- *OESetupMCSFragDatabaseOptions*
- *OEMCSFragDatabaseOptions.SetOptions* method
- *OEMatchedPairOptions.SingleCuts* constant
- *OEMatchedPairOptions.DoubleCuts* constant
- *OEMatchedPairOptions.TripleCuts* constant
- *OEMatchedPairOptions.AllCuts* constant

HeavyAtomLimit

Passing this constant to the *OEConfigureMCSFragDatabaseOptions* function results in generating the following default interface to configure the limit for the maximum number of heavy atoms allowed for an indexed molecule.

```
Contents of parameter -atomlimit
    Aliases : -alim
    Type : int
    Allow list : false
    Default : 100
```

(continues on next page)

(continued from previous page)

```
    Simple : true
    Required : false
    Brief : Heavy atom limit
```

See also

- *OEMCSFragDatabaseOptions.SetMaxAtomFilter* method
- *OESetupMCSFragDatabaseOptions*

IndexHydrogenSites

Passing this constant to the *OEConfigureMCSFragDatabaseOptions* function results in generating the following default interface to configure whether matched pairs that differ simply by hydrogen substitution is allowed.

```
Contents of parameter -indexhydsites
    Aliases : -indexh
    Type : bool
    Allow list : false
    Default : true
    Simple : true
    Required : false
    Brief : Index hydrogen sites
```

See also

- *OEMCSFragDatabaseOptions.SetOptions* method
- *OEMatchedPairOptions.IndexHSites* constant
- *OESetupMCSFragDatabaseOptions*

RingCuts

Passing this constant to the *OEConfigureMCSFragDatabaseOptions* function results in generating the following default interface to configure the maximum number of ring cuts allowed.

```
Contents of parameter -ringcuts
    Aliases : -ring
    Type : int
    Allow list : false
    Default : 0
    Simple : true
    Required : false
    Brief : Maximum number of ring fragmentation cuts allowed (0: disabled)
```

See also

- *OEMCSFragDatabaseOptions.SetRingFragLimit* method
- *OESetupMCSFragDatabaseOptions*

OEMCSScoreType

This namespace contains constants representing various supported similarity computations.

Undefined

This constant represents an undefined or illegal context state.

AtomCount

This constant represents a request for atom counts that relate the fragment core, query and indexed molecule.

BondCount

This constant represents a request for bond counts that relate the fragment core, query and indexed molecule.

Default

This constant represents the default score type setting of *OEMCSScoreType.AtomCount*.

OEMatchedPairAnalyzerFileType

This namespace contains constants representing supported matched pair analysis serialization export types.

MMPIndex

This constant identifies the only currently supported serialization type for an *OEMatchedPairAnalyzer* object.

MCSFrags

This constant identifies the only currently supported serialization type for an *OEMCSFragDatabase* object.

UNDEFINED

This constant represents any unknown file type for index serialization.

OEMatchedPairContext

This namespace contains constants representing various supported context requests for matched pair transformations.

Undefined

This constant represents an undefined or illegal context state.

AllBonds

This constant represents a request for the most explicit chemistry context for the transformation information.

Bond0

This constant represents a request for 0-bonds of chemistry context in the returned transformation information. This constant represents a request for the most general context for the transformation information.

Bond1

This constant represents a request for 1-bond of chemistry context in the returned transformation information.

Bond2

This constant represents a request for 2-bonds of chemistry context in the returned transformation information.

Bond3

This constant represents a request for 3-bonds of chemistry context in the returned transformation information.

Default

This constant represents the default transformation context setting of *OEMatchedPairContext.Bond1*.

OEMatchedPairIndexSetup

This namespace encodes symbolic constants used as bit-masks to indicate which matched pair indexing parameters are being created when invoking the *OEConfigureMatchedPairIndexOptions* function.

See also

- *OEMatchedPairAnalyzerOptions* class

All

The combination of following constants:

- *OEMatchedPairIndexSetup.BondFragLimit*
- *OEMatchedPairIndexSetup.FragMinMaxRange*

- *OEMatchedPairIndexSetup.FragmentationCuts*
- *OEMatchedPairIndexSetup.HeavyAtomLimit*
- *OEMatchedPairIndexSetup.IndexHydrogenSites*
- *OEMatchedPairIndexSetup.ProcessorType*
- *OEMatchedPairIndexSetup.UniquesOnly*

BondFragLimit

Passing this constant to the *OEConfigureMatchedPairIndexOptions* function results in generating the following default interface to configure maximum number of fragmentation bonds allowed.

```
Contents of parameter -bondfraglimit
    Aliases : -blim
    Type : int
    Allow list : false
    Default : 20
    Simple : true
    Required : false
    Brief : Bond fragmentation limit
```

See also

- *OEMatchedPairAnalyzerOptions.SetFragmentationLimit* method

FragMinMaxRange

Passing this constant to the *OEConfigureMatchedPairIndexOptions* function results in generating the following default interface to configure the min and max allowed fragment sizes to be indexed.

```
Contents of parameter -fragGe
    Aliases : -ge
    Type : float
    Allow list : false
    Default : 85.0
    Simple : true
    Required : false
    Legal ranges :
        0.0 to 100.0
    Brief : fragments greater than or equal to this heavy atom percentage

Contents of parameter -fragLe
    Aliases : -le
    Type : float
    Allow list : false
    Default : 100.0
    Simple : true
    Required : false
```

(continues on next page)

(continued from previous page)

```
    Legal ranges :
        0.0 to 100.0
    Brief : fragments less than or equal to this heavy atom percentage
```

See also

- *OEMatchedPairAnalyzerOptions.SetIndexableFragmentRange* method

FragmentationCuts

Passing this constant to the *OEConfigureMatchedPairIndexOptions* function results in generating the following default interface to configure the fragmentation bond cut combinations to be performed.

```
Contents of parameter -fragcuts
    Aliases : -cuts
    Type : string
    Allow list : true
    Default : All
    Simple : true
    Required : false
    Legal values : Single Double Triple All
    Brief : Fragmentation bond combinations
    Detail
```

See also

- *OEMatchedPairAnalyzerOptions.SetOptions* method
- *OEMatchedPairOptions.SingleCuts* constant
- *OEMatchedPairOptions.DoubleCuts* constant
- *OEMatchedPairOptions.TripleCuts* constant
- *OEMatchedPairOptions.AllCuts* constant

HeavyAtomLimit

Passing this constant to the *OEConfigureMatchedPairIndexOptions* function results in generating the following default interface to configure the limit for the maximum number of heavy atoms allowed for an indexed molecule.

```
Contents of parameter -atomlimit
    Aliases : -alim
    Type : int
    Allow list : false
    Default : 100
    Simple : true
```

(continues on next page)

(continued from previous page)

```
   Required : false
   Brief : Heavy atom limit
```

See also

- *OEMatchedPairAnalyzerOptions.SetMaxAtomFilter* method

IndexHydrogenSingleCuts

Passing this constant to the *OEConfigureMatchedPairIndexOptions* function results in generating the following default interface to configure whether matched pairs that differ simply by hydrogen substitution specifically at single cuts sites only.

```
Contents of parameter -indexhyd1cuts
   Aliases : -indexh1cuts
   Type : bool
   Allow list : false
   Default : false
   Simple : true
   Required : false
   Brief : Index hydrogen members for single cuts only
```

See also

- *OEMatchedPairAnalyzerOptions.SetOptions* method
- *OEMatchedPairOptions.IndexHSingleCuts* constant

IndexHydrogenSites

Passing this constant to the *OEConfigureMatchedPairIndexOptions* function results in generating the following default interface to configure whether matched pairs that differ simply by hydrogen substitution is allowed for all fragmentation cut types.

```
Contents of parameter -indexhydsites
    Aliases : -indexh
    Type : bool
    Allow list : false
    Default : true
    Simple : true
    Required : false
    Brief : Index hydrogen sites
```

See also

- *OEMatchedPairAnalyzerOptions.SetOptions* method

- *OEMatchedPairOptions.IndexHSites* constant

ProcessorType

Passing this constant to the *OEConfigureMatchedPairIndexOptions* function results in generating the following default interface to select the type of fragmentation processor to be used during indexing.

```
Contents of parameter -processortype
    Aliases : -ptype
    Type : string
    Allow list : false
    Default : Default
    Simple : true
    Required : false
    Legal values : Undefined Default HussainRea BemisMurcko HR BM
    Brief : Processor type
    Detail
```

See also

- *OEMatchedPairAnalyzerOptions.SetProcessorType* method
- *OEMatchedPairProcessor* namespace

UniquesOnly

Passing this constant to the *OEConfigureMatchedPairIndexOptions* function results in generating the following default interface to configure whether duplicate structures are allowed to be indexed.

```
Contents of parameter -unique
    Aliases : -u
    Type : bool
    Allow list : false
    Default : true
    Simple : true
    Required : false
    Brief : Unique molecules only
```

See also

- *OEMatchedPairAnalyzerOptions.SetOptions* method
- *OEMatchedPairOptions.UniquesOnly* constant

OEMatchedPairIndexStatus

This namespace contains constants representing status information returned from matched pair indexing.

BinaryIOError

This value indicates a serialization or deserialization issue during molecule fragmentation import or export.

CopyMolError

This value indicates that an internal error was encountered when attempting to copy the input structure for indexing manipulations.

Deactivated

This value indicates that an index loaded from serialization output encountered an internal loading failure and cannot be used as an active index.

DuplicateRecordID

This value indicates that the record id provided for the structure is already in use.

DuplicateStructure

This value indicates that the structure is already present in the index.

See also

- *OEMatchedPairAnalyzerOptions.SetOptions*
- *OEMatchedPairOptions.UniquesOnly*

FragmentRangeFilter

This value indicates that the structure was not added to the index as a result of exceeding the allowed range of the indexable fragment filter.

See also

- *OEMatchedPairAnalyzerOptions.SetIndexableFragmentRange*

FragmentationLimitFilter

This value indicates that the structure was not added to the index as a result of exceeding the number of allowed fragmentation bonds.

See also

- *OEMatchedPairAnalyzerOptions.SetFragmentationLimit*

HeavyAtomFilter

This value indicates that the structure was not added to the index due to exceeding the heavy atom limit.

See also

- *OEMatchedPairAnalyzerOptions.SetMaxAtomFilter*

IncompatibleOptions

This value indicates that an internal mismatch in fragmentation arguments and those for the target *OEMatchedPairAnalyzer* instance was identified. Contact support@eyesopen.com with details.

NoError

This value indicates a successful indexing status.

NoFragmentationBonds

This value indicates the structure contains no fragmentation bonds.

NoStructure

This value indicates that the structure is empty, or contains no heavy atoms.

ProcessorError

This value indicates that the internal processing engine has rejected the structure for unspecified reasons.

ProcessorFilter

This value indicates that the internal processing engine has rejected the structure based on the filter steps prior to indexing. One possible reason for the rejection is the presence of one or more unspecified atom types in the structure being indexed.

Unspecified

This value indicates an abnormal indexing status return of an unspecified nature.

OEMatchedPairOptions

This namespace contains constants representing various supported options to control a matched pair analysis.

AllCuts

This constant requests analysis of all fragmentation cuts from the processor-specific fragmentation strategy.

ComboCuts

This constant requests analysis of all single, double, triple combination fragmentation cuts from the processor-specific fragmentation strategy.

Default

This constant represents the default options setting for matched pair analysis which is *OEMatchedPairOptions.AllCuts*, *OEMatchedPairOptions.UniquesOnly*, *OEMatchedPairOptions.IndexHSites* and *OEMatchedPairOptions.ExportHKeys*.

DefaultFragIndex

This constant represents the default options setting for MCS fragment database indexing, which is *OEMatchedPairOptions.AllCuts* and *OEMatchedPairOptions.IndexHSites*.

DoubleCuts

This constant requests analysis of only double fragmentation cuts from the processor-specific fragmentation strategy.

ExportCompression

This constant affects the serialization of information from a *OEMatchedPairAnalyzer* instance. If enabled, singleton index nodes are *not* exported, thereby dramatically reducing the size of the serialized information. This option should *only* be used to export information from a static index since reading of this compressed index information and indexing of additional input structures miss valid matched molecular pairs.

ExportHKeys

This constant affects the serialization of information from a *OEMatchedPairAnalyzer* instance. If enabled, extra index information that supports the `OEMatchedPairOptions.IndexHSites` capability is serialized to the output file to increase index loading performance. If this information is suppressed, the serialized file size will be 10-20% smaller and this information will be regenerated as needed. If `OEMatchedPairOptions.IndexHSites` is not enabled, this option has no effect on serialization or deserialization.

IndexHSingleCuts

This constant activates additional indexing activity in order to capture matched pairs that differ by hydrogen presence at one or more of the substitution sites for SingleCuts only.

IndexHSites

This constant activates additional indexing activity in order to capture matched pairs that differ by hydrogen presence at one or more of the substitution sites for all fragmentation cut types.

SingleCuts

This constant requests analysis of only single fragmentation cuts from the processor-specific fragmentation strategy.

TripleCuts

This constant requests analysis of only triple fragmentation cuts from the processor-specific fragmentation strategy.

UniquesOnly

This constant requests uniqueness checking using canonical SMILES of the input structures to avoid duplicated indexed fragments.

ValidAlterOpts

This constant is a bit mask of all the options that can be changed on an active *OEMatchedPairAnalyzer* instance.

ValidFragIndexOpts

This constant is a bit mask of all the valid read-only option values for an active *OEMCSFragDatabase* instance.

ValidIndexOpts

This constant is a bit mask of all the valid read-only option values for an active *OEMatchedPairAnalyzer* instance.

ValidOptions

This constant is a bit mask of all the valid option values and is a union of *OEMatchedPairOptions.ValidAlterOpts* and *OEMatchedPairOptions.ValidIndexOpts*

OEMatchedPairProcessor

This namespace contains constants representing various supported processor strategies for a matched pair analysis.

Undefined

This constant represents an undefined or illegal context state.

Default

This constant represents a request for the default processor type which is currently specified to be *OEMatchedPairProcessor.HussainRea*

HussainRea

This constant represents a request for a [Hussain-2010] processor type.

BemisMurcko

This constant represents a request for a [Bemis-1996] processor type.

Custom

This constant identifies a custom type of processor engine that may be available in future versions.

OEMatchedPairTransformExtractMode

This namespace contains constants that control transform generation during retrieval from the matched pair index.

Unspecified

This value indicates an unspecified or null mode value.

Default

This mode represents the default extraction mode of *OEMatchedPairTransformExtractMode.Forward* and *OEMatchedPairTransformExtractMode.Backward*.

AddAttachmentPts

This extraction mode requests that attachment point information be added to the returned transforms to indicate the number of neighbor atoms adjacent to the desired context atoms.

AddMCSCorrespondence

This extraction mode returns correspondence information for the matched pairs by annotating the maximum common substructure with atom mapping information to indicate the MCS core.

Backward

This mode requests that only backward direction transforms be returned where 'backward' indicates the direction of later indexed structures over earlier indexed structures.

Forward

This mode requests that only forward direction transforms be returned where 'forward' indicates the direction of earlier indexed structures over later indexed structures.

NoIsotopes

This mode suppresses isotopic information in the transformation to allow more general use of the transformation with mixed isotopic input structures.

NoMatchedPairData

This extraction mode suppress populating the delta data for the matched pairs in the returned transforms, for example if the transformation and the matched pair ids are all that is required.

NoMatchedPairs

This extraction mode suppress populating the transformations with matched pair id information, for example, if the transformation itself is the only information of interest.

NoSMARTS

This mode suppresses the addition of SMARTS qualifiers to the transformation which is present to ensure more accurate application of the transformation. If the transform is being used for purposes other than performing transformations, the suppression of SMARTS qualifiers may provide simplification for downstream use of the transformation information.

Sorted

Since the direction of the transformations returned is dependent on input order of the indexed structures, this mode attempts to perform a simple sorting of the right hand and left hand sides of the transformation to collapse forward and backward transformations to a more deterministic form.

