

OpenEye
SCIENTIFIC

Spruce TK – C#

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Introduction **3**

2 Theory **5**
- 2.1 Brief Definitions . . . 5
- 2.2 Asymmetric Unit . . . 6
- 2.3 Biological Unit . . . 6
- 2.4 Biological Design Unit (BioDU) . . . 6
- 2.5 Design Unit . . . 6
- 2.6 Spruce TK Preparation Workflow . . . 7
- 2.7 Proton Placement and Optimization . . . 8
- 2.8 Protein Side Chain Modeling . . . 8
- 2.9 Protein Loop Modeling . . . 8
- 2.10 Protein Superposition . . . 9
- 2.11 How to Correctly Read a Macromolecular File Format (mmCIF/PDB) . . . 10

3 Iridium **11**
- 3.1 Categorization . . . 11

4 Advanced Macromolecular Preparation Workflow **15**
- 4.1 Inputs . . . 15
- 4.2 Filtering and Standardization . . . 16
- 4.3 Generating the Biological Unit . . . 17
- 4.4 Generating Packing Residues . . . 19
- 4.5 Enumerating Alternate Locations . . . 19
- 4.6 Componentizing the System . . . 20
- 4.7 Building Missing Pieces . . . 22
  - 4.7.1 Building Missing Side Chains . . . 22
  - 4.7.2 Building Missing Loops and Tails . . . 24
  - 4.7.3 Building Capping Groups . . . 25
- 4.8 Optimizing the Hydrogen Bond Network . . . 26
- 4.9 Assignment on Radii and Partial Charges . . . 26
- 4.10 Prepared Biological Units (BioDUs) . . . 27
- 4.11 Enumerating Sites / Pockets . . . 27
- 4.12 Assigning an Iridium Category . . . 28
- 4.13 Generating a Docking Receptor . . . 29
- 4.14 Superposition to a Common Frame of Reference . . . 29
- 4.15 Design Unit Validation . . . 29

5 Examples **31**
- 5.1 Spruce TK Examples . . . 31

5.1.1 Creating OEDesignUnits from a PDB file . . . 31
5.1.2 Creating apo OEDesignUnits from a PDB file . . . 33
5.1.3 Creating mmcif file from an input DU . . . 33
5.1.4 Calculating superposition . . . 34
5.1.5 Extracting Biounits from PDB header remarks . . . 35
5.1.6 Extracting Biounits by aligning to a reference protein . . . 35
5.1.7 Finding pockets and their properties (like, Surface Area) from a PDB file . . . 36
5.1.8 Write a metadata JSON file from a FASTA file . . . 36
5.1.9 Validate Structure of the Design Unit . . . 36
5.1.10 Add Receptor Grid to Design Unit . . . 36
5.1.11 Creating OEDesignUnits Using Provided Metadata . . . 37

6 OESpruce API **39**
6.1 OESpruce Classes . . . 39
6.1.1 OEBioUnitExtractionOptions . . . 39
6.1.2 SetMaxAtoms . . . 40
6.1.3 SetMaxParts . . . 40
6.1.4 SetMinScore . . . 40
6.1.5 SetPreferAuthorRecord . . . 40
6.1.6 SetSuperpose . . . 40
6.1.7 OECapBuilderOptions . . . 41
6.1.8 OEDesignUnitBuildOptions . . . 42
6.1.9 SetBuildLoops . . . 43
6.1.10 SetBuildSidechains . . . 43
6.1.11 SetCapCTermini . . . 43
6.1.12 SetCapNTermini . . . 44
6.1.13 SetSidechainBuilderOptions . . . 44
6.1.14 SetCapBuilderOptions . . . 44
6.1.15 SetLoopBuilderOptions . . . 44
6.1.16 OEDesignUnitEnumerateSitesOptions . . . 44
6.1.17 SetAddInteractionHints . . . 45
6.1.18 SetAddStyle . . . 45
6.1.19 SetCollapseNonSiteAlts . . . 46
6.1.20 SetDuplicateRemoval . . . 46
6.1.21 SetEnumerateCofactorSites . . . 46
6.1.22 SetRestrictToRefSite . . . 46
6.1.23 SetSiteSize . . . 46
6.1.24 OEDesignUnitMutationOptions . . . 46
6.1.25 OEDesignUnitPrepOptions . . . 47
6.1.26 SetAssignPartialChargesAndRadii . . . 49
6.1.27 SetBuildOptions . . . 49
6.1.28 SetChargeEngine . . . 49
6.1.29 SetEnumerateSitesOptions . . . 49
6.1.30 SetProtonate . . . 49
6.1.31 SetStrictProtonationMode . . . 49
6.1.32 SetProtonateOptions . . . 49
6.1.33 OEDesignUnitSplitOptions . . . 50
6.1.34 OEHeterogenMetadata . . . 53
6.1.35 OEIsModeledAtom . . . 55
6.1.36 OELoopBuilderOptions . . . 57
6.1.37 OEMakeDesignUnitOptions . . . 63
6.1.38 OEPocket . . . 65
6.1.39 OEPocketOptions . . . 67
6.1.40 SetMinSurfaceArea . . . 67

6.1.41 SetMaxSurfaceArea . . . 68
6.1.42 SetBurialFactor . . . 68
6.1.43 OEProtonateDesignUnitOptions . . . 68
6.1.44 SetGenerateTautomers . . . 69
6.1.45 SetHetGroupNbrDist . . . 69
6.1.46 SetOptimizeExpProtons . . . 69
6.1.47 SetPlaceHydrogensOptions . . . 69
6.1.48 OESidechainBuilderOptions . . . 69
6.1.49 OESequenceMetadata . . . 72
6.1.50 OEStructureMetadata . . . 73
6.1.51 OESecondaryStructureSuperposition . . . 77
6.1.52 OESpruceFilter . . . 78
6.1.53 StandardizeAndFilter . . . 79
6.1.54 GetMessages . . . 80
6.1.55 GetAtoms . . . 80
6.1.56 OESpruceFilterOptions . . . 80
6.1.57 SetDeleteFloatingRes . . . 81
6.1.58 SetFixBackboneAtoms . . . 81
6.1.59 SetFixBondsToMetals . . . 81
6.1.60 SetFixChainIDs . . . 81
6.1.61 SetFixInvalidCovalentBonds . . . 82
6.1.62 SetFixNames . . . 82
6.1.63 SetFixResStates . . . 82
6.1.64 OEStructuralSuperposition . . . 82
6.1.65 OESuperpositionOptions . . . 84
6.1.66 OESuperpose . . . 87
6.1.67 OESuperposeOptions . . . 90
6.1.68 GetMethod . . . 90
6.1.69 SetMethod . . . 90
6.1.70 GetUseConstraints . . . 91
6.1.71 SetUseConstraints . . . 91
6.1.72 GetValidRMSD . . . 91
6.1.73 SetValidRMSD . . . 91
6.1.74 GetValidTanimoto . . . 92
6.1.75 SetValidTanimoto . . . 92
6.1.76 GetValidSeqScore . . . 92
6.1.77 SetValidSeqScore . . . 93
6.1.78 GetUseSeqScoreChainMatching . . . 93
6.1.79 SetUseSeqScoreChainMatching . . . 93
6.1.80 OESuperposeResults . . . 94
6.1.81 OEValidateDesignUnit . . . 97
6.2 OESpruce Constants . . . 98
6.2.1 OEAlternateLocationOption . . . 98
6.2.2 OEDesignUnitIssueCodes . . . 99
6.2.3 OEExperimentType . . . 100
6.2.4 OEHeterogenType . . . 101
6.2.5 OESuperpositionType . . . 102
6.2.6 OESpruceFilterIssueCodes . . . 103
6.2.7 OESuperposeMethod . . . 104
6.3 OESpruce Functions . . . 106
6.3.1 OEBuildLoops . . . 106
6.3.2 OEBuildSidechains . . . 108
6.3.3 OEBuildSingleLoop . . . 108
6.3.4 OECapCTermini . . . 110

6.3.5 OECapNTermini . . . 110
6.3.6 OECapTermini . . . 111
6.3.7 OEEnumerateSites . . . 112
6.3.8 OEExtractBioUnits . . . 113
6.3.9 OEFindPockets . . . 114
6.3.10 OEFixBackbone . . . 115
6.3.11 OEGetAlternateLocationOptionID . . . 115
6.3.12 OEGetAlternateLocationOptionName . . . 115
6.3.13 OEGetExperimentTypeID . . . 115
6.3.14 OEGetExperimentTypeName . . . 116
6.3.15 OEGetHeterogenTypeID . . . 116
6.3.16 OEGetHeterogenTypeName . . . 116
6.3.17 OEGetPartialResidues . . . 116
6.3.18 OEHeterogenMetadataFromJson . . . 117
6.3.19 OEHeterogenMetadataToJson . . . 118
6.3.20 OEIsRedundantDesignUnit . . . 118
6.3.21 OEMakeBioDesignUnits . . . 119
6.3.22 OEMakeDesignUnit . . . 120
6.3.23 OEMakeDesignUnits . . . 121
6.3.24 OEMakeDesignUnitFromPocket . . . 122
6.3.25 OEMutateResidue . . . 123
6.3.26 OEMutateResidues . . . 124
6.3.27 OEProtonateDesignUnit . . . 124
6.3.28 OESplitBioUnitToComponents . . . 125
6.3.29 OESpruceGetArch . . . 126
6.3.30 OESpruceGetLicensee . . . 126
6.3.31 OESpruceGetPlatform . . . 126
6.3.32 OESpruceGetRelease . . . 126
6.3.33 OESpruceGetSite . . . 126
6.3.34 OESpruceGetVersion . . . 126
6.3.35 OESpruceIsLicensed . . . 127
6.3.36 OESequenceMetadataFromJson . . . 127
6.3.37 OESequenceMetadataToJson . . . 128
6.3.38 OEStructureMetadataFromJson . . . 128
6.3.39 OEStructureMetadataToJson . . . 129
6.3.40 OEGetSuperposeMethodFromName . . . 130
6.3.41 OEGetSuperposeMethodName . . . 131

7 Release History **133**
7.1 Spruce TK 1.9.0 . . . 133
7.1.1 New Features . . . 133
7.1.2 Minor Bug Fixes . . . 133
7.2 Spruce TK 1.8.0 . . . 133
7.2.1 New Features . . . 133
7.2.2 Major Bug Fixes . . . 134
7.2.3 Minor Bug Fixes . . . 134
7.2.4 Documentation Changes . . . 134
7.3 Spruce TK 1.7.0 . . . 134
7.3.1 New Features . . . 134
7.3.2 Major Bug Fixes . . . 134
7.3.3 Minor Bug Fixes . . . 135
7.4 Spruce TK 1.6.1.1 . . . 135
7.4.1 Major bug fixes . . . 135
7.5 Spruce TK 1.6.1 . . . 135

7.5.1 New features . . . 135
7.5.2 Major bug fixes . . . 135
7.5.3 Minor bug fixes . . . 135
7.5.4 Documentation changes . . . 135
7.6 Spruce TK 1.6.0 . . . 136
7.6.1 New features . . . 136
7.6.2 Major bug fixes . . . 136
7.6.3 Minor bug fixes . . . 136
7.6.4 Documentation changes . . . 136
7.7 Spruce TK 1.5.3 . . . 136
7.7.1 New features . . . 136
7.7.2 Major bug fixes . . . 137
7.7.3 Minor bug fixes . . . 137
7.8 Spruce TK 1.5.2 . . . 137
7.8.1 New features . . . 137
7.8.2 Major bug fixes . . . 137
7.8.3 Minor bug fixes . . . 137
7.8.4 Documentation changes . . . 137
7.9 Spruce TK 1.5.1 . . . 137
7.9.1 New features . . . 137
7.9.2 Major bug fixes . . . 138
7.10 Spruce TK 1.5.0 . . . 138
7.10.1 New features . . . 138
7.10.2 Major bug fixes . . . 138
7.10.3 Minor bug fixes . . . 138
7.11 Spruce TK 1.4.0 . . . 138
7.11.1 New features . . . 138
7.11.2 Major bug fixes . . . 139
7.11.3 Minor bug fixes . . . 139
7.12 Spruce TK 1.3.0 . . . 139
7.12.1 New features . . . 139
7.12.2 Minor bug fixes . . . 140
7.12.3 Documentation changes . . . 140
7.13 Spruce TK 1.2.0 . . . 140
7.13.1 New features . . . 140
7.13.2 Major bug fixes . . . 140
7.13.3 Minor bug fixes . . . 141
7.14 Spruce TK 1.1.0 . . . 141
7.14.1 New features . . . 141
7.14.2 Major bug fixes . . . 142
7.14.3 Minor bug fixes . . . 142
7.14.4 Documentation changes . . . 142
7.15 Spruce TK 1.0.0 . . . 143
7.15.1 New features . . . 143
7.15.2 Major bug fixes . . . 143
7.15.3 Minor bug fixes . . . 143
7.15.4 Documentation changes . . . 143
7.16 Spruce TK 0.9.2 . . . 144
7.16.1 New features . . . 144
7.16.2 Major bug fixes . . . 144
7.16.3 Minor bug fixes . . . 144
7.17 Spruce TK 0.9.1 . . . 144
7.17.1 New features . . . 144
7.17.2 API changes . . . 145

7.17.3 Major bug fixes . . . 145
7.17.4 Minor bug fixes . . . 145
7.18 Spruce TK 0.9.0 . . . 145
7.18.1 New features . . . 145

8 Copyright and Trademarks **147**

9 Sample Code **149**

10 Citation **151**
10.1 Orion® . . . 151
10.2 Orion Floes . . . 151
10.3 Toolkits and Applications . . . 151
10.4 Publications for Bibliographies . . . 153
10.4.1 Orion . . . 153
10.4.2 AFITT and FLYNN Applications . . . 153
10.4.3 OEDocking Application and Toolkit . . . 153
10.4.4 OMEGA Application and Toolkit . . . 153
10.4.5 ROCS Application . . . 154
10.5 OpenEye MMDS Web Service . . . 154

11 Technology Licensing **155**
11.1 GCC . . . 227
11.1.1 GCC RUNTIME LIBRARY EXCEPTION . . . 228
11.1.2 GNU GENERAL PUBLIC LICENSE . . . 229

Index **243**

> **Attention**
>
> **Spruce TK** is currently only available in C++ and Python.

CHAPTER

ONE

INTRODUCTION

OESpruce TK provides functionalities to prepare biomolecules (that is, proteins and nucleic acids) for modeling tasks.

CHAPTER

TWO

THEORY

Spruce TK is used to process PDB or mmCIF files containing the macromolecular structures, resulting from X-ray crystallography, Nuclear Magnetic Resonance (NMR) spectroscopy, or electron microscopy (EM) experiments into molecules usable for molecular modeling. Due to the nature of the experiments and data, some processing is required before use in modeling, such as adding explicit hydrogens and building pieces that were unobserved in the experimental data, yet required to properly simulate the system.

Spruce TK provides a complete preparation workflow to prepare macrocomolecular structures for modeling. Various options are available to control for desired behaviors. Additional metadata can be provided, along with the structural data to improve perception of small molecule (heterogen) *bondage and valence* states, as well as protein sequence information (if it is not in the structure file already).

2.1 Brief Definitions

In order to fully interpret **Spruce TK**, it is important to first identify some terms. Very brief definitions are provided in this section, followed by more detailed explanations below.

1. *Asymmetric Unit (ASU)*: The contents of a PDB or mmCIF file from X-ray crystallography containing an asymmetric unit as the output of the experiment.
2. *Biological Unit (BU)*: Also know as a biological assembly, this is the biologically relevant unit for a protein to use in modeling. For example, the BU is a monomer in Trypsin, but in HIV Protease it is a homo-dimer.
3. *Bio Design Unit (BioDU)*: This is a prepared biological unit. This is the result of the **Spruce** preparation from the extracted BU. It is a fully prepared macromolecular system, but a ligand and associated binding site have not yet been identified. A Bio Design Unit (*BioDU*) is considered an intermediate work product of the final design unit.
4. *Design Unit (DU)*: As the name indicates, this is an object used for structure-based drug design. It is a specific instance of a BioDU, where a binding site has been identified. If a ligand is present in that binding site, it has been moved into the ligand component; similarly, the cofactor component has been filled in if any are present in the binding site. This data structure can also have an OEReceptor stored on it for the purposes of docking calculations. Most OpenEye tools use DUs rather than BioDUs, as they utilize the binding site and bound ligand information.
5. **Alternate locations (alt locs)**: X-ray experiments often contain results with atoms occupying more than one configuration (alt location). Crystallographers denote this with a measure of the amount of time an atom occupies a given set of coordinates. A well-resolved atom will have an occupancy of 1.0, meaning that 100% of the time it is in that location. Sometimes, however, the atom exists in two positions—alternate locations—and these appear in the input file with single letter designations and with occupancy < 1.0. Alternate locations are by default enumerated (but only primary can be selected). When these are enumerated, the generated design units will be checked to see if there are alternate locations in the binding site; if not, they are deduplicated at the end of the preparation workflow (this behavior can be turned off). This is to reduce the number of design units generated for a single input file.

Note

For more information about these, there are two worked examples: *the simple example* and a more *advanced example*.

2.2 Asymmetric Unit

The asymmetric unit (ASU) is the smallest piece of a crystal structure to which symmetry operations can be applied in order to generate a complete unit cell (the crystal repeating unit). It contains the unique part of a crystal structure [Zardecki-2022]. The ASU may be equivalent to the BU, but often requires manipulation to create a correct BU. **Spruce TK** can be used to expand the AU to the BU. The data structure is stored in an `OEMolBase`.

2.3 Biological Unit

In short, the biological unit (BU) is an object that contains the biologically relevant parts of an ASU that have not been split into various molecular components and are not yet prepped for modeling. It is the quaternary structure of a protein that is believed to be the main functional form of the molecule. It can be a single chain or a collection of multiple chains. The BU may include more than one functional form if it can change under different conditions. The data structure is stored in an `OEMolBase`.

For a more detailed explanation of BUs, refer to the Introduction to Biological Assemblies and the PDB Archive.

A BU can be constructed from a single macromolecular structure file, from the metadata, or by using a sequence alignment to an input reference structure. To extract a BU or set of BUs from an asymmetric unit, one should use the helper function in **Spruce TK** called *`OEExtractBioUnits`*. The following example shows how to extract BUs from the metadata.

Using the same function, one can also extract BUs from an asymmetric unit using an input reference protein. The following example shows how to use *`OEExtractBioUnits`* to extract BUs from a given reference.

2.4 Biological Design Unit (BioDU)

Spruce extracts and prepares parts of a single BU to create BioDUs. A biological design unit (*BioDU*) is a fully prepared macromolecular system considered ready for modeling (generally considered Molecular Dynamics ready). Heterogen components are componentized into other_ligands, other_cofactors, counter_ions, metals, and so on. Thus, it has all the information required to generate a design unit, but is not yet focused on a specific binding site and as such does not contain receptor grids required for docking or other actions. The BioDU has limited usage, as most OpenEye tools require binding sites and their related information. The data structure is stored in an `OEDesignUnit`.

BioDUs are only applicable when a user does not need or want a binding site. The primary uses are:

- Cryptic Pocket Detection Floes Package
- Plain Molecular Dynamics Floe
- Solvate and Run MD Floe

2.5 Design Unit

Design unit (DU) is an idea first discussed by Pfizer [Gehlhaar-2022]. The `OEDesignUnit` class is a technical term for a specific data structure created to capture all the information in a macromolecular system and is unique to OpenEye. An OEDesignUnit file format is an oedu.

A DU is a specific instance of a BioDU. It is binding site focused. One BioDU may yield several DUs if there are multiple binding sites. If there are multiple ligands in a BioDU, each will include its own information related to the binding site.

A DU includes among other components:

1. A protein and/or nucleic acid
2. Site residues
3. A bound ligand in the binding site (if present, otherwise apo)
4. Cofactors in the binding site
5. Packing residues (if any exist near the site)
6. Excipients (if any exist near the site)

One can interact with each part of the DU through APIs, which are listed under `OEDesignUnit`.

2.6 Spruce TK Preparation Workflow

Spruce TK provides an end-to-end preparation workflow using the *`OEMakeDesignUnits`* API. Options (see *OEMakeDesignUnitOptions* and associated options classes) are available to control for the desired behavior. Metadata (see *OEStructureMetadata*) can be supplied to infuse experimental data, like the proper variant sequence, and also to fix common problems reading from PDB files (e.g., bond order perception). **Spruce** has a built-in Chemical Components Dictionary matching residue codes to SMILES, derived from the corresponding entry at the RCSB, but curated to fix improper molecule forms. The workflow automatically runs through the following steps to produce OEDesignUnit objects.

1. Biological unit extraction (see *`OEExtractBioUnits`*).
2. Generation of packing residues for visualization of crystal contacts (if the structure is from X-ray).
3. Assignment or enumeration of alternate locations (see the *`OEAlternateLocationOption`* namespace).
4. Separation of the system into components, such as identifying ligands, cofactors, and excipients.
5. Construction of missing side chains, modeling loops, and capping chain breaks (see *`OEBuildSidechains`*, *`OEBuildLoops`*, and *`OECapTermini`*).
6. Hydrogen placement and optimization, including searching ligand tautomer states (see *`OEProtonateDesignUnit`*).
7. Assignment of partial charges to the entire system (see OEDesignUnitCharges).
8. Superposition to a common reference frame (see *OEStructuralSuperposition* and *OESecondaryStructureSuperposition*).
9. Visualization and more of an encoded OEInteractionHintContainer.
10. Estimation of the model quality using Iridium [Warren-2012], stored in OEIridiumData.

> **Note**
>
> Multiple design units are often produced. This results from multiple biological units generated during step 1, and further expands that set when enumerating the alternate locations in step 3. The Iridium categorization (see *Categorization*) should be helpful in choosing which (if not all) of the produced units to use in downstream modeling tasks. We find that the Iridium categories are also helpful for managing modeling expectations, having a quality measure of the input data for modeling.

Note

Included with **Spruce TK** is a large database of loop templates generated by parsing the structures in the public PDB repository pulled from the RCSB. This database is available from db-downloads in a platform-independent format. The **SPRUCE** loop database is large (~13G) and will take time to download with a good internet connection. When the download is complete, move the file to a convenient location; a network drive is not recommended for performance.

To use it during preparation (OEMakeDesignUnits), set the path in SetLoopDBFilename method on the *OELoopBuilderOptions* class.

The database can be appended to (or updated) using the **LoopDB_Builder** utility program that ships with the **SPRUCE** application. This would be in the event that the user has a collection of internal or proprietary structures, or if specific structures are released to the public PDB between OpenEye updates of the database that are crucial for a given target that is less well described by the existing templates.

2.7 Proton Placement and Optimization

Traditionally, most biomolecular structures have been generated with either no explicit hydrogens or only polar hydrogens. However, having all atoms explicitly represented and positioned to make appropriate hydrogen bonds is sometimes necessary. The function `OEPlaceHydrogens` does this by adding and placing hydrogens and by "flipping" certain functional groups (e.g., side chain amides and imidazoles) as required for an optimal hydrogen bonding network. **Spruce TK** leverages this functionality, but builds on top of this by taking the ligand protomer and tautomer states into account in the *`OEProtonateDesignUnit`* function. The function identifies the heterogens (ligands, cofactors, etc.) in the structure and enumerates their states using the `OEGetReasonableTautomers` function or by using user supplied states. The hydrogen bonding network of the biomolecule is then optimized in the presence of each state independently, and the most favorable state of the complex is chosen based on interaction energies. If a binding site holds two heterogens, such as a ligand and a cofactor, and each have multiple tautomer states, the combinatorial number of states are optimized and evaluated. To do this efficiently, only the binding site(s) are initially optimized. After selecting the appropriate states of each of the binding sites, the remaining parts of the system are optimized, keeping these states fixed.

2.8 Protein Side Chain Modeling

Amino acid side chains are sometimes missing in protein structures due to low or missing density from X-ray diffraction crystallography experiments. This can be due to high flexibility making the assignment of a specific location of the atoms difficult. However, most molecule simulation software requires a position of these atoms, and the lack of them can cause incorrect results. Partial side chains are identified with *`OEGetPartialResidues`*. Using *`OEBuildSidechains`*, these residues are then divided into groups based on proximity and side chain orientation. The groups are independently optimized taking into account their local environment. In each group, the side chains are built and a standard rotamer library from the `OERotamerLibrary` namespace is used. For each side chain, the set of rotamers is evaluated based on an interaction energy with the nearest environment and the most favorable state is chosen. In the case where multiple residues are being built in a group, an iterative procedure attempts to find the optimal energy for the entire set of residues. In the event a side chain cannot be built because poor input geometry causes clashes, the entire cluster is skipped. Water locations that block side chain rotamers result in those water molecules being deleted (optionally) since their placement is most likely an artifact.

2.9 Protein Loop Modeling

Similar to missing side chains above, there are sometimes gaps in protein structures, where the position of the entire amino acid residues cannot be resolved based on the experimental data. The function *`OEBuildLoops`* is able to build

missing gaps (loops). The workflow is illustrated in the figure below. The function identifies missing loops by comparing the SEQRES section of the PDB header with the structure using a sequence alignment. The user can optionally input their own sequence using *OESequenceMetadata* if the SEQRES in the header is incorrect or missing. With the gaps identified, the function loops for templates matching the gap, filters them based on the ability to fit the gap without clashing, and eventually does an optimization of the top candidates in the local environment before picking the most favorable conformation. If multiple results are desired, these can be retrieved using `OEBuildSingleLoop`.

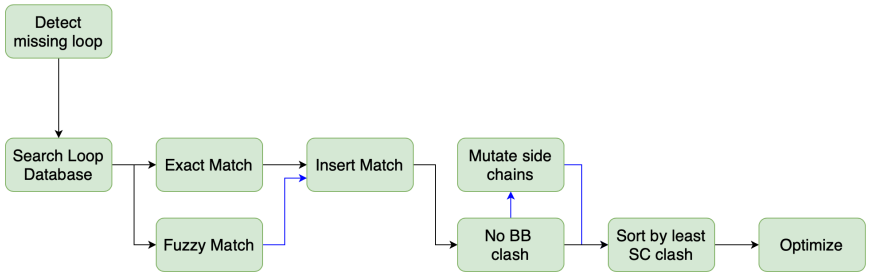


Fig. 1: **Figure 1.** OEBuildLoops: Gaps are searched, filtered, inserted, and evaluated for the best fit.

The loop template database is built using all the structures in the PDB downloaded from the RCSB and can be downloaded from db-downloads.

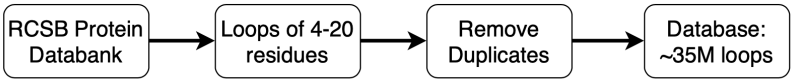


Fig. 2: **Figure 2.** Loop template database. Loops are extracted from the RCSB PDB, deduplicated, compressed, and indexed for easy retrieval.

2.10 Protein Superposition

A protein can be structurally superimposed onto a reference protein structure using **Spruce TK**. Proteins can be superimposed with either atomic coordinates in the *OEStructuralSuperposition* class, or with secondary structure elements using the *OESecondaryStructureSuperposition* class.

The *OEStructuralSuperposition* class can superimpose proteins using any of the following four methods:

1. The global method (default for *OEStructuralSuperposition*). This method uses all matching heavy atoms from the reference and fit proteins as the region for performing the superposition calculation (see `OESuperpositionType.Global`).

2. The **D**ifference **D**istance **M**atrix method. This method calculates the pairwise distance matrix of C-alpha atoms for both the reference and fit proteins, then takes the difference of these two matrices to find the difference distance matrix (DDM).

   For a given conformation a, the elements of the distance matrix D_{ij}^{a} are the distances between atoms *i* and *j* in a molecule,

   $$D_{ij}^{a} = |r_i - r_j|$$

   where r_i and r_j are the Cartesian coordinate vectors of atoms *i* and *j*. The elements Δ_{ij}^{ab} of the difference-distance matrix for two conformations *a* and *b* are

   $$\Delta_{ij}^{ab} = D_{ij}^{a} - D_{ij}^{b}$$

If Δ_{ij}^{ab} is positive, the interatomic vector between between i and j in conformation b is contracted with respect to a. Conversely, negative elements of the DDM indicate an expansion of the interatomic vector. The longest contiguous region of the resulting difference distance matrix (DDM) is used for the structural superposition calculation (see *OESuperpositionType.DDM*).

3. The weighted-DDM method. This method uses the DDM matrix and calculates Gaussian weighting factors for all matching C-alpha atoms. These Gaussian weights are used as a weighting function in the superposition calculation (see *OESuperpositionType.Weighted*).

4. The site residue method. This method uses a set of site residues (given as a set of unique residue strings) as a constraint on the protein superposition. Site residues must be set with the *SetSiteResidues* member function of the *OESuperpositionOptions* class (see *OESuperpositionType.Site*).

The *OESecondaryStructureSuperposition* class can superimpose proteins using:

1. The secondary structure superposition method. This method takes two proteins and performs a structural superposition on them based on the shape overlap of the secondary structure elements of the two proteins.

Note

All structural superposition methods in the *OEStructuralSuperposition* class have a corresponding score from the sequence alignment that was used to find the matching atoms of both proteins. This score comes from the output of the OESequenceAlignment class, where a larger score indicates a better sequence alignment, and scores below a small threshold (around 200) should be considered a bad alignment.

Note

All structural superposition methods in the *OEStructuralSuperposition* class have a corresponding RMSD value for superposition that can be loosely associated with the quality of the superposition. The *OESecondaryStructureSuperposition* class does not have an RMSD value, but instead uses the Tanimoto score from the underlying shape overlap calculation.

2.11 How to Correctly Read a Macromolecular File Format (mmCIF/PDB)

Reading a PDB file correctly for use in subsequent modeling tasks can be challenging. To correctly read a PDB, one must be aware that PDB header information as well as information about alternate location codes within the PDB file will be lost unless a specific combination of PDB-centric `OEIFlavor`s are used. Furthermore, the protein itself must be processed by :oe:cls:`OEAltLocationFactory` in order to create a molecule with all alternate location atoms retained. **Spruce** handles this for the user, so they are not handled by this reader. However, for uses other than **Spruce**, the user needs to either enumerate the different alternate locations or pick a specific location. With that in mind, we recommend reading PDB files for use in **Spruce TK** using the following pattern as shown in **ReadMacromolecule** below:

CHAPTER

THREE

IRIDIUM

Iridium [Warren-2012] is a metric used to estimate the model quality of a structure resulting from an X-ray crystallography experiment. The metric evaluates what we deem essential considerations for using protein–ligand structures in drug discovery, particularly virtual screening.

The metric was developed by aggregating previously published docking validation sets, evaluating each of them by eye, and even re-refining some of the data. The high quality structures were then deposited into a new dataset called Iridium HT [Warren-2012].

The metric takes a number of parameters into account and categorizes structures into four different categories, grading their trustworthiness to use in modeling.

- HT - Highly Trustworthy
- MT - Mildly Trustworthy
- NT - Not Trustworthy
- NA - Not Applicable

The fourth category is used when no electron density data is available, or when the structure being evaluated is not coming from an X-ray crystallography experiment. Electron density data is available for at least 85% of the public PDB data and is required for all new depositions. Historical data is also being recovered, so that percentage is only expected to increase.

Note

Regarding applicability:

- Iridium is not currently an appropriate metric for electron microscopy (EM) experiments, in part because the contour level, which we use as 1 sigma in X-ray experiments, is determined on a per experiment basis, and also in part due to different diffraction behavior of heavy atoms compared to the X-ray.
- Iridium is not currently an appropriate metric for neutron diffraction experiments, due to the different diffraction behavior for heavy atoms (e.g., sulfur and phosphorus are largely invisible in neutron diffraction data).
- Iridium relies on a bound ligand and is therefore not appropriate for apo structures.

3.1 Categorization

The initial Iridium categorization was done by eye. The rules developed were then formalized to be able to rigorously assess the Iridium category on every X-ray structure. In some cases, the original publication is not explicit about threshold values, but the metric below follows the paper in spirit and is approved by the original authors. The outcome of formalizing the rules resulted in minor category discrepancies on the published dataset and is likely due to the more stringent adherence to the rules.

The criteria to consider include both global and local quality metrics and are as follows (the acronyms in parentheses are used when **Spruce** logs the Iridium details):

- DPI - The diffraction-component precision index or global precision estimate [Cruickshank-1999]
- Rfree value
- Crystallographic resolution
- Density coverage of the ligand heavy atoms (LaD)
- Density coverage of the active site residue heavy atoms (including cofactors) (ASaD)
- Occupancy of ligand heavy atoms (POL)
- Occupancy of active site heavy atoms (POAS)
- Alternate locations of the ligand (AltConfs)
- Alternate locations of the active site residues (AltConfs)
- Presence of crystal packing residues near binding site (PackRes)
- Presence of excipients near binding site (Excp)
- Whether ligand is potentially covalently bound, but the covalent bond is not properly perceived (PossCov)

First, the initial consideration is based on the density coverage of the ligand and the active site residues.

	HT	MT	NT
Ligand	> 0.90	< 0.90 and > 0.50	< 0.50
Active Site	> 0.95	< 0.95 and > 0.50	< 0.50

Subsequently, a structure can be demoted from HT to MT if any of the conditions below are true:

- DPI > 0.50
- Presence of alternate locations of the ligand or the active site residues
- Any ligand heavy atoms with an occupancy < 0.90
- Any active site heavy atoms with an occupancy < 0.50
- Perception of interactions between packing residues and the ligand using OEPerceiveInteractionOptions with default settings.
- Perception of interactions between any excipient and the ligand using OEPerceiveInteractionOptions with default settings.
- Perception of unperceived covalent interactions of the ligand in the active site using OEPerceiveInteractionOptions with default settings.

Furthermore, any structure will be demoted to NT if the Rfree value is larger than 0.45 and the resolution is below 3.5A, as we consider that an irrational Rfree value (IrrRFree).

Note

Not all structures report a DPI, and in such cases it is calculated using `OECalculateDPI`.

See also

- OEIridiumData class
- `OECalculateDPI` class
- `OEDrawIridiumData` function in **Grapheme TK** to visualize OEIridiumData

CHAPTER

FOUR

ADVANCED MACROMOLECULAR PREPARATION WORKFLOW

This example provides a walk-through of the steps taken by **SPRUCE** to go from an asymmetric unit of an X-ray crystallography experiment to a model-ready design unit. The system being prepared is a structure of the DNA gyrase using the PDB code: 5CDO.

An example script to create a OEDesignUnit can be found at

Code Example

- *Creating OEDesignUnits from a PDB file*

See also

- *Spruce Theory* documentation

4.1 Inputs

The data required for **Spruce** to prepare a macromolecular system is:

- A structure file: preferably MMCIF but PDB is also supported.
  - We recommend reading the files using the SpruceDefault read flavors for these file formats. See more in this example: *How to Correctly Read a PDB File*.
  - SSBOND and LINK records are read. If the bonds that are read pass some sanity checks, they are applied to the structure if not automatically perceived by the molecule reader.
- Metadata about the sequence and any bound small molecules (heterogens). This can be provided in the metadata/header sections of PDB or MMCIF files. Alternatively, the information can be provided using `OEStructureMetadata` for applications and floes, the content of which can be stored in a JSON file format. The metadata required is:
  - **Experimental and space group information**
    - CRYST1
    - EXPDAT
  - **Title information**
    - HEADER
  - **Revision information**
    - REVDAT

  - **Biological unit information**
    - REMARK 300
    - REMARK 350
  - **Sequence information (using the PDB metadata notation; equivalent keys in MMCIF are also considered)**
    - SEQRES
    - SEQADV
    - DBREF, DBREF1/DBREF2
    - MODRES
  - **Refinement information to be used for Iridium classification**
    - EXPDTA
    - REMARK 3 (if available)

In the example walk-through here, all of the metadata is contained in the PDB file. However, we will use a custom metadata file for the heterogen to augment the bond order and valence states of one of the heterogens. Heterogen information can be provided using the *OEHeterogenMetadata* class within the *OEStructureMetadata* class.

