

OpenEye
SCIENTIFIC

GraphSim TK – C#

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Introduction **1**
- 1.1 Visualizing Molecule Similarity . . . 1

2 Theory **3**
- 2.1 Fingerprint Generation . . . 3
- 2.2 Fingerprint Types . . . 6
- 2.3 Storage and Retrieval . . . 10
- 2.4 Similarity Measures . . . 14
- 2.5 Fingerprint Database . . . 19
- 2.6 User-defined Fingerprint . . . 25
- 2.7 Fingerprint Coverage . . . 34
- 2.8 Fingerprint Overlap . . . 36
- 2.9 Fingerprint Patterns . . . 39

3 GPU Fast Fingerprints **43**
- 3.1 Prerequisites . . . 43
- 3.2 Usage . . . 44

4 Examples **45**
- 4.1 GraphSim TK Examples . . . 45
- 4.2 Python Cookbook Examples . . . 57

5 API **59**
- 5.1 OEGraphSim Classes . . . 59
- 5.2 OEGraphSim Constants . . . 102
- 5.3 OEGraphSim Functions . . . 129

6 Release History **153**
- 6.1 GraphSim TK 2.7.1 . . . 153
- 6.2 GraphSim TK 2.7.0 . . . 153
- 6.3 GraphSim TK 2.6.2 . . . 153
- 6.4 GraphSim TK 2.6.1 . . . 153
- 6.5 GraphSim TK 2.6.0 . . . 153
- 6.6 GraphSim TK 2.5.7 . . . 154
- 6.7 GraphSim TK 2.5.6 . . . 154
- 6.8 GraphSim TK 2.5.5 . . . 154
- 6.9 GraphSim TK 2.5.4 . . . 154
- 6.10 GraphSim TK 2.5.3 . . . 154
- 6.11 GraphSim TK 2.5.2 . . . 154
- 6.12 GraphSim TK 2.5.1 . . . 154
- 6.13 GraphSim TK 2.5.0 . . . 154

6.14 GraphSim TK 2.4.3 . . . 155
6.15 GraphSim TK 2.4.2 . . . 155
6.16 GraphSim TK 2.4.1 . . . 155
6.17 GraphSim TK 2.4.0 . . . 156
6.18 GraphSim TK 2.3.1 . . . 156
6.19 GraphSim TK 2.3.0 . . . 157
6.20 GraphSim TK 2.2.6 . . . 158
6.21 GraphSim TK 2.2.5 . . . 158
6.22 GraphSim TK 2.2.4 . . . 158
6.23 GraphSim TK 2.2.3 . . . 159
6.24 GraphSim TK 2.2.2 . . . 159
6.25 GraphSim TK 2.2.1 . . . 159
6.26 GraphSim TK 2.2.0 . . . 159
6.27 GraphSim TK 2.1.1 . . . 160
6.28 GraphSim TK 2.1.0 . . . 160
6.29 GraphSim TK 2.0.7 . . . 160
6.30 GraphSim TK 2.0.6 . . . 161
6.31 GraphSim TK 2.0.5 . . . 161
6.32 GraphSim TK 2.0.4 . . . 161
6.33 GraphSim TK 2.0.3 . . . 161
6.34 GraphSim TK 2.0.2 . . . 161
6.35 GraphSim TK 2.0.1 . . . 161
6.36 GraphSim TK 2.0.0 . . . 162
6.37 GraphSim TK 1.0.1 . . . 163
6.38 GraphSim TK 1.0.0 . . . 163

7 Copyright and Trademarks **165**

8 Sample Code **167**

9 Citation **169**
9.1 Orion® . . . 169
9.2 Orion Floes . . . 169
9.3 Toolkits and Applications . . . 169
9.4 Publications for Bibliographies . . . 171
9.5 OpenEye MMDS Web Service . . . 172

10 Technology Licensing **173**
10.1 GCC . . . 245

Index **261**

CHAPTER

ONE

INTRODUCTION

This manual is to familiarize the user with the **GraphSim TK** functionalities. It does not provide explanations of basic **OEChem TK** classes and functions. Therefore reading the **OEChem TK** manual beforehand is highly recommended.

The **GraphSim TK** library provides a facility for encoding 2D molecular graph information into fingerprints. The following basic fingerprint functionalities are available:

- Generation (see *Fingerprint Generation* chapter)
- Storage (see *Storage and Retrieval* chapter)
- Comparison (see *Similarity Measures* chapter)

Emphasis has been placed on speed, with attention to rapid in-memory fingerprint search (see *Fingerprint Database* chapter).

1.1 Visualizing Molecule Similarity

All images in this manual are generated utilizing the OEDepict TK and Grapheme TK .

Molecule similarity based on fingerprints is visualized by the following method: (See example in *Figure: Example of depiction of molecule similarity based on fingerprints*)

- The `OEGetFPOverlap` function is utilized to return all common fragments found between two molecules based on a given fingerprint type. These common fragments reveal the similar parts of the two molecules being compared.
- A "yellow to dark green" linear color gradient is then used to color the molecules by their similarity. Where there is similarity detected between the two molecules *i.e.* there are common fragments, the color green is used to highlight the bonds. The color gets darker with increasing similarity *i.e.* as more fragments are shared by the two molecules.
- Color pink indicates parts of the molecules that are not sharing any common fragments *i.e.* they are 2D dissimilar based on the given fingerprint type.

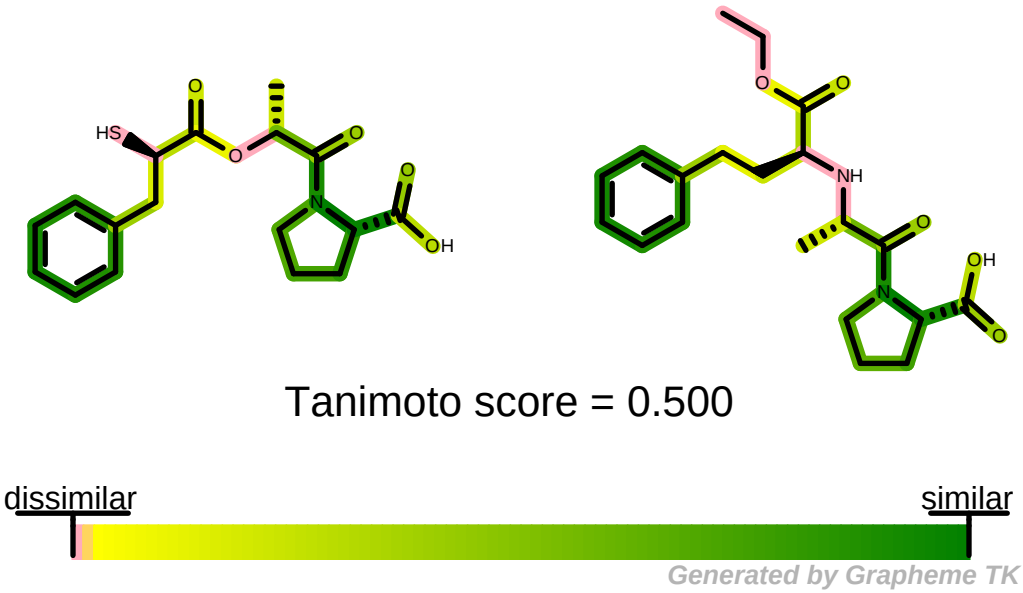


Fig. 1: **Example of depiction of molecule similarity based on fingerprints**

Hint

Visualizing similarity of two molecules based on their fingerprints provides insight into molecule similarity beyond a single numerical score and reveals information about the underlying fingerprint methods.

See also

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

CHAPTER

TWO

THEORY

2.1 Fingerprint Generation

The fingerprint types implemented in the **GraphSim TK** encode the 2D graph features of molecules. Fingerprints can be used in applications such as similarity searches, molecular characterization, molecular diversity and chemical database clustering.

The following five types of fingerprints are implemented:

1. *MACCS* (*OEFPType.MACCS166*)
2. *LINGO* (*OEFPType.Lingo*)
3. *Circular* (*OEFPType.Circular*)
4. *Path* (*OEFPType.Path*)
5. *Tree* (*OEFPType.Tree*)

2.1.1 MACCS

MACCS keys are 166 bit structural key descriptors in which each bit is associated with a SMARTS pattern.

The following code snippets demonstrate two **separate** ways to create a MACCS keys fingerprint:

```
OEFingerPrint fp = new OEFingerPrint();
OEGraphSim.OEMakeMACCS166FP(fp, mol);
```

```
OEGraphSim.OEMakeFP(fp, mol, OEFPType.MACCS166);
```

2.1.2 LINGO

The **GraphSim TK** provides fingerprint API for the LINGO similarity search method implemented in **OEChem TK**.

The following code snippets demonstrate two **separate** ways to create a LINGO fingerprint:

```
OEFingerPrint fp = new OEFingerPrint();
OEGraphSim.OEMakeLingoFP(fp, mol);
```

```
OEGraphSim.OEMakeFP(fp, mol, OEFPType.Lingo);
```

2.1.3 Circular

A circular fingerprint is generated by **exhaustively** enumerating **all** circular fragments grown radially from each heavy atom of the molecule up to the given radius and then hashing these fragments into a fixed-length bitvector. See *Figure: Example of enumerating circular fragments with various radii*.

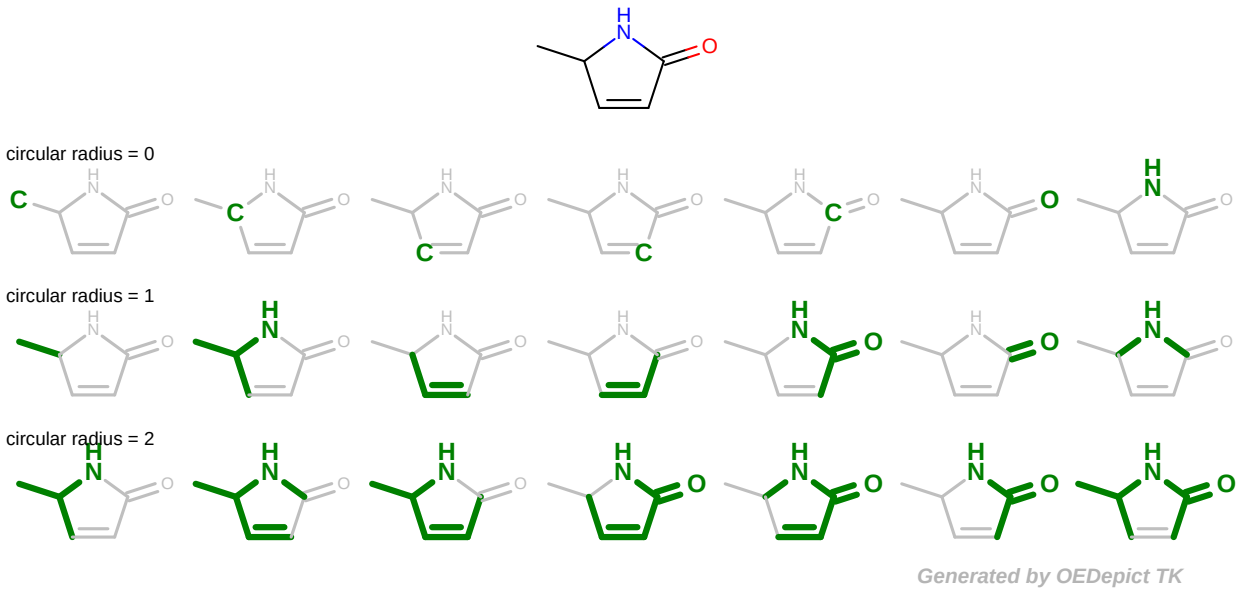


Fig. 1: **Example of enumerating circular fragments with various radii**

The following code snippets demonstrate two **separate** ways to create a circular fingerprint with default parameters:

```
OEFingerPrint fp = new OEFingerPrint();
OEGraphSim.OEMakeCircularFP(fp, mol);
```

```
OEGraphSim.OEMakeFP(fp, mol, OEFPType.Circular);
```

2.1.4 Path

A path fingerprint is generated by **exhaustively** enumerating **all** linear fragments of a molecular graph up to a given size and then hashing these fragments into a fixed-length bitvector. See *Figure: Example of enumerating path fragments with various lengths*.

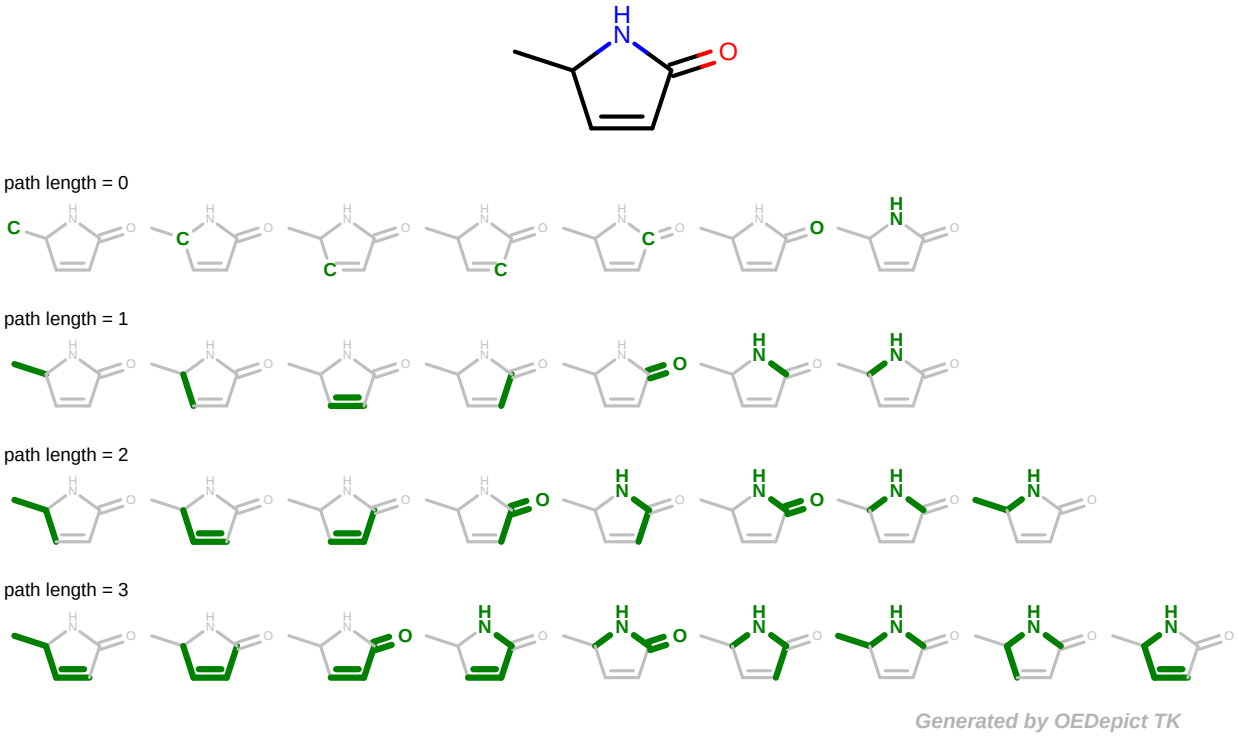


Fig. 2: **Example of enumerating path fragments with various lengths**

The following code snippets demonstrate two **separate** ways to create a path fingerprint with default parameters:

```
OEFingerPrint fp = new OEFingerPrint();
OEGraphSim.OEMakePathFP(fp, mol);
```

```
OEGraphSim.OEMakeFP(fp, mol, OEFPType.Path);
```

2.1.5 Tree

A tree fingerprint is generated by **exhaustively** enumerating **all** tree fragments of a molecular graph up to a given size and then hashing these fragments into a fixed-length bitvector. See *Figure: Example of enumerating tree fragments with various lengths*.

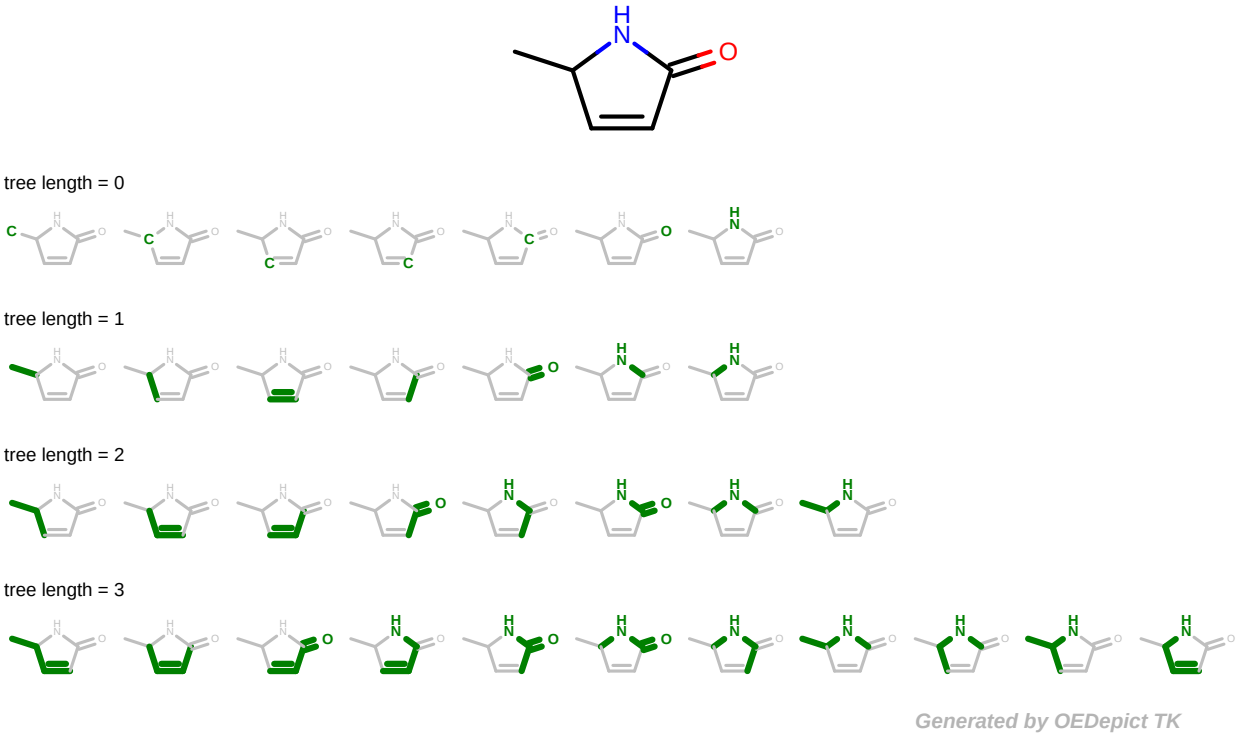


Fig. 3: **Example of enumerating tree fragments with various lengths**

The following code snippets demonstrate two **separate** ways to create a tree fingerprint with default parameters:

```
OEFingerPrint fp = new OEFingerPrint();
OEGraphSim.OEMakeTreeFP(fp, mol);
```

```
OEGraphSim.OEMakeFP(fp, mol, OEFPType.Tree);
```

Hint

GraphSim TK also provides the ability to parameterize the circular, path and tree fingerprint generation with arbitrary sets of properties. For more details see the *User-defined Fingerprint* chapter.

2.2 Fingerprint Types

A fingerprint is a bitvector. To reflect this the *OEFingerPrint* class derives from the OEBitVector class. The difference is that *OEFingerPrint* has a type that represents how the fingerprint is generated. Fingerprints may only be compared if they are generated in the same way. Therefore, the following restriction is introduced:

Warning

When two fingerprints are subjected to similarity calculation their type has to be identical.

Listing 1 shows how to create different fingerprint objects (*OEFingerPrint*) and identify or compare their types.

Listing 1: Fingerprint type

```
public class FPType
{
    public static int Main(string[] args)
    {
        OEFingerPrint fpA = new OEFingerPrint();
        OEFingerPrint fpB = new OEFingerPrint();

        if (!fpA.IsValid())
        {
            Console.WriteLine("uninitialized fingerprint");
        }

        OEGraphMol mol = new OEGraphMol();
        OEChem.OESmilesToMol(mol, "c1ccccc1");

        OEGraphSim.OEMakeFP(fpA, mol, OEFPType.Path);
        OEGraphSim.OEMakeFP(fpB, mol, OEFPType.Lingo);

        if (OEGraphSim.OEIsFPType(fpA, OEFPType.Lingo))
        {
            Console.WriteLine("Lingo");
        }

        if (OEGraphSim.OEIsFPType(fpA, OEFPType.Path))
        {
            Console.WriteLine("Path");
        }

        if (OEGraphSim.OEIsSameFPType(fpA, fpB))
        {
            Console.WriteLine("same fingerprint types");
        }
        else
        {
            Console.WriteLine("different fingerprint types");
        }
        return 0;
    }
}
```

The output of *Listing 1* is the following:

```
uninitialized fingerprint
Path
different fingerprint types
```

Two fingerprints are considered to be equivalent only if they have the same fingerprint type (*OEFPTypeBase*) and have identical bit-vectors (OEBitVector). The following code snippet shows how to compare two *OEFingerPrint* objects.

```
if (fpA == fpB)
{
    Console.WriteLine("same fingerprints");
```

(continues on next page)

(continued from previous page)

```
}
else
{
    Console.WriteLine("different fingerprints");
}
```

The following code snippet shows how to initialize a *OEFingerPrint* object by using the type of another fingerprint. The type of a fingerprint is accessed by the *OEFingerPrint.GetFPTypeBase* method.

```
OEFingerPrint fpA = new OEFingerPrint();
OEGraphSim.OEMakePathFP(fpA, mol);

OEFingerPrint fpB = new OEFingerPrint();
OEGraphSim.OEMakeFP(fpB, mol, fpA.GetFPTypeBase());
```

2.2.1 Fingerprint parameters

The *User-defined Fingerprint* chapter gives examples of how user defined fingerprints can be generated by defining, for example, the atom and bond properties that will be encoded into the fingerprints.

In order to ensure that only equivalent fingerprints can be compared, the fingerprint type stores the parameters being used in the generation process. The *OEFPTypeBase.GetFPTypeString* method returns the string representation of the fingerprint type that includes information about the parameters being used.

```
OEFingerPrint fpA = new OEFingerPrint();
OEGraphSim.OEMakeFP(fpA, mol, OEFPType.Path);
Console.WriteLine(fpA.GetFPTypeBase().GetFPTypeString());
```

The output of the preceding snippet is the following:

```
Path,ver=2.0.0,size=4096,bonds=0-5,atype=AtmNum|Arom|Chiral|FCharge|HvyDeg|Hyb|EqHalo,
btype=Order|Chiral
```

> **Note**
>
> The returned string does not include newline characters, the string was broken into two separate lines here only for better readability.

The following *Listing 2* shows how to extract the parameters of a fingerprint from a string representation by using the *OEFPTypeParams* class.

Listing 2: Fingerprint parameters

```
public class FPTypeParams
{
    public static int Main(string[] args)
    {
        OEFPTypeBase fptype = OEGraphSim.OEGetFPType(OEFPType.Path);
        OEFPTypeParams prms = new OEFPTypeParams(fptype.GetFPTypeString());
        Console.WriteLine("version = " + OEGraphSim.OEGetFingerPrintVersionString(prms.
↪GetVersion()));
```

(continues on next page)

(continued from previous page)

```
        Console.WriteLine("number of bits = " + prms.GetNumBits());
        Console.WriteLine("min bonds = " + prms.GetMinDistance());
        Console.WriteLine("max bonds = " + prms.GetMaxDistance());
        Console.WriteLine("atom types = " + OEGraphSim.OEGetFPAtomType(prms.
↪GetAtomTypes()));
        Console.WriteLine("bond types = " + OEGraphSim.OEGetFPBondType(prms.
↪GetBondTypes()));
        return 0;
    }
}
```

The output of *Listing 2* is the following:

```
version = 2.0.0
number of bits = 4096
min bonds = 0
max bonds = 5
atom types = AtmNum|Arom|Chiral|FCharge|HvyDeg|Hyb|EqHalo
bond types = Order|Chiral
```

See also

- *User-defined Fingerprint* chapter
- *OEIsValidFPTypeString* function
- *OEGetFPType* function
- *OEFPAtomType* namespace
- *OEGetFPAtomType* function
- *OEFPBondType* namespace
- *OEGetFPBondType* function

2.2.2 Fingerprint version number

Each fingerprint type additionally has a version number. Version numbers are introduced in order to keep track of changes in the fingerprint generation algorithm itself. The *OEFPTypeBase.GetFPVersionString* method returns the string representation of the fingerprint version.

```
OEFingerPrint fpB = new OEFingerPrint();
OEGraphSim.OEMakeFP(fpB, mol, OEFPType.Circular);
Console.WriteLine(fpB.GetFPTypeBase().GetFPVersionString());
```

The output of the preceding snippet is the following:

```
2.0.0
```

Warning

The version number of the fingerprints will not be changed with each release. It will be incremented only if modifications or bug fixes to the corresponding algorithm would result in generating a different bit-vector for the same

molecules.

Fingerprints with an old version number will be still readable and comparable with each other but not with fingerprints which have different version number.

2.3 Storage and Retrieval

The *OEFingerPrint* does not store any reference to the molecule from which it was generated. The user has to keep track of which fingerprint corresponds to which molecule. One way to do this is to attach the fingerprint, as generic data, to the molecule. *Listing 3* shows how store and retrieve fingerprints as generic data.

Listing 3: Storing and retrieving fingerprint as generic data

```
public class FPData
{
    public static int Main(string[] args)
    {
        string tag = "FP_DATA";
        OEGraphMol mol = new OEGraphMol();
        OEChem.OESmilesToMol(mol, "c1ccccc1");

        OEFingerPrint fp = new OEFingerPrint();
        OEGraphSim.OEMakeLingoFP(fp, mol);
        OEGraphSim.OESetFP(mol, tag, fp);

        if (mol.HasData(tag))
        {
            OEFingerPrint f = OEGraphSim.OEGetFP(mol, tag);
            if (f.IsValid())
            {
                string fptype = f.GetFPTypeBase().GetFPTypeString();
                Console.WriteLine("{0} fingerprint with `{1}` identifier", fptype, tag);
            }
        }
        return 0;
    }
}
```

It is good practice to check the validity of the fingerprint after retrieving it. The *OEFingerPrint.IsValid* method returns true if the fingerprint was successfully initialized.

Listing 4 demonstrates how to create an `OEB` binary file that stores molecules along with their fingerprints. When reading the `OEB` file that was generated by this program, the pre-calculated fingerprints can be accessed rapidly with the `PATH_FP` tag. This eliminates the *on-the-fly* generation of the fingerprints.

See also

Additional examples in *Listing 10* and *Listing 12* of the *Fingerprint Database* chapter.

Listing 4: Fingerprint generation and storage in OEB

```
public class FP2OEB
{
    public static int Main(string[] args)
    {
        if (args.Length != 2)
        {
            OEChem.OEThrow.Usage("FP2OEB <infile> <outfile>");
        }

        oemolistream ifs = new oemolistream();
        if (!ifs.open(args[0]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[0] + " for reading");
        }

        oemolostream ofs = new oemolostream();
        if (!ofs.open(args[1]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[1] + " for writing");
        }

        if (ofs.GetFormat() != OEFormat.OEB)
        {
            OEChem.OEThrow.Fatal(args[1] + " output file has to be an OEBinary file");
        }

        OEFingerPrint fp = new OEFingerPrint();
        foreach (OEGraphMol mol in ifs.GetOEGraphMols())
        {
            OEGraphSim.OEMakeFP(fp, mol, OEFPType.Path);
            OEGraphSim.OESetFP(mol, "PATH_FP", fp);
            OEChem.OEWriteMolecule(ofs, mol);
        }
        return 0;
    }
}
```

The following code snippets shows how to generate a bitstring from an *OEFingerPrint* object.

Listing 5: Accessing a fingerprint as a bitstring

```
    static string GetBitString(OEFingerPrint fp)
    {
        string bitstring = "";
        for (uint b = 0; b < fp.GetSize(); b++)
        {
```

(continues on next page)

(continued from previous page)

```
            if (fp.IsBitOn(b))
            {
                bitstring += '1';
            }
            else
            {
                bitstring += '0';
            }
        }

        return bitstring;
    }
```

See also

- OEBitVector class

Warning

Even though the **GraphSim TK** library provides a fingerprint API for the LINGO similarity search method, it is not implemented as a *real* fingerprint, so bitstrings that are generated from LINGO fingerprints are meaningless.

Fingerprints can also be stored in SDF files. The *Listing 6* demonstrates how to create an SDF and store fingerprints as hexadecimal strings.

After the fingerprint is generated, it is attached to the molecule as an SD data tag. The identifier of the fingerprint in the SDF file will be the string representation of the fingerprint type. The bitvector of the fingerprint is converted to a hexadecimal string with the `OEBitVector.ToHexString` method.

Listing 6: Storing fingerprint in SDF

```
public class FP2SDF
{
    public static int Main(string[] args)
    {

        if (args.Length != 2)
        {
            OEChem.OEThrow.Usage("FP2SDF <infile> <outfile>");
        }

        oemolistream ifs = new oemolistream();
        if (!ifs.open(args[0]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[0] + " for reading");
        }

        oemolostream ofs =  new oemolostream();
        if (!ofs.open(args[1]))
        {
```

(continues on next page)

(continued from previous page)

```
            OEChem.OEThrow.Fatal("Unable to open " + args[1] + " for writing");
        }
        if (ofs.GetFormat() != OEFormat.SDF)
        {
            OEChem.OEThrow.Fatal(args[1] + " output file has to be an SDF file");
        }

        OEFingerPrint fp = new OEFingerPrint();
        OEGraphMol mol = new OEGraphMol();
        while (OEChem.OEReadMolecule(ifs, mol))
        {
            OEGraphSim.OEMakeFP(fp, mol, OEFPType.Circular);
            string fptypestr = fp.GetFPTypeBase().GetFPTypeString();
            string fphexdata = fp.ToHexString();
            OEChem.OESetSDData(mol, fptypestr, fphexdata);
            OEChem.OEWriteMolecule(ofs, mol);
        }
        ofs.close();
        return 0;
    }
}
```

The following example (*Listing 7*) shows how to retrieve fingerprints from an SDF file. When looping over the SD data the *OEIsValidFPTypeString* functions can be used to identify SD data that stores a fingerprint. The tag of the data is the string representation of the fingerprint type. This string representation can be used to generate the corresponding *OEFPTypeBase* object by using the *OEGetFPType* function. The type of the *OEFingerPrint* object then has to be set by the *OEFingerPrint.SetFPTypeBase* method. Finally, the bitvector of the fingerprint then can be initialized from the hexadecimal string by using the `OEBitVector.FromHexString` method.

Listing 7: Retrieving fingerprint from SDF

```
public class SDF2FP
{
    public static int Main(string[] args)
    {
        if (args.Length != 1)
            OEChem.OEThrow.Usage("SDF2FP <infile>");

        oemolistream ifs = new oemolistream();
        if (!ifs.open(args[0]))
            OEChem.OEThrow.Fatal("Unable to open " + args[0] + " for reading");
        if (ifs.GetFormat() != OEFormat.SDF)
            OEChem.OEThrow.Fatal(args[0] + " input file has to be an SDF file");

        int molcounter = 0;
        int fpcounter  = 0;
        OEGraphMol mol = new OEGraphMol();
        while (OEChem.OEReadMolecule(ifs, mol))
        {
            molcounter += 1;
            foreach (OESDDataPair dp in OEChem.OEGetSDDataPairs(mol))
            {
```

(continues on next page)

(continued from previous page)

```
                if (OEGraphSim.OEIsValidFPTypeString(dp.GetTag()))
                {
                    fpcounter += 1;
                    String fptypestr = dp.GetTag();
                    String fphexdata = dp.GetValue();
                    OEFingerPrint fp = new OEFingerPrint();
                    OEFPTypeBase fptype = OEGraphSim.OEGetFPType(fptypestr);
                    fp.SetFPTypeBase(fptype);
                    fp.FromHexString(fphexdata);
                }
            }
        }

        Console.WriteLine("Number of molecules = " + molcounter);
        Console.WriteLine("Number of fingerprints = " + fpcounter);

        return 0;
    }
}
```

See also

- *OEFPTypeBase.GetFPTypeString* method
- OEBitVector class
- **SD Tagged Data Manipulation** chapter in the **OEChem TK** manual

2.4 Similarity Measures

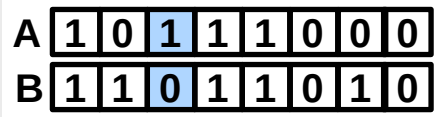
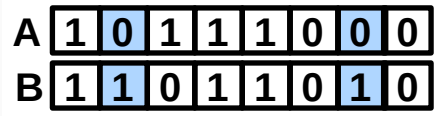
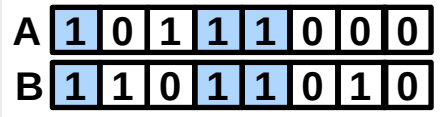
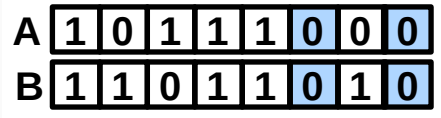
The basic idea underlying similarity-based measures is that molecules that are structurally similar are likely to have similar properties. In a fingerprint the presence or absence of a structural fragment is represented by the presence or absence of a set bit. This means that two molecules are judged as being similar if they have a large number of bits in common.

Measuring molecular similarity or dissimilarity has two basic components: the representation of molecular characteristics (such as fingerprints) and the similarity coefficient that is used to quantify the degree of resemblance between two such representations.

2.4.1 Built-in Similarity Measures

Since different similarity coefficients quantify different types of structural resemblance, several built-in similarity measures are available in the **GraphSim TK** (see *Table: Basic bit count terms of similarity calculation*) The table below defines the four basic bit count terms that are used in fingerprint-based similarity calculations:

Table 1: **Basic bit count terms of similarity calculation**

Symbol	Description	
$onlyA$	number of bits set "on" in fingerprint A but not in B	
$onlyB$	number of bits set "on" in fingerprint B but not in A	
$bothAB$	number of bits set "on" in both fingerprints	
$neitherAB$	number of bits set "off" in both fingerprints	
$\|A\|$	number of bits set "on" in fingerprint A	
$\|B\|$	number of bits set "on" in fingerprint B	
$fpsize$	length of fingerprint in bits	

Cosine

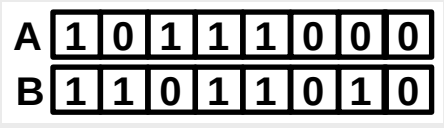
Formula:

$$Sim_{Cosine}(A,B) = \frac{bothAB}{\sqrt{|A|*|B|}} = \frac{bothAB}{\sqrt{(onlyA+bothAB)*(onlyB+bothAB)}}$$

Range:

$[0.0, 1.0]$

Example:



$$\frac{bothAB}{\sqrt{(onlyA+bothAB)*(onlyB+bothAB)}} = \frac{3}{\sqrt{(1+3)*(2+3)}} = \frac{3}{\sqrt{20}} = 0.67$$

Calculates the ratio of the bits in common to the geometric mean of the number of "on" bits in the two fingerprints.

Dice

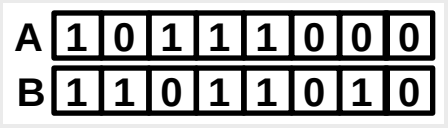
Formula:

$$Sim_{Dice}(A,B) = \frac{2*bothAB}{|A|+|B|} = \frac{2*bothAB}{onlyA+onlyB+2*bothAB}$$

Range:

$[0.0, 1.0]$

Example:



$$\frac{2*bothAB}{onlyA+onlyB+2*bothAB} = \frac{2*3}{1+2+2*3} = \frac{6}{9} = 0.666$$

Calculates the ratio of the bits in common to the arithmetic mean of the number of "on" bits in the two fingerprints.

Euclidean

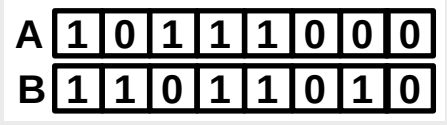
Formula:

$$Sim_{Euclid}(A,B) = \sqrt{\frac{bothAB+neitherAB}{fpsize}} = \sqrt{\frac{bothAB+neitherAB}{onlyA+onlyB+bothAB+neitherAB}}$$

Range:

$[0.0, 1.0]$

Example:



$$\sqrt{\frac{bothAB+neitherAB}{onlyA+onlyB+bothAB+neitherAB}} = \sqrt{\frac{3+2}{1+2+3+2}} = \sqrt{\frac{5}{8}} = 0.791$$

Manhattan

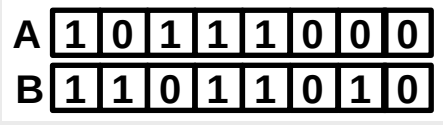
Formula:

$$Sim_{Manhattan}(A,B) = \frac{onlyA+onlyB}{fpsize} = \frac{onlyA+onlyB}{onlyA+onlyB+bothAB+neitherAB}$$

Range:

$[0.0, 1.0]$

Example:



$$\frac{onlyA+onlyB}{onlyA+onlyB+bothAB+neitherAB} = \frac{1+2}{1+2+3+2} = \frac{3}{8} = 0.375$$

Note

Although $Sim_{Manhattan}$ shares the same range with other similarity measures, it acts more like a **distance** measure, scoring more similar fingerprints lower. Identical fingerprints have $Sim_{Manhattan}$ of 0.0 (as opposed to 1.0 in any other measure).

Tanimoto

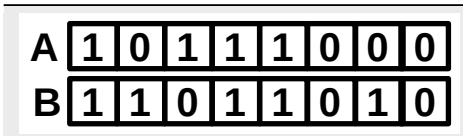
Formula:

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

Range:

$[0.0, 1.0]$

Example:



$\frac{bothAB}{onlyA+onlyB+bothAB} = \frac{3}{1+2+3} = \frac{3}{6} = 0.5$

Note

The calculation of the `OEFPType.Lingo` fingerprint is based on fragmenting canonical isomeric SMILES into overlapping four character long substrings. If any of the two SMILES being compared is shorter than four characters, then their Tanimoto score will be:

- *1.0*, if the two SMILES are identical
- *0.0*, otherwise.

Tversky

Formula:

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

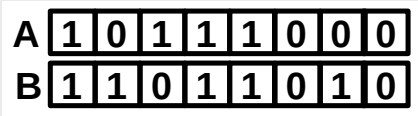
The Tversky similarity measure is asymmetric. Setting the parameters $\alpha = \beta = 1.0$ is identical to using the *Tanimoto* measure.

The factor α weights the contribution of the first 'reference' molecule. The larger α becomes, the more weight is put on the bit setting of the reference molecule.

Range:

$[0.0, 1.0]$

Example:



$\frac{bothAB}{\alpha * onlyA + \beta * onlyB + bothAB}(\alpha = 2.0, \beta = 1.0) = \frac{3}{2.0*1+1.0*2+3} = \frac{3}{7} = 0.429$

Note

Although $Sim_{Tversky}$ shares the same range with other similarity measures, its scaling can vary by orders of magnitude depending on the choice of α and β parameters.

2.4.2 Similarity Calculation

The following example demonstrates how to calculate `Tanimoto` similarity scores for the molecules depicted in *Figure: Example molecules*.