SubstLimitSkip

This extraction mode option indicates that any indexed core structures that exceed the maximum substituent limit are to be entirely skipped during the extraction process. By default, any MMP cores with large numbers of substituents are truncated to the maximum value to avoid the combinatorics of generating huge numbers of matched molecular pair transformations.

See also

OEMatchedPairTransformExtractOptions

SuppressNodeSort

This extraction mode option suppresses an internal sort which is used to return the largest MCS cores from the matched pairs. This is largely a debugging and analysis tool to allow examination of all matched pair cores.

OERegionType

This namespace contains constants representing various supported regions types for the *OEGetBemisMurcko* function.

See also

[Bemis-1996]

Ring

This constant identifies a ring region from the perception.

Linker

This constant identifies linker regions from the perception.

Framework

This constant identifies the framework region from the perception.

Sidechain

This constant identifies sidechain regions from the perception.

All

This constant requests that all regions be returned from the perception

4.1.3 OEMedChem Functions

OEApplyChEMBL18SolubilityTransforms

Warning

This is a deprecated API.

This version is based on older [ChEMBL18-2014] data and will be deprecated in a future version of **OEMedChem TK**, please use *OEApplyChEMBL24SolubilityTransforms* instead.

```
OESystem::OEIterBase<OEChem::OEMolBase>*␣
↪OEApplyChEMBL18SolubilityTransforms(OEChem::OEMolBase& input,
                                                                    int context␣
↪= OEMatchedPairContext::Bond2,
                                                                    unsigned␣
↪int requireNumPairs = 5u)
```

Given an input molecule, apply transformations derived from solubility data obtained from the [ChEMBL18-2014] database. The context argument controls the amount of chemistry information that should be included for the transformation reaction, see *OEMatchedPairContext*. This function supports only the *OEMatchedPairContext.Bond0* or *OEMatchedPairContext.Bond2* context values. The `minMMPThreshold` argument will only apply transformations that meet or exceed the specified number of matched pairs. Use a `minMMPThreshold` value of `0` to apply all transformations regardless of the number of matched pairs associated with them.

Note

This function does not perform any validation or filtering on the input molecule to be transformed. The caller is expected to perform validity and/or size checking to ensure sensible inputs are provided.

See also

- *OEApplyChEMBL24SolubilityTransforms* function

OEApplyChEMBL24SolubilityTransforms

```
OESystem::OEIterBase<OEChem::OEMolBase>*␣
↪OEApplyChEMBL24SolubilityTransforms(OEChem::OEMolBase& input,
                                                                    int context␣
↪= OEMatchedPairContext::Bond2,
                                                                    unsigned␣
↪int requireNumPairs = 5u)
```

Given an input molecule, apply transformations derived from solubility data obtained from the [ChEMBL24-2018] database. The context argument controls the amount of chemistry information that should be included for the transformation reaction, see *OEMatchedPairContext*. This function supports only the *OEMatchedPairContext.Bond0* or *OEMatchedPairContext.Bond2* context values. The `minMMPThreshold` argument will only apply transformations that meet or exceed the specified number of matched pairs. Use a `minMMPThreshold` value of `0` to apply all transformations regardless of the number of matched pairs associated with them.

Note

This function does not perform any validation or filtering on the input molecule to be transformed. The caller is

expected to perform validity and/or size checking to ensure sensible inputs are provided.

In the examples below, the input structures are transformed by the ChEMBL solubility transforms and exported to a file format that supports SD data. Each transformed structure will contain information about the solubility transform (as SMIRKS) that generated it, and the matched pair information associated with each transform (ChEMBL identifiers and solubility data). The added annotation data will contain the data fields, `OEMMP_normalized_value (uM)`, `OEMMP_published_value`, `OEMMP_examples (SMILES)`, and `OEMMP_transform (SMILES)` for subsequent analysis.

```
        // number of bonds of chemistry context at site of change
        //  for the applied transforms
        int totalmols = 0;
        int xformctxt = OEMatchedPairContext.Bond2;
        uint molidx = 0;
        OEGraphMol mol = new OEGraphMol();
        while (OEChem.OEReadMolecule(ifs, mol))
        {
            ++molidx;
            // consider only the largest input fragment
            OEChem.OEDeleteEverythingExceptTheFirstLargestComponent(mol);

            int smolcnt = 0;
            // only consider solubility transforms having at least 5 matched pairs
            foreach (OEMolBase solMol in OEMedChem.
↪OEApplyChEMBL24SolubilityTransforms(mol, xformctxt, 5))
            {
                // compute net change in solubility from MMP data
                if (OEChem.OEHasSDData(solMol, "OEMMP_normalized_value (uM)"))
                {
                    List<Double> deltasol = new List<Double>();
                    String sddata = OEChem.OEGetSDData(solMol, "OEMMP_normalized_value␣
↪(uM)");
                    String[] sdlines = sddata.Split('\n');
                    foreach (String sditem in sdlines)
                    {
                        String[] sdvalues = sditem.Split(',');
                        if (sdvalues.Length < 4)
                            continue;
                        if (sdvalues[2].Length == 0 || sdvalues[2].Length == 0)
                            continue;
                        deltasol.Add(Convert.ToDouble(sdvalues[3]) - Convert.
↪ToDouble(sdvalues[2]));
                    }
                    if (deltasol.Count == 0)
                        continue;

                    double avgsol = average(deltasol);

                    // reject examples with net decrease in solubility
                    if (avgsol < 0.0)
                        continue;
                    double sdev = stddev(deltasol);
```

(continues on next page)

(continued from previous page)

```
                    // annotate with average,stddev,num
                    OEChem.OEAddSDData(solMol,
                                       "OEMMP_average_delta_normalized_value", String.
↪Format("{0:F1},{1:F1},{2:D}", avgsol, sdev, deltasol.Count));
                }

                // export solubility transformed molecule with SDData annotations
                if (OEChem.OEWriteMolecule(ofs, solMol) == OEWriteMolReturnCode.Success)
                    smolcnt += 1;
            }

            OEChem.OEThrow.Info(String.Format("{0:D}: Exported molecule count, {1:D}",␣
↪molidx, smolcnt));
            totalmols += smolcnt;
        }
```

OECalculateComboBelief

```
float OECalculateComboBelief(float tanimotoCombo)
```

> **Note**
>
> See documentation for this function in the context of other belief function api points at *OECalculateETComboBelief*

OECalculateETComboBelief

```
float OECalculateComboBelief(float tanimotoCombo)
float OECalculateETComboBelief(float etCombo)
```

This pair of functions is based on [Muchmore-2008]. In that paper, the scores assessed for ROCS were the historic 'color score', in which the color was not normalized by self score, resulting in bias toward highly colored molecules. All recent versions of OEShape and ROCS use Tanimoto combo for ranking which is the sum of a shape Tanimoto and a color Tanimoto. The belief curve in *OECalculateComboBelief* is based on fitting of the data described in [Muchmore-2008] to ROCS Tanimoto combo scores [Brown-Muchmore-2008]. EON results are ranked on the ET combo score, which is a sum of a shape Tanimoto and an electrostatic Tanimoto. The belief curve in *OECalculateETComboBelief* is based on fitting of the data described in [Muchmore-2008] to ET combo scores [Brown-Muchmore-2008].

> **Warning**
>
> For use with other OpenEye toolkits, the full scores with range (0,2) should be passed into these functions.

OEConfigureMatchedPairIndexOptions

```
bool OEConfigureMatchedPairIndexOptions(OESystem::OEInterface& itf,
                                        unsigned int config =␣
↪OEMatchedPairIndexSetup::All)
```

Configures the given interface to add matched pair indexing parameters.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface This value has to be from the *OEMatchedPairIndexSetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEMatchedPairIndexSetup* namespace
- *OESetupMatchedPairIndexOptions* function

OEConfigureMCSFragDatabaseOptions

```
bool OEConfigureMCSFragDatabaseOptions(OESystem::OEInterface& itf,
                                       unsigned int config = OEMCSFragDatabaseSetup::All)
```

Configures the given interface to add MCS fragment database indexing parameters.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface This value has to be from the *OEMCSFragDatabaseSetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEMCSFragDatabaseSetup* namespace
- *OESetupMCSFragDatabaseOptions* function

OECreateIDString

```
std::string OECreateIDString(const OEChem::OEMolBase& mol)
```

Note

See documentation for this function in the context of other graph-edit similarity function api points at *OECreateIDStrings*

OECreateIDStrings

```
OESystem::OEIterBase<const std::string> * OECreateIDStrings(const OEChem::OEMolBase& mol,
                                                            const std::string& atomTag =
↪"")
```

This pair of functions exposes the graph-edit similarity used in clustering BROOD hitlists. Because this is a naturally O(N^2), the API is designed for efficiency using an asymmetric approach. For any 'accepted' molecule, the *OECreateIDStrings* function is used to generate a set of all the canonical SMILES of fragments which would be considered similar to the input molecule (this can be stored for fast lookup). If one wants to determine if a new molecule is similar to the original molecule, *OECreateIDString* is called, and the single string generated is compared to all the strings previously created. If the new string is an exact match to any of the multiple strings created from the prior molecule, then the two molecules are deemed similar.

The optional `atomTag` in the *OECreateIDStrings* function enables working with a subset of atoms in the molecule. This is useful in reducing the scope of the function when clustering BROOD hitlists for multi-query replacements.

OECreateMMPIndexFile

```
OECreateMMPIndexStatus OECreateMMPIndexFile(const std::string& mmpdbfname,
                                            const std::string& molfname,
                                            const OECreateMMPIndexOptions& opts)
```

This function reads the input structures from the provided `molfname` filename and generates an *OEMatchedPairAnalyzer* index from those structures which is saved to the provided `mmpdbfname` output file. Options for indexing are provided in the *OECreateMMPIndexOptions* argument.

In general, the provided input structures should have been thoroughly normalized, deduplicated or have data merged onto unique input structures for duplicate structures. There are no options for structure normalization or cleanup activities as part of this automatic indexing.

See also

- *OECreateMMPIndexStatus*
- *OECreateMMPIndexOptions*

OEDetermineConnectiveComplexity

```
float OEDetermineConnectiveComplexity(const OEChem::OEMolBase& mol)
```

Note

See documentation for this function in the context of other complexity api points at *OEDetermineMolecularComplexity*

OEDetermineElementalComplexity

```
float OEDetermineElementalComplexity(const OEChem::OEMolBase& mol)
```

Note

See documentation for this function in the context of other complexity api points at *OEDetermineMolecularComplexity*

OEDetermineMolecularComplexity

```
float OEDetermineMolecularComplexity(const OEChem::OEMolBase& mol)
float OEDetermineConnectiveComplexity(const OEChem::OEMolBase& mol)
float OEDetermineElementalComplexity(const OEChem::OEMolBase& mol)
float OEDetermineSymmetryComplexity(const OEChem::OEMolBase& mol)
```

Note

The molecule arguments are `const`, so the `OEPerceiveChiral` perception activity must be performed prior to calling these functions.

From [Bertz-1981]. Includes connectivity and elemental complexity. Though aromaticity is not addressed in the brief literature note, aromatic bonds are not counted as double bonds for the entropy calculation in this implementation. The three additional api points return the three individual terms that make up the Molecular Complexity as defined by Bertz.

See also

- *OEDetermineRingComplexity*
- *OENormalMolecularComplexity*

OEDetermineRingComplexity

```
float OEDetermineRingComplexity(const OEChem::OEMolBase& mol)
```

Note

The molecule argument is `const`, so this function requires that the `OEFindRingAtomsAndBonds` perception activity has been performed prior to calling this function.

This is a non-SSSR adaptation of the ring complexity algorithm of [Gasteiger-1979]. The values of this complexity are generally somewhat lower, but are well-correlated and follow the spirit of the SSSR method.

See also

- *OEDetermineMolecularComplexity*

OEDetermineStereoComplexity

```
float OEDetermineStereoComplexity(const OEChem::OEMolBase& mol)
```

Note

The molecule argument is `const`, so the `OEPerceiveChiral` perception activity must be performed prior to calling this function.

In [Boda-2007] a term was added for tetrahedral stereo complexity to the [Bertz-1981] approach. The function here reproduces that function with the addition of chiral bond stereo to the tetrahedral atom stereo.

See also

- *OENormalStereoComplexity*

OEDetermineSymmetryComplexity

```
float OEDetermineSymmetryComplexity(const OEChem::OEMolBase& mol)
```

Note

See documentation for this function in the context of other complexity api points at *OEDetermineMolecularComplexity*

OEGetBemisMurcko

```
OESystem::OEIterBase<OEChem::OEAtomBondSet>* OEGetBemisMurcko(OEChem::OEMolBase& mol,
                                                              const OEBemisMurckoOptions&
↪ options = OEBemisMurckoOptions())
```

Warning

The following function is deprecated. Please use the more generic function above.

```
OESystem::OEIterBase<OEChem::OEAtomBondSet>* OEGetBemisMurcko(OEChem::OEMolBase& mol,
                                                              unsigned perceivetype)
```

The *OEGetBemisMurcko* function partitions the given molecule into ring, linker and side chain fragments as defined in [Bemis-1996] . It returns an iterator over OEAtomBondSet objects where each OEAtomBondSet container stores the atoms and the bonds of an identified region. Each OEAtomBondSet returned is tagged with a role that identifies which group the set identifies.

The *OEBemisMurckoOptions* class can be used to specify both the *OERegionType* and also add additional functional groups to fragments. The *perceiveType* parameter only allows modification of the *OERegionType* of the fragments returned.

See also

- OEAtomBondSet and OEMolBase classes in the *OEChem TK* manual
- OERole in the *OEChem TK* manual

- *OEBemisMurckoOptions*
- *OERegionType*
- *OEGetRingLinkerSideChainFragments*
- *Molecule Fragmentation* chapter

OEGetFuncGroupFragments

```
OESystem::OEIterBase<OEChem::OEAtomBondSet>* OEGetFuncGroupFragments(const␣
↪OEChem::OEMolBase& mol)
```

The *OEGetFuncGroupFragments* function partitions the given molecule into functional groups and returns an iterator over OEAtomBondSet objects. Each OEAtomBondSet container stores the atoms and the bonds of an identified fragment. See example in *Figure: Example of fragments returned by the OEGetFuncGroupFragments function* where each fragment returned by the *OEGetFuncGroupFragments* function is highlighted with a different color.

The *OEGetFuncGroupFragments* function uses the following heuristics to fragment a molecule:

- rings are left intact *i.e.* considered as a unit
- exo-ring double bonds are not cleaved from adjacent rings
- hetero atoms next to each other are kept together
- sp and sp2 atoms next to each other are kept together
- in order to avoid generating one atom fragments isolated atoms are attached to the smallest neighbor fragment

Note

The molecule argument is `const`, so this function requires that the `OEFindRingAtomsAndBonds` perception activity has been performed prior to calling this function.