- **Heterogen information**
  - Title/residue name, such as ATP
  - Optionally a corporate ID, such as OE123456
  - SMILES, used for automated tautomer generation with OEQuacpac `OEGetReasonableTautomers` function
  - A list of tautomers if the user wants to provide externally generated tautomers.

4.2 Filtering and Standardization

OESpruceFilter performs a standardization and a filter check before preparing this structure (*OESpruceFilterOptions*), which includes:

- Fixing backbone atom states *OEFixBackbone*
- Fixing bonds to metal
- Checking electron density overlap
- Checking sequence alignment
- Reordering the atoms into PDB order `OEPDBOrderAtoms`
- Including standard residue perception for PDB files and Chem Comp residue perception for CIF files
- Fixing invalid covalent bonds
- Fixing problematic residue namespace
- Fixing chain IDs

Preparation for 5CDO builds missing OXT atoms when fixing the backbone:

```
Debug: Built missing OXT atom in backbone of terminal residue LEU 490   A 1
Debug: Built missing OXT atom in backbone of terminal residue LEU 490   C 3
Debug: Built missing OXT atom in backbone of terminal residue LEU 490   R 7
Debug: Built missing OXT atom in backbone of terminal residue LEU 490   T 9

Debug: Built missing OXT atom in backbone of terminal residue VAL 638   B 2
Debug: Built missing OXT atom in backbone of terminal residue VAL 638   D 4
Debug: Built missing OXT atom in backbone of terminal residue VAL 638   S 8

Debug: Built missing OXT atom in backbone of terminal residue ARG 252   R 7
Debug: Built missing OXT atom in backbone of terminal residue ALA 637   U 10
```

4.3 Generating the Biological Unit

Generation of biological units from asymmetric units is done by reading the _pdbx_struct_assembly, _pdbx_struct_assembly_gen, and _pdbx_struct_oper_list categories in the MMCIF file.

The asymmetric unit contains two biological units as noted in the MMCIF header:

Listing 1: MMCIF format

```
loop_
_pdbx_struct_assembly.id
_pdbx_struct_assembly.details
_pdbx_struct_assembly.method_details
_pdbx_struct_assembly.oligomeric_details
_pdbx_struct_assembly.oligomeric_count
1 author_defined_assembly ? hexameric 6
2 author_defined_assembly ? hexameric 6
```

In the case of the PDB file, generation of biological units from asymmetric units is done by reading the REMARK 300 and 350 in the header.

Listing 2: PDB format

```
REMARK 300 BIOMOLECULE: 1, 2
REMARK 300 SEE REMARK 350 FOR THE AUTHOR PROVIDED AND/OR PROGRAM
REMARK 300 GENERATED ASSEMBLY INFORMATION FOR THE STRUCTURE IN
REMARK 300 THIS ENTRY. THE REMARK MAY ALSO PROVIDE INFORMATION ON
REMARK 300 BURIED SURFACE AREA.
```

By default, **Spruce** uses the author-generated remarks and prefers them over the software-generated remarks. Generating the biological unit from the asymmetric unit can take various forms, including symmetry operations to generate the correct biological form specified.

In the event the end user knows the biological unit of interest, and it is different from the author- or software-provided one, a reference structure can be provided that will be used along with the space group information to generate the desired biological unit(s).

Read more in the *Biological Unit* section and the toolkit API *OEExtractBioUnits*.

In this specific case, the biological units are generated by splitting the asymmetric unit.

Listing 3: MMCIF format

```
loop_
_pdbx_struct_assembly_gen.assembly_id
_pdbx_struct_assembly_gen.oper_expression
_pdbx_struct_assembly_gen.asym_id_list
1 1 A,B,C,D,E,F,M,N,O,P,Q,R,S,T,U,V,W,HA,IA,JA,KA,LA,MA
2 1 G,H,I,J,K,L,X,Y,Z,AA,BA,CA,DA,EA,FA,GA,NA,OA,PA,QA,RA,SA
#
_pdbx_struct_oper_list.id                   1
_pdbx_struct_oper_list.type                 'identity operation'
_pdbx_struct_oper_list.name                 1_555
_pdbx_struct_oper_list.symmetry_operation   x,y,z
_pdbx_struct_oper_list.matrix[1][1]         1.0000000000
_pdbx_struct_oper_list.matrix[1][2]         0.0000000000
_pdbx_struct_oper_list.matrix[1][3]         0.0000000000
_pdbx_struct_oper_list.vector[1]            0.0000000000
_pdbx_struct_oper_list.matrix[2][1]         0.0000000000
_pdbx_struct_oper_list.matrix[2][2]         1.0000000000
_pdbx_struct_oper_list.matrix[2][3]         0.0000000000
_pdbx_struct_oper_list.vector[2]            0.0000000000
_pdbx_struct_oper_list.matrix[3][1]         0.0000000000
_pdbx_struct_oper_list.matrix[3][2]         0.0000000000
_pdbx_struct_oper_list.matrix[3][3]         1.0000000000
_pdbx_struct_oper_list.vector[3]            0.0000000000
```

Listing 4: PDB format

```
REMARK 350 COORDINATES FOR A COMPLETE MULTIMER REPRESENTING THE KNOWN
REMARK 350 BIOLOGICALLY SIGNIFICANT OLIGOMERIZATION STATE OF THE
REMARK 350 MOLECULE CAN BE GENERATED BY APPLYING BIOMT TRANSFORMATIONS
REMARK 350 GIVEN BELOW.  BOTH NONCRYSTALLOGRAPHIC AND
REMARK 350 CRYSTALLOGRAPHIC OPERATIONS ARE GIVEN.
REMARK 350
REMARK 350 BIOMOLECULE: 1
REMARK 350 AUTHOR DETERMINED BIOLOGICAL UNIT: HEXAMERIC
REMARK 350 APPLY THE FOLLOWING TO CHAINS: A, B, C, D, E, F
REMARK 350   BIOMT1   1  1.000000  0.000000  0.000000        0.00000
REMARK 350   BIOMT2   1  0.000000  1.000000  0.000000        0.00000
REMARK 350   BIOMT3   1  0.000000  0.000000  1.000000        0.00000
REMARK 350
REMARK 350 BIOMOLECULE: 2
REMARK 350 AUTHOR DETERMINED BIOLOGICAL UNIT: HEXAMERIC
REMARK 350 APPLY THE FOLLOWING TO CHAINS: R, S, T, U, V, W
REMARK 350   BIOMT1   1  1.000000  0.000000  0.000000        0.00000
REMARK 350   BIOMT2   1  0.000000  1.000000  0.000000        0.00000
REMARK 350   BIOMT3   1  0.000000  0.000000  1.000000        0.00000
```

Now we have two biological units one with chains A, B, C, D, E, F, and the other with chains R, S, T, U, V, W.

4.4 Generating Packing Residues

Packing residues are by default generated by using the crystallographic information to build the crystal lattice around each biological unit and extracting the potentially relevant pieces close to the biological unit. The packing residues are included in the `OEDesignUnit` object for visualization, and they are also used in determination of the *Iridium category* for a given biological unit.

4.5 Enumerating Alternate Locations

By default, **SPRUCE** enumerates all alternate locations. It has two behaviors, one pairing all A locations together, all B locations together, and so on, and another where it picks the primary alternate location based on occupancy and mixes and matches in each alternate location cluster. See the `OEAltLocationFactory` class.

In this specific case, there are three residues with alternate locations, two in chain A and one in chain R. Since each of these chains are in separate biological units, an A and a B form are generated for each of the biological units.

Listing 5: MMCIF format

```
ATOM   1302  N  N      A ARG A  1  84  ? 39.022  -34.720 51.558  0.50 28.55  ? 92   ARG␣
↪A N      1
ATOM   1303  N  N      B ARG A  1  84  ? 39.030  -34.712 51.565  0.50 28.56  ? 92   ARG␣
↪A N      1
...
ATOM   4046  N  N      A ARG A  1  264 ? 34.135  -51.092 43.755  0.50 46.45  ? 272  ARG␣
↪A N      1
ATOM   4047  N  N      B ARG A  1  264 ? 34.162  -51.078 43.762  0.50 46.44  ? 272  ARG␣
↪A N      1
...
ATOM   23895 N  N      A ARG G  1  114 ? 70.262  -22.784 -8.389  0.50 30.89  ? 122  ARG␣
↪R N      1
ATOM   23896 N  N      B ARG G  1  114 ? 70.284  -22.755 -8.432  0.50 30.91  ? 122  ARG␣
↪R N      1
...
```

Listing 6: PDB format

```
ATOM   1302  N   AARG A  92      39.022 -34.720  51.558  0.50 28.55           N
ATOM   1303  N   BARG A  92      39.030 -34.712  51.565  0.50 28.56           N
...
ATOM   4046  N   AARG A 272      34.135 -51.092  43.755  0.50 46.45           N
ATOM   4047  N   BARG A 272      34.162 -51.078  43.762  0.50 46.44           N
...
ATOM  23901  N   AARG R 122      70.262 -22.784  -8.389  0.50 30.89           N
ATOM  23902  N   BARG R 122      70.284 -22.755  -8.432  0.50 30.91           N
...
```

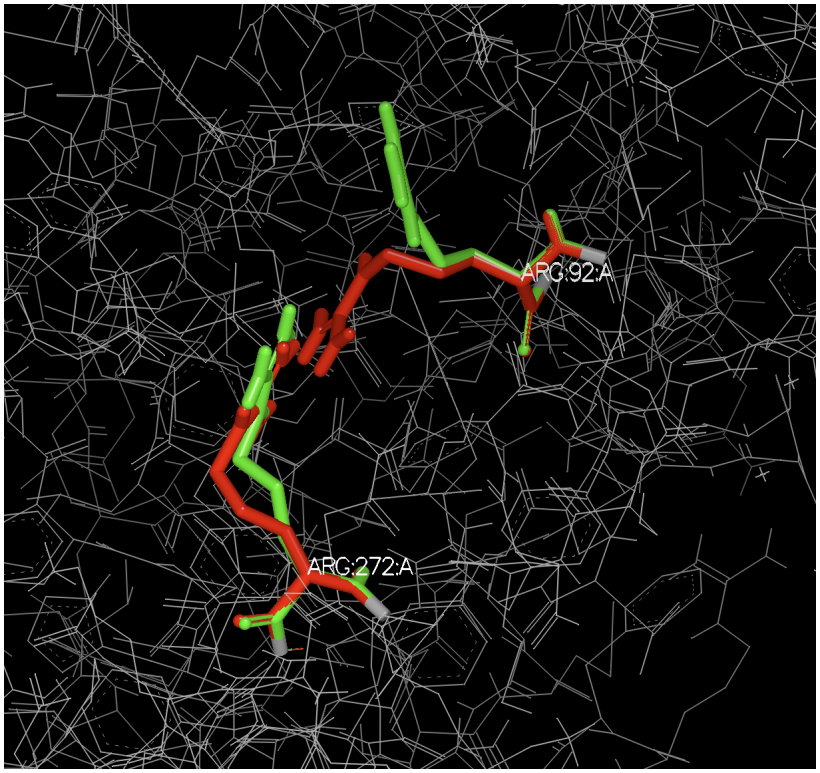


Fig. 1: **Figure 1.** Alternate locations A and B of Arg 92 and 272 in chain A.

4.6 Componentizing the System

Spruce next splits the system into components. It does so for various reasons, but mainly to make it easier for end users and downstream applications to extract the relevant parts for modeling. An example would be to exclude excipients when doing subsequent modeling.

The available list of components can be found in the `OEDesignUnitComponents` namespace.

There are several control parameters for ligand detection. The default parameters generally work very well, but sometimes will need user intervention.

In this specific instance, we focus on the biological unit composed of chains A, B, C, D, E, F. The components being filled are:

- protein

- nucleic
- solvent
- metals
- counter_ions
- packing_residues
- excipients
- other_ligands

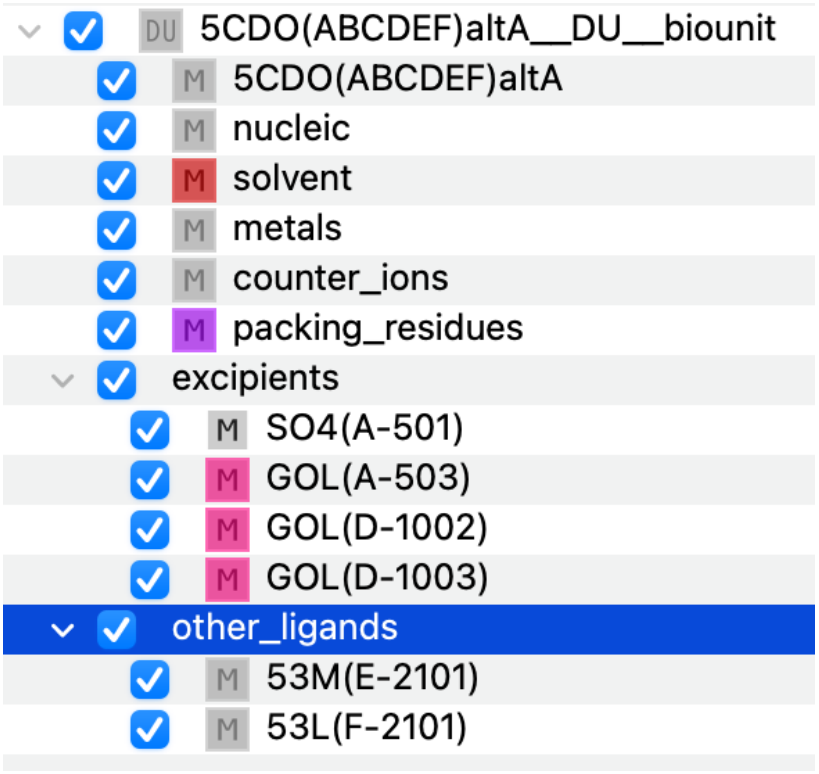


Fig. 2: **Figure 2.** Initial componentization of the biological unit.

The A and B forms of the alternate location enumeration are componentized identically. The unit comprised of chains R, S, T, U, V, W is componentized similarly, but that is not always the case.

Note

There are no binding sites, ligands, or cofactors assigned at this stage. That is done at a later stage when sites in the system are enumerated (*Enumerating Sites / Pockets*). The potential ligands and cofactors are stored in the other_ligands and metals components for the time being; there can also be an other_cofactors component, depending on the nature of the cofactors.

4.7 Building Missing Pieces

Spruce will attempt to build missing pieces that were not observed in the experimental record.

Note

Built residues are styled with a specific color scheme, making built items easy to see in 3D.

4.7.1 Building Missing Side Chains

For the biological unit comprising of chains A, B, C, D, E, F, there were 31 residues that were missing side chains. **Spruce** builds these using built-in rotamer libraries `OERotamerLibrary` and *`OEBuildSidechains`*.

```
Debug:    Residue: ILE 9   A 1
Debug:    Residue: GLU 11   A 1
Debug:    Residue: ARG 231   A 1
Debug:    Residue: ARG 258   A 1
Debug:    Residue: LYS 284   A 1
Debug:    Residue: ARG 299   A 1
Debug:    Residue: LYS 310   A 1
Debug:    Residue: ASP 311   A 1
Debug:    Residue: LYS 405   A 1
Debug:    Residue: GLN 413   A 1
Debug:    Residue: LYS 416   A 1
Debug:    Residue: LEU 490   A 1
Debug:    Residue: LYS 417   B 2
Debug:    Residue: LYS 424   B 2
Debug:    Residue: ARG 450   B 2
Debug:    Residue: LYS 581   B 2
Debug:    Residue: LYS 607   B 2
Debug:    Residue: ARG 12   C 3
Debug:    Residue: ASP 311   C 3
Debug:    Residue: LYS 405   C 3
Debug:    Residue: GLN 413   C 3
Debug:    Residue: LYS 416   C 3
Debug:    Residue: GLU 447   C 3
Debug:    Residue: GLU 475   C 3
Debug:    Residue: LYS 424   D 4
Debug:    Residue: ARG 450   D 4
Debug:    Residue: LYS 497   D 4
Debug:    Residue: GLU 585   D 4
Debug:    Residue: ASN 587   D 4
Debug:    Residue: GLU 593   D 4
Debug:    Residue: LYS 607   D 4
```

Every built side chain is optimized after being built; in cases where multiple side chains need to be built near each other, they are optimized as a cluster. By default, **Spruce** will delete water molecules clashing with built side chains, because it is believed that the water positions are likely not correct but an artifact of missing the side chain during crystallographic refinement.

```
Warning: No rotamer of residue: LYS 497   D 4   passed a clash test. Will check again␣
↪after minimization
```

For the Lysine residue 497 in chain D, no known rotamer would fit the local environment, but **Spruce** is able to find a proper configuration using minimization. In this case, the backbone configuration is not ideal, but a configuration is found that works. It is expected that some backbone rearrangement would happen during a whole system minimization and molecular dynamics simulation.

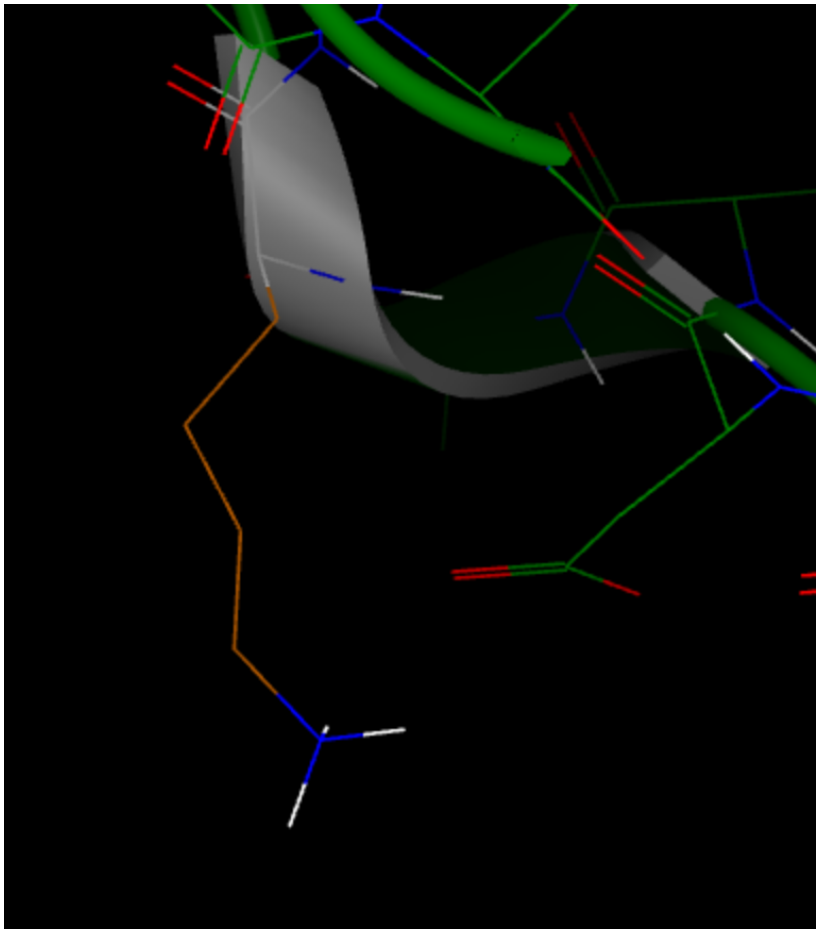


Fig. 3: **Figure 3.** Built side chain of LYS 497 in chain D.

See also

- *OESidechainBuilderOptions* class

4.7.2 Building Missing Loops and Tails

Spruce uses a knowledge-based loop database to fill in missing gaps; see the LoopDBBuilder overview for more information. It uses the SEQRES or sequence data provided using `OEStructureMetadata` to compare the structure to the known sequence. The knowledge-based database currently allows building of loops up to 20 residues in length. For example, in chain R:

```
Debug: Gap detection sequence alignment:
chainR  -NERNITSEMRESFLDYAMSVIVARALPDVRDGLKPVHRRILYGLNEQGM 49
        |||||||||||||||||||||||||||||||||||||||||||||||||||
chain   INERNITSEMRESFLDYAMSVIVARALPDVRDGLKPVHRRILYGLNEQGM 50

chainR  TPDKSYKKSARIVGDVMGKYHPHGDSSIYEAMVRMAQDFSYRYPLVDGQG 99
        |||||||||||||||||||||||||||||||||||||||||||||||||||
chain   TPDKSYKKSARIVGDVMGKYHPHGDSSIYEAMVRMAQDFSYRYPLVDGQG 100

chainR  NFGSMDGDGAAAMRXTEARMTKITLELLRDINKDTIDFIDNYDGNEREPS 149
        |||||||||||||||||||||||||||||||||||||||||||||||||||
chain   NFGSMDGDGAAAMRXTEARMTKITLELLRDINKDTIDFIDNYDGNEREPS 150

chainR  VLPARFPNLLANGASGIAVGMATNIPPHNLTELINGVLSLSKNPDISIAE 199
        |||||||||||||||||||||||||||||||||||||||||||||||||||
chain   VLPARFPNLLANGASGIAVGMATNIPPHNLTELINGVLSLSKNPDISIAE 200

chainR  LMEDIEGPDFPTAGLILGKSGIRRAYETGRGSIQMRSRAVIEER----QR 245
        |||||||||||||||||||||||||||||||||||||||||||||    ||
chain   LMEDIEGPDFPTAGLILGKSGIRRAYETGRGSIQMRSRAVIEERGGGRQR 250

chainR  IVVTEIPFQVNKARMIEKIAELVRDKKIDGITDLRDETSLRTGVRVVIDV 295
        |||||||||||||||||||||||||||||||||||||||||||||||||||
chain   IVVTEIPFQVNKARMIEKIAELVRDKKIDGITDLRDETSLRTGVRVVIDV 300

chainR  RKDANASVILNNLYKQTPLQTSFGVNMIALVNGRPKLINLKEALVHYLEH 345
        |||||||||||||||||||||||||||||||||||||||||||||||||||
chain   RKDANASVILNNLYKQTPLQTSFGVNMIALVNGRPKLINLKEALVHYLEH 350

chainR  QKTVVRRRTQYNLRKAKDRAHILEGLRIALDHIDEIISTIRESDTDKVAM 395
        |||||||||||||||||||||||||||||||||||||||||||||||||||
chain   QKTVVRRRTQYNLRKAKDRAHILEGLRIALDHIDEIISTIRESDTDKVAM 400

chainR  ESLQQRFKLSEKQAQAILDMRLRRLTGLERDKIEAEYNELLNYISELETI 445
        |||||||||||||||||||||||||||||||||||||||||||||||||||
chain   ESLQQRFKLSEKQAQAILDMRLRRLTGLERDKIEAEYNELLNYISELETI 450

chainR  LADEEVLLQLVRDELTEIRDRFGDDRRTEIQL 477
        |||||||||||||||||||||||||||||||
chain   LADEEVLLQLVRDELTEIRDRFGDDRRTEIQL 482
```

(continues on next page)

(continued from previous page)

```
Found unresolved N-terminal with 1 residues before ASN 10   R 7  , with sequence 'I'
Found 4 residue gap between ARG 252   R 7   and GLN 257   R 7  , with sequence 'GGGR'
```

As the log summarizes for chain R, a residue was missing at the N-terminal of the protein, and a 4-residue gap was found with the sequence GGGR between residues 252 and 257.

Terminal residues or "tails" are not built by default, partly because the build time for long tails can be significant as the problem is less constrained than a missing gap in the middle of the protein.

When multiple loops need to be built in a structure, **Spruce** builds the shortest ones first, as they are more geometrically constrained, to avoid a scenario where a loop cannot be built due to the construction of another.

When building a loop, **Spruce** crops a piece of the existing protein (by default, one residue on either side of the gap) and builds a slightly longer loop. This is in part to avoid the scenario where the anchoring residues are in a bad configuration as a result of crystallographic refinement with missing residues.

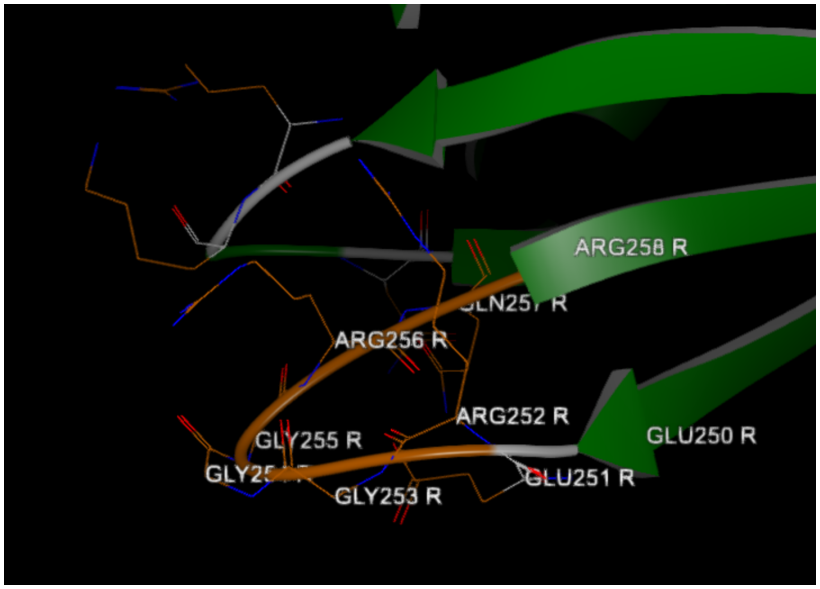


Fig. 4: **Figure 4.** Built loop between Arg 252 and Gln 257 in chain R.

See also

- *OELoopBuilderOptions* class
- *OESidechainBuilderOptions* class
- `OERotamerLibrary` namespace
- *OEStructureMetadata* class

4.7.3 Building Capping Groups

If, for some reason, a missing loop/gap could not be built, **Spruce** adds acetyl (ACE) and N-methyl (NME) capping groups to the terminal residues to avoid having artificial formal charges on the structure. It will, however, not add capping groups to the true biological termini as they are expected to be charged at neutral pH. The information about the true terminals is derived from the SEQRES or provided sequence metadata, similar to the loop.

When building a cap, if the capping group for any reason clashes with other parts of the system (not waters), the residue being capped is instead converted into a capping group. This is to avoid building a system with clashes. The behavior can be modified if desired.

Note

An N-terminal capping group (ACE) is numbered with N-1 to the residue it is capping.

A C-terminal capping group (NME) retains the residue number of the residue it is capping, but an insert code is indented. This is to avoid capping groups and ligands or other features having the same residue number, as it is standard practice to number heterogens following the protein chain using the same chain ID.

See also

- *OECapTermini*
- *OEStructureMetadata* class

4.8 Optimizing the Hydrogen Bond Network

Spruce optimizes the hydrogen bond network by adding explicit hydrogens and resolving clashes particularly around the binding site. The process considers the protonation and tautomer states of the components to produce an optimized protein–ligand complex. A specific tautomer state can be provided as part of metadata if it needs to be enforced.

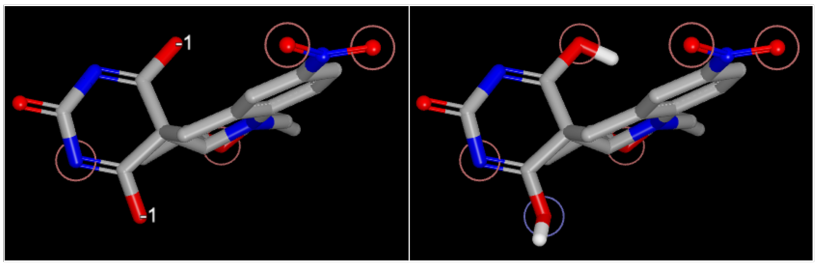


Fig. 5: **Figure 5.** Ligand tautomer state without specified tautomer state (*left*) and with a specified tautomer (C[C@@H]1C[N@@]2c3ccc(cc3CC4([C@H]2[C@@H](O1)C)C(=NC(=O)N=C4O)O)N(=O)=O, *right*).

See also

- *Creating OEDesignUnits Using Provided Metadata*
- Adding Hydrogens and Optimizing Hydrogen Bonds
- `OEPlaceHydrogens`
- *OEProtonateDesignUnit*

4.9 Assignment on Radii and Partial Charges

For more information about the assignment of partial charges to the entire system, please see OEDesignUnitCharges. By default, charges are assigned as follows:

- Standard protein residues: OEAmberFF94Charges
- Nonstandard protein residues: OEAM1BCCCharges
- Nucleic acids: OEMMFF94Charges
- Ligands and or any other heterogens in the system (e.g., cofactors): OEAM1BCCCharges
- Solvent: uses the **SZMAP** water model
- Excipients: OEMMFF94Charges

4.10 Prepared Biological Units (BioDUs)

The above-mentioned stages transformed an asymmetric unit to a biological unit, which was then componentized to generate a prepared biological unit (BioDU).

4.11 Enumerating Sites / Pockets

This is the final stage of creating a design unit from a prepared biological unit.

As was noted in the earlier section, there were two potential ligands found and componentized into the other_ligands category. When ligands are detected, **Spruce** will take the prepared biological unit and generate a design unit for each binding site.

The design unit will now have a binding site defined, and specific styling will be applied to the system. As part of the process, the ligand bound in that binding site will be promoted from the other_ligand category to the ligand category, and cofactors, either from other_cofactors or metals, will be moved to the cofactors category. If a ligand is present, interaction hints are also encoded and stored in the design unit. For more information, see OEInteractionHintContainer.

In the case here, there were two biological units. In the biological units with:

- chains A, B, C, D, E, and F, there were two ligands 53L and 53M.
- chains R, S, T, U, V, and W, there were two ligands 50M and 53M.

Therefore, for each biological unit, two design units were generated. Note that since there were both an A and a B alternate location form of each biological unit, there are really eight design units in the end.

1. 5CDO(ABCDEF)altA > 53L(F-2101)
2. 5CDO(ABCDEF)altB > 53L(F-2101)
3. 5CDO(ABCDEF)altA > 53M(E-2101)
4. 5CDO(ABCDEF)altB > 53M(E-2101)
5. 5CDO(RSTUVW)altA > 50M(S-6002)
6. 5CDO(RSTUVW)altB > 50M(S-6002)
7. 5CDO(RSTUVW)altA > 53M(W-2101)
8. 5CDO(RSTUVW)altB > 53M(W-2101)

If a ligand was not present, the *OEFindPockets* algorithm will be employed to attempt to detect potential binding sites. If multiple sites are found, the BioDU will be enumerated into a DU for each detected site.

4.12 Assigning an Iridium Category

The Iridium [Warren-2012] categorization process itself is well explained in the *Iridium section*, but in short, the Iridium categorization can be used to select between different protein structures if multiple are available for a given target, and also in the event that multiple design units are generated from a single experimental structure. The Iridium category and derived information is stored in OEIridiumData.

In the case here, the enumeration from a single structure file to eight design units gives a little perspective on why this filtering step can be necessary.

There are design units for three different ligands here: 50M, 53M, 53L.

```
5CDO(ABCDEF)altA > 53L(F-2101), Iridium Category: MT, LaD: 0.85, ASaD: 0.88, DPI: 0.47,␣
↪POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false, IrrRFree: false,
↪ PossCov: false
5CDO(ABCDEF)altB > 53L(F-2101), Iridium Category: MT, LaD: 0.85, ASaD: 0.88, DPI: 0.47,␣
↪POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false, IrrRFree: false,
↪ PossCov: false

5CDO(RSTUVW)altA > 50M(S-6002), Iridium Category: MT, LaD: 0.85, ASaD: 0.86, DPI: 0.47,␣
↪POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false, IrrRFree: false,
↪ PossCov: false
5CDO(RSTUVW)altB > 50M(S-6002), Iridium Category: MT, LaD: 0.85, ASaD: 0.86, DPI: 0.47,␣
↪POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false, IrrRFree: false,
↪ PossCov: false
```

For 50M and 53L, there are two forms due to alternate locations for each. The scores are identical and in this case either form is likely okay to use.

```
5CDO(ABCDEF)altA > 53M(E-2101), Iridium Category: MT, LaD: 0.81, ASaD: 0.85, DPI: 0.47,␣
↪POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false, IrrRFree: false,
↪ PossCov: false
5CDO(ABCDEF)altB > 53M(E-2101), Iridium Category: MT, LaD: 0.81, ASaD: 0.85, DPI: 0.47,␣
↪POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false, IrrRFree: false,
↪ PossCov: false
5CDO(RSTUVW)altA > 53M(W-2101), Iridium Category: MT, LaD: 0.74, ASaD: 0.88, DPI: 0.47,␣
↪POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false, IrrRFree: false,
↪ PossCov: false
5CDO(RSTUVW)altB > 53M(W-2101), Iridium Category: MT, LaD: 0.74, ASaD: 0.88, DPI: 0.47,␣
↪POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false, IrrRFree: false,
↪ PossCov: false
```

For 53M, there are four forms: two resulting from different biological units (chains), and two of each of those resulting from alternate locations. The alternate locations forms have identical scores and do not differentiate the forms. However, for the two biological units, the Iridium information is very different even if their overall scores are both MT. In particular, the fraction of ligand atoms covered by the experimental density is 0.81 versus 0.74, and the fraction of active site residue atoms covered by density is 0.85 versus 0.88. In this case, it will likely take a modeler or crystallographer to decide which is the better pose, and in some cases, both should be used for proper modeling.

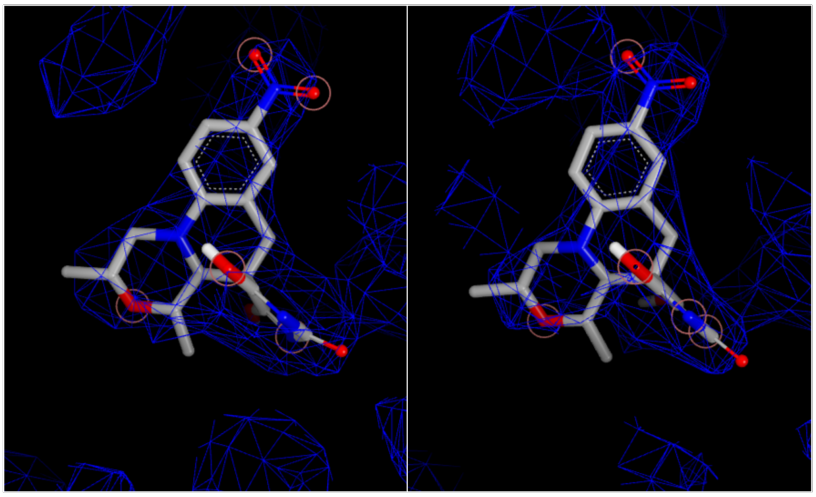


Fig. 6: **Figure 6.** Experimental density and ligand overlap for the two biological units.