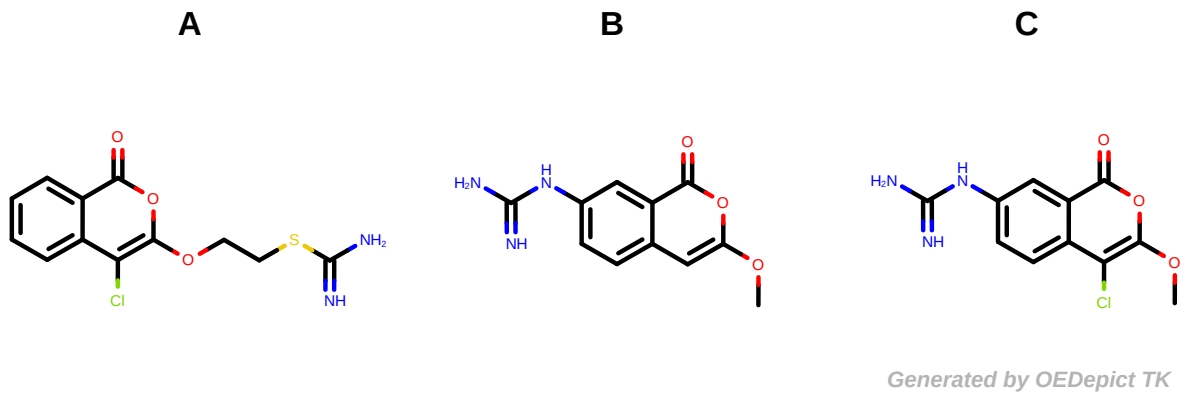


Fig. 4: **Example molecules**

Listing 8: Calculating Tanimoto index

```
public class FPTanimoto
{
    public static int Main(string[] args)
    {
        OEGraphMol molA = new OEGraphMol();
        OEChem.OESmilesToMol(molA, "c1ccc2c(c1)c(c(oc2=O)OCCSC(=N)N)Cl");
        OEFingerPrint fpA = new OEFingerPrint();
        OEGraphSim.OEMakeFP(fpA, molA, OEFPType.MACCS166);

        OEGraphMol molB = new OEGraphMol();
        OEChem.OESmilesToMol(molB, "COc1cc2ccc(cc2c(=O)o1)NC(=N)N");
        OEFingerPrint fpB = new OEFingerPrint();
        OEGraphSim.OEMakeFP(fpB, molB, OEFPType.MACCS166);

        OEGraphMol molC = new OEGraphMol();
        OEChem.OESmilesToMol(molC, "COc1c(c2ccc(cc2c(=O)o1)NC(=N)N)Cl");
        OEFingerPrint fpC = new OEFingerPrint();
        OEGraphSim.OEMakeFP(fpC, molC, OEFPType.MACCS166);

        Console.WriteLine("Tanimoto(A,B) = {0:0.000}", OEGraphSim.OETanimoto(fpA, fpB));
        Console.WriteLine("Tanimoto(A,C) = {0:0.000}", OEGraphSim.OETanimoto(fpA, fpC));
        Console.WriteLine("Tanimoto(B,C) = {0:0.000}", OEGraphSim.OETanimoto(fpB, fpC));
        return 0;
    }
}
```

Molecules B and C (shown in *Figure: Example Molecules*) have the largest Tanimoto value since they share the largest number of common structural features.

For these example molecule the output of *Listing 8* is the following:

```
Tanimoto(A,B) = 0.618
Tanimoto(A,C) = 0.709
Tanimoto(B,C) = 0.889
```

2.4.3 User-defined Similarity Measures

The following code snippet demonstrates how implement the Yule similarity measure with the following formula:

$Sim_{Yule}(A,B) = \sqrt{\frac{(bothAB*neitherAB)-(onlyA*onlyB)}{(bothAB*neitherAB)+(onlyA*onlyB)}}$

```
    public static float CalculateYule(OEFingerPrint fpA, OEFingerPrint fpB)
    {
        uint onlyA, onlyB, bothAB, neitherAB;
        OEChem.OEGetBitCounts(fpA, fpB, out onlyA, out onlyB, out bothAB, out neitherAB);
        float yule = 0.0f;
        yule = (float)(bothAB * neitherAB) -
               (float)(onlyA * onlyB);
        yule /= (float)(bothAB * neitherAB) +
                (float)(onlyA * onlyB);
        return yule;
    }
```

The *OEGetBitCounts* function returns the four basic values (namely $onlyA$, $onlyB$, $bothAB$ and $neitherAB$) from which any similarity measures can be calculated. For the definition of these values see *Table: Basic bit count terms*

```
OEGraphSim.OEMakeFP(fpA, molA, OEFPType.Path);
OEGraphSim.OEMakeFP(fpB, molB, OEFPType.Path);
OEGraphSim.OEMakeFP(fpC, molC, OEFPType.Path);

Console.WriteLine("Yule(A,B) = {0:0.000}", CalculateYule(fpA, fpB));
Console.WriteLine("Yule(A,C) = {0:0.000}", CalculateYule(fpA, fpC));
Console.WriteLine("Yule(B,C) = {0:0.000}", CalculateYule(fpB, fpC));
```

> **Warning**
>
> User-defined similarity measures can only be used with circular, path, tree, and MACCS key fingerprints but **not** with LINGO (*OEFPType.Lingo*).

2.5 Fingerprint Database

The following four examples perform the same task, detailed below:

1. reading a query structure
2. printing out the similarity score between the fingerprint of this query and the fingerprint generated for each molecule read from a database file.

In *Listing 9*, after importing the query structure and generating its path fingerprint, the program loops over the database file creating a path fingerprint for each structure. Then the program calculates the Tanimoto similarity between the fingerprint of the query and the database entry by calling the *OETanimoto* function.

Listing 9: Similarity calculation from file

```
public class SimCalcFromFile
{
    public static int Main(string[] args)
    {
```

(continues on next page)

(continued from previous page)

```
        if (args.Length != 2)
        {
            OEChem.OEThrow.Usage("SimCalcFromFile <queryfile> <targetfile>");
        }

        oemolistream ifs = new oemolistream();
        if (!ifs.open(args[0]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[0] + " for reading");
        }

        OEGraphMol qmol = new OEGraphMol();
        if (!OEChem.OEReadMolecule(ifs, qmol))
        {
            OEChem.OEThrow.Fatal("Unable to read query molecule");
        }

        OEFingerPrint qfp = new OEFingerPrint();
        OEGraphSim.OEMakeFP(qfp, qmol, OEFPType.Path);

        if (!ifs.open(args[1]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[1] + " for reading");
        }

        OEFingerPrint tfp = new OEFingerPrint();
        foreach (OEGraphMol tmol in ifs.GetOEGraphMols())
        {
            OEGraphSim.OEMakeFP(tfp, tmol, OEFPType.Path);
            Console.WriteLine("{0:0.000}", OEGraphSim.OETanimoto(qfp, tfp));
        }
        return 0;
    }
}
```

In *Listing 10* only the code block that is different from *Listing 9* is shown.

In this example, it is assumed that the fingerprints are pre-calculated and stored in an `OEB` binary file as generic data attached to the corresponding molecules. The program loops over the file and accesses the pre-generated fingerprints or calculates them if they are not available.

The obvious advantage of this process is that the fingerprints one have to be generated once when the binary file is created. This can be significantly faster, than generating the fingerprints on-the-fly every time the program is executed.

See also

The *Storage and Retrieval* section shows an example of how to generate an `OEB` binary file which stores molecule along with their corresponding fingerprints.

Listing 10: Similarity calculation from OEB file

```
        OEFingerPrint tfp = new OEFingerPrint();
        foreach (OEGraphMol tmol in ifs.GetOEGraphMols())
        {
            if (tmol.HasData("PATH_FP"))
            {
                tfp = OEGraphSim.OEGetFP(tmol, "PATH_FP");
            }
            else
            {
                OEChem.OEThrow.Warning("Unable to access fingerprint for" + tmol.
↪GetTitle());
                OEGraphSim.OEMakeFP(tfp, tmol, OEFPType.Path);
            }

            Console.WriteLine("{0:0.000}", OEGraphSim.OETanimoto(qfp, tfp));
        }
```

Listing 11 differs from *Listing 9* in that it uses an *OEFPDatabase* object to store the generated fingerprints. The *OEFPDatabase* class is designed to perform in-memory fingerprint searches.

Listing 11: Similarity calculation with fingerprint database from file

```
        OEFPDatabase fpdb = new OEFPDatabase(qfp.GetFPTypeBase());
        foreach (OEGraphMol tmol in ifs.GetOEGraphMols())
        {
            fpdb.AddFP(tmol);
        }

        foreach (OESimScore score in fpdb.GetScores(qfp))
        {
            Console.WriteLine("{0:0.000}", score.GetScore());
        }
```

After building the fingerprint database, the scores can be accessed by the *OEFPDatabase.GetScores* method. This will return an iterator over the similarity scores calculated.

Note

The *OEFPDatabase* only stores fingerprints and not the molecules from which they are generated. A correspondence between a molecule and its fingerprint stored in the database can be established by using the index returned by the *OEFPDatabase.AddFP* method.

See also

Listing 13 shows how to keep track of the correspondence between a fingerprint added to a *OEFPDatabase* object and a molecule from which it is calculated.

In the last example (*Listing 12*), *OEFPDatabase* is used again to store the fingerprints. If the fingerprint is read from the OEB input binary file, then it is directly added to the database, otherwise the fingerprint is generated on-the-fly when passing the OEMolBase molecule itself to the *OEFPDatabase.AddFP* method.

Listing 12: Similarity calculation with fingerprint database from OEB

```
        OEFPDatabase fpdb = new OEFPDatabase(qfp.GetFPTypeBase());
        OEFingerPrint tfp = new OEFingerPrint();
        foreach (OEGraphMol tmol in ifs.GetOEGraphMols())
        {
            if (tmol.HasData("PATH_FP"))
            {
                tfp = OEGraphSim.OEGetFP(tmol, "PATH_FP");
                fpdb.AddFP(tfp);
            }
            else
            {
                OEChem.OEThrow.Warning("Unable to access fingerprint for" + tmol.
↪GetTitle());
                fpdb.AddFP(tmol);
            }
        }

        foreach (OESimScore score in fpdb.GetScores(qfp))
        {
            Console.WriteLine("{0:0.000}", score.GetScore());
        }
```

2.5.1 Sorted Search

Similarity searching based on a 2D representation of molecular structure (such as fingerprints) is one of the most common approaches for virtual screening. A molecule that is structurally similar to an active molecule is more likely to be active.

A virtual screening strategy involves going through a molecule database and calculating the similarity between a reference structure and each of the molecules, followed by ranking the similarity scores in descending order to identify molecules that are the most similar to the reference structure.

Listing 13 shows how to search a molecule database using the *OEFPDatabase.GetSortedScores* method to identify analogs based on their fingerprint similarity. First the molecules are imported using an OEMolDatabase object. Then a *OEFPDatabase* object is created that will store the corresponding fingerprints. Iterating over the molecules a fingerprints are added to the database by calling the *OEFPDatabase.AddFP* method. In case when a molecule can not be accessed from the OEMolDatabase object, an empty fingerprint is added to the *OEFPDatabase* object. This ensures that the indices of the two databases are synchronized. After generating the fingerprints, the program reads reference molecules (in the SMILES format) from standard input. Then the input SMILES string is parsed and the fingerprint database is searched to identify structures with the highest similarity scores. Finally, the SMILES string of the best hits are written to standard output.

Listing 13: Similarity search in memory

```
public class MemSimSearch
{
    public static void Main(string[] args)
    {
        if (args.Length != 1)
            OEChem.OEThrow.Usage("MemSimSearch <database>");

        oemolistream ifs = new oemolistream();
        if (!ifs.open(args[0]))
            OEChem.OEThrow.Fatal("Cannot open database molecule file!");

        // load molecules

        OEMolDatabase moldb = new OEMolDatabase(ifs);
        uint nrmols = moldb.GetMaxMolIdx();

        // generate fingerprints

        OEFPDatabase fpdb = new OEFPDatabase(OEFPType.Path);

        OEFingerPrint emptyfp = new OEFingerPrint();
        emptyfp.SetFPTypeBase(fpdb.GetFPTypeBase());

        OEGraphMol mol = new OEGraphMol();
        for (uint idx = 0; idx < nrmols; ++idx)
        {
            if (moldb.GetMolecule(mol, idx))
                fpdb.AddFP(mol);
            else
                fpdb.AddFP(emptyfp);
        }
        uint nrfps = fpdb.NumFingerPrints();

        OEWallTimer timer = new OEWallTimer();

        while (true)
        {
            try
            {
                // read query SMILES from System.in

                Console.Write("Enter SMILES> ");
                StreamReader isr = new StreamReader(Console.OpenStandardInput());

                // parse query

                string str = isr.ReadLine();
                OEGraphMol query = new OEGraphMol();
                if (!OEChem.OEParseSmiles(query, str))
                {
                    OEChem.OEThrow.Warning("Invalid SMILES String");
```

(continues on next page)

(continued from previous page)

```
                    continue;
                }

                // calculate similarity scores

                timer.Start();
                OESimScoreIter siter = fpdb.GetSortedScores(query, 5);
                Console.WriteLine("{0} seconds to search {1} fingerprints",
                                  timer.Elapsed(), nrfps);

                OEGraphMol hit = new OEGraphMol();
                for (siter.ToFirst(); siter.IsValid(); siter.Increment())
                {
                    if (moldb.GetMolecule(hit, siter.Target().GetIdx()))
                    {
                        string smiles = OEChem.OEMolToSmiles(hit);
                        Console.WriteLine("Tanimoto score {0,4:0.000} {1}",
                                          siter.Target().GetScore(), smiles);
                    }
                }
            }
            catch (Exception e)
            {
                Console.WriteLine(e);
            }
        }
    }
}
```

As mentioned before, *OEFPDatabase* is a fingerprint container and does not store the corresponding molecule. Therefore the molecules have to be stored in a separate container in this case in an OEMolDatabase object. When the *OEFPDatabase.GetSortedScores* returns the iterator over the best similarity scores, the associated index can be utilized to access the corresponding structure in the OEMolDatabase object.

In the above example, the entire database was searched to identify structurally similar molecules. However, the user can also specify a segment of the database to be searched by providing a begin and end index.

See also

- OEMolDatabase class in the **OEChem TK** manual
- Examples of fingerprint searches in the API section:
  - *OEFPDatabase.GetScores* method
  - *OEFPDatabase.GetSortedScores* method

2.5.2 Searching with User-defined Similarity Measures

By default, the *Tanimoto* similarity is used when calling either the *OEFPDatabase.GetScores* method or the *OEFPDatabase.GetSortedScores* method. The user can set other types of similarity measures to be applied by calling the *OEFPDatabase.SetSimFunc* method with a value from the *OESimMeasure* namespace. Each of the constants from this namespace corresponds to one of the built-in similarity calculation methods.

There is also a facility to use user-defined similarity measures when searching a fingerprint database. The following example shows how a similarity calculation can be implemented by deriving from the *OESimFuncBase* class.

Formula: $Sim_{Simpson}(A, B) = \sqrt{\frac{bothAB}{min(onlyA+bothAB),(onlyB+bothAB))}}$

```
private class SimpsonSimFunc : OESimFuncBase
{
    public override float Call(OEFingerPrint fpA, OEFingerPrint fpB)
    {
        uint onlyA, onlyB, bothAB;
        OEChem.OEGetBitCounts(fpA, fpB, out onlyA, out onlyB, out bothAB);

        float sim = (float)bothAB;
        sim /= (float)Math.Min((onlyA + bothAB), (onlyB + bothAB));
        return sim;
    }

    public override OESimFuncBase CreateCopy()
    {
        OESimFuncBase copy = new SimpsonSimFunc();
        copy.ReleaseOwnership();
        return copy;
    }

    public override string GetSimTypeString()
    {
        return "Simpson";
    }
}
```

After implementing the similarity calculation, it can be added to an *OEFPDatabase* object, henceforth this new similarity calculation will be used.

```
OEFPDatabase fpdb = new OEFPDatabase(OEFPType.Path);
fpdb.SetSimFunc(new SimpsonSimFunc());
```

See also

- The *User-defined Similarity Measures* section

2.6 User-defined Fingerprint

The previous *Fingerprint Generation* chapter showed how to create circular, path and tree fingerprints with default parameters. These default parameters are calibrated on the Briem-Lessel [Briem-Lessel-2000], Hert-Willett [Hert-Willett-2004] and Grant [Grant-2006] benchmarks.

However, the **GraphSim TK** also provides facilities to construct user-defined fingerprints. When constructing a user-defined fingerprint, the following parameters have to be considered:

1. Atom and bond typing that define which atom and bond properties are encoded into the fingerprints (see the *Atom and Bond Typing* section)
2. Size of the fragments that are exhaustively enumerated during the fingerprint generation (see the *Fragment Size* section)

3. Size of the generated fingerprint (in bits) (see the *Fingerprint Size* section)

The following code snippet shows how to generate a `1024` bit long fingerprint that encodes paths from `0` up to `5` bonds in length with default atom and bond properties defined by the *OEFPAtomType.DefaultAtom* and *OEFPBondType.DefaultBond* constants, respectively.

```
uint numbits = 1024;
uint minbonds = 0;
uint maxbonds = 5;
OEGraphSim.OEMakePathFP(fp,
                        mol,
                        numbits,
                        minbonds,
                        maxbonds,
                        OEFPAtomType.DefaultAtom,
                        OEFPBondType.DefaultBond);
```

Warning

Two fingerprints which are generated with different parameters will have different fingerprint types!

In *Listing 14*, two fingerprints are generated with different parameters, namely they have a different number of bits. This means that they also have different types, therefore, no similarity value can be calculated between them.

Listing 14: Example of different path fingerprint types

```
        OEFingerPrint fpA = new OEFingerPrint();
        uint numbits = 1024;
        uint minbonds = 0;
        uint maxbonds = 5;
        OEGraphSim.OEMakePathFP(fpA,
                                mol,
                                numbits,
                                minbonds,
                                maxbonds,
                                OEFPAtomType.DefaultAtom,
                                OEFPBondType.DefaultBond);
        OEFingerPrint fpB = new OEFingerPrint();
        numbits = 2048;
        OEGraphSim.OEMakePathFP(fpB,
                                mol,
                                numbits,
                                minbonds,
                                maxbonds,
                                OEFPAtomType.DefaultAtom,
                                OEFPBondType.DefaultBond);

        Console.WriteLine("same fingerprint types = " + OEGraphSim.OEIsSameFPType(fpA,␣
↪fpB));
        Console.WriteLine("{0:0.000}", OEGraphSim.OETanimoto(fpA, fpB));
```

The output of *Listing 14* is the following:

```
same fingerprint types = False
Fatal: fingerprint type mismatch!
```

2.6.1 Atom and Bond Typing

Listing 15 shows how to generate fingerprints for two molecules with various atom and bond types (depicted in *Example molecules*). Reducing the number of atom and bond properties increases the similarity between the two molecules (*i.e.* their *Tanimoto* similarity). At the end, when only the topology of two molecules is considered, *i.e.*, whether or not their atoms and bonds belong to any ring system, the fingerprints of the two molecules become identical.

These effects are illustrated in *Table: Examples of depiction molecule similarity based on fingerprints*.

See also

- *Visualizing Molecule Similarity* section

Listing 15: Similarity calculation with various atom/bond typing

```
public class FPAtomTyping
{
    public static int Main(string[] args)
    {
        OEGraphMol molA = new OEGraphMol();
        OEChem.OESmilesToMol(molA, "Oc1c2c(cc(c1)CF)CCCC2");
        OEGraphMol molB = new OEGraphMol();
        OEChem.OESmilesToMol(molB, "c1ccc2c(c1)c(cc(n2)CCl)N");

        PrintTanimoto(molA, molB, OEFPAtomType.DefaultAtom, OEFPBondType.DefaultBond);
        PrintTanimoto(molA, molB, OEFPAtomType.DefaultAtom | OEFPAtomType.EqAromatic,␣
↪OEFPBondType.DefaultBond);
        PrintTanimoto(molA, molB, OEFPAtomType.Aromaticity, OEFPBondType.DefaultBond);
        PrintTanimoto(molA, molB, OEFPAtomType.InRing, OEFPBondType.InRing);
        return 0;
    }

    private static void PrintTanimoto(OEMolBase molA, OEMolBase molB, uint atype, uint␣
↪btype)
    {
        OEFingerPrint fpA = new OEFingerPrint();
        OEFingerPrint fpB = new OEFingerPrint();
        uint numbits = 2048;
        uint minb = 0;
        uint maxb = 5;
        OEGraphSim.OEMakePathFP(fpA, molA, numbits, minb, maxb, atype, btype);
        OEGraphSim.OEMakePathFP(fpB, molB, numbits, minb, maxb, atype, btype);
        Console.WriteLine("Tanimoto(A,B) = {0:0.000}", OEGraphSim.OETanimoto(fpA, fpB));
    }
}
```

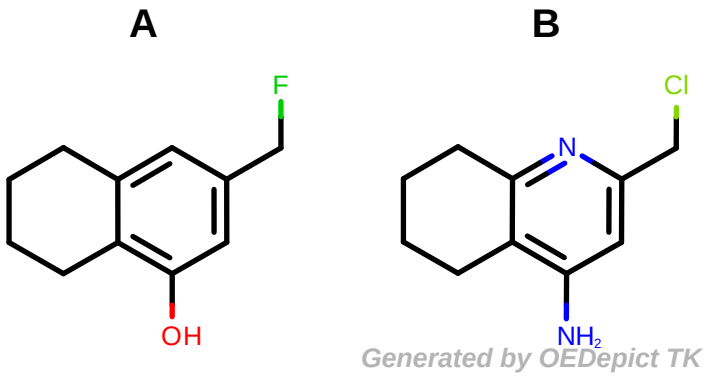


Fig. 5: **Example molecules**

The output of *Listing 15* is the following:

```
Tanimoto(A,B) = 0.166
Tanimoto(A,B) = 0.241
Tanimoto(A,B) = 0.592
Tanimoto(A,B) = 1.000
```

Table 2: **Examples of depiction molecule similarity based on fingerprints**

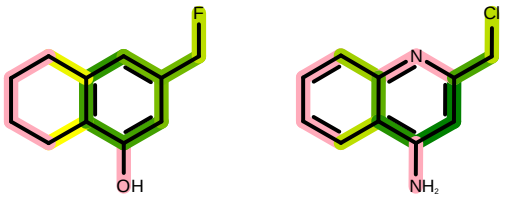
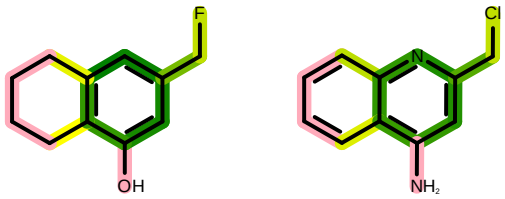
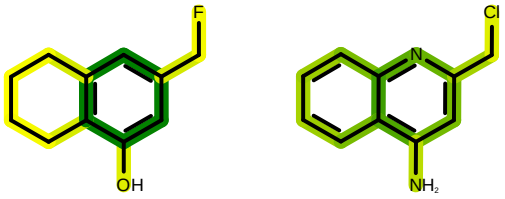
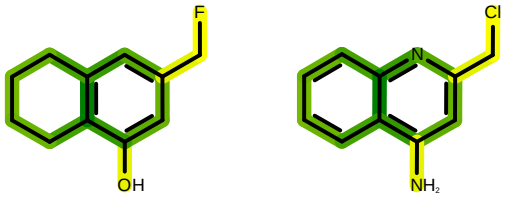
DefaultPathAtom and DefaultPathBond A B  Tanimoto score = 0.166 Generated by Grapheme TK	**DefaultPathAtom | EqAromatic and DefaultBond** A B  Tanimoto score = 0.241 Generated by Grapheme TK
Aromaticity and DefaultPathBond A B  Tanimoto score = 0.592 Generated by Grapheme TK	**InRing and InRing** A B  Tanimoto score = 1.000 Generated by Grapheme TK

Table: Atom typing options and *Table: Bond typing options* list the currently available typing options.

Table 3: Atom typing options

atom typing constant	encoded atom property
OEFPAtomType.Aromaticity	OEAtomBase.IsAromatic
OEFPAtomType.AtomicNumber	OEAtomBase.GetAtomicNum
OEFPAtomType.Chiral	OEAtomBase.IsChiral
OEFPAtomType.FormalCharge	OEAtomBase.GetFormalCharge
OEFPAtomType.HCount	OEAtomBase.GetTotalHCount
OEFPAtomType.HvyDegree	OEAtomBase.GetHvyDegree
OEFPAtomType.Hybridization	OEAtomBase.GetHyb
OEFPAtomType.InRing	OEAtomBase.IsInRing

Table 4: Atomic number modifiers

OEFPAtomType.EqAromatic
OEFPAtomType.EqHalogen
OEFPAtomType.EqHBondAcceptor
OEFPAtomType.EqHBondDonor

Table 5: Bond typing options

bond typing constant	encoded bond property
OEFPBondType.BondOrder	GetOrder
OEFPBondType.Chiral	OEBondBase.IsChiral
OEFPBondType.InRing	OEBondBase.IsInRing

See also

- *OEFPAtomType* namespace
- *OEFPBondType* namespace

2.6.2 Fragment Size

Circular, path and tree-based fingerprint generation involves molecular graph traversal to identify all unique radial, linear or branched fragments, respectively. When a path or tree fingerprint is initialized, the minimum and maximum number of bonds of the fragments that are encoded into the fingerprint can be specified. See *Figure: Example of enumerated path fragments with increasing number of bonds* and *Figure: Example of enumerated tree fragments with increasing number of bonds*.

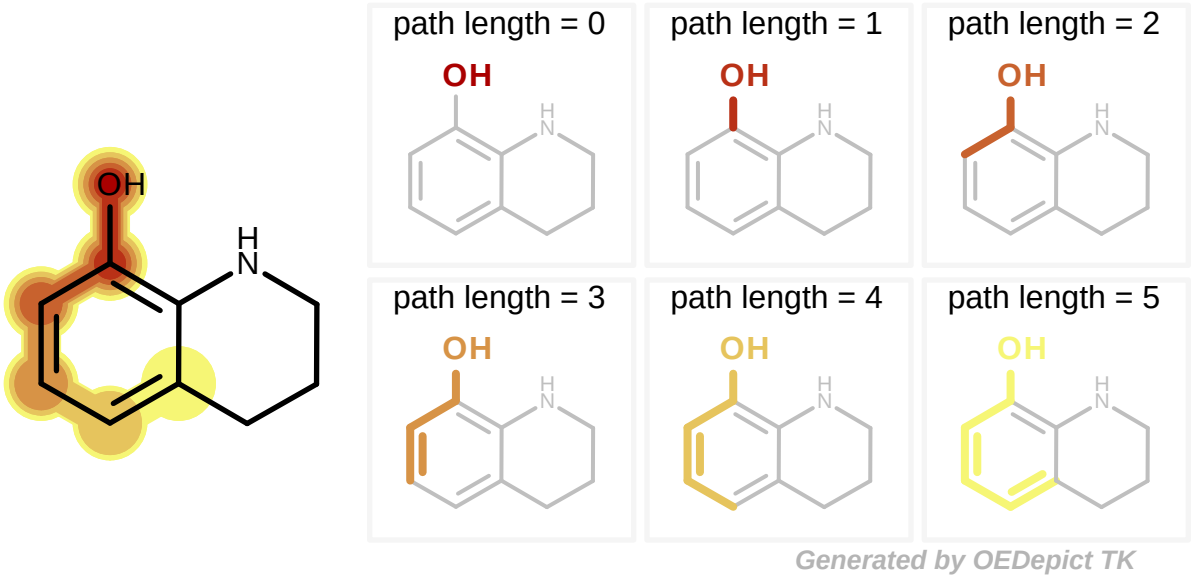


Fig. 6: **Example of enumerated path fragments with increasing number of bonds**

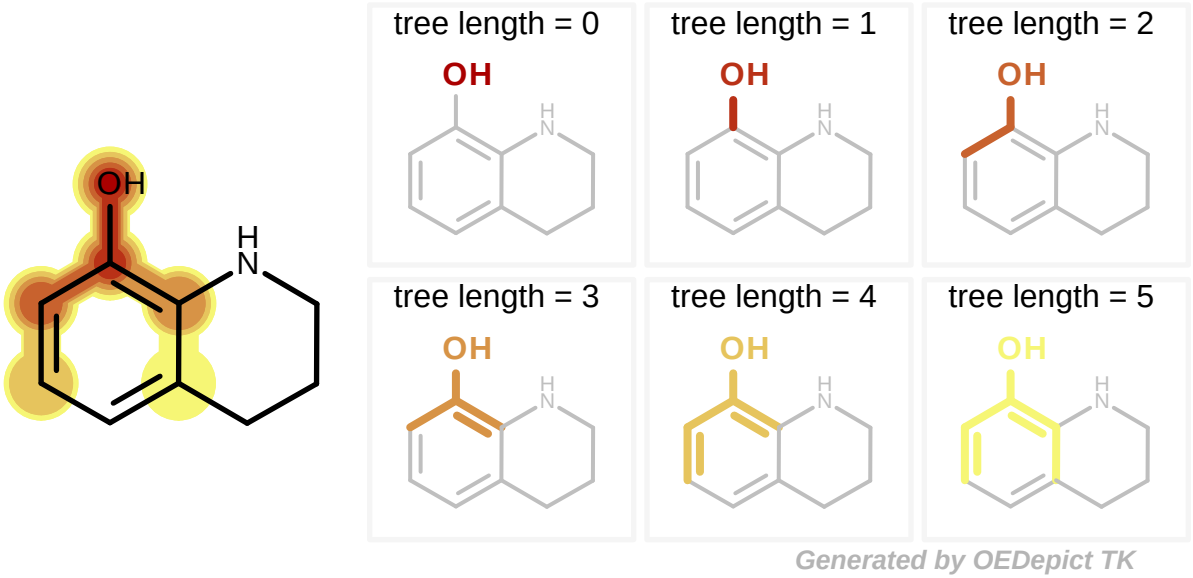


Fig. 7: **Example of enumerated tree fragments with increasing number of bonds**

In case of a circular fingerprint, the minimum and maximum radius of the enumerated fragments can be specified. See *Figure: Example of enumerated circular fragments with increasing radius*

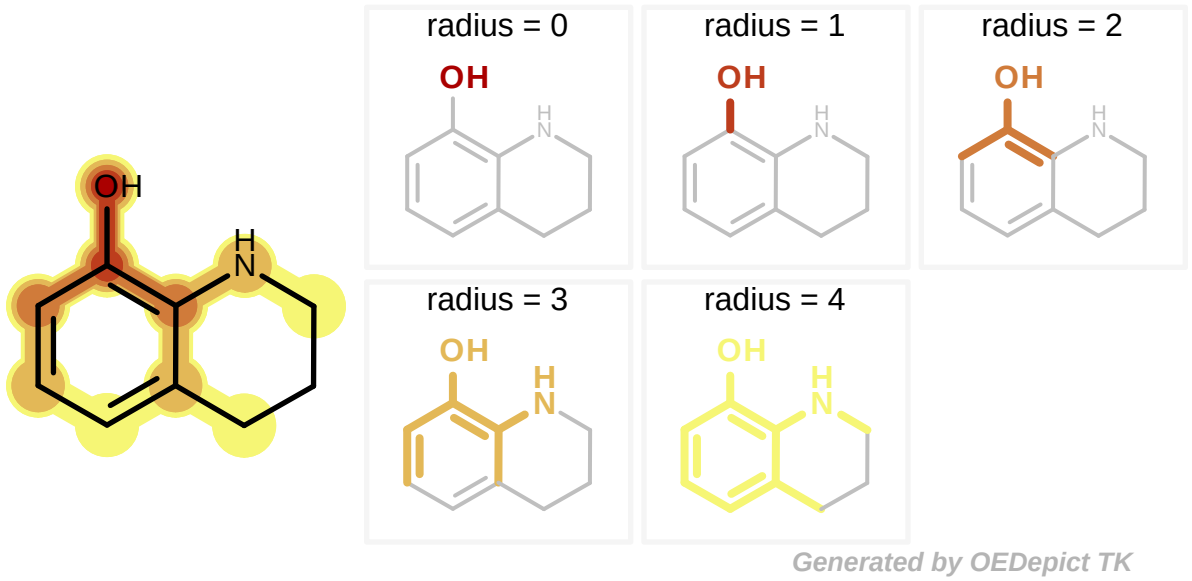


Fig. 8: **Example of enumerated circular fragments with increasing radius**

For example, when generating a fingerprint of the molecule shown in *Figure: Example Molecule* with minimum and maximum length set to `0` and `3`, respectively, only paths listed in the first four rows in *Table: Enumerated Paths*, are encoded into the fingerprint.

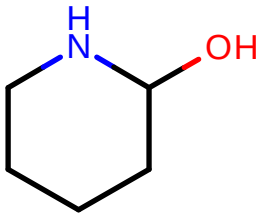


Fig. 9: **Example molecule**

Table 6: Enumerated paths

Path length (in bonds)	Generated Unique Paths
0	`C`, `N`, `O`
1	`C-C`, `C-N`, `C-O`
2	`C-C-C`, `C-C-N`, `C-C-O`, `C-N-C`, `N-C-O`
3	`C-C-C-C`, `C-C-C-N`, `C-C-C-O`, `C-C-N-C`, `C-N-C-O`,
4	`C-C-C-C-C`, `C-C-C-C-N`, `C-C-C-C-O`, `C-C-C-N-C`, `C-C-N-C-C`, `O-C-N-C-C`
5	`C-C-C-C-C-N`, `C-C-C-C-C-O`, `C-C-C-C-N-C`, `C-C-C-N-C-C`, `C-C-C-N-C-O`

Figure: Example of enumerated paths depicts the six unique paths of length four that are generated for the example molecule. Each unique path is encoded only once without considering its frequency.

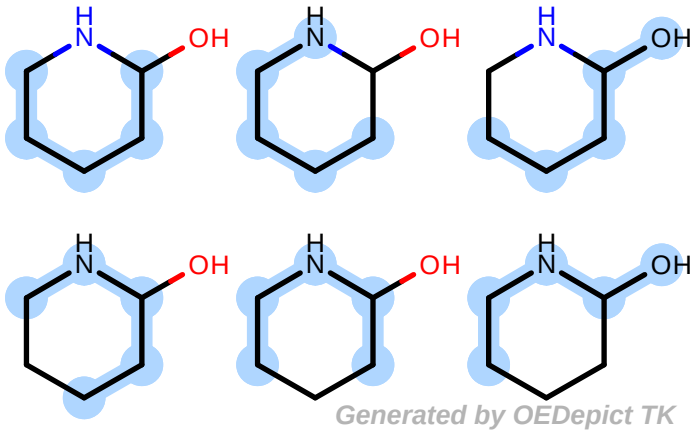


Fig. 10: **Example of enumerated paths**

In the example shown in *Listing 16*, fingerprints with various minimum and maximum path length are generated for pyrrole and pyridine. When enumerating only paths that are shorter than four bonds, the fingerprints generated for the two molecules are identical. Since the four bond-length pattern `ccccc` is present in pyridine but not in pyrrole, the fingerprints become different, resulting in a smaller Tanimoto similarity score.

Listing 16: Similarity calculation with various path lengths

```
public class FPPathLength
{
    public static int Main(string[] args)
    {
        OEGraphMol molA = new OEGraphMol();
        OEChem.OESmilesToMol(molA, "c1ccncc1");
        OEGraphMol molB = new OEGraphMol();
        OEChem.OESmilesToMol(molB, "c1cc[nH]c1");

        PrintTanimoto(molA, molB, 0, 3);
        PrintTanimoto(molA, molB, 1, 3);
        PrintTanimoto(molA, molB, 0, 4);
        PrintTanimoto(molA, molB, 0, 5);
        return 0;
    }

    private static void PrintTanimoto(OEMolBase molA, OEMolBase molB, uint minb, uint␣
↪maxb)
    {
        OEFingerPrint fpA = new OEFingerPrint();
        OEFingerPrint fpB = new OEFingerPrint();
        uint numbits = 2048;
        uint atype = OEFPAtomType.DefaultAtom;
        uint btype = OEFPBondType.DefaultBond;
        OEGraphSim.OEMakePathFP(fpA, molA, numbits, minb, maxb, atype, btype);
        OEGraphSim.OEMakePathFP(fpB, molB, numbits, minb, maxb, atype, btype);
        Console.WriteLine("Tanimoto(A,B) = {0:0.000}", OEGraphSim.OETanimoto(fpA, fpB));
    }
```

(continues on next page)

(continued from previous page)

```
}
```

The output of *Listing 16* is the following:

```
Tanimoto(A,B) = 1.000
Tanimoto(A,B) = 1.000
Tanimoto(A,B) = 0.950
Tanimoto(A,B) = 0.731
```

2.6.3 Fingerprint Size

The previous sections explain how the atom and bond typing and encoded fragment size can effect the similarity scores. Selecting an adequate fingerprint size is also very crucial. The number of unique circular, path or tree fragments present in molecular structures can be extremely large, therefore the generated fragments have to be hashed into the fixed-length fingerprint. This means that a bit in a fingerprint does not correspond to a unique pattern exclusively (as it does in structural key). Also a bit has no particular structural meaning, *i.e.*, each bit represents the presence of a number of structural patterns.

The smaller the size of the fingerprints, the more dense they become, raising the probability of collisions. A collision occurs when different fragments are mapped to the same bit. This will inherently result in information loss and weaken the power to discriminate between structurally similar and dissimilar molecules. On the other hand, when the size of the fingerprints is too large they become very sparse, which will reduce information loss. However, the time spent to calculate similarity scores will increase.

The following table shows the number of unique paths generated for benzylpenicillin (depicted in *Figure: Benzylpenicillin*).

Note

The more atom and bond properties that are taken into account and the larger the size of paths to enumerate, the larger the size of the fingerprint has to be in order to encode the enumerated fragments without a significant number of bit collisions.

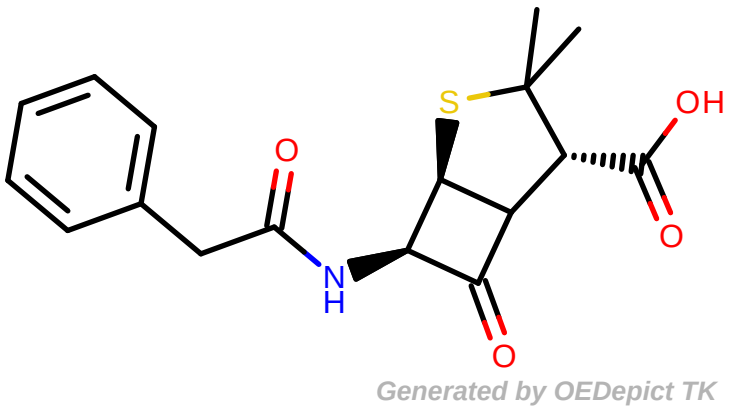


Fig. 11: **Benzylpenicillin**

Table 7: Number of unique paths generated for Benzylpenicillin

Atom/Bond typing	path 0-3	path 0-5	path 0-7
AtomicNumber, *BondOrder*	56	149	297
AtomicNumber | *HvyDegree*, *BondOrder*	111	265	453
AtomicNumber | *HvyDegree* | *Aromaticity*, *BondOrder* | *InRing*	126	297	499
DefaultAtom, *DefaultBond*	147	362	617

2.7 Fingerprint Coverage

Fingerprints are usually generated by enumerating various fragments (patterns) of a molecule and then hashing them into a fixed-length bitvector. The *OEGetFPCoverage* function provides access to these fragments by returning an iterator over OEAtomBondSet objects, each of which storing the atoms and bonds of a specific fragment.