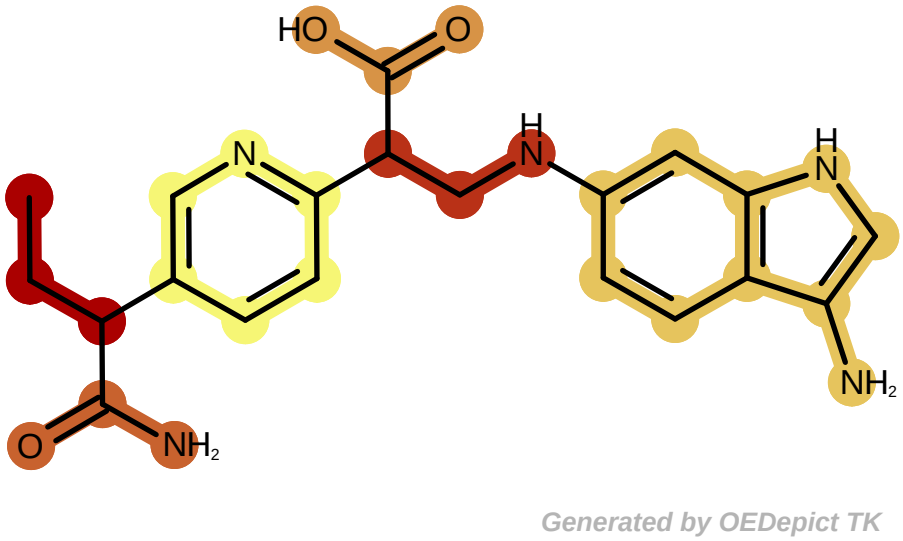


Fig. 1: **Example of fragments returned by the OEGetFuncGroupFragments function**

See also

- OEAtomBondSet and OEMolBase classes in the *OEChem TK* manual
- *Molecule Fragmentation* chapter

OEGetMatchedMolecularPairs

```
OESystem::OEIterBase<OEMedChem::OEMatchedPair> *
```

OESystem::OEIterBase<OEMedChem::OEMatchedPair>* OEGetMatchedMolecularPairs(const OEMedChem::OEMatchedPairAnalyzer &mmpindex, unsigned int molID, const OEMedChem::OEMatchedPairTransformExtractOptions &extractOpts=OEMedChem::OEMatchedPairTransformExtractOptions())
OESystem::OEIterBase<OEMedChem::OEMatchedPair> *

OESystem::OEIterBase<OEMedChem::OEMatchedPair>* OEGetMatchedMolecularPairs(const OEMedChem::OEMatchedPairAnalyzer &mmpindex, const OEChem::OEMolBase &probe, const OEChem::OEAtomBondSet &probeAttachSites, const OEMedChem::OEMatchedPairTransformExtractOptions &extractOpts=OEMedChem::OEMatchedPairTransformExtractOptions())

The first function returns the matched pairs associated with a specific molecule ID in the *OEMatchedPairAnalyzer* index.

The second overloaded function returns matched molecular pairs related to changes at specific substitution point(s) on the provided probe structure. Additionally, any *atoms (atomic number 0) on the probe structure can be used to indicate the site(s) that are open valences and which can be arbitrarily substituted.

See also

- *OEMatchedPairGetTransforms* function
- *OEMatchedPairAnalyzer* class
- *OEMatchedPairTransformExtractOptions* class
- OEAtomBondSet class

OEGetMatchedPairAnalyzerFileType

```
unsigned int OEGetMatchedPairAnalyzerFileType(const char* ext)
```

Returns a symbolic constant from the *OEMatchedPairAnalyzerFileType* namespace for the file extension 'ext'.

OEGetRegionType

```
unsigned int OEGetRegionType (const std::string& regionName)
```

Given the region name, return the *OERegionType* value.

See also

- *OERegionType*

OEGetRegionTypeName

```
const char* OEGetRegionTypeName (unsigned int regionType)
```

Given the *OERegionType* value, return a string representation of the region name.

See also

- *OERegionType*

OEGetRingChainFragments

```
OESystem::OEIterBase<OEChem::OEAtomBondSet>* OEGetRingChainFragments(const␣
↪OEChem::OEMolBase& mol)
```

The *OEGetRingChainFragments* function partitions the given molecule into ring and chain fragments and returns an iterator over OEAtomBondSet objects. Each OEAtomBondSet container stores the atoms and the bonds of an identified fragment. See example in *Figure: Example of fragments returned by the OEGetRingChainFragments function* where each fragment returned by the *OEGetRingChainFragments* function is highlighted with a different color.

The perceived regions are based upon the heavy atoms in the molecule. If the molecule contains explicit hydrogen atoms, those explicit hydrogens will be returned in the same perception set(s) as their alpha heavy atom neighbor. If

the perception set(s) returned are used to subset or fragment the molecule, a mixture of implicit and explicit hydrogens may result, so care should be taken to normalize the structure for subsequent use.

Note

The molecule argument is `const`, so this function requires that the `OEFindRingAtomsAndBonds` perception activity has been performed prior to calling this function.

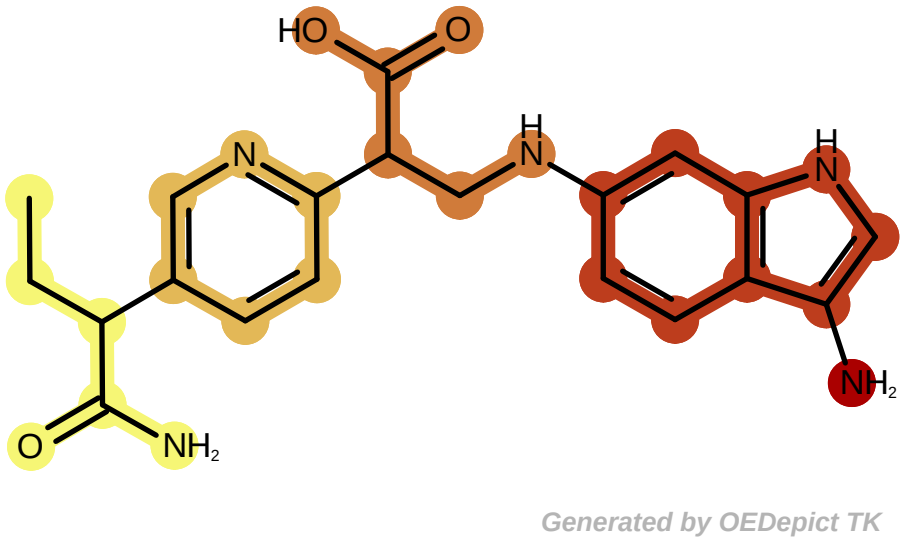


Fig. 2: **Example of fragments returned by the OEGetRingChainFragments function**

See also

- OEAtomBondSet and OEMolBase classes in the *OEChem TK* manual
- *Molecule Fragmentation* chapter

OEGetRingLinkerSideChainFragments

```
OESystem::OEIterBase<OEChem::OEAtomBondSet>* OEGetRingLinkerSideChainFragments(const␣
↪OEChem::OEMolBase& mol,
                                                                                const␣
↪OEBemisMurckoOptions& options = OEBemisMurckoOptions())
```

Warning

The following function is deprecated. Please use the more generic function above.

```
OESystem::OEIterBase<OEChem::OEAtomBondSet>* OEGetRingLinkerSideChainFragments(const␣
↪OEChem::OEMolBase& mol,
                                                                                unsigned␣
↪perceiveType)
```

The *OEGetRingLinkerSideChainFragments* function partitions the given molecule into ring, linker and side chain fragments as defined in [Bemis-1996] .

The *OEBemisMurckoOptions* class can be used to specify both the *OERegionType* and also add additional functional groups to fragments. The *perceiveType* parameter only allows modification of the *OERegionType* of the fragments returned.

It returns an iterator over OEAtomBondSet objects where each OEAtomBondSet container stores the atoms and the bonds of an identified fragment. See example in *Figure: Example of fragments returned by the OEGetRingLinkerSideChainFragments function* where each fragment returned by the *OEGetRingLinkerSideChainFragments* function is highlighted with a different color.

The perceived regions are based upon the heavy atoms in the molecule. If the molecule contains explicit hydrogen atoms, those explicit hydrogens will be returned in the same perception set(s) as their alpha heavy atom neighbor. If the perception set(s) returned are used to subset or fragment the molecule, a mixture of implicit and explicit hydrogens may result, so care should be taken to normalize the structure for subsequent use.

Definitions from [Bemis-1996]:

ring systems
: Cycles within the graph representation of molecules and cycles sharing an edge (a connection between two atoms or a bond).

linker atoms
: Atoms that are on the direct path connecting two ring systems are defined as linker atoms.

side chain atoms
: Any nonring, nonlinker atoms are defined as side chain atoms.

Note

The molecule argument is `const`, so this function requires that the `OEFindRingAtomsAndBonds` perception activity has been performed prior to calling this function.

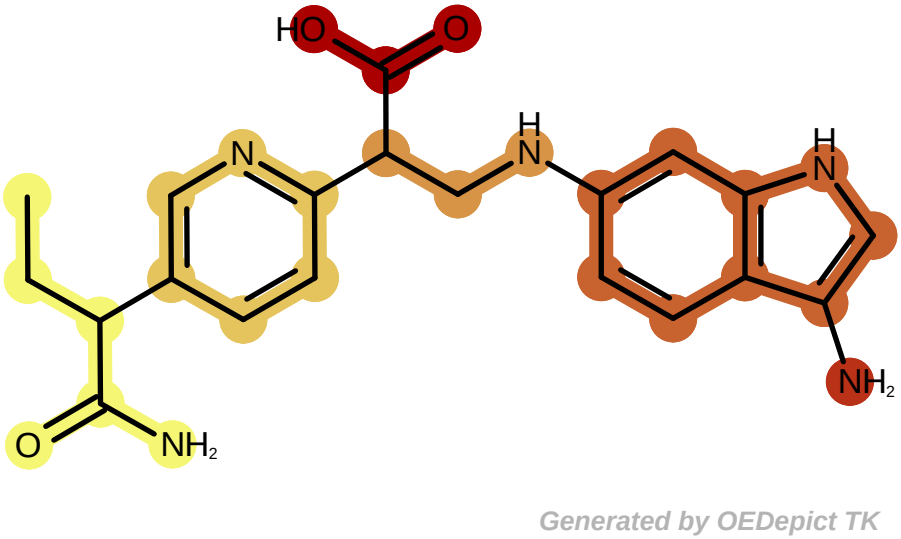


Fig. 3: **Example of fragments returned by the OEGetRingLinkerSideChainFragments function**

See also

- OEAtomBondSet and OEMolBase classes in the *OEChem TK* manual
- *Molecule Fragmentation* chapter

OEGetRingValue

```
float OEGetRingValue(const OEChem::OEMolBase& mol)
```

This is a function that assigns a value to a molecule or fragment based on the number and type of ring systems it contains. Ring systems that are favored by small-molecule medicinal chemistry receive higher scores. No preference is given between aromatic and aliphatic rings. Ring fusions are favored over spiro ring systems and common ring sizes (5,6) are favored over unusual ring sizes. This value is not recommended for filtering or cutoffs, yet can be useful for ordering compounds from a set of potential candidates.

OEIsMatchedPairAnalyzerFileType

```
bool OEIsMatchedPairAnalyzerFileType(unsigned int type)
bool OEIsMatchedPairAnalyzerFileType(const std::string& filename)
```

This function returns a boolean indicating whether the provided *OEMatchedPairAnalyzerFileType* value or filename is a serialized index type, i.e., has a file extension of `.mmpidx`.

OEIsMCSFragDatabaseFileType

```
bool OEIsMCSFragDatabaseFileType(unsigned int type)
bool OEIsMCSFragDatabaseFileType(const std::string& filename)
```

This function returns a boolean indicating whether the provided *OEMatchedPairAnalyzerFileType* value or filename is a serialized fragment index type, i.e., has a file extension of `.mcsfrag`.

OEIsReadableMatchedPairAnalyzer

Warning

The following functions are deprecated. Please use the more generic *OEIsMatchedPairAnalyzerFileType* to test for the appropriate file extension for a serialized index.

```
bool OEIsReadableMatchedPairAnalyzer(unsigned int type)
bool OEIsReadableMatchedPairAnalyzer(const std::string& filename)
```

This function returns a boolean indicating whether the provided *OEMatchedPairAnalyzerFileType* value or filename is a readable serialized index.

Warning

This function is deprecated as of 2017.Feb and will be removed in a future version of **OEMedChem TK**, please use *OEIsMatchedPairAnalyzerFileType* instead.

OEIsWriteableMatchedPairAnalyzer

Warning

The following functions are deprecated. Please use the more generic *OEIsMatchedPairAnalyzerFileType* to test for the appropriate file extension for a serialized index.

```
bool OEIsWriteableMatchedPairAnalyzer(unsigned int type)
bool OEIsWriteableMatchedPairAnalyzer(const std::string& filename)
```

This function returns a boolean indicating whether the provided *OEMatchedPairAnalyzerFileType* value or filename is a writeable serialized index.

> **Warning**
>
> This function is deprecated as of 2017.Feb and will be removed in a future version of **OEMedChem TK**, please use *OEIsMatchedPairAnalyzerFileType* instead.

OEMatchedPairApplyTransforms

```
OESystem::OEIterBase<OEChem::OEMolBase> *
OESystem::OEIterBase<OEChem::OEMolBase>* OEMatchedPairApplyTransforms(const␣
↪OEChem::OEMolBase& mol,
                                                                      ␣
↪OEMatchedPairAnalyzer& mmp,
                                                                       int context =␣
↪OEMatchedPairContext::Default,
                                                                       unsigned int␣
↪requireNumPairs = 0u)
```

The *OEMatchedPairApplyTransforms* function will take the matched pair analysis provided by the *OEMatchedPairAnalyzer* engine to generate a set of transformations to apply to the input molecule. The output molecule(s) will be annotated with SD data that identifies the original pair of molecules from the matched pair indexing, any data associated with those original molecules, and the transformation that resulted in the output molecule. The context argument controls the amount of chemistry information that should be used in constructing the transformation reaction, see *OEMatchedPairContext*. The `minMMPThreshold` argument will only apply transformations that meet or exceed the specified number of matched pairs. A `minMMPThreshold` value of `0` will apply transformations regardless of the number of matched pairs associated with them.

OEMatchedPairContextName

```
const char* OEMatchedPairContextName(int context)
```

Given a *OEMatchedPairContext* value, return a string representation of the context.

OEMatchedPairContextType

```
int OEMatchedPairContextType(const std::string& scontext)
```

Given the context name, return the *OEMatchedPairContext* value.

OEMatchedPairGetTransforms

```
OESystem::OEIterBase<OEMedChem::OEMatchedPairTransform>*␣
↪OEMatchedPairGetTransforms(const OEMatchedPairAnalyzer& mmpindex,
                                                                        int␣
↪context,
                                                                       ␣
↪unsigned int extractMode = OEMatchedPairTransformExtractMode::Default)
OESystem::OEIterBase<OEMedChem::OEMatchedPairTransform>*␣
↪OEMatchedPairGetTransforms(const OEMatchedPairAnalyzer& mmpindex,
                                                                       ␣
↪const OEChem::OEMolBase& probemol,
                                                                        int␣
↪context,
```

(continues on next page)

(continued from previous page)

```
                                                                                ␣
↪unsigned int extractMode = OEMatchedPairTransformExtractMode::Default)
OESystem::OEIterBase<OEMedChem::OEMatchedPairTransform>*␣
↪OEMatchedPairGetTransforms(const OEMatchedPairAnalyzer& mmpindex,
                                                                                ␣
↪const OEMatchedPairTransformExtractOptions& extractOpts =␣
↪OEMatchedPairTransformExtractOptions())
OESystem::OEIterBase<OEMedChem::OEMatchedPairTransform>*␣
↪OEMatchedPairGetTransforms(const OEMatchedPairAnalyzer& mmpindex,
                                                                                ␣
↪const OEChem::OEMolBase& probemol,
                                                                                ␣
↪const OEMatchedPairTransformExtractOptions& extractOpts =␣
↪OEMatchedPairTransformExtractOptions())
OESystem::OEIterBase<OEMedChem::OEMatchedPairTransform>*␣
↪OEMatchedPairGetTransforms(const OEMatchedPairAnalyzer& mmpindex,
                                                                                ␣
↪const OEChem::OEMolBase& probemol,
                                                                                ␣
↪const OEMatchedPairAnalyzerOptions& fragOpts,
                                                                                ␣
↪const OEMatchedPairTransformExtractOptions& extractOpts =␣
↪OEMatchedPairTransformExtractOptions())
OESystem::OEIterBase<OEMedChem::OEMatchedPairTransform>*␣
↪OEMatchedPairGetTransforms(const OEMatchedPairAnalyzer& mmpindex,
                                                                                ␣
↪const OEChem::OEMolBase& probemol,
                                                                                ␣
↪const OEChem::OEAtomBondSet& probesubst,
                                                                                ␣
↪const OEMatchedPairTransformExtractOptions& extractOpts =␣
↪OEMatchedPairTransformExtractOptions())
OESystem::OEIterBase<OEMedChem::OEMatchedPairTransform>*␣
↪OEMatchedPairGetTransforms(const OEMatchedPairAnalyzer& mmpindex,
                                                                                ␣
↪const OEChem::OEMolBase& probemol,
                                                                                ␣
↪const char* probesubstDataField,
                                                                                ␣
↪const OEMatchedPairTransformExtractOptions& extractOpts =␣
↪OEMatchedPairTransformExtractOptions())
```

Given an analyzer containing indexed structures, retrieves the transformations derived from the discovered matched pair information.

