

OpenEye
SCIENTIFIC

Szybki TK – C#

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Introduction **1**

2 Theory **3**
- 2.1 SZYBKI Theory . . . 3
  - 2.1.1 Force Field . . . 3
  - 2.1.2 Entropy evaluation . . . 3
- 2.2 Nonplanar Aromatic Rings in Protein-Bound Ligands . . . 4
- 2.3 *freeform* Theory . . . 5
  - 2.3.1 Conformer Free Energies . . . 5
  - 2.3.2 Solvation energy estimation . . . 7

3 API **13**
- 3.1 OESz Classes . . . 13
  - 3.1.1 OEBoundEntropyOptions: public OELigandChargeOptions . . . 13
  - 3.1.2 OEEntropyResults . . . 14
  - 3.1.3 OEFixedProteinLigandOptimizer . . . 15
  - 3.1.4 Energy . . . 15
  - 3.1.5 Optimize . . . 16
  - 3.1.6 SetProtein . . . 16
  - 3.1.7 OEFlexProteinLigandOptimizer . . . 16
  - 3.1.8 Energy . . . 16
  - 3.1.9 Optimize . . . 17
  - 3.1.10 OEFreeFormConf . . . 18
  - 3.1.11 OEFreeFormConfAdvanced . . . 19
  - 3.1.12 OEFreeFormConfOptions . . . 21
  - 3.1.13 SetCorrSolventOptions . . . 22
  - 3.1.14 OEFreeFormConfResults . . . 24
  - 3.1.15 OEFreeFormSolvOptions . . . 25
  - 3.1.16 OEFreeFormSolvResults . . . 27
  - 3.1.17 OELigandChargeOptions . . . 28
  - 3.1.18 OELigandEntropyOptions: public OELigandChargeOptions . . . 29
  - 3.1.19 OEProteinFlexOptions . . . 31
  - 3.1.20 OEProteinLigandOptOptions . . . 32
  - 3.1.21 OEProteinLigandOptResults . . . 34
  - 3.1.22 OERestrictionEnergyResult . . . 35
  - 3.1.23 OESingleConfResult . . . 35
  - 3.1.24 OESzybki . . . 38
  - 3.1.25 SetTorsionConstraint . . . 42
  - 3.1.26 UnsetProtein . . . 42
  - 3.1.27 OESzybkiEnsembleResults . . . 42

3.1.28 OESzybkiGeneralOptions . . . 44
3.1.29 OESzybkiOptions . . . 47
3.1.30 OESzybkiOptOptions . . . 48
3.1.31 OESzybkiProteinOptions . . . 50
3.1.32 OESzybkiResults . . . 52
3.1.33 OESzybkiSolventOptions . . . 56
3.1.34 OETorsionScanOptions . . . 58
3.1.35 GetUseSinglePointPB . . . 60
3.1.36 GetNumStarts . . . 60
3.1.37 GetOverlapDiv . . . 60
3.1.38 SetDelta . . . 60
3.1.39 SetForceField . . . 60
3.1.40 SetForceFieldType . . . 60
3.1.41 SetMaxIter . . . 61
3.1.42 SetSolvationType . . . 61
3.1.43 SetPenaltyForceConstant . . . 61
3.1.44 SetTolerance . . . 61
3.1.45 SetUseInternalCoord . . . 61
3.1.46 SetUserDefinedAngles . . . 61
3.1.47 SetUseSinglePointPB . . . 61
3.1.48 SetNumStarts . . . 62
3.1.49 SetOverlapDiv . . . 62
3.1.50 OETorsionScanResult . . . 62
3.1.51 OETorsionScanner . . . 63
3.1.52 GetLowestEnergyConfs . . . 63
3.1.53 GetLowestEneResults . . . 63
3.1.54 GetScannedAngles . . . 64
3.1.55 GetSinglePointConfs . . . 64
3.1.56 Scan . . . 64
3.2 OESz Constants . . . 64
3.2.1 OEEntropyMethod . . . 64
3.2.2 OEEnvType . . . 64
3.2.3 OEForceFieldType . . . 65
3.2.4 OEOptType . . . 67
3.2.5 OEPotentialTerms . . . 67
3.2.6 OEProteinElectrostatics . . . 72
3.2.7 OERunType . . . 73
3.2.8 OESolventModel . . . 73
3.2.9 OEAtomicRadii . . . 74
3.2.10 OEProtFlex . . . 74
3.2.11 OEFreeFormIonicState . . . 75
3.2.12 OEFreeFormReturnCode . . . 75
3.2.13 OEFreeFormWarning . . . 77
3.3 OESz Functions . . . 78
3.3.1 OEEstimateConfFreeEnergies . . . 78
3.3.2 OEEstimateSolvFreeEnergy . . . 78
3.3.3 OEGetEnergyTermName . . . 78
3.3.4 OEGetFreeFormError . . . 78
3.3.5 OEGetSzybkiError . . . 78
3.3.6 OESzybkiGetArch . . . 79
3.3.7 OESzybkiGetLicensee . . . 79
3.3.8 OESzybkiGetPlatform . . . 79
3.3.9 OESzybkiGetRelease . . . 79
3.3.10 OESzybkiGetSite . . . 79

3.3.11 OESzybkiGetVersion . . . 79
3.3.12 OETorsionScan . . . 79
3.3.13 OEBoundLigandEntropy . . . 80
3.3.14 OELigandEntropy . . . 81

4 Preliminary API **83**
4.1 Preliminary OESz Classes . . . 83
4.1.1 OELigandEnergy . . . 83
4.1.2 OELigandOptOptions . . . 85
4.1.3 GetForceField . . . 86
4.1.4 GetForceFieldType . . . 86
4.1.5 GetLigandCharge . . . 86
4.1.6 GetLigandChargeType . . . 86
4.1.7 GetNonBondCutoff . . . 86
4.1.8 GetGradTolerance . . . 86
4.1.9 GetMaxIter . . . 86
4.1.10 GetSolventModel . . . 86
4.1.11 SetForceField . . . 87
4.1.12 SetLigandCharge . . . 87
4.1.13 SetNonBondCutoff . . . 87
4.1.14 SetGradTolerance . . . 87
4.1.15 SetMaxIter . . . 87
4.1.16 SetSolventModel . . . 87
4.1.17 OELigandOptResults . . . 87
4.1.18 OETorConstraintOptOptions . . . 89
4.1.19 GetPenaltyForceConstant . . . 90
4.1.20 GetUseInternalCoord . . . 90
4.1.21 SetPenaltyForceConstant . . . 90
4.1.22 SetUseInternalCoord . . . 90
4.1.23 OETorConstraintOptimizer . . . 90
4.1.24 Setup . . . 91
4.1.25 Optimize . . . 91
4.2 Preliminary OESz Constants . . . 91
4.2.1 OEComplexFFType . . . 91
4.2.2 OELigandChargeType . . . 92
4.2.3 OEOptimizationType . . . 93
4.2.4 OESzybkiReturnCode . . . 93

5 Examples: Working with Szybki TK **97**
5.1 Ligand Energetics and Optimization . . . 97
5.1.1 Single ligand in vacuum . . . 97
5.1.2 Single ligand in vacuum using SMIRNOFF . . . 98
5.1.3 Optimization of a set of ligands . . . 100
5.1.4 Optimization of a single ligand with the Newton-Raphson method . . . 103
5.1.5 Optimization of all conformers of a ligand . . . 105
5.1.6 Optimization of a single bound ligand . . . 106
5.2 Protein-Ligand Energetics and Optimization . . . 108
5.2.1 Optimization of a set of bound ligands in a rigid receptor . . . 108
5.2.2 Optimization of a set of bound ligands in a partially flexible receptor . . . 111
5.2.3 Optimization of a bound ligand in a partially flexible receptor . . . 115
5.2.4 Estimation of PB binding for a set of ligands . . . 117
5.2.5 Optimization of a bound ligand using Newton-Raphson method . . . 120
5.3 DU Protein-Ligand Optimization with FF14SB-Parsley . . . 121
5.3.1 Optimization of ligand in a rigid active site . . . 122