See also

- *Fingerprint Patterns* chapter that shows how to access more information about the fragments enumerated during the fingerprint generation

The following example shows how the retrieve the unique fragments that are enumerated when generating a path fingerprint. The obtained fragments are depicted in *Table: Example of path fragments*

Listing 17: Example of accessing patterns encoded into a fingerprint

```
public class FPCoverage
{
    public static int Main(string[] args)
    {
        OEGraphMol mol = new OEGraphMol();
        OEChem.OESmilesToMol(mol, "CCNCC");

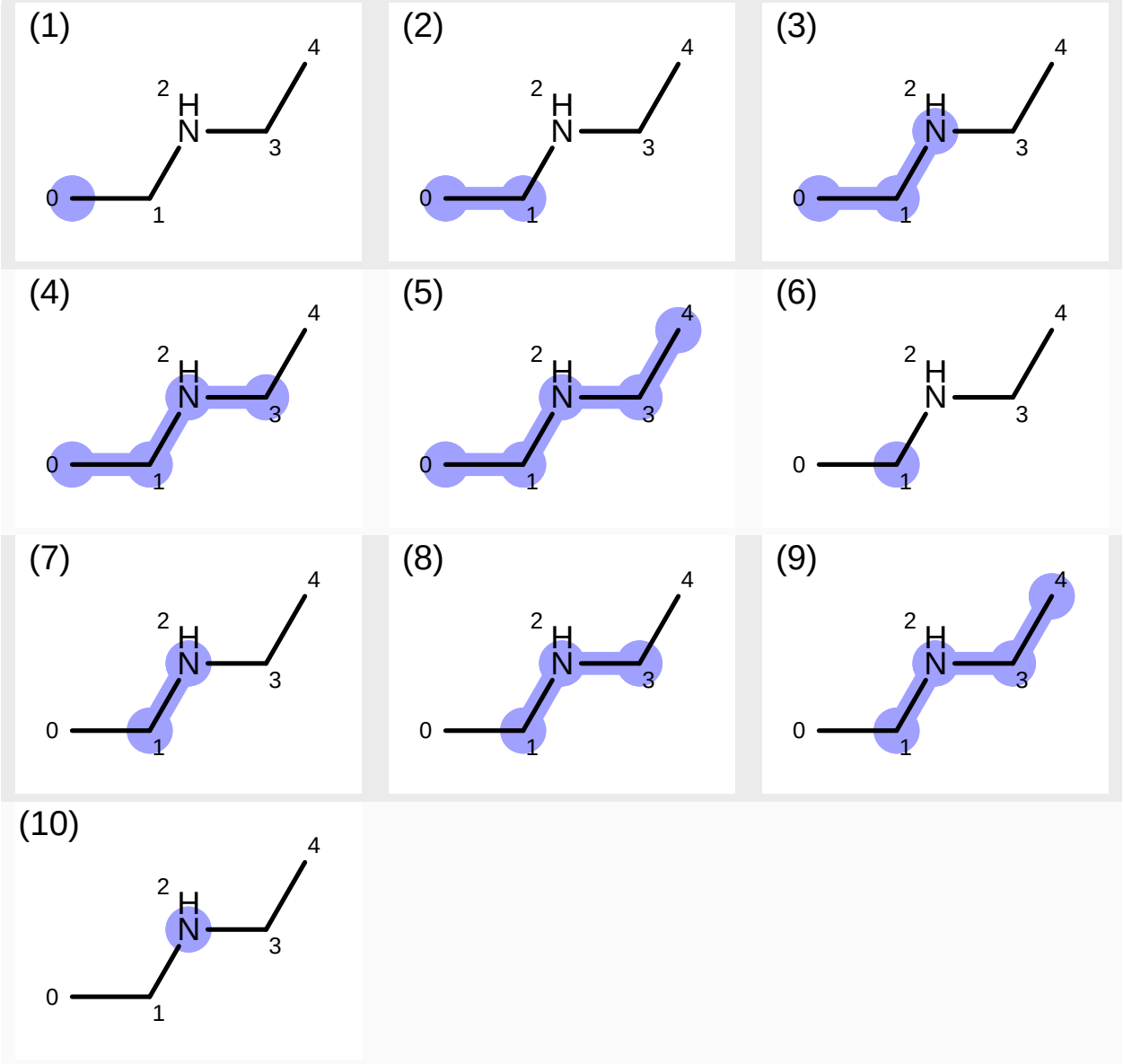
        OEFPTypeBase fptype = OEGraphSim.OEGetFPType(OEFPType.Path);

        bool unique = true;
        uint idx = 0;
        foreach (OEAtomBondSet abset in OEGraphSim.OEGetFPCoverage(mol, fptype, unique))
        {
            idx++;
            Console.Write("{0,2} ", idx);
            foreach (OEAtomBase a in abset.GetAtoms())
            {
                Console.Write(" {0} {1}", a.GetIdx(), OEChem.OEGetAtomicSymbol(a.
↪GetAtomicNum()));
            }
            Console.WriteLine();
        }
        return 0;
    }
}
```

The output of *Listing 17* is the following:

```
 1  0 C
 2  0 C  1 C
 3  0 C  1 C  2 N
 4  0 C  1 C  2 N  3 C
 5  0 C  1 C  2 N  3 C  4 C
 6  1 C
 7  1 C  2 N
 8  1 C  2 N  3 C
 9  1 C  2 N  3 C  4 C
10  2 N
```

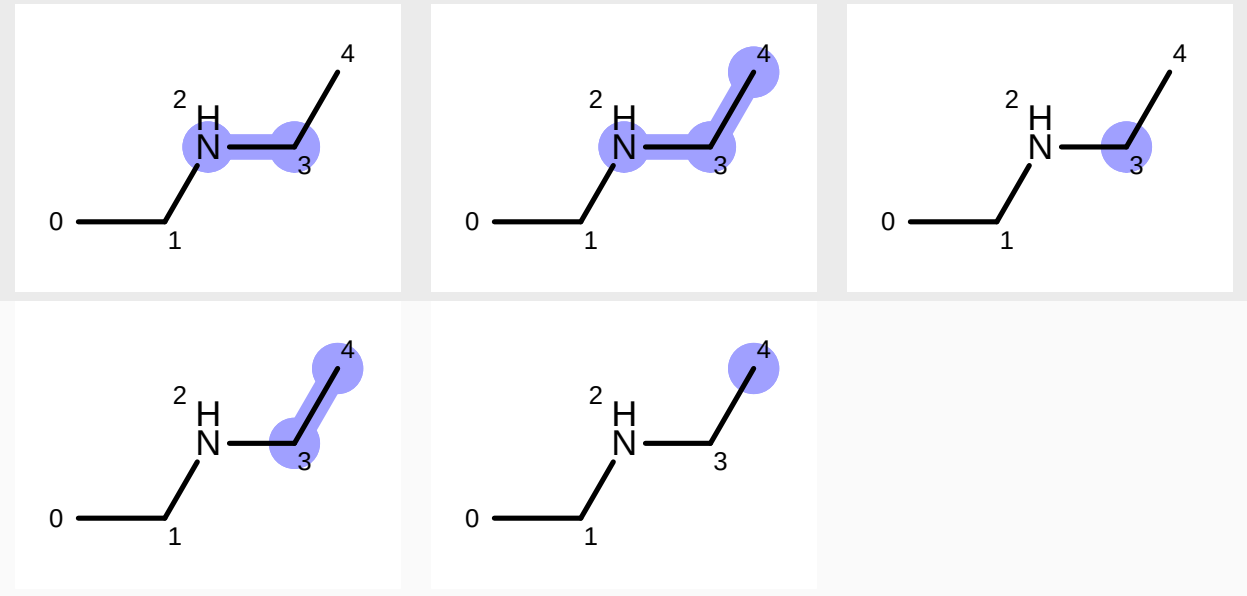
Table 8: **Example of unique path fragments.** The numbers displayed next to atoms are the atom indices.



The *OEGetFPCoverage* function in the *Listing 17* example is called with a unique options. This means that it returns

only unique fragments, where a fragment (*i.e.* subgraph) is considered unique, if it differs from all other subgraphs identified previously by at least one atom or bond. For example, executing the same code with a non-unique option would generate five additional paths depicted in *Table: Example of additional non-unique path fragments*

Table 9: **Example of additional non-unique path fragments.** The numbers displayed next to atoms are the atom indices.



See also

- *OEGetFPOverlap* function
- *Fingerprint Overlap* chapter

2.8 Fingerprint Overlap

The *OEGetFPOverlap* function provides access to the fragments of two molecules that are considered equivalent based on a specific fingerprint type. This means that the returned fragment-pairs set the same bit "on" when fingerprints are generated. The following example shows how to retrieve the common five bond-length patterns of two molecules.

Listing 18: Example of accessing common patterns based on a fingerprint

```
public class FPOverlap
{
    public static int Main(string[] args)
    {
        OEGraphMol pmol = new OEGraphMol();
        OEChem.OESmilesToMol(pmol, "c1cnc2c(c1)CC(CC2O)CF");

        OEGraphMol tmol = new OEGraphMol();
        OEChem.OESmilesToMol(tmol, "c1cc2c(cc1)CC(CCl)CC2N");

        OEFPTypeBase fptype = OEGraphSim.OEGetFPType("Tree,ver=2.0.0,size=4096,bonds=5-5,
```

(continues on next page)

(continued from previous page)

```
↪atype=AtmNum|HvyDeg|EqHalo,btype=Order");

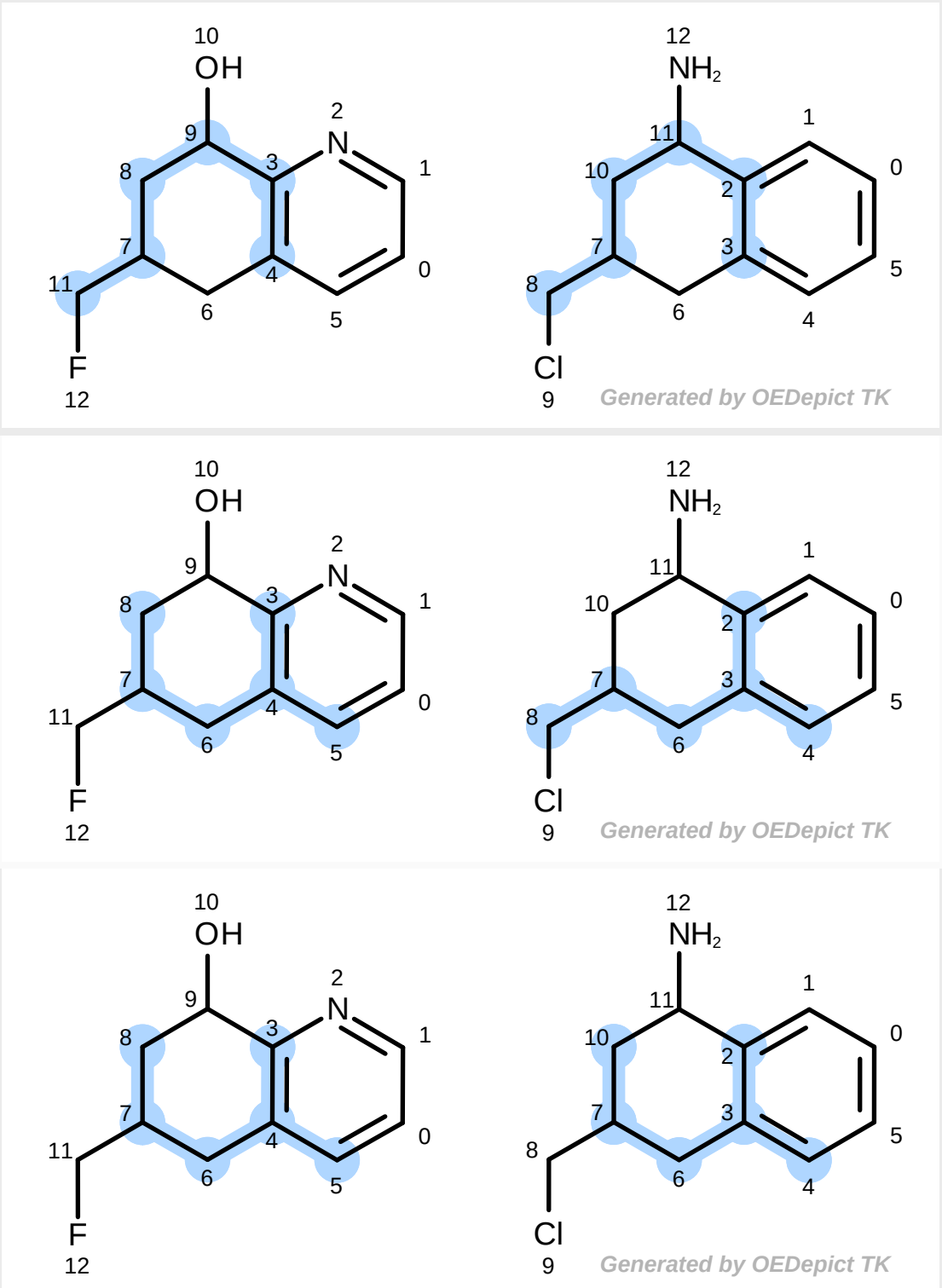
        int idx = 0;
        foreach (OEMatchBase match in OEGraphSim.OEGetFPOverlap(pmol, tmol, fptype))
        {
            idx++;
            Console.Write("match {0,2}:", idx);
            foreach (OEMatchPairAtom mpair in match.GetAtoms())
            {
                Console.Write(" {0}{1}-{2}{3}", mpair.pattern.GetIdx(), OEChem.
↪OEGetAtomicSymbol(mpair.pattern.GetAtomicNum()),
                              mpair.target.GetIdx(), OEChem.OEGetAtomicSymbol(mpair.
↪target.GetAtomicNum()));
            }
            Console.WriteLine();
        }
        return 0;
    }
}
```

The first three matches returned by the *Listing 18* are depicted in the next table. The output of code is the following:

```
match  1: 3C-2C  9C-11C  4C-3C  8C-10C  7C-7C  11C-8C
match  2: 3C-2C  4C-3C  5C-4C  6C-6C  7C-7C  8C-8C
match  3: 3C-2C  4C-3C  5C-4C  6C-6C  7C-7C  8C-10C
match  4: 3C-2C  4C-3C  5C-4C  6C-6C  7C-7C  11C-8C
match  5: 3C-2C  4C-3C  5C-4C  6C-6C  7C-7C  11C-10C
match  6: 3C-2C  9C-11C  4C-3C  6C-6C  7C-7C  11C-8C

... truncated  ...
```

Table 10: **Example of matches returned by the OEGetFPOverlap function.** The numbers depicted next to the atoms are the atom indices.



Warning

Even though the *OEGetFPOverlap* function returns an iterator of OEMatchBase objects, they are not matches in the traditional sense, i.e. the atom-pair and bond-pair correspondences between the pattern and the target atoms and bonds are not guaranteed. See example depicted below.

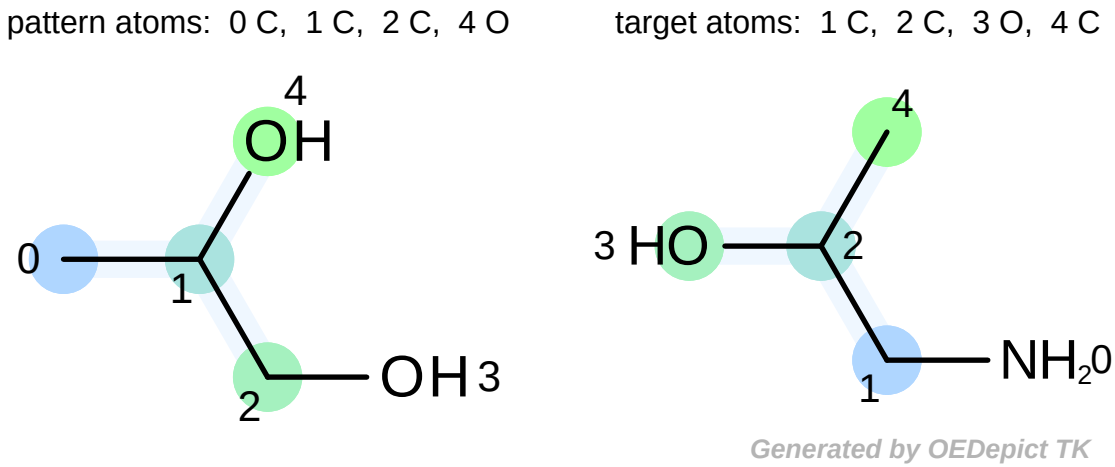


Fig. 12: **The two highlighted patterns set the same bit when fingerprints are generated, but the returned match is not pairwise**. The numbers depicted next to the atoms are the atom indices.

See also

- *OEGetFPCoverage* function
- *Fingerprint Coverage* chapter

Hint

The *OEGetFPOverlap* can be used to visualize molecule similarity based on a given fingerprint type. See more details in:

- *Visualizing Molecule Similarity* section
- Depicting Molecule Similarity section in OpenEye Python Cookbook

2.9 Fingerprint Patterns

Fingerprints are usually generated by enumerating various fragments (patterns) of a molecule and then hashing them into a fixed-length bitvector. The *OEGetFPPatterns* function provides access to these patterns by returning an iterator over *OEFPPattern* objects, each of which has the atoms and bonds of a specific fragment along with the following information:

- the bit that is set when the pattern is hashed into a fixed-length bitvector
- the unhashed unique integer encoding of the pattern

- the SMARTS representation of the pattern that encodes the atom and bond properties that are specified in the given fingerprint type (*OEFPTypeBase*)

The following example shows how the retrieve the patterns that are enumerated when generating a tree fingerprint.

Listing 19: Example of accessing patterns encoded into a fingerprint

```
public class FPPattern
{
    public static int Main(string[] args)
    {
        OEGraphMol mol = new OEGraphMol();
        OEChem.OESmilesToMol(mol, "NCC(=O)[O-]");

        OEFPTypeBase fptype = OEGraphSim.OEGetFPType("Tree,ver=2.0.0,size=4096,bonds=0-4,
↪atype=AtmNum|Arom|FCharge|HvyDeg,btype=Order");

        uint idx = 0;
        foreach (OEFPPattern pattern in OEGraphSim.OEGetFPPatterns(mol, fptype))
        {
            idx++;
            Console.Write("{0,2} {1,5} {2,50}", idx, pattern.GetBit(), pattern.
↪GetSmarts());
            foreach (OEAtomBase a in pattern.GetAtoms())
            {
                Console.Write(" {0} {1}", a.GetIdx(), OEChem.OEGetAtomicSymbol(a.
↪GetAtomicNum()));
            }
            Console.WriteLine();
        }
        return 0;
    }
}
```

The output of *Listing 19* is the following:

```
 1   849                                                [N;D1]  0 N
 2  1611                                                [C;D2]  1 C
 3   740                                                [C;D3]  2 C
 4  2841                                                [O;D1]  3 O
 5  1589                                               [O-;D1]  4 O
 6  2326                                     [N;D1]-[C;D2]  0 N  1 C
 7  3764                           [C;D2](-[C;D3])-[N;D1]  0 N  1 C  2 C
 8  2727                 [C;D2](-[C;D3]=[O;D1])-[N;D1]  0 N  1 C  2 C  3 O
 9   859           [C;D2](-[C;D3](-[O-;D1])=[O;D1])-[N;D1]  0 N  1 C  2 C  3 O  4 O
10  3537                 [C;D2](-[C;D3]-[O-;D1])-[N;D1]  0 N  1 C  2 C  4 O
11  3083                                     [C;D3]-[C;D2]  1 C  2 C
12   793                           [C;D2]-[C;D3]=[O;D1]  1 C  2 C  3 O
13  1574                 [C;D2]-[C;D3](-[O-;D1])=[O;D1]  1 C  2 C  3 O  4 O
14  3538                           [C;D2]-[C;D3]-[O-;D1]  1 C  2 C  4 O
15  1707                                     [O;D1]=[C;D3]  2 C  3 O
16  1472                           [C;D3](-[O-;D1])=[O;D1]  2 C  3 O  4 O
17  1397                                     [O-;D1]-[C;D3]  2 C  4 O
```

See also

- *OEFPPattern* class
- *OEGetFPPatterns* function
- *OEGetFPType* function
- *Fingerprint Coverage* chapter
- *Fingerprint Overlap* chapter

CHAPTER

THREE

GPU FAST FINGERPRINTS

3.1 Prerequisites

Fast fingerprint searching in **Graphsim TK** is GPU-enabled. In order to run searches on a GPU please refer to the guidelines below:

3.1.1 GPU-Related Requirements

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

Supported Platforms

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

Supported GPUs

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

NVIDIA Drivers

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

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

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

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

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

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

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

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

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

Performance Tuning

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

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

This will automatically enable persistence mode after reboot.

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

3.2 Usage

An example of how to perform fast fingerprint searches can be found at *Searching fast fingerprint database*. To run using a GPU the *OEFastFPDatabaseMemoryType.CUDA* constant is used to construct the *OEFastFPDatabase* object. If using the example script aforementioned, simply run the script with `-memorytype CUDA`.

> **See also**
>
> - *OEFastFPDatabaseMemoryType* namespace
> - *OEFastFPDatabase* class
> - *OEFPDatabaseOptions* class

CHAPTER

FOUR

EXAMPLES

4.1 GraphSim TK Examples

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

Program	Description
SearchFP	building and searching fingerprint database
MakeFastFP	generating fingerprint file for fast fingerprint search
SearchFastFP	searching fast fingerprint database

Examples:

4.1.1 Building and searching fingerprint database

A program that builds a fingerprint database on-the-fly and then performs fingerprint search to identify molecules which are similar to a given query molecule

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> SearchFP --help
```

will generate the following output:

```
Simple parameter list

 fingerprint database search options
   -cutoff : Similarity cutoff value
   -descending : Order of similarity scores
   -limit : Maximum number of similarity scores
   -simfunc : Similarity measure

 fingerprint options
   -atomtype : Fingerprint atom type
   -bondtype : Fingerprint bond type
   -fptype : Fingerprint type
   -maxdistance : Maximum number of bonds/radius in path/circular/tree
   -mindistance : Minimum number of bonds/radius in path/circular/tree
   -numbits : Size of bitvector
```

(continues on next page)

(continued from previous page)

```
 input/output options
   -molfname : Input molecule filename
   -out : Output molecule filename
   -query : Input query filename
```

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.
*/
/*****************************************************************************
 * Searching fingerprint database
 *****************************************************************************/

using System;
using OpenEye.OEChem;
using OpenEye.OEGraphSim;

public class SearchFP
{
    public static int Main(string[] args)
    {
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, interfaceData);

        OEFPDatabaseOptions defopts = new OEFPDatabaseOptions(10, OESimMeasure.Tanimoto);
        OEGraphSim.OEConfigureFPDatabaseOptions(itf, defopts);
        OEGraphSim.OEConfigureFingerPrint(itf, OEGraphSim.OEGetFPType(OEFPType.Tree));

        if (!OEChem.OEParseCommandLine(itf, args, "SearchFP")) {
            OEChem.OEThrow.Fatal("Unable to interpret command line!");
        }

        String qfname = itf.GetString("-query");
        String mfname = itf.GetString("-molfname");
```

(continues on next page)

(continued from previous page)

```
        String ofname = itf.GetString("-out");

        // initialize databases

        OEWallTimer timer = new OEWallTimer();
        timer.Start();

        oemolistream ifs = new oemolistream();
        if (!ifs.open(qfname))
            OEChem.OEThrow.Fatal("Cannot open input file!");

        OEGraphMol query = new OEGraphMol();
        if (!OEChem.OEReadMolecule(ifs, query))
            OEChem.OEThrow.Fatal("Cannot read query molecule!");

        OEMolDatabase moldb = new OEMolDatabase();
        if (!moldb.Open(mfname))
            OEChem.OEThrow.Fatal("Cannot open molecule database!");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(ofname))
            OEChem.OEThrow.Fatal("Cannot open output file!");

        OEFPTypeBase fptype = OEGraphSim.OESetupFingerPrint(itf);
        OEChem.OEThrow.Info("Using fingerprint type " + fptype.GetFPTypeString());
        OEFPDatabase fpdb = new OEFPDatabase(fptype);

        OEFingerPrint emptyfp = new OEFingerPrint();
        emptyfp.SetFPTypeBase(fptype);

        uint nrmols = moldb.GetMaxMolIdx();

        OEGraphMol mol = new OEGraphMol();
        for (uint idx = 0; idx < nrmols; ++idx) {
            if (moldb.GetMolecule(mol, idx))
                fpdb.AddFP(mol);
            else
                fpdb.AddFP(emptyfp);
        }

        uint nrfps = fpdb.NumFingerPrints();
        OEChem.OEThrow.Info(timer.Elapsed().ToString("##0.00") + " sec to initialize␣
↪databases");

        OEFPDatabaseOptions opts = new OEFPDatabaseOptions();
        OEGraphSim.OESetupFPDatabaseOptions(opts, itf);
        // search fingerprint database

        timer.Start();
        OESimScoreIter siter = fpdb.GetSortedScores(query, opts);
        OEChem.OEThrow.Info(timer.Elapsed().ToString("##0.00") + " sec to search " +␣
↪nrfps + " fingerprints");
```

(continues on next page)

(continued from previous page)

```
        timer.Start();
        OEGraphMol hit = new OEGraphMol();
        for (siter.ToFirst(); siter.IsValid(); siter.Increment()) {
            if (moldb.GetMolecule(hit, siter.Target().GetIdx()))
                OEChem.OEWriteMolecule(ofs, hit);
        }
        OEChem.OEThrow.Info(timer.Elapsed().ToString("##0.00") + " sec to write " + opts.
↪GetLimit() + " hits");
        return 0;
    }

    private static string interfaceData=@"
!BRIEF [-query] <molfile> [-molfname] <molfile> [-out] <molfile>

!CATEGORY ""input/output options""

  !PARAMETER -query
    !ALIAS -q
    !TYPE string
    !REQUIRED true
    !KEYLESS 1
    !VISIBILITY simple
    !BRIEF Input query filename
  !END

  !PARAMETER -molfname
    !ALIAS -mol
    !TYPE string
    !REQUIRED true
    !KEYLESS 2
    !VISIBILITY simple
    !BRIEF Input molecule filename
  !END

  !PARAMETER -out
    !ALIAS -o
    !TYPE string
    !REQUIRED true
    !KEYLESS 3
    !VISIBILITY simple
    !BRIEF Output molecule filename
  !END

!END
";
}
```

See also

- *OEFPDatabaseOptions* class

- *OEConfigureFPDatabaseOptions* and *OESetupFPDatabaseOptions* functions
- *OEConfigureFingerPrint* and *OESetupFingerPrint* functions
- OEMolDatabase class in **OEChem TK** manual
- *OEFPDatabase* class

4.1.2 Generating fingerprint file for fast fingerprint search

A program that generates a binary fingerprint file for fast fingerprint search.

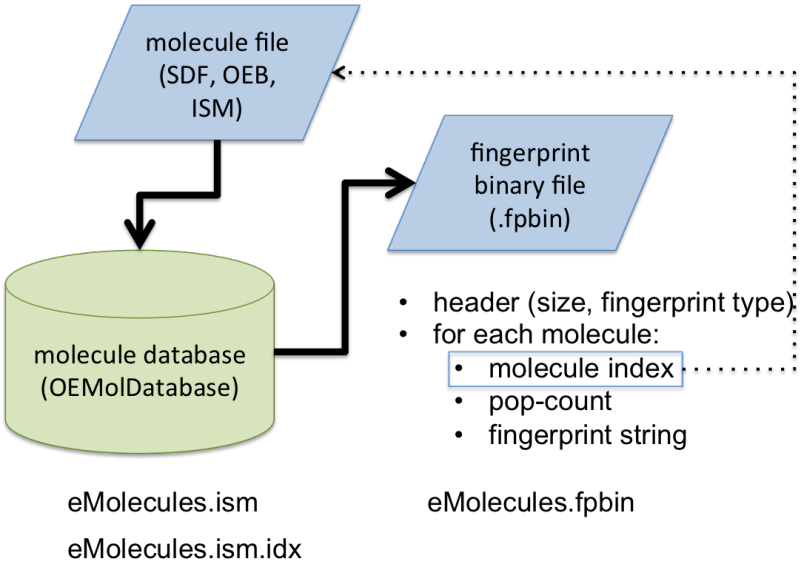


Fig. 1: **Schematic representation of fast fingerprint generation process**

See also

- Example code in the *Searching fast fingerprint database* section
- Rapid Similarity Searching of Large Molecule File OpenEye Python Cookbook example

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> MakeFastFP --help
```

will generate the following output:

```
Simple parameter list

 fingerprint options
   -atomtype : Fingerprint atom type
   -bondtype : Fingerprint bond type
   -fptype : Fingerprint type
```

(continues on next page)

(continued from previous page)

```
   -maxdistance : Maximum number of bonds/radius in path/circular/tree
   -mindistance : Minimum number of bonds/radius in path/circular/tree
   -numbits : Size of bitvector

 input/output options
   -fpdbfname : Output fingerprint database filename
   -in : Input molecule filename
```

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.
*/
/*****************************************************************************
 * Generates binary fingerprint file for fast fingerprint search
 *****************************************************************************/

using System;
using System.IO;
using OpenEye.OEChem;
using OpenEye.OEGraphSim;

public class MakeFastFP
{
    public static void Main(string[] args)
    {
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, interfaceData);
        OEGraphSim.OEConfigureFingerPrint(itf, OEGraphSim.OEGetFPType(OEFPType.Tree));

        if (!OEChem.OEParseCommandLine(itf, args, "MakeFastFP"))
        {
            OEChem.OEThrow.Fatal("Unable to interpret command line!");
        }

        String ifname = itf.GetString("-in");
```

(continues on next page)

(continued from previous page)

```
        String ffname = itf.GetString("-fpdbfname");

        String ext = OEChem.OEGetFileExtension(ffname);
        if (!ext.Equals("fpbin"))
            OEChem.OEThrow.Fatal("Fingerprint database file should have '.fpbin' file␣
↪extension!");

        String idxfname = OEChem.OEGetMolDatabaseIdxFileName(ifname);

        if (!File.Exists(idxfname)) {
            if (!OEChem.OECreateMolDatabaseIdx(ifname))
                OEChem.OEThrow.Warning("Unable to create " + idxfname + "  molecule␣
↪index file");
        }
        OEChem.OEThrow.Info("Using " + idxfname + " index molecule file");

        OEMolDatabase moldb = new OEMolDatabase();
        if (!moldb.Open(ifname))
            OEChem.OEThrow.Fatal("Cannot open molecule database file!");

        uint nrmols = moldb.GetMaxMolIdx();

        OEFPTypeBase fptype = OEGraphSim.OESetupFingerPrint(itf);
        OEChem.OEThrow.Info("Using fingerprint type " + fptype.GetFPTypeString());

        OECreateFastFPDatabaseOptions opts = new OECreateFastFPDatabaseOptions(fptype);
        opts.SetTracer(new OEDots(100000, 1000, "fingerprints"));
        OEChem.OEThrow.Info("Generating fingerprints with " + opts.GetNumProcessors() +
↪" threads");

        OEWallTimer timer = new OEWallTimer();

        if (!OEGraphSim.OECreateFastFPDatabaseFile(ffname, ifname, opts))
            OEChem.OEThrow.Fatal("Cannot create fingerprint database file!");

        OEChem.OEThrow.Info(String.Format("{0:F2}", timer.Elapsed()) + " sec to generate
↪" + nrmols + " fingerprints");
    }

    private static string interfaceData=@"
!BRIEF [-in] <input> [-fpdbfname] <output>

!CATEGORY ""input/output options""

  !PARAMETER -in
    !ALIAS -i
    !TYPE string
    !REQUIRED true
    !KEYLESS 1
    !VISIBILITY simple
    !BRIEF Input molecule filename
  !END
```

(continues on next page)

(continued from previous page)

```
  !PARAMETER -fpdbfname
    !ALIAS -fpdb
    !TYPE string
    !REQUIRED true
    !KEYLESS 2
    !VISIBILITY simple
    !BRIEF Output fingerprint database filename
  !END

!END
";
}
```

See also

- *OEConfigureFingerPrint* and *OESetupFingerPrint* functions
- `OEGetMolDatabaseIdxFileName` and `OECreateMolDatabaseIdx` functions in **OEChem TK** manual
- OEMolDatabase class in **OEChem TK** manual
- *OECreateFastFPDatabaseOptions* class
- *OECreateFastFPDatabaseFile* function

4.1.3 Searching fast fingerprint database

A program that initializes a fast fingerprint database with the fingerprint file generated from the previous example and then performs a fingerprint search to identify molecules which are similar to a given query molecule.

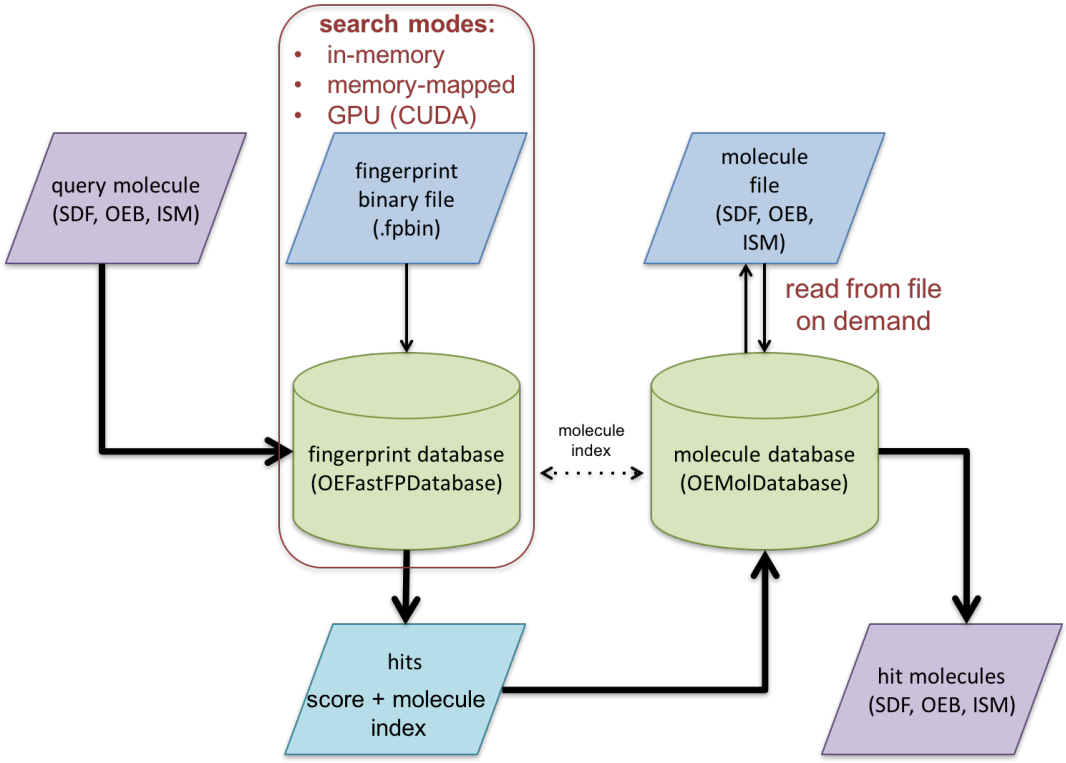


Fig. 2: **Schematic representation of fast fingerprint search process**

See also

- Example code in the *Generating fingerprint file for fast fingerprint search* section
- Rapid Similarity Searching of Large Molecule File OpenEye Python Cookbook example

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> SearchFastFP --help
```

will generate the following output:

```
Simple parameter list

 fingerprint database options
   -memorytype : Fingerprint database memory type

 fingerprint database search options
   -cutoff : Similarity cutoff value
   -descending : Order of similarity scores
   -limit : Maximum number of similarity scores
   -simfunc : Similarity measure
```

(continues on next page)

(continued from previous page)

```
 input/output options
   -fpdbfname : Input fast fingerprint database filename
   -molfname : Input molecule filename
   -out : Output molecule filename
   -query : Input query filename
```

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.
*/
/*****************************************************************************
 * Searching fast fingerprint database
 *****************************************************************************/

using System;
using OpenEye.OEChem;
using OpenEye.OEGraphSim;

public class SearchFastFP
{
    public static int Main(string[] args)
    {
        OEInterface itf = new OEInterface();
        OEChem.OEConfigure(itf, interfaceData);

        OEFPDatabaseOptions defopts = new OEFPDatabaseOptions(10, OESimMeasure.Tanimoto);
        OEGraphSim.OEConfigureFPDatabaseOptions(itf, defopts);
        OEGraphSim.OEConfigureFPDatabaseMemoryType(itf);

        if (!OEChem.OEParseCommandLine(itf, args, "SearchFastFP"))
        {
            OEChem.OEThrow.Fatal("Unable to interpret command line!");
        }

        String qfname = itf.GetString("-query");
```

(continues on next page)

(continued from previous page)

```
        String mfname = itf.GetString("-molfname");
        String ffname = itf.GetString("-fpdbfname");
        String ofname = itf.GetString("-out");

        // initialize databases

        OEWallTimer timer = new OEWallTimer();
        timer.Start();

        oemolistream ifs = new oemolistream();
        if (!ifs.open(qfname))
            OEChem.OEThrow.Fatal("Cannot open input file!");

        OEGraphMol query = new OEGraphMol();
        if (!OEChem.OEReadMolecule(ifs, query))
            OEChem.OEThrow.Fatal("Cannot read query molecule!");

        OEMolDatabase moldb = new OEMolDatabase();
        if (!moldb.Open(mfname))
            OEChem.OEThrow.Fatal("Cannot open molecule database!");

        uint memtype = OEGraphSim.OEGetFPDatabaseMemoryType(itf);

        OEFastFPDatabase fpdb = new OEFastFPDatabase(ffname, memtype);
        if (!fpdb.IsValid())
            OEChem.OEThrow.Fatal("Cannot open fingerprint database!");
        long nrfps = fpdb.NumFingerPrints();
        String memtypestr = fpdb.GetMemoryTypeString();

        OEFPTypeBase fptype = fpdb.GetFPTypeBase();
        OEChem.OEThrow.Info("Using fingerprint type " + fptype.GetFPTypeString());

        oemolostream ofs = new oemolostream();
        if (!ofs.open(ofname))
            OEChem.OEThrow.Fatal("Cannot open output file!");

        if (!OEGraphSim.OEAreCompatibleDatabases(moldb, fpdb))
            OEChem.OEThrow.Fatal("Databases are not compatible!");

        OEChem.OEThrow.Info(timer.Elapsed().ToString("##0.00") + " sec to initialize␣
↪databases");

        OEFPDatabaseOptions opts = new OEFPDatabaseOptions();
        OEGraphSim.OESetupFPDatabaseOptions(opts, itf);

        // search fingerprint database

        timer.Start();
        OESimScoreIter siter = fpdb.GetSortedScores(query, opts);
        OEChem.OEThrow.Info(timer.Elapsed().ToString("##0.00") + " sec to search " +
                            nrfps + " fingerprints " +  memtypestr);
```

(continues on next page)

(continued from previous page)

```
        timer.Start();
        uint nrhits = 0;
        OEGraphMol hit = new OEGraphMol();
        for (siter.ToFirst(); siter.IsValid(); siter.Increment())
        {
            if (moldb.GetMolecule(hit, siter.Target().GetIdx()))
            {
                nrhits += 1;
                OEChem.OESetSDData(hit, "Similarity score", String.Format("{0:F3}",␣
↪siter.Target().GetScore()));
                OEChem.OEWriteMolecule(ofs, hit);
            }
        }
        OEChem.OEThrow.Info(timer.Elapsed().ToString("##0.00") + " sec to write " +␣
↪nrhits + " hits");
        return 0;
    }

    private static string interfaceData=@"
!BRIEF [-query] <molfile> [-molfname] <molfile> [-fpdbfname] <fpfile>  [-out] <molfile>

!CATEGORY ""input/output options""

  !PARAMETER -query
    !ALIAS -q
    !TYPE string
    !REQUIRED true
    !KEYLESS 1
    !VISIBILITY simple
    !BRIEF Input query filename
  !END

  !PARAMETER -molfname
    !ALIAS -mol
    !TYPE string
    !REQUIRED true
    !KEYLESS 2
    !VISIBILITY simple
    !BRIEF Input molecule filename
  !END

  !PARAMETER -fpdbfname
    !ALIAS -fpdb
    !TYPE string
    !REQUIRED true
    !KEYLESS 3
    !VISIBILITY simple
    !BRIEF Input fast fingerprint database filename
  !END

  !PARAMETER -out
    !ALIAS -o
```

(continues on next page)

(continued from previous page)

```
    !TYPE string
    !REQUIRED true
    !KEYLESS 4
    !VISIBILITY simple
    !BRIEF Output molecule filename
  !END

!END
";
}
```

See also

- *OEFPDatabaseOptions* class
- *OEConfigureFPDatabaseOptions* and *OESetupFPDatabaseOptions* functions
- *OEConfigureFPDatabaseMemoryType* and *OEGetFPDatabaseMemoryType* function
- OEMolDatabase class in **OEChem TK** manual
- *OEFastFPDatabaseMemoryType* namespace
- *OEFastFPDatabase* class
- *OEAreCompatibleDatabases* function

4.2 Python Cookbook Examples

There are large number of code examples available in OpenEye Python Cookbook that use **OpenEye Toolkits** to solve a wide range of cheminformatics and molecular modeling problems.

- 2D Depiction chapter contains examples that illustrate how the **OEDepict TK** and **Grapheme TK** can be utilized to depict molecules and visualize properties calculated with other OpenEye toolkits.
- Visualizing 3D Information chapter contains examples that illustrate how the **Grapheme TK** can be utilized to project complex 3D information into the 2D molecular graph.
- Cheminformatics chapter contains examples that solve various cheminformatics problems such as similarity search, ring perception and manipulating molecular graphs.
- Peptide Informatics chapter contains examples that solve various peptide informatics problems such as generating custom monomer sets, parsing and generating HELM representations, and visualizing peptides.