Overloaded functions support the ability to pass in an arbitrary structure and return related matched molecular pair transformations that share any of the identified substituents from the probe structure. The probe molecule is not added to the index, it is only used to locate related transformations. The probe molecule is subject to the same filtering constraints as the structures present in the index. An additional overloaded function signature is provided that allows passing different fragmentation filtering constraints for the probe structure than the indexed structures to allow a more generalized extraction of transformations.

The final two overloaded functions support the ability to pass in an arbitrary structure and return related matched molecular pair transformations at one or more identified attachment site(s) on the probe structure.

The context argument controls the amount of chemistry context to include in the returned transforms. The extract mode argument provides some control over the transformation string returned. The extraction options object allows the most control over the content of the extracted transformations.

See also

- *OEGetMatchedMolecularPairs* function
- *OEMatchedPairContext*
- *OEMatchedPairTransformExtractMode*
- *OEMatchedPairTransformExtractOptions*
- *OEMatchedPairAnalyzer*
- *OEMatchedPairTransform*
- *OEMatchedPair*

OEMatchedPairIndexStatusName

```
const char* OEMatchedPairIndexStatusName(int status)
```

Given a *OEMatchedPairIndexStatus* value, return a string representation of the status.

OEMatchedPairIndexStatusType

```
int OEMatchedPairIndexStatusType(const std::string& scontext)
```

Given a status name, return the *OEMatchedPairIndexStatus* value.

OEMatchedPairOptionsName

```
const char* OEMatchedPairOptionsName(unsigned int option)
```

Given a *OEMatchedPairOptions* value, return a string representation of the option.

OEMatchedPairOptionsType

```
unsigned int OEMatchedPairOptionsType(const std::string& sOption)
```

Given the option name, return the *OEMatchedPairOptions* type.

OEMatchedPairProcessorName

```
const char* OEMatchedPairProcessorName(unsigned int type)
```

Given a *OEMatchedPairProcessor* type, return a string representation of the processor type.

OEMatchedPairProcessorType

```
unsigned int OEMatchedPairProcessorType(const std::string& sname)
```

Given the processor name, return the *OEMatchedPairProcessor* type.

OEMCSFragDatabaseCoreQuery

```
bool OEMCSFragDatabaseCoreQuery(const std::string& MCScoreSMI,
                                std::string& MCScoreSMARTS)
```

This function can be used to convert the MCS SMILES core string into a more restrictive SMARTS query for use in highlighting or mapping activities that use the MCS core and the query or target structures.

See also

- *OEMCSMolSimScore* class
- *OEMCSFragDatabase* class
- *OEMCSFragDatabase.GetScores* method
- *OEMCSFragDatabase.GetSortedScores* method

OEMedChemGetArch

```
const char* OEMedChemGetArch()
```

Returns the internal build string used by OpenEye, Cadence Molecular Sciences to identify the hardware architecture. The format of these strings may change over time.

OEMedChemGetLicensee

```
std::string OEMedChemGetLicensee()
bool OEMedChemGetLicensee(std::string& licensee)
```

Returns the licensee name from the license file.

OEMedChemGetPlatform

```
const char* OEMedChemGetPlatform()
```

Returns the internal build string used by OpenEye, Cadence Molecular Sciences to identify a platform. The format of these strings may change over time, and future distributions may contain different values even when using the same operating system, compiler and processor. For example, on a `x86_64` Red Hat Enterprise Linux box this would return `redhat-RHEL5-g++4.1-x64`.

OEMedChemGetRelease

```
const char* OEMedChemGetRelease()
```

Returns the release name of the *OEMedChem* library being used. This returns a value similar to `0.10.0` for production versions of the library, and `0.10.0 debug` for the checking version of the library.

OEMedChemGetSite

```
std::string OEMedChemGetSite()
bool OEMedChemGetSite(std::string& site)
```

Returns the physical site location of the licensee as registered in the license file.

OEMedChemGetVersion

```
unsigned int OEMedChemGetVersion()
```

Returns the version number of the library being used. This is an unsigned integer value indicating the date on which the library was built, for example `20141001`, for the 1st of October 2014. This value should be used when reporting problems, and unlike the release string, may be used in comparisons if needed.

OEMedChemIsLicensed

```
bool OEMedChemIsLicensed(const char* feature = 0,
                         unsigned int* expdate = 0)
```

Determines whether a valid license file is present. This function may be called without a legitimate run-time license to determine whether it is safe to call any **OEMedChem TK** functionality.

The 'feature' argument can be used to check for a valid license to **OEMedChem TK** along with the following features: 'python', 'java', or 'clr' (for CSharp).

The 'expdate' argument can be used to obtain the expiration date of the current **OEMedChem TK** license.

See also

- `OEChemIsLicensed` function

OEMoleculeToCores

```
OESystem::OEIterBase<const std::string>* OEMoleculeToCores(const OEChem::OEMolBase& mol,
                                                           const␣
↪OEMCSFragDatabaseOptions& opts,
                                                           bool permuteFragments = true)
```

Given a molecule, return the fragmentation cores using the provided fragmentation options. If the `permuteFragments` argument is `true`, all combinations of the generated fragmentation cores are generated, otherwise a unique set of multi-fragment cores is returned representing all combinations of bond fragmentations between the min and max cut limits.

Shown below is a version that uses the free function and custom options setup from command line arguments:

```
            // set the MCS fragment database options from the command-line arguments
            OEMCSFragDatabaseOptions fragopts = new OEMCSFragDatabaseOptions();
            if (!OEMedChem.OEConfigureMCSFragDatabaseOptions(itf))
                OEChem.OEThrow.Fatal("Error configuring options");
            if (!OEMedChem.OESetupMCSFragDatabaseOptions(fragopts, itf)) {
                OEChem.OEThrow.Fatal("Error setting options");
            }

            // use the custom options to fragment an arbitrary input molecule
            Console.WriteLine("MoleculeToCores using command-line options:");
            List<string> cores = new List<string>();
            foreach (String c in OEMedChem.OEMoleculeToCores(mol, fragopts))
                cores.Add(c);
            cores.Sort();
            int corenum = 0;
            foreach (String core in cores)
```

(continues on next page)

(continued from previous page)

```
            {
                Console.WriteLine(corenum + ": " + core);
                ++corenum;
            }
```

See also

- *OEMCSFragDatabase* class
- *OEMCSFragDatabaseOptions* class

OENormalMolecularComplexity

```
float OENormalMolecularComplexity(const OEChem::OEMolBase& mol)
```

Note

The molecule argument is `const`, so the `OEPerceiveChiral` perception activity must be performed prior to calling this function.

This function is a convenience that scales the value from *OEDetermineMolecularComplexity* to [0-1] assuming the underlying complexity is [0-1000] which is the case from most small molecules.

OENormalStereoComplexity

```
float OENormalStereoComplexity(const OEChem::OEMolBase& mol)
```

Note

The molecule argument is `const`, so the `OEPerceiveChiral` perception activity must be performed prior to calling this function.

This function is a convenience that scales the value from *OEDetermineStereoComplexity* to a range of approximately [0-1] assuming a non-normalized complexity range of [0-4.3] which corresponds to between 0 and 20 stereo centers.

OEReadMatchedPairAnalyzer

```
bool OEReadMatchedPairAnalyzer(const std::string& fname,
                               OEMatchedPairAnalyzer& mmpindex,
                               bool enableMerge = false,
                               bool regenHMemberKeys = false)
bool OEReadMatchedPairAnalyzer(OEPlatform::oeistream& ifs,
                               OEMatchedPairAnalyzer& mmpindex,
                               bool enableMerge = false,
                               bool regenHMemberKeys = false)
```

This function reads the serialized contents of a matched pair index and populates the *OEMatchedPairAnalyzer* instance as below. An `enableMerge` argument of `false` is used to indicate that the *OEMatchedPairAnalyzer* instance should

be cleared before the read, or `true` to indicate that the read index information should add (and update) existing data. If the `regenHMemberKeys` argument is `true` and the index includes extra H-member indexing information, the serialized H-member information will be ignored and regenerated during the read.

```
        OEMatchedPairAnalyzer mmpimp = new OEMatchedPairAnalyzer();
        if (!OEMedChem.OEIsMatchedPairAnalyzerFileType(mmpimport))
            OEChem.OEThrow.Fatal("Not a valid matched pair index input file, " +␣
↪mmpimport);
        else if (!OEMedChem.OEReadMatchedPairAnalyzer(mmpimport, mmpimp))
            OEChem.OEThrow.Fatal("Index deserialization failed");
        else
            OEChem.OEThrow.Info("Index deserialization complete");
```

See also

- *OEIsReadableMatchedPairAnalyzer*
- *OEWriteMatchedPairAnalyzer*

OEReadMCSFragDatabase

```
bool OEReadMCSFragDatabase(const std::string& fname,
                           OEMCSFragDatabase& mcsfragdb,
                           bool enableMerge = false)
bool OEReadMCSFragDatabase(OEPlatform::oeistream& ifs,
                           OEMCSFragDatabase& mcsfragdb,
                           bool enableMerge = false)
```

This function reads the serialized contents of an MCS fragment index and populates the *OEMCSFragDatabase* instance. An `enableMerge` argument of `false` is used to indicate that the *OEMCSFragDatabase* instance should be cleared before the read, or `true` to indicate that the read index information should add (and update) existing data.

See also

- *OEIsMCSFragDatabaseFileType*
- *OEWriteMCSFragDatabase*

OEReadMMPIndexMolecules

```
bool OEReadMMPIndexMolecules(OEPlatform::oeistream& ifs,
                             OEMatchedPairAnalyzer& mmpindex,
                             bool regenHKeys = false)
bool OEReadMMPIndexMolecules(const std::string& fragbytes,
                             OEMatchedPairAnalyzer& mmpindex,
                             bool regenHKeys = false)
```

This function deserializes Matched Pair fragmentation information from the passed string or oeistream and adds the information to the *OEMatchedPairAnalyzer* provided.

The goal of this function is to stream the precomputed fragmentation information directly into the Matched Molecular Pair index with maximum efficiency.

See also

- *OEWriteMMPIndexMolecules*

OESetupMatchedPairIndexOptions

```
bool OESetupMatchedPairIndexOptions(OEMatchedPairAnalyzerOptions& opts,
                                    const OESystem::OEInterface& itf)
```

Initializes an *OEMatchedPairAnalyzerOptions* object from the parameters of a given interface.

See also

- *OEMatchedPairAnalyzerOptions* class
- *OEConfigureMatchedPairIndexOptions* function

OESetupMCSFragDatabaseOptions

```
bool OESetupMCSFragDatabaseOptions(OEMCSFragDatabaseOptions& opts,
                                   const OESystem::OEInterface& itf)
```

Initializes an *OEMCSFragDatabaseOptions* object from the parameters of a given interface.

See also

- *OEMCSFragDatabaseOptions* class
- *OEConfigureMCSFragDatabaseOptions* function

OETotalMolecularComplexity

```
float OETotalMolecularComplexity(const OEChem::OEMolBase& mol)
```

Note

The molecule argument is `const`, so the `OEPerceiveChiral` perception activity must be performed prior to calling this function.

This is the final function to use for assessing molecular complexity. *OETotalMolecularComplexity* = *OENormalMolecularComplexity* + *OENormalStereoComplexity*. In turn, *OENormalMolecularComplexity* = *OEDetermineConnectiveComplexity* + *OEDetermineElementalComplexity*. Ring complexity is not included because it did not contribute significantly to reproduction of chemists' notion of synthetic accessibility. Ring complexity is available in *OEDetermineRingComplexity*.

See also

- *OEDetermineConnectiveComplexity*

- *OEDetermineElementalComplexity*
- *OEDetermineMolecularComplexity*
- *OEDetermineRingComplexity*
- *OEDetermineSymmetryComplexity*
- *OENormalMolecularComplexity*
- *OENormalStereoComplexity*

OEWriteMatchedPairAnalyzer

```
bool OEWriteMatchedPairAnalyzer(const std::string& fname,
                                const OEMatchedPairAnalyzer& mmpindex)
bool OEWriteMatchedPairAnalyzer(OEPlatform::oeostream& ofs,
                                const OEMatchedPairAnalyzer& mmpindex)
```

This function serializes the state of the *OEMatchedPairAnalyzer* instance argument to a file or stream as below:

```
        OEMatchedPairAnalyzer mmp = new OEMatchedPairAnalyzer();
        int recindex = 0;
        OEGraphMol mol = new OEGraphMol();
        while (OEChem.OEReadMolecule(ifs, mol))
        {
            ++recindex;

            int status = mmp.AddMol(mol, recindex);
            if (status != recindex)
                OEChem.OEThrow.Warning(String.Format("{0}: molecule indexing error,␣
↪status={1}",
                                                     recindex, OEMedChem.
↪OEMatchedPairIndexStatusName(status)));

        }
        Console.WriteLine("Index complete, matched pairs = {0:D}", mmp.
↪NumMatchedPairs());

        // check for output filename with .mmpidx extension
        String mmpexport = itf.GetString("-output");
        if (!OEMedChem.OEIsMatchedPairAnalyzerFileType(mmpexport))
            OEChem.OEThrow.Info("Not a valid matched pair index output file, " +␣
↪mmpexport);
        else if (!OEMedChem.OEWriteMatchedPairAnalyzer(mmpexport, mmp))
            OEChem.OEThrow.Fatal("Index serialization failed");
        else
            OEChem.OEThrow.Info("Index serialization complete");
```

If the matched pair index information being exported is complete in the sense that additional structures are *not* intended to be added to the index upon a subsequent deserialization, significant savings in output file size is possible by suppressing the export of singleton nodes - those index nodes which do not contain any matched pairs. Conversely, if the exported index must be able to support adding additional structures to the index after deserialization, export compression should be avoided, otherwise significant matched pairs may be missed. To *enable* export compression (by default, *disabled*), see the following:

```
            // create analyzer class with defaults
            //   - compression option disabled by default
            OEMatchedPairAnalyzer mmpAnalyzer = new OEMatchedPairAnalyzer();
            // for serialization, enable export compression to
            //   remove singleton index nodes by modifying analyzer
            mmpAnalyzer.ModifyOptions(OEMatchedPairOptions.ExportCompression, 0);
```

```
            // create options class with defaults
            //   - compression option disabled by default
            OEMatchedPairAnalyzerOptions mmpOpts = new OEMatchedPairAnalyzerOptions();
            // for serialization, enable export compression to
            //   remove singleton index nodes by modifying analyzer
            mmpOpts.SetOptions(OEMatchedPairOptions.Default |
                               OEMatchedPairOptions.ExportCompression);
            // create analyzer class with compression option enabled
            OEMatchedPairAnalyzer mmpAnalyzer = new OEMatchedPairAnalyzer(mmpOpts);
```

See also

- *OEIsWriteableMatchedPairAnalyzer*
- *OEReadMatchedPairAnalyzer*

OEWriteMCSFragDatabase

```
bool OEWriteMCSFragDatabase(const std::string& fname,
                            const OEMCSFragDatabase& mcsfragdb)
bool OEWriteMCSFragDatabase(OEPlatform::oeostream& ofs,
                            const OEMCSFragDatabase& mcsfragdb)
```

This function serializes the state of the *OEMCSFragDatabase* instance argument to a file or stream.

See also

- *OEIsMCSFragDatabaseFileType*
- *OEReadMCSFragDatabase*

OEWriteMMPIndexMolecules

```
int OEWriteMMPIndexMolecules(OEPlatform::oeostream& ofs,
                             unsigned int recID,
                             const OEChem::OEMolBase& inmol,
                             const OEMatchedPairAnalyzer& mmpindex,
                             bool initStream = false)
int OEWriteMMPIndexMolecules(std::string& fragbytes,
                             unsigned int recID,
                             const OEChem::OEMolBase& inmol,
                             const OEMatchedPairAnalyzer& mmpindex,
                             bool writeHeader = false);
```

This function serializes Matched Pair fragmentation information from the passed molecule based on the indexing setting provided in the *OEMatchedPairAnalyzer* argument.