5.3.2 Optimization of ligand in a partially flexible active site . . . 122
5.4 Entropy estimation . . . 122
5.4.1 Estimation of solution ligand entropy . . . 122
5.4.2 Estimation of bound ligand entropy . . . 123
5.5 Solvation free energy estimation . . . 124
5.6 Conformations free energy estimation . . . 126
5.6.1 Simple free energy estimation . . . 126
5.6.2 Simple restriction energy estimation . . . 127
5.6.3 Advanced free energy estimation . . . 129
5.6.4 Advanced restriction energy estimation . . . 131
5.6.5 Finding similar conformers . . . 134
5.6.6 Torsion scan . . . 135

6 Release History **137**
6.1 Szybki TK 2.9.2 . . . 137
6.2 Szybki TK 2.9.1 . . . 137
6.3 Szybki TK 2.9.0 . . . 137
6.3.1 New Features . . . 137
6.3.2 Major Bug Fixes . . . 138
6.3.3 Minor Bug Fixes . . . 138
6.4 Szybki TK 2.8.0 . . . 138
6.4.1 New features . . . 138
6.5 Szybki TK 2.7.1.3 . . . 138
6.6 Szybki TK 2.7.1.0 . . . 138
6.7 Szybki TK 2.7.0 . . . 138
6.7.1 New features . . . 138
6.7.2 Major bug fixes . . . 139
6.7.3 Minor bug fixes . . . 139
6.8 Szybki TK 2.6.0.1 . . . 139
6.9 Szybki TK 2.6.0.0 . . . 139
6.9.1 New features . . . 139
6.9.2 Major bug fixes . . . 139
6.9.3 Minor bug fixes . . . 140
6.10 Szybki TK 2.5.1.2 . . . 140
6.11 Szybki TK 2.5.1.1 . . . 140
6.11.1 New features . . . 140
6.11.2 Major bug fixes . . . 140
6.11.3 Minor bug fixes . . . 140
6.12 Szybki TK 2.5.0 . . . 140
6.12.1 New Features . . . 140
6.13 Szybki TK 2.4.0 . . . 141
6.13.1 New features . . . 141
6.13.2 Minor bug fixes . . . 141
6.14 Szybki TK 2.3.1 . . . 141
6.14.1 New features . . . 141
6.15 Szybki TK 2.3.0 . . . 141
6.15.1 New features . . . 141
6.15.2 Minor bug fixes . . . 142
6.16 Szybki TK 2.2.0 . . . 142
6.16.1 New features . . . 142
6.16.2 Major bug fixes . . . 142
6.16.3 Minor bug fixes . . . 142
6.17 Szybki TK 2.1.0 . . . 143
6.17.1 New features . . . 143

6.17.2 Minor bug fixes . . . 143
6.17.3 Documentation changes . . . 143
6.18 Szybki TK 2.0.4 . . . 144
6.18.1 Major bug fixes . . . 144
6.18.2 Minor bug fixes . . . 144
6.19 Szybki TK 2.0.3 . . . 144
6.19.1 New features . . . 144
6.19.2 Documentation changes . . . 144
6.20 Szybki TK 2.0.2 . . . 144
6.20.1 New features . . . 144
6.20.2 Minor bug fixes . . . 144
6.21 Szybki TK 2.0.1 . . . 144
6.21.1 Minor bug fixes . . . 144
6.22 Szybki TK 2.0.0 . . . 145
6.22.1 Minor bug fixes . . . 145
6.22.2 Documentation changes . . . 145
6.23 Szybki TK 1.9.2 . . . 145
6.23.1 New features . . . 145
6.23.2 Minor bug fixes . . . 149
6.24 Szybki TK 1.9.1 . . . 149
6.24.1 New features . . . 149
6.24.2 Minor bug fixes . . . 149
6.24.3 Documentation fixes . . . 149
6.25 Szybki TK 1.9.0 . . . 149
6.25.1 New features . . . 149
6.25.2 Minor bug fixes . . . 150
6.26 Szybki TK 1.8.7 . . . 150
6.26.1 New features . . . 150
6.26.2 Major bug fixes . . . 150
6.26.3 Minor bug fixes . . . 150
6.27 Szybki TK 1.8.6 . . . 151
6.27.1 Major bug fixes . . . 151
6.27.2 Minor bug fixes . . . 151
6.27.3 Documentation fixes . . . 151
6.28 Szybki TK 1.8.5 . . . 151
6.28.1 New features . . . 151
6.28.2 Major bug fixes . . . 152
6.28.3 Minor bug fixes . . . 152
6.28.4 Documentation fixes . . . 152
6.29 Szybki TK 1.8.4 . . . 152
6.29.1 New features . . . 152
6.29.2 Major bug fixes . . . 152
6.29.3 Minor bug fixes . . . 152
6.29.4 API Changes . . . 153
6.29.5 Documentation fixes . . . 153
6.29.6 Java- and C#-specific changes . . . 153
6.30 Szybki TK 1.8.3 . . . 153
6.30.1 New features . . . 153
6.30.2 Major bug fixes . . . 154
6.30.3 Minor bug fixes . . . 154
6.30.4 Documentation fixes . . . 155
6.31 Szybki TK 1.8.2 . . . 155
6.31.1 New features . . . 155
6.31.2 Major bug fixes . . . 155

6.31.3 Minor bug fixes . . . 155
6.31.4 Documentation fixes . . . 156
6.32 Szybki TK 1.8.1 . . . 156
6.32.1 New features . . . 156
6.32.2 Major bug fixes . . . 156
6.32.3 Minor bug fixes . . . 156
6.32.4 Documentation fixes . . . 157
6.33 Szybki TK 1.7.5 . . . 157
6.33.1 New features . . . 157
6.33.2 Minor bug fixes . . . 157
6.34 Szybki TK 1.7.4 . . . 157
6.34.1 Bug fixes . . . 157
6.35 Szybki TK 1.7.3 . . . 157
6.35.1 New features . . . 157
6.35.2 Major bug fixes . . . 158
6.35.3 Minor bug fixes . . . 158
6.36 Szybki TK 1.7.2 . . . 158
6.36.1 New features . . . 158
6.36.2 Major bug fixes . . . 158
6.36.3 Minor bug fixes . . . 158
6.37 Szybki TK 1.7.1 . . . 159
6.37.1 New features . . . 159
6.37.2 Bug fixes . . . 159
6.38 Szybki TK 1.7.0 . . . 159
6.38.1 New features . . . 159
6.38.2 Bug fixes . . . 160
6.39 Szybki TK 1.6.0 . . . 161
6.39.1 New features . . . 161
6.39.2 Bug fixes . . . 161
6.40 Szybki TK 1.5.0 . . . 161
6.40.1 New features . . . 161
6.40.2 Bug fixes . . . 162
6.41 Szybki TK 1.4.0 . . . 162
6.41.1 New features . . . 162
6.41.2 Bug fixes . . . 162
6.42 Szybki TK 1.3.4 . . . 162
6.42.1 New features . . . 162
6.42.2 Bug fixes . . . 163

7 Copyright and Trademarks **165**

8 Sample Code **167**

9 Citation **169**
9.1 Orion® . . . 169
9.2 Orion Floes . . . 169
9.3 Toolkits and Applications . . . 169
9.4 Publications for Bibliographies . . . 171
9.4.1 Orion . . . 171
9.4.2 AFITT and FLYNN Applications . . . 171
9.4.3 OEDocking Application and Toolkit . . . 171
9.4.4 OMEGA Application and Toolkit . . . 171
9.4.5 ROCS Application . . . 172
9.5 OpenEye MMDS Web Service . . . 172

10 Technology Licensing **173**
10.1 GCC 245
10.1.1 GCC RUNTIME LIBRARY EXCEPTION 246
10.1.2 GNU GENERAL PUBLIC LICENSE 247

Index **261**

CHAPTER

ONE

INTRODUCTION

The *Szybki Toolkit* library provides a facility for force field based optimization and thermodynamic property estimations of small molecule ligands or protein-ligand complexes with small molecule ligands.

The following basic functionalities are available:

- Entropy Evaluation (see *Entropy evaluation* chapter)
- Conformer free Energies (see *Conformer Free Energies* chapter)
- Solvation free Energies (see *Solvation energy estimation* chapter)

Note

Units of energy in various **Szybki TK** outputs are in kcal/mol.

The aim of this manual is to familiarize the user with the *Szybki Toolkit* functionalities, however, it does not provide explanations of basic *OEChem* classes and functions. Therefore reading the *OEChem* manual beforehand is highly recommended.