4.13 Generating a Docking Receptor

The toolkit does not automatically generate a docking receptor, but the **SPRUCE** application and floe both generate a docking grid using default settings with the `OEMakeReceptor` function. The receptor can later be modified or regenerated with the toolkit or the **Make Receptor** application if a more custom approach is desired.

4.14 Superposition to a Common Frame of Reference

Spruce is able to superpose molecules and design units onto a common reference frame using *`OESuperpose`*. If a common reference structure is provided, all generated design units will be superposed to that reference structure. If a common reference structure is not provided, but each design unit generated is similar to each other (same protein, same binding site), **Spruce** will superpose those similar units onto each other.

4.15 Design Unit Validation

Spruce validates the design unit using *`OEValidateDesignUnit`*. It looks for partial side chains, partial charges, implicit hydrogens, incorrect terminal residues, heavy atom overlap, and broken chains.

CHAPTER

FIVE

EXAMPLES

5.1 Spruce TK Examples

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

Examples:

5.1.1 Creating OEDesignUnits from a PDB file

Preparation of a biological structure file (PDB, mmCIF) to a fully charged, hydrogenated, molecular componentized object (design unit; DU; OEDesignUnit) is one of the more advanced functionalities offered through **Spruce TK**. This example shows how to construct DUs using an input PDB file and the *OEMakeDesignUnits* function.

Command Line Interface

This example uses an input PDB file, and will output a set of DUs from it to a temporary directory (see *OEMakeDesignUnits* for details on the API).

```
make_design_units <input biomolecular PDB> [<electron density mtz>] [
↪<LoopModelingTemplateDB>]
```

Code

Example

will generate the following output:

```
DPI: 0.06, RFree: 0.18, Resolution: 1.60
Processing BU # 0 with title: BETA-SECRETASE 1, chains A, alt: A
Warning: For residue ARG -4   A 1   removing clashing solvent molecule HOH 597   A 2
Warning: For residue ARG -4   A 1   removing clashing solvent molecule HOH 487 A A 1
Warning: For residue ARG 7   A 1   removing clashing solvent molecule HOH 498   A 2
Warning: For residue ARG 7   A 1   removing clashing solvent molecule HOH 730   A 2
Warning: For residue ARG 7   A 1   removing clashing solvent molecule HOH 653   A 2
Warning: For residue ARG 128   A 1   removing clashing solvent molecule HOH 523   A 2
Warning: For residue ARG 128   A 1   removing clashing solvent molecule HOH 654   A 2
Warning: For residue LYS 142   A 1   removing clashing solvent molecule HOH 550   A 2
Warning: For residue LYS 142   A 1   removing clashing solvent molecule HOH 691   A 2
Warning: For residue ARG 205   A 1   removing clashing solvent molecule HOH 423   A 2
Warning: For residue ARG 205   A 1   removing clashing solvent molecule HOH 703   A 2
Warning: For residue LYS 256   A 1   removing clashing solvent molecule HOH 604   A 2
Found gap between ALA 157   A 1   and VAL 170   A 1   , with sequence GFPLNQSEVLAS
```

(continues on next page)

(continued from previous page)

```
Found gap between GLU 310   A 1   and THR 314   A 1  , with sequence DVA
Opened database spruce_bace.loop_db
LoopDatabase Info:
    276 loops from RSCB last synced on 03-19-2020, were added to LoopTemplateDatabase on␣
↪03-19-2020 using Spruce Toolkit 1.0.0.a
    The loop database was built with a max loop length of 22, a termini crop length of 2,
↪ and excluding regular secondary structures
Opened database spruce_bace.loop_db
LoopDatabase Info:
    276 loops from RSCB last synced on 03-19-2020, were added to LoopTemplateDatabase on␣
↪03-19-2020 using Spruce Toolkit 1.0.0.a
    The loop database was built with a max loop length of 22, a termini crop length of 2,
↪ and excluding regular secondary structures
Processing BU # 1 with title: BETA-SECRETASE 1, chains A, alt: B
Warning: For residue ARG -4   A 1   removing clashing solvent molecule HOH 597   A 2
Warning: For residue ARG -4   A 1   removing clashing solvent molecule HOH 487 A A 1
Warning: For residue LYS 9   A 1   removing clashing solvent molecule HOH 675   A 2
Warning: For residue ARG 128   A 1   removing clashing solvent molecule HOH 523   A 2
Warning: For residue ARG 128   A 1   removing clashing solvent molecule HOH 654   A 2
Warning: For residue LYS 142   A 1   removing clashing solvent molecule HOH 448   A 2
Warning: For residue LYS 142   A 1   removing clashing solvent molecule HOH 735   A 2
Warning: For residue ARG 205   A 1   removing clashing solvent molecule HOH 423   A 2
Warning: For residue ARG 205   A 1   removing clashing solvent molecule HOH 703   A 2
Warning: For residue LYS 256   A 1   removing clashing solvent molecule HOH 604   A 2
Found gap between ALA 157   A 1   and VAL 170   A 1  , with sequence GFPLNQSEVLAS
Found gap between GLU 310   A 1   and THR 314   A 1  , with sequence DVA
Opened database spruce_bace.loop_db
LoopDatabase Info:
    276 loops from RSCB last synced on 03-19-2020, were added to LoopTemplateDatabase on␣
↪03-19-2020 using Spruce Toolkit 1.0.0.a
    The loop database was built with a max loop length of 22, a termini crop length of 2,
↪ and excluding regular secondary structures
Opened database spruce_bace.loop_db
LoopDatabase Info:
    276 loops from RSCB last synced on 03-19-2020, were added to LoopTemplateDatabase on␣
↪03-19-2020 using Spruce Toolkit 1.0.0.a
    The loop database was built with a max loop length of 22, a termini crop length of 2,
↪ and excluding regular secondary structures
DU: BETA-SECRETASE 1(A)altA > 5HA(A-999), Iridium Category: HT, LaD: 1.00, ASaD: 1.00,␣
↪DPI: 0.06, POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false,␣
↪IrrRFree: false, PossCov: false
DU: BETA-SECRETASE 1(A)altB > 5HA(A-999), Iridium Category: HT, LaD: 1.00, ASaD: 1.00,␣
↪DPI: 0.06, POL: false, POAS: false, AltConfs: false, PackRes: false, Excp: false,␣
↪IrrRFree: false, PossCov: false
Skipping redundant DU with alts outside the site of interest, renaming existing to␣
↪collapse alts
Discarding redundant alt DU with title BETA-SECRETASE 1(A)altB > 5HA(A-999)
```

See also

- *OEMakeDesignUnits* function

- `OEReadDesignUnit` and `OEWriteDesignUnit` functions in the **OEChem TK** manual
- OEDesignUnit class in the **OEChem TK** manual

5.1.2 Creating apo OEDesignUnits from a PDB file

Preparation of a biological structure file (PDB, mmCIF) to a fully charged, hydrogenated, molecular componentized object (design unit; DU; OEDesignUnit) is one of the more advanced functionalities offered through **Spruce TK**. This example shows how to construct apo DUs using an input PDB file and the *`OEMakeDesignUnits`* function.

Command Line Interface

This example uses an input PDB file, and will output a set of DUs from it to a temporary directory (see *`OEMakeDesignUnits`* for details on the API).

```
make_apo_design_units <input biomolecular PDB> <site_residue> [<electron density mtz>]
↪<LoopModelingTemplateDB>]
```

Code

Example

will generate the following output (failing to add partial charges to iodine):

```
DPI: 0.13, RFree: 0.28, Resolution: 1.75
Processing BU # 0 with title: BETA-SECRETASE 1, chains A
Found gap between ALA 157   A 1   and ALA 168   A 1   , with sequence GFPLNQSEVL
Opened database spruce_bace.loop_db
LoopDatabase Info:
    276 loops from RSCB last synced on 03-19-2020, were added to LoopTemplateDatabase on␣
↪03-19-2020 using Spruce Toolkit 1.0.0.a
    The loop database was built with a max loop length of 22, a termini crop length of 2,
↪ and excluding regular secondary structures
Warning: Failed to charge miscellaneous components
Warning: Unable to add partial charges and radii to DesignUnit: BETA-SECRETASE 1(A)__DU__
↪biounit
Iridium Category: NA, DPI: 0.13, RFree: 0.28, Resolution: 1.75
```

See also

- *`OEMakeDesignUnits`* function
- `OEReadDesignUnit` and `OEWriteDesignUnit` functions in the **OEChem TK** manual
- OEDesignUnit class in the **OEChem TK** manual

5.1.3 Creating mmcif file from an input DU

This example shows how to write an output mmcif file from a spruce generated design unit.

Command Line Interface

This example uses an input design unit (oedu) file, and will output a mmcif file from it to a temporary directory (see *OEMakeDesignUnits* for details on the API).

```
prompt> du2mmcif <input design unit>
```

Code

Example

See also

- *OEMakeDesignUnits* function
- OEReadDesignUnit and OEWriteDesignUnit functions in the **OEChem TK** manual
- OEDesignUnit class in the **OEChem TK** manual

5.1.4 Calculating superposition

The superposition calculation is a functionality offered through the **Spruce TK**. This example shows how to calculate structural superposition using either the global, DDM, weighted-DDM, site sequence, or SSE methods.

Command Line Interface

This example takes in a reference and a fit protein (from either a PDB or OEDU file) and aligns them with one of the supported methods (see *OESuperposeMethod* for methods and descriptions).

```
prompt> superpose <reference protein PDB> <fit protein PDB>␣
↪[global|ddm|weighted|sse|site] [site-residue file]
```

Code

Example

will generate the following output:

```
Superposing 1CLL.pdb to 1A29.pdb using ddm
Writing superimposed fit protein to 1CLL_sp.oeb.gz
RMSD: 0.596203 Angstroms
SeqScore: 642
```

See also

- *OESuperpose* class
- *OESuperposeOptions* class
- *OESuperposeResults* class
- *OESuperposeMethod* namespace

5.1.5 Extracting Biounits from PDB header remarks

Biounit (BU) extraction is one of the more basic functionalities offered through the **Spruce TK**. This example shows how to extract BUs using the PDB header remarks.

Command Line Interface

This example takes in a PDB file, and optionally the maximum number of atoms to be considered a BU as well as a boolean flag to instruct the code to prefer author or software generated headers (see *OEExtractBioUnits* for details on the API).

```
prompt> extract_biounits_remarks <extract protein PDB> [max atoms] [prefer author]
```

Code

Example

will generate the following output:

```
Writing biounit 0 to 4re6_BU_0.oeb.gz
Writing biounit 1 to 4re6_BU_1.oeb.gz
```

See also

- *OEExtractBioUnits* function
- *OEBioUnitExtractionOptions* class

5.1.6 Extracting Biounits by aligning to a reference protein

Biounit (BU) extraction is one of the more basic functionalities offered through the **Spruce TK**. This example shows how to extract BUs using a sequence alignment to a reference protein.

Command Line Interface

This example takes in an extraction PDB file, a reference PDB file, and optionally the minimum score for the alignment to the reference as well as a boolean flag to instruct the code to superimpose the output BU to the input reference protein (see *OEExtractBioUnits* for details on the API).

```
prompt> extract_biounits_ref <extract protein PDB> <reference protein PDB> [min score]␣
↪[superpose]
```

Code

Example

will generate the following output:

```
Writing biounit 0 to 4OBD_BU_0.oeb.gz
Writing biounit 1 to 4OBD_BU_1.oeb.gz
```

See also

- *OEExtractBioUnits* function

- *OEBioUnitExtractionOptions* class

5.1.7 Finding pockets and their properties (like, Surface Area) from a PDB file

This example shows how to find pockets in an input PDB file using the `OEFindPockets` function. OEDesignUnit file (OEDesignUnit) can also be used as input to find the pockets.

Command Line Interface

This example uses an input PDB file, and will print surface area and list of residues for each found pocket in given PDB file.

```
prompt> findpockets <input protein molecule or DesignUnit file>
```

Code

Example

See also

- *OEPocketOptions* class

5.1.8 Write a metadata JSON file from a FASTA file

This example shows how to read in a FASTA to structure metadata and write to a JSON file.

Example

Code

5.1.9 Validate Structure of the Design Unit

This example validates the structure of the design unit.

See also

- *OEValidateDesignUnit* class
- *OEDesignUnitIssueCodes* namespace

Example

Code

5.1.10 Add Receptor Grid to Design Unit

This example adds a receptor grid to the design unit. Providing the pred and mask arguments allows the user to control which component of the design unit is going to be treated as the receptor.

See also

- OEMakeReceptor function

Example

Code

5.1.11 Creating OEDesignUnits Using Provided Metadata

Preparation of a biological structure file (PDB, mmCIF) to a fully charged, hydrogenated, molecular componentized object (design unit, or DU; OEDesignUnit) is one of the more advanced functionalities offered through **Spruce TK**. Metadata can be provided to **Spruce TK** to identify the target ligand, the desired ligand tautomeric state, and the DU title. In the case where SEQRES data is not present for the structure, the sequence specified in the metadata can be used to build missing gaps and identify natural termini versus termini that need to be capped.

Metadata can be provided in the form of a JSON. The example below shows a template for it. It was generated by exporting an *OEStructureMetadata* class.

```
{
   "StructureID": "",
   "Author": "",
   "Keywords": [],
   "RevisionDate": "",
   "Revision": "",
   "ExperimentDate": "",
   "ExperimentType": "UNKNOWN",
   "IridiumData": {
      "Category": "NA",
      "LaD": 0.0,
      "ASaD": 0.0,
      "POL": false,
      "POAS": false,
      "AltConfs": false,
      "PackRes": false,
      "Excp": false,
      "IrrRFree": false,
      "PossCov": false,
      "DPI": 20.0,
      "RFree": 0.0,
      "Resolution": 0.0,
      "HasMTZ": false
   },
   "SequenceMetadata": [],
   "HeterogenMetadata": []
}
```

This example shows how to construct DUs with a specific ligand tautomer by providing ligand metadata.

Command Line Interface

This example uses an input PDB/mmCIF file and a metadata JSON file, and will output a set of DUs with the specified ligand tautomers. Setting the -generate_tautomers option to false ensures that only the tautomers specified in the metadata file are considered. Setting the -opt_expt_protons option to true ensures that experimental protons are optimized and thus the tautomeric state can be changed based on the provided metadata. The provided script can be used with a modified metadata file to accomplish the other mentioned examples as well.

```
prompt> make_design_unit_using_metadata.py -in <PDB/mmCIF file> -loop_db <loop database>␣
↪-metadata <metadata JSON> -generate_tautomers false -opt_expt_protons true
```

Code

See also

- *OEMakeDesignUnits* function
- `OEReadDesignUnit` and `OEWriteDesignUnit` functions in the **OEChem TK** manual
- OEDesignUnit class in the **OEChem TK** manual
- *OEStructureMetadata* class

CHAPTER

SIX

OESPRUCE API

6.1 OESpruce Classes

Attention

This API is currently available in C++ and Python.

6.1.1 OEBioUnitExtractionOptions

```
class OEBioUnitExtractionOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets for biounit extraction for use in the *OEExtractBioUnits* function.

Code Example

- *Extracting Biounits by aligning to a reference protein* example
- *Extracting Biounits from PDB header remarks* example

Constructors

```
OEBioUnitExtractionOptions()
OEBioUnitExtractionOptions(const OEBioUnitExtractionOptions &)
```

Default and copy constructor that constructs a *OEBioUnitExtractionOptions* object.

GetMaxAtoms

```
unsigned GetMaxAtoms() const
```

Returns the maximum number of atoms for the extracted biounit.

GetMaxParts

```
unsigned GetMaxParts() const
```

Returns the maximum number of parts for a given transformation required to generate a biounit.

GetMinScore

```
unsigned GetMinScore() const
```

Returns the minimum score required for the biounit extraction's sequence alignment.

GetPreferAuthorRecord

```
bool GetPreferAuthorRecord() const
```

Returns the boolean whether or not to prefer the author or software record for biounit extraction.

GetSuperpose

```
bool GetSuperpose() const
```

Returns the boolean whether or not to superpose the output biounit to the input reference structure.

6.1.2 SetMaxAtoms

```
bool SetMaxAtoms(unsigned maxAtoms)
```

Sets the maximum number of atoms allowed for the extracted biounit.

6.1.3 SetMaxParts

```
bool SetMaxParts(unsigned maxParts)
```

Sets the maximum number of parts allowed for a given transformation required to generate a biounit.

6.1.4 SetMinScore

```
bool SetMinScore(const unsigned value)
```

Sets the minimum score required for the biounit extraction's sequence alignment.

6.1.5 SetPreferAuthorRecord

```
bool SetPreferAuthorRecord(bool preferAuthor)
```

Sets the boolean of whether or not to prefer the author or software record for biounit extraction.

6.1.6 SetSuperpose

```
bool SetSuperpose(bool superpose)
```

Sets the boolean whether or not to superpose the output biounit to the input reference structure.

See also

- *OEMakeDesignUnitOptions* class

6.1.7 OECapBuilderOptions

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OECapBuilderOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets for building caps in the *OECapTermini*, *OECapCTermini*, and *OECapNTermini* functions.

Constructors

```
OECapBuilderOptions()
OECapBuilderOptions(const OECapBuilderOptions &)
```

Default and copy constructor that constructs a *OECapBuilderOptions* object.

GetDeleteClashingSolvent

```
bool GetDeleteClashingSolvent() const
```

Returns the boolean whether or not to delete solvent waters that clash with other atoms.

GetAllowTruncate

```
bool GetAllowTruncate() const
```

Returns the boolean whether or not it is okay to make a terminal residue the cap if building one would result in a clash, or if the backbone of the residue itself is incomplete

GetForceCapping

```
bool GetForceCapping() const
```

Returns the boolean of whether or not to force caping in the event the cap would clash with something in the system

SetDeleteClashingSolvent

```
bool SetDeleteClashingSolvent(bool deleteClashingSolvent)
```

Sets the boolean of whether or not to delete solvent waters that clash with other atoms.

SetAllowTruncate

```
bool SetAllowTruncate(bool allowTruncate)
```

Sets the boolean of whether or not it is okay to make a terminal residue the cap if building one would result in a clash, or if the backbone of the residue itself is incomplete

SetForceCapping

```
bool SetForceCapping(bool forceCapping)
```

Sets the boolean of whether or not to force capping in the event the cap would clash with something in the system. This setting supersedes the settings for *OECapBuilderOptions.GetAllowTruncate* and *OECapBuilderOptions.GetDeleteClashingSolvent*.

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- *OEDesignUnitBuildOptions* class
- *OECapTermini* function
- *OECapCTermini* function
- *OECapNTermini* functions

6.1.8 OEDesignUnitBuildOptions

Attention

This API is currently available in C++ and Python.

```
class OEDesignUnitBuildOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets for building the missing structural components of a design unit for use in the *OEMakeDesignUnitOptions* class.

Constructors

```
OEDesignUnitBuildOptions()
OEDesignUnitBuildOptions(const OEDesignUnitBuildOptions &)
```

Default and copy constructor that constructs a *OEDesignUnitBuildOptions* object.

GetBuildLoops

```
bool GetBuildLoops() const
```

Returns the boolean whether or not to build loops.

GetBuildSidechains

```
bool GetBuildSidechains() const
```

Returns the boolean whether or not to build side chains.

GetCapCTermini

```
bool GetCapCTermini() const
```

Returns the boolean whether or not to cap all C-termini in the system with NME.

GetCapNTermini

```
bool GetCapNTermini() const
```

Returns the boolean whether or not to cap all N-termini in the system with ACE.

GetSidechainBuilderOptions

```
OESidechainBuilderOptions& GetSidechainBuilderOptions()
const OESidechainBuilderOptions& GetSidechainBuilderOptions() const
```

Returns a reference to the stored sidechain builder options (*OESidechainBuilderOptions*) class.

GetLoopBuilderOptions

```
OELoopBuilderOptions& GetLoopBuilderOptions()
const OELoopBuilderOptions& GetLoopBuilderOptions() const
```

Returns a reference to the stored loop builder options (*OELoopBuilderOptions*) class.

GetCapBuilderOptions

```
OECapBuilderOptions& GetCapBuilderOptions()
const OECapBuilderOptions& GetCapBuilderOptions() const
```

Returns a reference to the stored cap builder options (*OECapBuilderOptions*) class.

6.1.9 SetBuildLoops

```
bool SetBuildLoops(bool buildLoops)
```

Sets the boolean of whether or not to build loops.

6.1.10 SetBuildSidechains

```
bool SetBuildSidechains(bool buildSidechains)
```

Sets the boolean of whether or not to build side chains.

6.1.11 SetCapCTermini

```
bool SetCapCTermini(bool capCTermini)
```

Sets the boolean of whether or not to cap all C-termini in the system with NME.

6.1.12 SetCapNTermini

```
bool SetCapNTermini(bool capNTermini)
```

Sets the boolean of whether or not to cap all N-termini in the system with ACE.

6.1.13 SetSidechainBuilderOptions

```
void SetSidechainBuilderOptions(const OESidechainBuilderOptions& value)
```

Set the sidechain builder options class (*OESidechainBuilderOptions*).

6.1.14 SetCapBuilderOptions

```
void SetCapBuilderOptions(const OECapBuilderOptions& value)
```

Set the cap builder options class (*OECapBuilderOptions*).

6.1.15 SetLoopBuilderOptions

```
void SetLoopBuilderOptions(const OELoopBuilderOptions& value)
```

Set the loop builder options class (*OELoopBuilderOptions*).

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- *OESidechainBuilderOptions* class
- *OELoopBuilderOptions* class
- *OECapBuilderOptions* class

6.1.16 OEDesignUnitEnumerateSitesOptions

Attention

This API is currently available in C++ and Python.

```
class OEDesignUnitEnumerateSitesOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets for design unit binding site enumeration for use in the *OEEnumerateSites* function.

Constructors

```
OEDesignUnitEnumerateSitesOptions()
OEDesignUnitEnumerateSitesOptions(const OEDesignUnitEnumerateSitesOptions &)
```

Default and copy constructor that constructs a *OEDesignUnitEnumerateSitesOptions* object.

GetAddInteractionHints

```
bool GetAddInteractionHints() const
```

Returns the boolean whether or not to add interaction hints.

GetAddStyle

```
bool GetAddStyle() const
```

Returns the boolean whether or not to add style.

GetCollapseNonSiteAlts

```
bool GetCollapseNonSiteAlts() const
```

Returns the boolean whether or not to collapse non site alternate locations.

GetDuplicateRemoval

```
bool GetDuplicateRemoval() const
```

Returns the boolean whether or not to remove duplicate sites.

GetEnumerateCofactorSites

```
bool GetEnumerateCofactorSites() const
```

Returns the boolean whether or not to enumerate cofactor sites.

GetRestrictToRefSite

```
bool GetRestrictToRefSite() const
```

Returns the boolean whether or not to restrict enumeration to the site of the reference structure.

GetSiteSize

```
double GetSiteSize() const
```

Returns the size of the site.

6.1.17 SetAddInteractionHints

```
bool SetAddInteractionHints(bool addInteractions)
```

Sets the boolean whether or not to add interactions.

6.1.18 SetAddStyle

```
bool SetAddStyle(bool addStyle)
```

Sets the boolean whether or not to add style.

6.1.19 SetCollapseNonSiteAlts

```
bool SetCollapseNonSiteAlts(bool collapseNonSiteAlts)
```

Sets the boolean whether or not to collapse non site alternate locations.

6.1.20 SetDuplicateRemoval

```
bool SetDuplicateRemoval(bool duplicateRemoval)
```

Sets the boolean whether or not to remove duplicate sites.

6.1.21 SetEnumerateCofactorSites

```
bool SetEnumerateCofactorSites(bool enumCofactorSites)
```

Sets the boolean whether or not to enumerate cofactor sites.

6.1.22 SetRestrictToRefSite

```
bool SetRestrictToRefSite(bool restrictToRefSite)
```

Sets the boolean whether or not to restrict enumeration to the reference site.

6.1.23 SetSiteSize

```
bool SetSiteSize(double siteSize)
```

Sets the size of the site.

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- *OEDesignUnitEnumerateSitesOptions* class

6.1.24 OEDesignUnitMutationOptions

Attention

This API is currently available in C++ and Python.

```
class OEDesignUnitMutationOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets used in design unit mutation for use in the *OEMutateResidues* function.

Constructors

```
OEDesignUnitMutationOptions()
OEDesignUnitMutationOptions(const OEDesignUnitMutationOptions &)
```

Default and copy constructors.

AddMutation

```
void AddMutation(const OEChem::OEResidue mutationRes,
                 const unsigned targetIdx)
```

Adds a mutation to make based on an input OEResidue object, along with a constant representing the target component which is taken from the `OEDesignUnitComponents` namespace.

ClearMutations

```
void ClearMutations()
```

Clears all mutations.

GetMutations

```
std::map<OEChem::OEResidue, unsigned> GetMutations() const
```

Returns the stored *std::map* of mutations, which is keyed on an OEResidue object with a value of the target component which is taken from the `OEDesignUnitComponents` namespace.

SetMutations

```
void SetMutations(const std::map<OEChem::OEResidue, unsigned>& mutations)
```

Sets the stored *std::map* of mutations, which is keyed on an OEResidue object with a value of the target component which is taken from the `OEDesignUnitComponents` namespace.

See also

- *OESidechainBuilderOptions* class

6.1.25 OEDesignUnitPrepOptions

Attention

This API is currently available in C++ and Python.

```
class OEDesignUnitPrepOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets used in design unit preparation for use in the *OEMakeDesignUnitOptions* class.

Constructors

```
OEDesignUnitPrepOptions()
OEDesignUnitPrepOptions(const OEDesignUnitPrepOptions &)
OEDesignUnitPrepOptions(const OEDesignUnitBuildOptions& buildOpts,
                        const OESpruce::OEProtonateDesignUnitOptions& protonateOpts,
                        const OEDesignUnitEnumerateSitesOptions& enumOpts)
```

Default, copy, and advanced constructor that constructs a *OEDesignUnitPrepOptions* object. The advanced constructor take three separate sub-options classes as arguments: an options class that controls prep (*OEDesignUnitPrepOptions*), an options class that controls the building options (*OEDesignUnitBuildOptions*), and an options class that controls alternate location handling (*OEDesignUnitEnumerateSitesOptions*).

GetAssignPartialChargesAndRadii

```
bool GetAssignPartialChargesAndRadii() const
```

Returns the boolean whether or not to assign partial charges and radii.

GetBuildOptions

```
OEDesignUnitBuildOptions& GetBuildOptions()
const OEDesignUnitBuildOptions& GetBuildOptions() const
```

Returns a reference to the stored build options (*OEDesignUnitBuildOptions*) class.

GetChargeEngine

```
OEProton::OEDesignUnitCharges& GetChargeEngine()
```

Returns a reference to the stored charge engine (OEDesignUnitCharges) class.

GetEnumerateSitesOptions

```
OEDesignUnitEnumerateSitesOptions& GetEnumerateSitesOptions()
const OEDesignUnitEnumerateSitesOptions& GetEnumerateSitesOptions() const
```

Returns a reference to the stored design unit site enumeration options (*OEDesignUnitEnumerateSitesOptions*) class.

GetProtonate

```
bool GetProtonate() const
```

Returns the boolean of whether or not to protonate the design unit.

GetStrictProtonationMode

```
bool GetStrictProtonationMode() const
```

Returns the boolean of whether to fail design unit preparation if the protonation stage fails

GetProtonateOptions

```
OESpruce::OEProtonateDesignUnitOptions& GetProtonateOptions()
const OESpruce::OEProtonateDesignUnitOptions& GetProtonateOptions() const
```

Returns a reference to the stored design unit protonation options (*OEProtonateDesignUnitOptions*) class.

6.1.26 SetAssignPartialChargesAndRadii

```
bool SetAssignPartialChargesAndRadii(bool assignPartialChargesAndRadii)
```

Sets the boolean of whether or not to assign partial charges and radii.

6.1.27 SetBuildOptions

```
void SetBuildOptions(const OEDesignUnitBuildOptions& buildOpts)
```

Sets the design unit building options class (*OEDesignUnitBuildOptions*).

6.1.28 SetChargeEngine

```
void SetChargeEngine(OEProton::OEDesignUnitCharges chargeEngine)
```

Sets the charging engine when charges are added (OEDesignUnitCharges).

6.1.29 SetEnumerateSitesOptions

```
void SetEnumerateSitesOptions(const OEDesignUnitEnumerateSitesOptions&␣
↪enumOpts)
```

Sets the design unit site enumeration options class (*OEDesignUnitEnumerateSitesOptions*).

6.1.30 SetProtonate

```
bool SetProtonate(const bool protonate)
```

Sets the boolean of whether or not to protonate the design unit.

6.1.31 SetStrictProtonationMode

```
bool SetStrictProtonationMode(bool strictProtonate)
```

Sets the boolean of whether or not to fail design unit preparation if the protonation stage fails

6.1.32 SetProtonateOptions

```
void SetProtonateOptions(const OESpruce::OEProtonateDesignUnitOptions&␣
↪protonateOpts)
```

Sets the design unit protonation options class (*OEProtonateDesignUnitOptions*).

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- OEDesignUnitCharges class
- *OEDesignUnitEnumerateSitesOptions* class
- *OEProtonateDesignUnitOptions* class
- OEPlaceHydrogensOptions class
- *OEDesignUnitBuildOptions* class
- *OELoopBuilderOptions* class
- *OESidechainBuilderOptions* class
- *OELoopBuilderOptions* class
- *OECapBuilderOptions* class

6.1.33 OEDesignUnitSplitOptions

Attention

This API is currently available in C++ and Python.

```
class OEDesignUnitSplitOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets for splitting a structure into design unit components for use in the *OEMakeDesignUnitOptions* class.

Constructors

```
OEDesignUnitSplitOptions()
OEDesignUnitSplitOptions(const OEDesignUnitSplitOptions &)
```

Default and copy constructor that constructs a *OEDesignUnitSplitOptions* object.

AddCofactorCode

```
void AddCofactorCode(const std::string argValue)
```

Explicitly sets the name of the molecule(s) to be considered a cofactor.

AddExcipientCode

```
void AddExcipientCode(const std::string argValue)
```

Explicitly sets the name of the molecule(s) to be considered an excipient.

AddLipidCode

```
void AddLipidCode(const std::string argValue)
```

Explicitly sets the name of the molecule(s) to be considered an lipid.

ClearCofactorCodes

```
void ClearCofactorCodes()
```

Removes the name(s) of the molecules considered to be cofactors.

ClearExcipientCodes

```
void ClearExcipientCodes()
```

Removes the name(s) of the molecules considered to be excipients.

ClearLipidCodes

```
void ClearLipidCodes()
```

Removes the name(s) of the molecules considered to be lipids.

GetAlternateLocationHandling

```
unsigned GetAlternateLocationHandling() const
```

Returns the value of the alternate location handling mode, which is given by the *OEAlternateLocationOption* namespace.

GetCofactorCodes

```
std::vector<std::string> GetCofactorCodes() const
```

Returns a reference to the vector of the name(s) of the molecule(s) considered to be cofactors.

GetExcipientCodes

```
std::vector<std::string> GetExcipientCodes() const
```

Returns a reference to the stored vector of strings representing names of molecules that should be considered excipients.

GetLipidCodes

```
std::vector<std::string> GetLipidCodes() const
```

Returns a reference to the stored vector of strings representing names of molecules that should be considered lipids.

GetMakePackingResidues

```
bool GetMakePackingResidues() const
```

Returns the boolean whether or not to make packing residues.

GetMaxLigAtoms

```
unsigned GetMaxLigAtoms() const
```

Returns the maximum number of atoms for a molecule to still be considered a ligand.

GetMaxLigResidues

```
unsigned GetMaxLigResidues() const
```

Returns the maximum number of residues for a molecule to still be considered a ligand.

GetMaxSystemAtoms

```
unsigned GetMaxSystemAtoms() const
```

Returns the maximum number of atoms for a system to be split.

GetMinLigAtoms

```
unsigned GetMinLigAtoms() const
```

Returns the minimum number of atoms for a molecule to still be considered a ligand.

GetTargetComponentID

```
unsigned GetTargetComponentID() const
```

Returns the constant of the target component, which is taken from the `OEDesignUnitComponents` namespace.

SetAlternateLocationHandling

```
bool SetAlternateLocationHandling(unsigned altLocHandling)
```

Sets the constant that controls how alternate location enumeration will be handled, which is taken from the *OEAlternateLocationOption* namespace.

SetCofactorCodes

```
bool SetCofactorCodes(const std::vector<std::string>& argValue)
```

Sets the vector of strings representing names of molecules that should be considered cofactors.

SetExcipientCodes

```
bool SetExcipientCodes(const std::vector<std::string>& argValue)
```

Sets the vector of strings representing names of molecules that should be considered excipients.

SetLipidCodes

```
bool SetLipidCodes(const std::vector<std::string>& argValue)
```

Sets the vector of strings representing names of molecules that should be considered lipids.

SetMakePackingResidues

```
bool SetMakePackingResidues(bool makePackingResidues)
```

Sets the boolean of whether or not to make a packing residue component.

SetMaxLigAtoms

```
void SetMaxLigAtoms(unsigned maxLigAtoms)
```

Sets the maximum number of atoms for a molecule to be considered a ligand.

SetMaxLigResidues

```
void SetMaxLigResidues(unsigned maxLigResidues)
```

Sets the maximum number of residues for a molecule to be considered a ligand.

SetMaxSystemAtoms

```
bool SetMaxSystemAtoms(unsigned maxSystemAtoms)
```

Sets the maximum number of atoms for a system to be split.

SetMinLigAtoms

```
void SetMinLigAtoms(unsigned minLigAtoms)
```

Sets the minimum number of atoms for a molecule to be considered a ligand.

SetTargetComponentID

```
bool SetTargetComponentID(unsigned targetCompID)
```

Sets the constant of the target component, which is taken from the `OEDesignUnitComponents` namespace.

See also

- *OEMakeDesignUnitOptions* class

6.1.34 OEHeterogenMetadata

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEHeterogenMetadata
```

This class stores metadata for a heterogen molecule, including its title, ID, SMILES representation, heterogen type (as defined in the *OEHeterogenType* namespace), and any user-specified tautomers. A map of `OEHeterogenMetadata` objects keyed by SMILES can be passed to *OEProtonateDesignUnit* to guide protonation of specific heterogen components.

The title is the string used to map to the name in the molecule file, historically a 3-character string in PDB file format e.g. LIG, 5HA (from wwPDB 3TPP), but can now be a 5 character string using the new mmCIF format and updated wwPDB conventions.

The ID is meant for users who want to then title the ligand molecule, the ligand name in the design unit title and output design unit filenames with an internal compound ID, e.g. "OE-000123".

Constructors

```
OEHeterogenMetadata()
OEHeterogenMetadata(const OEHeterogenMetadata &)
```

Default and copy constructors.

AddTautomer

```
void AddTautomer(const std::string& tautomer)
```

Add a tautomer using a Smiles string.

AddTautomers

```
void AddTautomers(const std::vector<std::string>& tautomers)
```

Add a set of tautomers of the ligand using a *std::vector* of Smiles strings.

ClearTautomers

```
void ClearTautomers()
```

Clears all tautomer strings.

GetID

```
std::string GetID() const
```

Returns the stored ID string for the ligand.

GetSmiles

```
std::string GetSmiles() const
```

Returns the stored Smiles string for the ligand.

GetTautomers

```
std::vector<std::string>& GetTautomers()
const std::vector<std::string>& GetTautomers() const
```

Returns a reference of the set of stored tautomers for the ligand.

GetTitle

```
std::string GetTitle() const
```

Returns the ligand's title.

GetType

```
unsigned GetType() const
```

Returns the type of heterogen of the ligand as defined in the *OEHeterogenType* namespace.

SetID

```
void SetID(const std::string id)
```

Sets a string of the ligand's ID.

SetSmiles

```
void SetSmiles(const std::string smiles)
```

Sets the Smiles string for the ligand.

SetTitle

```
void SetTitle(const std::string title)
```

Sets the ligand's title.

SetType

```
void SetType(const unsigned type)
```

Sets the type of heterogen of the ligand as defined in the *OEHeterogenType* namespace.

6.1.35 OEIsModeledAtom

Attention

This API is currently available in C++ and Python.