The goal of this function is to precompute all the fragmentation information needed to allow the provided input molecule to be recorded in the Matched Molecular Pair index with maximum efficiency. For multi-threaded or multi-processing index creation, moving most of the fragmentation processing activity outside of the index creation activity results in better indexing throughput overall.

See also

- *OEReadMMPIndexMolecules*

CHAPTER

FIVE

RELEASE HISTORY

5.1 OEMedChem TK 1.2.6

5.1.1 New Features

- A new overload of the *OECreateIDStrings* function has been added to enable speeding up **BROOD** hit list clustering when working with multiquery replacements.

5.2 OEMedChem TK 1.2.5

Minor internal improvements have been made.

5.3 OEMedChem TK 1.2.4

5.3.1 Minor Bug Fixes

- A bug causing a crash on Windows builds has been fixed.

5.4 OEMedChem TK 1.2.3

Minor internal improvements have been made.

5.5 OEMedChem TK 1.2.2

Minor internal improvements have been made.

5.6 OEMedChem TK 1.2.1

Minor internal improvements have been made.

5.7 OEMedChem TK 1.2.0

5.7.1 Internal changes

- Internal functions used to constrain Matched Molecular Pair transforms by applying SMARTS constraints have been replaced with calls to the new preliminary API functions in OEChem. See `OECreateAtomSmartsString`, `OECreateBondSmartsString`, and `OECreateSmartsString`.

5.8 OEMedChem TK 1.1.7

Minor internal improvements have been made.

5.9 OEMedChem TK 1.1.6

- Minor internal improvements have been made.

5.10 OEMedChem TK 1.1.5

Fall 2021

- Minor internal improvements have been made.

5.11 OEMedChem TK 1.1.4

July 2021

5.11.1 Minor bug fixes

- *OEMatchedPairOptions.ComboCuts* has been added by default to all cut options.

5.12 OEMedChem TK 1.1.3

Fall 2020

5.12.1 New features

- The preliminary API points, *OEReadMMPIndexMolecules* and *OEWriteMMPIndexMolecules* are now fully supported.
- Bemis-Murcko fragmentation has been extended to allow inclusion of custom substituents on the framework, using either a custom subsearch or a flag to include unsaturated bonds to hetero atoms on the framework. The class *OEBemisMurckoOptions* allows specification of these options. The APIs for the functions *OEGetBemisMurcko* and *OEGetRingLinkerSideChainFragments* have changed to include an options argument.
- The versions of functions *OEGetBemisMurcko* and *OEGetRingLinkerSideChainFragments* that take a single integer argument to specify fragmentation region are now deprecated.

5.12.2 Documentation changes

- Examples of custom Bemis-Murcko fragmentation have been added to the fragmentation documentation in the *Molecule Fragmentation* chapter.
- The *BemisMurckoPerception* example script has been modified to include examples of custom Bemis-Murcko fragmentation, using the flag `-unsatHetero` to include unsaturated bonds to hetero atoms on the framework and the flag `-smartsSubstituents` to include substituents specified by given SMARTS patterns.

5.13 OEMedChem TK 1.1.2

5.13.1 Minor bug fixes

- An internal issue in *OECreateIDStrings* that resulted in hanging processes or crashes for some simple structures has been fixed.

- A minor issue that occurred when using a combination of *OEMatchedPairOptions.IndexHSingleCuts* and *OEMatchedPairOptions.IndexHSites* has been fixed. The *OEMatchedPairOptions.IndexHSingleCuts* option now takes precedence.

5.14 OEMedChem TK 1.1.1

5.14.1 New features

- The following APIs for fast multithreaded generation of matched pair indices, introduced in the 2018.Oct release, have graduated from preliminary API to stable API status:
  - *OECreateMMPIndexFile* function
  - *OECreateMMPIndexStatus* class
  - *OECreateMMPIndexOptions* class

  They are now also available in Python, Java, and C#.

5.14.2 Documentation changes

- The following examples that create a Matched Molecule Pair index were modified to fix an issue with the SD data handling (options `-keepSD`, `-clearSD`, `-allSD`):
  - *Matched Pair analysis generation of a MMP index*
  - *Matched Pair analysis using a multi-process generation of an MMP index*

5.15 OEMedChem TK 1.1.0

5.15.1 Minor bug fixes

- An issue with Matched Pair index generation that occurred when input structures contained reaction mapping information has been fixed.
- Several issues in the probesite-based preliminary API function *OEGetMatchedMolecularPairs* have been fixed.

5.16 OEMedChem TK 1.0.7

5.16.1 New features

- The performance of appending segments of matched pair indices together has been significantly improved over previous versions. This activity is a common use-case with parallel index creation approaches.
- A new *OEMatchedPairAnalyzer.GetMaxMolIdx* method has been added to make identifying the largest molecule index registered to a matched pair index more convenient.
- A new *OEMatchedPairAnalyzer.GetMolecule* method has been added to make retrieving a molecule from the MMP index more convenient and to eliminate the need to maintain a separate OEMolDatabase for this activity.
- A new *OEApplyChEMBL24SolubilityTransforms* function has been added to return modified structures using ChEMBL24 solubility data. The existing *OEApplyChEMBL18SolubilityTransforms* API will be deprecated in future versions.

- A new indexing option, *OEMatchedPairOptions.IndexHSingleCuts*, has been added to limit the H-member matched pair index information to single-cut transformations only. The current *OEMatchedPairOptions.IndexHSites* option will continue to provide H-member index information for all of single-, double-, or triple-cut transformations. *OEMatchedPairOptions.IndexHSingleCuts* takes precedence over *OEMatchedPairOptions.IndexHSites* if both options are provided.

5.16.2 New features (Preliminary API)

- The following preliminary APIs have been added to provide fast multithreaded generation of matched pair indices:
  - *OECreateMMPIndexFile* function
  - *OECreateMMPIndexOptions* option class
  - *OECreateMMPIndexStatus* class
- New *OEReadMMPIndexMolecules* and *OEWriteMMPIndexMolecules* functions have been implemented as preliminary APIs to support multithreaded and parallel matched pair index creation activities.
- New function overloads to *OEMatchedPairGetTransforms* have been added to allow extracting transforms and related matched molecular pairs using 1 or more identified site(s) of substitution on a probe structure.
- New free functions and overloads, *OEGetMatchedMolecularPairs*, have been added to allow extracting related matched molecular pairs using either a molecule ID or a probe structure with one or more identified site(s) of substitution.

5.16.3 API changes

- A public class, *OEMatchedPairTransform*, has been altered to remove unnecessary implementation details. Several of the methods exposed were previously inaccessible or unnecessary for wrapped language use. The simplified API retains the common methods for interrogating the transforms and matched pairs returned from *OEMatchedPairGetTransforms*. This is potentially a breaking API change. For custom code using the removed methods, please contact support@eyesopen.com for details on migration activity that might be required.
- The `OEMedChem::OEIsMCSFragDatabaseFiletype` function has been renamed *OEIsMCSFragDatabaseFileType* to follow **OpenEye Toolkit**'s naming scheme.
- The *OEMCSFragDatabase.CoreToMolecules*, *OEMCSFragDatabase.CoreToMoleculeCount*, *OEMCSFragDatabase.GetMaxCoreMolecule*, *OEMCSFragDatabase.GetMaxCoreMoleculeCount*, and *OEMCSFragDatabase.MoleculeToCores* methods and the *OEMoleculeToCores* function have graduated from preliminary API to stable API status for the 2018.Oct release.

5.16.4 Minor bug fixes

- The *OEGetBemisMurcko* function has been corrected so that chain bonds between rings are no longer returned in the *OERegionType.Ring* role sets.
- *OEMatchedPairGetTransforms* has been fixed and no longer returns unusual matched pair transforms related to group bond-insertion changes.
- A bug that seemed to allow reactions to be added to a matched pair index has been fixed so that reaction input is now explicitly rejected. Attempting to add a reaction to the index will now return a status of *OEMatchedPairIndexStatus.NoStructure*.
- New options were added to *OEMatchedPairIndexStatus* to return additional indexing failure conditions.
- *OEMatchedPairIndexStatus.NoFragmentationBonds* may now be returned from *OEMatchedPairAnalyzer.AddMol* when the input structure has no fragmentation bonds, depending on the H-member indexing option in use.

5.16.5 Python-specific changes

- A new match pair index example has been added to demonstrate the creation of an MMP index via parallel processing using the Python multiprocessing module.

5.16.6 Documentation changes

- The MCS fragment database examples and match pair index examples have been modified to separate the index creation step from database query activities. See the list of currently available examples in the *OEMedChem TK Examples* section.

5.17 OEMedChem TK 1.0.6

5.17.1 New features

- A new status return has been added to indicate when a molecule submitted for indexing by *OEMCSFragDatabase* has no fragmentation bonds (e.g., is a complete ring system without non-ring substituents). The new constant returned is *OEMatchedPairIndexStatus.NoFragmentationBonds*.

5.17.2 Minor bug fixes

- An issue with hydrogen atoms on input molecules used with *OEApplyChEMBL18SolubilityTransforms* has been fixed. This function is now tolerant of explicit hydrogen atoms on the input molecule.

5.18 OEMedChem TK 1.0.5

5.18.1 New features

- *OEWriteMatchedPairAnalyzer* now supports a new option, *OEMatchedPairOptions.ExportHKeys*, which will serialize additional information to reduce the loading time from a *OEReadMatchedPairAnalyzer* call. The saved information results in slightly larger index files (+10-20%) but substantially reduced load times (>10x). *OEMatchedPairOptions.ExportHKeys* is enabled by default.

> **Warning**
>
> The change in the serialized data format requires that any existing saved matched pair analyzer index files be regenerated if they were created using a previous **OpenEye Toolkit** version. Indices serialized from this version cannot be read by older versions.

- *OEReadMatchedPairAnalyzer* now supports a new Boolean argument that controls the regeneration of index information during the read. By default, the regeneration is suppressed and index data read from the file is used.
- A new *OEMatchedPairTransformExtractOptions* option class has been added to control the transformations returned from the *OEMatchedPairGetTransforms* functions. Several new function overloads have been provided that use this new option class.

5.18.2 Minor bug fixes

- Output to `stdout` from within *OEMatchedPairAnalyzer* has been removed.
- Several tetrahedral stereo issues in the transformations extracted by *OEMatchedPairGetTransforms* have been fixed.
- Erroneous warnings had been thrown about duplicate structures when appending matched pair indices if duplicate structures are allowed. This has been fixed.

- An issue with exported matched pair transformations that occurred when search qualifiers were requested and matched pair core atoms contained charged atoms has been fixed. The search qualifiers erroneously duplicated charge information in the output string, causing transformation mappings to fail.

5.18.3 Documentation changes

- Examples have been updated to use `OEDeleteEverythingExceptTheFirstLargestComponent` rather than the deprecated function `OETheFunctionFormerlyKnownAsStripSalts`.

5.19 OEMedChem TK 1.0.4

5.19.1 New features

- A major overhaul of the fragmentation internals of *OEMCSFragDatabase* now allows an arbitrary number of fragmentation cuts. Additionally, an experimental capability for performing limited ring bond fragmentation cuts has been provided. Because of the combinatorial explosion of fragments generated with increased fragmentation cuts, care should be used when determining fragmentation cut limits. The enhanced ability to do arbitrary N-cut fragmentation makes the MCS results from the fragmentation approach much closer to an OEMCSSearch mapping result, with better overall performance than a mapping approach. To specify the fragmentation cut limits, see *OEMCSFragDatabaseOptions.SetFragLimits*, *OEMCSFragDatabaseOptions.GetMinFragLimit*, *OEMCSFragDatabaseOptions.GetMaxFragLimit*, *OEMCSFragDatabaseOptions.SetRingFragLimit*, and *OEMCSFragDatabaseOptions.GetRingFragLimit*.
- A new method, *OEMCSFragDatabase.IsIndexed*, has been added to determine whether a particular molecule index has been registered in the database. Depending on the indexing option settings, a particular molecule may be rejected from indexing. This method allows that condition to be interrogated.
- A new method, *OEMatchedPairAnalyzerOptions.HasIndexableFragmentHeavyAtomRange*, has been added to allow an options object to indicate that a `min-max` range has been specified via *OEMatchedPairAnalyzerOptions.SetIndexableFragmentRange*.
- The *OEMatchedPairAnalyzerOptions* options class now supports the `operator==` comparison to more easily determine the equality of two options objects.
- *OEConfigureMatchedPairIndexOptions* and *OEConfigureMCSFragDatabaseOptions* now support an `-allfrags` option to clear the `min-max` indexing constraint and to allow the indexing of all fragments. Indexing all fragments has indexing size and indexing performance implications for *OEMCSFragDatabase* and *OEMatchedPairAnalyzer* objects.
- The *OEConfigureMCSFragDatabaseOptions* function now supports the newly added `-minfragcuts`, `-maxfragcuts`, and `-ringcuts` options.
- *OEReadMatchedPairAnalyzer* now supports the ability to merge indices generated from segments of a large set of input structures. This capability can be used in a parallel processing approach to increase the throughput of the time-consuming structure indexing phase.

5.19.2 Preliminary Fragmentation API

- New Preliminary API methods have been added to allow the interrogation of intermediate fragmentation output and internal index information from the *OEMCSFragDatabase* object. See *OEMCSFragDatabase.MoleculeToCores*, *OEMCSFragDatabase.CoreToMolecules*, *OEMCSFragDatabase.CoreToMoleculeCount*, *OEMCSFragDatabase.GetMaxCoreMolecule*, and *OEMCSFragDatabase.GetMaxCoreMoleculeCount*.
- A new example has been provided to demonstrate using the new experimental fragment API of *OEMCSFragDatabase*. See `MCSFragOccurrence.py`.

5.19.3 Minor bug fixes

- A bug that resulted in the return of `[R1][R2]` fragments in the Matched Molecular Pair transforms returned by *OEMatchedPairGetTransforms* has been fixed.
- The *OEMCSFragDatabase.GetScores* has been changed to make the API more consistent with OEFPDatabase. The end range argument is now exclusive instead of inclusive when a non-zero value is specified.
- The *OEMCSFragDatabase.GetScores* and *OEMCSFragDatabase.GetSortedScores* methods previously returned an arbitrary *OEMCSMolSimScore* when different identified cores had the same score (in particular for *OEMCSScoreType.AtomCount* scoring). This has been fixed.
- The undocumented internal debugging method `OEMedChem::OEMCSFragDatabase::Dump` has been removed from the public API.

5.19.4 Python-specific changes

- Minor changes to some examples were made to synchronize the behavior and output with other wrapped language examples.

5.19.5 Java-specific changes

- Minor changes to some examples were made to synchronize the behavior and output with other wrapped language examples.

5.19.6 C#-specific changes

- Minor changes to some examples were made to synchronize the behavior and output with other wrapped language examples.

5.19.7 Documentation changes

- New methods have been documented in their respective classes.

5.20 OEMedChem TK 1.0.3

5.20.1 New features

- A new class, *OEMCSFragDatabase*, has been implemented that supports an MCS fragmentation approach for returning Tanimoto and Tversky MCS similarity scores from a query against an indexed set of input structures. This approach is similar to the *OEMatchedPairAnalyzer* and the API is functionally equivalent to the OEFPDatabase API for ease of use. Because the indexing phase of the input structures can be time-consuming, new functions to capture and reload the MCS fragmentation index are provided as *OEReadMCSFragDatabase* and *OEWriteMCSFragDatabase*.
- A new options class, *OEMCSFragDatabaseOptions*, has been added to control the indexing phase of *OEMCSFragDatabase* usage.
- A new constant, *MCSFrags*, has been added to support *OEMCSFragDatabase* serialization and may be returned by *OEGetMatchedPairAnalyzerFileType*.
- Two new functions, *OEConfigureMCSFragDatabaseOptions* and *OESetupMCSFragDatabaseOptions*, have been added to allow configuration of indexing parameters from the command line for *OEMCSFragDatabaseOptions*.
- A new function, *OEMCSFragDatabaseCoreQuery*, has been added to annotate MCS fragment SMILES cores returned from *OEMCSFragDatabase.GetSortedScores* and *OEMCSMolSimScore* with SMARTS qualifiers. This allows more restrictive and meaningful mapping highlights to be applied for depicting similarity hit results.