See also

- C# Quick Start Manual for installation instructions.
- Importing C# Toolkits <section_csharp_tk_import section

CHAPTER

TWO

THEORY

2.1 SZYBKI Theory

2.1.1 Force Field

Theory documentation for forcefields is available from OEFF theory.

2.1.2 Entropy evaluation

Ligand entropy is evaluated as a sum of configurational entropy (S_c) and solvation entropy (ΔS_s):

$$S = S_c + \Delta S_s$$

Configurational entropy

Configurational entropy is calculated as:

$$S_c = kN\left[1 + \ln\left(\frac{q}{N}\right) + \frac{T}{q}\frac{\partial q}{\partial T}\right]$$

where q is the conformation dependent partition function:

$$q = q_t \sum_{i=1}^{n_c} e^{-\frac{\epsilon_i}{kT}} q_{iv} q_{ir}$$

Here q_t, q_{ir} and q_{iv} are the translational, rotational and vibrational partition functions respectively, n_c is the number of unique conformations in the ensemble. All 3 partition functions are calculated from the classical statistical mechanics expressions which could be found in [McQuarrie-1976]. Vibrational frequencies for each conformation, needed for evaluation of q_{ir} are derived from diagonalization of a Hessian matrix obtained from Quasi-Newton optimization when convergence is achieved. Eigenvalues λ_i of the mass-weighted Hessian:

$$\mathbf{H}^m = \mathbf{M}^{-1/2}\mathbf{H}\mathbf{M}^{-1/2}$$

are converted into wavenumbers $\tilde{\nu}_i$ according to:

$$\tilde{\nu}_i = \frac{1}{2\pi c}\sqrt{\lambda_i}$$

Solvation entropy

Solvation entropy is split into electrostatic and hydrophobic parts:

$$\Delta S_s = \Delta S_{s,elec} + \Delta S_{s,hyd}$$

The electrostatic part of solvation entropy is divided in to the bulk component and tight electrostatic polar solute - water interactions (hydrogen bonds). The bulk contribution is estimated from the temperature dependence of the solvent dielectric constant as:

$$\Delta S_{s,elec_bulk} = -\left(\frac{\partial \Delta G_s}{\partial \epsilon_{solv}}\right)\left(\frac{\partial \epsilon_{solv}}{\partial T}\right)$$

The second term of the electrostatic solvation entropy is estimated as a constant of 28 J/(mol K).

The hydrophobic term, $\Delta S_{s,hyd}$, is evaluated as:

$$\Delta S_{s,hyd} = -\left(\frac{\partial \Delta G_{s,hyd}}{\partial T}\right)$$

where $\Delta G_{s,hyd}$ consists of 3 components:

$$\Delta G_{s,hyd} = \Delta G_{cav} + \Delta G_{VdW} + \Delta G_{Ind}$$

describing the free energy of cavitation, solute-solvent van der Waals and inductive terms respectively. The cavity formation term is calculated from Scaled Particle Theory [Pierotti-1976]. Analytical expressions for ΔG_{VdW} and ΔG_{Ind} terms are also taken from the 1976 Pierotti review.

Protein-bound ligand entropy

Configurational entropy of a protein bound ligand is calculated totally as vibrational entropy for 3N modes, assuming that 3 rotational and 3 translational degrees of freedom of a solution ligand are transformed into low-vibrational degrees of freedom for the bound ligand.

Solvation entropy for a ligand in the active site is assumed to be a sum of its fractional value in solution determined by the percentage of the ligand surface exposed to the solvent, $f\Delta S_s$, and a partial desolvation entropy of the protein active site, ΔS_{des}

$$S_{protein} = S_v + f\Delta S_s + \Delta S_{des}$$

where f is the fraction of ligand surface exposed to the solvent. It is important to notice that $S_{protein}$ is not an experimentally measurable value, and that only the difference between $S_{protein}$ and $S = S_c + \Delta S_s$ might be compared with experimental binding entropy.

2.2 Nonplanar Aromatic Rings in Protein-Bound Ligands

Optimization of a ligand in a protein active site, especially if the protein conformation is held fixed during optimization, can often lead to distortions of the aromatic ring structures. Aromatic rings in optimized ligands are more prone to distortions when the initial pose structures are the output of docking, since tools such as **OEDocking** or **POSIT** try to accommodate a ligand into a rigid receptor pocket. Optimization of a ligand in the protein pocket helps relieve both intramolecular close contacts within the ligand and intermolecular clashes between the ligand and the protein. However, such relaxation of the van der Waals clashes can sometimes come at the expense of nonplanarity in the aromatic ring structures.

Ligand force fields used in optimization, such as `Sage`, `Parsley`, and `MMFF`, put a low energy constant in defining the out-of-plane potential of aromatic rings. The low energy penalty for aromatic rings to deviate from planarity in the force fields is also supported by quantum mechanical calculations which estimate only a 0.5 kcal/mol penalty for ~5 degrees of deviation for a benzene ring. The out-of-plane potential being the lowest vibrational motion during optimization means it can easily be bent by a few degrees during optimization.

2.3 *freeform* Theory

freeform has two distinct functions:

"freeform -calc conf…": Evaluation of the conformer free energy in solution, meaning the free energy required to select a particular conformer out of the whole conformational ensemble in solution.

"freeform -calc solv…": Fast estimation of the small molecule solvation free energy and the XLogP calculated partition coefficient (with graphical representation of fragment-based contributions to these physical quantities).

2.3.1 Conformer Free Energies

Calculating the partition function

The method is based upon combining ligand conformational search, energy minimization, and entropy estimation in the workflow shown in *Figure: Stages and workflow in the conformer free energy calculation* below.

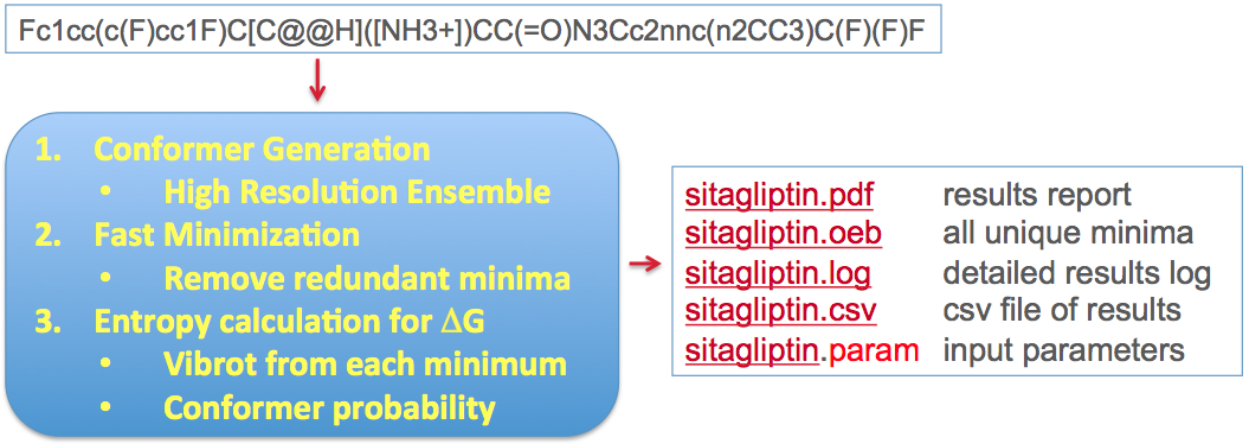


Fig. 1: **Stages and workflow in the conformer free energy calculation**

The conformational search is done at a very high resolution using the conformation generator OMEGA [Hawkins-2010]. The energy minimization uses Halgren's MMFF94s force field [Halgren-VI-1999]and consists of several stages, all aimed at arriving at the most thorough sampling of the accessible conformational space of the ligand. Duplicate conformers are removed from the final minimized set, leaving an ensemble of n_c unique conformers which we use to create an approximation to the partition function of the unbound ligand. This is illustrated conceptually in *Figure: Concept of ligand entropy as a sum-over-states of conformers*.