OpenEye Python Cookbook examples using OEGraphSim TK

- Similarity Searching of Large Molecule Files
- Rapid Similarity Searching of Large Molecule File
- Finding Core Fragment of a Molecule Series
- Depicting Molecule Similarity Based on Fingerprints
- Drawing Fingerprint Score Histogram

CHAPTER

FIVE

API

5.1 OEGraphSim Classes

5.1.1 OECreateFastFPDatabaseOptions

```
class OECreateFastFPDatabaseOptions
```

Encapsulates the options that control how the binary fingerprint database file is generated when calling the *OECreateFastFPDatabaseFile* function.

See also

- *OECreateFastFPDatabaseFile* function
- Example code in *Generating fingerprint file for fast fingerprint search* section

Constructors

```
OECreateFastFPDatabaseOptions(const OEFPTypeBase* fptype)
```

Constructs an *OECreateFastFPDatabaseOptions* object with the given fingerprints type.

See also

- *OEFPTypeBase* class

Note

GraphSim TK currently only supports the popcount search method for fingerprints with the size of multiple of `256`. This means that the *`OEFPType.MACCS166`* fingerprint type is currently not supported. When using other custom fingerprint types (see *User-defined Fingerprint* section) the size of the fingerprint must be a multiple of `256` in order to be able to search it with the popcount method.

GetFPType

```
const OEFPTypeBase* GetFPType() const
```

Returns the fingerprint type in which the *OECreateFastFPDatabaseOptions* was constructed.

See also

- *OEFPTypeBase* class

GetNumProcessors

```
unsigned int GetNumProcessors() const
```

Returns the number of threads that are used to generate fingerprints when calling the *`OECreateFastFPDatabaseFile`* function.

See also

- *`OECreateFastFPDatabaseOptions.SetNumProcessors`* method

SetNumProcessors

```
void SetNumProcessors(unsigned int nrps)
```

Sets the number of threads that are used to generate fingerprints when calling the *`OECreateFastFPDatabaseFile`* function. When set to `0`, the number of processors used will be the number returned by the `OEGetNumProcessors` function.

SetTracer

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

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

See also

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

5.1.2 OEFastFPDatabaseParams

```
class OEFastFPDatabaseParams
```

Stores parameters retrieved from a binary fingerprint database file generated with the *OECreateFastFPDatabaseFile* function.

See also

- *OECreateFastFPDatabaseFile* function

Constructors

```
OEFastFPDatabaseParams(const std::string& fpdbfname)
```

Constructor that parses the header of the given fingerprint file.

GetFPTypeBase

```
const OEFPTypeBase* GetFPTypeBase() const
```

Returns the type of fingerprints stored in the file.

See also

OEFPTypeBase class

GetMolDatabaseFilename

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

Returns the name of the molecule file from which the fingerprint file has been generated.

IsValid

```
bool IsValid() const
```

Returns whether the header of the fingerprint file can be successfully parsed.

NumFingerPrints

```
size_t NumFingerPrints() const
```

Returns the number of fingerprints stored in the fingerprint file.

5.1.3 OEFastFPDatabase

```
class OEFastFPDatabase
```

The *OEFastFPDatabase* class is designed to perform rapid CUDA-accelerated, in-memory or memory-mapped fingerprint searches using the popcount method. Each *OEFastFPDatabase* object is associated with a fingerprint type (*OEFPTypeBase*) that is set when the database is initialized from a pre-generated binary fingerprint file.

For CUDA-accelerated fingerprint searching please see the *prerequisites for OpenEye's GPU-accelerated software*.

Note

GraphSim TK currently only supports the popcount search method for fingerprints with the size of multiple of `256`. This means that the *OEFastFPDatabase* class currently does not support:

- `OEFPType.MACCS166` fingerprint type
- `OEFPType.Lingo` fingerprint type
- other custom fingerprint types (see *User-defined Fingerprint* section) with size that is not a multiple of `256`

For the fingerprint types listed above, the original *OEFPDatabase* class can be utilized.

Note

OEFastFPDatabase gives identical results to *OEFPDatabase*. However, *OEFPDatabase* calculates similarity scores in single precision (float) while *OEFastFPDatabase* uses double precision. As a result, small similarity score differences can be observed.

See also

- *OEFPDatabase* class

Code Example

- Example code in the *Generating fingerprint file for fast fingerprint search* section
- Example code in the *Searching fast fingerprint database* section
- Rapid Similarity Searching of Large Molecule File **OpenEye Python Cookbook** recipe
- Depicting Molecule Similarity Based on Fingerprints **OpenEye Python Cookbook** recipe

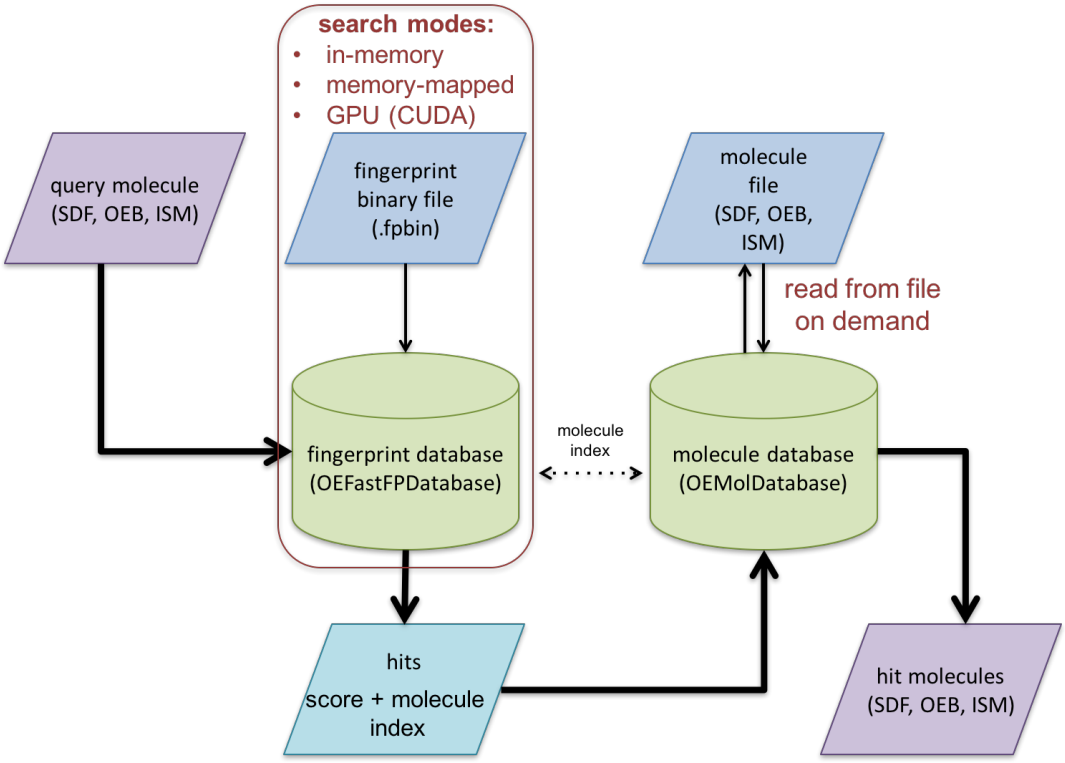


Fig. 1: **Schematic representation of fast fingerprint search process**

Constructors

```
OEFastFPDatabase(const std::string& dbfile,
                 unsigned int memtype = OEFastFPDatabaseMemoryType::Default)
```

Constructs an *OEFastFPDatabase* object.

dbfile
: The name of the file which contains the fingerprint data. The file has to be generated with the *OECreateFastFPDatabaseFile* function.

memtype
: Defines whether the fingerprints are pre-loaded into GPU-memory, CPU-memory, or memory-mapped during the search process. This value has to be from the *OEFastFPDatabaseMemoryType* namespace.

Note

If the *OEFastFPDatabase* object can not be initialized with the *OEFastFPDatabaseMemoryType.CUDA* option, the following warning message will be throw:

```
Warning: OEFastFPDatabase::OEFastFPDatabase() : no CUDA-enabled
device available falling back to memory-mapped type!
```

As a rule of thumb, 1 million finger prints requires 0.6GB of GPU memory. GPU memory can be queried using the `nvidia-smi` command from terminal.

See also

- *OEFastFPDatabaseMemoryType* namespace.

GetAllScores

```
OESystem::OEIterBase<float>* GetAllScores(const OEFPDatabaseOptions& opts) const
```

Performs NxN similarity calculations between all pairs of fingerprints stored in the *OEFastFPDatabase* object. It returns an iterator over the calculated similarity scores. The scores are not sorted, but returned as a flattened square matrix.

For all similarity measures other than *Tversky*, the returned 'matrix' will be symmetrical.

opts
: The *OEFPDatabaseOptions* object controls all the parameters that determine the search (i.e. similarity measure parameters). Cutoff and order parameters are ignored as the results are not filtered or sorted.

Note

This operation scales with n^2 in memory and can easily overwhelm the system memory for larger databases. Therefore, the *OEFastFPDatabase.GetRawScores* method returns a single row of the similarity matrix and is the recommended usage for querying large databases.

See also

- *OEFPDatabaseOptions* class

GetFingerPrint

```
bool GetFingerPrint(OEFingerPrint& fp, size_t idx) const
```

Returns the idx^{th} fingerprint of the database.

Warning

This function returns `false` if the fingerprint index is not identical to the corresponding molecule index. This can occur if the fingerprint binary file is generated in multi-thread process. If direct access to the fingerprint is required when using the *OEFastFPDatabase.GetFingerPrint* method, the fingerprint file should be generated in a single-threaded mode. This can be done by setting *SetNumProcessors(1)* for the option class used for creating the binary file.

See also

- *OEFingerPrint* class
- *OECreateFastFPDatabaseOptions.GetNumProcessors* method
- *OECreateFastFPDatabaseOptions.SetNumProcessors* method

GetFPTypeBase

```
const OEFPTypeBase* GetFPTypeBase() const
```

Returns the fingerprint type of the *OEFastFPDatabase* object. An *OEFastFPDatabase* object can only store fingerprints with identical types.

GetHistogram

```
OEFPHistogram GetHistogram(const OEFPDatabaseOptions& opts,
                           const size_t nrbins = 200u) const
```

Performs similarity calculations between all pairs of fingerprints stored in the *OEFastFPDatabase* object. It returns the histogram over the calculated similarity scores in a *OEFPHistogram* object.

opts
: The *OEFPDatabaseOptions* object contains the settings available to control the search (i.e. similarity measure, α and β parameters for *Tversky* similarity). Cutoff and order parameters are ignored as the results are not filtered or sorted.

nrbins
: Number of bins in the returned *OEFPHistogram* object.

Note

For all similarity measures other than *Tversky*, the histogram only contains the upper-triangular similarity scores (excluding the diagonal). In case of the asymmetric *Tversky* similarity measure, the histogram of the whole NxN matrix is returned.

Note

When the *OEFastFPDatabase* object is initialized with *OEFastFPDatabaseMemoryType.CUDA*, `nrbins` is limited to at most 1024.

Hint

This method calculates similarities identically to *OEFastFPDatabase.GetAllScores* but is not bound by the system memory. It can be used to quickly obtain statistics on larger databases.

See also

- *OEFPDatabaseOptions* class
- *OEFPHistogram* class

GetMemoryType

```
unsigned int GetMemoryType() const
```

Returns the memory type of the fingerprint database. The return value is taken from the *OEFastFPDatabaseMemoryType* namespace.

See also

- *OEFastFPDatabase.GetMemoryTypeString* method

GetMemoryTypeString

```
std::string GetMemoryTypeString() const
```

Returns the string representation if memory type of the fingerprint database.

See also

- *OEFastFPDatabase.GetMemoryType* method

GetMoleculeIndex

```
size_t GetMoleculeIndex(const size_t fpidx) const
```

Returns the molecule index that corresponds to the fingerprint index.

Note

When building fingerprint databases using *OECreateFastFPDatabaseOptions*, the molecule index is always the same as the fingerprint index. However, there are private database building APIs that allow to specify the molecule index associated with each fingerprint. This *OEFastFPDatabase.GetMoleculeIndex* method allows to handle these private databases.

GetRawScores

```
OESystem::OEIterBase<double>* GetRawScores(const size_t fpidx,
                                           const OEFPDatabaseOptions& opts) const
OESystem::OEIterBase<double>* GetRawScores(const OEFingerPrint& fp,
                                           const OEFPDatabaseOptions& opts) const
OESystem::OEIterBase<double>* GetRawScores(const OEChem::OEMolBase& mol,
                                           const OEFPDatabaseOptions& opts) const
```

Performs similarity calculations between a molecule or a fingerprint and the fingerprints stored in the *OEFastFPDatabase* object. It returns an iterator over the calculated similarity scores. The scores are not sorted, but returned in the same order as the database. The number of elements in the returned iterator is equal to the number of fingerprints in the database.

fpidx
: If the method is called with an integer index, the query fingerprint is taken from the *OEFastFPDatabase* object with the given index.

mol
: If the method is called with an OEMolBase object, a fingerprint is generated from this molecule before looping over the fingerprints of the database and calculating similarities.

fp
: If the method is called with an *OEFingerPrint* object, its type has to match the type of the *OEFastFPDatabase*.

opts
: The *OEFPDatabaseOptions* object contains the settings available to control the search (i.e. similarity measure parameters). Cutoff and order parameters are ignored as the results are not filtered or sorted.

See also

- *OEFPDatabaseOptions* class
- *OEFastFPDatabase.GetAllScores* method

GetScores

```
OESystem::OEIterBase<OESimScore>* GetScores(const size_t idx,
                                            const OEFPDatabaseOptions& opts) const
OESystem::OEIterBase<OESimScore>* GetScores(const OEFingerPrint& fp,
                                            const OEFPDatabaseOptions& opts) const
OESystem::OEIterBase<OESimScore>* GetScores(const OEChem::OEMolBase& mol,
                                            const OEFPDatabaseOptions& opts) const
```

Performs similarity calculations between a molecule or fingerprint and the fingerprints stored in the *OEFastFPDatabase* object. It returns an iterator over the calculated similarity scores (*OESimScore*). The results are filtered according to the cutoff and order parameters specified in `opts`, but are not sorted.

fpidx
: If the method is called with an integer index, the query fingerprint is taken from the *OEFastFPDatabase* object with the given index.

mol
: If the method is called with an OEMolBase object, a fingerprint is generated from this molecule before looping over the fingerprints of the database and calculating similarities.

fp
: If the method is called with an *OEFingerPrint* object, its type has to match the type of the *OEFastFPDatabase*.

opts
: The *OEFPDatabaseOptions* object controls all the parameters that determine the search (i.e. similarity measure parameters, score cutoff and order).

See also

- *OEFPDatabaseOptions* class

GetSortedScores

```
OESystem::OEIterBase<OESimScore>* GetSortedScores(const size_t idx,
                                                  const OEFPDatabaseOptions& opts) const
OESystem::OEIterBase<OESimScore>* GetSortedScores(const OEFingerPrint& fp,
```

(continues on next page)

(continued from previous page)

```
                                                    const OEFPDatabaseOptions& opts) const
OESystem::OEIterBase<OESimScore>* GetSortedScores(const OEChem::OEMolBase& mol,
                                                    const OEFPDatabaseOptions& opts) const
```

Performs similarity calculations between a molecule or fingerprint and the fingerprints stored in the *OEFastFPDatabase* object. It returns an iterator over the calculated similarity scores (*OESimScore*) in sorted order. Each *OESimScore* holds a similarity score and index of the corresponding fingerprint of the database.

fpidx
: If the method is called with an integer index, the query fingerprint is taken from the *OEFastFPDatabase* object with the given index.

mol
: If the method is called with an OEMolBase object, a fingerprint is generated from this molecule before looping over the fingerprints of the database and calculating similarities.

fp
: If the method is called with an *OEFingerPrint* object, its type has to match the type of the *OEFastFPDatabase*.

opts
: The *OEFPDatabaseOptions* object controls all the parameters that determine the search (i.e. similarity measure parameters, score cutoff, order and limit).

See also

- *OEFPDatabaseOptions* class

GetSparseMatrix

```
OESystem::OEIterBase<OESimScorePair>* GetSparseMatrix(const OEFPDatabaseOptions& opts)␣
↪const
```

Performs NxN similarity calculations between all pairs of fingerprints stored in the *OEFastFPDatabase* object and returns either the top `K` scores for each fingerprint or all scores above a cutoff for each fingerprint. The sparse matrix is returned as an iterator over *OESimScorePair* objects. The limit of scores to return can be set using *OEFPDatabaseOptions.SetLimit* and the cutoff of scores to return can be set using *OEFPDatabaseOptions.SetCutoff*. If no limit is set, all scores will be returned above the cutoff value. If a limit is set, only *OEFPDatabaseOptions.GetLimit* scores will be returned regardless of whether more scores fall within the cutoff range. A `limit` should be set for best performance.

opts
: The *OEFPDatabaseOptions* object controls all the parameters that determine the search (i.e. similarity measure parameters).

Hint

This operation scales with n^2 in memory when `limit = 0` (or when `limit` is not set), so it can easily overwhelm the system memory for larger databases. The best practice is to set a reasonable `limit` that will capture the scores of interest.

See also

- *OEFPDatabaseOptions* class
- *OESimScorePair* class

GetVariogram

```
OEFPVariogram GetVariogram(const std::vector<float>& obsdata,
                           const OEFPDatabaseOptions& opts,
                           const size_t nrbins = 200u) const
```

Performs similarity calculations between all pairs of fingerprints stored in the *OEFastFPDatabase* object. It returns the empirical variogram over the calculated similarity scores with respect to the measurements provided in the `obsdata` parameter in a *OEFPVariogram* object.

obsdata
User-provided empirical measurements for each fingerprint in the database.

opts
The *OEFPDatabaseOptions* object controls all the parameters that determine the scoring (i.e. similarity measure parameters). Cutoff and order parameters are ignored as the results are not filtered or sorted.

nrbins
Number of bins in the returned *OEFPVariogram* object.

Note

The empirical variogram is defined over **distances rather than similarities**. It is therefore not possible to calculate a variogram using *Tversky* similarity. For all other similarity measures, empirical variogram is calculated using $distance = 1 - similarity$.

Hint

This method calculates similarities identically to *OEFastFPDatabase.GetAllScores* but is not bound by the system memory. It can be used to quickly obtain statistics on larger databases.

Note

When the *OEFastFPDatabase* object is initialized with *OEFastFPDatabaseMemoryType.CUDA*, `nrbins` is limited to at most 1024.

Hint

The returned *OEFPVariogram* object also contains a histogram, but note that this histogram is over **distances** rather than similarities.

See also

- *OEFPDatabaseOptions* class
- *OEFPVariogram* class

IsValid

```
bool IsValid() const
```

Returns whether the database was initialized correctly.

NumFingerPrints

```
size_t NumFingerPrints() const
```

Returns the number of *OEFingerPrint* objects stored in the database.

SortedSearch

```
unsigned SortedSearch(OESimSearchResult& result,
                      const OEChem::OEMolBase& mol,
                      const OEFPDatabaseOptions& opts) const
```

Performs **multi-threaded** similarity calculations between a molecule and the fingerprints stored in the *OEFastFPDatabase* object. The method combines the functionality of the *OEFastFPDatabase.GetSortedScores* and the *OEFastFPDatabase.GetHistogram* methods.

result
: The *OESimSearchResult* object that stores the result of the search along with the progress of the search and the histogram of all scores.

mol
: If the method is called with an OEMolBase object, a fingerprint is generated from this molecule before looping over the fingerprints of the database and calculating similarities.

opts
: The *OEFPDatabaseOptions* object controls all the parameters that determine the search (i.e. similarity measure parameters). The *OEFastFPDatabase.SortedSearch* method can use multiple threads to accelerate the search process. The number of processors used can be controlled by the *OEFPDatabaseOptions.SetNumProcessors* method.

The *OEFastFPDatabase.SortedSearch* method returns:

- *OESimSearchStatus.Uninitialized* if the similarity search can not be executed
- *OESimSearchStatus.Finished*, if the similarity search successfully finished

Note

This method is currently only available in *OEFastFPDatabaseMemoryType.MemoryMapped* and *OEFastFPDatabaseMemoryType.InMemory* modes.

Example:

```
OEGraphMol query = new OEGraphMol();
OEChem.OESmilesToMol(query, "Cc1c(c2cc(ccc2n1C(=0)c3ccc(cc3)Cl)OC)CC(=0)O");

uint limit = 5;
OEFPDatabaseOptions opts = new OEFPDatabaseOptions(limit, OESimMeasure.Tanimoto);

uint nrbins = 5;
OESimSearchResult result = new OESimSearchResult(nrbins);
uint status = fpdb.SortedSearch(result, query, opts);
Console.WriteLine("Search status = ", OEGraphSim.OESimSearchStatusToName(status));
Console.WriteLine("Number of searched ", result.NumSearched());

// print scores
foreach(OESimScore score in result.GetSortedScores())
    Console.WriteLine(score.GetScore().ToString("0.000"));

// print histogram
OEFPHistogram hist = result.GetHistogram();
OEDoubleIter bound = new OEDoubleIter(hist.GetBinBoundaries());
OEUnsignedIter count = new OEUnsignedIter(hist.GetCounts());
while (bound.IsValid() && count.IsValid())
{
    double bgn = bound.Target();
    bound.Increment();
    double end = bound.Target();
    Console.WriteLine("[{0,4:0.000} - {1,4:0.000}] = {2}", bgn, end, count.Target());
    count.Increment();
}
```

The output of the code snippet above might look like this:

```
Search status = Finished
Number of searched = 1000
0.950
0.912
0.901
0.886
0.859
[0.000-0.200] = 428
[0.200-0.400] = 312
[0.400-0.600] = 225
[0.600-0.800] = 25
[0.800-1.000] = 10
```

See also

- *OEFastFPDatabase.GetSortedScores* method
- *OEFastFPDatabase.GetHistogram* method
- *OESimSearchStatus* namespace

5.1.4 OEFingerPrint

```
class OEFingerPrint : public OESystem::OEBitVector
```

OEFingerPrint class is used to encode molecular properties. An *OEFingerPrint* object is a *typed* bitvector (OEBitVector). The type of an *OEFingerPrint* object is set when it is initialized.

See also

- *OEFingerPrint.GetFPTypeBase*
- *OEFingerPrint.SetFPTypeBase*

The following methods are publicly inherited from OEBitVector:

operator<	FirstBit	PrevBit
operator=	FromHexString	SetBitOff
operator-=	GetData	SetBitOn
operator[]	GetSize	SetData
operator^=	IsBitOn	SetRangeOff
operator|=	IsEmpty	SetRangeOn
operator &=	LastBit	SetSize
ClearBits	NegateBits	ToHexString
CountBits	NextBit	ToggleBit

Constructors

```
OEFingerPrint()
```

Default constructor that creates an *OEFingerPrint* object with an uninitialized type, zero pointer (*OEFPTypeBase*).

```
OEFingerPrint(const OEFingerPrint& rhs)
```

Copy constructor.

operator bool

```
bool IsValid()
```

Returns whether the *OEFingerPrint* has been initialized, *i.e.*, has a valid type.

GetFPTypeBase

```
const OEFPTypeBase* GetFPTypeBase() const
```

Returns a *const* pointer to the fingerprint type (*OEFPTypeBase*) of the *OEFingerPrint* object. This method will return `0` if the *OEFingerPrint* object has not been initialized.

SetFPTypeBase

```
void SetFPTypeBase(const OEFPTypeBase* t)
```

Sets the fingerprint type of a *OEFingerPrint* object.

The following functions set the type of the fingerprint when an *OEFingerPrint* object is initialized:

- *OEMakeFP*
- *OEMakeMACCS166FP*
- *OEMakeLingoFP*
- *OEMakePathFP*
- *OEMakeCircularFP*
- *OEMakeTreeFP*

> **Warning**
>
> Use this method with caution.
>
> The type of a *OEFingerPrint* object (that is an *OEFPTypeBase*) encodes how the fingerprint is generated. Changing this type can mean that the information represented by bitvector of the fingerprint will be misinterpreted.

5.1.5 OEFPDatabase

```
class OEFPDatabase
```

The *OEFPDatabase* class is designed to perform rapid in-memory fingerprint searches. Each *OEFPDatabase* object is associated with a fingerprint type (*OEFPTypeBase*) that is specified when the database is constructed. An *OEFPDatabase* object can only store fingerprints (*OEFingerPrint*) with this specified type.

> **See also**
>
> - *OEFastFPDatabase* class

Constructors

```
OEFPDatabase(unsigned int fptype)
```

Creates an *OEFPDatabase* object that can store *OEFingerPrint* objects with a given type.

fptype
: The type of the *OEFingerPrint* object stored in the *OEFPDatabase*. This value has to be from the *OEFPType* namespace.

```
OEFPDatabase(const OEGraphSim::OEFPTypeBase *)
```

Creates an *OEFPDatabase* object that can store *OEFingerPrint* objects with *OEFPTypeBase* type.

Note

By default, an *OEFPDatabase* object is constructed with:

- no cutoff value
- *Tanimoto* similarity measure
- descending order to return scores when calling the *OEFPDatabase.GetSortedScores* method

These default values can be altered by the following methods:

- *OEFPDatabase.SetCutoff*
- *OEFPDatabase.SetSimFunc*

AddFP

```
unsigned int AddFP(const OEChem::OEMolBase &mol)
```

Generates an *OEFingerPrint* object from the OEMolBase molecule with the fingerprint type of the *OEFPDatabase*. The generated *OEFingerPrint* object is then inserted into the database returning its index. This method will return `-1`, if the fingerprint generation was unsuccessful.

```
unsigned int AddFP(const OEGraphSim::OEFingerPrint &fp)
```

Creates a copy of the *OEFingerPrint* object, inserts it into the database, and then returns its index. If the type of the passed fingerprint was different from the type of the database, than the insertion is unsuccessful and this method will return `-1`.

Note

The index returned by the *OEFPDatabase.AddFP* method is a unique number starting from zero. This index can be used as a reference number to associate the fingerprint with the molecule from which is it generated.

ClearCutoff

```
void ClearCutoff()
```

Removes the cutoff value previously set by the *OEFPDatabase.SetCutoff* method. After clearing the cutoff value *OEFPDatabase.HasCutoff* method will return `false`.

See also

- *OEFPDatabase.GetCutoff*
- *OEFPDatabase.HasCutoff*
- *OEFPDatabase.SetCutoff*

GetCutoff

```
float GetCutoff() const
```

Returns the cutoff value previously set by the *OEFPDatabase.SetCutoff* method.

See also

- *OEFPDatabase.ClearCutoff*
- *OEFPDatabase.HasCutoff*
- *OEFPDatabase.SetCutoff*

GetDescendingOrder

```
bool GetDescendingOrder() const
```

Returns the order in which the calculated scores are returned by the *OEFPDatabase.GetSortedScores* method.

See also

- *OEFPDatabase.SetDescendingOrder*

GetFPTypeBase

```
const OEGraphSim::OEFPTypeBase* GetFPTypeBase() const
```

Returns the fingerprint type of the *OEFPDatabase* object.

GetFingerPrints

```
OESystem::OEIterBase<const OEGraphSim::OEFingerPrint>* GetFingerPrints() const
```

Returns an iterator pointer over fingerprints (*OEFingerPrint*) stored in the *OEFPDatabase* object. The returned *OEFingerPrint* objects can only be accessed by *const* methods or functions, *i.e.*, they can not be modified.

GetScores

Table 1: **The overloaded versions of the GetScores method**

Link	Description
GetScores(mol, bgn, end)	returns similarities for a molecule
GetScores(fp, bgn, end)	returns similarities for a fingerprint
GetScores(mol, opts)	returns similarities for a molecule with the given options
GetScores(fp, opts)	returns similarities for a fingerprint with the given options

Similarity calculation:

```
OESystem::OEIterBase<OEGraphSim::OESimScore> * GetScores(const␣
↪OEChem::OEMolBase &mol,
```

(continues on next page)

(continued from previous page)

```
                                                    unsigned int bgn = 0,
                                                    unsigned int end = 0)␣
↪const
```

```
OESystem::OEIterBase<OEGraphSim::OESimScore> * GetScores(const␣
↪OEGraphSim::OEFingerPrint &fp,
                                                    unsigned int bgn = 0,
                                                    unsigned int end = 0)␣
↪const
```

Performs similarity calculation between a given molecule or a fingerprint and the fingerprints stored in the *OEFPDatabase* object. It returns an iterator over the calculated similarity scores (*OESimScore*). Each *OESimScore* holds a similarity score and index of the corresponding fingerprint of the database.

mol

If the method is called with an OEMolBase object, then a fingerprint is generated from this molecule before looping over the fingerprints of the database and calculating similarities.

fp

If the method is called with an *OEFingerPrint* object, then its type has to match with the type of the *OEFPDatabase*.

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 fingerprint database.

Note

- By default, the *OEFPDatabase.GetScores* methods calculates *Tanimoto* similarity scores and will always return these scores in the order in which the corresponding *OEFingerPrint* objects are added to the database.
- By using the *OEFPDatabase.SetSimFunc* method, similarity measures other than the default can be performed.
- The behavior of *OEFPDatabase.GetSortedScores* is also influenced by the cutoff value (*OEFPDatabase.SetCutoff*) of the database and the order of the scores specified by the *OEFPDatabase.SetSimFunc* method. If a cutoff value was set, then the *OEFPDatabase.GetScores* method:
  - return scores that are equal to or larger than the specified cutoff value if the order is descending
  - return scores that are equal to or smaller than the specified cutoff value if the order is ascending

See also

- *OEFPDatabase.SetSimFunc* method
- *OEFPDatabase.SetCutoff* method

Examples:

- Calculates the *Tanimoto* similarity on the first 100 entries of the database and returns scores that are equal to or larger than `0.1`.

```
bool descending = true;
fpdb.SetSimFunc(OESimMeasure.Tanimoto, descending);
fpdb.SetCutoff(0.1f);
foreach (OESimScore score in fpdb.GetScores(qfp, 0, 100))
{
    Console.WriteLine("{0:0.000}", score.GetScore());
}
```

- Calculates the *Tversky* similarity (with $\alpha = 0.9$ and $\beta = 0.1$) on the entire database and returns all scores.

```
fpdb.SetSimFunc(new OETverskySim(0.9f));
foreach (OESimScore score in fpdb.GetScores(qfp))
{
    Console.WriteLine("{0:0.000}", score.GetScore());
}
```

- Calculates the *Dice* similarity beginning at the 100th entry of the database and returns scores that are equal to or smaller than `0.5`.

```
descending = true;
fpdb.SetSimFunc(OESimMeasure.Dice, !descending);
fpdb.SetCutoff(0.5f);
foreach (OESimScore score in fpdb.GetScores(qfp, 100))
{
    Console.WriteLine("{0:0.000}", score.GetScore());
}
```

Similarity calculation with an option class:

Performs similarity calculation between a given molecule or a fingerprint and the fingerprints stored in the *OEFPDatabase* object. It returns an iterator over the calculated similarity scores (*OESimScore*). Each *OESimScore* holds a similarity score and index of the corresponding fingerprint of the database.

mol

If the method is called with an OEMolBase object, then a fingerprint is generated from this molecule before looping over the fingerprints of the database and calculating similarities.

fp

If the method is called with an *OEFingerPrint* object, then its type has to match with the type of the *OEFPDatabase*.

opts

The *OEFPDatabaseOptions* object controls all the parameters that determine the sorted search. For example, similarity measure, score cutoff, descending or ascending order etc.

See also

- *OEFPDatabaseOptions* class

Example:

- Calculates the *Cosine* similarity on the entire database and returns scores that are equal to or larger than *0.3*.

```
OEFPDatabaseOptions opts = new OEFPDatabaseOptions();
opts.SetCutoff(0.3f);
opts.SetSimFunc(OESimMeasure.Cosine);
foreach (OESimScore score in fpdb.GetScores(qfp, opts))
{
    Console.WriteLine("{0:0.000}", score.GetScore());
}
```

GetSortedScores

Table 2: **The overloaded versions of the GetSortedScores method**

Link	Description
GetSortedScores(mol, limit, bgn, end)	returns sorted similarities for a molecule
GetSortedScores(fp, limit, bgn, end)	returns sorted similarities for a fingerprint
GetSortedScores(mol, opts)	returns sorted similarities for a molecule with the given options
GetSortedScores(fp, opts)	returns similarities for a fingerprint with the given options

Similarity calculation:

```
OESystem::OEIterBase<OEGraphSim::OESimScore> * GetSortedScores(const␣
↪OEChem::OEMolBase &mol,
                                                                unsigned int␣
↪limit = 0,
                                                                unsigned int␣
↪bgn = 0,
                                                                unsigned int␣
↪end = 0) const
```

```
OESystem::OEIterBase<OEGraphSim::OESimScore> * GetSortedScores(const␣
↪OEGraphSim::OEFingerPrint &fp,
                                                                unsigned int␣
↪limit = 0,
                                                                unsigned int␣
↪bgn = 0,
                                                                unsigned int␣
↪end = 0) const
```

Performs similarity calculations between a molecule or a fingerprint and the fingerprints stored in the *OEFPDatabase* object. It returns an iterator over the calculated similarity scores (*OESimScore*) in sorted order. Each *OESimScore* holds a similarity score and index of the corresponding fingerprint of the database.

mol

If the method is called with an OEMolBase object, then a fingerprint is generated from this molecule before looping over the fingerprints of the database and calculating similarities.

fp

If the method is called with an *OEFingerPrint* object, then its type has to match with the type of the *OEFPDatabase*.

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 fingerprint database.

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

Note

- By default, the *OEFPDatabase.GetSortedScores* method calculates *Tanimoto* similarity scores and returns them in descending order.
- By using the *OEFPDatabase.SetSimFunc* method, similarity measures other than the default can be performed.
- The behavior of *OEFPDatabase.GetSortedScores* is also influenced by the cutoff value (*OEFPDatabase.SetCutoff*) of the database and the order of the scores specified by the *OEFPDatabase.SetSimFunc* method. If a cutoff value was set, then the *OEFPDatabase.GetSortedScores* method:
  - return scores that are equal to or larger than the specified cutoff value if the order is descending
  - return scores that are equal to or smaller than the specified cutoff value if the order is ascending

See also

- *OEFPDatabase.SetSimFunc* method
- *OEFPDatabase.SetCutoff* method

Examples:

- Calculates the *Tanimoto* similarity on the entire database and returns the 10 best scores in descending order.

```
fpdb.ClearCutoff();                                    // default
fpdb.SetSimFunc(OESimMeasure.Tanimoto, descending); // defualt
foreach (OESimScore score in fpdb.GetSortedScores(qfp, 10))
{
    Console.WriteLine("{0:0.000}", score.GetScore());
}
```

- Calculates the *Dice* similarity on the first 100 entries of the database and returns scores that are equal to or larger than 0.5 in descending order.

```
descending = true;
fpdb.SetSimFunc(OESimMeasure.Dice, descending);
fpdb.SetCutoff(0.5f);
foreach (OESimScore score in fpdb.GetSortedScores(qfp, 0, 0, 100))
{
```

(continues on next page)

(continued from previous page)

```
    Console.WriteLine("{0:0.000}", score.GetScore());
}
```

- Calculates *Manhattan* similarity beginning at the 100th entry of the database and returns the "worst" 5 scores that are equal to or smaller than `0.3` in ascending order.

```
descending = true;
fpdb.SetSimFunc(OESimMeasure.Manhattan, !descending);
fpdb.SetCutoff(0.3f);
foreach (OESimScore score in fpdb.GetSortedScores(qfp, 5, 100))
{
    Console.WriteLine("{0:0.000}", score.GetScore());
}
```

Sorted similarity calculation with an option class:

```
OESystem::OEIterBase<OESimScore>* GetSortedScores(const OEChem::OEMolBase& mol,
                                                  const OEFPDatabaseOptions&␣
↪opts) const GetSortedScores(const OEChem::OEMolBase &mol, const␣
↪OEFPDatabaseOption &opts) const
```

```
OESystem::OEIterBase<OESimScore>* GetSortedScores(const OEFingerPrint& fp,
                                                  const OEFPDatabaseOptions&␣
↪opts) const GetSortedScores(const OEGraphSim::OEFingerPrint &fp, const␣
↪OEFPDatabaseOption &opts) const
```

Performs similarity calculations between a molecule or a fingerprint and the fingerprints stored in the *OEFPDatabase* object. It returns an iterator over the calculated similarity scores (*OESimScore*) in sorted order. Each *OESimScore* holds a similarity score and index of the corresponding fingerprint of the database.

mol

If the method is called with an OEMolBase object, then a fingerprint is generated from this molecule before looping over the fingerprints of the database and calculating similarities.

fp

If the method is called with an *OEFingerPrint* object, then its type has to match with the type of the *OEFPDatabase*.

opts

The *OEFPDatabaseOptions* object controls all the parameters that determine the sorted search. For example, similarity measure, score cutoff, descending or ascending order etc.

See also

- *OEFPDatabaseOptions* class

Example:

- Calculates *Tversky* similarity on the entire database and returns the best `10` scores that are equal to or larger than `0.3` in descending order.

```
OEFPDatabaseOptions opts = new OEFPDatabaseOptions();
opts.SetDescendingOrder(true);
```

(continues on next page)

(continued from previous page)

```
opts.SetCutoff(0.3f);
opts.SetSimFunc(OESimMeasure.Tversky);
opts.SetTverskyCoeffs(0.9f, 0.1f);
opts.SetLimit(10);
foreach (OESimScore score in fpdb.GetSortedScores(qfp, opts))
{
    Console.WriteLine("{0:0.000}", score.GetScore());
}
```

HasCutoff

```
bool HasCutoff() const
```

Returns whether the cutoff value of the *OEFPDatabase* object has been set by the *OEFPDatabase.SetCutoff* method.

See also

- *OEFPDatabase.ClearCutoff*
- *OEFPDatabase.GetCutoff*
- *OEFPDatabase.SetCutoff*

NumFingerPrints

```
unsigned int NumFingerPrints() const
```

Returns the number of *OEFingerPrint* objects stored in the database.

SetCutoff

```
void SetCutoff(float)
```

Sets the cutoff value of the *OEFPDatabase* object. The cutoff value influences the behavior of both the *OEFPDatabase.GetScores* and the *OEFPDatabase.GetSortedScores* methods.

See also

- *OEFPDatabase.ClearCutoff*
- *OEFPDatabase.HasCutoff*
- *OEFPDatabase.SetCutoff*

SetDescendingOrder

```
void SetDescendingOrder(bool descending)
```

Sets the order in which the calculated scores are returned by the *OEFPDatabase.GetSortedScores* method.

See also

- *OEFPDatabase.GetDescendingOrder*

SetSimFunc

Sets the method used to evaluate fingerprint similarity when calling either the *OEFPDatabase.GetScores* or the *OEFPDatabase.GetSortedScores* methods.

```
void SetSimFunc(unsigned int simtype, bool descending = true)
```

Sets the similarity calculation by specifying a similarity method with a constant from the *OESimMeasure* namespace. The second argument defines the order in which the calculated scores are returned by the *OEFPDatabase.GetSortedScores* method.

```
void SetSimFunc(const OEGraphSim::OESimFuncBase &,
                bool descending = true)
```

Creates a copy of the *OESimFuncBase* object and uses its *OESimFuncBase.operator()* method to evaluate similarity between two *OEFingerPrint* objects. The second argument defines the order in which the calculated scores are returned by the *OEFPDatabase.GetSortedScores* method.

Note

By default, both the *OEFPDatabase.GetScores* and the *OEFPDatabase.GetSortedScores* methods calculate *Tanimoto* similarity scores. While the *OEFPDatabase.GetScores* always returns these scores in the order in which the corresponding *OEFingerPrint* objects are added to the database, the latter method returns them in descending order, by default.