```
class OEIsModeledAtom : public OESystem::OEUnaryPredicate<OEChem::OEAtomBase>
```

This predicate is used to identify atoms that have been built using **Spruce TK** either by capping terminal residues, building out partial residues, or by mutating a residue. Otherwise it returns `false`.

The following methods are publicly inherited from OEUnaryPredicate:

operator()	CreateCopy	CreatePredicateCopy

The following methods are publicly inherited from OEUnaryFunction:

operator()	CreateCopy

Constructors

```
OEIsModeledAtom()
```

Default constructor.

operator()

```
bool operator()(const OEChem::OEAtomBase& atom) const
```

Returns `true` if the atom has been built using **Spruce TK** either by capping terminal residues, building out partial residues, or by mutating a residue.

CreateCopy

```
OESystem::OEUnaryFunction<OEChem::OEAtomBase, bool>* CreateCopy() const
```

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

See also

- *OECapTermini* function
- *OECapCTermini* function
- *OECapNTermini* function
- *OEBuildSidechains* function
- *OEMutateResidue* function

6.1.36 OELoopBuilderOptions

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OELoopBuilderOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets for building loops in the *OEBuildLoops* and *OEBuildSingleLoop* functions.

Constructors

```
OELoopBuilderOptions()
OELoopBuilderOptions(const OELoopBuilderOptions &)
```

Default and copy constructor that constructs a *OELoopBuilderOptions* object.

GetLoopDBFilename

```
std::string GetLoopDBFilename() const
```

Returns the filename of the loop database used when searching for compatible loops.

GetUsePackingResidues

```
bool GetUsePackingResidues() const
```

Returns the boolean whether or not to use crystal packing residues during loop modeling (if available).

GetCropLength

```
unsigned GetCropLength() const
```

Returns the number of residues on either side of a gap to remove, since these are known to sometimes be poorly resolved and therefore have configurations that make it difficult to build loops.

GetTransformThreshold

```
unsigned GetTransformThreshold() const
```

Returns the number of loops from the database to transform, meaning mutate from their database sequence to target sequence, if they are not 100% sequence matching. Structures with 100% sequence match are not affected by this threshold.

GetStrictProlineMatch

```
bool GetStrictProlineMatch() const
```

Returns the boolean of whether or not fuzzy sequence matches have to have proline residues in the correct position. This can be relevant since they have a different backbone conformation than other amino acid residues.

GetConnectBufferDistance

```
double GetConnectBufferDistance() const
```

Returns the buffer distance allowed between the anchor atoms (connection points) when searching for fuzzy sequence matches.

GetBackboneClashRejectThreshold

```
double GetBackboneClashRejectThreshold() const
```

Returns the threshold of clashing backbone atoms. If the clash level is above this threshold a loop from the database is rejected, and is not processed further.

GetDeleteExistingResidues

```
bool GetDeleteExistingResidues() const
```

Returns whether or not to delete residues already present at site where new loops needs to be built.

GetLoopClashRejectThreshold

```
double GetLoopClashRejectThreshold() const
```

Returns the threshold of clashing loops atoms. This threshold excludes already considered clashing backbone atoms. If the clash level is above this threshold a loop from the database is rejected, and is not processed further.

GetOptimizationShell

```
double GetOptimizationShell() const
```

Returns the radius of atoms being considered when minimizing the loops in place. Since multiple loops are being optimized, it is the union of atoms inside the radius around each starting loop configuration that is considered. This allows comparison of the energies between each loop conformation as the systems are of identical size.

GetOptimizationTolerance

```
double GetOptimizationTolerance() const
```

Returns the tolerance used by the optimizer.

GetOptimizationStage1IterMultiplier

```
unsigned GetOptimizationStage1IterMultiplier() const
```

Returns the multiplier of optimization steps per residue in the loop used during the initial steepest descent optimization.

GetOptimizationStage2IterMultiplier

```
unsigned GetOptimizationStage2IterMultiplier() const
```

Returns the multiplier of optimization steps per residue in the loop used during the secondary BFGS optimization.

GetOptimizationUseSolventModel

```
bool GetOptimizationUseSolventModel() const
```

Returns whether or not to use a simplistic solvent model during the loop optimization.

GetOptimizationInclSurfaceAreaTerm

```
bool GetOptimizationInclSurfaceAreaTerm() const
```

Returns whether or not to include the surface area term when scoring the loops after optimization.

GetOptimizationMaxLoops

```
unsigned GetOptimizationMaxLoops() const
```

Returns the number of loops to optimize, the higher the number the longer the calculation will take, but also includes more structural diversity.

GetAllowBuildDisulfideBridges

```
bool GetAllowBuildDisulfideBridges() const
```

Returns whether or not to allow building disulfide bridges between the loop and the protein if possible, requires cystine residues in proximity.

GetRejectClashingLoopsPostMinimization

```
bool GetRejectClashingLoopsPostMinimization() const
```

Returns whether or not to reject clashing loops post-minimization.

GetSeqAlignMethod

```
unsigned GetSeqAlignMethod() const
```

Returns the sequence alignment method, from the `OESeqAlignmentMethod` namespace, used to compare the provided sequence and the structure for gaps.

GetSeqAlignGapPenalty

```
int GetSeqAlignGapPenalty() const
```

Returns the sequence alignment gap penalty, used to compare the provided sequence and the structure for gaps with the `OEGetSimpleAlignment` function.

GetSeqAlignExtendPenalty

```
int GetSeqAlignExtendPenalty() const
```

Returns the sequence alignment extend penalty, used to compare the provided sequence and the structure for gaps with the `OEGetSimpleAlignment` function.

GetBuildTails

```
bool GetBuildTails() const
```

Returns whether or not to allow building missing tails at the N and C termini.

SetLoopDBFilename

```
bool SetLoopDBFilename(std::string filename)
```

Sets the filename of the loop database to use when searching for compatible loops.

SetUsePackingResidues

```
bool SetUsePackingResidues(bool usePackingResidues)
```

Sets whether or not to use crystal packing residues during loop modeling (if available).

SetCropLength

```
bool SetCropLength(unsigned cropLength)
```

Returns the number of residues on either side of a gap to remove, since these are known to sometimes be poorly resolved and therefore have configurations that make it difficult to build loops.

SetTransformThreshold

```
bool SetTransformThreshold(unsigned transformThreshold)
```

Sets the number of loops from the database to transform, meaning mutate from their database sequence to target sequence, if they are not 100% sequence matching. Structures with 100% sequence match are not affected by this threshold.

SetStrictProlineMatch

```
bool SetStrictProlineMatch(bool strictProline)
```

Sets whether or not fuzzy sequence matches have to have proline residues in the correct position. This can be relevant since they have a different backbone conformation than other amino acid residues.

SetConnectBufferDistance

```
bool SetConnectBufferDistance(double buffer)
```

Sets the buffer distance allowed between the anchor atoms (connection points) when searching for fuzzy sequence matches.

SetBackboneClashRejectThreshold

```
bool SetBackboneClashRejectThreshold(double bbClashThreshold)
```

Sets the threshold of clashing backbone atoms. If the clash level is above this threshold a loop from the database is rejected, and is not processed further.

SetDeleteExistingResidues

```
bool SetDeleteExistingResidues(bool deleteExistingResidues)
```

Sets whether or not to delete residues already present at site where new loops needs to be built. See also *SetRejectClashingLoopsPostMinimization*.

SetLoopClashRejectThreshold

```
bool SetLoopClashRejectThreshold(double loopClashThreshold)
```

Sets the threshold of clashing loops atoms. This threshold excludes already considered clashing backbone atoms. If the clash level is above this threshold a loop from the database is rejected, and is not processed further.

SetOptimizationShell

```
bool SetOptimizationShell(double optimizationShellRadius)
```

Set the radius of atoms being considered when minimizing the loops in place. Since multiple loops are being optimized, it is the union of atoms inside the radius around each starting loop configuration that is considered. This allows comparison of the energies between each loop conformation as the systems are of identical size.

SetOptimizationTolerance

```
bool SetOptimizationTolerance(double optimizationTolerance)
```

Sets the tolerance used by the optimizer.

SetOptimizationStage1IterMultiplier

```
bool SetOptimizationStage1IterMultiplier(unsigned stage1multiplier)
```

Sets the multiplier of optimization steps per residue in the loop used during the initial steepest descent optimization.

SetOptimizationStage2IterMultiplier

```
bool SetOptimizationStage2IterMultiplier(unsigned stage2multiplier)
```

Sets the multiplier of optimization steps per residue in the loop used during the secondary BFGS optimization.

SetOptimizationUseSolventModel

```
bool SetOptimizationUseSolventModel(bool useSolventModel)
```

Sets whether or not to use a simplistic solvent model during the loop optimization.

SetOptimizationInclSurfaceAreaTerm

```
bool SetOptimizationInclSurfaceAreaTerm(bool useSATerm)
```

Sets whether or not to include the surface area term when scoring the loops after optimization.

SetOptimizationMaxLoops

```
bool SetOptimizationMaxLoops(unsigned maxOptLoops)
```

Sets the number of loops to optimize, the higher the number the longer the calculation will take, but also includes more structural diversity.

SetAllowBuildDisulfideBridges

```
bool SetAllowBuildDisulfideBridges(bool allowBuildDisulfideBridges)
```

Sets whether or not to allow building disulfide bridges between the loop and the protein if possible, requires cystine residues in proximity.

SetRejectClashingLoopsPostMinimization

```
bool GetRejectClashingLoopsPostMinimization() const
```

Set whether or not to reject clashing loops post-minimization. See also *SetDeleteExistingResidues*.

SetSeqAlignMethod

```
bool SetSeqAlignMethod(unsigned seqAlignMethod)
```

Sets the sequence alignment method, from the `OESeqAlignmentMethod` namespace, used to compare the provided sequence and the structure for gaps.

SetSeqAlignGapPenalty

```
bool SetSeqAlignGapPenalty(int gapPenalty)
```

Set the sequence alignment gap penalty, used to compare the provided sequence and the structure for gaps with the `OEGetSimpleAlignment` function.

SetSeqAlignExtendPenalty

```
bool SetSeqAlignExtendPenalty(int extendPenalty)
```

Set the sequence alignment extend penalty, used to compare the provided sequence and the structure for gaps with the `OEGetSimpleAlignment` function.

SetBuildTails

```
bool SetBuildTails(bool BuildTails)
```

Sets whether or not to allow building missing tails at the C and N termini.

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- *OEDesignUnitBuildOptions* class
- *OESidechainBuilderOptions* class
- `OEBuildLoops` function
- `OEBuildSingleLoop` function
- `OESeqAlignmentMethod` namespace
- `OEGetSimpleAlignment` function
- OESequenceAlignment class

6.1.37 OEMakeDesignUnitOptions

Attention

This API is currently available in C++ and Python.

The *OEMakeDesignUnitOptions* class holds other options classes that are used during the structure preparation processes. The organization of those options classes is illustrated in the figure below. Default constructing this options class, constructs all the necessary options classes in the illustration, which can then be accessed and modified. This class stores the optional parameter sets for design unit construction for use in the `OEMakeDesignUnits` and `OEMakeDesignUnit` functions.

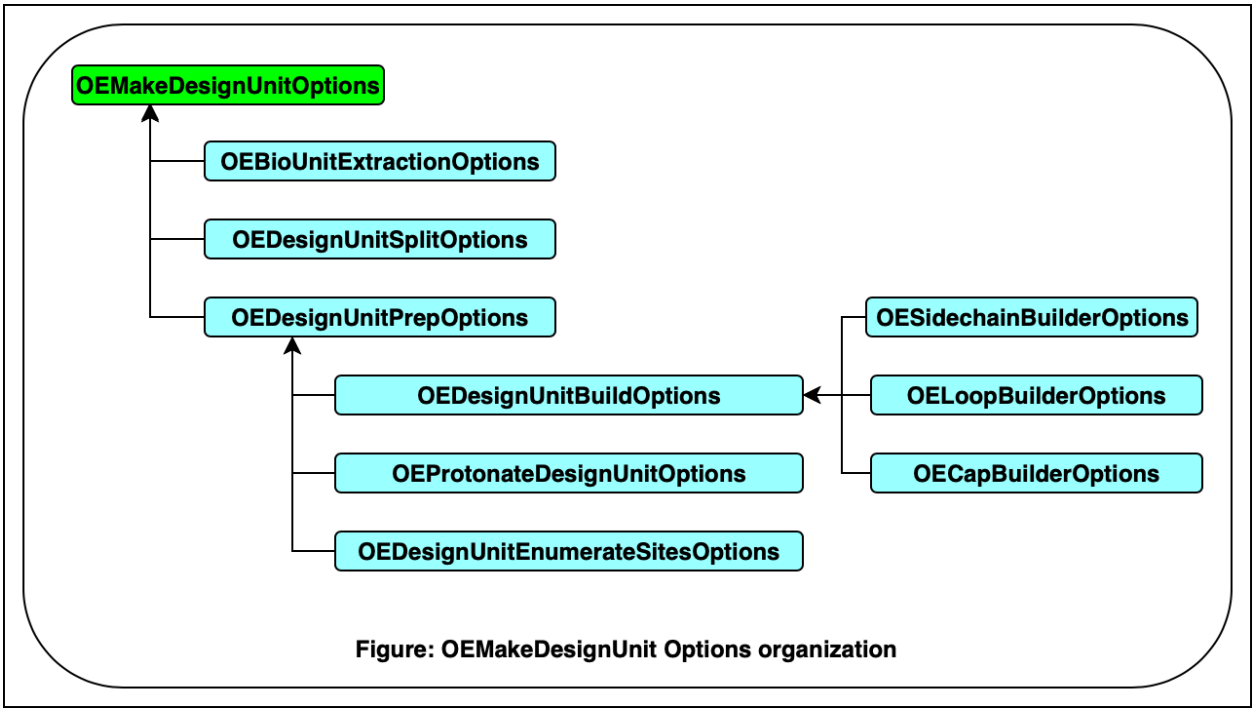


Figure: OEMakeDesignUnit Options organization

Constructors

```
OEMakeDesignUnitOptions()
OEMakeDesignUnitOptions(const OEMakeDesignUnitOptions &)
OEMakeDesignUnitOptions(const OEDesignUnitSplitOptions& splitOpts,
                        const OEDesignUnitPrepOptions& prepOpts,
                        const OEBioUnitExtractionOptions& buExtractOpts)
```

Default, copy, and advanced constructor that constructs a *OEMakeDesignUnitOptions* object. The advanced constructor take three separate sub-options classes as arguments: an options class that controls componentization of the system (*OEDesignUnitSplitOptions*), an options class that controls prep (*OEDesignUnitPrepOptions*), and an options class that controls biounit extraction (*OEBioUnitExtractionOptions*).

GetBioUnitExtractionOptions

```
OEBioUnitExtractionOptions& GetBioUnitExtractionOptions()
const OEBioUnitExtractionOptions& GetBioUnitExtractionOptions() const
```

Returns a reference to the stored biounit extraction options (*OEBioUnitExtractionOptions*) class.

GetPrepOptions

```
OEDesignUnitPrepOptions& GetPrepOptions()
const OEDesignUnitPrepOptions& GetPrepOptions() const
```

Returns a reference to the stored design unit prep options (*OEDesignUnitPrepOptions*) class.

GetSplitOptions

```
OEDesignUnitSplitOptions& GetSplitOptions()
const OEDesignUnitSplitOptions& GetSplitOptions() const
```

Returns a reference to the stored options for splitting options (*OEDesignUnitSplitOptions*) class.

GetSuperpose

```
bool GetSuperpose() const
```

Returns the boolean whether or not to superpose the design units onto a common reference.

GetSuperpositionMethod

```
unsigned GetSuperpositionMethod() const
```

Returns the stored superposition method taken from the *OESuperpositionType* namespace.

SetBioUnitExtractionOptions

```
void SetBioUnitExtractionOptions(const OESpruce::OEBioUnitExtractionOptions& value)
```

Sets the biounit extraction options class (*OEBioUnitExtractionOptions*).

SetPrepOptions

```
void SetPrepOptions(const OESpruce::OEDesignUnitPrepOptions& value)
```

Sets the design unit prep options class (*OEDesignUnitPrepOptions*).

SetSplitOptions

```
void SetSplitOptions(const OESpruce::OEDesignUnitSplitOptions& value)
```

Sets the system molecular componentization options class (*OEDesignUnitSplitOptions*).

SetSuperpose

```
bool SetSuperpose(bool superpose)
```

Sets the boolean of whether or not to superpose design units to a common reference.

SetSuperpositionMethod

```
bool SetSuperpositionMethod(unsigned superposeMethod)
```

Sets the superposition method from a value in the *OESuperpositionType* namespace.

See also

- *OEBioUnitExtractionOptions* class
- *OEDesignUnitSplitOptions* class
- *OEDesignUnitPrepOptions* class
- OEDesignUnitCharges class
- *OEDesignUnitEnumerateSitesOptions* class
- *OEProtonateDesignUnitOptions* class
- OEPlaceHydrogensOptions class
- *OEDesignUnitBuildOptions* class
- *OELoopBuilderOptions* class
- *OESidechainBuilderOptions* class
- *OELoopBuilderOptions* class
- *OECapBuilderOptions* class

6.1.38 OEPocket

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEPocket
```

This class stores the pockets resulting from running *OEFindPockets* function.

Constructors

```
OEPocket(const OESpicoli::OESurface& surf, OEChem::OEMolBase& mol)
OEPocket(const OEPocket &)
```

Default and copy constructor that constructs a *OEPocket* object.

GetSurfaceArea

```
double GetSurfaceArea() const
```

Returns the surface area for the pocket surface.

GetLocalDensity

```
double GetLocalDensity() const
```

Returns the local density score for the pocket. A score > 50 is considered high quality.

GetPocketPredicate

```
const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& GetPocketPredicate() const
```

Returns an atom predicate for the pocket. The pocket is residue based and can therefore be transferred to structures in the same target.

GetPocketSurface

```
const OESpicoli::OESurface& GetPocketSurface() const
OESpicoli::OESurface& GetPocketSurface()
```

Returns a reference to the surface of the pocket.

GetAtoms

```
OESystem::OEIterBase<OEChem::OEAtomBase>* GetAtoms()
```

Returns an iterator of the atoms making up the pocket surface.

GetResidues

```
OESystem::OEIterBase<OEHierResidue>* GetResidues()
```

Returns an iterator of the OEResidues making up the pocket.

See also

- *OEFindPockets* function
- *OEPocketOptions* class
- *OEMakeDesignUnitFromPocket* function

6.1.39 OEPocketOptions

Attention

This API is currently available in C++ and Python.

```
class OEPocketOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets for pocket finding using *OEFindPockets* function.

Constructors

```
OEPocketOptions()
OEPocketOptions(const OEPocketOptions &)
```

Default and copy constructor that constructs a *OEPocketOptions* object.

GetMinSurfaceArea

```
double GetMinSurfaceArea() const
```

Returns the minimum surface area for a detected pocket surface. [Default: $150\ Angstrom^2$]

GetMaxSurfaceArea

```
double GetMaxSurfaceArea() const
```

Returns the maximum surface area for a detected pocket surface. [Default: $3000\ Angstrom^2$]

GetBurialFactor

```
double GetBurialFactor() const
```

Returns the factor used to distinguish pockets from the general surface. [Default: 1.4]

6.1.40 SetMinSurfaceArea

```
bool SetMinSurfaceArea(const double value)
```

Sets the minimum surface area for a detected pocket surface.

6.1.41 SetMaxSurfaceArea

```
bool SetMaxSurfaceArea(const double value)
```

Sets the maximum surface area for a detected pocket surface.

6.1.42 SetBurialFactor

```
bool SetBurialFactor(const double value)
```

Sets the factor used to distinguish pockets from the general surface.

See also

- *OEFindPockets* function
- *OEPocket* class

6.1.43 OEProtonateDesignUnitOptions

Attention

This API is currently available in C++ and Python.

```
class OEProtonateDesignUnitOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets used in design unit protonation for use in the *OEProtonateDesignUnit* function.

Constructors

```
OEProtonateDesignUnitOptions()
OEProtonateDesignUnitOptions(const OEBio::OEPlaceHydrogensOptions& placeHOpts)
OEProtonateDesignUnitOptions(const OEProtonateDesignUnitOptions& rhs)
```

Default and copy constructors.

GetGenerateTautomers

```
bool GetGenerateTautomers() const
```

Returns the boolean whether or not to generate tautomers for the ligand and cofactors.

GetHetGroupNbrDist

```
double GetHetGroupNbrDist() const
```

Returns the distance cutoff for two groups to be considered within the same cluster.

GetOptimizeExpProtons

```
bool GetOptimizeExpProtons() const
```

Returns the boolean whether or not to optimize the position of experimental protons.

GetPlaceHydrogensOptions

```
OEBio::OEPlaceHydrogensOptions& GetPlaceHydrogensOptions()
const OEBio::OEPlaceHydrogensOptions& GetPlaceHydrogensOptions() const
```

Returns a reference to the stored place hydrogen options (OEPlaceHydrogensOptions) class.

6.1.44 SetGenerateTautomers

```
bool SetGenerateTautomers(const bool generateTautomers)
```

Sets the boolean of whether or not to generate tautomers.

6.1.45 SetHetGroupNbrDist

```
bool SetHetGroupNbrDist(const double hetGroupNbrDist)
```

Sets the distance cutoff for two groups to be considered within the same cluster.

6.1.46 SetOptimizeExpProtons

```
bool SetOptimizeExpProtons(const bool optimizeExpProtons)
```

Sets the boolean whether or not to optimize the position of experimental protons.

6.1.47 SetPlaceHydrogensOptions

```
void SetPlaceHydrogensOptions(const OEBio::OEPlaceHydrogensOptions& value)
```

Sets the place hydrogens options class (OEPlaceHydrogensOptions).

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- OEPlaceHydrogensOptions class

6.1.48 OESidechainBuilderOptions

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESidechainBuilderOptions : public OESystem::OEOptions
```

This class stores the optional parameter sets for building sidechains in the *OEBuildSidechains*, *OEMutateResidue*, *OEMutateResidues*, *OEBuildLoops*, *OEBuildSingleLoop* functions.

Constructors

```
OESidechainBuilderOptions()
OESidechainBuilderOptions(const OESidechainBuilderOptions &)
OESidechainBuilderOptions(bool minimizeSidechainsShell)
```

Default, copy, and parameterized constructors for the *OESidechainBuilderOptions* object.

GetDeleteClashingSolvent

```
bool GetDeleteClashingSolvent() const
```

Returns the boolean whether or not to delete solvent waters that clash with other atoms.

GetMaxClusterSize

```
unsigned GetMaxClusterSize() const
```

Returns the maximum number of partial residues allowed in a cluster. Clusters are made based on proximity of partial residues to each other.

GetMinimizeSidechains

```
bool GetMinimizeSidechains() const
```

Returns the boolean whether or not to minimize the sidechains of the target residues.

GetMinimizeSidechainsShell

```
bool GetMinimizeSidechainsShell() const
```

Returns the boolean whether or not to minimize the sidechains of the residues around the target residues.

GetRotamerLibrary

```
unsigned GetRotamerLibrary() const
```

Returns the rotamer library used for side chain rebuilding.

GetStrictMode

```
bool GetStrictMode() const
```

Returns the boolean whether or not to fail early if any of the partial or missings sidechain could not be built

SetRotamerCoverage

```
bool SetRotamerCoverage(double rotamerCoverage)
```

Sets the value for the rotamer coverage percentage to be used for side chain rebuilding.

SetRotamerLibrary

```
bool SetRotamerLibrary(unsigned rotamerLibrary)
```

Sets the boolean of the rotamer library.

SetDeleteClashingSolvent

```
bool SetDeleteClashingSolvent(bool deleteClashingSolvent)
```

Sets the boolean of whether or not to delete solvent waters that clash with other atoms.

SetMaxClusterSize

```
bool SetMaxClusterSize(unsigned)
```

Sets the maximum number of partial residues allowed in a cluster. Clusters are made based on proximity of partial residues to each other.

SetMinimizeSidechains

```
bool SetMinimizeSidechains(bool)
```

Sets the boolean whether or not to minimize the sidechains of the target residues.

SetMinimizeSidechainsShell

```
bool SetMinimizeSidechainsShell(bool)
```

Sets the boolean whether or not to minimize the sidechains of the residues around the target residues.

SetStrictMode

```
bool SetStrictMode(bool)
```

Sets the boolean whether or not to fail early if any of the partial or missings sidechain could not be built

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- *OEDesignUnitBuildOptions* class
- *OEBuildSidechains* function
- *OEMutateResidue* function
- *OEMutateResidues* function

- *OEBuildLoops* function
- *OEBuildSingleLoop* function

6.1.49 OESequenceMetadata

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESequenceMetadata
```

This class stores the sequence data used in biomolecule rebuilding in the *OEMakeDesignUnits* function.

Constructors

```
OESequenceMetadata()
OESequenceMetadata(const OESequenceMetadata &)
```

Default and copy constructors.

GetChainID

```
std::string GetChainID() const
```

Returns the string of the chain ID of the sequence.

GetEndResidueInsertCode

```
char GetEndResidueInsertCode() const
```

Returns a char representing the insertion code of the last residue.

GetEndResidueNumber

```
int GetEndResidueNumber() const
```

Returns the residue number.

GetSequence

```
std::string GetSequence() const
```

Returns the string sequence.

GetStartResidueInsertCode

```
char GetStartResidueInsertCode() const
```

Returns a char representing the insertion code of the first residue.

GetStartResidueNumber

```
int GetStartResidueNumber() const
```

Returns the residue number (*int*) of the first residue.

SetChainID

```
void SetChainID(const std::string& cid)
```

Sets the string of the chain ID of the sequence.

SetEndResidueInsertCode

```
void SetEndResidueInsertCode(const char endInsCode)
```

Sets the char representing the insertion code of the last residue.

SetEndResidueNumber

```
void SetEndResidueNumber(const int endResNum)
```

Sets the residue number (*int*) of the last residue.

SetSequence

```
void SetSequence(const std::string& seq)
```

Sets the string of the sequence. The format should be a three letter code per residue separated by spaces.

SetStartResidueInsertCode

```
void SetStartResidueInsertCode(const char startInsCode)
```

Sets the char representing the insertion code of the first residue.

SetStartResidueNumber

```
void SetStartResidueNumber(const int startResNum)
```

Sets the residue number (*int*) of the first residue.

6.1.50 OEStructureMetadata

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEStructureMetadata
```

This class stores structural information used in design unit preparation within the *OEMakeDesignUnits* function.

Constructors

```
OEStructureMetadata()
OEStructureMetadata(const OEStructureMetadata &)
OEStructureMetadata(const std::vector<OEHeterogenMetadata>& hetData)
OEStructureMetadata(const std::vector<OEHeterogenMetadata>& hetData,
                    const std::vector<OESequenceMetadata>& seqData)
```

Default, copy, and advanced constructor that constructs an *OEStructureMetadata* object. The advanced constructor takes a combination of two separate data classes as arguments: a class that stores the structural metadata (*OEStructureMetadata*), and a class that stores the heterogen metadata (*OEHeterogenMetadata*). A separate overload that takes a *std::vector* of each of those data classes is also available.

AddHeterogenMetadata

```
void AddHeterogenMetadata(const OEHeterogenMetadata& hetData)
void AddHeterogenMetadata(const std::vector<OEHeterogenMetadata>& hetData)
```

Adds the heterogen metadata (*OEHeterogenMetadata*) as a single object or as a *std::vector* of objects.

AddKeyword

```
void AddKeyword(const std::string& keyword)
```

Adds a string keyword.

AddKeywords

```
void AddKeywords(const std::vector<std::string>& keywords)
```

Adds a *std::vector* of string keywords.

AddSequenceMetadata

```
void AddSequenceMetadata(const OESequenceMetadata& seqData)
void AddSequenceMetadata(const std::vector<OESequenceMetadata>& seqData)
```

Adds the sequence metadata (*OESequenceMetadata*) as a single object or as a *std::vector* of objects.

ClearHeterogenMetadata

```
void ClearHeterogenMetadata()
```

Clears all heterogen metadata.

ClearKeywords

```
void ClearKeywords()
```

Clears all keywords.

ClearSequenceMetadata

```
void ClearSequenceMetadata()
```

Clears all sequence metadata.

GetAuthor

```
std::string GetAuthor() const
```

Returns the name of the author.

GetExperimentDate

```
std::string GetExperimentDate() const
```

Returns the string version of the experiment date.

GetExperimentType

```
unsigned GetExperimentType() const
```

Returns the experiment type as defined by the *OEExperimentType* namespace.

GetHeterogenMetadata

```
std::vector<OEHeterogenMetadata>& GetHeterogenMetadata()
const std::vector<OEHeterogenMetadata>& GetHeterogenMetadata() const
```

Returns a reference to the *std::vector* of stored heterogen metadata (*OEHeterogenMetadata*) objects.

GetIridiumData

```
OEBio::OEIridiumData& GetIridiumData()
const OEBio::OEIridiumData& GetIridiumData() const
```

Returns a reference to the Iridium data (OEIridiumData) object.

GetKeywords

```
std::vector<std::string>& GetKeywords()
const std::vector<std::string>& GetKeywords() const
```

Returns a reference to the vector of keyword strings.

GetRevision

```
std::string GetRevision() const
```

Returns the revision.

GetRevisionDate

```
std::string GetRevisionDate() const
```

Returns the string version of the revision date.

GetSequenceMetadata

```
std::vector<OESequenceMetadata>& GetSequenceMetadata()
const std::vector<OESequenceMetadata>& GetSequenceMetadata() const
```

GetStructureID

```
std::string GetStructureID() const
```

Returns the structure's ID.

SetAuthor

```
void SetAuthor(const std::string& author)
```

Sets the author's name.

SetExperimentDate

```
void SetExperimentDate(const std::string& exptDate)
```

Sets the string version of the experiment date.

SetExperimentType

```
void SetExperimentType(unsigned exprType)
```

Sets the experiment type as defined by the *OEExperimentType* namespace.

SetIridiumData

```
void SetIridiumData(const OEBio::OEIridiumData& iridData)
```

Sets the Iridium data from an OEIridiumData object.

SetRevision

```
void SetRevision(const std::string& revVersion)
```

Sets the revision.

SetRevisionDate

```
void SetRevisionDate(const std::string& revDate)
```

Sets the string version of the revision date.

SetStructureID

```
void SetStructureID(const std::string& structureID)
```

Sets the string version of the structure's ID.

6.1.51 OESecondaryStructureSuperposition

> **Warning**
>
> This is a deprecated API. Please use *OESuperpose* class instead.

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESecondaryStructureSuperposition
```

This class represents *OESecondaryStructureSuperposition* that performs a structural superposition on fit protein to a reference using a shape-based alignment of the secondary structure elements.

> **See also**
>
> - *OESuperpositionOptions* class
> - *OEStructuralSuperposition* class

Constructors

```
OESecondaryStructureSuperposition(const OEChem::OEMolBase& refMol,
                                  const OEChem::OEMolBase& fitMol,
                                  const OESuperpositionOptions& argOpts =␣
↪OESuperpositionOptions())
```

Constructor that superimposes the `fitMol` to the `refMol` by using a shape-based alignment to the secondary structure elements. If no *OESuperpositionOptions* are passed in as `argOpts`, then the default behavior of using a superposition to all globally matched atoms will be used.

```
OESecondaryStructureSuperposition(const OESecondaryStructureSuperposition& rhs)
```

Copy constructor.

operator bool

```
operator bool() const
```

Returns whether the *OESecondaryStructureSuperposition* object is valid.

GetFitChains

```
std::vector<std::string> GetFitChains() const
```

Returns 1-letter code of all names of fit protein chains.

See also

- *GetRefChains* method

GetRefChains

```
std::vector<std::string> GetRefChains() const
```

Returns 1-letter code of all names of reference protein chains.

See also

- *GetFitChains <OESpruce_OESecondaryStructureSuperposition_GetFitChains>* method

GetRotMatrix

```
bool GetRotMatrix(double* rmat) const
```

Returns by reference the rotation matrix of the superposition.

GetTanimoto

```
double GetTanimoto() const
```

Returns the Tanimoto score for the structural superposition.

GetTranslation

```
bool GetTranslation(double* trans) const
```

Returns by reference the translation vector of the superposition.

Transform

```
void Transform(OEChem::OEMolBase& mol)
```

Transforms the input molecule using the stored rotation matrix and translation vector from the superposition.

6.1.52 OESpruceFilter

Attention

This API is currently available in C++ and Python.

Note

Note: We strongly recommend running *OESpruceFilter* for standardization and filtering prior to running the design unit preparation.

```
class OESpruceFilter
```

This class performs standardization and quality filtering on a biomolecular structure prior to design unit preparation with `OEMakeDesignUnit` or `OEMakeDesignUnits`. Checks include residue name validity, residue states, covalent bonds to metals, and crystal symmetry.

See also

- *OESpruceFilterOptions* class.
- *OEMakeDesignUnit* function.
- *OEMakeBioDesignUnit* function.

Constructors

```
OESpruceFilter()
OESpruceFilter(const OESpruceFilterOptions&,
               const OEMakeDesignUnitOptions&)
```

Constructor for spruce filter given the specified options class. If no *OESpruceFilterOptions* and *OEMakeDesignUnitOptions* are passed in as `fopts` and `DUOpts`, then the default options class will be passed into the constructor.

See also

- *OESpruceFilterOptions* class
- *OEMakeDesignUnitOptions* class

```
OESpruceFilter(const OESpruceFilter& rhs)
```

Copy constructor.

6.1.53 StandardizeAndFilter

```
unsigned StandardizeAndFilter(OEChem::OEMolBase& mol,
                              const OESystem::OESkewGrid& eDensGrid,
                              const OEStructureMetadata& mdata =␣
↪OEStructureMetadata())
```

Returns a code which indicates the issues that are detected for this structure

6.1.54 GetMessages

```
std::string GetMessages(const unsigned errorTypes =␣
↪OESpruceFilterIssueCodes::ALL)
```

Returns the error message according to the provided error code

6.1.55 GetAtoms

```
OESystem::OEIterBase<OEChem::OEAtomBase>* GetAtoms(const unsigned errorTypes =␣
↪OESpruceFilterIssueCodes::ALL)
```

Returns the problematic atoms that have the issues indicated by the input error code

6.1.56 OESpruceFilterOptions

Attention

This API is currently available in C++ and Python.