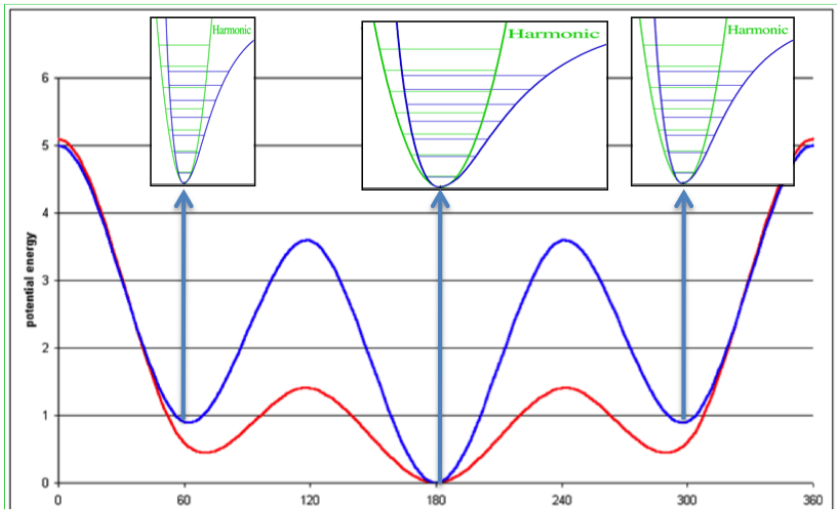


Fig. 2: **Concept of ligand entropy as a sum-over-states of conformer entropies**

To the molecule's gas-phase potential based on the MMFF94 force field (shown by the red line), Sheffield solvation energies are added to give a solvation-corrected potential (shown by the blue line) for which the minima are identified. The analytic second derivative of each minimum is then used to calculate the entropy contribution of each energy well according to [Wlodek-2010]. An energy minimum with a wider than average energy well will have a higher entropy, which will decrease the conformer free energy to select that conformer, while the converse will be true for a narrower than average energy well.

With this, the contribution Q_i of each energy minimum to the overall partition function Q can be calculated:

$$Q_i = q_t q_{ir} q_{iv} e^{-\epsilon_i / RT}$$

$$Q = \sum_{i=1}^{n_c} Q_i$$

Here ϵ_i is the relative internal energy (including solvation energy) of the conformer relative to the global minimum conformation. q_t, $q_i r$ and $q_i v$ are the translational, rotational and vibrational partition functions respectively, describing the entropic contributions from that conformation. n_c is the number of unique conformations in the ensemble. More details on these terms are given in the theory section of the szybki documentation.

Computing the conformer free energies

Based upon this approximation to the partition function, we calculate conformer free energies: the free energy required to select a particular conformation from the unbound ensemble of all conformers equilibrating in aqueous solution. Strictly speaking, this is a Helmholtz free energy, not a Gibb's free energy, because we are neglecting the differential PV (Pressure*Volume) contribution between conformers, but this should be negligible. Based on the above conformer and ensemble partition functions, the conformer free energy for any conformer is calculated straightforwardly as:

$$\Delta A_i = -RT(\ln Q_i - \ln Q)$$

In addition to the graphic output file which shows a plot of the calculated free energies of conformations vs. their relative intrinsic energies, numerical results are stored in the log and csv files. For each conformation the following data are reported: rotational and vibrational entropies, total force field energy, solvation energy, the sum of solvation energy and force field energy, relative energy, free energy, and heavy atoms RMSD displacement from the input 3D structure (the latter only when the `-useInput3D` option is used).

Ligand strain in the input 3D structure

When conformer free energies are calculated with the `-useInput3D` flag, *freeform* pays attention to the 3-D structure in the input file and keeps track of it, writing out the conformer free energy quantities associated with it. The idea is that the user can input a bioactive confirmation from an x-ray crystal structure, or a hypothetical model-built structure of interest, and get a direct readout of the energy required to select that particular conformation out of the aqueous ensemble. A key issue, however, is that the input 3D coordinates may be of artificially high energy, usually due to small deviations of bond lengths and bond angles from optimum; this is often true particularly of ligands from Xray structures. For this reason a "cleanup" pre-optimization is done using harmonic constraints to the input coordinates, thus retaining the shape of the ligand but resolving the artificially high energy. This "cleanup" energy is discarded. Since conformer free energies are calculated from energy minima in order to characterize the entropy, an unconstrained minimization is now required to find the closest energy minimum to the cleaned-up structure, which we refer to as the input 3D minimum. This minimization allows the torsions in the input structure to relax away from where they may have been distorted from the nearby local minimum, for example in order to fit into the active site. *freeform* keeps track of this "distortion" strain energy as Estrain. From this point, *freeform* simply includes the relaxed input 3D minimum as just another energy minimum in the aqueous ensemble and calculates its conformer free energy, which is specifically reported in the pdf and log output (see examples). Keep in mind that the free energy required to adopt the input 3D minimum may be significant and probably should be considered a component of the strain energy also. Thus, within *freeform*, the overall strain energy for the input 3D structure would be considered to consists of a) the conformer free energy required to adopt the input 3D minimum (i.e. the closest local minimum to the input structure), and b) the additional energy `Estrain` required to distort the torsions of that minimum so the ligand can adopt the shape of the actual input 3D structure.

2.3.2 Solvation energy estimation

How it is calculated

The term "Solvation energy" is defined here as the standard free energy of transferring a compound from the gaseous phase into dilute aqueous solution, ΔG^0_{solv}.

Solvation energy is a crucial component of the solubility energetics, and is directly related to hydrophilicity/hydrophobicity of a compound, therefore one can expect it is correlated with another measure of hydrophobicity logP, as illustrated on the diagram below (*Figure: Solvation, solubility and hydrophobicity*). Such a correlation, particularly when functional groups contribution is considered might not exist. *freeform* shows side-by-side the graphical group contributions to solvation free energy and XLogP.

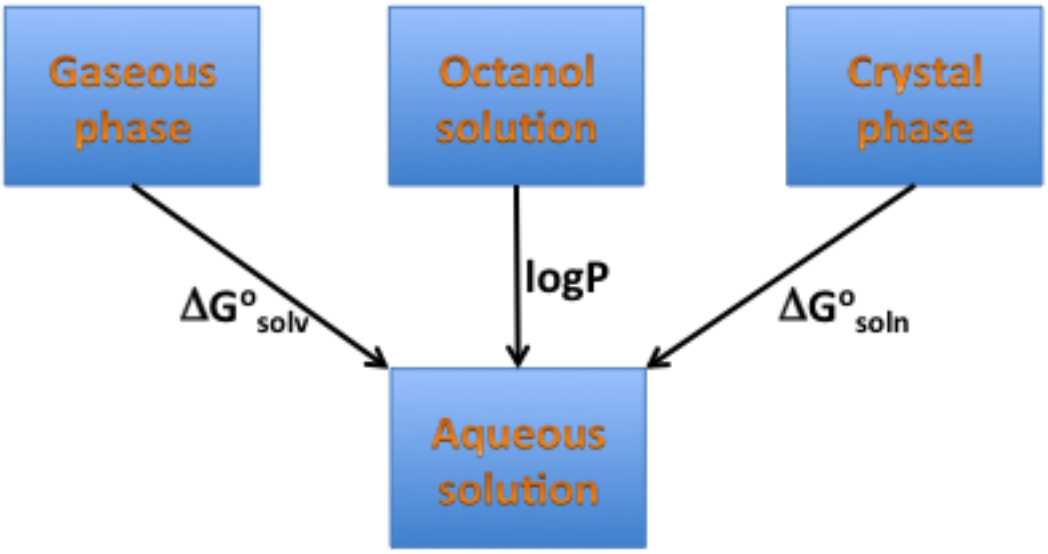


Fig. 3: **Solvation, solubility and hydrophobicity**

The method is based on the continuum solvent model in which a PB solver ZAP [Grant-2001] is used. The calculation is performed on the lowest energy gas-phase conformation found with the MMFF94 force field from the optimized set of conformations generated by the conformation generator OMEGA [Hawkins-2010]. A set of atomic radii ZAP9 and AM1BCC partial charges as described in [Nicholls-2010]is applied for all PB calculations. Atomic and group contributions to the free energy of solvation for the input compound are calculated from atomic electrostatic potentials and atomic area terms representing hydrophobic part of solvation. Assuming that a functional group has n atoms, its calculated solvation free energy ΔG_s is:

$$\Delta G_s = \sum_i^n [(V_{i,s} - V_{i,v})q_i + S_i]$$

where $V_{i,s}$ and $V_{i,v}$ are calculated electrostatic potentials on atom i in solution and in vacuum respectively, q_i is atom partial charge and S_i is atomic contribution to hydrophobic part of solvation evaluated from a simple surface area model assuming 6.3 cal/(mol Å) as a value for microscopic surface tension coefficient. Graphical representation of group (or atomic) contributions to the free energy of solvation is performed with the OE toolkits OEDepict and Grapheme.