See also

- *Searching with User-defined Similarity Measures* section
- *User-defined Similarity Measures* section
- *OESimMeasure* namespace
- *OETanimotoSim* class
- *OETverskySim* class

5.1.6 OEFPDatabaseOptions

```
class OEFPDatabaseOptions
```

The *OEFPDatabaseOptions* class allows setting options on a per-search basis. This class is used to control the behavior of:

- the following methods of the *OEFPDatabase* class
  - *OEFPDatabase.GetScores*
  - *OEFPDatabase.GetSortedScores*

- the following methods of the *OEFastFPDatabase* class
  - *OEFastFPDatabase.GetAllScores*
  - *OEFastFPDatabase.GetHistogram*
  - *OEFastFPDatabase.GetSortedScores*
  - *OEFastFPDatabase.GetRawScores*
  - *OEFastFPDatabase.SortedSearch*

See also

- Example code in *Similarity calculation with an option class* section
- Example code in *Sorted similarity calculation with an option class* section
- Example code in *Searching fast fingerprint database* section

The *OEFPDatabaseOptions* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / type
cutoff value	*GetCutoff*	*SetCutoff* / *HasCutoff*	floating point number
order of sorting	*GetDescendingOrder*	*SetDescendingOrder*	boolean
max number of scores	*GetLimit*	*SetLimit*	integer `0` means no limit
similarity method	*GetSimFunc*	*SetSimFunc*	*OESimMeasure*
α and β for *Tversky*	*GetTverskyAlpha* / *GetTverskyBeta*	*SetTverskyCoeffs*	floating point number
numper of processors	*GetNumProcessors*	*SetNumProcessors*	integer `0` means `OEGetNumProcessors`

Constructors

```
OEFPDatabaseOptions()
```

Default constructor.

Table 3: **Default parameters of OEFPDatabaseOptions**

Property	Default value
cutoff value	no cutoff set
order of sorting	true (means descending order)
max number of scores	0 (means no limit)
similarity method	*OESimMeasure.Tanimoto*
α and β for *Tversky*	$\alpha = 0.95, \beta = 0.05$
number of processors	1

```
OEFPDatabaseOptions(const unsigned limit,
                    const unsigned simfunc = OESimMeasure::Tanimoto)
```

Constructs an *OEFPDatabaseOptions* object with a limit on the number of similarity scores returned and a similarity method from the *OESimMeasure* namespace. The returned scores will be in descending order and applying no cutoff value.

```
OEFPDatabaseOptions(const OEFPDatabaseOptions& rhs)
```

Copy constructors.

ClearCutoff

```
void ClearCutoff()
```

Removes the cutoff value previously set by the *OEFPDatabaseOptions.SetCutoff* method. After clearing the cutoff value, *OEFPDatabaseOptions.HasCutoff* method will return `false`.

See also

- *OEFPDatabaseOptions.HasCutoff* method
- *OEFPDatabaseOptions.GetCutoff* method
- *OEFPDatabaseOptions.SetCutoff* method

GetCutoff

```
float GetCutoff() const
```

Returns the cutoff value previously set by the *OEFPDatabase.SetCutoff* method.

See also

- *OEFPDatabaseOptions.ClearCutoff* method
- *OEFPDatabaseOptions.HasCutoff* method
- *OEFPDatabaseOptions.SetCutoff* method

GetDescendingOrder

```
bool GetDescendingOrder() const
```

Returns whether the similarity scores will be returned in descending or in ascending order.

See also

- *OEFPDatabaseOptions.SetDescendingOrder* method

GetLimit

```
unsigned GetLimit() const
```

Returns the maximum number of similarity scores returned.

See also

- *OEFPDatabaseOptions.SetLimit* method

GetSimFunc

```
unsigned GetSimFunc() const
```

Returns the similarity measure used to evaluate fingerprint similarities. The return value is taken from the *OESimMeasure* namespace.

See also

- *OEFPDatabaseOptions.SetSimFunc* method

GetNumProcessors

```
unsigned GetNumProcessors() const
```

Returns the number of threads that are used to search fingerprints when calling the *OEFastFPDatabase.SortedSearch* method.

See also

- *OEFPDatabaseOptions.SetNumProcessors* method

GetTverskyAlpha

```
float GetTverskyAlpha() const
```

Returns the α parameter of the *Tversky* similarity.

See also

- *OEFPDatabaseOptions.SetTverskyCoeffs* method
- *OETverskySim* class

GetTverskyBeta

```
float GetTverskyBeta() const
```

Returns the β parameter of the *Tversky* similarity.

See also

- *OEFPDatabaseOptions.SetTverskyCoeffs* method

- *OETverskySim* class

HasCutoff

```
bool HasCutoff() const
```

Returns whether the cutoff value of the *OEFPDatabaseOptions* object has been set by the *OEFPDatabaseOptions.SetCutoff* method.

See also

- *OEFPDatabaseOptions.ClearCutoff* method
- *OEFPDatabaseOptions.GetCutoff* method
- *OEFPDatabaseOptions.SetCutoff* method

SetCutoff

```
void SetCutoff(const float cutoff)
```

Sets the cutoff value of the *OEFPDatabaseOptions* object.

See also

- *OEFPDatabaseOptions.ClearCutoff* method
- *OEFPDatabaseOptions.GetCutoff* method
- *OEFPDatabaseOptions.HasCutoff* method

SetDescendingOrder

```
void SetDescendingOrder(const bool descending = true)
```

Sets whether the similarity scores will be returned in descending or in ascending order.

Note

This method has no effect on the *OEFPDatabase.GetScores* method.

See also

- *OEFPDatabaseOptions.GetDescendingOrder* method

SetLimit

```
void SetLimit(const unsigned limit)
```

Sets the maximum number of similarity scores returned.

limit
If it is set to zero, then all similarity scores are returned.

Note

This method has no effect on the *OEFPDatabase.GetScores* method.

See also

- *OEFPDatabaseOptions.GetLimit* method

SetSimFunc

```
void SetSimFunc(const unsigned simtype)
```

Sets the similarity measure used to evaluate fingerprint similarities.

simtype
This value has to be from the *OESimMeasure* namespace.

See also

- *OEFPDatabaseOptions.GetSimFunc* method

SetNumProcessors

```
void SetNumProcessors(const unsigned nrps)
```

Sets the number of threads that are used to search fingerprints when calling the *OEFastFPDatabase.SortedSearch* method. When set to `0`, the number of processors used will be the number returned by the `OEGetNumProcessors` function.

Multi-threaded fingerprint search is currently only available in *OEFastFPDatabaseMemoryType.MemoryMapped* and *OEFastFPDatabaseMemoryType.InMemory* modes.

SetTverskyCoeffs

```
void SetTverskyCoeffs(const float alpha = 0.95f,
                      const float beta = 0.05f)
```

Sets α and β parameters for the *Tversky* similarity.

See also

- *OEFPDatabaseOptions.GetTverskyAlpha* method
- *OEFPDatabaseOptions.GetTverskyBeta* method
- *OETverskySim* class

5.1.7 OEFPHistogram

```
class OEFPHistogram
```

The *OEFPHistogram* class is used to hold a histogram of similarity scores.

This class is an efficient way to obtain statistics on large databases which would be unfeasible to keep as full NxN similarity matrices.

The following classes derive from this class:

- *OEFPVariogram*

See also

- *OEFastFPDatabase.GetHistogram* method

Constructors

```
OEFPHistogram(const size_t numBins = 200u, const double min = 0.0,
              const double max = 1.0);
OEFPHistogram(const OEFPHistogram&)
```

Creates an *OEFPHistogram* object with given number of bins and range.

nrbins
Number of bins in the histogram. (default=200)

min
Minimum bound (inclusive) of the histogram range. (default = 0.0)

max
Maximum bound (inclusive) of the histogram range. (default = 1.0)

Note

The default range of [0,1] covers all built-in similarity measures.

AddSample

```
size_t AddSample(const float sample)
size_t AddSample(const double sample)
```

Adds the sample into the histogram by incrementing the count in the relevant bin. Returns the bin index that the sample went into.

GetBinBoundaries

```
OESystem::OEIterBase<double>* GetBinBoundaries() const
```

Returns an iterator over the boundaries for each bin.

> **Note**
>
> The number of elements in this iterator is equal to *OEFPHistogram.NumBins* **+1**.

GetBinCenters

```
OESystem::OEIterBase<double>* GetBinCenters() const
```

Returns an iterator over the center value for each bin.

> **Note**
>
> The number of elements in this iterator is equal to *OEFPHistogram.NumBins*.

> **Hint**
>
> It is generally reasonable to plot the results of *OEFPHistogram.GetCounts* or *OEFPHistogram.GetDensity* against the bin centers returned by this method.

GetBinIdx

```
size_t GetBinIdx(const float sample) const
size_t GetBinIdx(const double sample) const
```

Returns the bin index that the sample would go into without actually adding the sample into the bin.

GetBinWidth

```
double GetBinWidth() const
```

Returns the width of bins in the histogram.

GetCounts

```
OESystem::OEIterBase<unsigned int>* GetCounts() const
```

Returns an iterator over the bin counts.

> **Note**
>
> Even though counts are returned as unsigned integers, they are internally kept as size_t. If the counts are overflowing the unsigned int type of the system, it may be more appropriate to use the normalized *OEFPHistogram.GetDensity* instead.

GetDensity

```
OESystem::OEIterBase<double>* GetDensity() const
```

Returns an iterator over the probability density for each bin.

> **Note**
>
> Result is equivalent to *OEFPHistogram.GetCounts* normalized by *OEFPHistogram.NumSamples*.

GetMax

```
double GetMax() const
```

Returns upper bound (inclusive) of the histogram range.

GetMin

```
double GetMin() const
```

Returns lower bound (inclusive) of the histogram range.

Mean

```
double Mean()
```

Approximates and returns the sample mean over the histogram.

> **Note**
>
> This approximation is affected by number of bins in the histogram.

NumBins

```
size_t NumBins() const
```

Returns the number of bins in the histogram.

NumSamples

```
size_t NumSamples() const
```

Returns the total number of samples in the histogram.

Std

```
double Std()
```

Approximates and returns the sample standard deviation over the histogram.

> **Note**
>
> This approximation is affected by number of bins in the histogram.

5.1.8 OEFPPattern

```
class OEFPPattern : public OEChem::OEAtomBondSet
```

A container that stores patterns that are enumerated when fingerprints such as *OEFPType.Path*, *OEFPType.Circular* and *OEFPType.Tree* are generated.

> **See also**
>
> - *Fingerprint Patterns* chapter
> - *OEGetFPPatterns* function

The following methods are publicly inherited from OEAtomBondSet:

operator=	ClearBonds	Remove
Add	CreateCopy	RemoveAtom
AddAtom	GetAtoms	RemoveAtoms
AddAtoms	GetBonds	RemoveBond
AddBond	HasAtom	RemoveBonds
AddBonds	HasBond	Sweep
Clear	IsEmpty	Translate
ClearAtoms	NumAtoms	
ClearAtomsAndBonds	NumBonds	

Constructors

```
OEFPPattern(OESystem::OEIter<OEChem::OEAtomBase>& atoms,
            OESystem::OEIter<OEChem::OEBondBase>& bonds,
            const std::string& smarts, unsigned int unhashed,
            unsigned int setbit)
```

Constructor.

GetBit

```
unsigned int GetBit() const
```

Returns the bit that is set when the given pattern is hashed into fixed-length bitvector.

GetSmarts

```
std::string GetSmarts() const
```

Returns the smarts representation of the given pattern.

GetUnhashed

```
unsigned int GetUnhashed() const
```

Returns the unhashed unique integer representation of the given pattern.

5.1.9 OEFPTypeBase

```
class OEFPTypeBase
```

The *OEFPTypeBase* class is the abstract interface for representing *OEFingerPrint* types.

> **Note**
>
> When a *OEFingerPrint* object is initialized, its type is set to a type that is derived from this abstract base class.

Constructors

```
OEFPTypeBase(unsigned short v)
```

Default constructor.

GetFPType

```
unsigned int GetFPType() const = 0
```

Returns the type of the *OEFPTypeBase* object from the *OEFPType* namespace.

GetFPTypeString

```
std::string GetFPTypeString() const = 0
```

Returns a string representation of *OEFPTypeBase*. This string representation encodes information about the fingerprinting method and the parameters being used to generate the fingerprint.

Note

The string returned by the *OEFPTypeBase.GetFPTypeString* method does not include new line characters. The strings presented here were broken into separate lines only for better readability.

The string representation of the default *OEFPType.Circular* fingerprint is the following:

```
Circular,ver=2.0.0,size=4096,radius=0-5,atype=AtmNum|Arom|Chiral|FCharge|HCount|EqHalo,
btype=Order
```

The string representation of the default *OEFPType.Lingo* fingerprint is the following:

```
Lingo,ver=2.0.0
```

The string representation of the default *OEFPType.MACCS166* fingerprint is the following:

```
MACCS166,ver=2.0.0
```

The string representation of the default *OEFPType.Path* fingerprint is the following:

```
Path,ver=2.0.0,size=4096,bonds=0-5,atype=AtmNum|Arom|Chiral|FCharge|HvyDeg|Hyb|EqHalo,
btype=Order|Chiral
```

The string representation of the default *OEFPType.Tree* fingerprint is the following:

```
Tree,ver=2.0.0,size=4096,bonds=0-4,atype=AtmNum|Arom|Chiral|FCharge|HvyDeg|Hyb,
btype=Order
```

See also

- *OEFPTypeParams* class

GetFPVersion

```
unsigned short GetFPVersion() const
```

Returns the version number of the fingerprint.

GetFPVersionString

```
std::string GetFPVersionString() const
```

Returns the string format of the version of the fingerprint.

5.1.10 OEFPTypeParams

```
class OEFPTypeParams
```

This class represents *OEFPTypeParams*.

See also

- *Fingerprint parameters* section

Constructors

```
OEFPTypeParams(const std::string& typestr)
```

Constructor that parses the given string as a representation of a fingerprint and extracts the parameters that are used to generate the fingerprint.

See also

- *OEFPTypeBase.GetFPTypeString* method

```
OEFPTypeParams(const OEFPTypeBase* fptype)
```

Constructor that extracts the parameters that are used to generate the fingerprint type.

See also

- *OEFPTypeBase* class

GetAtomTypes

```
unsigned int GetAtomTypes() const
```

Returns

- the atom types extracted from the string for *OEFPType.Path*, *OEFPType.Circular* and *OEFPType.Tree* fingerprint types
- zero for *OEFPType.Lingo* and *OEFPType.MACCS166* fingerprint types

The return value is taken from the *OEFPAtomType* namespace.

See also

- *OEGetFPAtomType* function

GetBondTypes

```
unsigned int GetBondTypes() const
```

Returns

- the bond types extracted from the string for *OEFPType.Path*, *OEFPType.Circular* and *OEFPType.Tree* fingerprint types
- zero for *OEFPType.Lingo* and *OEFPType.MACCS166* fingerprint types

The return value is taken from the *OEFPBondType* namespace.

See also

- *OEGetFPBondType* function

GetFPType

```
unsigned int GetFPType() const
```

Returns the fingerprint type extracted from the string. The return value is taken from the *OEFPType* namespace.

GetMaxDistance

```
unsigned int GetMaxDistance() const
```

Returns:

- the size of the largest path fragments (in bonds), if the fingerprint type is *OEFPType.Path*
- the size of the largest circular fragments (in radius), if the fingerprint type is *OEFPType.Circular*
- the size of the largest tree fragments (in bonds), if the fingerprint type is *OEFPType.Tree*
- zero for *OEFPType.Lingo* and *OEFPType.MACCS166* fingerprint types

GetMinDistance

```
unsigned int GetMinDistance() const
```

Returns:

- the size of the smallest path fragments (in bonds), if the fingerprint type is *OEFPType.Path*
- the size of the smallest circular fragments (in radius), if the fingerprint type is *OEFPType.Circular*
- the size of the smallest tree fragments (in bonds), if the fingerprint type is *OEFPType.Tree*
- zero for *OEFPType.Lingo* and *OEFPType.MACCS166* fingerprint types

GetNumBits

```
unsigned int GetNumBits() const
```

Returns:

- the size of the fingerprint in bits for *OEFPType.Path*, *OEFPType.Circular* and *OEFPType.Tree* fingerprint types
- 166 for *OEFPType.MACCS166* fingerprint type
- zero for *OEFPType.Lingo* fingerprint type

GetVersion

```
unsigned short GetVersion() const
```

Returns the version number.

See also

- *OEGetFingerPrintVersionString* function

IsValid

```
bool IsValid() const
```

Returns whether the string was valid from which the *OEFPTypeParams* object was initialized.

5.1.11 OEFPVariogram

```
class OEFPVariogram : public OEFPHistogram
```

OEFPVariogram class is used to generate a variogram of similarity scores and related empirical measurements. An *OEFPVariogram* object is a *OEFPHistogram* with additional variogram data regarding user provided measurements.

The empirical variogram has the form :

$\hat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{(i,j)\in N(h)} |z_i - z_j|^2$

Where $N(h)$ is the set of pairs that have a **distance** close to h and z_i is the measurement provided for the i th molecule.

For all of our built-in similarity measures, except for Tversky, distance is calculated a $1 - similarity$. It is not possible to calculate an empirical variogram using Tversky similarity.

The **measurement** in the context of a variogram can be any user provided value per fingerprint e.g. bioassay data or calculated properties.

The following methods are publicly inherited from *OEFPHistogram*:

AddSample	*GetCounts*	*NumBins*
GetBinBoundaries	*GetDensity*	*NumSamples*
GetBinCenters	*GetMax*	*Std*
GetBinIdx	*GetMin*	
GetBinWidth	*Mean*	

See also

- *OEFastFPDatabase.GetVariogram* method

Constructors

```
OEFPVariogram(size_t numBins = 200, double min = 0, double max = 1)
OEFPVariogram(const OEFPVariogram&)
```

Creates an *OEFPVariogram* object with the given number of bins and similarity range.

numBins
Number of bins in the histogram. (default = 200)

min
Minimum bound (inclusive) of the similarity range. (default = 0.0)

max
Maximum bound (inclusive) of the similarity range. (default = 1.0)

AddSample

```
size_t AddSample(const float sim, const float obs)
```

Adds a sample representing a pair of molecules into the variogram

sim
Similarity between fingerprints i and j

obs
Measured difference between the relevant fingerprints $z_i - z_j$

Note

Since we are dealing with $|z_i - z_j|^2$ this relationship is actually symmetrical.

GetVariogram

```
OESystem::OEIterBase<double>* GetVariogram() const
```

Returns an iterator over the empirical variogram.

5.1.12 OESimFuncBase

```
class OESimFuncBase
```

OESimFuncBase is an abstract base class which defines the interface necessary to perform similarity calculations of *OEFingerPrint* objects.

See also

- *User-defined Similarity Measures* section

The following classes derive from this class:

- *OETanimotoSim*
- *OETverskySim*

Constructors

```
OESimFuncBase()
```

Default constructor.

operator()

```
float operator()(const OEGraphSim::OEFingerPrint& fpA, const OEGraphSim::OEFingerPrint&␣
↪fpB) const = 0
```

Virtual *const* operator that takes two *OEFingerPrint* objects and returns their evaluated similarity value.

CreateCopy

```
OESimFuncBase* CreateCopy() const = 0
```

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

GetSimTypeString

```
std::string GetSimTypeString() const = 0
```

Returns a string representation of the concrete derived class.

5.1.13 OESimScore

```
class OESimScore
```

This class represents *OESimScore* that holds a similarity value with an index that identifies the fingerprint in the *OEFPDatabase* object from which the score is calculated.

See also

- *OEFPDatabase.GetScores*
- *OEFPDatabase.GetSortedScores*

Constructors

```
OESimScore(size_t i, double s)
```

Initializes an *OESimScore* from an index `i` and a score `s`.

GetIdx

```
size_t GetIdx() const
```

Returns the index of the fingerprint corresponding to the similarity score.

GetScore

```
double GetScore() const
```

Returns the similarity score of the fingerprint corresponding to the index.

5.1.14 OESimScorePair

```
class OESimScorePair
```

This class represents *OESimScorePair* that holds a similarity scores with an index `i` and an index `j` that identifies the fingerprints in the *OEFastFPDatabase* object from an NxN similarity calculation with *OEFastFPDatabase.GetSparseMatrix*.

See also

- *OEFastFPDatabase.GetSparseMatrix*

Constructors

```
OESimScorePair(size_t i, size_t j, double s)
```

Initializes an *OESimScorePair* from an index `i`, an index `j` and a score `s`.

GetIdxA

```
size_t GetIdxA() const
```

Returns the index `i` of the fingerprint corresponding to the similarity score.

GetIdxB

```
size_t GetIdxB() const
```

Returns the index `j` of the fingerprint corresponding to the similarity score.

GetScore

```
double GetScore() const
```

Returns the similarity score of the fingerprint corresponding to the index `i` and index `j` fingerprints.

5.1.15 OESimSearchResult

```
class OESimSearchResult
```

The *OESimSearchResult* class is used to store the results of a similarity search along with reporting the progress of the search and the histogram of the similarity scores.

See also

- *OEFastFPDatabase.SortedSearch* method

Constructor

```

```

Constructor an empty result object with parameters for the score histogram.

nrbins
: Number of bins in the histogram (default = 50).

minv
: Minimum bound (inclusive) of the histogram range (default = 0.0).

maxv
: Maximum bound (inclusive) of the histogram range (default = 1.0).

GetHistogram

```
OEFPHistogram GetHistogram() const
```

Returns an *OEFPHistogram* object that stores the histogram of similarity scores.

GetSearchStatus

```
unsigned GetSearchStatus() const
```

Returns the status of the similarity search defined in the *OESimSearchStatus* namespace.

GetSortedScores

```
OESystem::OEIterBase<OESimScore>* GetSortedScores()
```

Returns an iterator over the calculated similarity scores (*OESimScore*) in the sorted order. Each *OESimScore* holds a similarity score and index of the corresponding fingerprint of the database.

IsValid

```
bool IsValid() const
```

Returns whether the *OESimSearchResult* object is valid.

NumSearched

```
size_t NumSearched() const
```

Returns the number of fingerprints that have been already searched. This counter is continuously updated and provides the progress of the similarity search.

NumTargets

```
size_t NumTargets() const
```

Returns the number of targets (i.e. the number of fingerprints in the database).

5.1.16 OETanimotoSim

```
class OETanimotoSim : public OEGraphSim::OESimFuncBase
```

This class represents *OETanimotoSim* that calculates the *Tanimoto* similarity score.

Formula:

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

See also

- *Tanimoto* section
- *OETanimoto* function

The following methods are publicly inherited from *OESimFuncBase*:

operator()	*CreateCopy*	*GetSimTypeString*

Constructors

```
OETanimotoSim()
```

Default constructor.

operator()

```
float operator()(const OEGraphSim::OEFingerPrint& fpA, const OEGraphSim::OEFingerPrint&␣
↪fpB) const
```

Returns the *Tanimoto* similarity coefficient of the two given *OEFingerPrint* objects.

CreateCopy

```
OEGraphSim::OESimFuncBase* CreateCopy() const
```

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

GetSimTypeString

```
std::string GetSimTypeString() const
```

Returns a string representation of the *OETanimotoSim* class.

5.1.17 OETverskySim

```
class OETverskySim : public OEGraphSim::OESimFuncBase
```

This class represents *OETverskySim* that calculates the *Tversky* similarity score.

Formula:

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

See also

- *Tversky* section
- *OETversky* function

The following methods are publicly inherited from *OESimFuncBase*:

operator()	*CreateCopy*	*GetSimTypeString*

Constructors

```
OETverskySim(float a = 0.95f)
```

Default constructor ($\alpha = 0.95$, $\beta = 1.0 - \alpha = 0.05$)

```
OETverskySim(float a, float b)
```

Constructor with arbitrary α and β parameters.

operator()

```
float operator()(const OEGraphSim::OEFingerPrint& fpA, const OEGraphSim::OEFingerPrint&␣
↪fpB) const
```

Returns the *Tversky* similarity coefficient of the two given *OEFingerPrint* objects.

CreateCopy

```
OEGraphSim::OESimFuncBase* CreateCopy() const
```

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

GetAlpha

```
float GetAlpha() const
```

Returns the α parameter of the *Tversky* similarity calculation.

GetBeta

```
float GetBeta() const
```

Returns the β parameter of the *Tversky* similarity calculation.

GetSimTypeString

```
std::string GetSimTypeString() const
```

Returns a string representation of the *OETverskySim* class.

5.2 OEGraphSim Constants

5.2.1 OEFastFPDatabaseMemoryType

This namespace contains memory type constants representing how fingerprints are stored and searched for in an *OEFastFPDatabase* object.

See also

- *OEFastFPDatabase* class

Default

The default memory type is `OEFastFPDatabaseMemoryType.MemoryMapped`.

CUDA

If a CUDA-enabled GPU is available, *OEFastFPDatabase* can take advantage of the device for similarity calculations. When an *OEFastFPDatabase* is initialized with the `OEFastFPDatabaseMemoryType.CUDA` type, similarity calculations are performed on the device. Initialization incurs a one-time load penalty, similar to `OEFastFPDatabaseMemoryType.InMemory`.

Requirements:

- *Prerequisites for OpenEye's GPU-accelerated software*
- Fingerprint sizes of multiples of 1024 bits.

Note

At present, multi-GPU support is not available. If there are multiple CUDA devices on the system, the first compatible device detected will be used. To enforce the use of a particular device, set the environment variable `CUDA_VISIBLE_DEVICES`.

Note

If device memory limits are exceeded or a CUDA-enabled device is not detected this implementation falls back to `OEFastFPDatabaseMemoryType.MemoryMapped`

InMemory

When an *OEFastFPDatabase* is initialized with `OEFastFPDatabaseMemoryType.InMemory` type the fingerprints are pre-loaded into memory and the similarity search performed in-memory when calling the `OEFastFPDatabase.GetSortedScores` method.

Hint

`OEFastFPDatabaseMemoryType.InMemory` presents the fastest search at the expense of load time and the search is limited by the size of the memory.

MemoryMapped

When an *OEFastFPDatabase* is initialized with `OEFastFPDatabaseMemoryType.MemoryMapped` type there is no load time penalty or memory limitation but the search itself is slower.

See also

- Example code in *Searching fast fingerprint database* section

5.2.2 OEFPAtomType

This namespace contains atom typing options that can be used when generating *Circular*, *Path* or *Tree* fingerprints. Atom type options control how the atoms of the enumerated circular, path or tree fragments are encoded during the fingerprint generation.

The `OEFPAtomType` namespace contains the following constants:

Constant name	Combination of
`OEFPAtomType.Aromaticity`	
`OEFPAtomType.AtomicNumber`	
`OEFPAtomType.Chiral`	
`OEFPAtomType.DefaultAtom`	`AtomicNumber` | `Aromaticity` | `Chiral` | `FormalCharge` | `HvyDegree` | `Hybridization` | `EqHalogen`
`OEFPAtomType.DefaultCircularAtom`	`AtomicNumber` | `Aromaticity` | `Chiral` | `FormalCharge` | `HCount` | `EqHalogen`
`OEFPAtomType.DefaultCircularVSAtom`	`AtomicNumber` | `Aromaticity` | `Chiral` | `EqHalogen`
`OEFPAtomType.DefaultCircularVSAtom1`	`AtomicNumber` | `Aromaticity` | `Chiral` | `HCount` | `EqHalogen`
`OEFPAtomType.DefaultCircularVSAtom2`	`AtomicNumber` | `Aromaticity` | `Chiral` | `EqHalogen`
`OEFPAtomType.DefaultPathAtom`	`AtomicNumber` | `Aromaticity` | `Chiral` | `FormalCharge` | `HvyDegree` | `Hybridization` | `EqHalogen`
`OEFPAtomType.DefaultPathVSAtom`	`AtomicNumber` | `Aromaticity` | `Chiral` | `HvyDegree` | `EqHalogen`
`OEFPAtomType.DefaultTreeAtom`	`AtomicNumber` | `Aromaticity` | `Chiral` | `FormalCharge` | `HvyDegree` | `Hybridization`
`OEFPAtomType.DefaultTreeVSAtom`	`AtomicNumber` | `Aromaticity` | `Chiral` | `HvyDegree`
`OEFPAtomType.EqAromatic`	
`OEFPAtomType.EqHalogen`	
`OEFPAtomType.EqHBondAcceptor`	
`OEFPAtomType.EqHBondDonor`	
`OEFPAtomType.FormalCharge`	
`OEFPAtomType.HCount`	
`OEFPAtomType.HvyDegree`	
`OEFPAtomType.Hybridization`	
`OEFPAtomType.InRing`	
`OEFPAtomType.None`	

Hint

The virtual screening variant (`VS`) version of the default atom types, treats certain functional group identically regardless of their pKa state e.g. protonated and unprotonated carboxylic acids.

Note

The constants of the *`OEFPAtomType`* namespace can be **combined** using the bitwise `OR` operation.

Note

The images in this sections visualize the effect of the various atom typing options.

See also: *Visualizing Molecule Similarity* section

See also

- *`OEFPBondType`* namespace
- *`OEMakeCircularFP`* function
- *`OEMakePathFP`* function
- *`OEMakeTreeFP`* function
- *Atom and Bond Typing* section

Note

All explicit hydrogens are suppressed of the molecule before generating any fingerprints. (See example in *Example of molecules that are considered to be equivalent due to suppressing their explicit hydrogens*).

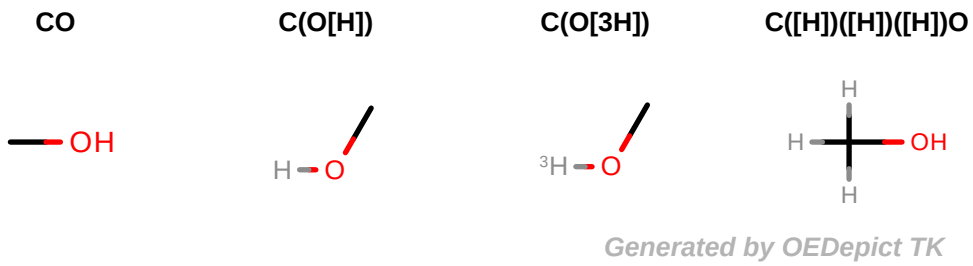
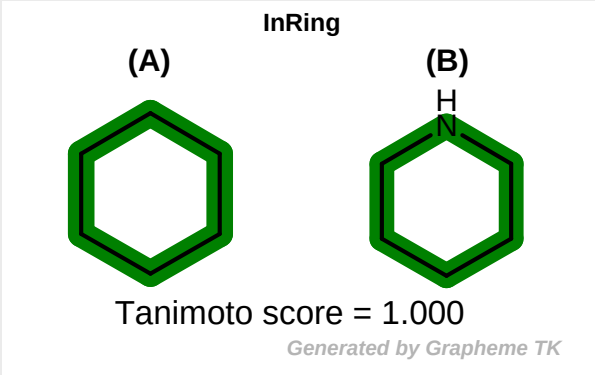


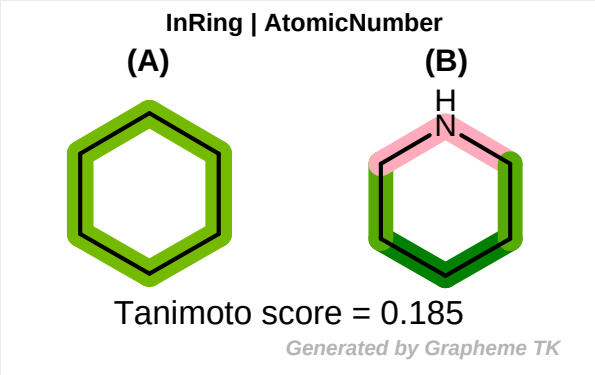
Fig. 2: **Example of molecules that are considered to be equivalent due to suppressing their explicit hydrogens**

AtomicNumber

This flag indicates that atomic number (the value returned by the `OEAtomBase.GetAtomicNum` method) is encoded into the generated fingerprint, *i.e.*, if two fragments (either circular, paths or tree) are composed of atoms with different atomic numbers, then the two fragments will be mapped to different bits of the fingerprint. *Table: Example of using the AtomicNumber option* demonstrates the effect of using the *OEFPAtomType.AtomicNumber* flag.

Table 4: **Example of using the *OEFPAtomType.AtomicNumber* option** (*Path*, numbits=4096, bonds=0-5, bond typing = *BondOrder*).

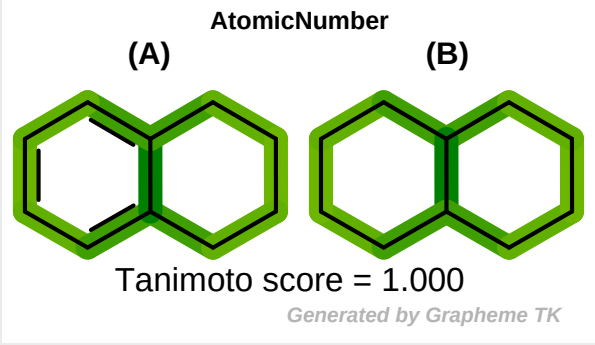


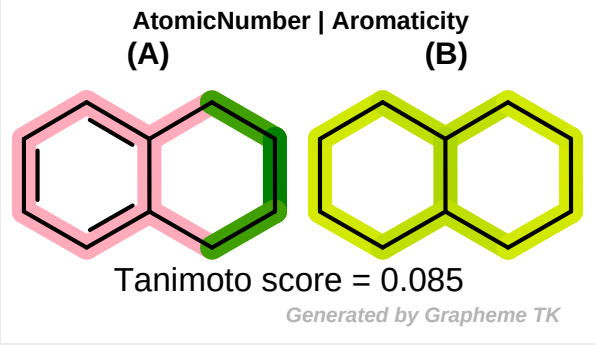


Aromaticity

This flag indicates that aromaticity (the value returned by the `OEAtomBase.IsAromatic` method) is encoded into the generated fingerprint, *i.e.*, an aromatic and an aliphatic fragment will be mapped to different bits of the fingerprint. *Table: Example of using the Aromatic option* demonstrates the effect of using the *OEFPAtomType.Aromaticity* flag.

Table 5: **Example of using the *OEFPAtomType.Aromaticity* option** (*Path*, numbits=4096, bonds=0-5, bond typing = *InRing*).





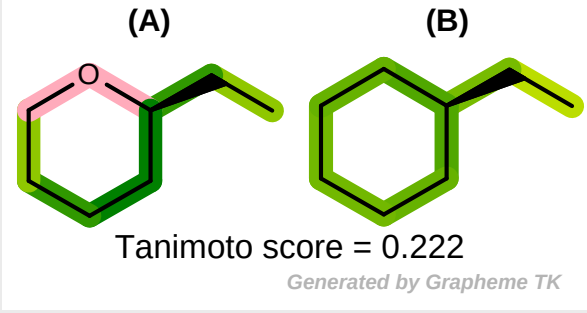
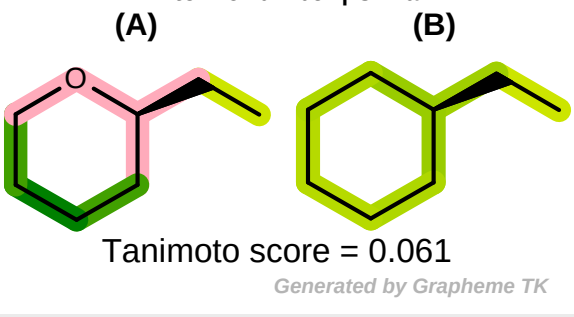
Note

Prior to generating a fingerprint, the aromaticity of the molecule is re-perceived using the `OEAroModel.OpenEye` aromaticity model.

Chiral

This flag indicates that chiral and non-chiral atoms (the value returned by the `OEAtomBase.IsChiral` method) are distinguished during the fingerprint generation. *Table: Example of using the Chiral option* demonstrates the effect of using the *`OEFPAtomType.Chiral`* flag.

Table 6: **Example of using the *`OEFPAtomType.Chiral`* option** (*`Path`*, numbits=4096, bonds=0-5, bond typing = *`BondOrder`*).

AtomicNumber	AtomicNumber | Chiral
	

Note

Different stereoisomers of molecules can not be distinguished when the *`OEFPAtomType.Chiral`* flag is set. (See example in *Figure: Example of molecule similarity of stereoisomers*).

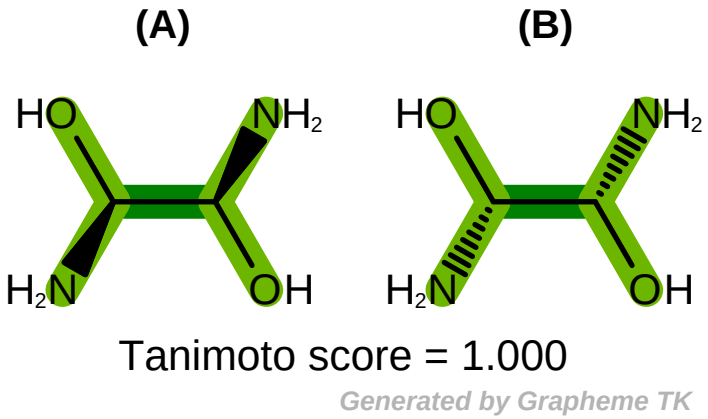
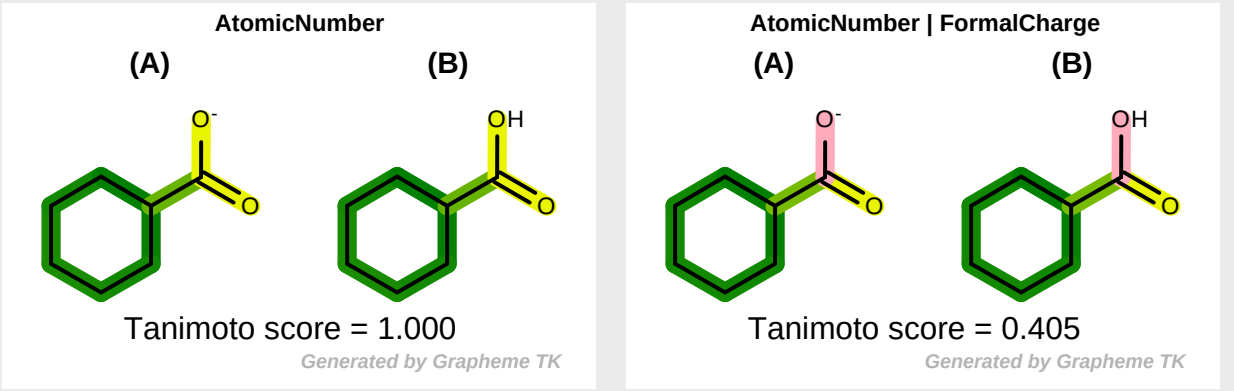


Fig. 3: **Example of molecule similarity of stereoisomers**

FormalCharge

This flag indicates that formal charge (the value returned by the `OEAtomBase.GetFormalCharge` method) is encoded into the generated fingerprint. *Table: Example of using the FormalCharge option* demonstrates the effect of using the *`OEFPAtomType.FormalCharge`* flag.

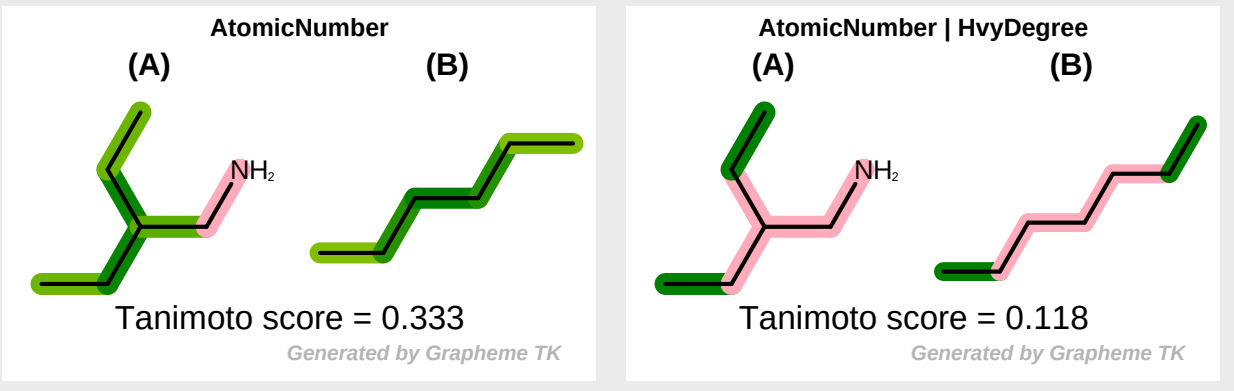
Table 7: **Example of using the** *OEFPAtomType.FormalCharge* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *BondOrder*).



HvyDegree

This flag indicates that heavy degree information (the value returned by the `OEAtomBase.GetHvyDegree` method) is encoded into the generated fingerprint. *Table: Example of using the HvyDegree option* demonstrates the effect of using the *OEFPAtomType.HvyDegree* flag.