The *OESpruceFilterOptions* class holds options classes that are used for structural checks. Default constructing this options class, constructs all the necessary options classes, which can then be accessed and modified. This class stores the optional parameter sets for use in the *OEMakeDesignUnits* and *OEMakeDesignUnit* functions.

Constructors

```
OESpruceFilterOptions()
OESpruceFilterOptions(const OESpruceFilterOptions &)
```

Default, copy, and advanced constructor that constructs a *OESpruceFilterOptions* object.

GetDeleteFloatingRes

```
bool GetDeleteFloatingRes() const
```

Returns the boolean whether or not to delete floating residues

GetFixBackboneAtoms

```
bool GetFixBackboneAtoms() const
```

Returns the boolean whether or not to fix incorrect backbone atoms. If FixBackboneAtoms is set to true, spruce filter will delete incorrect partial backbone atoms.

GetFixBondsToMetals

```
bool GetFixBondsToMetals() const
```

Returns the boolean whether or not to fix covalent bonds to metals. If FixBondsToMetals option is set to true, spruce filter function will set bond order of the incorrect covalent bonds to metals to 0.

GetFixChainIDs

```
bool GetFixChainIDs() const
```

Returns the boolean whether or not to fix chain IDs. If FixChainIDs is set to true, spruce filter will assign incorrect chain IDs based on structural context.

GetFixInvalidCovalentBonds

```
bool GetFixInvalidCovalentBonds() const
```

Returns the boolean of whether or not to fix covalent bond structure. If FixInvalidCovalentBonds option is set to true, bonds deemed invalid will be deleted or flagged, depending on whether structural modifications are allowed.

GetFixNames

```
bool GetFixNames() const
```

Returns the boolean whether or not to fix incorrect names for all components. If FixNames is set to true, spruce filter function will rename ligands named "UNL" to "LIG" and residues named "NMA" will change to "NME" with CA atoms renamed to "CH3". Spruce filter will fail if residue has name "UNK".

GetFixResStates

```
bool GetFixResStates() const
```

Returns the boolean of whether or not to fix protein residue states

6.1.57 SetDeleteFloatingRes

```
bool SetDeleteFloatingRes(bool DeleteFloatingRes)
```

Sets the boolean of whether or not to delete floating residues

6.1.58 SetFixBackboneAtoms

```
bool SetFixBackboneAtoms(bool FixBackboneAtoms)
```

Sets the boolean of whether or not to fix incorrect backbone atoms

6.1.59 SetFixBondsToMetals

```
bool SetFixBondsToMetals(bool FixBondOrder)
```

Sets the boolean of whether or not to fix covalent bonds to metals

6.1.60 SetFixChainIDs

```
bool SetFixChainIDs(bool FixChainIDs)
```

Sets the boolean of whether or not to fix fix chainIDs

6.1.61 SetFixInvalidCovalentBonds

```
bool SetFixInvalidCovalentBonds(bool FixInvalidCovalentBonds)
```

Returns the boolean of whether or not to fix covalent bond structure

6.1.62 SetFixNames

```
bool SetFixNames(bool FixNames)
```

Sets the boolean of whether or not to fix incorrect names for all components.

6.1.63 SetFixResStates

```
bool GetFixResStates() const
```

Sets the boolean of whether or not to fix protein residue states

6.1.64 OEStructuralSuperposition

Warning

This is a deprecated API. Please use *OESuperpose* class instead.

Attention

This API is currently available in C++ and Python.

```
class OEStructuralSuperposition
```

This class represents *OEStructuralSuperposition* that performs a structural superposition on fit protein to a reference protein, using a superposition method given as an optional parameter.

See also

- *OESuperpositionOptions* class
- *OESecondaryStructureSuperposition* class

Constructors

```
OEStructuralSuperposition(const OEChem::OEMolBase& argRefMol,
                          const OEChem::OEMolBase& argFitMol,
                          const OESuperpositionOptions& argOpts =␣
↪OESuperpositionOptions())
```

Constructor that superimposes the `argFitMol` to the `argRefMol`. If no *OESuperpositionOptions* are passed in as `argOpts`, then the default behavior of using a superposition to all globally matched atoms will be used.

Both `argRefMol` and `argFitMol` must have 3D coordinates assigned.

```
OEStructuralSuperposition(const OEStructuralSuperposition& rhs)
```

Copy constructor.

operator bool

```
operator bool()
```

Returns whether the *OEStructuralSuperposition* object is valid.

ConstraintsMet

```
bool ConstraintsMet() const
```

Returns whether or not the constraints were met.

GetFitChains

```
std::vector<std::string> GetFitChains() const
```

Returns 1-letter code of all names of fit protein chains.

See also

- *OEStructuralSuperposition.GetRefChains* method

GetRMSD

```
double GetRMSD() const
```

Returns the RMSD of the structural superposition.

GetRefChains

```
std::vector<std::string> GetRefChains() const
```

Returns 1-letter code of all names of reference protein chains.

See also

- *OEStructuralSuperposition.GetFitChains* method

GetRegion

```
std::vector<int> GetRegion() const
```

Returns the index of all residues involved in the alignment.

GetRotMatrix

```
void GetRotMatrix(double* rmat) const
```

Returns by reference the rotation matrix of the superposition.

GetScore

```
int GetScore() const
```

Returns the sequence alignment score for the structural superposition.

GetTranslation

```
void GetTranslation(double* trans) const
```

Returns by reference the translation vector of the superposition.

Transform

```
void Transform(OEChem::OEMolBase& mol)
```

Transforms the input molecule using the stored rotation matrix and translation vector from the superposition.

6.1.65 OESuperpositionOptions

Attention

This API is currently available in C++ and Python.

```
class OESuperpositionOptions
```

This class represents *OESuperpositionOptions* that sets the optional parameters for structural superposition using the *OEStructuralSuperposition* class.

See also

- *OEStructuralSuperposition* class
- *OESecondaryStructureSuperposition* class

Constructors

```
OESuperpositionOptions()
```

Default constructor.

```
OESuperpositionOptions(const OESuperpositionOptions& rhs)
```

Copy constructor.

AddSiteResidue

```
void AddSiteResidue(const std::string& argSiteResidue)
```

Add a single site residue for *OESuperpositionType.Site* -based superpositions based on a site-specific string that takes the form: `<residue name>:<residue number>:<insertion code>:<chain id>`.

See also

- *OESuperpositionOptions.SetSiteResidues* method
- *OESuperpositionOptions.GetSiteResidues* method
- *OESuperpositionOptions.ClearSiteResidues* method

ClearLigandConstraints

```
void ClearLigandConstraints()
```

Clear the stored ligand constraints, if they exist.

See also

- *OESuperpositionOptions.SetLigandConstraints* method
- *OESuperpositionOptions.GetLigandConstraints* method
- *OESuperpositionOptions.HasLigandConstraints* method

ClearSiteResidues

```
void ClearSiteResidues()
```

Remove any existing site residues from the *OESuperpositionOptions* object that would be used in the *OESuperpositionType.Site* -based superposition calculations.

See also

- *OESuperpositionOptions.AddSiteResidue* method
- *OESuperpositionOptions.SetSiteResidues* method
- *OESuperpositionOptions.GetSiteResidues* method

GetDist

```
double GetDist() const
```

Get the cutoff distance used in the nearest neighbors calculation for the site residue constraints (default = `5.0 Angstroms`) for *OESuperpositionType.Site* -based superpositions.

GetLigandConstraints

```
void GetLigandConstraints(OEChem::OEMolBase& refLig,
                          OEChem::OEMolBase& fitLig) const
```

Returns by reference the reference and fit ligand constraints that are stored.

See also

- *OESuperpositionOptions.SetLigandConstraints* method
- *OESuperpositionOptions.HasLigandConstraints* method
- *OESuperpositionOptions.ClearLigandConstraints* method

GetSiteResidues

```
std::vector<std::string> GetSiteResidues() const
```

Get the site residues used in the *OESuperpositionType.Site* -based calculation.

GetSuperpositionType

```
unsigned GetSuperpositionType() const
```

Get the type of superposition to be performed. Types are given from the constants given in *OESuperpositionType*. The default type is *OESuperpositionType.Default*.

HasLigandConstraints

```
bool HasLigandConstraints() const
```

Returns whether or not ligand constraints have been set.

See also

- *OESuperpositionOptions.SetLigandConstraints* method
- *OESuperpositionOptions.GetLigandConstraints* method
- *OESuperpositionOptions.ClearLigandConstraints* method

SetDist

```
void SetDist(double argDist)
```

Set the cutoff distance used in the nearest neighbors calculation for the site residue constraints for *OESuperpositionType.Site* -based superpositions.

SetLigandConstraints

```
void SetLigandConstraints(const OEChem::OEMolBase& refLig,
                          const OEChem::OEMolBase& fitLig)
```

Sets the ligand constraints that will be used for the superposition.

See also

- *OESuperpositionOptions.GetLigandConstraints* method
- *OESuperpositionOptions.HasLigandConstraints* method
- *OESuperpositionOptions.ClearLigandConstraints* method

SetSiteResidues

```
void SetSiteResidues(const std::vector<std::string>& argSiteResidues)
```

Set all site residues at once for *OESuperpositionType.Site* -based superpositions using a string vector, where each string takes the form `<residue name>:<residue number>:<insertion code>:<chain id>`.

See also

- *OESuperpositionOptions.AddSiteResidue* method
- *OESuperpositionOptions.GetSiteResidues* method
- *OESuperpositionOptions.ClearSiteResidues* method

SetSuperpositionType

```
bool SetSuperpositionType(unsigned type)
```

Set the type of superposition to be performed. Types are taken from the constants given in *OESuperpositionType*.

6.1.66 OESuperpose

Attention

This API is currently available in C++ and Python.

```
class OESuperpose
```

This class performs a structural superposition of a fit protein onto a reference protein. The superposition method (see *OESuperposeMethod*) and other options are controlled through the *OESuperposeOptions* class. Results including RMSD and TM-score are returned via *OESuperposeResults*.

See also

- *OESuperposeOptions* class

- *OESuperposeResults* class

Code Example

- *Calculating superposition* example

Constructors

```
OESuperpose(const OESuperposeOptions& opts =␣
↪OESuperposeOptions(OESuperposeMethod::Default))
```

Constructor for superposing molecules given the specified method and other options. If no *OESuperposeOptions* are passed in as `opts`, then the default behavior of using a superposition to all globally matched atoms will be used.

```
OESuperpose(const OESuperpose& rhs)
```

Copy constructor.

HasRef

```
bool HasRef() const
```

Returns whether or not the reference is set.

HasConstraintMol

```
bool HasConstraintMol() const
```

Returns whether or not the constraint molecule is set.

Clear

```
void Clear()
```

Clears the reference and the constraint molecule.

SetOptions

```
bool SetOptions(const OESuperposeOptions& opts)
```

Sets the options for the superposition.

SetupRef

```
bool SetupRef(const OEBio::OEDesignUnit& refDU,
              const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& refPred =␣
↪OEChem::OEIsTrueAtom())
bool SetupRef(const OEChem::OEMolBase& refMol,
              const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& refPred =␣
```

(continues on next page)

(continued from previous page)

```
↪OEChem::OEIsTrueAtom())
bool SetupRef(const OEChem::OEMolBase& refMol,
              const OEChem::OEMolBase& refConstraintMol,
              const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& refPred =␣
↪OEChem::OEIsTrueAtom())
bool SetupRef(const OEChem::OEMolBase& refMol,
              const std::vector<std::string>& siteResidues,
              const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& refPred =␣
↪OEChem::OEIsTrueAtom())
bool SetupRef(const OEChem::OEMolBase& refMol,
              const OEChem::OEMolBase& refConstraintMol,
              const std::vector<std::string>& siteResidues,
              const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& refPred =␣
↪OEChem::OEIsTrueAtom())
```

Sets up the reference molecule, constraint molecule, or site residues if given. A predicate (`refPred`) can be passed to subset the reference. If the reference is given as an OEDesignUnit object, then the constraint molecule (ligand) and the site residues will be set if such information is available in the design unit. The function returns true if the setup is successful.

The `refMol` argument must have 3D coordinates assigned.

SetMethod

```
bool SetMethod(unsigned value)
```

Sets the type of superposition to be performed. Types are taken from the constants given in *OESuperposeMethod*.

> **Note**
>
> This switches the method used during superposition, but not the default, user defined, valid RMSD, seqScore, or Tanimoto values. This means switching between a shape and sequence based method requires the user to explicitly set proper values for these or, alternatively, reconstruct the class with a new method.

Superpose

```
bool Superpose(OESuperposeResults& results,
               const OEChem::OEMolBase& fitMol,
               const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& fitPred =␣
↪OEChem::OEIsTrueAtom()) const
bool Superpose(OESuperposeResults& results,
               const OEChem::OEMolBase& fitMol,
               const OEChem::OEMolBase& fitConstraintMol,
               const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& fitPred =␣
↪OEChem::OEIsTrueAtom()) const
bool Superpose(OESuperposeResults& results,
               const OEBio::OEDesignUnit& fitDU,
               const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& fitPred =␣
↪OEChem::OEIsTrueAtom()) const
```

Superimposes the fit molecule onto the reference. A predicate (`fitPred`) can be passed to subset the reference. The results for the superposition will be stored in `results`. The function returns `results.IsValid()`.

6.1.67 OESuperposeOptions

Attention

This API is currently available in C++ and Python.

```
class OESuperposeOptions
```

This class represents *OESuperposeOptions* that sets the optional parameters for structural superposition using the *OESuperpose* class.

See also

- *OESuperpose* class
- *OEStructuralSuperposition* class
- *OESecondaryStructureSuperposition* class

Code Example

- *Calculating superposition* example

Constructors

```
OESuperposeOptions(const unsigned method = OESuperposeMethod::Default)
```

Default constructor.

```
OESuperposeOptions(const OESuperposeOptions& rhs)
```

Copy constructor.

6.1.68 GetMethod

```
unsigned GetMethod() const
```

Returns the type of superposition to be performed. Types are given from the constants given in *OESuperposeMethod*. The default type is *Default*.

6.1.69 SetMethod

```
bool SetMethod(unsigned value)
```

Sets the type of superposition to be performed. Types are taken from the constants given in *OESuperposeMethod*.

6.1.70 GetUseConstraints

```
bool GetUseConstraints() const
```

Returns whether or not ligand or site-residue constraints should be used to validate the superposition.

See also

- *SetUseConstraints* method
- *SetupRef* method
- *Superpose* method

6.1.71 SetUseConstraints

```
bool SetUseConstraints(bool value)
```

Sets whether or not ligand or site-residue constraints should be used to validate the superposition.

See also

- *GetUseConstraints* method
- *SetupRef* method
- *Superpose* method

6.1.72 GetValidRMSD

```
double GetValidRMSD() const
```

Returns the RMSD threshold for validating the superposition of the reference and fit molecules. *OESuperposeResults.IsValid* returns false if the RMSD of the superposed molecules exceeds this threshold. The default value is 10.0 Angstroms for sequence-based methods and -1.0 for shape-based methods (RMSD calculation is only applicable to sequence-based methods).

See also

- *SetValidRMSD* method
- *IsValid* method
- *GetRMSD* method

6.1.73 SetValidRMSD

```
bool SetValidRMSD(double value)
```

Sets the RMSD threshold for validating the superposition of the reference and fit molecules.

Warning

RMSD calculation is only applicable to sequence-based methods. Please avoid using this function when running shape-based superposition.

See also

- *SetValidRMSD* method

6.1.74 GetValidTanimoto

```
double GetValidTanimoto() const
```

Returns the Tanimoto threshold for validating the superposition of the reference and fit molecules. *OESuperposeResults.IsValid* returns false if the Tanimoto of the superposed molecules is smaller than this threshold. The default value is 0.0 for shape-based methods and -1.0 for sequence-based methods (Tanimoto calculation is only applicable to shape-based methods).

See also

- *SetValidTanimoto* method
- *IsValid* method
- *GetTanimoto* method

6.1.75 SetValidTanimoto

```
bool SetValidTanimoto(double value)
```

Sets the Tanimoto threshold for validating the superposition of the reference and fit molecules.

Warning

Tanimoto calculation is only applicable to shape-based methods. Please avoid using this function when running sequence-based superposition.

See also

- *GetValidRMSD* method

6.1.76 GetValidSeqScore

```
int GetValidSeqScore() const
```

Returns the sequence alignment score threshold for validating the superposition of the reference and fit molecules. *OESuperposeResults.IsValid* returns false if the sequence alignment score of the super-

posed molecules is smaller than this threshold. The default value is 200 for sequence-based methods and -1000 for shape-based methods (sequence alignment score is only applicable to sequence-based methods).

See also

- *SetValidSeqScore* method
- *IsValid* method
- *GetSeqScore* method

6.1.77 SetValidSeqScore

```
bool SetValidSeqScore(int value)
```

Sets the sequence alignment score threshold for validating the superposition of the reference and fit molecules.

Warning

sequence alignment score is only applicable to sequence-based methods. Please avoid using this function when running shape-based superposition.

See also

- *GetValidRMSD* method

6.1.78 GetUseSeqScoreChainMatching

```
bool GetUseSeqScoreChainMatching() const
```

Returns whether or not combined sequence alignment score should be used to match chains for multi-chain complexes.

See also

- *SetUseSeqScoreChainMatching* method

6.1.79 SetUseSeqScoreChainMatching

```
bool SetUseSeqScoreChainMatching(bool value)
```

Sets whether or not combined sequence alignment score should be used to match chains for multi-chain complexes.

See also

- *GetUseSeqScoreChainMatching* method

6.1.80 OESuperposeResults

Attention

This API is currently available in C++ and Python.

```
class OESuperposeResults
```

This class stores superposition results generated by *OESuperpose*.

See also

- *OESuperpose* class

Code Example

- *Calculating superposition* example

Constructors

```
OESuperposeResults()
```

Default constructor.

```
OESuperposeResults(const OESuperposeResults& rhs)
```

Copy constructor.

```
OESuperposeResults(OESuperposeResults&&)
```

Move constructor.

operator bool

```
operator bool() const
```

Returns true if the superposition is valid and false if not, which can be also obtained by calling *IsValid*.

See also

- *IsValid* method

IsValid

```
bool IsValid() const
```

Returns true if the superposition is valid and false if not. The superposition is valid if the scores (RMSD, Tanimoto, and sequence alignment score) satisfy the thresholds set in the *OESuperposeOptions* class.

See also

- *GetValidRMSD* method
- *GetValidTanimoto* method
- *GetValidSeqScore* method

Transform

```
bool Transform(OEChem::OEMolBase& mol) const
bool Transform(OEBio::OEDesignUnit& du) const
```

Returns true and transforms the input molecule using the stored transform from the superposition if they are valid. Otherwise, the function does nothing and returns false.

See also

- OETrans class
- *GetTransform* method

GetTransform

```
OEChem::OETrans& GetTransform() const
```

Returns the OETrans object that stores the rotation matrix and translation vector from the superposition.

See also

- OETrans class
- *Transform* method
- *GetRotMatrix* method
- *GetTranslation* method

GetRotMatrix

```
bool GetRotMatrix(double* rmat) const
```

Returns by reference the rotation matrix of the superposition.

GetTranslation

```
bool GetTranslation(double* trans) const
```

Returns by reference the translation vector of the superposition.

HasRMSD

```
bool HasRMSD() const
```

Returns whether RMSD was calculated for the superposed molecules. RMSD is only applicable to sequence-based methods, such as *Global*, *Site*, *DDM*, and *Weighted*.

HasTanimoto

```
bool HasTanimoto() const
```

Returns whether Tanimoto was calculated for the superposed molecules. Tanimoto is only applicable to shape-based methods, such as *SSE*.

HasSeqScore

```
bool HasSeqScore() const
```

Returns whether sequence alignment score was calculated for the superposed molecules. sequence alignment score is only applicable to sequence-based methods, such as *Global*, *Site*, *DDM*, and *Weighted*.

GetRMSD

```
double GetRMSD() const
```

Returns the RMSD of the structural superposition.

GetTanimoto

```
double GetTanimoto() const
```

Returns the Tanimoto of the structural superposition.

GetSeqScore

```
int GetSeqScore() const
```

Returns the sequence alignment score of the structural superposition. The score is calculated by summing the similarity scores of matched residues and gap penalties.

GetRefChains

```
const std::vector<std::string>& GetRefChains() const
```

Returns 1-letter code of all names of reference protein chains.

See also

- *GetFitChains* method

GetFitChains

```
const std::vector<std::string>& GetFitChains() const
```

Returns 1-letter code of all names of fit protein chains.

See also

- *GetRefChains* method

6.1.81 OEValidateDesignUnit

Attention

This API is currently available in C++ and Python.

```
class OEValidateDesignUnit
```

This class performs a structural check after running Spruce preparation.

It looks for:

- Partial side chains
- Partial charges
- Implicit hydrogens
- Incorrect terminal residues
- Heavy atom overlap
- Broken chains

OEDesignUnitIssueCodes shows the type of issue codes.

Code Example

- *Validate Structure of the Design Unit*

Constructors

```
OEValidateDesignUnit()
OEValidateDesignUnit(const OEValidateDesignUnit& rhs)
```

Default and copy constructors for the OEValidateDesignUnit class.

Validate

```
unsigned Validate(OEBio::OEDesignUnit& du,
                  const OESpruce::OEStructureMetadata& mdata =␣
↪OESpruce::OEStructureMetadata())
```

Returns a code which indicates the issues that are detected for this structure.

GetMessages

```
std::string GetMessages(const unsigned errorTypes = OEDesignUnitIssueCodes::ALL)
```

Returns the error message according to the provided error code.

GetProblematicResidues

```
std::vector<OEChem::OEResidue> GetProblematicResidues(const unsigned errorTypes =␣
↪OEDesignUnitIssueCodes::ALL)
```

Returns the problematic residues that have the issues indicated by the input error code.

6.2 OESpruce Constants

6.2.1 OEAlternateLocationOption

> **Attention**
>
> This API is currently available in C++ and Python.

This namespace contains methods to enumerate the alternate locations of a biomolecular structure.

Combinatorial

Make all possible combinations of enumerated alternate locations.

Default

Enumerate is the default option.

Enumerate

Enumerate through all possible alternate locations.

Primary

Use only the primary (highest occupancy) alternate location.

Unknown

The alternate location method has not been set.

Max

6.2.2 OEDesignUnitIssueCodes

Attention

This API is currently available in C++ and Python.

This namespace contains constants for design unit validation issue codes, used by the *OEValidateDesignUnit* class.

See also

- The *OEMakeDesignUnitOptions* class.

Success

This issue code indicates that the design unit does not have errors.

PartialSideChain

This issue code indicates that the design unit contains partial side chains.

MissingLoop

This issue code indicates that the input design unit contains missing loops.

ImplicitH

This issue code indicates that the input design unit contains implicit hydrogens.

TerminiNotCapped

This issue code indicates that the input design unit contains incorrect terminal residues.

HeavyAtomsOverlap

This issue code indicates that the design unit has heavy atoms overlapping with each other.

BrokenChains

This issue code indicates the design unit contains broken chains.

All

This issue code indicates that the design unit contains all the issues listed above.

6.2.3 OEExperimentType

> **Attention**
>
> This API is currently available in C++ and Python.

This namespace contains constants that describe the type of experiment from which the 3-D structure was determined.

ElectronCrystallography

The structure was determined by electron crystallography.

ElectronMicroscopy

The structure was determined by electron cryo-microscopy.

FiberDiffraction

The structure was determined by fiber diffraction using either X-rays, electrons, or neutrons.

NeutronDiffraction

The structure was determined by elastic neutron scattering.

PowderDiffraction

The structure was determined by powder diffraction using either X-rays, electrons, or neutrons.

SolidStateNMR

The structure was determined by NMR methods in the solid state.

SolutionNMR

The structure was determined by NMR methods in the liquid state.

SolutionScattering

The structure was determined by small angle X-ray scattering (SAXS) in solution.

TheoreticalModel

The structure was predicted/computed using a theoretical/computational model.

Unknown

The experiment type is not known.

XRayDiffraction

The structure was determined using X-ray diffraction.

6.2.4 OEHeterogenType

> **Attention**
>
> This API is currently available in C++ and Python.

This namespace contains constants that define the heterogen molecule type used in the *OEHeterogenMetadata* class.

Cofactors

The heterogen type are cofactors.

CounterIons

The heterogen type are counter ions.

Excipients

The heterogen type are excipients.

Ligand

The heterogen type is a ligand.

Lipids

The heterogen types are lipids.

Metals

The heterogen type are metals.

Polymers

The heterogen type are polymeric molecules.

Solvent

The heterogen type are solvent molecules.

Sugars

The heterogen type are sugars.

Unknown

The heterogen type are not known.

6.2.5 OESuperpositionType

Attention

This API is currently available in C++ and Python.

This namespace contains constants for superpositioning methods.

See also

- *OESuperpositionOptions* class

Default

The default superpositioning method is *OESuperpositionType.Global*.

DDM

Calculate superposition using the **D**istance **D**ifference **M**atrix method.

Global

Calculate superposition using the all matched atoms.

Site

Calculate superposition using the input site residues. Site residues must be set with the *OESuperpositionOptions.SetSiteResidues* member function of the *OESuperpositionOptions* class.

SSE

Calculate superposition using an overlap of **S**econdary **S**tructure **E**lements.

See also

- *OESecondaryStructureSuperposition* class

Undefined

Superposition method is undefined. No superposition will take place.

Weighted

Calculate a weighted superposition using weights taken from the Difference Distance Matrix.

6.2.6 OESpruceFilterIssueCodes

Attention

This API is currently available in C++ and Python.

This namespace contains constants for Spruce filter issue codes.

See also

- *OESpruceFilterOptions* class

Success

This issue code indicates that the input structure does not have errors and passes the spruce filter

InvalidNames

This issue code indicates that the input structure contains invalid names.

InvalidResidueStates

This issue code indicates that the input structure contains residues at invalid state.

InvalidMetalBond

This issue code indicates that the input structure contains invalid covalent bonds to metals.

InvalidCRYST1

This issue code indicates that the input structure does not have a valid CRYST1 header.

InvalidElectronDensityOverlap

This issue code indicates that the provided electron density map does not overlap with the input structure

InvalidSeqAlign

This issue code indicates that the sequence does not align with the input structure.

All

This issue code indicates that the structure contains all the issues listed above.

6.2.7 OESuperposeMethod

Attention

This API is currently available in C++ and Python.

This namespace contains constants for superpositioning methods.

See also

- *OESuperposeOptions* class

Code Example

- *Calculating superposition* example

Default

The default superpositioning method is GlobalCarbonAlpha.

GlobalCarbonAlpha

Calculate superposition using the all matched alpha carbon atoms.

Global

Alias for `OESuperposeMethod.GlobalCarbonAlpha`.

SiteSequence

Calculate superposition using the input site residues. Site residues must be set with the `OESuperpose.SetupRef` member function of the *OESuperpose* class.

Site

Alias for SiteSequence.

DifferenceDistanceMatrix

Calculate superposition using the **D**istance **D**ifference **M**atrix method. See *Protein Superposition* for more information.

DDM

Alias for DifferenceDistanceMatrix.

WeightedDifferenceDistanceMatrix

Calculate a weighted superposition using weights taken from the Difference Distance Matrix. See *Protein Superposition* for more information.

Weighted

Alias for WeightedDifferenceDistanceMatrix.

SecondaryStructureElements

Calculate superposition using an overlap of **S**econdary **S**tructure **E**lements.

See also

- *OESecondaryStructureSuperposition* class

SSE

Alias for SecondaryStructureElements.

SiteHopper

Calculates the superposition using SiteHopper

See also

- Introduction

Undefined

Superposition method is undefined. No superposition will take place.

6.3 OESpruce Functions

6.3.1 OEBuildLoops

Attention

This API is currently available in C++ and Python.

These functions build missing loops and tails in protein structures. They return false if any of the loops or tails cannot be built, but it will return with as many built loops and tails as possible. The functions require a loop database be provided in the options class. Using the **loopdb_builder** application it is possible to build your own database for a given target family, or a set of proprietary structure. OpenEye provides a database built of the structures in the entire PDB repository at the time of toolkit release. The datastamps are noted in the info table and logged during loop building. It is possible to append local/proprietary structures to this database if desired, again using the desktop application. The options class allows different behaviors for the loop builder.

Note

Explicit hydrogens are not added to the structure during the build processes, therefore no partial charges are assigned.

```
bool OEBuildLoops(OEChem::OEMolBase& mol,
                  const OESidechainBuilderOptions& scOpts = OESidechainBuilderOptions(),
                  const OELoopBuilderOptions& lOpts = OELoopBuilderOptions())
```

This function takes a molecule and builds any missing loops that can be detected based on the information in the PDB header metadata. The function also takes options classes for the loop building, but also for the side-chain re-building, in case the loop in the database is not a 100% identity match. The function calls OEOmega to generate possible conformations for tails.

```
bool OEBuildLoops(OEChem::OEMolBase& mol,
                  const OEStructureMetadata& data,
                  const OESidechainBuilderOptions& scOpts = OESidechainBuilderOptions(),
                  const OELoopBuilderOptions& lOpts = OELoopBuilderOptions())
```

This function works identically to the above function, only metadata is explicitly provided containing information about the sequence, so gaps can be detected.

The input `mol` must have 3D coordinates assigned.

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- *OEDesignUnitBuildOptions* class
- *OELoopBuilderOptions* class
- *OESidechainBuilderOptions* class
- `OERotamerLibrary` namespace
- *OEStructureMetadata* class
- *`OEBuildSingleLoop`* function

6.3.2 OEBuildSidechains

Attention

This API is currently available in C++ and Python.

These functions build out partial sidechains using a rotamer library. The rotamer chosen is the one with the best interaction energy with the surrounding environment without taking solvent effects into account. If multiple sidechains are built that interact with one another, the energy chosen is the best collective interaction energy. The functions will return false if a sidechain can not be built due to clashes, but will still return a structure with all the ones it could build.

Note

Explicit hydrogens are not added to the structure during the build processes, therefore no partial charges are assigned.

```
bool OEBuildSidechains(OEChem::OEMolBase& mol,
                       const OESpruce::OESidechainBuilderOptions& opts =␣
↪OESpruce::OESidechainBuilderOptions())
```

This function takes a molecule and builds any partial sidechain of standard protein residues determined using the *OEGetPartialResidues* function. An options class defining a rotamer library from the `OERotamerLibrary` namespace can be set, otherwise the default library from that namespace is chosen. Additionally, in the options the number of rotamers used can be reduced by picking a rotamer coverage below the default 100%. The percentage is calculated cumulatively, so the most probably rotamer is tested first, up to the rotamer that takes it above the percentage limit. The options class also defines the behavior of removing water molecules that clash with the built sidechain.

```
bool OEBuildSidechains(OEChem::OEMolBase& mol,
                       const std::vector<OEChem::OEResidue>& residues,
                       const OESpruce::OESidechainBuilderOptions& opts =␣
↪OESpruce::OESidechainBuilderOptions())
```

This function works identically to the above function, but only builds the residues identified in the vector residues.

The input `mol` must have 3D coordinates assigned.

See also

- *OESidechainBuilderOptions* class
- `OERotamerLibrary` namespace
- *OEGetPartialResidues* function

6.3.3 OEBuildSingleLoop

Attention

This API is currently available in C++ and Python.

This function builds a specified missing loop or tail in a protein structures. The function returns false if the loop or tail cannot be built. The function requires a loop database be provided in the options class. Using the **loopdb_builder** application it is possible to build your own database for a given target family, or a set of proprietary structure. OpenEye provides a database built of the structures in the entire PDB repository at the time of toolkit release. The datastamps are noted in the info table and logged during loop building. It is possible to append local/proprietary structures to this database if desired, again using the desktop application. The options classes allows different behaviors for the loop builder.

Note

Explicit hydrogens are not added to the structure during the build processes, therefore no partial charges are assigned.

```
bool OEBuildSingleLoop(OEChem::OEMCMolBase& outputMol,
                       const OEChem::OEMolBase& inputMol,
                       const std::string fasta,
                       const OEChem::OEResidue& cTerminalAttachment,
                       const OEChem::OEResidue& nTerminalAttachment,
                       const OESidechainBuilderOptions& scOpts =␣
↪OESidechainBuilderOptions(),
                       const OELoopBuilderOptions& lOpts = OELoopBuilderOptions())
```

The function takes an empty multi-conformer molecule which will contain the protein with the loop built if the function is successful. Multiple conformers of the loop will be returned based on the passed options, the results are sorted by a force field energy score. The remaining arguments are, the molecule with the gap, the loop protein sequence in 1-letter codes as a string (e.g. "PVHTAL"), and the c- and n-terminal attachment (or anchor) residues. The function also takes options classes for the loop building, and side-chain re-building, in case the loop in the database is not a 100% identity match. The function calls OEOmega to generate possible conformations for tails.

The input `inputMol` must have 3D coordinates assigned.

See also

- *OEMakeDesignUnitOptions* class
- *OEDesignUnitPrepOptions* class
- *OEDesignUnitBuildOptions* class
- *OELoopBuilderOptions* class
- *OESidechainBuilderOptions* class
- `OERotamerLibrary` namespace
- *OEStructureMetadata* class
- *`OEBuildLoops`* function

6.3.4 OECapCTermini

> **Attention**
>
> This API is currently available in C++ and Python.

```
bool OECapCTermini(OEChem::OEMolBase& mol,
                   const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& exclude =␣
↪OEChem::OEIsFalseAtom(),
                   const OESpruce::OECapBuilderOptions& =␣
↪OESpruce::OECapBuilderOptions())
```

This function caps all the C-termini in a protein with an N-methyl (`OEResidueIndex.NME`) group. The N-methyl is built assuming ideal backbone geometry.

A predicate can be passed to exclude certain termini from being capped, such as the true biological C-terminal residue since this should be charged to match the biophysical experiments.

The function takes an options class, where behavior can be controlled if building the cap results in a clash, and depending on the type of class what actions can be taken, e.g. a solvent molecule blocking the protein backbone is not very realistic and the solvent molecule is likely placed to account for density that should have belonged to the residue not resolved in the crystal structure.

The cap will be numbered identically to the residue number of the terminal residue, but the insertion code will be incremented. The reason for this is to avoid residue number clashes with heterogens following the protein chain in the PDB file.

> **Note**
>
> If the backbone of the residue being capped is incomplete, or a cap can not be built because of a clash, the residue being capped can be turned into the cap by removing atoms in the residue. A clash can either be into a non-solvent or into a solvent with the *deleteClashingSolvent* option set to false.

The input `mol` must have 3D coordinates assigned.

> **See also**
>
> - *OECapBuilderOptions* class
> - *OECapTermini* function
> - *OECapNTermini* function
> - OEIsCTerminalAtom predicate

6.3.5 OECapNTermini

Attention

This API is currently available in C++ and Python.

```
bool OECapNTermini(OEChem::OEMolBase& mol,
                   const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& exclude =␣
↪OEChem::OEIsFalseAtom(),
                   const OESpruce::OECapBuilderOptions& =␣
↪OESpruce::OECapBuilderOptions())
```

This function caps all the N-termini in a protein with an acetyl (`OEResidueIndex.ACE`) group. The acetyl is built assuming ideal backbone geometry, the ϕ value of the residue being capped is set to match the ϕ value of the previous residue (if it's a single residue in space, a value of $-90.0°$ degrees, is set which is in the allowed range for both an alpha helix and beta sheet secondary structure).