Counterintuitive group contributions to solvation energies

Solvation free energy for molecular ions might not be trivial to interpret because of very strong intrinsic electrostatic interactions. One can most easily begin to understand these effects by considering zwitterions. A common zwitterion will contain both a group with a positive formal charge and a group with a negative formal charge. It is well known that zwitterions, though still having high solvation energies, have significantly smaller solvation energies than compounds with two formal charges of the same sign. In fact, zwitterion solvation energies are typically smaller than the solvation energy of the analog molecule with a single charge. For instance, as the following figures show *Figure: Solvation energy of phenylalanine*, *Figure: Solvation energy of amide analog of phenylalanine* and *Figure: Solvation energy of methoxy analog of phenylalanine* the solvation energy of zwitterionic phenylalanine is -32.8 kcal/mol, while the solvation energy of the formamide analog and the methyl ester analog are -73.4 and -63.4 kcal/mol respectively.

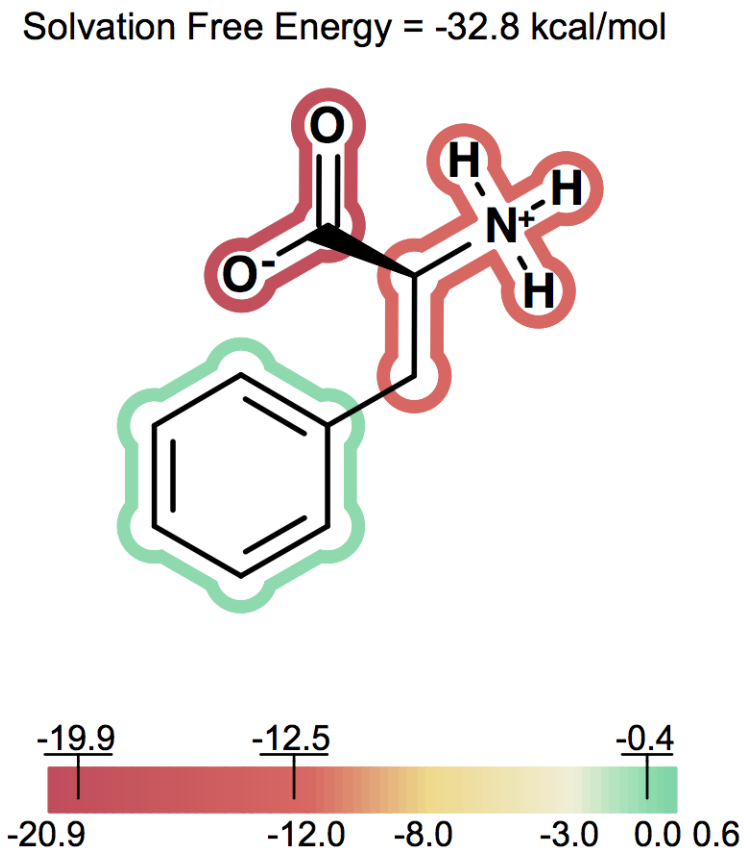


Fig. 4: **Solvation energy of phenylalanine**

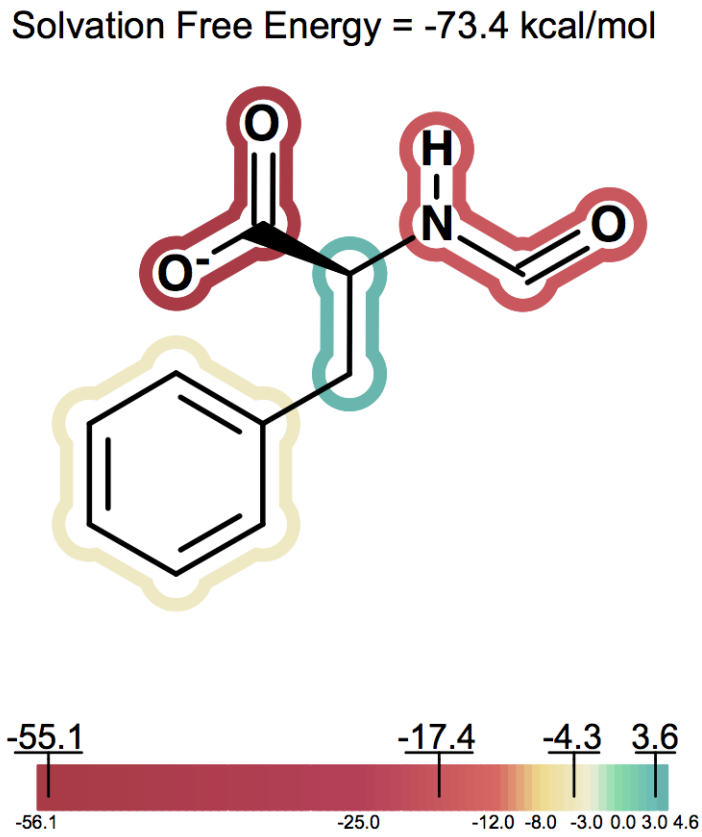


Fig. 5: **Solvation energy of amide analog of phenylalanine**

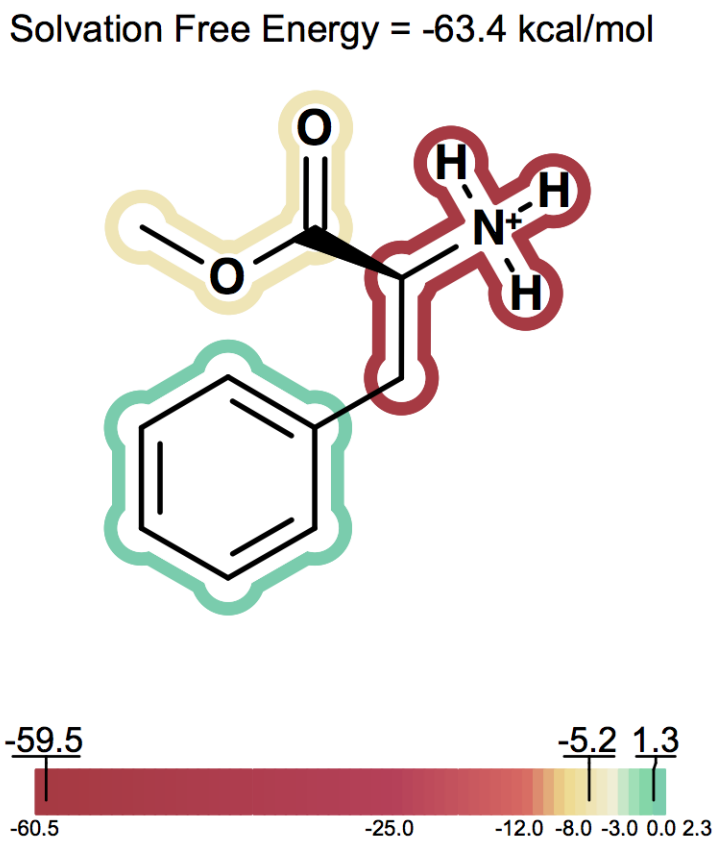


Fig. 6: **Solvation energy of methoxy analog of phenylalanine**

One can consider then, that adding a charge that is the opposite charge of the current molecular ion contributes a positive 30 to 40 kcal/mol to the solvation energy. A similar situation, perhaps even more dramatic, occurs in the case of intramolecular salt bridges, where positively and negatively charged groups are positioned in a very short distance (that kind of interaction not only exists in macromolecules: NMR and hydrogen exchange rate experiments have confirmed the presence of salt bridges between arginine or lysine and aspartate or glutamate side chains also in short peptides [Otter-1989], [Mayne-1998]). Making positive contributions to the solvation energy is generally considered a hydrophobic effect, yet here the contribution is made by a charged species – something that is somewhat counterintuitive at first. Nevertheless, in this context, one can understand how, under the right conditions, a normally hydrophilic functional group can make a significant positive contribution to the solvation energy. The more confusing circumstance is when, rather than a simple zwitterion, we see a similar effect in a molecule with either multiple charges of one type (either positive or negative) and a single charge of the opposite sign. Consider a molecule with 2 negative ions, one positive ion and a total charge of -1, like the one shown on *Figure: Example of counterintuitive group contributions to solvation* below.