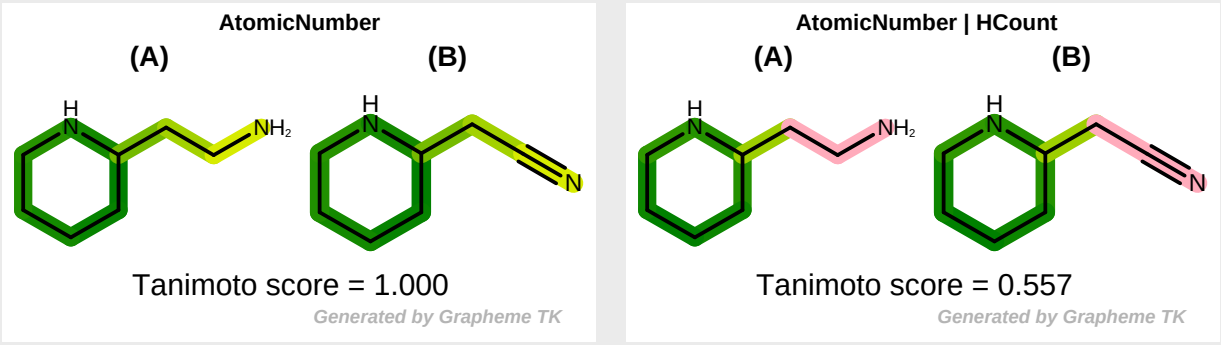
Table 8: **Example of using the** *OEFPAtomType.FormalCharge* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *BondOrder*).



HCount

This flag indicates that number of hydrogens (the value returned by the `OEAtomBase.GetTotalHCount` method) is encoded into the generated fingerprint. *Table: Example of using the HCount option* demonstrates the effect of using the *OEFPAtomType.HCount* flag.

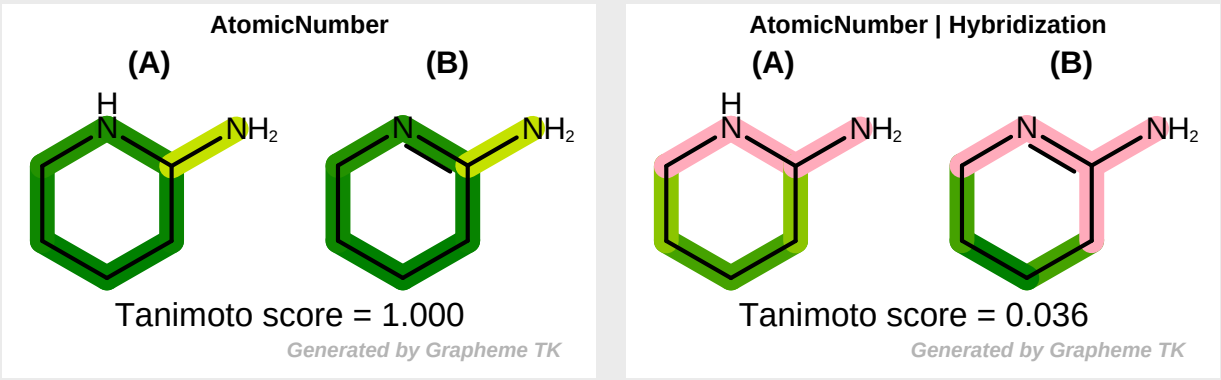
Table 9: **Example of using the** *OEFPAtomType.HCount* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *InRing*).



Hybridization

This flag indicates that hybridization (the value returned by the `OEAtomBase.GetHyb` method) is encoded into the generated fingerprint. *Table: Example of using the Hybridization option* demonstrates the effect of using the *OEFPAtomType.Hybridization* flag.

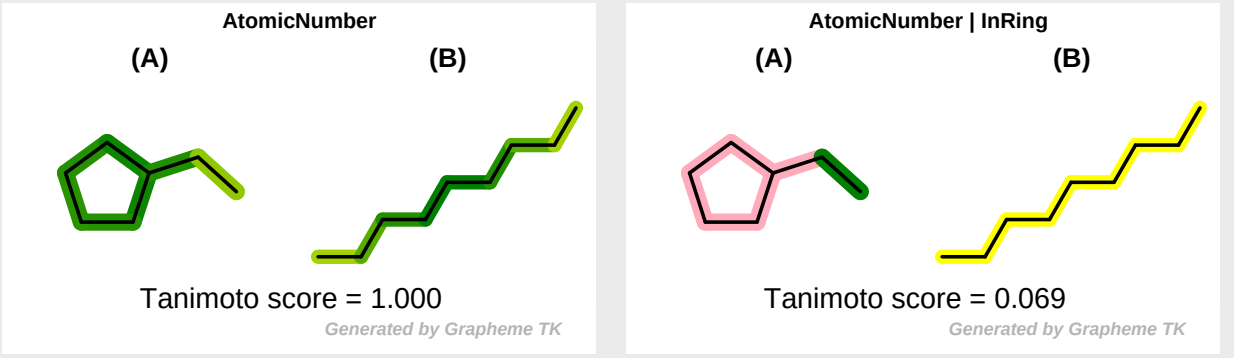
Table 10: **Example of using the** *OEFPAtomType.Hybridization* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *InRing*).



InRing

This flag indicates that atom topology (the value returned by the `OEAtomBase.IsInRing` method) is encoded into the generated fingerprint, *i.e.*, if two fragments (either circular, path or tree) are composed of atoms with different atom topology, then the two fragments will be mapped to different bits of the fingerprint. *Table: Example of using the InRing option* demonstrates the effect of using the *OEFPAtomType.InRing* flag.

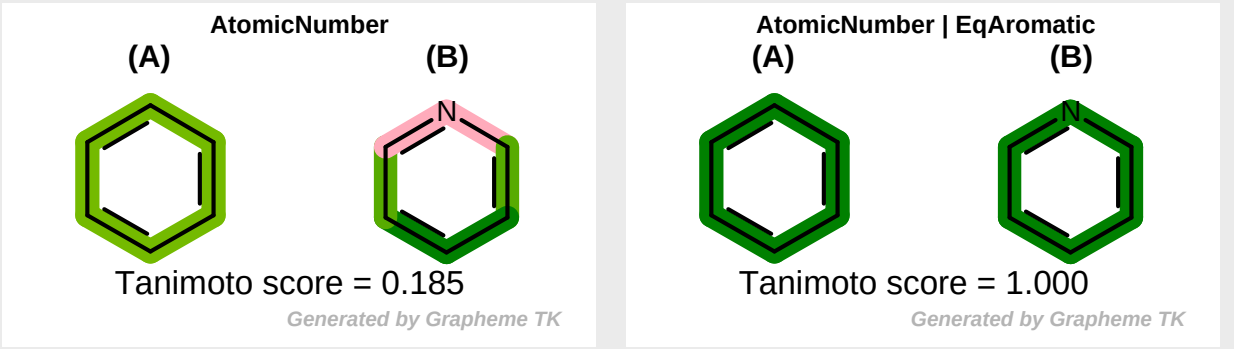
Table 11: **Example of using the** *OEFPAtomType.InRing* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *BondOrder*).



EqAromatic

This flag modifies the meaning of the *OEFPAtomType.AtomicNumber* flag. If the *OEFPAtomType.EqAromatic* flag is set then aromatic atoms are considered equivalent during the fingerprint generation. *Table: Example of using the EqAromatic option* demonstrates the effect of using the *OEFPAtomType.EqAromatic* flag.

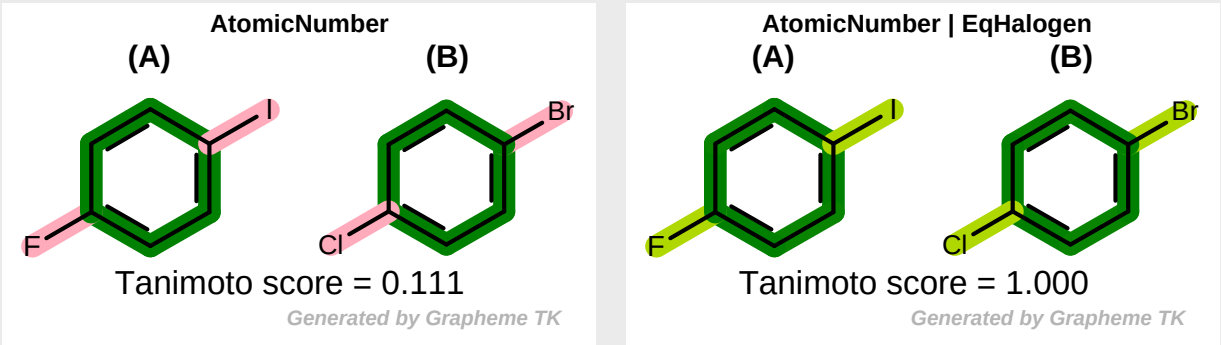
Table 12: **Example of using the** *OEFPAtomType.EqAromatic* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *BondOrder*).



EqHalogen

This flag modifies the meaning of *OEFPAtomType.AtomicNumber* flag. If the *OEFPAtomType.EqHalogen* flag is set then halide atoms (`OEElemNo.F`, `OEElemNo.Cl`, `OEElemNo.Br`, and `OEElemNo.I`) are considered equivalent during the fingerprint generation. *Table: Example of using the EqHalogen option* demonstrates the effect of using the *OEFPAtomType.EqHalogen* flag.

Table 13: **Example of using the** *OEFPAtomType.EqHalogen* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *BondOrder*).



EqHBondAcceptor

This flag modifies the meaning of the *OEFPAtomType.AtomicNumber* flag. If the *OEFPAtomType.EqHBondAcceptor* flag is set then atoms that are perceived as hydrogen bonding acceptors are considered equivalent during the fingerprint generation. The **GraphSim TK** uses the same definition as the MolProp TK to identify hydrogen bond acceptors. See examples in *Figure: Molecules with hydrogen bond acceptor annotation. Table: Example of using the EqHBondAcceptor option* demonstrates the effect of using the *OEFPAtomType.EqHBondAcceptor* flag.

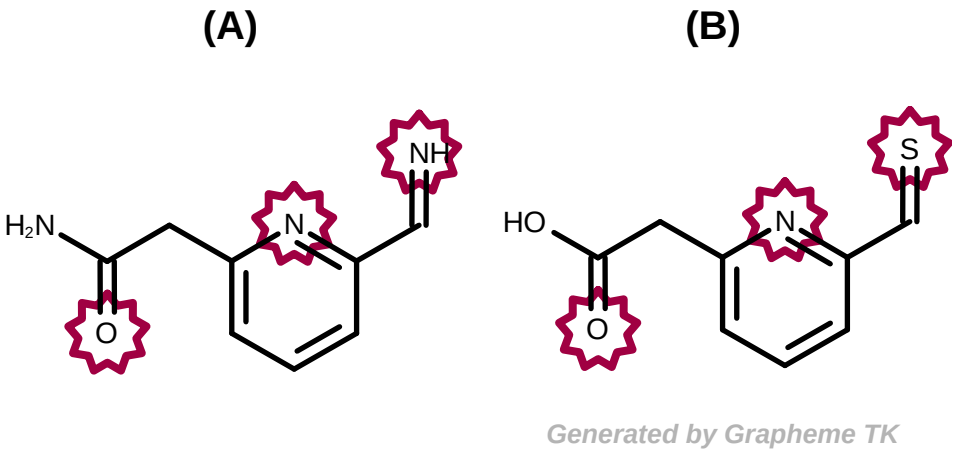
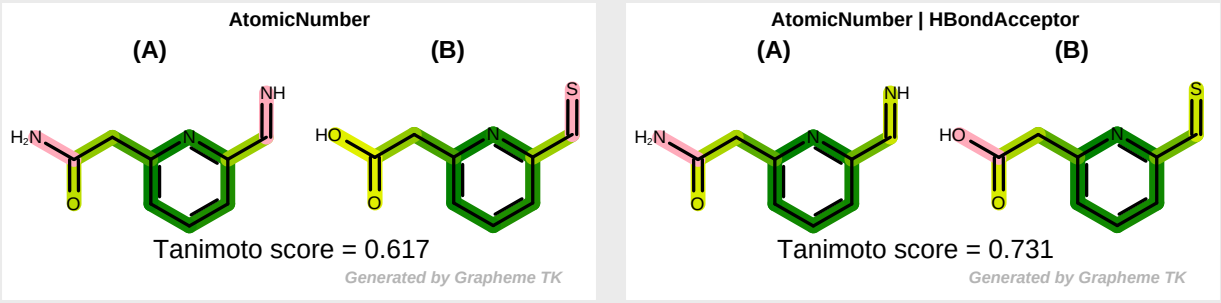


Fig. 4: **Example of molecules with hydrogen bond acceptor annotation**

Table 14: **Example of using the** *OEFPAtomType.EqHBondAcceptor* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *BondOrder*).



EqHBondDonor

This flag modifies the meaning of the *OEFPAtomType.AtomicNumber* flag. If the *OEFPAtomType.EqHBondDonor* flag is set then atoms that are perceived as hydrogen bonding donors are considered equivalent during the fingerprint generation. The **GraphSim TK** uses the same definition as the MolProp TK to identify hydrogen bond donors. See examples in *Figure: Molecules with hydrogen bond donor annotation*. *Table: Example of using the EqHBondDonor option* demonstrates the effect of using the *OEFPAtomType.EqHBondDonor* flag.

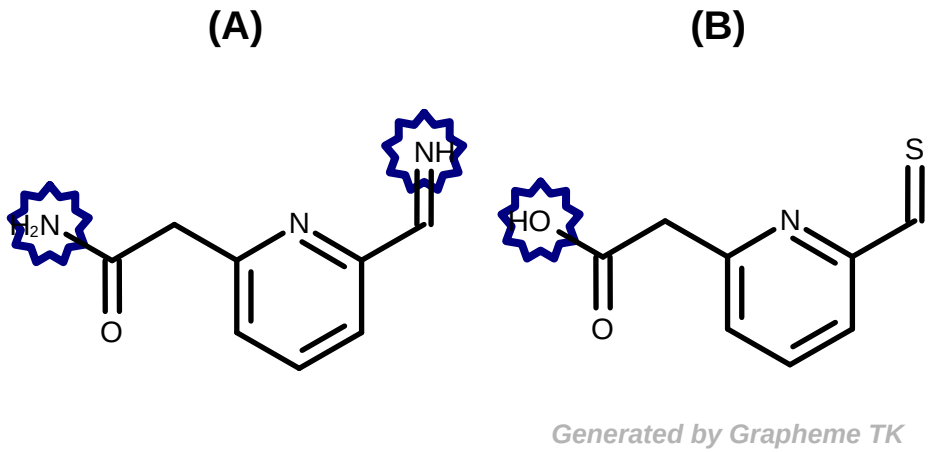
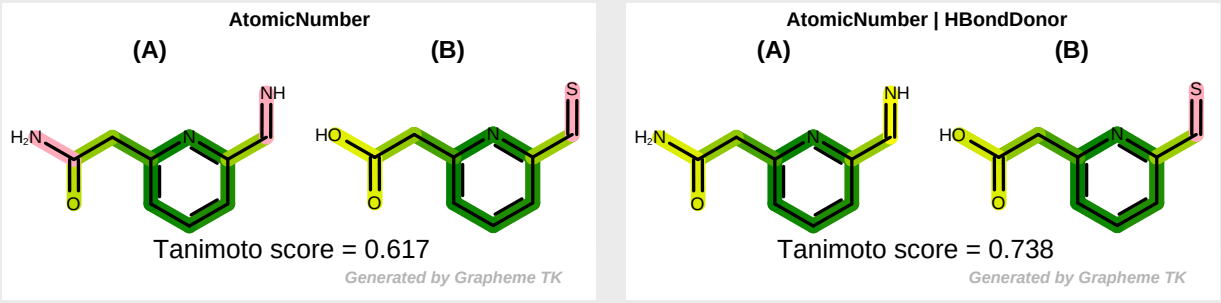


Fig. 5: **Molecules with hydrogen bond donor annotation**

Table 15: **Example of using the** *OEFPAtomType.EqHBondDonor* **option** (*Path*, numbits=4096, bonds=0-5, bond typing = *BondOrder*).



DefaultAtom

Same as *DefaultPathAtom* constant.

DefaultCircularAtom

The bitwise OR'd value of the following atom typing options:

- *OEFPAtomType.AtomicNumber*
- *OEFPAtomType.Aromaticity*
- *OEFPAtomType.Chiral*
- *OEFPAtomType.FormalCharge*
- *OEFPAtomType.HCount*
- *OEFPAtomType.EqHalogen*

See example in *Figure: Circular fingerprint similarity with default circular atom and bond typing*. This constant is used as atom typing parameter when a default *Circular* fingerprint is generated by the following functions:

- *OEMakeFP* (*OEFingerPrint* &, const OEMolBase &, *OEFPType.Circular*)
- *OEMakeCircularFP* (*OEFingerPrint* &, const OEMolBase &)

See also

- *OEFPBondType.DefaultCircularBond* constant

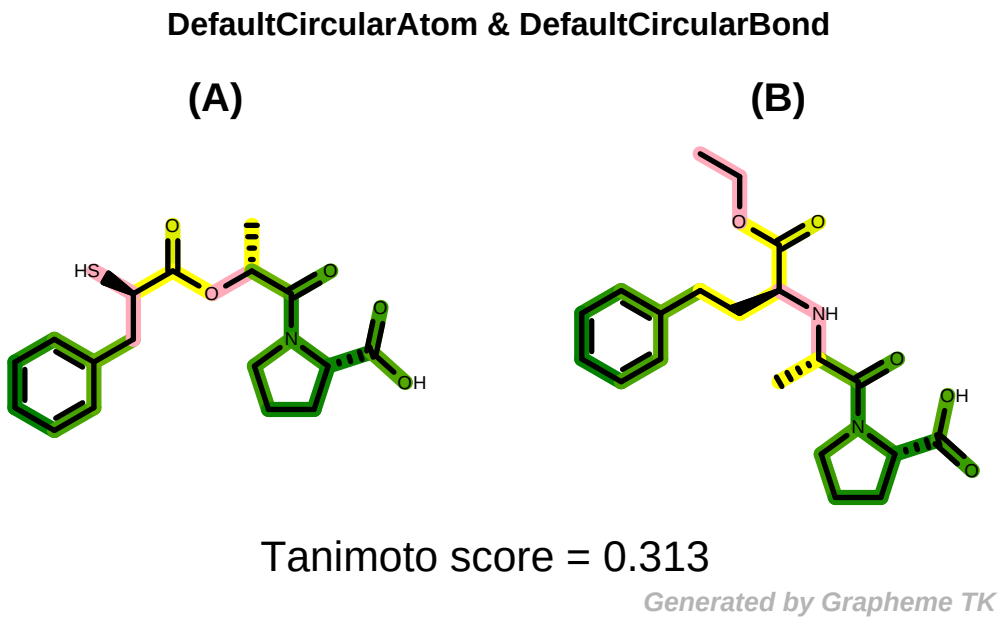


Fig. 6: **Circular fingerprint similarity with default circular atom and bond typing**

DefaultCircularVSAtom

The default atom typing for the circular fingerprint that is designed for virtual screening. It is the same as *DefaultCircularAtom* without *OEFPAtomType.FormalCharge* and *OEFPAtomType.HCount*

DefaultCircularVSAtom1

It is the same as *DefaultCircularAtom* without *OEFPAtomType.FormalCharge*

DefaultCircularVSAtom2

Same as *OEFPAtomType.DefaultCircularVSAtom* constant.

DefaultPathAtom

The bitwise OR'd value of the following atom typing options:

- *OEFPAtomType.AtomicNumber*
- *OEFPAtomType.Aromaticity*

- *OEFPAtomType.Chiral*
- *OEFPAtomType.FormalCharge*
- *OEFPAtomType.HvyDegree*
- *OEFPAtomType.Hybridization*
- *OEFPAtomType.EqHalogen*

See example in *Figure: Path fingerprint similarity with default path atom and bond typing*. This constant is used as atom typing parameter when a default *Path* fingerprint is generated by the following functions:

- *OEMakeFP* (*OEFingerPrint* &, const OEMolBase &, *OEFPType.Path*)
- *OEMakePathFP* (*OEFingerPrint* &, const OEMolBase &)

See also

- *OEFPBondType.DefaultPathBond* constant

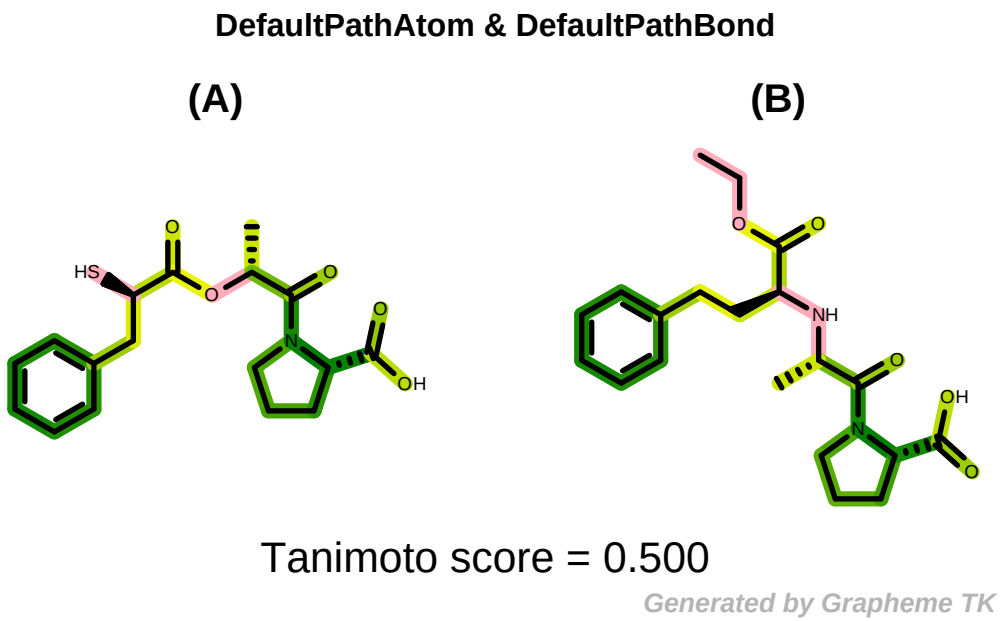


Fig. 7: **Path fingerprint similarity with default path atom and bond typing**

DefaultPathVSAtom

The default atom typing for the path fingerprint that is designed for virtual screening. It is the same as *DefaultPathAtom* without *OEFPAtomType.FormalCharge* and *OEFPAtomType.Hybridization*.

DefaultTreeAtom

The bitwise OR'd value of the following atom typing options:

- *OEFPAtomType.AtomicNumber*
- *OEFPAtomType.Aromaticity*
- *OEFPAtomType.Chiral*
- *OEFPAtomType.FormalCharge*
- *OEFPAtomType.HvyDegree*
- *OEFPAtomType.Hybridization*

See example in *Figure: Tree fingerprint similarity with default tree atom and bond typing*. This constant is used as atom typing parameter when a default *Tree* fingerprint is generated by the following functions:

- *OEMakeFP* (*OEFingerPrint* &, const OEMolBase &, *OEFPType.Tree*)
- *OEMakeTreeFP* (*OEFingerPrint* &, const OEMolBase &)

See also

- *OEFPBondType.DefaultTreeBond* constant

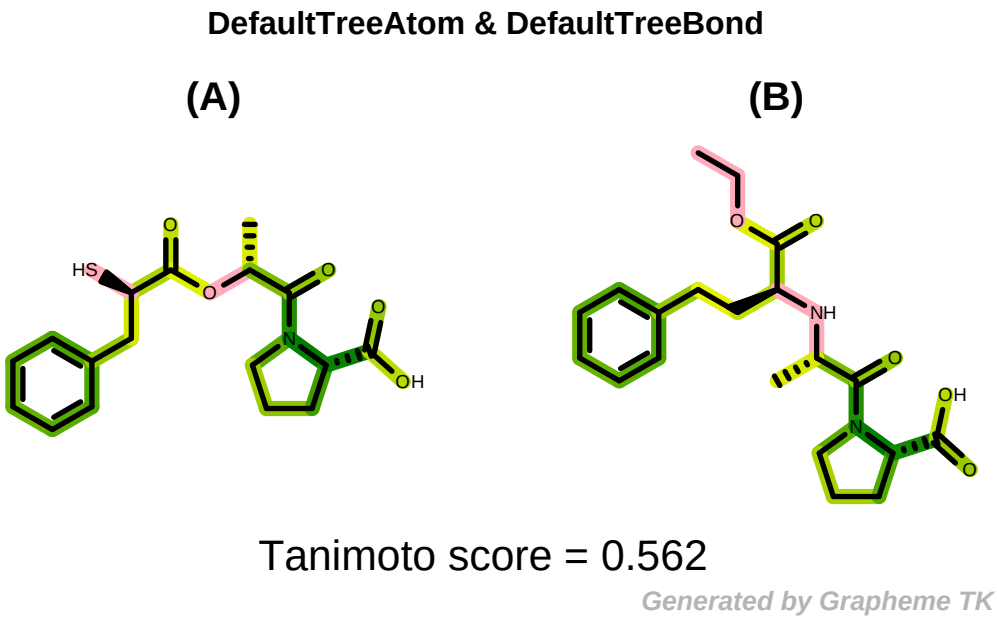


Fig. 8: **Tree fingerprint similarity with default tree atom and bond typing**

DefaultTreeVSAtom

The default atom typing for the tree fingerprint that is designed for virtual screening. It is the same as *DefaultTreeAtom* without *OEFPAtomType.FormalCharge* and *OEFPAtomType.Hybridization*.

None

No atom properties are encoded when generating a fingerprint.

5.2.3 OEFPBondType

This namespace contains bond typing options that can be used when generating *Circular*, *Path* or *Tree* fingerprints. Bond type options control how the bonds of the enumerated circular, path or tree fragments are encoded during the fingerprint generation.

The *OEFPBondType* namespace contains the following constants:

Constant name	Combination of
OEFPBondType.BondOrder	
OEFPBondType.Chiral	
OEFPBondType.DefaultBond	*BondOrder* | *Chiral*
OEFPBondType.DefaultCircularBond	*BondOrder*
OEFPBondType.DefaultCircularVSBond	*BondOrder*
OEFPBondType.DefaultCircularVSBond1	*BondOrder*
OEFPBondType.DefaultCircularVSBond2	*BondOrder*
OEFPBondType.DefaultPathBond	*BondOrder* | *Chiral*
OEFPBondType.DefaultPathVSBond	*BondOrder* | *Chiral*
OEFPBondType.DefaultTreeBond	*BondOrder*
OEFPBondType.DefaultTreeVSBond	*BondOrder*
OEFPBondType.InRing	
OEFPBondType.None	

Note

The constants of the *OEFPBondType* namespace can be **combined** using the bitwise OR operation.

Note

The images in this sections visualize the effect of the various bond typing options.

See also: *Visualizing Molecule Similarity* section

See also

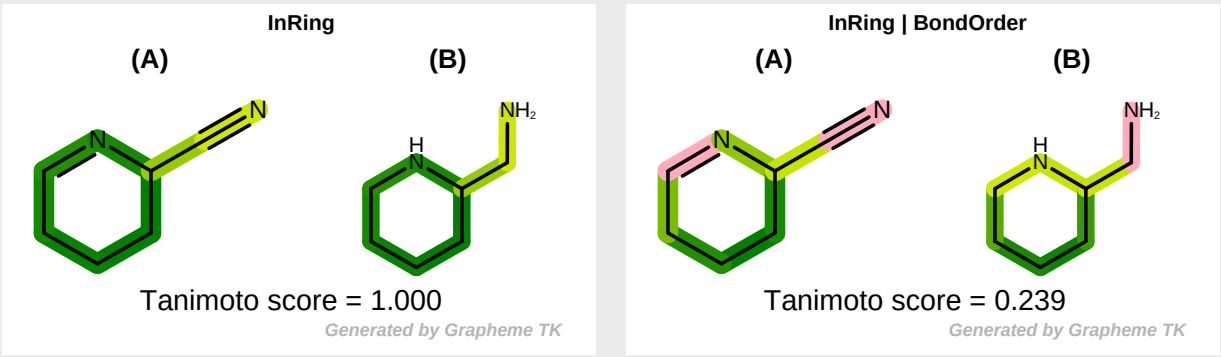
- *OEFPAtomType* namespace

- *OEMakeCircularFP* function
- *OEMakePathFP* function
- *OEMakeTreeFP* function
- *Atom and Bond Typing* section

BondOrder

This flag indicates that bond order information is encoded into the generated fingerprint, *i.e.*, if two fragments (either circular, path or tree) are composed of bonds with different bond orders (the value returned by the `GetOrder` method) then the two fragments will be mapped to different bits of the fingerprint. *Table: Example of using the BondOrder option* demonstrates the effect of using the *OEFPBondType.BondOrder* flag.

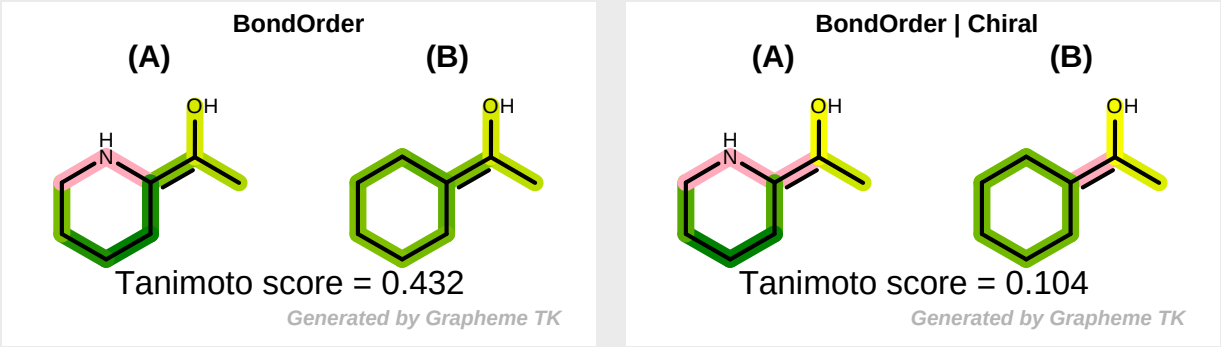
Table 16: **Example of using the *OEFPBondType.BondOrder* option** (*Path*, numbits=4096, bonds=0-5, atom typing = *AtomicNumber*).



Chiral

This flag indicates that chiral and non-chiral bonds (the value returned by the `OEBondBase.IsChiral` method) are distinguished during the circular, path, the tree fingerprint generation. *Table: Example of using the Chiral option* demonstrates the effect of using the *OEFPBondType.Chiral* flag.

Table 17: **Example of using the *OEFPBondType.Chiral* option** (*Path*, numbits=4096, bonds=0-5, atom typing = *AtomicNumber*).



Note

Cis and *trans* chiral bonds can not be distinguished when the `OEFPBondType.Chiral` flag is set. Fingerprints generated for molecules `C\C=C/C` and `C/C=C/C` will always be identical. (See example in *Figure: Example of molecule similarity of stereoisomers*).

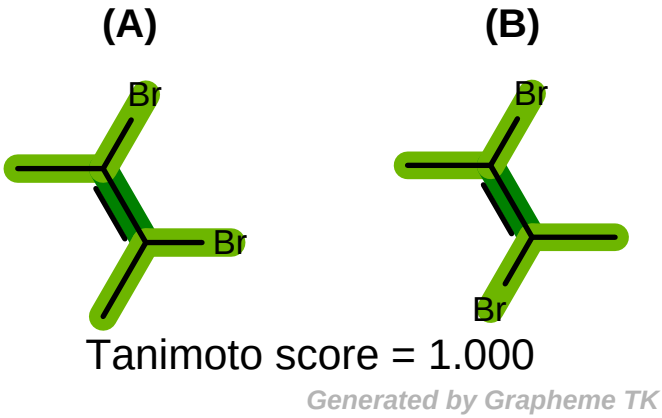


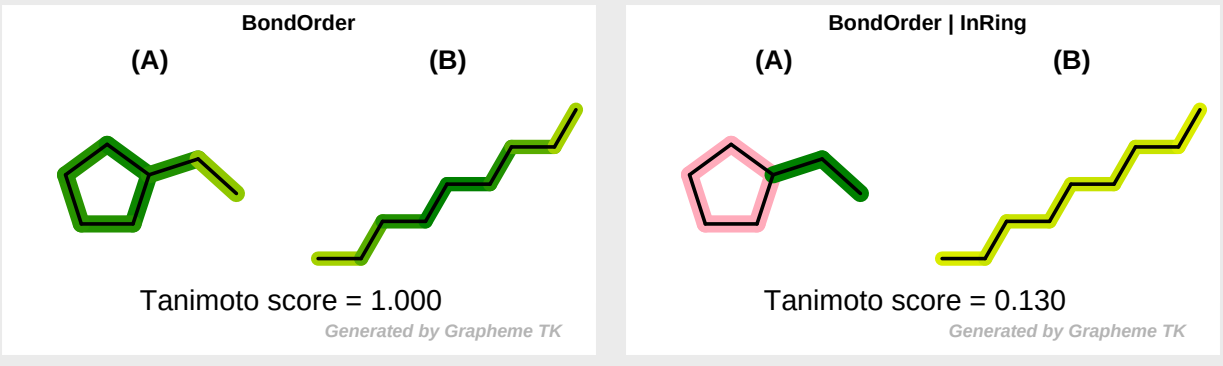
Fig. 9: **Example of molecule similarity of cis-trans stereoisomers**

InRing

This flag indicates that bond topology information is encoded into the generated fingerprint, *i.e.*, if two fragments (either circular, path or tree) are composed of bonds with different bond topology (the value returned by the `OEBondBase.IsInRing` method) then the two fragments will be mapped to different bits of the fingerprint.

Table: Example of using the InRing option demonstrates the effect of using the `OEFPBondType.InRing` flag.

Table 18: **Example of using the `OEFPBondType.InRing` option** (`Path`, numbits=4096, bonds=0-5, atom typing = `AtomicNumber`).



DefaultBond

Same as *OEFPBondType.DefaultPathBond* constant.

DefaultCircularBond

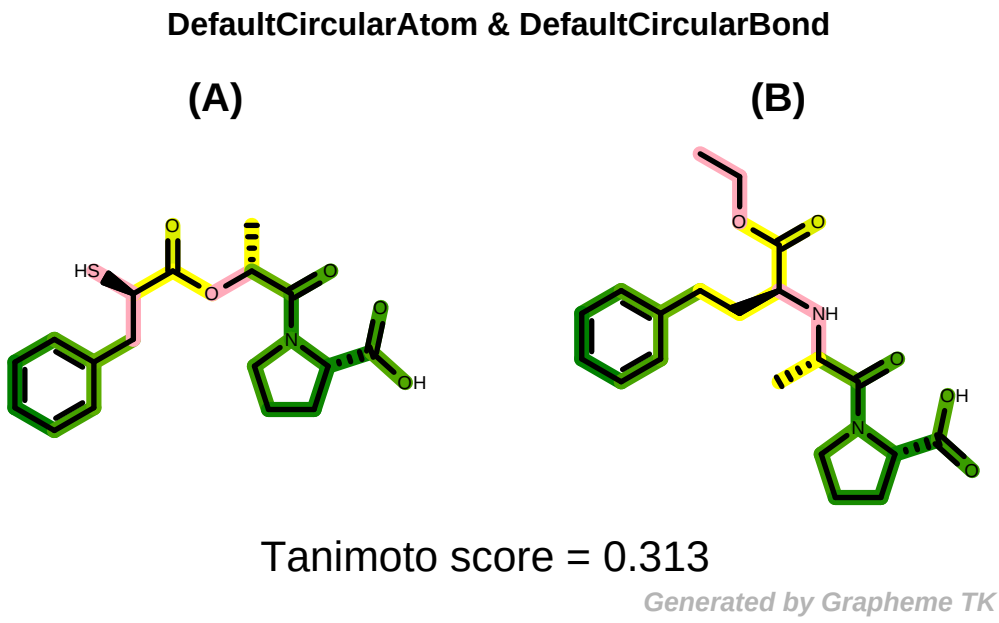
Same as the *OEFPBondType.BondOrder* constant.

See example in *Figure: Circular fingerprint similarity with default circular atom and bond typing*. This constant is used as bond typing parameter when a default *Circular* fingerprint is generated by the following functions:

- *OEMakeFP* (*OEFingerPrint* &, const OEMolBase &, *OEFPType.Circular*)
- *OEMakeCircularFP*

See also

- *OEFPAtomType.DefaultCircularAtom* constant



DefaultCircularVSBond

The default bond typing for the circular fingerprint that is designed for virtual screening. Currently it's the same as *OEFPBondType.DefaultCircularBond*.

DefaultCircularVSBond1

Same as *OEFPBondType.DefaultCircularBond* constant.

DefaultCircularVSBond2

Same as *OEFPBondType.DefaultCircularBond* constant.

DefaultPathBond

The bitwise OR'd value of the following bond typing options:

- *OEFPBondType.BondOrder*
- *OEFPBondType.Chiral*

See example in *Figure: Path fingerprint similarity with default path atom and bond typing*. This constant is used as bond typing parameter when a default *Path* fingerprint is generated by the following functions:

- *OEMakeFP* (*OEFingerPrint* &, const OEMolBase &, *OEFPType.Path*)
- *OEMakePathFP*

See also

- *OEFPAtomType.DefaultPathAtom* constant

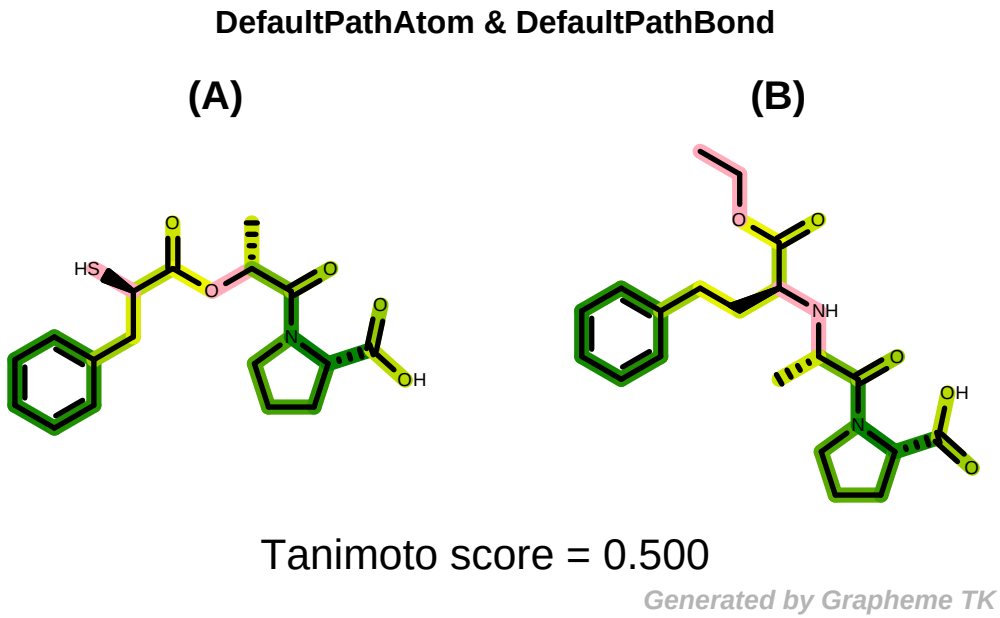


Fig. 10: **Path fingerprint similarity with default tree atom and bond typing**

DefaultPathVSBond

The default bond typing for the tree fingerprint that is designed for virtual screening. Currently it's the same as *OEFPBondType.DefaultPathBond*.

DefaultTreeBond

Same as the *OEFPBondType.BondOrder* constant.