- The *OEMCSFragDatabase* class now supports user-customization of the similarity score returned by implementing a class derived from *OEMCSSimFuncBase*. Two concrete implementations of scoring functions, *OEMCSTanimotoSim* and *OEMCSTverskySim*, have been added.
- The indexing performance of *OEMatchedPairAnalyzer* has been significantly improved. While this is highly dependent on the input structures and the requested indexing parameters, the provided default indexing settings with a drug-like set of structures show an indexing improvement of ~25%.
- A new overloaded function, `OEIsMatchedPairAnalyzerFileType`, has been added to simplify the validation of matched pair index files. The now unnecessary `OEIsReadableMatchedPairAnalyzer` and `OEIsWriteableMatchedPairAnalyzer` functions have been deprecated.

5.20.2 Documentation changes

- New documentation has been added for the *OEMCSFragDatabase* API and the ancillary support functions, classes, and constants.

5.21 OEMedChem TK 1.0.2

5.21.1 Minor bug fixes

- An issue that could lead to corrupted stereochemistry alpha to fragmentation bonds during Matched Pair analysis has been repaired.
- Minor internal improvements have been made.

5.22 OEMedChem TK 1.0.1

5.22.1 Minor bug fixes

- The `OECreateIDStrings` function has been repaired and will no longer crash.

5.23 OEMedChem TK 1.0.0

5.23.1 Licensing changes

- In preparation for new licensing requirements for **OEMedChem TK**, license checks have been added to *OEMatchedPairAnalyzer* constructor. Visit https://www.eyesopen.com/contact if you need a license for matched pair activities.

5.23.2 OEMedChem functionalities

This is the first official release of **OEMedChem** functionalities, previously released in beta form. An overview of its capabilities is provided below.

5.23.3 Matched molecular pair analysis

A powerful analysis engine and functions support the identification of matched molecular pairs that differ by both heavy-atom and hydrogen-atom substitutions. Additionally, the analyzer supports the ability to be saved and restored to disk or stream. The analyzer includes the following functions:

- *OEMatchedPairAnalyzer*
- `OEMatchedPairApplyTransforms`
- `OEReadMatchedPairAnalyzer`

- *OEWriteMatchedPairAnalyzer*
- *OEMatchedPairAnalyzerOptions*
- *OEConfigureMatchedPairIndexOptions*
- *OESetupMatchedPairIndexOptions*

Functions and classes are provided to allow extraction and inspection of transformations based on matched molecular pairs found during structure indexing as well as the specific matched pairs and data associated with those transformations:

- *OEMatchedPair*
- *OEMatchedPairTransform*
- *OEMatchedPairGetTransforms*

ChEMBL18 solubility data has been indexed with the matched pair analyzer to modify input structures according to the solubility transforms and to annotate modified structures with the net change in solubility data from ChEMBL:

- *OEApplyChEMBL18SolubilityTransforms*

5.23.4 Fragmentation and perception capabilities

The following functions are provided to return perceived regions for fragmentation strategies:

- *OEGetBemisMurcko*
- *OEGetFuncGroupFragments*
- *OEGetRegionTypeName*
- *OEGetRegionType*
- *OEGetRingChainFragments*
- *OEGetRingLinkerSideChainFragments*

5.23.5 Belief theory

Functions are provided to return information for belief theory usage based on [Muchmore-2008]:

- *OECalculateComboBelief*
- *OECalculateETComboBelief*

5.23.6 Similarity

Functions are provided to return the graph-edit similarity used in clustering BROOD hitlists:

- *OECreateIDString*
- *OECreateIDStrings*

5.23.7 Complexity measures

Functions are provided to return individual terms that make up the molecular complexity as defined by [Bertz-1981]:

- *OEDetermineConnectiveComplexity*
- *OEDetermineElementalComplexity*
- *OEDetermineMolecularComplexity*
- *OEDetermineSymmetryComplexity*

- *OENormalMolecularComplexity*

Functions are provided to return a ring complexity measure similar to that defined by [Gasteiger-1979]:

- *OEDetermineRingComplexity*
- *OEGetRingValue*

Functions are provided to return a measure of stereo complexity as defined by [Boda-2007]:

- *OEDetermineStereoComplexity*
- *OENormalStereoComplexity*

A function is provided to return a measure of the total molecular complexity, which is a sum of the provided individual complexity measures:

- *OETotalMolecularComplexity*

5.23.8 Minor bug fixes

- As a result of a minor bug fix in **OEChem TK**, *OEDetermineStereoComplexity*, *OENormalStereoComplexity*, and *OENormalStereoComplexity* functions might return slightly different values for molecules containing transition metals, lanthanides, or actinides.

5.23.9 Documentation changes

- New documentation code snippets for C++, Java, and C# have been added to demonstrate matched pair index creation, serialization, and deserialization.
- Examples have been updated to allow explicit specification of the `mmpMinPairThreshold` arguments for *OEMatchedPairApplyTransforms* and *OEApplyChEMBL18SolubilityTransforms*.
- Several molecular complexity functions that were inadvertently removed from the documentation have been re-added.
- The *Matched Pair analysis and transformations* example has been modified to reduce the amount of output that is typical for a general indexing activity.

5.24 OEMedChem TK 0.9.6

5.24.1 Beta-API changes

- New functions have been added to support matched pair index serialization (see *OEGetMatchedPairAnalyzerFileType*, *OEIsReadableMatchedPairAnalyzer*, *OEIsWriteableMatchedPairAnalyzer*, *OEReadMatchedPairAnalyzer*, *OEWriteMatchedPairAnalyzer*).
- A new option to remove unnecessary index node information during the index export activity is now available (see *OEMatchedPairOptions.ExportCompression*).
- New methods have been added to the analyzer and option classes to allow altering options that do not invalidate an active index (see *OEMatchedPairOptions.ValidAlterOpts*, *OEMatchedPairOptions.ValidIndexOpts*, *OEMatchedPairAnalyzer.ModifyOptions*, *OEMatchedPairAnalyzerOptions.ModifyOptions*).
- A new method has been added to the analyzer options class to support command-line specification of options (see *OEMatchedPairAnalyzerOptions.SetProcessorType*).
- A new argument has been added to the functions that apply matched pair transformations that limits application of transformations only to those that meet or exceed the specified number of matched pairs (see *OEApplyChEMBL18SolubilityTransforms* and *OEMatchedPairApplyTransforms*).

5.24.2 New features

- The matched pair analyzer now supports the ability to be saved and restored to disk or stream. This allows checkpoint backups during a long analysis run or a long analysis run to be restored from disk for subsequent use without the need to re-run the analysis (see *OEReadMatchedPairAnalyzer* and *OEWriteMatchedPairAnalyzer*).
- The ChEMBL18 solubility data has been re-indexed with the latest version of the matched pair analyzer to include hydrogen-member transformations. The result is a larger number of transformations and potentially a larger number of transformed and annotated structures returned from *OEApplyChEMBL18SolubilityTransforms*. For a 10K test set of structures, using *OEMatchedPairContext.Bond0* resulted in an average of three times more output structures and *OEMatchedPairContext.Bond2* produced an average of 4.5 times the number of output structures as the previous release when applying all transformations by overriding the default value of `minMMPThreshold` parameter with `0`. Using the default value (5) for the `minMMPThreshold` parameter, the number of annotated structures from the 10K test set compared to the previous version is approximately the same for the *OEMatchedPairContext.Bond0* context and approximately 25% less structures for the *OEMatchedPairContext.Bond2* context. The number of structures produced is input dataset-dependent, so tuning of the `minMMPThreshold` parameter is highly encouraged.
- New functions to configure matched pair indexing options and to set indexing options from the command line have been added (see *OEConfigureMatchedPairIndexOptions* and *OESetupMatchedPairIndexOptions*). Example scripts have been updated to demonstrate the use of these new functions.
- A new method has been added to return a count of the number of active index nodes (see *OEMatchedPairAnalyzer.NumIndexNodes*).

5.24.3 Minor bug fixes

- Structures returned from the application of matched pair transformations will now have their coordinates cleared. Previously, newly added atoms were simply placed at the origin, possibly affecting stereochemistry and generating odd depiction output.

5.24.4 Documentation fixes

- Documentation of the new serialization support is now available for *OEGetMatchedPairAnalyzerFileType*, *OEIsReadableMatchedPairAnalyzer*, *OEIsWriteableMatchedPairAnalyzer*, *OEReadMatchedPairAnalyzer*, *OEWriteMatchedPairAnalyzer*, *OEMatchedPairAnalyzer.ModifyOptions*, *OEMatchedPairOptions.ExportCompression*, *OEMatchedPairAnalyzerFileType*, *OEMatchedPairOptions.ValidAlterOpts* and *OEMatchedPairOptions.ValidIndexOpts*.
- Documentation of the new command-line configuration support is now available for *OEConfigureMatchedPairIndexOptions* and *OESetupMatchedPairIndexOptions*.

5.25 OEMedChem TK 0.9.5

5.25.1 New features

- The matched pair analyzer now supports the identification of matched molecular pairs that differ by heavy-atom and hydrogen-atom substitutions. This additional indexing activity returns significantly more matched molecular pairs than the previous version. The hydrogen-atom indexing activity has been enabled by default. To perform heavy-atom indexing only, remove the *OEMatchedPairOptions.IndexHSites* option from *OEMatchedPairOptions.Default* and set the desired option(s) explicitly via *OEMatchedPairAnalyzerOptions.SetOptions*.
- The matched pair analyzer now supports single bond insertion differences between matched molecular pairs. Previously, a minimum change of 1-atom was identified in the substituent change between the matched pairs.

5.25.2 Beta-API changes

- *OEMatchedPairOptions.IndexHSites* is a new option that enables hydrogen-atom indexing support to the default indexing activities.
- *OEMatchedPairTransformExtractMode.SuppressNodeSort* is a new option that suppresses the internal sorting of index nodes prior to extraction of transformations and matched pairs for debugging and analysis of indexing problems.
- *OEMatchedPairTransformExtractMode.AddMCSCorrespondence* is a new extraction option that returns the maximum common substructure correspondence between the matched molecular pairs in the form of atom-mapping information applied to the MCS cores.
- *OEMatchedPairTransformExtractMode.AddAttachmentPts* is a new extraction option that requests the addition of Rgroup attachment points on the returned matched molecular pair transforms.

5.25.3 Major bug fixes

- A number of bugs related to the identification of the maximum common substructure for matched molecular pairs have been corrected.
- The ChEMBL solubility data has been re-indexed to capture indexing fixes, potentially impacting the results returned from *OEApplyChEMBL18SolubilityTransforms*.
- An internal index node sort now includes the number of fragmentation cuts in the sort criterion to preferentially return single > double > triple cuts for the largest maximum common substructures.
- A significant refactoring of internal activities has been performed to prepare for additional public API exposure in future versions.

5.26 OEMedChem TK 0.9.4

5.26.1 Beta-API changes

- The *OEMatchedPairAnalyzer.AddMol* return value has been changed. Negative return values no longer indicate a duplicate record ID in the index, but instead return various status values to indicate specific details of an indexing failure.

> **Note**
>
> The interpretation of return values from this method in user-coded scripts may need to be altered.

5.26.2 New features

- The indexing of structures now supports the specification of a zero or positive integer record ID to be used for referencing the input structures. This simplifies cross-referencing to externally maintained or managed data structures. For example, see changes to following:
  - *OEMatchedPairAnalyzer*
  - *OEMatchedPairAnalyzer.AddMol*
  - *OEMatchedPairAnalyzer.IsIndexed*
  - *OEMatchedPairIndexStatus*
  - *OEMatchedPairIndexStatusName*
  - *OEMatchedPairIndexStatusType*

- Prioritization of matched pair index nodes is now performed based on the size of the MCS for the indexed pair(s). This results in a significant reduction of redundant and less interesting transformations extracted from the index. The ChEMBL solubility data was reanalyzed with this optimization and results in approximately 1/3 fewer (redundant) transformations to be applied via *OEApplyChEMBL18SolubilityTransforms*.
- A function has been provided to allow extraction and inspection of transformations based on the matched molecular pairs found during indexing as well as the specific matched pairs and data associated with those transformations. For example, see changes to the following:
  - *OEMatchedPairGetTransforms*
  - *OEMatchedPairTransform*
  - *OEMatchedPair*
- A new overload of *OEMatchedPairGetTransforms* now provides the ability to retrieve related transformations and matched molecular pairs based on a probe molecule. The probe molecule is fragmented based on the analyzer indexing strategy; matched pair index nodes that have common cores to the provided probe are used to retrieve related matched pair transformations.

5.26.3 Major bug fixes

- Stereoisomers are now indexed properly by *OEMatchedPairAnalyzer*. Previously, stereoisomers were incorrectly considered duplicates and were rejected during indexing.
- The perceived regions from *OEGetRingChainFragments* and *OEGetRingLinkerSideChainFragments* are based on the heavy atoms in the molecule. If the molecule contains explicit hydrogen atoms, those atoms will be returned in the same perception set(s) as their alpha-heavy atom neighbors. The documentation for these functions has been altered to reflect the new behavior as well as noting the requirement of the `OEFindRingAtomsAndBonds` perception activity prior to use of these functions.
- A crash in *OETotalMolecularComplexity* for highly bridged ring systems has been fixed. An issue that caused *OEDetermineSymmetryComplexity* to crash when the required call to `OEPerceiveChiral` had not been performed on the input molecule has also been fixed.
- The complexity computation functions have been altered to verify that the required perceptions have been performed. Due to the `const` arguments to these functions, the perceptions cannot be performed internally and will now throw an error if the perception state is out of date for the provided input molecule. The functions affected are:
  - *OEDetermineRingComplexity*
  - *OEDetermineSymmetryComplexity*
  - *OEDetermineStereoComplexity*
  - *OENormalStereoComplexity*
  - *OETotalMolecularComplexity*

5.26.4 Documentation fixes

- New examples have been added to demonstrate the new API functions.
- New examples have been added for C++, Python, Java, and C#.
- Minor naming changes to constants for the beta matched pair API were made for consistency.
- Minor fixes have been made to the complexity function documentation to reflect the correct API points.

5.27 OEMedChem TK 0.9.3

5.27.1 New features

- New experimental classes and a new function for Matched Molecular Pair analysis were added. These APIs are experimental and they will likely change in future versions.
  - *OEMatchedPairAnalyzer*
  - *OEMatchedPairAnalyzerOptions*
  - *OEMatchedPairApplyTransforms*
- New functions were added to return type to name conversions for new parameterized constants.
  - *OEMatchedPairContextName*
  - *OEMatchedPairContextType*
  - *OEMatchedPairProcessorName*
  - *OEMatchedPairProcessorType*
  - *OEMatchedPairOptionsName*
  - *OEMatchedPairOptionsType*
- New functions were added to return information for belief theory usage based on [Muchmore-2008].
  - *OECalculateComboBelief*
  - *OECalculateETComboBelief*
- New functions were added based on the implementation in *OEGetRingLinkerSideChainFragments* to return annotated Bemis Murcko regions and types from the perception as defined in [Bemis-1996] .
  - *OEGetBemisMurcko*
  - *OEGetRegionType*
  - *OEGetRegionTypeName*
- New functions were added to return the graph-edit similarity used in clustering BROOD hitlists.
  - *OECreateIDString*
  - *OECreateIDStrings*
- New functions were added which return individual terms that make up the Molecular Complexity as defined by [Bertz-1981].
  - *OEDetermineConnectiveComplexity*
  - *OEDetermineElementalComplexity*
  - *OEDetermineMolecularComplexity*
  - *OEDetermineSymmetryComplexity*
  - *OENormalMolecularComplexity*
- New functions were added for a ring complexity measure similar to that defined by [Gasteiger-1979].
  - *OEDetermineRingComplexity*
  - *OEGetRingValue*
- New functions were added which return a measure of stereo complexity as defined by [Boda-2007].