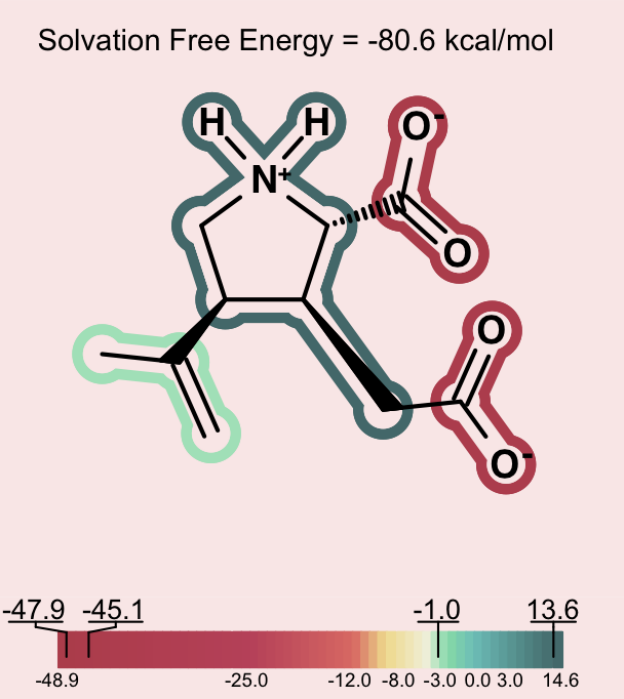


Fig. 7: **Counterintuitive group contributions to solvation**

In this case, the positive ion will be changing the formal charge from -2 to -1. This will make the total solvation energy significantly less negative (a positive contribution to the solvation energy). The positive ionic functional group in this case will have a large positive solvation energy. While a charged group is indeed not hydrophobic, in this instance, one can understand that it makes the solvation energy less negative in a manner similar to the way a hydrophobic group might, but to a larger degree. In more subtle cases, one can observe a similar though less dramatic hydrophobicity in a molecule with a single charge, either positive or negative and another fragment that makes a large neutralizing contribution to the partial charge of the molecule.

In order to alert the user, species which contain strong intramolecular electrostatic interactions of the nature described above are depicted on a light pink background. This should act as a reminder to users to consider the net charge of the molecule in interpreting the relative contribution of each fragment to the solvation energy.

Note

Units of all energies reported from **SzybkiTK** are in `kcal/mol`.

CHAPTER

THREE

API

3.1 OESz Classes

3.1.1 OEBoundEntropyOptions: public OELigandChargeOptions

```
class OEBoundEntropyOptions : public OESystem::OEOptions
```

Configuration options for *OEBoundLigandEntropy* bound-ligand entropy calculations.

See also

- *OEEntropyResults* class
- *OEBoundLigandEntropy* function

The following methods are publicly inherited from *OELigandChargeOptions*:

- *GetGenerateElfConfs* and *SetGenerateElfConfs*
- *GetLigandCharge*, *GetLigandChargeType*, and *SetLigandCharge*

The *OEBoundEntropyOptions* class defines the following public methods:

- *GetForceField*, *GetForceFieldType* and *SetForceField*

Constructors

```
OEBoundEntropyOptions()
OEBoundEntropyOptions(const OEBoundEntropyOptions &)
```

Default and copy constructors.

GetForceField

```
const OEFF::OEComplexFF* GetForceField() const
```

See *SetForceField* method.

GetForceFieldType

```
unsigned GetForceFieldType() const
```

See *SetForceField* method.

SetForceField

```
bool SetForceField(const unsigned)
bool SetForceField(const std::string&)
bool SetForceField(const OEFF::OEComplexFF&)
```

Sets the **force field** for optimization. The first overload takes an *unsigned* from the *OEComplexFFType* namespace, and the second overload takes corresponding string values. The final overload allows the use of a user-defined OEComplexFF. Method returns `false` when the invalid selection is chosen, `true` otherwise. **Default:** *OEComplexFFType.FF14SB_SAGE*

3.1.2 OEEntropyResults

```
class OEEntropyResults
```

Container for entropy estimation results returned by *OELigandEntropy* and *OEBoundLigandEntropy*.

See also

- *OEBoundEntropyOptions* class
- *OELigandEntropyOptions* class
- *OEBoundLigandEntropy* function
- *OELigandEntropy* function

Constructors

```
OEEntropyResults()
OEEntropyResults(const OEEntropyResults&)
```

Default and copy constructors.

GetTotalEntropy in J/(mol K).

```
double GetTotalEntropy() const
```

Returns the value of total ligand entropy.

GetConfigurationalEntropy

```
double GetConfigurationalEntropy() const
```

Returns the value of configurational entropy in J/(mol K).

GetTranslationalEntropy

```
double GetTranslationalEntropy() const
```

Returns the value of translational entropy in J/(mol K).

GetVibrationalRotationalEntropy

```
double GetVibrationalRotationalEntropy() const
```

Returns a sum of vibrational and rotational entropy in J/(mol K).

GetSolvationEntropy

```
double GetSolvationEntropy() const
```

Returns the value of solvation entropy in J/(mol K).

SetResults

```
void SetResults(const double data[], unsigned int nterms = 4)
```

Allows to set the *OEEntropyResults <OESz_OEEntropyResults>* object with numeric values of entropy components by passing a pointer to the array of doubles in the following order: data[0]: total entropy, data[1] = translational entropy, data[2] = rotvib entropy and data[3]: solvation entropy

3.1.3 OEFixedProteinLigandOptimizer

```
class OEFixedProteinLigandOptimizer
```

This class provides an interface to optimize a ligand in a protein active site, where the protein is kept fixed during the optimization. The protein active site can be defined either by using a OEDesignUnit or by simple passing in a protein molecule.

The *OEFixedProteinLigandOptimizer* class defines the following public methods:

- *Energy*
- *Optimize*
- *SetProtein*

Constructor

```
OEFixedProteinLigandOptimizer(const OEProteinLigandOptOptions& opts =␣
↪OEProteinLigandOptOptions())
OEFixedProteinLigandOptimizer(const OEFixedProteinLigandOptimizer &)
```

Default and copy constructors.

Assignment operator.

3.1.4 Energy

```
OESystem::OEIterBase<OEFF::OEComplexEnergies>* Energy(const␣
↪OEChem::OEMCMolBase& ligand) const
unsigned Energy(OEFF::OEComplexEnergies& res,
                const OEChem::OEMolBase& ligand) const
```

Calculate the single point energy of the ligand interacting with the protein. The intra-molecular protein energy components are ignored by this method. The second overload returns a value from the *OESzybkiReturnCode* namespace reporting the success or failure of the calculation.

3.1.5 Optimize

```
OESystem::OEIterBase<OESz::OEProteinLigandOptResults>*␣
↪Optimize(OEChem::OEMCMolBase& ligand) const
unsigned Optimize(OESz::OEProteinLigandOptResults& res,
                  OEChem::OEMolBase& ligand) const
```

Optimize the ligand in the protein active site. The intra-molecular protein energy components are ignored by this method. The second overload returns a value from the *OESzybkiReturnCode* namespace reporting the success or failure of the optimization. The return code for each conformer in the first overload can be obtained from the corresponding results object.

3.1.6 SetProtein

```
unsigned SetProtein(const OEBio::OEDesignUnit& du,
                    const unsigned proteinMask,
                    const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪proteinPred = OEChem::OEIsTrueAtom())
unsigned SetProtein(const OEChem::OEMolBase& mol)
```

Set the protein molecule for the calculation. In the first two overload, the `proteinMask` define the portion of the OEDesignUnit that should be used as the target protein active site. An optional `proteinPred` can be also provided along with the mask, to define additional residue components to be part of the target. Method returns a value from the *OESzybkiReturnCode* namespace reporting the success or failure.

3.1.7 OEFlexProteinLigandOptimizer

```
class OEFlexProteinLigandOptimizer
```

This class provides an interface to optimize a ligand in a protein active site, where the protein can be treated a partially flexible. The protein active site can be defined either by using a OEDesignUnit or by simple passing in a protein molecule.