See example in *Figure: Tree fingerprint similarity with default tree atom and bond typing*. This constant is used as bond typing parameter when a default *Tree* fingerprint is generated by the following functions:

- *OEMakeFP* (*OEFingerPrint* &, const OEMolBase &, *OEFPType.Tree*)
- *OEMakeTreeFP*

See also

- *OEFPAtomType.DefaultTreeAtom* constant

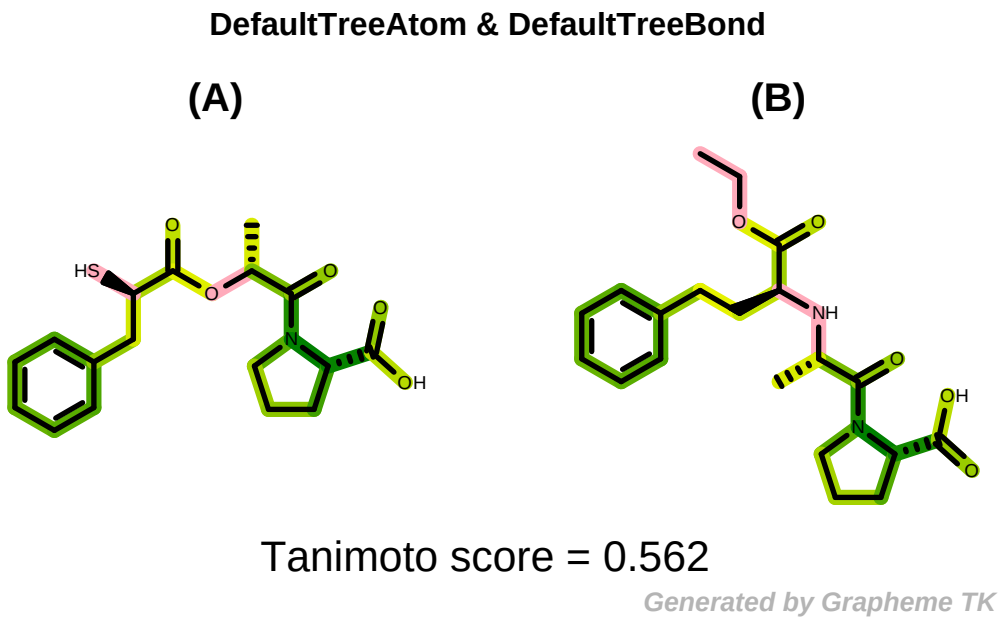


Fig. 11: **Tree fingerprint similarity with default tree atom and bond typing**

DefaultTreeVSBond

The default bond typing for the tree fingerprint that is designed for virtual screening. Currently it's the same as *OEFPBondType.DefaultTreeBond*.

None

No bond properties are encoded when generating a fingerprint.

5.2.4 OEFPDatabaseOptionsSetup

This namespace encodes symbolic constants used as bit-masks to indicate which highlighting interface parameters are being created when invoking the *OEConfigureFPDatabaseOptions* function.

The *OEFPDatabaseOptionsSetup* namespace contains the following constants:

All

The combination of following constants:

- *OEFPDatabaseOptionsSetup.Cutoff*
- *OEFPDatabaseOptionsSetup.DescendingOrder*
- *OEFPDatabaseOptionsSetup.Limit*
- *OEFPDatabaseOptionsSetup.SimFunc*

Cutoff

Passing this constant to the *OEConfigureFPDatabaseOptions* function result in generating the following default interface.

```
Contents of parameter -cutoff
   Aliases : -c
   Type : float
   Allow list : false
   Default : (parameter does not have a default)
   Simple : true
   Required : false
   Brief : Similarity cutoff value
   Detail

      API : OEFPDatabaseOptions::SetCutoff method
      By default there is no similarity cutoff
```

See also

- *OEFPDatabaseOptions.SetCutoff* method

DescendingOrder

Passing this constant to the *OEConfigureFPDatabaseOptions* function result in generating the following default interface.

```
Contents of parameter -descending
   Aliases : -d
   Type : bool
```

(continues on next page)

(continued from previous page)

```
   Allow list : false
   Default : true
   Simple : true
   Required : false
   Brief : Order of similarity scores
   Detail

      API : OEFPDatabaseOptions::SetDescendingOrder method
      If true scores are returned in descending order otherwise in ascending order
```

See also

- *OEFPDatabaseOptions.SetDescendingOrder* method

Limit

Passing this constant to the *OEConfigureFPDatabaseOptions* function result in generating the following default interface.

```
Contents of parameter -limit
   Aliases : -l
   Type : int
   Allow list : false
   Default : 10
   Simple : true
   Required : false
   Legal ranges :
      100000 to 0
   Brief : Maximum number of similarity scores
   Detail

      API : OEFPDatabaseOptions::SetLimit method
      If set to zero, then all of the similarity scores are returned
```

See also

- *OEFPDatabaseOptions.SetLimit* method

SimFunc

Passing this constant to the *OEConfigureFPDatabaseOptions* function result in generating the following default interface.

```
Contents of parameter -simfunc
   Aliases : -s
   Type : string
```

(continues on next page)

(continued from previous page)

```
  Allow list : false
  Default : Tanimoto
  Simple : true
  Required : false
  Legal values : Tanimoto Cosine Dice Euclid Tversky Manhattan
  Brief : Similarity measure
  Detail

     API : OEFPDatabaseOptions::SetSimFunc method and OESimMeasure namespace
```

See also

- *OEFPDatabaseOptions.SetSimFunc* method

5.2.5 OEFPType

This namespace contains constants representing various *OEFingerPrint* types available in the **GraphSim TK**.

See also

- *OEMakeFP*

Circular

This constant represents the circular fingerprint type.

Circular fingerprints are generated by **exhaustively** enumerating **all** circular fragments grown radially from each heavy atom of the molecule up to the given radius and then hashing these fragments into a fixed-length bitvector.

See also

- *Circular* section

Lingo

This constant represents the LINGO fingerprint type.

See also

- *LINGO* section
- LINGO

MACCS166

This constant represents the MACCS key fingerprint type.

See also

- *MACCS* section

Path

This constant represents the path fingerprint type.

Path fingerprints are generated by **exhaustively** enumerating **all** linear paths of a molecular graph up to a given size and then hashing these fragments into a fixed-length bitvector.

See also

- *Path* section

Tree

This constant represents the tree fingerprint type.

A tree fingerprint is generated by **exhaustively** enumerating **all** trees of a molecular graph up to a given size and then hashing these fragments into a fixed-length bitvector.

See also

- *Tree* section

5.2.6 OEPopCountMethod

This namespace contains constants representing possible popcount methods available in **GraphSim TK**.

See also

- *OEGetPopCountMethod* function
- *OEIsFastFPDatabaseReady* function

BitCount

The hardware does not support the popcount method but only a slower bitcount method.

NoSupport

No support available.

PopCount

The fast popcount method is available.

5.2.7 OESimMeasure

This namespace contains constants representing the built-in similarity indices of the **GraphSim TK**.

See also

- *OEFPDatabase.SetSimFunc* method
- *Searching with User-defined Similarity Measures* section

Cosine

This constant represents Cosine similarity index.

See also

- *Cosine* section
- *OECosine* function

Dice

This constant represents Dice similarity index.

See also

- *Dice* section
- *OEDice* function

Euclid

This constant represents the Euclidean similarity index.

See also

- *Euclidean* section
- `OEEuclid` function

Manhattan

This constant represents the Manhattan similarity index.

See also

- *Manhattan* section
- `OEManhattan` function

Tanimoto

This constant represents the Tanimoto similarity index.

See also

- *Tanimoto* section
- `OETanimoto` function
- *OETanimotoSim* class

Tversky

This constant represents the Tversky similarity index.

See also

- *Tversky* section
- `OETversky` function
- *OETverskySim* class

5.2.8 OESimSearchStatus

This namespace contains constants representing the various status of the similarity search (`OESubSearchResult.GetSearchStatus`).

Finished

This constant indicates that the similarity search finished successfully.

InProgress

This constant indicates that the similarity search is in progress.

Uninitialized

This constant indicates that the similarity search can not be initialized (for example, attempting a search on an invalid *OEFastFPDatabase* object).

See also

- *OEFastFPDatabase.SortedSearch* method
- *OESimSearchStatusToName* function

5.3 OEGraphSim Functions

5.3.1 OEAreCompatibleDatabases

```
bool OEAreCompatibleDatabases(const OEChem::OEMolDatabase& moldb,
                              const OEFastFPDatabase& fpdb)
```

Returns whether a molecule database and a fingerprint database are constructed from the same molecule file.

moldb
The OEMolDatabase object.

fpdb
The *OEFastFPDatabase* object.

See also

- *OECreateFastFPDatabaseFile* function
- *OEFastFPDatabaseParams* class
- Example code in *Searching fast fingerprint database* section

5.3.2 OEConfigureFPDatabaseMemoryType

```
bool OEConfigureFPDatabaseMemoryType(OESystem::OEInterface& itf,
                                     unsigned int defvalue =␣
↪OEFastFPDatabaseMemoryType::Default,
                                     std::string name = "-memorytype",
                                     std::string alias = "-m")
```

Adds a fingerprint database memory type parameter to the given interface.

itf
The interface being configured.

defvalue
Default parameter value.

name
The name of the parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEFastFPDatabaseMemoryType* namespace
- *OEGetFPDatabaseMemoryType* function
- Example code in *Searching fast fingerprint database* section

5.3.3 OEConfigureFPDatabaseOptions

```
bool OEConfigureFPDatabaseOptions(OESystem::OEInterface& itf,
                                  const OEFPDatabaseOptions& defopts)
```

Configures the given interface to add parameters for fingerprint database search options.

itf
The interface being configured.

defopts
The default fingerprint database search options.

```
bool OEConfigureFPDatabaseOptions(OESystem::OEInterface& itf,
                                  unsigned int config = OEFPDatabaseOptionsSetup::All)
```

Configures the given interface to add parameters for default fingerprint database search options.

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 *OEFPDatabaseOptionsSetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEFPDatabaseOptions* class
- *OESetupFPDatabaseOptions* function
- Example code in *Building and searching fingerprint database* section
- Example code in *Searching fast fingerprint database* section

5.3.4 OEConfigureFingerPrint

```
bool OEConfigureFingerPrint(OESystem::OEInterface& itf,
                            const OEFPTypeBase* deffptype)
```

Configures the given interface to add the following parameters for user-defined fingerprint type:

```
fingerprint options
 -atomtype : Fingerprint atom type
 -bondtype : Fingerprint bond type
 -fptype : Fingerprint type
 -maxdistance : Maximum number of bonds/radius in path/circular/tree
                fingerprint
 -mindistance : Minimum number of bonds/radius in path/circular/tree
                fingerprint
 -numbits : Size of bitvector
```

itf
The interface being configured.

deffptype
The default fingerprint type.

See also

- OEInterface class in the **OEChem TK** manual
- *OEFPTypeBase* class
- *OESetupFingerPrint* function
- Example code in *Generating fingerprint file for fast fingerprint search* section
- Example code in *Building and searching fingerprint database* section

5.3.5 OECosine

```
float OECosine(const OEFingerPrint& fpA, const OEFingerPrint& fpB)
```

Calculates the Cosine similarity value of two *OEFingerPrint* objects.

Formula:

$$Sim_{Cosine}(A,B) = \frac{bothAB}{\sqrt{(onlyA+bothAB)*(onlyB+bothAB)}}$$

See also

- *Cosine* section

5.3.6 OECreateFastFPDatabaseFile

```
bool OECreateFastFPDatabaseFile(const std::string& fpdbfname,
                                const std::string& molfname,
                                const OEFPTypeBase* fptype)
```

(continues on next page)

(continued from previous page)

```
bool OECreateFastFPDatabaseFile(const std::string& fpdbfname,
                                const std::string& molfname,
                                const OECreateFastFPDatabaseOptions& opts)
```

Generates a binary fingerprint file for the given molecule file.

fpdbfname
The name of the fingerprint file. The file name has to have a `.fpbin` extension.

molfname
The name of the molecule file.

fptype
The type of the fingerprint that will be generated to each molecule.

opts
The *OECreateFastFPDatabaseOptions* object that encapsulates properties that determine how fingerprints are generated.

Note

GraphSim TK currently only supports the popcount search method for fingerprints with the size of multiple of `256`. This means that the *`OEFPType.MACCS166`* fingerprint type is currently not supported. When using other custom fingerprint types (see *User-defined Fingerprint* section) the size of the fingerprint must be a multiple of `256` in order to be able to search it with the popcount method.

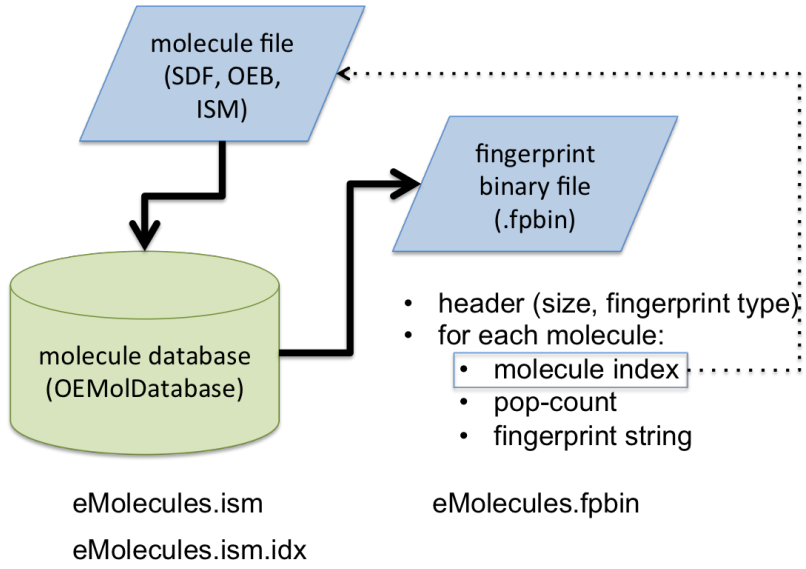


Fig. 12: **Schematic representation of fast fingerprint generation process**

See also

- *OEFastFPDatabaseParams* class
- Example code in *Generating fingerprint file for fast fingerprint search* section

5.3.7 OEDice

```
float OEDice(const OEFingerPrint& fpA, const OEFingerPrint& fpB)
```

Calculates the Dice similarity value of two *OEFingerPrint* objects.

Formula:

$$Sim_{Dice}(A,B) = \frac{2*bothAB}{onlyA+onlyB+2*bothAB}$$

See also

- *Dice* section

5.3.8 OEEuclid

```
float OEEuclid(const OEFingerPrint& fpA, const OEFingerPrint& fpB)
```

Calculates the Euclidean similarity value of two *OEFingerPrint* objects.

Formula:

$$Sim_{Euclid}(A,B) = \sqrt{\frac{bothAB+neitherAB}{onlyA+onlyB+bothAB+neitherAB}}$$

See also

- *Euclidean* section

5.3.9 OEGetBitCounts

```
bool OEGetBitCounts(const OEFingerPrint& fpA,
                    const OEFingerPrint& fpB, unsigned int* onlyA,
                    unsigned int* onlyB, unsigned int* bothAB,
                    unsigned int* neitherAB)
```

fpA*, *pfB
The two *OEFingerPrint* objects being compared.

onlyA
The number of bits set "on" in fingerprint *fpA* but not in *fpB*.

onlyB
The number of bits set "on" in fingerprint *fpB* but not in *fpA*.

bothAB
The number of bits set "on" in both *OEFingerPrint* objects.

neitherAB
The number of bits unset in both *OEFingerPrint* objects.

See also

- *User-defined Similarity Measures* section

5.3.10 OEGetCircularFPType

```
const OEFPTypeBase* OEGetCircularFPType(unsigned int numbits,
                                        unsigned int minradius,
                                        unsigned int maxradius,
                                        unsigned int atype,
                                        unsigned int btype)
```

Returns an *OEFPType.Circular* fingerprint type generated with the given parameters.

numbits
The size of the fingerprint in bits. This number has to be larger than or equal to 2^4 and smaller than 2^{16}.

minradius, ***maxradius***
The smallest and largest circular fragments (in radius) that are enumerated during the fingerprint generation. All enumerated circular fragments are hashed into the *OEFingerPrint* object.

atype
Defines which atom properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPAtomType* namespace.

btype
Defines which bond properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPBondType* namespace.

See also

- *OEFPTypeBase* class

5.3.11 OEGetFPAtomType

```
std::string OEGetFPAtomType(unsigned int value)
```

Returns the string representation of a value that is from the *OEFPAtomType* namespace. For example, the string representation of *OEFPAtomType.AtomicNumber* | *OEFPAtomType.Aromaticity* value is the `"AtmNum|Arom"` string.

Table 19: **The string representation of various fingerprint atom types**

Atom type	String representation
OEFPAtomType.AtomicNumber	"AtmNum"
OEFPAtomType.Aromaticity	"Arom"
OEFPAtomType.Chiral	"Chiral"
OEFPAtomType.FormalCharge	"FCharge"
OEFPAtomType.HvyDegree	"HvyDeg"
OEFPAtomType.Hybridization	"Hyb"
OEFPAtomType.InRing	"InRing"
OEFPAtomType.HCount	"HCount"
OEFPAtomType.EqAromatic	"EqArom"
OEFPAtomType.EqHalogen	"EqHalo"
OEFPAtomType.EqHBondAcceptor	"EqHBAcc"
OEFPAtomType.EqHBondDonor	"EqHBDon"

```
unsigned int OEGetFPAtomType(const std::string& expression)
```

This function is the inverse of the previous *OEGetFPAtomType* function *i.e.* it converts the string representation into a value from the *OEFPAtomType* namespace.

5.3.12 OEGetFPBondType

```
std::string OEGetFPBondType(unsigned int value)
```

Returns the string representation of a value that is from the *OEFPBondType* namespace. For example, the string representation of *OEFPBondType.BondOrder* | *OEFPBondType.Chiral* value is the: `"Order|Chiral"` string.

Table 20: **The string representation of various fingerprint bond types**

Bond type	String representation
OEFPBondType.BondOrder	"Order"
OEFPBondType.Chiral	"Chiral"
OEFPBondType.InRing	"InRing"

```
unsigned int OEGetFPBondType(const std::string& expression)
```

This function is the inverse of the previous *OEGetFPBondType* function *i.e.* it converts the string representation into a value from the *OEFPBondType* namespace.

5.3.13 OEGetFPCoverage

```
OESystem::OEIterBase<OEChem::OEAtomBondSet>* OEGetFPCoverage(const OEChem::OEMolBase&␣
↪mol,
                                                             const OEFPTypeBase* fptype,
                                                             bool unique = false)
```

Returns an iterator over the fragments that are generated when a fingerprint is constructed.

mol
: The OEMolBase objects from which the fingerprint fragments are generated.

fptype
: The type of the fingerprint that also stores the parameters that are used when the fingerprint is generated.

unique
: If true, then the *OEGetFPCoverage* function returns only unique fragments. A fragment (i.e. a subgraph) is considered unique if it differs from all other subgraphs identified previously by at least one atom or bond.

Note

The *OEGetFPCoverage* function is not available for the *OEFPType.Lingo* fingerprint type.

See also

- *Fingerprint Coverage* chapter
- *OEGetFPOverlap* function

- *OEGetFPPatterns* function

5.3.14 OEGetFPDatabaseMemoryType

```
unsigned int OEGetFPDatabaseMemoryType(OESystem::OEInterface& itf,
                                       const std::string name = "-memorytype")
```

Returns the fingerprint database memory type initialized in the given interface.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEFastFPDatabaseMemoryType* namespace
- *OEConfigureFPDatabaseMemoryType* function
- Example code in *Searching fast fingerprint database* section

5.3.15 OEGetFPOverlap

```
OESystem::OEIterBase<OEChem::OEMatchBase>* OEGetFPOverlap(const OEChem::OEMolBase& qmol,
                                                          const OEChem::OEMolBase& tmol,
                                                          const OEFPTypeBase* fptype)
```

Returns an iterator over all **unique** matches (common fragments) found between two molecules based on the given fingerprint type. This means that these common molecular fragments are mapped into the same bit when fingerprints are generated with the given type.

qmol, tmol
The two OEMolBase objects from which the fingerprint overlap is generated.

fptype
The type of the fingerprint that also stores the parameters that are used when the fingerprint is generated.

Note

The *OEGetFPOverlap* function is not available for the *OEFPType.Lingo* fingerprint type.

Note

Two matches are considered to be equivalent only if they store the same sets of target and pattern atoms and bonds. The correspondence between the pattern and target atoms and bonds are not considered.

See also

- *Fingerprint Overlap* chapter
- *OEGetFPCoverage* function
- *OEGetFPPatterns* function

5.3.16 OEGetFPPatterns

```
OESystem::OEIterBase<OEFPPattern>* OEGetFPPatterns(const OEChem::OEMolBase& mol,
                                                   const OEFPTypeBase* fptype)
```

Returns an iterator over the patterns (*OEFPPattern*) that are generated when a fingerprint is constructed.

mol
The OEMolBase objects from which the fingerprint patterns are generated.

fptype
The type of the fingerprint that also stores the parameters that are used when the fingerprint is generated.

Note

The *OEGetFPPatterns* function is not available for the *OEFPType.Lingo* and *OEFPType.MACCS166* fingerprint types.

See also

- *Fingerprint Patterns* chapter
- *OEGetFPCoverage* function
- *OEGetFPOverlap* function

5.3.17 OEGetFPType

```
const OEFPTypeBase* OEGetFPType(const unsigned int fptype,
                                const bool vs = false,
                                const unsigned vsversion = 1u)
```

Returns a *const* pointer of a default fingerprint type.

fptype
The type of the fingerprint. This value has to be from the *OEFPType* namespace.

vs
If `true` and the fingerprint type is either *Path*, *Tree* or *Circular*, then the atom and bond typings designed for virtual screening will be used. Any other cases this parameter will be ignored.

vsversion
The version for the virtual screening fingerprint. Passing a value of `0u` requests the latest version of the fingerprint from the current toolkit. Supported values are `1u` (pre-2025.1 toolkits), and `0u` or `2u` (post-2025.1 toolkits).

Note

The boolean argument signature for this function to request virtual screening fingerprints will be deprecated in favor of the explicit API function *OEGetVSFPType* in future toolkit releases.

See also

- *OEFPAtomType* namespace
- *OEFPBondType* namespace
- *OEGetVSFPType*

```
const OEFPTypeBase* OEGetFPType(const std::string& fptypestring)
```

Returns a *const* pointer of a fingerprint type of the given valid string representation. If the string representation is not valid, it will return a NULL pointer.

See also

- *OEIsValidFPTypeString* function
- *OEFPTypeBase.GetFPTypeString* method
- *OEFPTypeParams* class
- *OEGetCircularFPType* function
- *OEGetPathFPType* function
- *OEGetTreeFPType* function

5.3.18 OEGetFingerPrintVersion

```
unsigned short OEGetFingerPrintVersion(unsigned int fptype)
```

Returns the hashed number representation of the version number of a built-in fingerprint type. Please use the *OEGetFingerPrintVersionString* function to convert the returned number into human-readable representation.

The *OEGetFingerPrintVersion* function returns zero and throws a warning for unknown fingerprint type.

fptype
The type of the fingerprint. This value has to be from the *OEFPType* namespace.

See also

- *OEGetFingerPrintVersionString* function
- *OEFPTypeBase.GetFPVersion* method
- *OEFPTypeBase.GetFPVersionString* method

5.3.19 OEGetFingerPrintVersionString

```
std::string OEGetFingerPrintVersionString(unsigned short version)
```

Returns the string representation of a fingerprint version number returned by either *OEGetFingerPrintVersion* function or *OEFPTypeBase.GetFPVersion* method.

See also

- *OEGetFingerPrintVersion* function
- *OEFPTypeBase.GetFPVersion* method
- *OEFPTypeBase.GetFPVersionString* method

5.3.20 OEGetPathFPType

```
const OEFPTypeBase* OEGetPathFPType(unsigned int numbits,
                                    unsigned int minbonds,
                                    unsigned int maxbonds,
                                    unsigned int atype,
                                    unsigned int btype)
```

Returns an *OEFPType.Path* fingerprint type generated with the given parameters.

numbits
The size of the fingerprint in bits. This number has to be larger than or equal to 2^4 and smaller than 2^{16}.

minbonds*, *maxbonds
The smallest and largest paths (in bonds) that are enumerated during the fingerprint generation. All enumerated paths are hashed into the *OEFingerPrint* object.

atype
Defines which atom properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPAtomType* namespace.

btype
Defines which bond properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPBondType* namespace.

See also

- *OEFPTypeBase* class

5.3.21 OEGetPopCountMethod

```
unsigned int OEGetPopCountMethod()
```

Returns whether popcount method is supported on the hardware. The return value is taken from the *OEPopCountMethod* namespace.

See also

- *OEPopCountMethod* namespace
- *OEIsFastFPDatabaseReady* function

5.3.22 OEGetSimilarityMeasureName

```
std::string OEGetSimilarityMeasureName(unsigned int simtype)
```

Returns the string representation of a built-in similarity measure.

simtype
This value has to be from the *OESimMeasure* namespace.

5.3.23 OEGetTreeFPType

```
const OEFPTypeBase* OEGetTreeFPType(unsigned int numbits,
                                    unsigned int minbonds,
                                    unsigned int maxbonds,
                                    unsigned int atype,
                                    unsigned int btype)
```

Returns an *OEFPType.Tree* fingerprint type generated with the given parameters.

numbits
The size of the fingerprint in bits. This number has to be larger than or equal to 2^4 and smaller than 2^{16}.

minbonds*, *maxbonds
The smallest and largest tree fragments (in bonds) that are enumerated during the fingerprint generation. All enumerated tree fragments are hashed into the *OEFingerPrint* object.

atype
Defines which atom properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPAtomType* namespace.

btype
Defines which bond properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPBondType* namespace.

See also

- *OEFPTypeBase* class

5.3.24 OEGetVSFPType

```
const OEFPTypeBase* OEGetVSFPType(const unsigned int fptype,
                                  const unsigned vsversion = 0u)
```

This function returns a *const* pointer of a default virtual screening fingerprint type.

fptype
The type of the fingerprint. This value has to be from the *OEFPType* namespace. If the fingerprint type is either

Path, *Tree*, or *Circular*, then the atom and bond typings designed for virtual screening will be used. Any other fingerprint types that have no virtual screening equivalent will emit an error and return null.

vsversion
The version for the virtual screening fingerprint. Passing a value of `0u` requests the latest version of the fingerprint from the current toolkit. Supported values are `1u` (pre-2025.2 toolkits), and `0u` or `2u` (2025.2 toolkits onwards).

See also

- *OEFPAtomType* namespace
- *OEFPBondType* namespace
- *OEGetFPType*
- *OEMakeVSFP*

5.3.25 OEGraphSimGetArch

```
const char* OEGraphSimGetArch()
```

5.3.26 OEGraphSimGetLicensee

```
bool OEGraphSimGetLicensee(std::string& licensee)
```

5.3.27 OEGraphSimGetPlatform

```
const char* OEGraphSimGetPlatform()
```

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 an `x86_64` Red Hat Enterprise Linux box this would return `redhat-RHEL5-g++4.1-x64`.

5.3.28 OEGraphSimGetRelease

```
const char* OEGraphSimGetRelease()
```

Returns the release name of the **GraphSim TK** library being used. This returns a value similar to `1.0` for production versions of the library, and `1.0 debug` for the checking version of the library.

5.3.29 OEGraphSimGetSite

```
bool OEGraphSimGetSite(std::string& site)
```

5.3.30 OEGraphSimGetVersion

```
unsigned int OEGraphSimGetVersion()
```

5.3.31 OEGraphSimIsLicensed

```
bool OEGraphSimIsLicensed(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 **GraphSim TK** functionality.

The 'feature' argument can be used to check for a valid license to **GraphSim 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 **GraphSim TK** license.

See also

- `OEChemIsLicensed` function

5.3.32 OEGraphsimIsGPUReady

```
bool OEGraphsimIsGPUReady()
```

Determines whether a GPU is present and available for use on the current system.

See also

GPU-Related Requirements section in the *OEGraphSim TK* manual.

5.3.33 OEInitFPFieldHandler

```
OEDataFlow::OEFieldTypeHandler* OEInitFPFieldHandler()
```

Creates and returns the field type handler that enables *OEFingerPrint* objects to be stored and retrieved as typed fields on record objects. Call this function once at startup to register fingerprint support for use with the **Platform TK** data record system.

The caller takes ownership of the returned handler.

See also

- *OEInitFPVecFieldHandler* function

5.3.34 OEInitFPVecFieldHandler

```
OEDataFlow::OEFieldTypeHandler* OEInitFPVecFieldHandler()
```

Creates and returns the field type handler that enables vectors of *OEFingerPrint* objects to be stored and retrieved as typed fields on record objects. Call this function once at startup to register fingerprint vector support for use with the **Platform TK** data record system.

The caller takes ownership of the returned handler.

See also

- *OEInitFPFieldHandler* function

5.3.35 OEIsFPType

```
bool OEIsFPType(const OEFingerPrint& fp, unsigned int fptype)
```

Returns true if the given *OEFingerPrint* object's fingerprint type is identical to `fptype`. `fptype` has to be a value from the *OEFPType* namespace.

See also

- *OEIsSameFPType* function

5.3.36 OEIsFastFPDatabaseReady

```
bool OEIsFastFPDatabaseReady()
```

Returns whether any fast fingerprint (pop count) method supported on the hardware.

See also

- *OEGetPopCountMethod* function

5.3.37 OEIsSameFPType

```
bool OEIsSameFPType(const OEFingerPrint& fpA, const OEFingerPrint& fpB)
```

Returns whether or not the types (*OEFPTypeBase*) of the two *OEFingerPrint* objects are the same.

Note

Two user-defined path fingerprints have the same type only if they have been generated with the same parameters (*i.e.* number of bits, minimum and maximum path size and atom and bond typing options) and have the same version number.

Warning

Similarity measure functions (such as *OETanimoto*) only work on fingerprints with identical type (*OEFPTypeBase*).

5.3.38 OEIsValidFPTypeString

```
bool OEIsValidFPTypeString(const std::string& fptypestring)
```

Returns true if the given string is a valid representation of any available fingerprint types.

See also

- *OEFPTypeBase.GetFPTypeString* method

5.3.39 OEMakeCircularFP

```
bool OEMakeCircularFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol)
```

Generates the *OEFingerPrint* object as a *Circular* fingerprint using default parameters.

fp
: The *OEFingerPrint* object that is being initialized.

mol
: The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

The following code snippet shows how to print out the string representation of a *OEFPType.Circular* fingerprint. This string encodes information about default parameters used to generate a *OEFPType.Circular* fingerprint.

```
OEFPTypeBase fptype = OEGraphSim.OEGetFPType(OEFPType.Circular);
Console.WriteLine(fptype.GetFPTypeString());
```

The output of the above code snippet is:

```
Circular,ver=2.0.0,size=4096,radius=0-5,atype=AtmNum|Arom|Chiral|FCharge|HCount|EqHalo,
↪btype=Order
```

See also

- *OEFPTypeBase.GetFPTypeString* method

```
bool OEMakeCircularFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol,
                      unsigned int numbits, unsigned int minradius,
                      unsigned int maxradius, unsigned int atype,
                      unsigned int btype)
```

Generates the *OEFingerPrint* object as a *Circular* fingerprint using the given parameters.

fp
: The *OEFingerPrint* object that is being initialized.

mol
The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

numbits
The size of the fingerprint in bits. This number has to be larger than or equal to 2^4 and smaller than 2^{16}.

minradius, ***maxradius***
The smallest and largest circular fragments (in radius) that are enumerated during the fingerprint generation. All enumerated circular fragments are hashed into the *OEFingerPrint* object.

atype
Defines which atom properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPAtomType* namespace.

btype
Defines which bond properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPBondType* namespace.

Note

For an empty molecule, both *OEMakeCircularFP* function will return `false` and the type of the fingerprint will be set to `0` (*OEFingerPrint.GetFPTypeBase*).

See also

- *User-defined Fingerprint* chapter
- *OEFPAtomType* namespace
- *OEFPBondType* namespace

5.3.40 OEMakeFP

```
bool OEMakeFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol,
              const unsigned fptype, const bool vs = false)
```

Generates a fingerprint from the given molecule.

fp
The *OEFingerPrint* object that is being initialized.

mol
The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

fptype
The type of the fingerprint. This value has to be from the *OEFPType* namespace.

vs
If `true` and the fingerprint type is either *Path*, *Tree* or *Circular*, then the atom and bond typings designed for virtual screening will be used. Any other cases this parameter will be ignored.

Note

The boolean argument signature for this function to request virtual screening fingerprints will be deprecated in favor of the explicit API function *OEMakeVSFP* in future toolkit releases.

See also

- *OEFPAtomType* namespace
- *OEFPBondType* namespace
- *OEMakeVSFP*

```
bool OEMakeFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol,
              const OEFPTypeBase* fptype)
```

Generates a fingerprint from the given molecule.

fp
The *OEFingerPrint* object that is being initialized.

mol
The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

fptype
The type of the fingerprint specified by an *OEFPTypeBase* pointer.

Note

For an empty molecule, both *OEMakeFP* function will return `false` and the type of the fingerprint will be set to `0` (*OEFingerPrint.GetFPTypeBase*).

5.3.41 OEMakeLingoFP

```
bool OEMakeLingoFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol)
```

Generates the *OEFingerPrint* object as a *LINGO* fingerprint.

fp
The *OEFingerPrint* object that is being initialized.

mol
The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

Note

For an empty molecule, the *OEMakeLingoFP* function will return `false` and the type of the fingerprint will be set to `0` (*OEFingerPrint.GetFPTypeBase*).

See also

- *LINGO* section

5.3.42 OEMakeMACCS166FP

```
bool OEMakeMACCS166FP(OEFingerPrint& fp, const OEChem::OEMolBase& mol)
```

Generates the *OEFingerPrint* object as a *MACCS* fingerprint.

fp
The *OEFingerPrint* object that is being initialized.

mol
The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

Note

For an empty molecule, the *OEMakeMACCS166FP* function will return `false` and the type of the fingerprint will be set to `0` (*OEFingerPrint.GetFPTypeBase*).

See also

- *MACCS* section

5.3.43 OEMakePathFP

```
bool OEMakePathFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol)
```

Generates the *OEFingerPrint* object as a *Path* fingerprint using default parameters.

fp
The *OEFingerPrint* object that is being initialized.

mol
The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

The following code snippet shows how to print out the string representation of a *OEFPType.Path* fingerprint. This string encodes information about default parameters used to generate a *OEFPType.Path* fingerprint.

```
OEFPTypeBase fptype = OEGraphSim.OEGetFPType(OEFPType.Path);
Console.WriteLine(fptype.GetFPTypeString());
```

The output of the above code snippet is:

```
Path,ver=2.0.0,size=4096,bonds=0-5,atype=AtmNum|Arom|Chiral|FCharge|HvyDeg|Hyb|EqHalo,
↪btype=Order|Chiral
```

See also

- *OEFPTypeBase.GetFPTypeString* method

```
bool OEMakePathFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol,
                  unsigned int numbits, unsigned int minbonds,
                  unsigned int maxbonds, unsigned int atype,
                  unsigned int btype)
```

Generates the *OEFingerPrint* object as a *Path* fingerprint using the given parameters.

fp
The *OEFingerPrint* object that is being initialized.

mol
The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

numbits
The size of the fingerprint in bits. This number has to be larger than or equal to 2^4 and smaller than 2^{16}.

minbonds*, *maxbonds
The smallest and largest paths (in bonds) that are enumerated during the fingerprint generation. All enumerated paths are hashed into the *OEFingerPrint* object.

atype
Defines which atom properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPAtomType* namespace.

btype
Defines which bond properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPBondType* namespace.

Note

For an empty molecule, both *OEMakePathFP* function will return `false` and the type of the fingerprint will be set to `0` (*OEFingerPrint.GetFPTypeBase*).

See also

- *User-defined Fingerprint* chapter
- *OEFPAtomType* namespace
- *OEFPBondType* namespace

5.3.44 OEMakeTreeFP

```
bool OEMakeTreeFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol)
```

Generates the *OEFingerPrint* object as a *Tree* fingerprint using default parameters.

fp
The *OEFingerPrint* object that is being initialized.

mol
: The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

The following code snippet shows how to print out the string representation of a *OEFPType.Tree* fingerprint. This string encodes information about default parameters used to generate a *OEFPType.Tree* fingerprint.

```
OEFPTypeBase fptype = OEGraphSim.OEGetFPType(OEFPType.Tree);
Console.WriteLine(fptype.GetFPTypeString());
```

The output of the above code snippet is:

```
Tree,ver=2.0.0,size=4096,bonds=0-4,atype=AtmNum|Arom|Chiral|FCharge|HvyDeg|Hyb,
↪btype=Order
```

See also

- *OEFPTypeBase.GetFPTypeString* method

```
bool OEMakeTreeFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol,
                  unsigned int numbits, unsigned int minbonds,
                  unsigned int maxbonds, unsigned int atype,
                  unsigned int btype)
```

fp
: The *OEFingerPrint* object that is being initialized.

mol
: The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

numbits
: The size of the fingerprint in bits. This number has to be larger than or equal to 2^4 and smaller than 2^{16}.

minbonds*, *maxbonds
: The smallest and largest tree fragments (in bonds) that are enumerated during the fingerprint generation. All enumerated tree fragments are hashed into the *OEFingerPrint* object.

atype
: Defines which atom properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPAtomType* namespace.

btype
: Defines which bond properties are encoded during the fingerprint generation. This value has to be either a value or a set of bitwise OR'd values from the *OEFPBondType* namespace.

Note

For an empty molecule, both *OEMakeTreeFP* function will return `false` and the type of the fingerprint will be set to `0` (*OEFingerPrint.GetFPTypeBase*).

See also

- *User-defined Fingerprint* chapter
- *OEFPAtomType* namespace
- *OEFPBondType* namespace

5.3.45 OEMakeVSFP

```
bool OEMakeVSFP(OEFingerPrint& fp, const OEChem::OEMolBase& mol,
                const unsigned fptype, const unsigned vsversion = 0u)
```

This function generates a virtual screening fingerprint from the given molecule for the requested fingerprint type.

fp
The *OEFingerPrint* object that is being initialized.

mol
The OEMolBase object from which the fingerprint is generated. Aromaticity must be perceived before calling this function (e.g. via `OEAssignAromaticFlags`).

fptype
The type of the fingerprint. This value has to be from the *OEFPType* namespace. If the fingerprint type is either *Path*, *Tree*, or *Circular*, then the atom and bond typings designed for virtual screening will be used. Any other fingerprint types that have no virtual screening equivalent simply return the requested non-VS fingerprint.

vsversion
The version for the virtual screening fingerprint. Passing a value of `0u` requests the latest version of the fingerprint from the current toolkit. Supported values are `1u` (pre-2025.2 toolkits), and `0u` or `2u` (2025.2 toolkits onwards).

See also

- *OEFPAtomType* namespace
- *OEFPBondType* namespace
- *OEMakeFP*
- *OEGetVSFPType*

5.3.46 OEManhattan

```
float OEManhattan(const OEFingerPrint& fpA, const OEFingerPrint& fpB)
```

Calculates the Manhattan similarity value of two *OEFingerPrint* objects.

Formula:

$$Sim_{Manhattan}(A,B) = \frac{onlyA+onlyB}{onlyA+onlyB+bothAB+neitherAB}$$

See also

- *Manhattan* section

5.3.47 OESetFingerPrintValue

```
bool OESetFingerPrintValue(OERecord& rec,
                           const OEField<OEGraphSim::OEFingerPrint>& field,
                           const OEGraphSim::OEFingerPrint& fp)
```

Sets a fingerprint value on a record.

rec
The record on which the fingerprint value is set.

field
The field descriptor that identifies the fingerprint field on the record.

fp
The fingerprint to store.

Returns `true` if the value was successfully stored.

5.3.48 OESetupFPDatabaseOptions

```
bool OESetupFPDatabaseOptions(OEFPDatabaseOptions& opts,
                              const OESystem::OEInterface& itf)
```

Initializes an *OEFPDatabaseOptions* object from the parameters of a given interface.

See also

- OEInterface class in the **OEChem TK** manual
- *OEFPDatabaseOptions* class
- *OEConfigureFPDatabaseOptions* function
- Example code in *Building and searching fingerprint database* section
- Example code in *Searching fast fingerprint database* section

5.3.49 OESetupFingerPrint

```
const OEFPTypeBase* OESetupFingerPrint(const OESystem::OEInterface& itf)
```

Initializes an *OEFPTypeBase* object from the parameters of a given interface.

See also

- OEInterface class in the **OEChem TK** manual
- *OEFPTypeBase* class
- *OEConfigureFingerPrint* function
- Example code in *Generating fingerprint file for fast fingerprint search* section
- Example code in *Building and searching fingerprint database* section

5.3.50 OESimSearchStatusToName

```
std::string OESimSearchStatusToName(const unsigned status)
```

Returns the string representation of the constants in the *OESimSearchStatus* namespace.

See also

- *OEFastFPDatabase.SortedSearch* method

5.3.51 OETanimoto

```
float OETanimoto(const OEFingerPrint& fpA, const OEFingerPrint& fpB)
```

Calculates the *Tanimoto* similarity value of two *OEFingerPrint* objects.

Formula:

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

See also

- *Tanimoto* section

5.3.52 OETversky

```
float OETversky(const OEFingerPrint& fpA, const OEFingerPrint& fpB,
                float a, float b)
```

Calculates the *Tversky* similarity value of two *OEFingerPrint* objects.

Formula:

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

See also

- *Tversky* section

5.3.53 OEFPToNumpyArray

Attention

This functionality is only available in the Python toolkits.