  - *OEDetermineStereoComplexity*
  - *OENormalStereoComplexity*

- A new function was added to return a measure of the total molecular complexity which is a sum of individual complexity measures above.
  - *OETotalMolecularComplexity*

5.27.2 Known bugs

- The beta Matched Pair analyzer api internally removes stereochemistry from the internally indexed structures. It should more properly treat the presence of stereochemistry for duplicate checking and indexing equivalently. This issue will be repaired in the next release.
  - *OEMatchedPairAnalyzer*

5.27.3 Documentation fixes

- Added documentation for
  - *OEMedChemGetArch*
  - *OEMedChemGetLicensee*
  - *OEMedChemGetPlatform*
  - *OEMedChemGetRelease*
  - *OEMedChemGetSite*
  - *OEMedChemGetVersion*
  - *OEMedChemIsLicensed*

5.28 OEMedChem TK 0.9.2

5.28.1 Minor bug fixes

- *OEGetRingChainFragments*, *OEGetRingLinkerSideChainFragments*, *OEGetFuncGroupFragments* will now `OEThrow.Error` whenever the molecule passed in does not have `RingAtomsAndBonds` perceived.

5.29 OEMedChem TK 0.9.1

5.29.1 New features

- Several new experimental APIs added. These APIs are not officially supported yet and they will likely change in a future revision.

5.30 OEMedChem TK 0.9.0

5.30.1 New features

Providing the following fragmentation functions:

- *OEGetRingChainFragments*
- *OEGetRingLinkerSideChainFragments*
- *OEGetFuncGroupFragments*

CHAPTER

SIX

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

SEVEN

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

EIGHT

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.

8.1 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.

8.2 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.

8.3 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.

8.4 Publications for Bibliographies

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

8.4.1 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

8.4.2 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

8.4.3 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

8.4.4 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

8.4.5 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

8.5 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

NINE

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

9.1 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.

9.1.1 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

9.1.2 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)

```
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
contain the covered work, unless you entered into that arrangement,
```

(continues on next page)

(continued from previous page)

```
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
permissions.  However, no additional obligations are imposed on any
```

(continues on next page)

(continued from previous page)

```
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>

    This program is free software: you can redistribute it and/or modify
```

(continues on next page)