The *OEFlexProteinLigandOptimizer* class defines the following public methods:

- *Energy*
- *Optimize*

Constructor

```
OEFlexProteinLigandOptimizer(const OEProteinLigandOptOptions& opts =␣
↪OEProteinLigandOptOptions())
OEFlexProteinLigandOptimizer(const OEFlexProteinLigandOptimizer &)
```

Default and copy constructors.

Assignment operator.

3.1.8 Energy

```
unsigned Energy(OEFF::OEComplexEnergies& res,
                const OEBio::OEDesignUnit& du,
                const unsigned proteinMask,
                const unsigned ligandMask,
```

(continues on next page)

(continued from previous page)

```
                const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪protFlexPred,
                const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪proteinPred = OEChem::OEIsTrueAtom()) const
unsigned Energy(OEFF::OEComplexEnergies& res,
                const OEBio::OEDesignUnit& du,
                const unsigned proteinMask,
                const unsigned ligandMask,
                const OEProteinFlexOptions& flexOpts,
                const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪proteinPred = OEChem::OEIsTrueAtom()) const
unsigned Energy(OEFF::OEComplexEnergies& res,
                const OEChem::OEMolBase& protein,
                const OEChem::OEMolBase& ligand,
                const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪protFlexPred) const
unsigned Energy(OEFF::OEComplexEnergies& res,
                const OEChem::OEMolBase& protein,
                const OEChem::OEMolBase& ligand,
                const OEProteinFlexOptions& flexOpts) const
```

Calculate the single point energy of the protein ligand complex. Only the flexible portion of the intramolecular protein energy components are considered by this method. In the first four overloads, the `proteinMask` define the portion of the OEDesignUnit that should be used as the target protein active site. An optional `proteinPred` can be also provided along with the mask, to define additional residue components to be part of the target. The methods return a value from the *OESzybkiReturnCode* namespace reporting the success or failure of the calculation.

3.1.9 Optimize

```
unsigned Optimize(OESz::OEProteinLigandOptResults& res,
                  OEBio::OEDesignUnit& du,
                  const unsigned proteinMask,
                  const unsigned ligandMask,
                  const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪protFlexPred,
                  const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪proteinPred = OEChem::OEIsTrueAtom()) const
unsigned Optimize(OESz::OEProteinLigandOptResults& res,
                  OEBio::OEDesignUnit& du,
                  const unsigned proteinMask,
                  const unsigned ligandMask,
                  const OEProteinFlexOptions& flexOpts,
                  const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪proteinPred = OEChem::OEIsTrueAtom()) const
unsigned Optimize(OESz::OEProteinLigandOptResults& res,
                  OEChem::OEMolBase& protein,
                  OEChem::OEMolBase& ligand,
                  const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪protFlexPred) const
unsigned Optimize(OESz::OEProteinLigandOptResults& res,
                  OEChem::OEMolBase& protein,
```

(continues on next page)

(continued from previous page)

```
                    OEChem::OEMolBase& ligand,
                    const OEProteinFlexOptions& flexOpts) const
```

Optimize the protein ligand complex. Only the flexible portion of the intra-molecular protein energy components are considered by this method. In the first four overloads, the `proteinMask` define the portion of the OEDesignUnit that should be used as the target protein active site. An optional `proteinPred` can be also provided along with the mask, to define additional residue components to be part of the target. The methods return a value from the *OESzybkiReturnCode* namespace reporting the success or failure of the calculation.

3.1.10 OEFreeFormConf

```
class OEFreeFormConf
```

The *OEFreeFormConf* class defines high level interface for conformer Free Energies calculation. An advanced interface for conformer Free Energies is provided through the derived class *OEFreeFormConfAdvanced*. All the methods of *OEFreeFormConf* are also available through the *OEFreeFormConfAdvanced* class interface.

Constructors

```
OEFreeFormConf(const OEFreeFormConfOptions& opts = OEFreeFormConfOptions())
OEFreeFormConf(const OEFreeFormConf&)
```

Default and copy constructors.

EstimateFreeEnergies

```
unsigned int EstimateFreeEnergies(OEChem::OEMCMolBase& mol) const
unsigned int EstimateFreeEnergies(OEChem::OEMCMolBase& mol,
                                  OEChem::OEMCMolBase& restrictedMol) const
```

Estimates the Gibbs free energy to convert an ensemble of solution conformations into a state where only a single specific conformation is present. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace.

mol

Represents the input molecule to estimate the conformer free energies, and on return it is modified to represent the corresponding ensemble of solution conformers. The return ensemble also contains the estimated free energy components.

restrictedMol

The restricted conformers of the molecule for which to estimate the restriction energies. The estimated restriction energies are tagged on conformers on return.

FindSimilarConfs

```
unsigned int FindSimilarConfs(OEChem::OEMCMolBase& mol,
                              const OEChem::OEMCMolBase& ensemble,
                              const OEChem::OEConfBase& conf,
                              const OESystem::OEBinaryPredicate<OEChem::OEConfBase,␣
↪OEChem::OEConfBase>& condition) const
```

Find similar conformers to the specified one from the given ensemble of conformers, based on *condition* provided. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace. A failure is returned when no matches are found.

mol
Contains the conformers that are found on return. The found conformers are added to any existing conformers.

ensemble
Ensemble of conformers to search from.

conf
Reference conformer used for searching.

condition
Functor defining the similarity condition.

3.1.11 OEFreeFormConfAdvanced

```
class OEFreeFormConfAdvanced
```

Advanced interface for conformer free-energy calculations, extending *OEFreeFormConf* with additional controls.

The following methods are publicly inherited from *OEFreeFormConf*:

- *EstimateFreeEnergies*
- *FindSimilarConfs*

Constructors

```
OEFreeFormConfAdvanced(const OEFreeFormConfOptions& opts = OEFreeFormConfOptions())
OEFreeFormConfAdvanced(const OEFreeFormConfAdvanced&)
```

Default and copy constructors.

EstimateEnergies

```
unsigned int EstimateEnergies(OEChem::OEMCMolBase& mol) const
```

Estimates the Gibbs free energy to convert an ensemble of solution conformations into a state where only a single specific conformation is present. This method assumes that a duplicate free solution ensemble is provided as input. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace. The molecule conformers contain the estimated free energy components on return.

EstimateRestrictionEnergy

```
unsigned int EstimateRestrictionEnergy(OEChem::OEMCMolBase& freeMol,
                                       const OEChem::OEMCMolBase& restrictedMol) const
```

Estimates the restriction energies on the restricted conformers. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace.

freeMol
Molecule with unrestricted conformers identified from the set of solution ensemble, corresponding to the restricted conformers. The estimated restriction energies are tagged on conformers on return.

restrictedMol
Molecule with the restricted conformers for which to estimate the restriction energies.

IdentifyConformer

```
unsigned int IdentifyConformer(OEChem::OEMCMolBase& mol,
                               const OEChem::OEMCMolBase& ensemble) const
```

Identifies the given conformers from the solution ensemble. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace.

mol
: Molecule with conformers to be identified from the set of solution ensemble. Conformer free energies from identified conformers are tagged on these conformers on return.

ensemble
: Molecule with the solution ensemble with corresponding conformer free energies.

Optimize

```
unsigned int Optimize(OEChem::OEMCMolBase& mol) const
```

Optimize the molecule conformers using the optimization options. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace. The molecule conformers contain optimized geometry and are tagged with estimated energies on return.

OptimizeRestraint

```
unsigned int OptimizeRestraint(OEChem::OEMCMolBase& mol) const
```

Perform restraint optimization of the molecule conformers using the optimization options. A harmonic restraint is applied to retain the shape of the given conformers. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace. The molecule conformers contain optimized geometry and are tagged with estimated energies on return.

PreOptimizeEnsemble

```
unsigned int PreOptimizeEnsemble(OEChem::OEMCMolBase& mol) const
```

Optimize the molecule conformers using the pre-optimization options. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace. The molecule conformers contain optimized geometry and are tagged with estimated energies on return.

PrepareEnsemble

```
unsigned int PrepareEnsemble(OEChem::OEMCMolBase& mol) const
unsigned int PrepareEnsemble(OEChem::OEMCMolBase& mol,
                             OEChem::OEMCMolBase& restrictionMol,
                             OEChem::OEMCMolBase& freeMol) const
```

Generates a comprehensive set of conformer ensemble when desired, using OEOmega. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace. The input molecule is filled with the generated ensemble on return.