```
def OEFPToNumpyArray(fp, numpyCheck: bool = True) -> "NumpyArray"
def OEFPToNumpyArrayCpp(fp, numpyCheck: bool = True) -> "NumpyArray"
```

Creates a NumPy array from the bits stored in an *OEFingerPrint*. *OEFPToNumpyArray* ` is completely done through Python, while *OEFPToNumpyArrayCpp* creates the array in C++ and converts it into a NumPy array. The user is required to have NumPy installed in their Python environment.

CHAPTER

SIX

RELEASE HISTORY

6.1 GraphSim TK 2.7.1

6.1.1 Minor Bug Fixes

- The hydrogen counts of neighbor atoms when R-groups are removed prior to fingerprint generation have been adjusted.

6.2 GraphSim TK 2.7.0

6.2.1 New Features

- New API functions have been added to explicitly provide access to virtual screening fingerprints which were previously generated with a subtle boolean function argument. See *OEMakeVSFP* and *OEGetVSFPType*.
- New Python function has been added to create a NumPy Array from the bits stored in an *OEFingerPrint*. See *OEFPToNumpyArray*.

6.2.2 Minor Bug Fixes

- A minor tuning of `OEFPAtomType.DefaultCircularVSAtom` for virtual screening of `OEFPType.Circular` fingerprints improved the consistency of virtual screening scores when comparing `OEFPType.Tree`, `OEFPType.Circular`, and `OEFPType.Path` for VS fingerprints.

6.3 GraphSim TK 2.6.2

Minor internal improvements have been made.

6.4 GraphSim TK 2.6.1

6.4.1 Minor bug fixes

- The function *OEGraphsimIsGPUReady* has been safeguarded from a rare crash.

6.5 GraphSim TK 2.6.0

Minor internal improvements have been made.

6.6 GraphSim TK 2.5.7

Minor internal improvements have been made.

6.7 GraphSim TK 2.5.6

Minor internal improvements have been made.

6.8 GraphSim TK 2.5.5

Minor internal improvements have been made.

6.9 GraphSim TK 2.5.4

- Minor internal improvements have been made.

6.10 GraphSim TK 2.5.3

Fall 2021

- Minor internal improvements have been made.

6.11 GraphSim TK 2.5.2

July 2021

Minor internal improvements have been made.

6.12 GraphSim TK 2.5.1

Fall 2020

6.12.1 New features

- GPU-accelerated *OEFastFPDatabase* searches now require a minimum NVIDIA Driver version of 450.x and a minimum GPU compute capability of 3.5. See NVIDIA Drivers and Supported GPUs for more details.

6.13 GraphSim TK 2.5.0

6.13.1 New features

- The following new APIs have been added that enable multi-threaded similarity search and access to a histogram of the similarity score distributions:
  - *OEFastFPDatabase.SortedSearch* method
  - *OESimSearchResult* method
  - *OESimSearchStatus* namespace
  - *OESimSearchStatusToName* function

- Two new methods, *OEFPDatabaseOptions.SetNumProcessors* and *OEFPDatabaseOptions.GetNumProcessors*, have been added that control the number of threads used to search fingerprints when using *OEFastFPDatabase.SortedSearch*.
- New atom and bond typing combinations have been added to generate *tree*, *path*, and *circular* fingerprints that are optimized for virtual screening. The *OEMakeFP* and *OEGetFPType* methods can now generate these new types. See also the *OEFPAtomType* and *OEFPBondType* namespaces.
- Small optimization has been achieved by reducing cache misses during the fingerprint search when using the *OEFastFPDatabase* class.
- New overloads of the *OEFPHistogram.GetBinIdx* and *OEFPHistogram.AddSample* methods have been added that take a `double` parameter.

6.13.2 Major bug fixes

- The binary fingerprint database writer had been accidentally broken in the Oct.2019 release. This has been fixed.

6.13.3 Minor bug fixes

- The default limit of the *OEFPDatabaseOptions* class has been changed from 0 (meaning no limit) to 1000. See also the *OEFPDatabaseOptions.SetLimit* method.

6.13.4 General Notices

- In the C++ **GraphSim TK**, **Fast fingerprints** with the CUDA memory type is not available on RHEL8 or Ubuntu18.

6.14 GraphSim TK 2.4.3

6.14.1 New features

- A new method, *OEFastFPDatabase.GetMoleculeIndex*, has been added.

6.14.2 Minor bug fixes

- *OEIsValidFPTypeString* no longer throws an error and always return `false` if the given string cannot be interpreted.

6.15 GraphSim TK 2.4.2

- Minor internal improvements have been made.

6.16 GraphSim TK 2.4.1

6.16.1 New features

- A new *OEFastFPDatabase.GetFingerPrint* method has been added to allow retrieving fingerprints stored in fast fingerprint databases that are generated in a single-threaded mode.
- New *OEFPDatabaseOptionsSetup.Cutoff* and *OEFPDatabaseOptionsSetup.DescendingOrder* methods have been added to help generate consistent command line interfaces.
- A new *OEFastFPDatabase.GetSparseMatrix* method has been added to allow retrieving a sparse matrix of similarity scores to facilitate clustering. The method will retrieve `limit` scores per fingerprint or all scores above a desired `cutoff` value. The method is available in all memory modes, including *CUDA*.

6.16.2 Minor bug fixes

- The *OEGetFingerPrintVersion* function no longer throws an error for an invalid fingerprint type. It now throws a warning and returns zero.

6.17 GraphSim TK 2.4.0

6.17.1 New features

- **GraphSim TK** now has limited GPU support for fast fingerprints. *CUDA* has been added to the *OEFastFPDatabaseMemoryType* namespace for executing large 2D searches and to have GPU devices at their disposal. This new feature comes with the following additional API points:
  - *OEFastFPDatabase.GetRawScores*
  - *OEFastFPDatabase.GetAllScores*
  - *OEFastFPDatabase.GetHistogram* and *OEFPHistogram* (preliminary)
  - *OEFastFPDatabase.GetVariogram* and *OEFPVariogram* (preliminary)
  - *OEFastFPDatabase.GetMemoryType*
  - *OEFastFPDatabase.GetMemoryTypeString*

  For more information, see the documentation for *OEFastFPDatabaseMemoryType.CUDA*.
- The following API has been added to help generate consistent command line interfaces for 2D molecule similarity examples and utilities:
  - *OEFPDatabaseOptionsSetup*
  - *OEConfigureFPDatabaseOptions*
  - *OEConfigureFPDatabaseMemoryType*
  - *OEGetFPDatabaseMemoryType*

6.18 GraphSim TK 2.3.1

6.18.1 New features

- The following API has been added to provide access to underlying data generated for *OEFPType.Path*, *OEFPType.Tree*, and *OEFPType.Circular* fingerprint types:
  - *OEFPPattern*
  - *OEGetFPPatterns*
- The performance of the *OEGetFPCoverage* function has been improved.
- The following functions have been added to help generate consistent command line interfaces for 2D molecule similarity examples and utilities:
  - *OEConfigureFingerPrint* and the corresponding *OESetupFingerPrint*
  - *OEConfigureFPDatabaseOptions* and the corresponding *OESetupFPDatabaseOptions*

  Generated interfaces include correspondence between each command line option and its related entry point in **OEGraphSim TK**.

6.18.2 Documentation changes

- A new *Fingerprint Patterns* chapter has been added.

6.19 GraphSim TK 2.3.0

6.19.1 New features

- A new API has been added to perform rapid searching of fingerprints using the popcount method:
  - `OECreateFastFPDatabaseFile` function and *OECreateFastFPDatabaseOptions* class to generate binary fingerprint files
  - *OEFastFPDatabase* class and `OEFastFPDatabaseMemoryType` namespace to perform rapid in-memory or memory-mapped fingerprint searches using the popcount method
  - *OEFastFPDatabaseParams* utility class and `OEAreCompatibleDatabases` utility function
  - `OEIsFastFPDatabaseReady` and `OEGetPopCountMethod` functions and `OEPopCountMethod` namespace to check whether the popcount method is supported on the current hardware

> **Note**
>
> **GraphSim TK** currently only supports the popcount search method for fingerprints where the size is multiple of `256`. This means that the `OEFPType.MACCS166` fingerprint type is currently not supported. See the *User-defined Fingerprint* section.

> **Note**
>
> *OEFastFPDatabase* gives identical results to *OEFPDatabase*. However *OEFPDatabase* calculates similarity scores in single precision (float) while *OEFastFPDatabase* uses double precision. As a result small similarity score differences can be observed.

6.19.2 API Change

- The *OESimScore* class, which is utilized by both the *OEFPDatabase* class and the new *OEFastFPDatabase* class, stores a similarity value with a corresponding molecule index. This pair is now stored as "double" and "size_t" rather than "float" and "unsigned int". This **non-breaking** API change allows **GraphSim TK** to support molecular files that contain more than 2^{32} entries and supports double precision similarity scores calculated using the new *OEFastFPDatabase* class.

6.19.3 C++-specific changes

- The above-mentioned *OESimScore* class API change means that the following warning messages could be encountered when compiling old code with the new **GraphSim TK**:
  - `implicit conversion loses floating point precision: 'double' to 'float'`

```
for (OEIter<OESimScore> si = fpdb.GetScores(mol); si; ++si)
  float score = si->GetScore(); // GetScore() now returns double
```

  - `implicit conversion loses integer precision : 'size_t' to 'unsigned int'`

```
for (OEIter<OESimScore> si = fpdb.GetScores(mol); si; ++si)
  unsigned int molidx = si->GetIdx(); // GetIdx() now returns size_t
```

6.19.4 Java-specific changes

- The above-mentioned *OESimScore* class API change means that a similarity value with a corresponding molecule index is now expected to be "double" and "long", respectively.

6.19.5 Documentation changes

- A new section, *Building and searching fingerprint database*, has been added.
- New *Generating fingerprint file for fast fingerprint search* and *Searching fast fingerprint database* sections have been added with examples showing generation and searching of fingerprints using the *OEFastFPDatabase* class.

6.20 GraphSim TK 2.2.6

6.20.1 New features

- Three new functions, *OEGetCircularFPType*, *OEGetPathFPType*, and *OEGetTreeFPType*, have been added to enable generating fingerprint types with user-defined parameters.
- A new constructor, *OEFPTypeParams*, has been added that takes a fingerprint type (*OEFPTypeBase*).
- Four functions that automatically generate standard interfaces for **GraphSim TK** applications have been added:
  - *OEConfigureFingerPrint*
  - *OESetupFingerPrint*
  - *OEConfigureFPDatabaseOptions*
  - *OESetupFPDatabaseOptions*
- A new constructor, *OEFPDatabaseOption*, has been added that takes a similarity method from the *OESimMeasure* namespace and a limit on the number of similarity scores returned from a query.

6.21 GraphSim TK 2.2.5

- Minor internal improvements have been made.

6.22 GraphSim TK 2.2.4

6.22.1 New features

- *OEFPDatabaseOptions* class has been added to allow simultaneous fingerprint database searches.
- *OEFPDatabase.GetScores* and *OEFPDatabase.GetSortedScores* methods have been overloaded. The new versions take an *OEFPDatabaseOptions* search parameter.
- *OEFPDatabase.SetDescendingOrder* and *OEFPDatabase.GetDescendingOrder* methods have been added to allow setting the order in which the fingerprint scores are returned by the *OEFPDatabase.GetSortedScores* method to be set.

6.22.2 Minor bug fixes

- *OEFPDatabase.GetScores* and *OEFPDatabase.GetSortedScores* now throw warnings instead of error messages in the case of fingerprint type mismatches before returning an empty iterator.

6.23 GraphSim TK 2.2.3

- Minor internal improvements have been made.

6.23.1 Documentation changes

- The documentation for *OETversky* function has been restored.

6.24 GraphSim TK 2.2.2

- Minor internal improvements have been made.

6.25 GraphSim TK 2.2.1

- Minor internal improvements have been made.

> **Note**
>
> A mistake was made in the 2015.Feb release such that the *OEGraphSimGetRelease* function returns `2.1.2` rather than `2.2.0`.

6.26 GraphSim TK 2.2.0

6.26.1 Major bug fixes

- *OEFPType.MACCS166* fingerprints bit "91" and bit "92" have been fixed to behave properly. The previous behavior was the following:
  - Bit 91 could have been turned on by either `[#8]~[#6](~[#7])~[#6]` or `[#6H3]~[!#6;!#1]`.
  - Bit 92 was never turned on.

  The behavior has been fixed to do the following:
  - Bit 91 is now only turned on by `[#8]~[#6](~[#7])~[#6]`.
  - Bit 92 is now turned on by `[#6H3]~[!#6;!#1]`.

  > **Warning**
  >
  > This change was required to revise the version of the *OEFPType.MACCS166* fingerprints from `2.0.0` to `2.2.0`.
  >
  > For more information about version numbers, see the *Fingerprint version number* section.

- The customizable fingerprints (*OEFPType.Path*, *OEFPType.Tree*, and *OEFPType.Circular*) can now be generated without encoding any atom or bond properties by using the *OEFPAtomType.None* or *OEFPBondType.*

None constants, respectively. However, a warning will be thrown if using *OEFPAtomType.None* and *OEFPBondType.None* constants together.

6.26.2 Documentation fixes

- Code snippets have been added to the following functions that print the string representation of generated fingerprints:
  - *OEMakeCircularFP* function
  - *OEMakePathFP* function
  - *OEMakeTreeFP* function

> **Hint**
>
> The string representation reveals information about the fingerprinting method and the parameters being used to generate the fingerprint.

6.27 GraphSim TK 2.1.1

6.27.1 Documentation fixes

- Explaining the hexadecimal string representation of fingerprints generated by the `OEBitVector.ToHexString` method.

6.28 GraphSim TK 2.1.0

6.28.1 Minor bug fixes

- Removed unbounded stack allocations.

6.28.2 Documentation fixes

- The *MemSimSearch* documentation example demonstrates how to use a OEMolDatabase to minimize memory usage. See the *Sorted Search section* for a description of how this is done.
- A C# example for **GraphSim TK** added.

6.29 GraphSim TK 2.0.7

6.29.1 Documentation changes

- Added a section about *Visualizing Molecule Similarity*.
- Added note to the *OEFPAtomType.Aromaticity* constant.
- Added new images to the:
  - *OEFPAtomType* namespace to visualize the effect of atom typings.
  - *OEFPBondType* namespace to visualize the effect of bond typings.

6.30 GraphSim TK 2.0.6

- Minor internal improvements.

6.30.1 Documentation changes

Adding new images to the:

- *Fingerprint Generation* chapter to illustrate the enumeration of fragments of various fingerprints
- *Similarity Measures* chapter to show the basic bit count terms of similarity calculation
- *User-defined Fingerprint* chapter and `OEFPAtomType` namespace to visualize the effect of various atom and bond typings

6.31 GraphSim TK 2.0.5

6.31.1 Major bug fixes

- The performance of the `OEGetFPOverlap` function improved on average for drug-like structures:
  - 15% for `OEFPType.Path`
  - 30% for `OEFPType.Tree`
  - 10% for `OEFPType.Circular`

  The improvement is significantly more for very complex molecules such as buckyball structures.

6.32 GraphSim TK 2.0.4

- Minor internal improvements.

6.33 GraphSim TK 2.0.3

- Internal build system improvements.

6.34 GraphSim TK 2.0.2

- Internal build system improvements.

6.35 GraphSim TK 2.0.1

6.35.1 Minor bug fixes

- If a molecule being fingerprinted has only hydrogen atom(s), for example molecule `[H]`, then an empty fingerprint is generated, that is, all bits of the fingerprint are set to zero. All hydrogens of the molecule (polar, stereo, isotope, charge) are suppressed before generating its fingerprint.
- If the shell of the circular fingerprint can not be extended, for example, all atoms of the molecule are already being considered, then the search is terminated. This modification does not effect the generated circular fingerprints, it only makes the generation process faster.
- Some of the algorithms generating fingerprints can return the same fragment of a molecule more than once. For example the path fingerprint enumerates both `OC` and `CO`. These duplicates are now filtered out when calling the

OEGetFPOverlap function. Two overlaps are considered equivalent only if they store the same target and pattern atoms and bonds, not considering the correspondence between the pattern and target atoms and bonds.

6.36 GraphSim TK 2.0.0

6.36.1 New features

- Adding two new fingerprint types:
  - The *OEFPType.Circular* fingerprint is generated by exhaustively enumerating all circular fragments grown radially from each heavy atom of the input molecule up to the given radius. (See more details in *Circular Fingerprint* section and *OEMakeCircularFP* function)
  - The *OEFPType.Tree* fingerprint is generated by exhaustively enumerating all unique trees of a molecular graph. (See more details in *Tree Fingerprint* section and *OEMakeTreeFP* function)
- Adding the *OEGetFPCoverage* function that allows access to the patterns that are encoded into a fingerprint. (See example in *Fingerprint Coverage* chapter)
- Adding the *OEGetFPOverlap* function that returns the common patterns of two molecules based on their fingerprint. (See example in *Fingerprint Overlap* chapter)
- Adding version numbers for fingerprints that prevents the comparison of fingerprints that are generated by different versions of the algorithm. (See more details in *Fingerprint version number* section and *OEGetFingerPrintVersion* and *OEGetFingerPrintVersionString* functions)
- Adding three more atom typing options that can be used when generating path, circular or tree fingerprints:
  - *OEFPAtomType.HCount*
  - *OEFPAtomType.EqHBondAcceptor*
  - *OEFPAtomType.EqHBondDonor*
- Adding the *OEGetFPType* function that generates a fingerprint type from a valid string representation.
- Adding the *OEFPTypeParams* that allow to extract the parameters of the fingerprint types from their string representation:
- Adding 9 code examples in four different languages (C++, Python, Java, C#) that demonstrate the usage of fingerprints and functionalities.

6.36.2 Documentation changes

- Adding figures to the *Fingerprint* chapter that demonstrate how the *OEFPType.Circular*, *OEFPType.Path* and *OEFPType.Tree* fingerprints are generated.
- Adding a new chapter *Fingerprint Types*, that gives more details about fingerprint types.
- Adding two new examples to the *Storage and Retrieval* chapter that demonstrate how to store and retrieve fingerprints in SDF files.
- Rewriting part of the *User-defined Fingerprint* chapter, since the two new fingerprint types can also be customized.
- Adding two new chapters:
  - *Fingerprint Coverage* chapter
  - *Fingerprint Overlap* chapter

6.37 GraphSim TK 1.0.1

6.37.1 Documentation changes

- Adding an example to the *Storage and Retrieval* section that demonstrate how to generate a bitstring from an *OEFingerPrint* object.

6.38 GraphSim TK 1.0.0

- Provides the following features:
  - Generate three basic fingerprint types (MACCS key, LINGO and path-based)
  - Parameterize the path-based fingerprint generation
    - allowing various atom/bond typing
    - specifying the minimum and maximum length of the enumerated paths
    - setting the size of fingerprint
  - Store and retrieve fingerprints as generic data
  - Save/import fingerprints to/from OEB binary file format
- Implementation of six common similarity indices (Cosine, Dice, Euclidean, Manhattan, Tanimoto, Tversky)
- Implementation of fingerprint database that provides the following methods for rapid fingerprint-based similarity search:
  - apply user-defined similarity measures
  - set similarity cutoff value
  - access similarity in sorted order

CHAPTER

SEVEN

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

EIGHT

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

NINE

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.

9.1 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.

9.2 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.

9.3 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.

9.4 Publications for Bibliographies

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

9.4.1 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

9.4.2 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

9.4.3 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

9.4.4 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

9.4.5 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

9.5 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

TEN

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-strean	https://github.com/conda/conda-package-strean 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

10.1 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.

10.1.1 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

10.1.2 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

AddFP
    OEGraphSim::OEFPDatabase, 74
AddSample
    OEGraphSim::OEFPHistogram, 88
    OEGraphSim::OEFPVariogram, 97

C

ClearCutoff
    OEGraphSim::OEFPDatabase, 74
    OEGraphSim::OEFPDatabaseOptions, 84
Constructor
    OEGraphSim::OESimSearchResult, 99
Constructors
    OEGraphSim::OECreateFastFPDatabaseOptions, 59
    OEGraphSim::OEFastFPDatabase, 63
    OEGraphSim::OEFastFPDatabaseParams, 61
    OEGraphSim::OEFingerPrint, 72
    OEGraphSim::OEFPDatabase, 73
    OEGraphSim::OEFPDatabaseOptions, 83
    OEGraphSim::OEFPHistogram, 88
    OEGraphSim::OEFPPattern, 91
    OEGraphSim::OEFPTypeBase, 92
    OEGraphSim::OEFPTypeParams, 94
    OEGraphSim::OEFPVariogram, 96
    OEGraphSim::OESimFuncBase, 97
    OEGraphSim::OESimScore, 98
    OEGraphSim::OESimScorePair, 99
    OEGraphSim::OETanimotoSim, 101
    OEGraphSim::OETverskySim, 102
Cosine
    similarity measure, 15
CreateCopy
    OEGraphSim::OESimFuncBase, 97
    OEGraphSim::OETanimotoSim, 101
    OEGraphSim::OETverskySim, 102

D

Dice
    similarity measure, 15

E

Euclidean
    similarity measure, 16
Example Code
    makefastfp.cs, 50
    searchfastfp.cs, 54
    searchfp.cs, 46

G

GetAllScores
    OEGraphSim::OEFastFPDatabase, 64
GetAlpha
    OEGraphSim::OETverskySim, 102
GetAtomTypes
    OEGraphSim::OEFPTypeParams, 94
GetBeta
    OEGraphSim::OETverskySim, 102
GetBinBoundaries
    OEGraphSim::OEFPHistogram, 89
GetBinCenters
    OEGraphSim::OEFPHistogram, 89
GetBinIdx
    OEGraphSim::OEFPHistogram, 89
GetBinWidth
    OEGraphSim::OEFPHistogram, 89
GetBit
    OEGraphSim::OEFPPattern, 91
GetBondTypes
    OEGraphSim::OEFPTypeParams, 94
GetCounts
    OEGraphSim::OEFPHistogram, 89
GetCutoff
    OEGraphSim::OEFPDatabase, 74
    OEGraphSim::OEFPDatabaseOptions, 84
GetDensity
    OEGraphSim::OEFPHistogram, 90
GetDescendingOrder
    OEGraphSim::OEFPDatabase, 75
    OEGraphSim::OEFPDatabaseOptions, 84
GetFingerPrint
    OEGraphSim::OEFastFPDatabase, 64
GetFingerPrints

OEGraphSim::OEFPDatabase, 75
GetFPType
OEGraphSim::OECreateFastFPDatabaseOptions, 60
OEGraphSim::OEFPTypeBase, 92
OEGraphSim::OEFPTypeParams, 95
GetFPTypeBase
OEGraphSim::OEFastFPDatabase, 64
OEGraphSim::OEFastFPDatabaseParams, 61
OEGraphSim::OEFingerPrint, 72
OEGraphSim::OEFPDatabase, 75
GetFPTypeString
OEGraphSim::OEFPTypeBase, 92
GetFPVersion
OEGraphSim::OEFPTypeBase, 93
GetFPVersionString
OEGraphSim::OEFPTypeBase, 93
GetHistogram
OEGraphSim::OEFastFPDatabase, 65
OEGraphSim::OESimSearchResult, 99
GetIdx
OEGraphSim::OESimScore, 98
GetIdxA
OEGraphSim::OESimScorePair, 99
GetIdxB
OEGraphSim::OESimScorePair, 99
GetLimit
OEGraphSim::OEFPDatabaseOptions, 84
GetMax
OEGraphSim::OEFPHistogram, 90
GetMaxDistance
OEGraphSim::OEFPTypeParams, 95
GetMemoryType
OEGraphSim::OEFastFPDatabase, 65
GetMemoryTypeString
OEGraphSim::OEFastFPDatabase, 66
GetMin
OEGraphSim::OEFPHistogram, 90
GetMinDistance
OEGraphSim::OEFPTypeParams, 95
GetMolDatabaseFilename
OEGraphSim::OEFastFPDatabaseParams, 61
GetMoleculeIndex
OEGraphSim::OEFastFPDatabase, 66
GetNumBits
OEGraphSim::OEFPTypeParams, 95
GetNumProcessors
OEGraphSim::OECreateFastFPDatabaseOptions, 60
OEGraphSim::OEFPDatabaseOptions, 85
GetRawScores
OEGraphSim::OEFastFPDatabase, 66
GetScore
OEGraphSim::OESimScore, 98
OEGraphSim::OESimScorePair, 99
GetScores
OEGraphSim::OEFastFPDatabase, 67
OEGraphSim::OEFPDatabase, 75
GetSearchStatus
OEGraphSim::OESimSearchResult, 100
GetSimFunc
OEGraphSim::OEFPDatabaseOptions, 85
GetSimTypeString
OEGraphSim::OESimFuncBase, 98
OEGraphSim::OETanimotoSim, 101
OEGraphSim::OETverskySim, 102
GetSmarts
OEGraphSim::OEFPPattern, 92
GetSortedScores
OEGraphSim::OEFastFPDatabase, 67
OEGraphSim::OEFPDatabase, 78
OEGraphSim::OESimSearchResult, 100
GetSparseMatrix
OEGraphSim::OEFastFPDatabase, 68
GetTverskyAlpha
OEGraphSim::OEFPDatabaseOptions, 85
GetTverskyBeta
OEGraphSim::OEFPDatabaseOptions, 85
GetUnhashed
OEGraphSim::OEFPPattern, 92
GetVariogram
OEGraphSim::OEFastFPDatabase, 69
OEGraphSim::OEFPVariogram, 97
GetVersion
OEGraphSim::OEFPTypeParams, 95

H

HasCutoff
OEGraphSim::OEFPDatabase, 81
OEGraphSim::OEFPDatabaseOptions, 86

I

IsValid
OEGraphSim::OEFastFPDatabase, 70
OEGraphSim::OEFastFPDatabaseParams, 61
OEGraphSim::OEFPTypeParams, 96
OEGraphSim::OESimSearchResult, 100

M

makefastfp.cs
Example Code, 50
Manhattan
similarity measure, 16
Mean
OEGraphSim::OEFPHistogram, 90

N

NumBins

OEGraphSim::OEFPHistogram, 90
NumFingerPrints
OEGraphSim::OEFastFPDatabase, 70
OEGraphSim::OEFastFPDatabaseParams, 61
OEGraphSim::OEFPDatabase, 81
NumSamples
OEGraphSim::OEFPHistogram, 90
NumSearched
OEGraphSim::OESimSearchResult, 100
NumTargets
OEGraphSim::OESimSearchResult, 100

O

OEDataFlow::OEInitFPFieldHandler, 142
OEDataFlow::OEInitFPVecFieldHandler, 142
OEDataFlow::OESetFingerPrintValue, 150
OEGraphSim::OEAreCompatibleDatabases, 129
OEGraphSim::OEConfigureFingerPrint, 130
OEGraphSim::OEConfigureFPDatabaseMemoryType, 129
OEGraphSim::OEConfigureFPDatabaseOptions, 130
OEGraphSim::OECosine, 131
OEGraphSim::OECreateFastFPDatabaseFile, 131
OEGraphSim::OECreateFastFPDatabaseOptions, 59
Constructors, 59
GetFPType, 60
GetNumProcessors, 60
SetNumProcessors, 60
SetTracer, 60
OEGraphSim::OEDice, 132
OEGraphSim::OEEuclid, 133
OEGraphSim::OEFastFPDatabase, 61
Constructors, 63
GetAllScores, 64
GetFingerPrint, 64
GetFPTypeBase, 64
GetHistogram, 65
GetMemoryType, 65
GetMemoryTypeString, 66
GetMoleculeIndex, 66
GetRawScores, 66
GetScores, 67
GetSortedScores, 67
GetSparseMatrix, 68
GetVariogram, 69
IsValid, 70
NumFingerPrints, 70
SortedSearch, 70
OEGraphSim::OEFastFPDatabaseMemoryType, 102
OEGraphSim::OEFastFPDatabaseMemoryType::CUDA, 103
OEGraphSim::OEFastFPDatabaseMemoryType::Default, 103
OEGraphSim::OEFastFPDatabaseMemoryType::InMemory, 103
OEGraphSim::OEFastFPDatabaseMemoryType::MemoryMapped, 103
OEGraphSim::OEFastFPDatabaseParams, 60
Constructors, 61
GetFPTypeBase, 61
GetMolDatabaseFilename, 61
IsValid, 61
NumFingerPrints, 61
OEGraphSim::OEFingerPrint, 72
Constructors, 72
GetFPTypeBase, 72
operator bool, 72
SetFPTypeBase, 73
OEGraphSim::OEFPAtomType, 104
OEGraphSim::OEFPAtomType::Aromaticity, 106
OEGraphSim::OEFPAtomType::AtomicNumber, 105
OEGraphSim::OEFPAtomType::Chiral, 106
OEGraphSim::OEFPAtomType::DefaultAtom, 113
OEGraphSim::OEFPAtomType::DefaultCircularAtom, 113
OEGraphSim::OEFPAtomType::DefaultCircularVSAtom, 114
OEGraphSim::OEFPAtomType::DefaultCircularVSAtom1, 114
OEGraphSim::OEFPAtomType::DefaultCircularVSAtom2, 114
OEGraphSim::OEFPAtomType::DefaultPathAtom, 114
OEGraphSim::OEFPAtomType::DefaultPathVSAtom, 115
OEGraphSim::OEFPAtomType::DefaultTreeAtom, 115
OEGraphSim::OEFPAtomType::DefaultTreeVSAtom, 116
OEGraphSim::OEFPAtomType::EqAromatic, 110
OEGraphSim::OEFPAtomType::EqHalogen, 110
OEGraphSim::OEFPAtomType::EqHBondAcceptor, 111
OEGraphSim::OEFPAtomType::EqHBondDonor, 112
OEGraphSim::OEFPAtomType::FormalCharge, 107
OEGraphSim::OEFPAtomType::HCount, 108
OEGraphSim::OEFPAtomType::HvyDegree, 108
OEGraphSim::OEFPAtomType::Hybridization, 109
OEGraphSim::OEFPAtomType::InRing, 109
OEGraphSim::OEFPAtomType::None, 117
OEGraphSim::OEFPBondType, 117
OEGraphSim::OEFPBondType::BondOrder, 118
OEGraphSim::OEFPBondType::Chiral, 118
OEGraphSim::OEFPBondType::DefaultBond, 119
OEGraphSim::OEFPBondType::DefaultCircularBond, 120

OEGraphSim::OEFPBondType::DefaultCircularVSBond, 120
OEGraphSim::OEFPBondType::DefaultCircularVSBond2, 120
OEGraphSim::OEFPBondType::DefaultCircularVSBond3, 121
OEGraphSim::OEFPBondType::DefaultPathBond, 121
OEGraphSim::OEFPBondType::DefaultPathVSBond, 121
OEGraphSim::OEFPBondType::DefaultTreeBond, 122
OEGraphSim::OEFPBondType::DefaultTreeVSBond, 122
OEGraphSim::OEFPBondType::InRing, 119
OEGraphSim::OEFPBondType::None, 122
OEGraphSim::OEFPDatabase, 73
  AddFP, 74
  ClearCutoff, 74
  Constructors, 73
  GetCutoff, 74
  GetDescendingOrder, 75
  GetFingerPrints, 75
  GetFPTypeBase, 75
  GetScores, 75
  GetSortedScores, 78
  HasCutoff, 81
  NumFingerPrints, 81
  SetCutoff, 81
  SetDescendingOrder, 81
  SetSimFunc, 82
OEGraphSim::OEFPDatabaseOptions, 82
  ClearCutoff, 84
  Constructors, 83
  GetCutoff, 84
  GetDescendingOrder, 84
  GetLimit, 84
  GetNumProcessors, 85
  GetSimFunc, 85
  GetTverskyAlpha, 85
  GetTverskyBeta, 85
  HasCutoff, 86
  SetCutoff, 86
  SetDescendingOrder, 86
  SetLimit, 87
  SetNumProcessors, 87
  SetSimFunc, 87
  SetTverskyCoeffs, 87
OEGraphSim::OEFPDatabaseOptionsSetup, 123
OEGraphSim::OEFPDatabaseOptionsSetup::All, 123
OEGraphSim::OEFPDatabaseOptionsSetup::Cutoff, 123
OEGraphSim::OEFPDatabaseOptionsSetup::DescendingOrder, 123
OEGraphSim::OEFPDatabaseOptionsSetup::Limit, 124
OEGraphSim::OEFPDatabaseOptionsSetup::SimFunc, 124
OEGraphSim::OEFPHistogram, 88
  AddSample, 88
  Constructors, 88
  GetBinBoundaries, 89
  GetBinCenters, 89
  GetBinIdx, 89
  GetBinWidth, 89
  GetCounts, 89
  GetDensity, 90
  GetMax, 90
  GetMin, 90
  Mean, 90
  NumBins, 90
  NumSamples, 90
  Std, 91
OEGraphSim::OEFPPattern, 91
  Constructors, 91
  GetBit, 91
  GetSmarts, 92
  GetUnhashed, 92
OEGraphSim::OEFPToNumpyArray, 152
OEGraphSim::OEFPType, 125
OEGraphSim::OEFPType::Circular, 125
OEGraphSim::OEFPType::Lingo, 125
OEGraphSim::OEFPType::MACCS166, 125
OEGraphSim::OEFPType::Path, 126
OEGraphSim::OEFPType::Tree, 126
OEGraphSim::OEFPTypeBase, 92
  Constructors, 92
  GetFPType, 92
  GetFPTypeString, 92
  GetFPVersion, 93
  GetFPVersionString, 93
OEGraphSim::OEFPTypeParams, 93
  Constructors, 94
  GetAtomTypes, 94
  GetBondTypes, 94
  GetFPType, 95
  GetMaxDistance, 95
  GetMinDistance, 95
  GetNumBits, 95
  GetVersion, 95
  IsValid, 96
OEGraphSim::OEFPVariogram, 96
  AddSample, 97
  Constructors, 96
  GetVariogram, 97
OEGraphSim::OEGetBitCounts, 133

OEGraphSim::OEGetCircularFPType, 133
OEGraphSim::OEGetFingerPrintVersion, 138
OEGraphSim::OEGetFingerPrintVersionString, 138
OEGraphSim::OEGetFPAtomType, 134
OEGraphSim::OEGetFPBondType, 135
OEGraphSim::OEGetFPCoverage, 135
OEGraphSim::OEGetFPDatabaseMemoryType, 136
OEGraphSim::OEGetFPOverlap, 136
OEGraphSim::OEGetFPPatterns, 137
OEGraphSim::OEGetFPType, 137
OEGraphSim::OEGetPathFPType, 139
OEGraphSim::OEGetPopCountMethod, 139
OEGraphSim::OEGetSimilarityMeasureName, 140
OEGraphSim::OEGetTreeFPType, 140
OEGraphSim::OEGetVSFPType, 140
OEGraphSim::OEGraphSimGetArch, 141
OEGraphSim::OEGraphSimGetLicensee, 141
OEGraphSim::OEGraphSimGetPlatform, 141
OEGraphSim::OEGraphSimGetRelease, 141
OEGraphSim::OEGraphSimGetSite, 141
OEGraphSim::OEGraphSimGetVersion, 141
OEGraphSim::OEGraphsimIsGPUReady, 142
OEGraphSim::OEGraphSimIsLicensed, 141
OEGraphSim::OEIsFastFPDatabaseReady, 143
OEGraphSim::OEIsFPType, 143
OEGraphSim::OEIsSameFPType, 143
OEGraphSim::OEIsValidFPTypeString, 144
OEGraphSim::OEMakeCircularFP, 144
OEGraphSim::OEMakeFP, 145
OEGraphSim::OEMakeLingoFP, 146
OEGraphSim::OEMakeMACCS166FP, 147
OEGraphSim::OEMakePathFP, 147
OEGraphSim::OEMakeTreeFP, 148
OEGraphSim::OEMakeVSFP, 150
OEGraphSim::OEManhattan, 150
OEGraphSim::OEPopCountMethod, 126
OEGraphSim::OEPopCountMethod::BitCount, 126
OEGraphSim::OEPopCountMethod::NoSupport, 126
OEGraphSim::OEPopCountMethod::PopCount, 127
OEGraphSim::OESetupFingerPrint, 151
OEGraphSim::OESetupFPDatabaseOptions, 151
OEGraphSim::OESimFuncBase, 97
    Constructors, 97
    CreateCopy, 97
    GetSimTypeString, 98
    operator(), 97
OEGraphSim::OESimMeasure, 127
OEGraphSim::OESimMeasure::Cosine, 127
OEGraphSim::OESimMeasure::Dice, 127
OEGraphSim::OESimMeasure::Euclid, 127
OEGraphSim::OESimMeasure::Manhattan, 128
OEGraphSim::OESimMeasure::Tanimoto, 128
OEGraphSim::OESimMeasure::Tversky, 128
OEGraphSim::OESimScore, 98
    Constructors, 98
    GetIdx, 98
    GetScore, 98
OEGraphSim::OESimScorePair, 98
    Constructors, 99
    GetIdxA, 99
    GetIdxB, 99
    GetScore, 99
OEGraphSim::OESimSearchResult, 99
    Constructor, 99
    GetHistogram, 99
    GetSearchStatus, 100
    GetSortedScores, 100
    IsValid, 100
    NumSearched, 100
    NumTargets, 100
OEGraphSim::OESimSearchStatus, 128
OEGraphSim::OESimSearchStatus::Finished, 128
OEGraphSim::OESimSearchStatus::InProgress, 129
OEGraphSim::OESimSearchStatus::Uninitialized, 129
OEGraphSim::OESimSearchStatusToName, 151
OEGraphSim::OETanimoto, 152
OEGraphSim::OETanimotoSim, 100
    Constructors, 101
    CreateCopy, 101
    GetSimTypeString, 101
    operator(), 101
OEGraphSim::OETversky, 152
OEGraphSim::OETverskySim, 101
    Constructors, 102
    CreateCopy, 102
    GetAlpha, 102
    GetBeta, 102
    GetSimTypeString, 102
    operator(), 102
operator bool
    OEGraphSim::OEFingerPrint, 72
operator()
    OEGraphSim::OESimFuncBase, 97
    OEGraphSim::OETanimotoSim, 101
    OEGraphSim::OETverskySim, 102

S

searchfastfp.cs
    Example Code, 54
searchfp.cs
    Example Code, 46
SetCutoff
    OEGraphSim::OEFPDatabase, 81
    OEGraphSim::OEFPDatabaseOptions, 86
SetDescendingOrder

OEGraphSim::OEFPDatabase, 81
OEGraphSim::OEFPDatabaseOptions, 86
SetFPTypeBase
OEGraphSim::OEFingerPrint, 73
SetLimit
OEGraphSim::OEFPDatabaseOptions, 87
SetNumProcessors
OEGraphSim::OECreateFastFPDatabaseOptions, 60
OEGraphSim::OEFPDatabaseOptions, 87
SetSimFunc
OEGraphSim::OEFPDatabase, 82
OEGraphSim::OEFPDatabaseOptions, 87
SetTracer
OEGraphSim::OECreateFastFPDatabaseOptions, 60
SetTverskyCoeffs
OEGraphSim::OEFPDatabaseOptions, 87
similarity measure
Cosine, 15
Dice, 15
Euclidean, 16
Manhattan, 16
Tanimoto, 16
Tversky, 17
SortedSearch
OEGraphSim::OEFastFPDatabase, 70
Std
OEGraphSim::OEFPHistogram, 91

T

Tanimoto
similarity measure, 16
Tversky
similarity measure, 17