A predicate can be passed to exclude certain termini from being capped, such as the true biological N-terminal residue since this should be charged to match the biophysical experiments.

The function takes an options class, where behavior can be controlled if building the cap results in a clash, and depending on the type of class what actions can be taken, e.g. a solvent molecule blocking the protein backbone is not very realistic and the solvent molecule is likely placed to account for density that should have belonged to the residue not resolved in the crystal structure.

The cap will be numbered with -1 from the residue number of the terminal residue.

Note

If the backbone of the residue being capped is incomplete, or a cap can not be built because of a clash, the residue being capped can be turned into the cap by removing atoms in the residue. A clash can either be into a non-solvent or into a solvent with the *deleteClashingSolvent* option set to false.

The input `mol` must have 3D coordinates assigned.

See also

- *OECapBuilderOptions* class
- *OECapTermini* function
- *OECapCTermini* function
- OEIsNTerminalAtom predicate

6.3.6 OECapTermini

Attention

This API is currently available in C++ and Python.

```
bool OECapTermini(OEChem::OEMolBase& mol,
                  const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& exclude =␣
```

(continues on next page)

(continued from previous page)

```
↪OEChem::OEIsFalseAtom(),
                  const OESpruce::OECapBuilderOptions& = OESpruce::OECapBuilderOptions())
```

This function caps all the N- and C-termini in a protein with acetyl (`OEResidueIndex.ACE`) and N-methyl (`OEResidueIndex.NME`) groups, respectively.

A predicate can be passed to exclude certain termini from being capped, such as the true biological terminal residues since these should be charged to match the biophysical experiments.

The function takes an options class, where behavior can be controlled if building the cap results in a clash, and depending on the type of class what actions can be taken, e.g. a solvent molecule blocking the protein backbone is not very realistic and the solvent molecule is likely placed to account for density that should have belonged to the residue not resolved in the crystal structure.

Note

Explicit hydrogens are not added to the structure during the build processes.

The input `mol` must have 3D coordinates assigned.

See also

- *OECapBuilderOptions* class
- *OECapCTermini* function
- *OECapNTermini* function
- OEIsNTerminalAtom predicate
- OEIsCTerminalAtom predicate

6.3.7 OEEnumerateSites

Attention

This API is currently available in C++ and Python.

```
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEEnumerateSites(const OEBio::OEDesignUnit&␣
↪bioDesignUnit,
                                                            const␣
↪OEDesignUnitEnumerateSitesOptions& opts = OEDesignUnitEnumerateSitesOptions(),
                                                            const std::string␣
↪siteResidue = "")
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEEnumerateSites(const OEBio::OEDesignUnit&␣
↪bioDesignUnit,
                                                            const OESystem::OESkewGrid&␣
↪electronDensity,
                                                            const␣
↪OESpruce::OEDesignUnitEnumerateSitesOptions& opts =␣
↪OESpruce::OEDesignUnitEnumerateSitesOptions(),
```

(continues on next page)

(continued from previous page)

```
                                                                const std::string␣
↪siteResidue = "")
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEEnumerateSites(const OEBio::OEDesignUnit&␣
↪bioDesignUnit,
                                                                const OEBio::OEDesignUnit&␣
↪designUnitReference,
                                                                const␣
↪OESpruce::OEDesignUnitEnumerateSitesOptions& opts =␣
↪OESpruce::OEDesignUnitEnumerateSitesOptions())
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEEnumerateSites(const OEBio::OEDesignUnit&␣
↪bioDesignUnit,
                                                                const OEBio::OEDesignUnit&␣
↪designUnitReference,
                                                                const OESystem::OESkewGrid&␣
↪electronDensity,
                                                                const␣
↪OESpruce::OEDesignUnitEnumerateSitesOptions& opts =␣
↪OESpruce::OEDesignUnitEnumerateSitesOptions())
```

Return an iterator over all OEDesignUnit objects enumerated from the input *bioDesignUnit* argument based on the componentization of the internal heterogen molecules. The *bioDesignUnit* input argument should be taken from the iterator output of *OEMakeBioDesignUnits*.

6.3.8 OEExtractBioUnits

Attention

This API is currently available in C++ and Python.

```
OESystem::OEIterBase<OEChem::OEMolBase>*OEExtractBioUnits(const OEChem::OEMolBase&␣
↪extractProtein,
                                                          const OEChem::OEMolBase&␣
↪refProtein,
                                                          const␣
↪OEBioUnitExtractionOptions& opts = OEBioUnitExtractionOptions())
OESystem::OEIterBase<OEChem::OEMolBase>*OEExtractBioUnits(const OEChem::OEMolBase&␣
↪extractProtein,
                                                          const␣
↪OEBioUnitExtractionOptions& opts = OEBioUnitExtractionOptions())
```

Extracts BUs using either the PDB remarks when a single OEMolBase is provided, or a sequence alignment to a reference if two `OEMolBases` are provided. The *OEExtractBioUnits* is overloaded with to take an *OEBioUnitExtractionOptions* options class. This function returns an iterator of OEMolBase objects, one for each of the BUs.

Note

If atoms are on the symmetry axis causing them to be duplicated by symmetry expansion, they are handled specially to avoid clashes in the output. If the duplication results in atoms of a molecule that are perfectly overlapping, e.g. a water molecule oxygen, then the duplicated water is deleted. If the overlap is not complete, but instead creates partially overlapping molecules, then the BU is enumerated into multiple BUs. The nearly identical BUs are tagged

in generic data with an index number used by Spruce.

Parameters :

extractProtein
The protein from which the BUs will be extracted.

refProtein
The reference protein that will be used for sequence alignment.

OEBioUnitExtractionOptions
OEBioUnitExtractionOptions options class defining biological unit extraction options.

The `extractProtein` structure must have crystal symmetry set (e.g. loaded from a CRYST1 record) for biounit extraction to succeed.

Code Example

- *Extracting Biounits by aligning to a reference protein* example
- *Extracting Biounits from PDB header remarks* example

6.3.9 OEFindPockets

Attention

This API is currently available in C++ and Python.

```
OESystem::OEIterBase<OESpruce::OEPocket>* OEFindPockets(OEBio::OEDesignUnit& designUnit,
                                                        const unsigned componentMask =␣
↪OEBio::OEDesignUnitComponents::MacroMolComponents,
                                                        const OEPocketOptions& opts =␣
↪OESpruce::OEPocketOptions(),
                                                        const OESystem::OEUnaryPredicate
↪<OEChem::OEAtomBase>& restrictPred = OESystem::OEIsTrue<OEChem::OEAtomBase>())
OESystem::OEIterBase<OESpruce::OEPocket>* OEFindPockets(OEChem::OEMolBase& mol,
                                                        const OEPocketOptions& opts =␣
↪OESpruce::OEPocketOptions(),
                                                        const OESystem::OEUnaryPredicate
↪<OEChem::OEAtomBase>& restrictPred = OESystem::OEIsTrue<OEChem::OEAtomBase>())
```

Return an iterator over all *OEPocket* objects that can be found from the input OEMolBase or OEDesignUnit input structure. For design units, a mask can be supplied to determine, which of the internal components should be used for the pocket detection. The options input defines some parameters of the output pockets. If a specific pocket is required an atom based predicate can be supplied, the requirement is then that the one of the atom(s) in the predicate, is associated with the surface of the discovered pocket.

The input structure must have 3D coordinates assigned.

See also

- *OEPocketOptions* class
- *OEPocket* class
- *OEMakeDesignUnitFromPocket* function

6.3.10 OEFixBackbone

Attention

This API is currently available in C++ and Python.

```
bool OEFixBackbone(OEBio::OEDesignUnit& du)
```

The function attempts to correct missing oxygen atoms on the protein backbone, O and OXT, as well as correct the charge and valence state. If any of the core backbone atoms, N, CA, or C are missing it removes the residue. The proper residue can be modeled in later using *OEBuildLoops*.

6.3.11 OEGetAlternateLocationOptionID

Attention

This API is currently available in C++ and Python.

```
unsigned OEGetAlternateLocationOptionID(std::string optionName)
```

Returns the unsigned integer version of the string option name from the *OEAlternateLocationOption* namespace.

6.3.12 OEGetAlternateLocationOptionName

Attention

This API is currently available in C++ and Python.

```
std::string OEGetAlternateLocationOptionName(unsigned option)
```

Returns the string version of the option name from the *OEAlternateLocationOption* namespace.

6.3.13 OEGetExperimentTypeID

> **Attention**
>
> This API is currently available in C++ and Python.

```
unsigned OEGetExperimentTypeID(const std::string& expTypeName)
```

This function returns an unsigned integer version of the experiment type from the string argument *expTypeName*. The experiment types are contained in the *OEExperimentType* namespace.

6.3.14 OEGetExperimentTypeName

> **Attention**
>
> This API is currently available in C++ and Python.

```
std::string OEGetExperimentTypeName(unsigned expTypeID)
```

This function returns a string version of the experiment type from the unsigned integer argument *expTypeID*. The experiment types are contained in the *OEExperimentType* namespace.

6.3.15 OEGetHeterogenTypeID

> **Attention**
>
> This API is currently available in C++ and Python.

```
unsigned OEGetHeterogenTypeID(const std::string& typeName)
```

This function returns an unsigned integer version of the heterogen type from the string argument *typeName*. The heterogen types are contained in the *OEHeterogenType* namespace.

6.3.16 OEGetHeterogenTypeName

> **Attention**
>
> This API is currently available in C++ and Python.

```
std::string OEGetHeterogenTypeName(unsigned typeID)
```

This function returns a string version of the heterogen type from the unsigned integer argument *typeID*. The heterogen types are contained in the *OEHeterogenType* namespace.

6.3.17 OEGetPartialResidues

Attention

This API is currently available in C++ and Python.

```
bool OEGetPartialResidues(std::vector<OEChem::OEResidue>& partialResidues,
                          const OEChem::OEMolBase& mol)
```

This function looks through the molecule and generates a list of standard protein residues with partial sidechains, which can then be passed to *OEBuildSidechains*

See also

- *OEBuildSidechains* function

6.3.18 OEHeterogenMetadataFromJson

Attention

This API is currently available in C++ and Python.

```
bool OEHeterogenMetadataFromJson(OEHeterogenMetadata& metadata,
                                 const std::string& json)
```

Populates an *OEHeterogenMetadata* object from a JSON-formatted string. The function recognizes the following fields in the JSON object:

Field	Type	Description
`Title`	string	Human-readable title of the heterogen.
`Id`	string	Identifier for the heterogen (e.g. PDB ligand code).
`Smiles`	string	SMILES representation of the heterogen.
`Type`	string	Heterogen component type name, resolved via *OEGetHeterogenTypeID*.
`Tautomers`	array of strings	SMILES strings of known tautomeric forms of the heterogen.

Unrecognized or missing fields are silently ignored. Returns `true` if the JSON string was successfully parsed as an object; returns `false` if the string is not valid JSON or does not represent a JSON object.

See also

- *OEHeterogenMetadataToJson* function
- *OEHeterogenMetadata* class
- *OEGetHeterogenTypeID* function

6.3.19 OEHeterogenMetadataToJson

Attention

This API is currently available in C++ and Python.

```
std::string OEHeterogenMetadataToJson(const OEHeterogenMetadata& metadata,
                                      const bool prettyPrint = true)
```

Serializes an *OEHeterogenMetadata* object to a JSON-formatted string. The resulting JSON object contains the following fields:

Field	Type	Description
`Title`	string	Human-readable title of the heterogen.
`Id`	string	Identifier for the heterogen (e.g. PDB ligand code).
`Smiles`	string	SMILES representation of the heterogen.
`Type`	string	Heterogen component type name, as returned by *OEGetHeterogenTypeName*.
`Tautomers`	array of strings	SMILES strings of known tautomeric forms of the heterogen.

The *prettyPrint* argument controls formatting: when `true` (the default) the JSON is indented for readability; when `false` a compact single-line representation is produced.

See also

- *OEHeterogenMetadataFromJson* function
- *OEHeterogenMetadata* class
- *OEGetHeterogenTypeName* function

6.3.20 OEIsRedundantDesignUnit

Attention

This API is currently available in C++ and Python.

```
bool OEIsRedundantDesignUnit(const std::vector<OEBio::OEDesignUnit*>& dus,
                             const OEBio::OEDesignUnit& du,
                             const OESpruce::OEDesignUnitEnumerateSitesOptions& opts)
```

Returns `true` if the OEDesignUnit `du` is considered redundant with respect to the existing set of design units `dus`, using the criteria defined by the supplied *OEDesignUnitEnumerateSitesOptions* `opts`. A design unit is redundant when duplicate-removal logic (controlled by *OEDesignUnitEnumerateSitesOptions.SetDuplicateRemoval*) identifies it as equivalent to one already present in `dus`.

The input `du` must have 3D coordinates assigned.

See also

- *OEEnumerateSites* function
- *OEMakeDesignUnits* function
- *OEDesignUnitEnumerateSitesOptions* class

6.3.21 OEMakeBioDesignUnits

Attention

This API is currently available in C++ and Python.

```
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEMakeBioDesignUnits(const OEChem::OEMolBase&␣
↪structure,
                                                                const␣
↪OESpruce::OEStructureMetadata& metadata = OESpruce::OEStructureMetadata(),
                                                                const␣
↪OESpruce::OEMakeDesignUnitOptions& opts = OESpruce::OEMakeDesignUnitOptions())
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEMakeBioDesignUnits(const OEChem::OEMolBase&␣
↪structure,
                                                                const␣
↪OEBio::OEDesignUnit& designUnitReference,
                                                                const␣
↪OESpruce::OEStructureMetadata& metadata = OESpruce::OEStructureMetadata(),
                                                                const␣
↪OESpruce::OEMakeDesignUnitOptions& opts = OESpruce::OEMakeDesignUnitOptions())
```

Return an iterator over all OEDesignUnit objects that can be produced from the input OEMolBase structure. A reference structure to be used for biounit extraction and structural superposition can be passed in through the *designUnitReference* argument, while information regarding the experimental data as well as basic options for constructing the design units themselves can be provided through the *metadata* and *opts* arguments, respectively.

Note

We strongly recommend running *OESpruceFilter* for standardization and filtering prior to running the design unit preparation

The input `structure` must have 3D coordinates assigned.

See also

- *OEMakeDesignUnit*, and *OESpruceFilter*

6.3.22 OEMakeDesignUnit

Attention

This API is currently available in C++ and Python.

```
bool OEMakeDesignUnit(OEBio::OEDesignUnit& du,
                      const OEChem::OEMolBase& nonLigandComplex,
                      const OEChem::OEMolBase& ligand,
                      const OESpruce::OEMakeDesignUnitOptions& opts =␣
↪OESpruce::OEMakeDesignUnitOptions())
```

Constructs and populates an input OEDesignUnit object (*du*) from the input, a *nonLigandComplex* (typically a protein or DNA) and a *ligand*. An options *OEMakeDesignUnitOptions* object can be passed in to control the behavior during design unit preparation.

Note

We strongly recommend running *OESpruceFilter* for standardization and filtering prior to running the design unit preparation.

Note

The function will construct an OEDesignUnit (*du*), where the system's molecules have been binned into categories given by the `OEDesignUnitComponents` namespace. Although an `OEDesignUnitComponents.Metals` category exists, depending on the proximity of the metal to the design unit's binding site, the metal may be contained in either the `OEDesignUnitComponents.Cofactors` category if it falls within a cutoff to the binding site, or in the `OEDesignUnitComponents.Metals` category if not.

Code Example

- *Creating OEDesignUnits from a PDB file* example

See also

- *OEMakeDesignUnitOptions*, *OEMakeBioDesignUnits*, and *OESpruceFilter*

6.3.23 OEMakeDesignUnits

Attention

This API is currently available in C++ and Python.

```
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEMakeDesignUnits(const OEChem::OEMolBase&␣
↪structure,
                                                             const␣
↪OESpruce::OEStructureMetadata& metadata = OESpruce::OEStructureMetadata(),
                                                             const␣
↪OESpruce::OEMakeDesignUnitOptions& opts = OESpruce::OEMakeDesignUnitOptions(),
                                                             const std::string␣
↪siteResidue = "")
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEMakeDesignUnits(const OEChem::OEMolBase&␣
↪structure,
                                                             const OESystem::OESkewGrid&␣
↪electronDensity,
                                                             const␣
↪OESpruce::OEStructureMetadata& metadata = OESpruce::OEStructureMetadata(),
                                                             const␣
↪OESpruce::OEMakeDesignUnitOptions& opts = OESpruce::OEMakeDesignUnitOptions(),
                                                             const std::string␣
↪siteResidue = "")
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEMakeDesignUnits(const OEChem::OEMolBase&␣
↪structure,
                                                             const OEBio::OEDesignUnit&␣
↪designUnitReference,
                                                             const␣
↪OESpruce::OEStructureMetadata& metadata = OESpruce::OEStructureMetadata(),
                                                             const␣
↪OESpruce::OEMakeDesignUnitOptions& opts = OESpruce::OEMakeDesignUnitOptions(),
                                                             const std::string␣
↪siteResidue = "")
OESystem::OEIterBase<OEBio::OEDesignUnit>* OEMakeDesignUnits(const OEChem::OEMolBase&␣
↪structure,
                                                             const OESystem::OESkewGrid&␣
↪electronDensity,
                                                             const OEBio::OEDesignUnit&␣
↪designUnitReference,
                                                             const␣
↪OESpruce::OEStructureMetadata& metadata = OESpruce::OEStructureMetadata(),
                                                             const␣
↪OESpruce::OEMakeDesignUnitOptions& opts = OESpruce::OEMakeDesignUnitOptions(),
                                                             const std::string␣
↪siteResidue = "")
```

Return an iterator over all OEDesignUnit objects that can be produced from the input OEMolBase structure. A reference structure to be used for biounit extraction and structural superposition can be passed in through the *designUnitReference* argument, while information regarding the experimental data as well as basic options for constructing the design units themselves can be provided through the *OEStructureMetadata* (*metadata*), OESkewGrid (*electronDensity*) and *OEMakeDesignUnitOptions* (*opts*) arguments, respectively. The input option of providing a electron density map can be used to calculate Iridium score. For more information about the Iridium score, see *Iridium chapter of that toolkit*. The *OEStructureMetadata* class stores constants for use with OEFieldMeta. These attributes (e.g. symmetry) are used to construct DUs. There is an input option to specify a binding site using a single residue specification if apo (or holo). The format is "name:num:insert code:chainid", e.g. "ASP:25: :A" indicating Aspartic acid 25 in chain A. Note: A blank/whitespace character is used for the insert code, which is a typical use case.

Note

This function will return an iterator over OEDesignUnit pointers, where the system's molecules have been binned into categories given by the `OEDesignUnitComponents` namespace. Although an `OEDesignUnitComponents.Metals` category exists, depending on the proximity of the metal to the design unit's binding site, the metal may be contained in either the `OEDesignUnitComponents.Cofactors` category if it falls within a cutoff to the binding site, or in the `OEDesignUnitComponents.Metals` category if not.

Code Example

- *Creating OEDesignUnits from a PDB file* example

The input `structure` must have 3D coordinates assigned.

See also

- *OEMakeDesignUnitOptions* class

6.3.24 OEMakeDesignUnitFromPocket

Attention

This API is currently available in C++ and Python.

```
bool OEMakeDesignUnitFromPocket(OEBio::OEDesignUnit& du,
                                const OEBio::OEDesignUnit& designUnit,
                                const OESpruce::OEPocket& pocket)
```

Takes an existing OEDesignUnit object (*designunit*) and generates a new OEDesignUnit object (*du*), based on the input pocket. If the input designunit has a binding site defined, it will be reset in the output designunit (du) and switched to the site specified by the pocket.

```
bool OEMakeDesignUnitFromPocket(OEBio::OEDesignUnit& du,
                                const OEChem::OEMolBase& mol,
                                const OESpruce::OEPocket& pocket)
```

Takes an existing OEMolBase object (*mol*) and generates a new OEDesignUnit object (*du*), based on the input pocket. The molecule is not split into components but is placed as the 'protein' component of the design unit.

Note

This method is intended to be used for un-liganded pockets on a design unit.

See also

- *OEPocket* class
- *OEFindPockets* function
- *OEMakeDesignUnits* function
- *OEMakeBioDesignUnits* function
- *OEMakeDesignUnit* function

6.3.25 OEMutateResidue

Attention

This API is currently available in C++ and Python.

This function mutates a single standard protein residue in the molecule to another standard amino acid residue specified by the 3 letter residue name, it currently only supports the 20 standard amino acids. The function works by deleting the residue sidechain and rebuilds the new residue using the *OEBuildSidechains* function, but only building the mutated sidechain. The sidechain re-building can be controlled using the *OESidechainBuilderOptions* options class argument *opts*.

Note

Explicit hydrogens are not added to the structure during the build processes, therefore no partial charges are assigned.

```
bool OEMutateResidue(OEChem::OEMolBase& mol,
                     const OEChem::OEResidue& res,
                     const std::string newResName,
                     const OESpruce::OESidechainBuilderOptions& opts =␣
↪OESpruce::OESidechainBuilderOptions(true))
```

See also

- *OEBuildSidechains* function
- *OESidechainBuilderOptions* class

6.3.26 OEMutateResidues

Attention

This API is currently available in C++ and Python.

```
bool OEMutateResidues(OEChem::OEMolBase& mol,
                      const std::map<OEChem::OEResidue, std::string>& mutationMap,
                      const OESpruce::OESidechainBuilderOptions& opts =␣
↪OESpruce::OESidechainBuilderOptions(true))
```

This function takes an input OEMolBase (*mol*) and it mutates residues in the provided map. The sidechain re-building can be controlled using the *OESidechainBuilderOptions* options class argument *opts*.

See also

- *OEBuildSidechains* function
- *OESidechainBuilderOptions* class

6.3.27 OEProtonateDesignUnit

> **Attention**
>
> This API is currently available in C++ and Python.

```
bool OEProtonateDesignUnit(OEBio::OEDesignUnit& du,
                           const OESpruce::OEProtonateDesignUnitOptions& opts =␣
↪OEProtonateDesignUnitOptions())
bool OEProtonateDesignUnit(OEBio::OEDesignUnit& du,
                           const std::map<std::string, OEHeterogenMetadata>& metadata,
                           const OESpruce::OEProtonateDesignUnitOptions& opts =␣
↪OEProtonateDesignUnitOptions())
```

This function takes an input OEDesignUnit (*du*) and it intelligently places hydrogens across all internal molecular components. Optionally, one may pass in a *OEHeterogenMetadata* map keyed on the SMILES string for a given heterogen, as well as a set of general protonation options through the *OEProtonateDesignUnitOptions* argument, *opts*.

6.3.28 OESplitBioUnitToComponents

> **Attention**
>
> This API is currently available in C++ and Python.

```
bool OESplitBioUnitToComponents(OEBio::OEDesignUnit& du,
                                const OEChem::OEMolBase& bioUnit,
                                OEChem::OEMolBase& packingMol,
                                const OESpruce::OEDesignUnitSplitOptions& opts)
```

Splits the biological unit molecule `bioUnit` into the component categories of a OEDesignUnit (protein, ligand, solvent, metals, etc.) using the splitting criteria defined in *OEDesignUnitSplitOptions* `opts`. The resulting components are stored in `du`, and any packing residues are placed in `packingMol`. Returns `true` on success.

```
bool OESplitBioUnitToComponents(OEBio::OEDesignUnit& du,
                                const OEChem::OEMolBase& bioUnit,
                                OEChem::OEMolBase& packingMol,
                                const OESpruce::OEDesignUnitSplitOptions& opts,
                                const std::vector<std::string>& ligandNames)
```

As the first overload, but restricts ligand detection to residues whose three-letter codes appear in `ligandNames`. This allows explicit control over which heterogens are classified as ligands rather than cofactors or excipients.

```
bool OESplitBioUnitToComponents(OEBio::OEDesignUnit& du,
                                const OEChem::OEMolBase& bioUnit,
                                OEChem::OEMolBase& packingMol,
                                const OESpruce::OEDesignUnitSplitOptions& opts,
                                const std::map<unsigned, std::vector<std::string>>&␣
↪componentNames)
```

As the first overload, but accepts a map of `OEDesignUnitComponents` component IDs to lists of three-letter residue codes, giving fine-grained control over which residues are assigned to each component category.

The input `bioUnit` must have 3D coordinates assigned.

See also

- *OEDesignUnitSplitOptions* class
- *OEMakeDesignUnits* function
- *OEMakeBioDesignUnits* function
- OEDesignUnitComponents namespace

6.3.29 OESpruceGetArch

```
const char* OESpruceGetArch()
```

Returns the architecture of the current version of the Spruce toolkit. Examples include:

- microsoft-win32-x86
- Ubuntu-16.04-x64

6.3.30 OESpruceGetLicensee

```
bool OESpruceGetLicensee(std::string& licensee)
```

Returns the LICENSEE from the current valid **Spruce TK** license.

6.3.31 OESpruceGetPlatform

```
const char* OESpruceGetPlatform()
```

Returns the platform, including compiler version, of the current version of the Spruce toolkit. Examples include:

- microsoft-win32-msvc9-MD-x86
- Ubuntu-16.04-x64

6.3.32 OESpruceGetRelease

```
const char* OESpruceGetRelease()
```

Returns the version of this toolkit release. For example: `0.9.0`.

6.3.33 OESpruceGetSite

```
bool OESpruceGetSite(std::string& site)
```

Returns the SITE from the current valid **Spruce TK** license.

6.3.34 OESpruceGetVersion

```
unsigned int OESpruceGetVersion()
```

Returns the build date of the toolkit as an unsigned int. For example: `20180126`.

6.3.35 OESpruceIsLicensed

```
bool OESpruceIsLicensed(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 **Spruce TK** functionality.

The 'feature' argument can be used to check for a valid license to **Spruce 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 **Spruce TK** license.

See also

- `OEChemIsLicensed` function

6.3.36 OESequenceMetadataFromJson

Attention

This API is currently available in C++ and Python.

```
bool OESequenceMetadataFromJson(OESequenceMetadata& metadata,
                                const std::string& json)
```

Populates an *OESequenceMetadata* object from a JSON-formatted string. The function recognizes the following fields in the JSON object:

Field	Type	Description
`ChainID`	string	Chain identifier for this sequence segment.
`StartResNum`	integer	Residue number of the first residue in the sequence segment.
`StartResInsCode`	string (single character)	Insertion code of the first residue in the sequence segment.
`EndResNum`	integer	Residue number of the last residue in the sequence segment.
`EndResInsCode`	string (single character)	Insertion code of the last residue in the sequence segment.
`Sequence`	string	One-letter amino acid sequence for this segment.

Unrecognized or missing fields are silently ignored. Returns `true` if the JSON string was successfully parsed as an object; returns `false` if the string is not valid JSON or does not represent a JSON object.

See also

- *OESequenceMetadataToJson* function
- *OESequenceMetadata* class

6.3.37 OESequenceMetadataToJson

> **Attention**
>
> This API is currently available in C++ and Python.

```
std::string OESequenceMetadataToJson(const OESequenceMetadata& metadata,
                                     const bool prettyPrint = true)
```

Serializes an *OESequenceMetadata* object to a JSON-formatted string. The resulting JSON object contains the following fields:

Field	Type	Description
`ChainID`	string	Chain identifier for this sequence segment.
`StartResNum`	integer	Residue number of the first residue in the sequence segment.
`StartResInsCode`	string (single character)	Insertion code of the first residue in the sequence segment.
`EndResNum`	integer	Residue number of the last residue in the sequence segment.
`EndResInsCode`	string (single character)	Insertion code of the last residue in the sequence segment.
`Sequence`	string	One-letter amino acid sequence for this segment.

The *prettyPrint* argument controls formatting: when `true` (the default) the JSON is indented for readability; when `false` a compact single-line representation is produced.

> **See also**
>
> - *OESequenceMetadataFromJson* function
> - *OESequenceMetadata* class

6.3.38 OEStructureMetadataFromJson

Attention

This API is currently available in C++ and Python.

```
bool OEStructureMetadataFromJson(OEStructureMetadata& metadata,
                                 const std::string& json)
```

Populates an *OEStructureMetadata* object from a JSON-formatted string. The function recognizes the following top-level fields in the JSON object:

Field	Type	Description
`StructureID`	string	Identifier for the structure (e.g. PDB accession code).
`Author`	string	Name of the structure's depositing author.
`Keywords`	array of strings	Keywords describing the structure.
`RevisionDate`	string	Date of the most recent revision.
`Revision`	string	Revision version label.
`ExperimentDate`	string	Date the experiment was performed.
`ExperimentType`	string	Experiment method name, resolved via *OEGetExperimentTypeID*.
`IridiumData`	object	Nested object containing Iridium quality assessment data.
`SequenceMetadata`	array of objects	Array of sequence segment objects, each parsed according to the fields documented in *OESequenceMetadataFromJson*.
`HeterogenMetadata`	array of objects	Array of heterogen objects, each parsed according to the fields documented in *OEHeterogenMetadataFromJson*.

Unrecognized or missing fields are silently ignored. Returns `true` if the JSON string was successfully parsed as an object; returns `false` if the string is not valid JSON or does not represent a JSON object.

Code Example

- *Creating OEDesignUnits Using Provided Metadata* example

See also

- *OEStructureMetadataToJson* function
- *OESequenceMetadataFromJson* function
- *OEHeterogenMetadataFromJson* function
- *OEStructureMetadata* class
- *OEGetExperimentTypeID* function

6.3.39 OEStructureMetadataToJson

Attention

This API is currently available in C++ and Python.

```
std::string OEStructureMetadataToJson(const OEStructureMetadata& metadata,
                                      const bool prettyPrint = true)
```

Serializes an *OEStructureMetadata* object to a JSON-formatted string. The resulting JSON object contains the following top-level fields:

Field	Type	Description
`StructureID`	string	Identifier for the structure (e.g. PDB accession code).
`Author`	string	Name of the structure's depositing author.
`Keywords`	array of strings	Keywords describing the structure.
`RevisionDate`	string	Date of the most recent revision.
`Revision`	string	Revision version label.
`ExperimentDate`	string	Date the experiment was performed.
`ExperimentType`	string	Experiment method name, as returned by *OEGetExperimentTypeName*.
`IridiumData`	object	Nested object containing Iridium quality assessment data.
`SequenceMetadata`	array of objects	Array of sequence segment objects, each with the fields documented in *OESequenceMetadataToJson*.
`HeterogenMetadata`	array of objects	Array of heterogen objects, each with the fields documented in *OEHeterogenMetadataToJson*.

The *prettyPrint* argument controls formatting: when `true` (the default) the JSON is indented for readability; when `false` a compact single-line representation is produced.

Code Example

- *Creating OEDesignUnits Using Provided Metadata* example

See also

- *OEStructureMetadataFromJson* function
- *OESequenceMetadataToJson* function
- *OEHeterogenMetadataToJson* function
- *OEStructureMetadata* class
- *OEGetExperimentTypeName* function

6.3.40 OEGetSuperposeMethodFromName

> **Attention**
>
> This API is currently available in C++ and Python.

```
unsigned OEGetSuperposeMethodFromName(const std::string& name)
```

Returns the corresponding constant defined in the *OESuperposeMethod* namespace for the given superposition method name.

6.3.41 OEGetSuperposeMethodName

> **Attention**
>
> This API is currently available in C++ and Python.