(continued from previous page)

```
    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

A

AddConstMol
    OEMedChem::OEMCSFragDatabase, 68
AddIndexedMol
    OEMedChem::OECreateMMPIndexStatus, 66
AddMatchedPair
    OEMedChem::OEMatchedPairTransform, 86
AddMol
    OEMedChem::OEMatchedPairAnalyzer, 80
    OEMedChem::OEMCSFragDatabase, 68
AddSDData
    OEMedChem::OEMatchedPair, 84
AddUnindexedMol
    OEMedChem::OECreateMMPIndexStatus, 66

B

bemismurckoperception.cs
    Example Code, 14

C

chemblsolubility.cs
    Example Code, 18
ClearIndexableFragmentRange
    OEMedChem::OEMatchedPairAnalyzerOptions, 82
    OEMedChem::OEMCSFragDatabaseOptions, 73
ClearIndexedMols
    OEMedChem::OEMCSFragDatabase, 69
ClearSDData
    OEMedChem::OEMatchedPair, 85
Constructors
    OEMedChem::OEBemisMurckoOptions, 63
    OEMedChem::OECreateMMPIndexOptions, 64
    OEMedChem::OECreateMMPIndexStatus, 66
    OEMedChem::OEMatchedPair, 84
    OEMedChem::OEMatchedPairAnalyzer, 79
    OEMedChem::OEMatchedPairAnalyzerOptions, 82
    OEMedChem::OEMatchedPairTransform, 86
    OEMedChem::OEMatchedPairTransformExtractOptions, 89
    OEMedChem::OEMCSFragDatabase, 68
    OEMedChem::OEMCSFragDatabaseOptions, 73
    OEMedChem::OEMCSMolSimScore, 76
    OEMedChem::OEMCSSimFuncBase, 77
    OEMedChem::OEMCSTanimotoSim, 78
    OEMedChem::OEMCSTverskySim, 78
CoreToMoleculeCount
    OEMedChem::OEMCSFragDatabase, 69
CoreToMolecules
    OEMedChem::OEMCSFragDatabase, 69
CreateCopy
    OEMedChem::OEBemisMurckoOptions, 63
    OEMedChem::OEMatchedPair, 85
    OEMedChem::OEMatchedPairTransform, 87
    OEMedChem::OEMCSSimFuncBase, 77
    OEMedChem::OEMCSTanimotoSim, 78
    OEMedChem::OEMCSTverskySim, 79
createmcsfragdatabase.cs
    Example Code, 53
createmmpindex.cs
    Example Code, 22
createmmpindexthreaded.cs
    Example Code, 30

E

Example Code
    bemismurckoperception.cs, 14
    chemblsolubility.cs, 18
    createmcsfragdatabase.cs, 53
    createmmpindex.cs, 22
    createmmpindexthreaded.cs, 30
    matchedpairtransform.cs, 35
    matchedpairtransformlist.cs, 40
    matchedpairtransformprobe.cs, 46
    mcsfragdatabase.cs, 57
    oemedcheminfo.cs, 62

F

FromIndex
    OEMedChem::OEMatchedPair, 85
FromSmiles
    OEMedChem::OEMatchedPair, 85

G

GetAlpha
OEMedChem::OEMCSTverskySim, 79
GetAttachmentPoints
OEMedChem::OEMatchedPairTransformExtractOptions, 89
GetBeta
OEMedChem::OEMCSTverskySim, 79
GetContext
OEMedChem::OEMatchedPairTransformExtractOptions, 89
GetDataTags
OEMedChem::OEMatchedPair, 85
GetDirection
OEMedChem::OEMatchedPairTransformExtractOptions, 90
GetFragmentationLimit
OEMedChem::OEMatchedPairAnalyzerOptions, 82
OEMedChem::OEMCSFragDatabaseOptions, 73
GetFromSDData
OEMedChem::OEMatchedPair, 85
GetIdx
OEMedChem::OEMCSMolSimScore, 76
GetIndexableFragmentRangeMax
OEMedChem::OEMatchedPairAnalyzerOptions, 83
OEMedChem::OEMCSFragDatabaseOptions, 74
GetIndexableFragmentRangeMin
OEMedChem::OEMatchedPairAnalyzerOptions, 83
OEMedChem::OEMCSFragDatabaseOptions, 74
GetIsotopes
OEMedChem::OEMatchedPairTransformExtractOptions, 90
GetMatchedPairContext
OEMedChem::OEMatchedPairTransform, 87
GetMatchedPairData
OEMedChem::OEMatchedPairTransformExtractOptions, 90
GetMatchedPairs
OEMedChem::OEMatchedPairTransform, 87
OEMedChem::OEMatchedPairTransformExtractOptions, 90
GetMaxAtomFilter
OEMedChem::OEMatchedPairAnalyzerOptions, 83
OEMedChem::OEMCSFragDatabaseOptions, 74
GetMaxCoreMolecule
OEMedChem::OEMCSFragDatabase, 69
GetMaxCoreMoleculeCount
OEMedChem::OEMCSFragDatabase, 69
GetMaxFragLimit
OEMedChem::OEMCSFragDatabaseOptions, 74
GetMaxMMPLimit
OEMedChem::OEMatchedPairTransformExtractOptions, 90
GetMaxMolIdx
OEMedChem::OEMatchedPairAnalyzer, 80
OEMedChem::OEMCSFragDatabase, 69
GetMaxRecord
OEMedChem::OECreateMMPIndexOptions, 65
GetMaxSubstituentLimit
OEMedChem::OEMatchedPairTransformExtractOptions, 91
GetMaxSubstituentLimitSkip
OEMedChem::OEMatchedPairTransformExtractOptions, 91
GetMCSCore
OEMedChem::OEMCSMolSimScore, 76
GetMCSCorrespondence
OEMedChem::OEMatchedPairTransformExtractOptions, 90
GetMCSSimTypeString
OEMedChem::OEMCSSimFuncBase, 77
OEMedChem::OEMCSTanimotoSim, 78
OEMedChem::OEMCSTverskySim, 79
GetMinFragLimit
OEMedChem::OEMCSFragDatabaseOptions, 74
GetMinMMPLimit
OEMedChem::OEMatchedPairTransformExtractOptions, 91
GetMinSubstituentLimit
OEMedChem::OEMatchedPairTransformExtractOptions, 91
GetMMPOptions
OEMedChem::OECreateMMPIndexOptions, 65
GetMolecule
OEMedChem::OEMatchedPairAnalyzer, 80
OEMedChem::OEMatchedPairTransform, 87
GetMoleculeTitle
OEMedChem::OEMatchedPairTransform, 87
GetNumMatchedPairs
OEMedChem::OECreateMMPIndexStatus, 67
GetNumMols
OEMedChem::OECreateMMPIndexStatus, 67
GetNumThreads
OEMedChem::OECreateMMPIndexOptions, 65
GetOptions
OEMedChem::OEMatchedPairAnalyzer, 80
OEMedChem::OEMatchedPairAnalyzerOptions, 83
OEMedChem::OEMatchedPairTransformExtractOptions, 91
OEMedChem::OEMCSFragDatabase, 69
OEMedChem::OEMCSFragDatabaseOptions, 74
GetProcessorID
OEMedChem::OEMatchedPairAnalyzer, 80

GetProcessorType
    OEMedChem::OEMatchedPairAnalyzer, 81
    OEMedChem::OEMatchedPairAnalyzerOptions, 83
GetRegionType
    OEMedChem::OEBemisMurckoOptions, 63
GetRingFragLimit
    OEMedChem::OEMCSFragDatabaseOptions, 74
GetScore
    OEMedChem::OEMCSMolSimScore, 76
GetScores
    OEMedChem::OEMCSFragDatabase, 69
GetSDData
    OEMedChem::OEMatchedPair, 85
GetSMARTS
    OEMedChem::OEMatchedPairTransformExtractOptions, 92
GetSortedScores
    OEMedChem::OEMCSFragDatabase, 70
GetSubstituentSearch
    OEMedChem::OEBemisMurckoOptions, 63
GetToSDData
    OEMedChem::OEMatchedPair, 85
GetTotalMols
    OEMedChem::OECreateMMPIndexStatus, 67
GetTransform
    OEMedChem::OEMatchedPairTransform, 88
GetUnindexedMols
    OEMedChem::OECreateMMPIndexStatus, 67
GetUnsaturatedHeteroBonds
    OEMedChem::OEBemisMurckoOptions, 64
GetVerbose
    OEMedChem::OECreateMMPIndexOptions, 65

H

HasIndexableFragmentHeavyAtomRange
    OEMedChem::OEMatchedPairAnalyzerOptions, 83
    OEMedChem::OEMCSFragDatabaseOptions, 74
HasMatchedPair
    OEMedChem::OEMatchedPairTransform, 88
HasSDData
    OEMedChem::OEMatchedPair, 86

I

Invert
    OEMedChem::OEMatchedPairTransform, 88
IsIndexed
    OEMedChem::OEMatchedPairAnalyzer, 81
    OEMedChem::OEMCSFragDatabase, 71

M

matchedpairtransform.cs
    Example Code, 35
matchedpairtransformlist.cs
    Example Code, 40
matchedpairtransformprobe.cs
    Example Code, 46
mcsfragdatabase.cs
    Example Code, 57
ModifyOptions
    OEMedChem::OEMatchedPairAnalyzer, 81
    OEMedChem::OEMatchedPairAnalyzerOptions, 83
MoleculeToCores
    OEMedChem::OEMCSFragDatabase, 71

N

NumDataTags
    OEMedChem::OEMatchedPair, 86
NumFragments
    OEMedChem::OEMCSFragDatabase, 72
NumIndexNodes
    OEMedChem::OEMatchedPairAnalyzer, 81
NumMatchedPairs
    OEMedChem::OEMatchedPairAnalyzer, 81
    OEMedChem::OEMatchedPairTransform, 88
NumMols
    OEMedChem::OEMatchedPairAnalyzer, 81
    OEMedChem::OEMCSFragDatabase, 72

O

OEMedChem::OEApplyChEMBL18SolubilityTransforms, 111
OEMedChem::OEApplyChEMBL24SolubilityTransforms, 112
OEMedChem::OEBemisMurckoOptions, 63
    Constructors, 63
    CreateCopy, 63
    GetRegionType, 63
    GetSubstituentSearch, 63
    GetUnsaturatedHeteroBonds, 64
    SetRegionType, 64
    SetSubstituentSearch, 64
    SetUnsaturatedHeteroBonds, 64
    Validate, 64
OEMedChem::OECalculateComboBelief, 114
OEMedChem::OECalculateETComboBelief, 114
OEMedChem::OEConfigureMatchedPairIndexOptions, 114
OEMedChem::OEConfigureMCSFragDatabaseOptions, 115
OEMedChem::OECreateIDString, 115
OEMedChem::OECreateIDStrings, 115
OEMedChem::OECreateMMPIndexFile, 116
OEMedChem::OECreateMMPIndexOptions, 64
    Constructors, 64
    GetMaxRecord, 65

GetMMPOptions, 65
GetNumThreads, 65
GetVerbose, 65
SetMaxRecord, 65
SetNumThreads, 65
SetTracer, 66
SetVerbose, 66
OEMedChem::OECreateMMPIndexStatus, 66
AddIndexedMol, 66
AddUnindexedMol, 66
Constructors, 66
GetNumMatchedPairs, 67
GetNumMols, 67
GetTotalMols, 67
GetUnindexedMols, 67
SetNumMatchedPairs, 67
SetNumMols, 67
SetUnindexedMols, 67
SetValid, 67
OEMedChem::OEDetermineConnectiveComplexity, 116
OEMedChem::OEDetermineElementalComplexity, 116
OEMedChem::OEDetermineMolecularComplexity, 116
OEMedChem::OEDetermineRingComplexity, 117
OEMedChem::OEDetermineStereoComplexity, 117
OEMedChem::OEDetermineSymmetryComplexity, 118
OEMedChem::OEGetBemisMurcko, 118
OEMedChem::OEGetFuncGroupFragments, 119
OEMedChem::OEGetMatchedMolecularPairs, 120
OEMedChem::OEGetMatchedPairAnalyzerFileType, 121
OEMedChem::OEGetRegionType, 121
OEMedChem::OEGetRegionTypeName, 121
OEMedChem::OEGetRingChainFragments, 121
OEMedChem::OEGetRingLinkerSideChainFragments, 122
OEMedChem::OEGetRingValue, 124
OEMedChem::OEIsMatchedPairAnalyzerFileType, 124
OEMedChem::OEIsMCSFragDatabaseFileType, 124
OEMedChem::OEIsReadableMatchedPairAnalyzer, 125
OEMedChem::OEIsWriteableMatchedPairAnalyzer, 125
OEMedChem::OEMatchedPair, 84
AddSDData, 84
ClearSDData, 85
Constructors, 84
CreateCopy, 85
FromIndex, 85
FromSmiles, 85
GetDataTags, 85
GetFromSDData, 85
GetSDData, 85
GetToSDData, 85
HasSDData, 86
NumDataTags, 86
ToIndex, 86
ToSmiles, 86
OEMedChem::OEMatchedPairAnalyzer, 79
AddMol, 80
Constructors, 79
GetMaxMolIdx, 80
GetMolecule, 80
GetOptions, 80
GetProcessorID, 80
GetProcessorType, 81
IsIndexed, 81
ModifyOptions, 81
NumIndexNodes, 81
NumMatchedPairs, 81
NumMols, 81
OEMedChem::OEMatchedPairAnalyzerFileType, 99
OEMedChem::OEMatchedPairAnalyzerFileType::MCSFrags, 99
OEMedChem::OEMatchedPairAnalyzerFileType::MMPIndex, 99
OEMedChem::OEMatchedPairAnalyzerFileType::UNDEFINED, 99
OEMedChem::OEMatchedPairAnalyzerOptions, 82
ClearIndexableFragmentRange, 82
Constructors, 82
GetFragmentationLimit, 82
GetIndexableFragmentRangeMax, 83
GetIndexableFragmentRangeMin, 83
GetMaxAtomFilter, 83
GetOptions, 83
GetProcessorType, 83
HasIndexableFragmentHeavyAtomRange, 83
ModifyOptions, 83
SetFragmentationLimit, 83
SetIndexableFragmentRange, 84
SetMaxAtomFilter, 84
SetOptions, 84
SetProcessorType, 84
OEMedChem::OEMatchedPairApplyTransforms, 126
OEMedChem::OEMatchedPairContext, 99
OEMedChem::OEMatchedPairContext::AllBonds, 100
OEMedChem::OEMatchedPairContext::Bond0, 100
OEMedChem::OEMatchedPairContext::Bond1, 100
OEMedChem::OEMatchedPairContext::Bond2, 100
OEMedChem::OEMatchedPairContext::Bond3, 100
OEMedChem::OEMatchedPairContext::Default, 100
OEMedChem::OEMatchedPairContext::Undefined, 99

OEMedChem::OEMatchedPairContextName, 126
OEMedChem::OEMatchedPairContextType, 126
OEMedChem::OEMatchedPairGetTransforms, 126
OEMedChem::OEMatchedPairIndexSetup, 100
OEMedChem::OEMatchedPairIndexSetup::All, 100
OEMedChem::OEMatchedPairIndexSetup::BondFragLimit, 101
OEMedChem::OEMatchedPairIndexSetup::FragmentationMethod, 102
OEMedChem::OEMatchedPairIndexSetup::FragMinMaxChange, 101
OEMedChem::OEMatchedPairIndexSetup::HeavyAtomLimit, 102
OEMedChem::OEMatchedPairIndexSetup::IndexHydrogensMethod, 103
OEMedChem::OEMatchedPairIndexSetup::IndexHydrogensOnly, 103
OEMedChem::OEMatchedPairIndexSetup::ProcessorType, 104
OEMedChem::OEMatchedPairIndexSetup::UniquesOnly, 104
OEMedChem::OEMatchedPairIndexStatus, 105
OEMedChem::OEMatchedPairIndexStatus::BinaryIOError, 105
OEMedChem::OEMatchedPairIndexStatus::CopyMolError, 105
OEMedChem::OEMatchedPairIndexStatus::Deactivated, 105
OEMedChem::OEMatchedPairIndexStatus::DuplicateRecordID, 105
OEMedChem::OEMatchedPairIndexStatus::DuplicateStructure, 105
OEMedChem::OEMatchedPairIndexStatus::FragmentationError, 106
OEMedChem::OEMatchedPairIndexStatus::FragmentFilter, 105
OEMedChem::OEMatchedPairIndexStatus::HeavyAtomFilter, 106
OEMedChem::OEMatchedPairIndexStatus::IncompatibleOptions, 106
OEMedChem::OEMatchedPairIndexStatus::NoError, 106
OEMedChem::OEMatchedPairIndexStatus::NoFragmentationMethod, 106
OEMedChem::OEMatchedPairIndexStatus::NoStructure, 106
OEMedChem::OEMatchedPairIndexStatus::ProcessorFilter, 106
OEMedChem::OEMatchedPairIndexStatus::ProcessorFilter, 107
OEMedChem::OEMatchedPairIndexStatus::Unspecified, 107
OEMedChem::OEMatchedPairIndexStatusName, 128
OEMedChem::OEMatchedPairIndexStatusType, 128
OEMedChem::OEMatchedPairOptions, 107
OEMedChem::OEMatchedPairOptions::AllCuts, 107
OEMedChem::OEMatchedPairOptions::ComboCuts, 107
OEMedChem::OEMatchedPairOptions::Default, 107
OEMedChem::OEMatchedPairOptions::DefaultFragIndex, 107
OEMedChem::OEMatchedPairOptions::DoubleCuts, 107
OEMedChem::OEMatchedPairOptions::ExportCompression, 107
OEMedChem::OEMatchedPairOptions::ExportHKeys, 108
OEMedChem::OEMatchedPairOptions::IndexHSingleCuts, 108
OEMedChem::OEMatchedPairOptions::IndexHSites, 108
OEMedChem::OEMatchedPairOptions::SingleCuts, 108
OEMedChem::OEMatchedPairOptions::TripleCuts, 108
OEMedChem::OEMatchedPairOptions::UniquesOnly, 108
OEMedChem::OEMatchedPairOptions::ValidAlterOpts, 108
OEMedChem::OEMatchedPairOptions::ValidFragIndexOpts, 108
OEMedChem::OEMatchedPairOptions::ValidIndexOpts, 109
OEMedChem::OEMatchedPairOptions::ValidOptions, 109
OEMedChem::OEMatchedPairOptionsName, 128
OEMedChem::OEMatchedPairOptionsType, 128
OEMedChem::OEMatchedPairProcessor, 109
OEMedChem::OEMatchedPairProcessor::BemisMurcko, 109
OEMedChem::OEMatchedPairProcessor::Custom, 109
OEMedChem::OEMatchedPairProcessor::Default, 109
OEMedChem::OEMatchedPairProcessor::HussainRea, 109
OEMedChem::OEMatchedPairProcessor::Undefined, 109
OEMedChem::OEMatchedPairProcessorName, 128
OEMedChem::OEMatchedPairProcessorType, 128
OEMedChem::OEMatchedPairTransform, 86
AddMatchedPair, 86
Constructors, 86
CreateCopy, 87
GetMatchedPairContext, 87
GetMatchedPairs, 87
GetMolecule, 87
GetMoleculeTitle, 87

GetTransform, 88
HasMatchedPair, 88
Invert, 88
NumMatchedPairs, 88
SetMoleculeTitle, 88
SetTransform, 88
OEMedChem::OEMatchedPairTransformExtractMode, 109
OEMedChem::OEMatchedPairTransformExtractMode::AddAttachmentPoints, 110
OEMedChem::OEMatchedPairTransformExtractMode::AddMCSCorrespondence, 110
OEMedChem::OEMatchedPairTransformExtractMode::Backward, 110
OEMedChem::OEMatchedPairTransformExtractMode::Default, 109
OEMedChem::OEMatchedPairTransformExtractMode::Forward, 110
OEMedChem::OEMatchedPairTransformExtractMode::NoIsotopes, 110
OEMedChem::OEMatchedPairTransformExtractMode::NoMatchedPairData, 110
OEMedChem::OEMatchedPairTransformExtractMode::NoMatchedPairs, 110
OEMedChem::OEMatchedPairTransformExtractMode::NoSMARTS, 110
OEMedChem::OEMatchedPairTransformExtractMode::SortedCanSmiNode, 110
OEMedChem::OEMatchedPairTransformExtractMode::SuppressNoMatch, 110
OEMedChem::OEMatchedPairTransformExtractMode::SuppressNoMatchSort, 111
OEMedChem::OEMatchedPairTransformExtractMode::Unspecified, 109
OEMedChem::OEMatchedPairTransformExtractOptions, 89
Constructors, 89
GetAttachmentPoints, 89
GetContext, 89
GetDirection, 90
GetIsotopes, 90
GetMatchedPairData, 90
GetMatchedPairs, 90
GetMaxMMPLimit, 90
GetMaxSubstituentLimit, 91
GetMaxSubstituentLimitSkip, 91
GetMCSCorrespondence, 90
GetMinMMPLimit, 91
GetMinSubstituentLimit, 91
GetOptions, 91
GetSMARTS, 92
SetAttachmentPoints, 92
SetContext, 92
SetDirection, 92
SetIsotopes, 92
SetMatchedPairData, 93
SetMatchedPairs, 93
SetMaxSubstituentLimitSkip, 93
SetMCSCorrespondence, 92
SetMMPLimit, 93
SetOptions, 93
SetSMARTS, 94
SetSubstituentLimit, 93
OEMedChem::OEMCSFragDatabase, 68
AddMCSCorrespondence, 68
AddMol, 68
ClearIndexedMols, 69
Constructors, 68
CoreToMoleculeCount, 69
CoreToMolecules, 69
GetMaxCoreMolecule, 69
GetMaxCoreMoleculeCount, 69
GetMolIdx, 69
GetOptions, 69
GetSortedPairData, 69
GetSortedScores, 70
HasIndexedMols, 71
MoleculeToCores, 71
NumFragments, 72
NumMols, 72
PreProcessMol, 73
OEMedChem::OEMCSFragDatabaseCoreQuery, 128
OEMedChem::OEMCSFragDatabaseOptions, 73
ClearIndexableFragmentRange, 73
ClearMaxAtomFilter, 73
GetFragmentationLimit, 73
GetIndexableFragmentRangeMax, 74
GetIndexableFragmentRangeMin, 74
GetMaxAtomFilter, 74
GetMaxFragLimit, 74
GetMinFragLimit, 74
GetOptions, 74
GetRingFragLimit, 74
HasIndexableFragmentHeavyAtomRange, 74
SetFragLimits, 75
SetFragmentationLimit, 75
SetIndexableFragmentRange, 75
SetMaxAtomFilter, 75
SetOptions, 75
SetRingFragLimit, 75
OEMedChem::OEMCSFragDatabaseSetup, 94
OEMedChem::OEMCSFragDatabaseSetup::All, 94
OEMedChem::OEMCSFragDatabaseSetup::AllFrags, 94
OEMedChem::OEMCSFragDatabaseSetup::BondFragLimit, 95
OEMedChem::OEMCSFragDatabaseSetup::FragCutsRange, 95

OEMedChem::OEMCSFragDatabaseSetup::FragmentationMethod, 97
OEMedChem::OEMCSFragDatabaseSetup::FragMinMaxRange, 96
OEMedChem::OEMCSFragDatabaseSetup::HeavyAtomLimit, 97
OEMedChem::OEMCSFragDatabaseSetup::IndexHydrogenSmarts, 98
OEMedChem::OEMCSFragDatabaseSetup::RingCuts, 98
OEMedChem::OEMCSMolSimScore, 76
    Constructors, 76
    GetIdx, 76
    GetMCSCore, 76
    GetScore, 76
    Set, 76
OEMedChem::OEMCSScoreType, 99
OEMedChem::OEMCSScoreType::AtomCount, 99
OEMedChem::OEMCSScoreType::BondCount, 99
OEMedChem::OEMCSScoreType::Default, 99
OEMedChem::OEMCSScoreType::Undefined, 99
OEMedChem::OEMCSSimFuncBase, 77
    Constructors, 77
    CreateCopy, 77
    GetMCSSimTypeString, 77
    operator(), 77
OEMedChem::OEMCSTanimotoSim, 77
    Constructors, 78
    CreateCopy, 78
    GetMCSSimTypeString, 78
    operator(), 78
OEMedChem::OEMCSTverskySim, 78
    Constructors, 78
    CreateCopy, 79
    GetAlpha, 79
    GetBeta, 79
    GetMCSSimTypeString, 79
    operator(), 79
OEMedChem::OEMedChemGetArch, 129
OEMedChem::OEMedChemGetLicensee, 129
OEMedChem::OEMedChemGetPlatform, 129
OEMedChem::OEMedChemGetRelease, 129
OEMedChem::OEMedChemGetSite, 129
OEMedChem::OEMedChemGetVersion, 129
OEMedChem::OEMedChemIsLicensed, 130
OEMedChem::OEMoleculeToCores, 130
OEMedChem::OENormalMolecularComplexity, 131
OEMedChem::OENormalStereoComplexity, 131
OEMedChem::OEReadMatchedPairAnalyzer, 131
OEMedChem::OEReadMCSFragDatabase, 132
OEMedChem::OEReadMMPIndexMolecules, 132
OEMedChem::OERegionType, 111
OEMedChem::OERegionType::All, 111
OEMedChem::OERegionType::Framework, 111
OEMedChem::OERegionType::Linker, 111
OEMedChem::OERegionType::Ring, 111
OEMedChem::OERegionType::Sidechain, 111
OEMedChem::OESetupMatchedPairIndexOptions, 133
OEMedChem::OESetupMCSFragDatabaseOptions, 133
OEMedChem::OETotalMolecularComplexity, 133
OEMedChem::OEWriteMatchedPairAnalyzer, 134
OEMedChem::OEWriteMCSFragDatabase, 135
OEMedChem::OEWriteMMPIndexMolecules, 135
oemedcheminfo.cs
    Example Code, 62
operator()
    OEMedChem::OEMCSSimFuncBase, 77
    OEMedChem::OEMCSTanimotoSim, 78
    OEMedChem::OEMCSTverskySim, 79

P

ProcessMol
    OEMedChem::OEMCSFragDatabase, 73

S

Set
    OEMedChem::OEMCSMolSimScore, 76
SetAttachmentPoints
    OEMedChem::OEMatchedPairTransformExtractOptions, 92
SetContext
    OEMedChem::OEMatchedPairTransformExtractOptions, 92
SetDirection
    OEMedChem::OEMatchedPairTransformExtractOptions, 92
SetFragLimits
    OEMedChem::OEMCSFragDatabaseOptions, 75
SetFragmentationLimit
    OEMedChem::OEMatchedPairAnalyzerOptions, 83
    OEMedChem::OEMCSFragDatabaseOptions, 75
SetIndexableFragmentRange
    OEMedChem::OEMatchedPairAnalyzerOptions, 84
    OEMedChem::OEMCSFragDatabaseOptions, 75
SetIsotopes
    OEMedChem::OEMatchedPairTransformExtractOptions, 92
SetMatchedPairData
    OEMedChem::OEMatchedPairTransformExtractOptions, 93
SetMatchedPairs
    OEMedChem::OEMatchedPairTransformExtractOptions, 93
SetMaxAtomFilter

OEMedChem::OEMatchedPairAnalyzerOptions, 84
OEMedChem::OEMCSFragDatabaseOptions, 75
SetMaxRecord
OEMedChem::OECreateMMPIndexOptions, 65
SetMaxSubstituentLimitSkip
OEMedChem::OEMatchedPairTransformExtractOptions, 93
SetMCSCorrespondence
OEMedChem::OEMatchedPairTransformExtractOptions, 92
SetMMPLimit
OEMedChem::OEMatchedPairTransformExtractOptions, 93
SetMoleculeTitle
OEMedChem::OEMatchedPairTransform, 88
SetNumMatchedPairs
OEMedChem::OECreateMMPIndexStatus, 67
SetNumMols
OEMedChem::OECreateMMPIndexStatus, 67
SetNumThreads
OEMedChem::OECreateMMPIndexOptions, 65
SetOptions
OEMedChem::OEMatchedPairAnalyzerOptions, 84
OEMedChem::OEMatchedPairTransformExtractOptions, 93
OEMedChem::OEMCSFragDatabaseOptions, 75
SetProcessorType
OEMedChem::OEMatchedPairAnalyzerOptions, 84
SetRegionType
OEMedChem::OEBemisMurckoOptions, 64
SetRingFragLimit
OEMedChem::OEMCSFragDatabaseOptions, 75
SetSMARTS
OEMedChem::OEMatchedPairTransformExtractOptions, 94
SetSubstituentLimit
OEMedChem::OEMatchedPairTransformExtractOptions, 93
SetSubstituentSearch
OEMedChem::OEBemisMurckoOptions, 64
SetTracer
OEMedChem::OECreateMMPIndexOptions, 66
SetTransform
OEMedChem::OEMatchedPairTransform, 88
SetUnindexedMols
OEMedChem::OECreateMMPIndexStatus, 67
SetUnsaturatedHeteroBonds
OEMedChem::OEBemisMurckoOptions, 64
SetValid
OEMedChem::OECreateMMPIndexStatus, 67
SetVerbose
OEMedChem::OECreateMMPIndexOptions, 66

T

ToIndex
OEMedChem::OEMatchedPair, 86
ToSmiles
OEMedChem::OEMatchedPair, 86

V

Validate
OEMedChem::OEBemisMurckoOptions, 64