```
std::string OEGetSuperposeMethodName(const unsigned method)
```

Returns the name of the superposition method as a string.

CHAPTER

SEVEN

RELEASE HISTORY

7.1 Spruce TK 1.9.0

7.1.1 New Features

- The maximum cluster size for partial residues built using *OEBuildSidechains* can now be set in *OESidechainBuilderOptions* using *SetMaxClusterSize*.
- The chemical component dictionary has been updated. However, if using mmCIF, the metadata will be used as primary information and the dictionary only engages if metadata is unavailable.

7.1.2 Minor Bug Fixes

- An issue has been fixed in the routine to reassign ambiguous water and metals from packing residues to the protein chain of interest. The issue caused HEME analogs and connected amino acids to be soaked into a given chain ID.
- An issue has been fixed to allow `OEHeterogenMetadata` to inform the component splitter during design unit generation and preparation. This also fixes an issue where the splitter would not allow reclassification of a metal ion to an excipient.
- An issue has been fixed that caused the option to collapse redundant alternate locations to improperly engage if there were multiple binding sites in a protein.
- An issue has been fixed that caused the option to collapse redundant alternate locations to improperly engage for different ligands in the same binding site.
- An issue has been fixed that caused `OEMutateResidues` to incorrectly leave the backbone nitrogen of a residue in a bad valence state when mutating from Proline.
- An issue has been fixed in `OEBuildSidechains` with the shell mutation option turned on that would leave side chains clashing with water molecules. The water molecules are now removed if the option to allow for removal of clashing waters is set to `True`.

7.2 Spruce TK 1.8.0

7.2.1 New Features

- The function *OEBuildSingleLoop* now checks for loop clashes after minimization. The option can be set using *SetRejectClashingLoopsPostMinimization*.
- The function *OEBuildSingleLoop* can now delete residues already present at sites where a new loop needs to be built. The option can be set using *SetDeleteExistingResidues*.
- The function *OEBuildSidechains* can now partially build target residues in the event that some targets fail.

- Improved loop modeling has been added involving disulfide bridge creation by adding a bond length check.
- When running with *SetFixResStates* set to True, if the input molecule contains CIF header data, the filter will utilize the chemical component header section to inform a residue's bond order and subsequent formal charge.
- A new method, *SetUseSeqScoreChainMatching*, has been added to *OESuperposeOptions* which, when enabled, optimizes the sequence score over the overall RMSD when matching on multiple protein chains.

7.2.2 Major Bug Fixes

- An issue has been fixed where gaps in the protein structure were misidentified due to unconventional insert code usage.

7.2.3 Minor Bug Fixes

- Loop modeling results have been improved by setting *OELoopBuilderOptions StrictProlineMatch* to `False` by default. The updated default is due to an improved side chain building routine in a prior toolkit release.
- An issue has been fixed to ensure loops extracted from the loop database had the desired structural properties.
- When truncating a residue or converting a residue to a capping group, if that residue had a crosslink bond, the residue opposite would retain an atom in a bad valence state. That issue has been fixed.
- An issue has been fixed in the routine to reassign ambiguous water and metals from packing residues to the protein chain of interest. The issue caused HEME-analogs and connected amino acids to be soaked into a given chain ID.
- An issue has been fixed causing loop-relevant information with insert codes in an mmCIF header to be incorrectly read.
- The *OECapCTermini*, *OECapNTermini*, and *OECapTermini* functions address an issue that could result in an incorrectly labeled cap residue SequenceID.
- Several improvements were made to address performance issues in the *OESpruceFilter*.

7.2.4 Documentation Changes

- Several examples have been added, and the advanced protein preparation tutorial has been expanded.

7.3 Spruce TK 1.7.0

7.3.1 New Features

- The functions *OEMakeDesignUnit* and *OEMakeBioDesignUnits* will now use provided *OEHeterogenMetadata* in *OEStructureMetadata* as information for the splitting of a structure into a design unit, including such details as which parts should be considered cofactors.

7.3.2 Major Bug Fixes

- An issue has been fixed that caused charge and radii assignment to not happen, in paths where **Spruce** was also not allowed to protonate the structure.
- An issue has been fixed that caused radii to not be properly assigned to all atoms in the design unit.

7.3.3 Minor Bug Fixes

- An issue has been fixed which caused some mmCIF files to not be able to be prepared due to lack of missing metadata.
- An issue has been fixed where *OEMutateResidues* would crash if the structure contained the N-terminal capping group *NME*.

7.4 Spruce TK 1.6.1.1

7.4.1 Major bug fixes

- An issue has been fixed where *OEExtractBioUnits* function would fail if there was a mismatch in author assigned chain names and PDB assigned chain names in the case where a molecule was extracted from an mmCIF file.
- An issue was fixed in *OEProtonateDesignUnit* where a rule to flip clashing in-plane hydrogens on double bonded nitrogens was triggering incorrectly in specific instances.

7.5 Spruce TK 1.6.1

7.5.1 New features

- *OEBuildSidechains* now minimizes target side chains by default. Minimization can be controlled by setting *minimizeSidechains* and *minimizeSidechainsShell* options in the *OESidechainBuilderOptions* class.
- A feature has been added to support a next generation mmCIF feature, such that when an mmCIF file provides a connectivity record for the heterogens, **Spruce** will no longer look up the heterogen in the Chemical Component Dictionary but trust the input provided.
- A feature has been added that allows in-plane flipping of a single hydrogen on a double bonded nitrogen, when a clash is detected as part of hydrogen bond network optimization.

7.5.2 Major bug fixes

- An issue has been fixed related to using the *OEHeterogenMetadata* class, where the provided SMILES string was not respected for tautomer generation; when tautomers were not provided; and when the title (residue name) collided with existing residue names in the stored Chemical Component Dictionary.
- An issue has been fixed related to covalently bound ligands, where the pre-reaction input SMILES and related tautomers were not adjusted properly when accounting for the formed covalent bond.
- An issue has been fixed where the **Spruce** filter deleted single (disconnected) amino acid residues when they were marked HETATMs and were not real "floating" residues in part of the regular protein sequence.

7.5.3 Minor bug fixes

- *OEMutateResidues*, *OEMutateResidue*, and *OEBuildSidechains* now reduce failures arising from clashes by using shell minimization by default.
- An issue has been fixed that caused HEME-like molecules to be excluded from the stored chemical component database.

7.5.4 Documentation changes

- Information has been added about setting and getting minimizeSidechains and minimizeSidechainsShell options in the *OESidechainBuilderOptions* class.

7.6 Spruce TK 1.6.0

7.6.1 New features

- The chemical components dictionary has been updated with latest structures from the wwPDB.
- A feature has been added to *OEProtonateDesignUnit* to detect acceptor–acceptor clashes, modify the protonation state of one of the moieties, and reoptimize the hydrogen bond network.
- The following preliminary API classes have been made permanent.
  - *OECapBuilderOptions*
  - *OELoopBuilderOptions*
  - *OEPocket*
  - *OEPocketOptions*
  - *OESidechainBuilderOptions*
  - *OESpruceFilter*
  - *OESpruceFilterOptions*
  - *OEValidateDesignUnit*
- The following preliminary API functions have been made permanent.
  - *OEBuildLoops*
  - *OEBuildSingleLoop*
  - *OEFindPockets*
  - *OEMakeDesignUnitFromPocket*

7.6.2 Major bug fixes

- A crash in *OESpruceFilter* has been fixed in relation to atoms that were cleaned up during the standardization process.

7.6.3 Minor bug fixes

- An issue has been fixed that ensures radii are assigned to all atoms in a design unit.

7.6.4 Documentation changes

- An example has been added showing how to convert FASTA files to structure metadata for structures missing the SEQRES information in the PDB metadata and MMCIF equivalent.

7.7 Spruce TK 1.5.3

7.7.1 New features

- The chemical components dictionary was updated with latest data from the RCSB.

7.7.2 Major bug fixes

- An issue that caused *OESpruceFilter* to segfault has been fixed and safeguarded.

7.7.3 Minor bug fixes

- An issue has been fixed that caused incorrectly built loops when the protein structure contains nonstandard amino acids, particularly in the gap needing to be built or anchoring to it.
- An issue with sequence mutation for cysteine residues, where an explicit hydrogen was left over after the mutation, has been fixed.

7.8 Spruce TK 1.5.2

7.8.1 New features

- *OESpruceFilterOptions* has an option to automatically assign chain IDs (default is true) when they are missing either completely or partially in the structure for a structure processed by *OESpruceFilter*.
- The built-in heterogen database has been updated with the latest structures from the ligand-expo at the RCSB.

7.8.2 Major bug fixes

- An issue where tautomers passed using metadata were not respected has been fixed.
- An issue has been fixed where, in a small fraction of cases, an OXT atom was not correctly built out when needed at the protein C-terminals.
- An issue has been fixed where the option to fix the state of backbone atoms and bonds in *OESpruceFilterOptions* was sometimes not respected in *OESpruceFilter*, which led to issues such as OXT atoms being built even if the option was disabled.

7.8.3 Minor bug fixes

- An issue was fixed in filter options to not accept the disallowed residue names UNL and UNK. Instead, they are renamed to LIG, which is a change filter that applies in the structure upon standardization.
- An issue where the maximum number of atoms in the biological unit was not being respected has been fixed.

7.8.4 Documentation changes

- Minor documentation and example improvements have been fixed.

7.9 Spruce TK 1.5.1

7.9.1 New features

- If a site residue is provided to *OEMakeDesignUnits*, the output design unit title now includes the site residue in the title for apo structures. Previously, the residue in the title was the residue closest to the center of the binding site.
- Standardization using *Spruce Filter* now checks for metal chelation before forming disulfide bridges.

7.9.2 Major bug fixes

- An issue causing accumulation of generic data on the molecule components stored in the design unit object during prep has been fixed. This accumulation led to large memory consumption, slow preparation, and larger file size.

7.10 Spruce TK 1.5.0

7.10.1 New features

- The new *OESpruceFilter* class has been added to provide checks and corrections on input to **Spruce**. For input it can take an OEMolBase, *OEStructureMetadata*, or an options class, *OESpruceFilterOptions*, that holds the options to turn on each of the fixing options.
- The new *OEValidateDesignUnit* class has been added for validation of **Spruce** produced design units. For input it can take an OEDesignUnit or *OEStructureMetadata*.

7.10.2 Major bug fixes

- A bug that caused memory leaks in **OEBio TK** has been fixed.

7.10.3 Minor bug fixes

- An issue was fixed in *OEBuildLoops* to allow better sequence alignment for peptides.
- An issue was fixed in *OEBuildLoops* where free amino acids, as co-factors or ligands, were incorrectly included in sequence alignments and causing mismatched alignments to the SEQRES.
- An issue was fixed in *OEExtractBioUnits* to be able to lower the minimum sequence alignment score threshold.

7.11 Spruce TK 1.4.0

Fall 2021

7.11.1 New features

- A new method, *SetForceCapping*, has been added to enable capping even if it will result in a clash.
- Two new methods, *SetBuildTails* and *GetBuildTails*, have been added to set and get an option that controls whether to build tails when there is missing residues at the N- and C- termini.
- *SiteHopper* is now a supported superposition method in *OESuperpose*, and can be set using the SetMethod method of *OESuperpose*.
- All Options classes have been modified to derive from OEOptions. This affects classes:
  - *OEDesignUnitEnumerateSitesOptions*
  - *OEProtonateDesignUnitOptions*
  - *OEDesignUnitBuildOptions*
  - *OEDesignUnitPrepOptions*
  - *OEDesignUnitSplitOptions*
  - *OEBioUnitExtractionOptions*
  - *OEMakeDesignUnitOptions*
  - OEPlaceHydrogensOptions

7.11.2 Major bug fixes

- An issue causing capping a nonstandard residue to fail has been fixed.
- An issue causing covalent ligands to have their covalent bond broken when there were alternate locations in the structure, has been fixed.
- An issue causing side-chains to either not be built or being built in non-optimal configurations has been fixed.
- An issue with pockets identifying liganded sites in *OEMakeDesignUnitFromPocket* not generating a proper design unit with a ligand bound in the identified site has been fixed.

7.11.3 Minor bug fixes

- An issue causing the warning message 'Truncated cap due to clash' to appear when *SetAllowTruncate* is *False* has been fixed.
- An issue caused by Heuristic score offset being added before comparing to original structure therefore unable to pick the more energy favorable conformation has been fixed.
- An issue causing the gap finding code to mis-identify gaps in the protein structure due to residues with missing backbone atoms has been fixed. The function now removes these residues prior to gap detection inside *OEBuildLoops*.
- An issue caused by malformed PDB Header data, specifically with regards to REMARK 300 sections, has been fixed.

7.12 Spruce TK 1.3.0

July 2021

7.12.1 New features

- A molecule-based API for *OEMakeDesignUnitFromPocket* has been added.
- New high-level superposition APIs were added that unify the sequence-based (*OEStructuralSuperposition*) and shape-based (*OESecondaryStructureSuperposition*) APIs. This includes a new *OESuperpose* class that can calculate different types of structural superposition, a new *OESuperposeOptions* class that allows the users to control various behaviors of the superposition calculation, such as the superposition method, and a new *OESuperposeResults* class that holds the transformation and scores of the superposition. More detailed changes are as follows:
  - The new *OESuperpose* class can take either OEMol or OEDesignUnit as the reference for the fit molecule. The reference molecule can be set by using the *SetupRef* method, and the superposition can be calculated via *OESuperpose* with the fit molecule supplied as one of its arguments.
  - A new *OESuperposeMethod* namespace was added that contains the constants for different superposition methods. Functions *OEGetSuperposeMethodName* and *OEGetSuperposeMethodFromName* were added to obtain the superposition method's name (as a string literal) from its designated constant and vice versa.
  - Methods *OESuperposeOptions*, *OESuperposeOptions*, and *OESuperposeOptions* have been added. These methods can be used to set score thresholds for validating superposition results in the new *OESuperposeOptions* class. The validity of the superposition result can be obtained by *OESuperposeResults* from the new *OESuperposeResults* class.
  - The new *OESuperpose* API returns an *OESuperposeResults* object that can be used to transform the fit molecule to its superposed position with its *Transform* method. Alternatively, the users can use *GetRotMatrix* and *GetTranslation* to obtain the rotation matrix and translation vector, respectively, for their analyses. The RMSD and sequence alignment score after the superposition can be obtained using *OESuperposeResults*'s *GetRMSD* and *GetSeqScore* methods for sequence-based superpositions, and the Tanimoto score can be obtained using *GetTanimoto*.

Please refer to the documentation for more information on these new superposition APIs.

7.12.2 Minor bug fixes

- A bug that caused the loop modeling algorithm to crash in rare instances has been fixed.
- Examples have been modified to use appropriate command-line arguments.
- An issue that caused style on OEDesignUnit objects to disappear when calling *OEProtonateDesignUnit* has been fixed.
- An issue with the return value of superposition when two structures cannot be superposed has been fixed. It now properly returns a -1.00 RMSD.
- An issue in *OEMakeDesignUnitFromPocket* that caused style to remain on both the previous binding site and the one defined from the pocket has been fixed.
- A bug in superposition that resulted in a crash when using site residues that were not present on a design unit has been fixed.
- A bug that kept the insert code from its anchor residue when there was an N-terminal cap has been fixed.

7.12.3 Documentation changes

- Documentation has been added for the Iridium classification. *Iridium.*
- Documentation has been added for the new superposition APIs. Superposition related APIs are now fully supported. *Superposition.*

7.13 Spruce TK 1.2.0

Fall 2020

7.13.1 New features

- New methods have been added to *OEDesignUnitSplitOptions* (*AddLipidCode*, *SetLipidCodes*, *ClearLipidCodes*, and *GetLipidCodes*). They allow users to specify the names of molecules to be considered as lipids.
- Support has been added for HEME and related cofactors in *OEMakeDesignUnits*. The state of HEME, contains two negatively charged nitrogens to balance the +2 Fe formal charge. The nitrogens coordinating the iron atom are indicated by zero-order bonds.
- A new function, *OEFindPockets*, has been added that finds cavities in protein or nucleic acid complexes. The parameters for the search can be tuned with *OEPocketOptions*. The pockets are output as *OEPocket* objects in an iterator. The function can take a predicate that will control only outputting pockets with a given set of residues or atoms in it. This function is also used in *OEMakeDesignUnits* when a site residue is provided.
- A new function, *OEMakeDesignUnitFromPocket*, has been added that takes a prepared biological unit from an *OEMakeDesignUnit* and an *OEPocket* and coverts it into an OEDesignUnit does with a reference structure.
- New methods have been added in *OELoopBuilderOptions*, *SetSeqAlignMethod*, *SetSeqAlignGapPenalty*, *SetSeqAlignExtendPenalty*, *GetSeqAlignMethod*, *GetSeqAlignGapPenalty*, and *GetSeqAlignExtendPenalty* that allow users to control the sequence alignment done to find gaps in protein structures related to loop modeling.

7.13.2 Major bug fixes

- A bug that caused some modeled loops to be missing the backbone NH hydrogen where the modeled loops were attached has been fixed.

- Fragment numbers for build loops have been fixed to be consistent with the anchor residues where the loops are inserted.
- A bug in *OEFixBackbone* has been fixed to ensure that OXT atoms are always built on C-terminal residues.
- An issue with atom order for built pieces like loops or caps has been fixed to ensure that these are sequential with the location they are built.
- A bug in *OEBuildLoops* that in specific situations caused incorrect detection of gaps in protein structures has been fixed. Additionally, options have been exposed in *OELoopBuilderOptions* to allows users to control the underlying sequence alignment in the event of a failure.

7.13.3 Minor bug fixes

- Improvements have been made to standard amino acid residue perception when preparing structures.
- Site residues in *OEMakeDesignUnits* are now being checked against all the biological units so one can serve as a reference for the rest.
- A bug in loop modeling that caused the function to create residues with identical residue numbers has been fixed. The function now properly builds using insertion codes.
- A bug in loop modeling that caused problems for the underlying sequence alignment code and incorrect loops when a single residue was floating in space between two gaps has been fixed.
- The chemical component dictionary used internally in *OEMakeDesignUnits* has been updated with the latest changes from the RCSB, including some fixes for incorrect entries in the source data based on rules and literature searches.

7.14 Spruce TK 1.1.0

7.14.1 New features

- Two new APIs, `OEBuildSingleLoop` and `OEBuildLoops`, have been added that provide loop modeling capability. This capability relies on either the SEQRES in the PDB header (or equivalent for MMCIF files) or the user passing metadata about the sequences. The functions take an options class, *OELoopBuilderOptions*, as an argument to specify the file name of the required loop template database. This database is available for download in a platform-independent format.
- A new parameter, `siteResidue`, has been added to the `OEMakeDesignUnits` function. This allows users to specify a single residue in the binding site for APO structures when using the `OEEnumerateSites` function. The function checks whether the detected site has a large enough volume and whether there is anything in the binding site that the splitter failed to identify as a potential ligand.
- The options class *OEDesignUnitBuildOptions* has been updated to hold options classes for the different builder options available. This also changes the options class structure for *OEMakeDesignUnitOptions*.
- A new low level class, *OECapBuilderOptions*, that determines behavior when building caps on proteins, has been added to replace passing different parameters to the functions `OECapTermini`, `OECapNTermini`, and `OECapCTermini`.
- *OESidechainBuilderOptions*, a low level class that determines the behavior when building sidechains on proteins, replaces passing different parameters to the function `OEBuildSidechains`.
- The preliminary API options class `OEDesignUnitMutationOptions` has been removed. `OEMutateResidues` has been added to take a map of mutations, along with *OESidechainBuilderOptions* to control the behavior when building the mutated residue sidechain.

- A new function, *OEFixBackbone*, has been added that builds =O and/or OXT, if necessary, and ensures that all formal charges and valence states are correct on all backbone atoms. The function removes a residue completely if the primary backbone atoms, N, CA, and C, are missing. The residues can be built correctly later with *OEBuildLoops*.
- A strict protonation mode has been added to *OEDesignUnitPrepOptions*, enforcing that when an error occurs during addition and optimization of hydrogens to the system, the preparation processes now stop and fail.
- A new options class, OEPlaceHydrogensOptions, has been added to *OEProtonateDesignUnitOptions*. This option allows users more control over the protonation processes; for example, by passing a bypass or no-flip predicate that will prevent the protonation process from altering the residues in the predicate.
- The design units produced by *OEMakeDesignUnits* are now ordered by the quality of the Iridium classification data.

7.14.2 Major bug fixes

- A bug in *OEExtractBioUnits* that occurred when using a reference has been fixed. Previously, it resulted in incorrect biological units being produced under certain conditions. The algorithm has also been improved to more often produce the correct biological unit based on symmetry expansion.

7.14.3 Minor bug fixes

- Metal co-factors previously had not been properly identified in the binding site during design unit generation due to the distance threshold from a ligand that was too strict. This has been fixed.
- Ligands with an incorrect alternate location assignment but that were in the same location were previously not correctly identified and were assigned alternate locations. This was due to the threshold being too strict and has been fixed.
- In cases where a residue is itself converted to a cap and where building a cap would result in the cap strongly clashing, the cap number no longer adds an insert code or alters the residue number. In addition, valence states resulting from the residue conversion are being more thoroughly validated and fixed, if necessary.
- Some instances of functions changing the location of the `OEThrow` output and their levels have been fixed.
- An issue in *OEExtractBioUnits* had previously occurred when structures with the historical convention of putting waters and metals into specific chain IDs (i.e., W and S, respectively) caused the molecules, after symmetry expansion, to have identical residue information. This has been fixed.
- An issue with the side-chain builder miscounting side-chain atoms when involving a residue in cross-linking has been fixed.
- An issue that occurred when a single ligand atom (per design) was not properly detected as a ligand has been fixed.

7.14.4 Documentation changes

- Documentation and examples have been modified to omit the step where alternate locations are collapsed. The function *OEMakeDesignUnits* handles alternate location collapse or enumeration and should provide all available data.
- Examples in C++ and Python have been enhanced to include passing electron density maps, and a new example has been added for making APO design units using a site residue or reference structure. The examples also illustrate how to enable **Spruce TK** to build missing loops.

7.15 Spruce TK 1.0.0

7.15.1 New features

- The *OEExtractBioUnits* function has become a stable API. The final implementation includes an options class, *OEBioUnitExtractionOptions*, to control its behavior.
- A new function, *OEMakeDesignUnits*, has been added that generates OEDesignUnit from structures read from PDB or mmCIF files. The function takes a number of options to control its behavior; therefore, the following options classes have been defined:
  - *OEMakeDesignUnitOptions*: a high-level options class that holds the options classes below.
  - *OEDesignUnitPrepOptions*: a high-level options class that holds other options classes, determining how a design unit is prepped.
  - *OEDesignUnitBuildOptions*: a low-level class that sets whether to build caps and missing sidechains.
  - *OEProtonateDesignUnitOptions*: a low-level class that sets whether and how to protonate the design unit, including whether to generate tautomers.
  - *OEDesignUnitSplitOptions*: a low-level class that determines how to handle alternate locations, how a system is split, locating the ligand, etc.

  The *OEMakeDesignUnits* function also takes the *OEStructureMetadata* class as input. The *OEStructureMetadata* class holds experimental data if known and can be used to indicate which is the ligand, what the sequence of the protein is, and other relevant pieces of information.
- The function *OEMakeDesignUnits* is a high-level function that works on the two intermediate functions, *OEMakeBioDesignUnits* and *OEEnumerateSites*, which can be run separately if intermediate results are desired. The high-level *OEMakeDesignUnits* function has a deduplication step built in; this function is highly recommended.
- A new function, *OEMakeDesignUnit*, has been added that creates an OEDesignUnit.
- A new function, *OEProtonateDesignUnit*, has been added that ensures that protons in an OEDesignUnit are optimized. The high-level function *OEMakeDesignUnits* calls this by default unless it is turned off in options.
- The *OEBuildSidechains* function has been updated to allow building all possible clusters even if one fails due to clashes. The function still returns `false` in this case.

7.15.2 Major bug fixes

- An issue in the *OEExtractBioUnits* function that caused a segfault due to an improper PDB header format has been fixed.
- An issue in the alternate location expansion that left atoms on top of each other due to unconventional location definitions has been fixed.

7.15.3 Minor bug fixes

- An issue with *OECapTermini* preventing caps from being built in some very specific instances has been fixed.
- *OEBuildSidechains* no longer builds sidechains on packing residues.

7.15.4 Documentation changes

- A new example, *Creating OEDesignUnits from a PDB file*, has been added that illustrates the usage of the high-level *OEMakeDesignUnits* function for biomolecular structure preparation. This example takes a PDB file as input and outputs a set of fully prepared OEDesignUnit objects to a temporary file.

7.16 Spruce TK 0.9.2

- The product name in the license file for **Spruce TK** has been changed from `spruce` to `sprucetk`. This change does not impact the validity of current licenses, which will continue to work with this new version of the toolkit. For more information on **Spruce TK** licensing, please visit https://www.eyesopen.com/contact.
- **Spruce TK** is still preliminary.

7.16.1 New features

- A new predicate, *OEIsModeledAtom*, has been added that identifies atoms built by the **Spruce TK** functions.
- A new function, *OEBuildSidechains*, has been added to build partial or missing sidechains of standard amino acid residues in a protein.
- A new method, *OECapCTermini*, has been added to cap the C-termini of a protein.
- A new method, *OECapNTermini*, has been added to cap the N-termini of a protein.
- A new method, *OECapTermini*, has been added to cap both the N- and C-termini of a protein.
- A new method, *OEGetPartialResidues*, has been added to return a list of partial residues in a protein.
- A new method, *OEMutateResidue*, has been added to mutate a single residue in a protein.

7.16.2 Major bug fixes

- A memory leak in the *OESecondaryStructureSuperposition* class has been fixed.

7.16.3 Minor bug fixes

- *OEExtractBioUnits* has been updated to better handle atoms placed on a symmetry axis.
- *OEExtractBioUnits* now more reliably assigns components to a biological unit (BU) even when they are far away.

7.17 Spruce TK 0.9.1

7.17.1 New features

> **Note**
>
> **Spruce TK** now requires a specific license. For information on **Spruce TK** licensing, please visit https://www.eyesopen.com/contact.

- The *OESecondaryStructureSuperposition* class can now use ligand constraints initialized in the *OESuperpositionOptions* option class when handled by the following new methods:
  - *OESuperpositionOptions.SetLigandConstraints*
  - *OESuperpositionOptions.HasLigandConstraints*
  - *OESuperpositionOptions.GetLigandConstraints*
  - *OESuperpositionOptions.ClearLigandConstraints*
- New *OEExtractBioUnits* high-level functions have been added to extract biological assemblies, replacing the `OEExtractBioUnitsFromRef` and `OEExtractBioUnitsFromPDBRemarks` functions.

- The *OESecondaryStructureSuperposition* class now can be initialized with the *OESuperpositionOptions* option class.
- The `OESuperpositionType` namespace now contains an entry for the `SSE` superposition method. The default superpositioning method is now `OESuperpositionType.Global`.
- New *OESStructural Superposition Transform* and *OESecondaryStructureSuperposition Transform* methods have been added to transform a molecule based on superpositioning.
- New *OESStructural Superposition operator bool* and *OESecondaryStructureSuperposition operator bool* methods have been added to validate objects.

7.17.2 API changes

- The constructor of the *OEStructuralSuperposition* class has been changed.
- The low-level `OEExtractBioUnitsFromRef` and `OEExtractBioUnitsFromPDBRemarks` functions have been removed.

7.17.3 Major bug fixes

- A bug that caused an issue with extracting biological assemblies has been fixed in the low-level biological assembly factory class.
- Biological assembly extraction for large proteins has been improved for memory and CPU usage.

7.17.4 Minor bug fixes

- Superposition methods for using OETrans instead of raw double pointers for rotation matrices and translation vectors have been improved.
- A bug that caused an issue with extracting biological assemblies if they had a blank REMARK line in the PDB file has been fixed.

7.18 Spruce TK 0.9.0

7.18.1 New features

Spruce TK has a limited API that is in a preliminary state until the 2018.Oct release. The currently available classes in the API are *OESecondaryStructureSuperposition*, *OEStructuralSuperposition*, and *OESuperpositionOptions*. The currently available functions are `OEExtractBioUnitsFromRef` and `OEExtractBioUnitsFromPDBRemarks`.

CHAPTER

EIGHT

COPYRIGHT AND TRADEMARKS

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

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

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

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

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

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

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

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

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

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

CHAPTER

NINE

SAMPLE CODE

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

TERMS FOR USE OF SAMPLE CODE

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

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

CHAPTER

TEN

CITATION

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

For more information, please contact OpenEye Support.

10.1 Orion®

To cite use of Orion, please use the following:

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

For example:

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

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

10.2 Orion Floes

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

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

For example:

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

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

10.3 Toolkits and Applications

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

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

For example:

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

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

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

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

where <*product name*> is:

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

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

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

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

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

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

10.4 Publications for Bibliographies

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

10.4.1 Orion

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

10.4.2 AFITT and FLYNN Applications

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

10.4.3 OEDocking Application and Toolkit

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

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

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

10.4.4 OMEGA Application and Toolkit

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

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

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

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

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

10.4.5 ROCS Application

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

10.5 OpenEye MMDS Web Service

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

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

For example:

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

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

CHAPTER

ELEVEN

TECHNOLOGY LICENSING

OpenEye products use the following technology under license.

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

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

Table 1: Licensed Technology

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

continues on next page

Table 1 – continued from previous page

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

11.1 GCC

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

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

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

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

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

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

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

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

11.1.1 GCC RUNTIME LIBRARY EXCEPTION

```
Version 3.1, 31 March 2009

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

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

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

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

0. Definitions.

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

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

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

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

(continues on next page)

(continued from previous page)

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

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

1. Grant of Additional Permission.

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

2. No Weakening of GCC Copyleft.

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

See also

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

11.1.2 GNU GENERAL PUBLIC LICENSE

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

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

                            Preamble

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

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

(continues on next page)

(continued from previous page)

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

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

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

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

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

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

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

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

(continues on next page)

(continued from previous page)

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

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

                    TERMS AND CONDITIONS

  0. Definitions.

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

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

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

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

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

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

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

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

  1. Source Code.

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

(continues on next page)

(continued from previous page)

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

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

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

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

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

  2. Basic Permissions.

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

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

(continues on next page)

(continued from previous page)

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

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

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

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

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

  4. Conveying Verbatim Copies.

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


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

  5. Conveying Modified Source Versions.

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

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

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

(continues on next page)

(continued from previous page)

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

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

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

  6. Conveying Non-Source Forms.

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

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

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

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

(continues on next page)

(continued from previous page)

```
   with subsection 6b.

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

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

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

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

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

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

(continues on next page)

(continued from previous page)

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

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

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

  7. Additional Terms.

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

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

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

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

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

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

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

(continues on next page)

(continued from previous page)

```
   authors of the material; or

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

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

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

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

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

  8. Termination.

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

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

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

(continues on next page)

(continued from previous page)

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

  9. Acceptance Not Required for Having Copies.

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

  10. Automatic Licensing of Downstream Recipients.

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

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

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

  11. Patents.

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

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
```

(continues on next page)

```
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

AddCofactorCode
    OESpruce::OEDesignUnitSplitOptions, 50
AddExcipientCode
    OESpruce::OEDesignUnitSplitOptions, 50
AddHeterogenMetadata
    OESpruce::OEStructureMetadata, 74
AddKeyword
    OESpruce::OEStructureMetadata, 74
AddKeywords
    OESpruce::OEStructureMetadata, 74
AddLipidCode
    OESpruce::OEDesignUnitSplitOptions, 50
AddMutation
    OESpruce::OEDesignUnitMutationOptions, 47
AddSequenceMetadata
    OESpruce::OEStructureMetadata, 74
AddSiteResidue
    OESpruce::OESuperpositionOptions, 84
AddTautomer
    OESpruce::OEHeterogenMetadata, 54
AddTautomers
    OESpruce::OEHeterogenMetadata, 54

C

Clear
    OESpruce::OESuperpose, 88
ClearCofactorCodes
    OESpruce::OEDesignUnitSplitOptions, 51
ClearExcipientCodes
    OESpruce::OEDesignUnitSplitOptions, 51
ClearHeterogenMetadata
    OESpruce::OEStructureMetadata, 74
ClearKeywords
    OESpruce::OEStructureMetadata, 74
ClearLigandConstraints
    OESpruce::OESuperpositionOptions, 85
ClearLipidCodes
    OESpruce::OEDesignUnitSplitOptions, 51
ClearMutations
    OESpruce::OEDesignUnitMutationOptions, 47
ClearSequenceMetadata
    OESpruce::OEStructureMetadata, 74
ClearSiteResidues
    OESpruce::OESuperpositionOptions, 85
ClearTautomers
    OESpruce::OEHeterogenMetadata, 54
ConstraintsMet
    OESpruce::OEStructuralSuperposition, 83
Constructors
    OESpruce::OEBioUnitExtractionOptions, 39
    OESpruce::OECapBuilderOptions, 41
    OESpruce::OEDesignUnitBuildOptions, 42
    OESpruce::OEDesignUnitEnumerateSitesOptions, 44
    OESpruce::OEDesignUnitMutationOptions, 46
    OESpruce::OEDesignUnitPrepOptions, 47
    OESpruce::OEDesignUnitSplitOptions, 50
    OESpruce::OEHeterogenMetadata, 54
    OESpruce::OEIsModeledAtom, 56
    OESpruce::OELoopBuilderOptions, 57
    OESpruce::OEMakeDesignUnitOptions, 63
    OESpruce::OEPocket, 66
    OESpruce::OEPocketOptions, 67
    OESpruce::OEProtonateDesignUnitOptions, 68
    OESpruce::OESecondaryStructureSuperposition, 77
    OESpruce::OESequenceMetadata, 72
    OESpruce::OESidechainBuilderOptions, 70
    OESpruce::OESpruceFilter, 79
    OESpruce::OESpruceFilterOptions, 80
    OESpruce::OEStructuralSuperposition, 82
    OESpruce::OEStructureMetadata, 73
    OESpruce::OESuperpose, 88
    OESpruce::OESuperposeOptions, 90
    OESpruce::OESuperposeResults, 94
    OESpruce::OESuperpositionOptions, 84
    OESpruce::OEValidateDesignUnit, 97
CreateCopy
    OESpruce::OEIsModeledAtom, 56

G

GetAddInteractionHints

OESpruce::OEDesignUnitEnumerateSitesOptions, 44
GetAddStyle
OESpruce::OEDesignUnitEnumerateSitesOptions, 45
GetAllowBuildDisulfideBridges
OESpruce::OELoopBuilderOptions, 59
GetAllowTruncate
OESpruce::OECapBuilderOptions, 41
GetAlternateLocationHandling
OESpruce::OEDesignUnitSplitOptions, 51
GetAssignPartialChargesAndRadii
OESpruce::OEDesignUnitPrepOptions, 48
GetAtoms
OESpruce::OEPocket, 66
OESpruce::OESpruceFilter, 80
GetAuthor
OESpruce::OEStructureMetadata, 75
GetBackboneClashRejectThreshold
OESpruce::OELoopBuilderOptions, 58
GetBioUnitExtractionOptions
OESpruce::OEMakeDesignUnitOptions, 64
GetBuildLoops
OESpruce::OEDesignUnitBuildOptions, 42
GetBuildOptions
OESpruce::OEDesignUnitPrepOptions, 48
GetBuildSidechains
OESpruce::OEDesignUnitBuildOptions, 42
GetBuildTails
OESpruce::OELoopBuilderOptions, 60
GetBurialFactor
OESpruce::OEPocketOptions, 67
GetCapBuilderOptions
OESpruce::OEDesignUnitBuildOptions, 43
GetCapCTermini
OESpruce::OEDesignUnitBuildOptions, 42
GetCapNTermini
OESpruce::OEDesignUnitBuildOptions, 43
GetChainID
OESpruce::OESequenceMetadata, 72
GetChargeEngine
OESpruce::OEDesignUnitPrepOptions, 48
GetCofactorCodes
OESpruce::OEDesignUnitSplitOptions, 51
GetCollapseNonSiteAlts
OESpruce::OEDesignUnitEnumerateSitesOptions, 45
GetConnectBufferDistance
OESpruce::OELoopBuilderOptions, 57
GetCropLength
OESpruce::OELoopBuilderOptions, 57
GetDeleteClashingSolvent
OESpruce::OECapBuilderOptions, 41
OESpruce::OESidechainBuilderOptions, 70
GetDeleteExistingResidues
OESpruce::OELoopBuilderOptions, 58
GetDeleteFloatingRes
OESpruce::OESpruceFilterOptions, 80
GetDist
OESpruce::OESuperpositionOptions, 85
GetDuplicateRemoval
OESpruce::OEDesignUnitEnumerateSitesOptions, 45
GetEndResidueInsertCode
OESpruce::OESequenceMetadata, 72
GetEndResidueNumber
OESpruce::OESequenceMetadata, 72
GetEnumerateCofactorSites
OESpruce::OEDesignUnitEnumerateSitesOptions, 45
GetEnumerateSitesOptions
OESpruce::OEDesignUnitPrepOptions, 48
GetExcipientCodes
OESpruce::OEDesignUnitSplitOptions, 51
GetExperimentDate
OESpruce::OEStructureMetadata, 75
GetExperimentType
OESpruce::OEStructureMetadata, 75
GetFitChains
OESpruce::OESecondaryStructureSuperposition, 77
OESpruce::OEStructuralSuperposition, 83
OESpruce::OESuperposeResults, 97
GetFixBackboneAtoms
OESpruce::OESpruceFilterOptions, 80
GetFixBondsToMetals
OESpruce::OESpruceFilterOptions, 80
GetFixChainIDs
OESpruce::OESpruceFilterOptions, 80
GetFixInvalidCovalentBonds
OESpruce::OESpruceFilterOptions, 81
GetFixNames
OESpruce::OESpruceFilterOptions, 81
GetFixResStates
OESpruce::OESpruceFilterOptions, 81
GetForceCapping
OESpruce::OECapBuilderOptions, 41
GetGenerateTautomers
OESpruce::OEProtonateDesignUnitOptions, 68
GetHeterogenMetadata
OESpruce::OEStructureMetadata, 75
GetHetGroupNbrDist
OESpruce::OEProtonateDesignUnitOptions, 68
GetID
OESpruce::OEHeterogenMetadata, 54
GetIridiumData

OESpruce::OEStructureMetadata, 75
GetKeywords
OESpruce::OEStructureMetadata, 75
GetLigandConstraints
OESpruce::OESuperpositionOptions, 85
GetLipidCodes
OESpruce::OEDesignUnitSplitOptions, 51
GetLocalDensity
OESpruce::OEPocket, 66
GetLoopBuilderOptions
OESpruce::OEDesignUnitBuildOptions, 43
GetLoopClashRejectThreshold
OESpruce::OELoopBuilderOptions, 58
GetLoopDBFilename
OESpruce::OELoopBuilderOptions, 57
GetMakePackingResidues
OESpruce::OEDesignUnitSplitOptions, 51
GetMaxAtoms
OESpruce::OEBioUnitExtractionOptions, 39
GetMaxClusterSize
OESpruce::OESidechainBuilderOptions, 70
GetMaxLigAtoms
OESpruce::OEDesignUnitSplitOptions, 52
GetMaxLigResidues
OESpruce::OEDesignUnitSplitOptions, 52
GetMaxParts
OESpruce::OEBioUnitExtractionOptions, 39
GetMaxSurfaceArea
OESpruce::OEPocketOptions, 67
GetMaxSystemAtoms
OESpruce::OEDesignUnitSplitOptions, 52
GetMessages
OESpruce::OESpruceFilter, 79
OESpruce::OEValidateDesignUnit, 98
GetMethod
OESpruce::OESuperposeOptions, 90
GetMinimizeSidechains
OESpruce::OESidechainBuilderOptions, 70
GetMinimizeSidechainsShell
OESpruce::OESidechainBuilderOptions, 70
GetMinLigAtoms
OESpruce::OEDesignUnitSplitOptions, 52
GetMinScore
OESpruce::OEBioUnitExtractionOptions, 39
GetMinSurfaceArea
OESpruce::OEPocketOptions, 67
GetMutations
OESpruce::OEDesignUnitMutationOptions, 47
GetOptimizationInclSurfaceAreaTerm
OESpruce::OELoopBuilderOptions, 59
GetOptimizationMaxLoops
OESpruce::OELoopBuilderOptions, 59
GetOptimizationShell
OESpruce::OELoopBuilderOptions, 58
GetOptimizationStage1IterMultiplier
OESpruce::OELoopBuilderOptions, 58
GetOptimizationStage2IterMultiplier
OESpruce::OELoopBuilderOptions, 58
GetOptimizationTolerance
OESpruce::OELoopBuilderOptions, 58
GetOptimizationUseSolventModel
OESpruce::OELoopBuilderOptions, 59
GetOptimizeExpProtons
OESpruce::OEProtonateDesignUnitOptions, 68
GetPlaceHydrogensOptions
OESpruce::OEProtonateDesignUnitOptions, 69
GetPocketPredicate
OESpruce::OEPocket, 66
GetPocketSurface
OESpruce::OEPocket, 66
GetPreferAuthorRecord
OESpruce::OEBioUnitExtractionOptions, 40
GetPrepOptions
OESpruce::OEMakeDesignUnitOptions, 64
GetProblematicResidues
OESpruce::OEValidateDesignUnit, 98
GetProtonate
OESpruce::OEDesignUnitPrepOptions, 48
GetProtonateOptions
OESpruce::OEDesignUnitPrepOptions, 48
GetRefChains
OESpruce::OESecondaryStructureSuperposition, 78
OESpruce::OEStructuralSuperposition, 83
OESpruce::OESuperposeResults, 96
GetRegion
OESpruce::OEStructuralSuperposition, 83
GetRejectClashingLoopsPostMinimization
OESpruce::OELoopBuilderOptions, 59
GetResidues
OESpruce::OEPocket, 66
GetRestrictToRefSite
OESpruce::OEDesignUnitEnumerateSitesOptions, 45
GetRevision
OESpruce::OEStructureMetadata, 75
GetRevisionDate
OESpruce::OEStructureMetadata, 75
GetRMSD
OESpruce::OEStructuralSuperposition, 83
OESpruce::OESuperposeResults, 96
GetRotamerLibrary
OESpruce::OESidechainBuilderOptions, 70
GetRotMatrix
OESpruce::OESecondaryStructureSuperposition, 78

OESpruce::OEStructuralSuperposition, 83
OESpruce::OESuperposeResults, 95
GetScore
OESpruce::OEStructuralSuperposition, 84
GetSeqAlignExtendPenalty
OESpruce::OELoopBuilderOptions, 59
GetSeqAlignGapPenalty
OESpruce::OELoopBuilderOptions, 59
GetSeqAlignMethod
OESpruce::OELoopBuilderOptions, 59
GetSeqScore
OESpruce::OESuperposeResults, 96
GetSequence
OESpruce::OESequenceMetadata, 72
GetSequenceMetadata
OESpruce::OEStructureMetadata, 76
GetSidechainBuilderOptions
OESpruce::OEDesignUnitBuildOptions, 43
GetSiteResidues
OESpruce::OESuperpositionOptions, 86
GetSiteSize
OESpruce::OEDesignUnitEnumerateSitesOptions, 45
GetSmiles
OESpruce::OEHeterogenMetadata, 54
GetSplitOptions
OESpruce::OEMakeDesignUnitOptions, 64
GetStartResidueInsertCode
OESpruce::OESequenceMetadata, 72
GetStartResidueNumber
OESpruce::OESequenceMetadata, 72
GetStrictMode
OESpruce::OESidechainBuilderOptions, 70
GetStrictProlineMatch
OESpruce::OELoopBuilderOptions, 57
GetStrictProtonationMode
OESpruce::OEDesignUnitPrepOptions, 48
GetStructureID
OESpruce::OEStructureMetadata, 76
GetSuperpose
OESpruce::OEBioUnitExtractionOptions, 40
OESpruce::OEMakeDesignUnitOptions, 64
GetSuperpositionMethod
OESpruce::OEMakeDesignUnitOptions, 64
GetSuperpositionType
OESpruce::OESuperpositionOptions, 86
GetSurfaceArea
OESpruce::OEPocket, 66
GetTanimoto
OESpruce::OESecondaryStructureSuperposition, 78
OESpruce::OESuperposeResults, 96
GetTargetComponentID
OESpruce::OEDesignUnitSplitOptions, 52
GetTautomers
OESpruce::OEHeterogenMetadata, 55
GetTitle
OESpruce::OEHeterogenMetadata, 55
GetTransform
OESpruce::OESuperposeResults, 95
GetTransformThreshold
OESpruce::OELoopBuilderOptions, 57
GetTranslation
OESpruce::OESecondaryStructureSuperposition, 78
OESpruce::OEStructuralSuperposition, 84
OESpruce::OESuperposeResults, 95
GetType
OESpruce::OEHeterogenMetadata, 55
GetUseConstraints
OESpruce::OESuperposeOptions, 90
GetUsePackingResidues
OESpruce::OELoopBuilderOptions, 57
GetUseSeqScoreChainMatching
OESpruce::OESuperposeOptions, 93
GetValidRMSD
OESpruce::OESuperposeOptions, 91
GetValidSeqScore
OESpruce::OESuperposeOptions, 92
GetValidTanimoto
OESpruce::OESuperposeOptions, 92

H

HasConstraintMol
OESpruce::OESuperpose, 88
HasLigandConstraints
OESpruce::OESuperpositionOptions, 86
HasRef
OESpruce::OESuperpose, 88
HasRMSD
OESpruce::OESuperposeResults, 96
HasSeqScore
OESpruce::OESuperposeResults, 96
HasTanimoto
OESpruce::OESuperposeResults, 96

I

IsValid
OESpruce::OESuperposeResults, 94

O

OESpruce::OEAlternateLocationOption, 98
OESpruce::OEAlternateLocationOption::Combinatorial, 98
OESpruce::OEAlternateLocationOption::Default, 98
OESpruce::OEAlternateLocationOption::Enumerate, 98

OESpruce::OEAlternateLocationOption::Max, 99
OESpruce::OEAlternateLocationOption::Primary, 98
OESpruce::OEAlternateLocationOption::Unknown, 98
OESpruce::OEBioUnitExtractionOptions, 39
  Constructors, 39
  GetMaxAtoms, 39
  GetMaxParts, 39
  GetMinScore, 39
  GetPreferAuthorRecord, 40
  GetSuperpose, 40
  SetMaxAtoms, 40
  SetMaxParts, 40
  SetMinScore, 40
  SetPreferAuthorRecord, 40
  SetSuperpose, 40
OESpruce::OEBuildLoops, 106
OESpruce::OEBuildSidechains, 107
OESpruce::OEBuildSingleLoop, 108
OESpruce::OECapBuilderOptions, 40
  Constructors, 41
  GetAllowTruncate, 41
  GetDeleteClashingSolvent, 41
  GetForceCapping, 41
  SetAllowTruncate, 41
  SetDeleteClashingSolvent, 41
  SetForceCapping, 41
OESpruce::OECapCTermini, 109
OESpruce::OECapNTermini, 110
OESpruce::OECapTermini, 111
OESpruce::OEDesignUnitBuildOptions, 42
  Constructors, 42
  GetBuildLoops, 42
  GetBuildSidechains, 42
  GetCapBuilderOptions, 43
  GetCapCTermini, 42
  GetCapNTermini, 43
  GetLoopBuilderOptions, 43
  GetSidechainBuilderOptions, 43
  SetBuildLoops, 43
  SetBuildSidechains, 43
  SetCapBuilderOptions, 44
  SetCapCTermini, 43
  SetCapNTermini, 43
  SetLoopBuilderOptions, 44
  SetSidechainBuilderOptions, 44
OESpruce::OEDesignUnitEnumerateSitesOptions, 44
  Constructors, 44
  GetAddInteractionHints, 44
  GetAddStyle, 45
  GetCollapseNonSiteAlts, 45
  GetDuplicateRemoval, 45
  GetEnumerateCofactorSites, 45
  GetRestrictToRefSite, 45
  GetSiteSize, 45
  SetAddInteractionHints, 45
  SetAddStyle, 45
  SetCollapseNonSiteAlts, 45
  SetDuplicateRemoval, 46
  SetEnumerateCofactorSites, 46
  SetRestrictToRefSite, 46
  SetSiteSize, 46
OESpruce::OEDesignUnitIssueCodes, 99
OESpruce::OEDesignUnitIssueCodes::All, 100
OESpruce::OEDesignUnitIssueCodes::BrokenChains, 99
OESpruce::OEDesignUnitIssueCodes::HeavyAtomsOverlap, 99
OESpruce::OEDesignUnitIssueCodes::ImplicitH, 99
OESpruce::OEDesignUnitIssueCodes::MissingLoop, 99
OESpruce::OEDesignUnitIssueCodes::PartialSideChain, 99
OESpruce::OEDesignUnitIssueCodes::Success, 99
OESpruce::OEDesignUnitIssueCodes::TerminiNotCapped, 99
OESpruce::OEDesignUnitMutationOptions, 46
  AddMutation, 47
  ClearMutations, 47
  Constructors, 46
  GetMutations, 47
  SetMutations, 47
OESpruce::OEDesignUnitPrepOptions, 47
  Constructors, 47
  GetAssignPartialChargesAndRadii, 48
  GetBuildOptions, 48
  GetChargeEngine, 48
  GetEnumerateSitesOptions, 48
  GetProtonate, 48
  GetProtonateOptions, 48
  GetStrictProtonationMode, 48
  SetAssignPartialChargesAndRadii, 49
  SetBuildOptions, 49
  SetChargeEngine, 49
  SetEnumerateSitesOptions, 49
  SetProtonate, 49
  SetProtonateOptions, 49
  SetStrictProtonationMode, 49
OESpruce::OEDesignUnitSplitOptions, 50
  AddCofactorCode, 50
  AddExcipientCode, 50
  AddLipidCode, 50
  ClearCofactorCodes, 51
  ClearExcipientCodes, 51
  ClearLipidCodes, 51

Constructors, 50
GetAlternateLocationHandling, 51
GetCofactorCodes, 51
GetExcipientCodes, 51
GetLipidCodes, 51
GetMakePackingResidues, 51
GetMaxLigAtoms, 52
GetMaxLigResidues, 52
GetMaxSystemAtoms, 52
GetMinLigAtoms, 52
GetTargetComponentID, 52
SetAlternateLocationHandling, 52
SetCofactorCodes, 52
SetExcipientCodes, 52
SetLipidCodes, 52
SetMakePackingResidues, 53
SetMaxLigAtoms, 53
SetMaxLigResidues, 53
SetMaxSystemAtoms, 53
SetMinLigAtoms, 53
SetTargetComponentID, 53
OESpruce::OEEnumerateSites, 112
OESpruce::OEExperimentType, 100
OESpruce::OEExperimentType::ElectronCrystallography, 100
OESpruce::OEExperimentType::ElectronMicroscopy, 100
OESpruce::OEExperimentType::FiberDiffraction, 100
OESpruce::OEExperimentType::NeutronDiffraction, 100
OESpruce::OEExperimentType::PowderDiffraction, 100
OESpruce::OEExperimentType::SolidStateNMR, 100
OESpruce::OEExperimentType::SolutionNMR, 100
OESpruce::OEExperimentType::SolutionScattering, 101
OESpruce::OEExperimentType::TheoreticalModel, 101
OESpruce::OEExperimentType::Unknown, 101
OESpruce::OEExperimentType::XRayDiffraction, 101
OESpruce::OEExtractBioUnits, 113
OESpruce::OEFindPockets, 114
OESpruce::OEFixBackbone, 115
OESpruce::OEGetAlternateLocationOptionID, 115
OESpruce::OEGetAlternateLocationOptionName, 115
OESpruce::OEGetExperimentTypeID, 115
OESpruce::OEGetExperimentTypeName, 116
OESpruce::OEGetHeterogenTypeID, 116
OESpruce::OEGetHeterogenTypeName, 116
OESpruce::OEGetPartialResidues, 116
OESpruce::OEGetSuperposeMethodFromName, 130
OESpruce::OEGetSuperposeMethodName, 131
OESpruce::OEHeterogenMetadata, 53
AddTautomer, 54
AddTautomers, 54
ClearTautomers, 54
Constructors, 54
GetID, 54
GetSmiles, 54
GetTautomers, 55
GetTitle, 55
GetType, 55
SetID, 55
SetSmiles, 55
SetTitle, 55
SetType, 55
OESpruce::OEHeterogenMetadataFromJson, 117
OESpruce::OEHeterogenMetadataToJson, 118
OESpruce::OEHeterogenType, 101
OESpruce::OEHeterogenType::Cofactors, 101
OESpruce::OEHeterogenType::CounterIons, 101
OESpruce::OEHeterogenType::Excipients, 101
OESpruce::OEHeterogenType::Ligand, 101
OESpruce::OEHeterogenType::Lipids, 102
OESpruce::OEHeterogenType::Metals, 102
OESpruce::OEHeterogenType::Polymers, 102
OESpruce::OEHeterogenType::Solvent, 102
OESpruce::OEHeterogenType::Sugars, 102
OESpruce::OEHeterogenType::Unknown, 102
OESpruce::OEIsModeledAtom, 55
Constructors, 56
CreateCopy, 56
operator(), 56
OESpruce::OEIsRedundantDesignUnit, 118
OESpruce::OELoopBuilderOptions, 56
Constructors, 57
GetAllowBuildDisulfideBridges, 59
GetBackboneClashRejectThreshold, 58
GetBuildTails, 60
GetConnectBufferDistance, 57
GetCropLength, 57
GetDeleteExistingResidues, 58
GetLoopClashRejectThreshold, 58
GetLoopDBFilename, 57
GetOptimizationInclSurfaceAreaTerm, 59
GetOptimizationMaxLoops, 59
GetOptimizationShell, 58
GetOptimizationStage1IterMultiplier, 58
GetOptimizationStage2IterMultiplier, 58
GetOptimizationTolerance, 58
GetOptimizationUseSolventModel, 59
GetRejectClashingLoopsPostMinimization, 59
GetSeqAlignExtendPenalty, 59

GetSeqAlignGapPenalty, 59
GetSeqAlignMethod, 59
GetStrictProlineMatch, 57
GetTransformThreshold, 57
GetUsePackingResidues, 57
SetAllowBuildDisulfideBridges, 62
SetBackboneClashRejectThreshold, 60
SetBuildTails, 62
SetConnectBufferDistance, 60
SetCropLength, 60
SetDeleteExistingResidues, 61
SetLoopClashRejectThreshold, 61
SetLoopDBFilename, 60
SetOptimizationInclSurfaceAreaTerm, 61
SetOptimizationMaxLoops, 62
SetOptimizationShell, 61
SetOptimizationStage1IterMultiplier, 61
SetOptimizationStage2IterMultiplier, 61
SetOptimizationTolerance, 61
SetOptimizationUseSolventModel, 61
SetRejectClashingLoopsPostMinimization, 62
SetSeqAlignExtendPenalty, 62
SetSeqAlignGapPenalty, 62
SetSeqAlignMethod, 62
SetStrictProlineMatch, 60
SetTransformThreshold, 60
SetUsePackingResidues, 60
OESpruce::OEMakeBioDesignUnits, 119
OESpruce::OEMakeDesignUnit, 120
OESpruce::OEMakeDesignUnitFromPocket, 122
OESpruce::OEMakeDesignUnitOptions, 63
Constructors, 63
GetBioUnitExtractionOptions, 64
GetPrepOptions, 64
GetSplitOptions, 64
GetSuperpose, 64
GetSuperpositionMethod, 64
SetBioUnitExtractionOptions, 64
SetPrepOptions, 64
SetSplitOptions, 65
SetSuperpose, 65
SetSuperpositionMethod, 65
OESpruce::OEMakeDesignUnits, 121
OESpruce::OEMutateResidue, 123
OESpruce::OEMutateResidues, 124
OESpruce::OEPocket, 65
Constructors, 66
GetAtoms, 66
GetLocalDensity, 66
GetPocketPredicate, 66
GetPocketSurface, 66
GetResidues, 66
GetSurfaceArea, 66
OESpruce::OEPocketOptions, 67
Constructors, 67
GetBurialFactor, 67
GetMaxSurfaceArea, 67
GetMinSurfaceArea, 67
SetBurialFactor, 68
SetMaxSurfaceArea, 67
SetMinSurfaceArea, 67
OESpruce::OEProtonateDesignUnit, 124
OESpruce::OEProtonateDesignUnitOptions, 68
Constructors, 68
GetGenerateTautomers, 68
GetHetGroupNbrDist, 68
GetOptimizeExpProtons, 68
GetPlaceHydrogensOptions, 69
SetGenerateTautomers, 69
SetHetGroupNbrDist, 69
SetOptimizeExpProtons, 69
SetPlaceHydrogensOptions, 69
OESpruce::OESecondaryStructureSuperposition, 77
Constructors, 77
GetFitChains, 77
GetRefChains, 78
GetRotMatrix, 78
GetTanimoto, 78
GetTranslation, 78
operator bool, 77
Transform, 78
OESpruce::OESequenceMetadata, 72
Constructors, 72
GetChainID, 72
GetEndResidueInsertCode, 72
GetEndResidueNumber, 72
GetSequence, 72
GetStartResidueInsertCode, 72
GetStartResidueNumber, 72
SetChainID, 73
SetEndResidueInsertCode, 73
SetEndResidueNumber, 73
SetSequence, 73
SetStartResidueInsertCode, 73
SetStartResidueNumber, 73
OESpruce::OESequenceMetadataFromJson, 127
OESpruce::OESequenceMetadataToJson, 127
OESpruce::OESidechainBuilderOptions, 69
Constructors, 70
GetDeleteClashingSolvent, 70
GetMaxClusterSize, 70
GetMinimizeSidechains, 70
GetMinimizeSidechainsShell, 70
GetRotamerLibrary, 70
GetStrictMode, 70
SetDeleteClashingSolvent, 71

SetMaxClusterSize, 71
SetMinimizeSidechains, 71
SetMinimizeSidechainsShell, 71
SetRotamerCoverage, 70
SetRotamerLibrary, 71
SetStrictMode, 71
OESpruce::OESplitBioUnitToComponents, 125
OESpruce::OESpruceFilter, 78
Constructors, 79
GetAtoms, 80
GetMessages, 79
StandardizeAndFilter, 79
OESpruce::OESpruceFilterIssueCodes, 103
OESpruce::OESpruceFilterIssueCodes::All, 104
OESpruce::OESpruceFilterIssueCodes::InvalidCRYST1, 104
OESpruce::OESpruceFilterIssueCodes::InvalidElectronDensityQuerMat, 104
OESpruce::OESpruceFilterIssueCodes::InvalidMetalBond, 104
OESpruce::OESpruceFilterIssueCodes::InvalidNames, 104
OESpruce::OESpruceFilterIssueCodes::InvalidResidueTemplate, 104
OESpruce::OESpruceFilterIssueCodes::InvalidSeqAlign, 104
OESpruce::OESpruceFilterIssueCodes::Success, 104
OESpruce::OESpruceFilterOptions, 80
Constructors, 80
GetDeleteFloatingRes, 80
GetFixBackboneAtoms, 80
GetFixBondsToMetals, 80
GetFixChainIDs, 80
GetFixInvalidCovalentBonds, 81
GetFixNames, 81
GetFixResStates, 81
SetDeleteFloatingRes, 81
SetFixBackboneAtoms, 81
SetFixBondsToMetals, 81
SetFixChainIDs, 81
SetFixInvalidCovalentBonds, 81
SetFixNames, 82
SetFixResStates, 82
OESpruce::OESpruceGetArch, 126
OESpruce::OESpruceGetLicensee, 126
OESpruce::OESpruceGetPlatform, 126
OESpruce::OESpruceGetRelease, 126
OESpruce::OESpruceGetSite, 126
OESpruce::OESpruceGetVersion, 126
OESpruce::OESpruceIsLicensed, 126
OESpruce::OEStructuralSuperposition, 82
ConstraintsMet, 83
Constructors, 82
GetFitChains, 83
GetRefChains, 83
GetRegion, 83
GetRMSD, 83
GetRotMatrix, 83
GetScore, 84
GetTranslation, 84
operator bool, 83
Transform, 84
OESpruce::OEStructureMetadata, 73
AddHeterogenMetadata, 74
AddKeyword, 74
AddKeywords, 74
AddSequenceMetadata, 74
ClearHeterogenMetadata, 74
ClearKeywords, 74
ClearSequenceMetadata, 74
Constructors, 73
GetAuthor, 75
GetExperimentDate, 75
GetExperimentType, 75
GetHeterogenMetadata, 75
GetIridiumData, 75
GetKeywords, 75
GetRevision, 75
GetRevisionDate, 75
GetSequenceMetadata, 76
GetStructureID, 76
SetAuthor, 76
SetExperimentDate, 76
SetExperimentType, 76
SetIridiumData, 76
SetRevision, 76
SetRevisionDate, 76
SetStructureID, 76
OESpruce::OEStructureMetadataFromJson, 128
OESpruce::OEStructureMetadataToJson, 129
OESpruce::OESuperpose, 87
Clear, 88
Constructors, 88
HasConstraintMol, 88
HasRef, 88
SetMethod, 89
SetOptions, 88
SetupRef, 88
Superpose, 89
OESpruce::OESuperposeMethod, 104
OESpruce::OESuperposeMethod::DDM, 105
OESpruce::OESuperposeMethod::Default, 105
OESpruce::OESuperposeMethod::DifferenceDistanceMatrix, 105
OESpruce::OESuperposeMethod::Global, 105
OESpruce::OESuperposeMethod::GlobalCarbonAlpha, 105

OESpruce::OESuperposeMethod::SecondaryStructureElementSu, 106
OESpruce::OESuperposeMethod::Site, 105
OESpruce::OESuperposeMethod::SiteHopper, 106
OESpruce::OESuperposeMethod::SiteSequence, 105
OESpruce::OESuperposeMethod::SSE, 106
OESpruce::OESuperposeMethod::Undefined, 106
OESpruce::OESuperposeMethod::Weighted, 106
OESpruce::OESuperposeMethod::WeightedDifferenceSmarts, 106
OESpruce::OESuperposeOptions, 90
    Constructors, 90
    GetMethod, 90
    GetUseConstraints, 90
    GetUseSeqScoreChainMatching, 93
    GetValidRMSD, 91
    GetValidSeqScore, 92
    GetValidTanimoto, 92
    SetMethod, 90
    SetUseConstraints, 91
    SetUseSeqScoreChainMatching, 93
    SetValidRMSD, 91
    SetValidSeqScore, 93
    SetValidTanimoto, 92
OESpruce::OESuperposeResults, 94
    Constructors, 94
    GetFitChains, 97
    GetRefChains, 96
    GetRMSD, 96
    GetRotMatrix, 95
    GetSeqScore, 96
    GetTanimoto, 96
    GetTransform, 95
    GetTranslation, 95
    HasRMSD, 96
    HasSeqScore, 96
    HasTanimoto, 96
    IsValid, 94
    operator bool, 94
    Transform, 95
OESpruce::OESuperpositionOptions, 84
    AddSiteResidue, 84
    ClearLigandConstraints, 85
    ClearSiteResidues, 85
    Constructors, 84
    GetDist, 85
    GetLigandConstraints, 85
    GetSiteResidues, 86
    GetSuperpositionType, 86
    HasLigandConstraints, 86
    SetDist, 86
    SetLigandConstraints, 86
    SetSiteResidues, 87
    SetSuperpositionType, 87
OESpruce::OESuperpositionType, 102
OESpruce::OESuperpositionType::DDM, 103
OESpruce::OESuperpositionType::Default, 102
OESpruce::OESuperpositionType::Global, 103
OESpruce::OESuperpositionType::Site, 103
OESpruce::OESuperpositionType::SSE, 103
OESpruce::OESuperpositionType::Undefined, 103
OESpruce::OESuperpositionType::Weighted, 103
OESpruce::OEValidateDesignUnit, 97
    Constructors, 97
    GetMessages, 98
    GetProblematicResidues, 98
    Validate, 97
operator bool
    OESpruce::OESecondaryStructureSuperposition, 77
    OESpruce::OEStructuralSuperposition, 83
    OESpruce::OESuperposeResults, 94
operator()
    OESpruce::OEIsModeledAtom, 56

S

SetAddInteractionHints
    OESpruce::OEDesignUnitEnumerateSitesOptions, 45
SetAddStyle
    OESpruce::OEDesignUnitEnumerateSitesOptions, 45
SetAllowBuildDisulfideBridges
    OESpruce::OELoopBuilderOptions, 62
SetAllowTruncate
    OESpruce::OECapBuilderOptions, 41
SetAlternateLocationHandling
    OESpruce::OEDesignUnitSplitOptions, 52
SetAssignPartialChargesAndRadii
    OESpruce::OEDesignUnitPrepOptions, 49
SetAuthor
    OESpruce::OEStructureMetadata, 76
SetBackboneClashRejectThreshold
    OESpruce::OELoopBuilderOptions, 60
SetBioUnitExtractionOptions
    OESpruce::OEMakeDesignUnitOptions, 64
SetBuildLoops
    OESpruce::OEDesignUnitBuildOptions, 43
SetBuildOptions
    OESpruce::OEDesignUnitPrepOptions, 49
SetBuildSidechains
    OESpruce::OEDesignUnitBuildOptions, 43
SetBuildTails
    OESpruce::OELoopBuilderOptions, 62
SetBurialFactor
    OESpruce::OEPocketOptions, 68
SetCapBuilderOptions

OESpruce::OEDesignUnitBuildOptions, 44
SetCapCTermini
OESpruce::OEDesignUnitBuildOptions, 43
SetCapNTermini
OESpruce::OEDesignUnitBuildOptions, 43
SetChainID
OESpruce::OESequenceMetadata, 73
SetChargeEngine
OESpruce::OEDesignUnitPrepOptions, 49
SetCofactorCodes
OESpruce::OEDesignUnitSplitOptions, 52
SetCollapseNonSiteAlts
OESpruce::OEDesignUnitEnumerateSitesOptions, 45
SetConnectBufferDistance
OESpruce::OELoopBuilderOptions, 60
SetCropLength
OESpruce::OELoopBuilderOptions, 60
SetDeleteClashingSolvent
OESpruce::OECapBuilderOptions, 41
OESpruce::OESidechainBuilderOptions, 71
SetDeleteExistingResidues
OESpruce::OELoopBuilderOptions, 61
SetDeleteFloatingRes
OESpruce::OESpruceFilterOptions, 81
SetDist
OESpruce::OESuperpositionOptions, 86
SetDuplicateRemoval
OESpruce::OEDesignUnitEnumerateSitesOptions, 46
SetEndResidueInsertCode
OESpruce::OESequenceMetadata, 73
SetEndResidueNumber
OESpruce::OESequenceMetadata, 73
SetEnumerateCofactorSites
OESpruce::OEDesignUnitEnumerateSitesOptions, 46
SetEnumerateSitesOptions
OESpruce::OEDesignUnitPrepOptions, 49
SetExcipientCodes
OESpruce::OEDesignUnitSplitOptions, 52
SetExperimentDate
OESpruce::OEStructureMetadata, 76
SetExperimentType
OESpruce::OEStructureMetadata, 76
SetFixBackboneAtoms
OESpruce::OESpruceFilterOptions, 81
SetFixBondsToMetals
OESpruce::OESpruceFilterOptions, 81
SetFixChainIDs
OESpruce::OESpruceFilterOptions, 81
SetFixInvalidCovalentBonds
OESpruce::OESpruceFilterOptions, 81
SetFixNames
OESpruce::OESpruceFilterOptions, 82
SetFixResStates
OESpruce::OESpruceFilterOptions, 82
SetForceCapping
OESpruce::OECapBuilderOptions, 41
SetGenerateTautomers
OESpruce::OEProtonateDesignUnitOptions, 69
SetHetGroupNbrDist
OESpruce::OEProtonateDesignUnitOptions, 69
SetID
OESpruce::OEHeterogenMetadata, 55
SetIridiumData
OESpruce::OEStructureMetadata, 76
SetLigandConstraints
OESpruce::OESuperpositionOptions, 86
SetLipidCodes
OESpruce::OEDesignUnitSplitOptions, 52
SetLoopBuilderOptions
OESpruce::OEDesignUnitBuildOptions, 44
SetLoopClashRejectThreshold
OESpruce::OELoopBuilderOptions, 61
SetLoopDBFilename
OESpruce::OELoopBuilderOptions, 60
SetMakePackingResidues
OESpruce::OEDesignUnitSplitOptions, 53
SetMaxAtoms
OESpruce::OEBioUnitExtractionOptions, 40
SetMaxClusterSize
OESpruce::OESidechainBuilderOptions, 71
SetMaxLigAtoms
OESpruce::OEDesignUnitSplitOptions, 53
SetMaxLigResidues
OESpruce::OEDesignUnitSplitOptions, 53
SetMaxParts
OESpruce::OEBioUnitExtractionOptions, 40
SetMaxSurfaceArea
OESpruce::OEPocketOptions, 67
SetMaxSystemAtoms
OESpruce::OEDesignUnitSplitOptions, 53
SetMethod
OESpruce::OESuperpose, 89
OESpruce::OESuperposeOptions, 90
SetMinimizeSidechains
OESpruce::OESidechainBuilderOptions, 71
SetMinimizeSidechainsShell
OESpruce::OESidechainBuilderOptions, 71
SetMinLigAtoms
OESpruce::OEDesignUnitSplitOptions, 53
SetMinScore
OESpruce::OEBioUnitExtractionOptions, 40
SetMinSurfaceArea
OESpruce::OEPocketOptions, 67

SetMutations
    OESpruce::OEDesignUnitMutationOptions, 47
SetOptimizationInclSurfaceAreaTerm
    OESpruce::OELoopBuilderOptions, 61
SetOptimizationMaxLoops
    OESpruce::OELoopBuilderOptions, 62
SetOptimizationShell
    OESpruce::OELoopBuilderOptions, 61
SetOptimizationStage1IterMultiplier
    OESpruce::OELoopBuilderOptions, 61
SetOptimizationStage2IterMultiplier
    OESpruce::OELoopBuilderOptions, 61
SetOptimizationTolerance
    OESpruce::OELoopBuilderOptions, 61
SetOptimizationUseSolventModel
    OESpruce::OELoopBuilderOptions, 61
SetOptimizeExpProtons
    OESpruce::OEProtonateDesignUnitOptions, 69
SetOptions
    OESpruce::OESuperpose, 88
SetPlaceHydrogensOptions
    OESpruce::OEProtonateDesignUnitOptions, 69
SetPreferAuthorRecord
    OESpruce::OEBioUnitExtractionOptions, 40
SetPrepOptions
    OESpruce::OEMakeDesignUnitOptions, 64
SetProtonate
    OESpruce::OEDesignUnitPrepOptions, 49
SetProtonateOptions
    OESpruce::OEDesignUnitPrepOptions, 49
SetRejectClashingLoopsPostMinimization
    OESpruce::OELoopBuilderOptions, 62
SetRestrictToRefSite
    OESpruce::OEDesignUnitEnumerateSitesOptions, 46
SetRevision
    OESpruce::OEStructureMetadata, 76
SetRevisionDate
    OESpruce::OEStructureMetadata, 76
SetRotamerCoverage
    OESpruce::OESidechainBuilderOptions, 70
SetRotamerLibrary
    OESpruce::OESidechainBuilderOptions, 71
SetSeqAlignExtendPenalty
    OESpruce::OELoopBuilderOptions, 62
SetSeqAlignGapPenalty
    OESpruce::OELoopBuilderOptions, 62
SetSeqAlignMethod
    OESpruce::OELoopBuilderOptions, 62
SetSequence
    OESpruce::OESequenceMetadata, 73
SetSidechainBuilderOptions
    OESpruce::OEDesignUnitBuildOptions, 44
SetSiteResidues
    OESpruce::OESuperpositionOptions, 87
SetSiteSize
    OESpruce::OEDesignUnitEnumerateSitesOptions, 46
SetSmiles
    OESpruce::OEHeterogenMetadata, 55
SetSplitOptions
    OESpruce::OEMakeDesignUnitOptions, 65
SetStartResidueInsertCode
    OESpruce::OESequenceMetadata, 73
SetStartResidueNumber
    OESpruce::OESequenceMetadata, 73
SetStrictMode
    OESpruce::OESidechainBuilderOptions, 71
SetStrictProlineMatch
    OESpruce::OELoopBuilderOptions, 60
SetStrictProtonationMode
    OESpruce::OEDesignUnitPrepOptions, 49
SetStructureID
    OESpruce::OEStructureMetadata, 76
SetSuperpose
    OESpruce::OEBioUnitExtractionOptions, 40
    OESpruce::OEMakeDesignUnitOptions, 65
SetSuperpositionMethod
    OESpruce::OEMakeDesignUnitOptions, 65
SetSuperpositionType
    OESpruce::OESuperpositionOptions, 87
SetTargetComponentID
    OESpruce::OEDesignUnitSplitOptions, 53
SetTitle
    OESpruce::OEHeterogenMetadata, 55
SetTransformThreshold
    OESpruce::OELoopBuilderOptions, 60
SetType
    OESpruce::OEHeterogenMetadata, 55
SetupRef
    OESpruce::OESuperpose, 88
SetUseConstraints
    OESpruce::OESuperposeOptions, 91
SetUsePackingResidues
    OESpruce::OELoopBuilderOptions, 60
SetUseSeqScoreChainMatching
    OESpruce::OESuperposeOptions, 93
SetValidRMSD
    OESpruce::OESuperposeOptions, 91
SetValidSeqScore
    OESpruce::OESuperposeOptions, 93
SetValidTanimoto
    OESpruce::OESuperposeOptions, 92
StandardizeAndFilter
    OESpruce::OESpruceFilter, 79
Superpose

OESpruce::OESuperpose, 89

T

Transform
OESpruce::OESecondaryStructureSuperposition, 78
OESpruce::OEStructuralSuperposition, 84
OESpruce::OESuperposeResults, 95

V

Validate
OESpruce::OEValidateDesignUnit, 97