Prepares the conformer ensemble for subsequent FreeForm calculations, this includes generation of a comprehensive set of conformers using OEOmega as desired, validation of the structures and charge assignment. The second overload is provided for preparing an ensemble that would include un-restricted (free) conformers corresponding to a set of given restriction ensemble of conformers. Charges are also assigned on the given restriction ensemble of conformers. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace.

mol
Molecule with the solution ensemble prepared for FreeFormConf calculations.

restrictionMol
Molecule with the restricted conformers.

freeMol
Molecule with the un-restricted conformers, corresponding to the restricted conformers.

RemoveDuplicates

```
unsigned int RemoveDuplicates(OEChem::OEMCMolBase& mol) const
```

Removes duplicates from the given ensemble of molecule conformers. Function returns a success or failure code as *unsigned int* from the *OEFreeFormReturnCode* namespace.

3.1.12 OEFreeFormConfOptions

```
class OEFreeFormConfOptions : public OESystem::OEOptions
```

This class represents options for setting up FreeFormConf calculations.

Constructors

```
OEFreeFormConfOptions()
OEFreeFormConfOptions(const OEFreeFormConfOptions&)
```

Default and copy constructors.

GetCorrSolventOptions

```
OESzybkiSolventOptions& GetCorrSolventOptions()
const OESzybkiSolventOptions& GetCorrSolventOptions() const
```

Returns a reference to the *OESzybkiSolventOptions* instance as currently set for solvent correction. When desired, these options are used in applying energetic corrections to the free energy calculations.

GetIonicState

```
unsigned GetIonicState() const
```

Returns the ionic state to be used for OEFreeFormConf calculations, as *unsigned int* from the *OEFreeFormIonicState* namespace.

GetOmegaOptions

```
OEOmegaOptions& GetOmegaOptions()
const OEOmegaOptions& GetOmegaOptions() const
```

Returns a reference to the OEOmegaOptions instance as currently set for ensemble generation. When desired, these options are used in generating the initial ensemble of conformers.

GetOptimizeOptions

```
OESzybkiOptOptions& GetOptimizeOptions()
const OESzybkiOptOptions& GetOptimizeOptions() const
```

Returns a reference to the *OESzybkiOptOptions* instance as currently set for tight optimization. These options are used in tight optimization of conformers.

GetPreOptimizeOptions

```
OESzybkiOptOptions& GetPreOptimizeOptions()
const OESzybkiOptOptions& GetPreOptimizeOptions() const
```

Returns a reference to the *OESzybkiOptOptions* instance as currently set for pre-optimization. These options are used in loose pre-optimization of conformers to pre-screen and reduce the size of conformers ensemble.

GetSolventOptions

```
OESzybkiSolventOptions& GetSolventOptions()
const OESzybkiSolventOptions& GetSolventOptions() const
```

Returns a reference to the *OESzybkiSolventOptions* instance as currently set for solvent. These options are used in defining the solvent for conformers minimization and subsequent entropy calculation.

GetSzybkiGeneralOptions

```
OESzybkiGeneralOptions& GetSzybkiGeneralOptions()
const OESzybkiGeneralOptions& GetSzybkiGeneralOptions() const
```

Returns a reference to the *OESzybkiGeneralOptions* instance as currently set. These options are used in defining general environment for all szybki calculations performed from FreeFormConf.

GetUseInputEnsemble

```
bool GetUseInputEnsemble() const
```

Returns the current state of the use input ensemble flag. If true, the input set of ensemble is used as the initial conformers ensemble and no new conformers are generated during FreeFormConf calculations.

GetUseSolventCorr

```
bool GetUseSolventCorr() const
```

Returns the current state of the use solvent correction flag. If true, energetic correction due to solvent interactions is applied after entropy calculation.

3.1.13 SetCorrSolventOptions

```
void SetCorrSolventOptions(const OESzybkiSolventOptions&)
```

Sets the correction solvent options by passing in an *OESzybkiSolventOptions* instance. When desired, these options are used in applying energetic corrections to the free energy calculations. Default solvent model: *OESolventModel.PB*, default atomic radii type: *OEAtomicRadii.ZAP9*, default charge model: OEChargeEngineNoOp.

SetIonicState

```
bool SetIonicState(const unsigned)
```

Sets the ionic state to be used for OEFreeFormConf calculations, as *unsigned int* from the *OEFreeFormIonicState* namespace. The method returns `false` when the invalid selection is chosen, `true` otherwise. Default: *OEFreeFormIonicState.PH74*.

SetOmegaOptions

```
void SetOmegaOptions(const OEOmegaOptions&)
```

Sets the omega options by passing in an OEOmegaOptions instance. When desired, these options are used in generating the initial ensemble of conformers. Default search force field: `OEMMFFSheffieldFFType.MMFF94S_SHEFF`, default EnergyWindow: 10.0, default IncludeInput: `false`, default MaxRotors: 20.

SetOptimizeOptions

```
void SetOptimizeOptions(const OESzybkiOptOptions&)
```

Sets the optimization options by passing in an *OESzybkiOptOptions* instance. These options are used in tight optimization of conformers. Default optimizer type: *OEOptType.NEWTON*, default MaxIter: 1000.

SetPreOptimizeOptions

```
void SetPreOptimizeOptions(const OESzybkiOptOptions&)
```

Sets the pre-optimization options by passing in an *OESzybkiOptOptions* instance. These options are used in loose pre-optimization of conformers to pre-screen and reduce the size of conformers ensemble. Default optimizer type: *OEOptType.BFGS*, default MaxIter: 1000, default GradTolerance: 0.005.

SetSolventOptions

```
void SetSolventOptions(const OESzybkiSolventOptions&)
```

Sets the solvent options by passing in an *OESzybkiSolventOptions* instance. These options are used in defining the solvent for conformers minimization and subsequent entropy calculation. Default solvent model: *OESolventModel.Sheffield*, default atomic radii type: *OEAtomicRadii.ZAP9*, default charge model: OEAM1BCCELF10Charges.

SetSzybkiGeneralOptions

```
void SetSzybkiGeneralOptions(const OESzybkiGeneralOptions&)
```

Sets the general szybki options by passing in an *OESzybkiGeneralOptions* instance. These options are used in defining the general environment for all szybki calculations performed from FreeFormConf. Default force field: *OEForceFieldType.MMFF94S*, default temperature: 298.15K.

SetUseInputEnsemble

```
bool SetUseInputEnsemble(const bool)
```

Sets the current state of the use input ensemble flag. If true, the input set of ensemble is used as the initial conformers ensemble and no new conformers are generated during FreeFormConf calculations. Default: `false`.

SetUseSolventCorr

```
bool SetUseSolventCorr(const bool)
```

Sets the current state of the use solvent correction flag. If true, energetic correction due to solvent interactions is applied after entropy calculation. Default: `false`.

3.1.14 OEFreeFormConfResults

```
class OEFreeFormConfResults
```

Container for conformer free-energy calculation results produced by *OEFreeFormConf* and *OEFreeFormConfAdvanced*.

Constructors

```
OEFreeFormConfResults(const OEChem::OEMCMolBase&)
OEFreeFormConfResults(const OEFreeFormConfResults&)
```

Default and copy constructors. The first variation of the constructor populates the object by extracting the corresponding tagged values from the molecule.

GetConfResult

```
OESingleConfResult GetConfResult(const unsigned int) const
```

Returns the *OESingleConfResult* object corresponding to the specified conformer index.

GetLowestdGConfIdx

```
unsigned int GetLowestdGConfIdx() const
```

Returns the conformer index corresponding to the lowest dG.

GetLowestRelEneConfIdx

```
unsigned int GetLowestRelEneConfIdx() const
```

Returns the conformer index corresponding to the lowest relative energy.

GetNumUniqueConfs

```
unsigned int GetNumUniqueConfs() const
```

Returns the number of the unique conformations in the ensemble.

PrintTxt

```
void PrintTxt(OEPlatform::oeostream& fileStream,
              const std::string& molName) const
```

Prints the results in plain text format in the provided output filestream.

PrintCsv

```
void PrintCsv(OEPlatform::oeostream& fileStream,
              const std::string& molName) const
```

Prints the results in comma separated values (CSV) format in the provided output filestream.

GetResultsForConformations

```
OESystem::OEIterBase<OESingleConfResult>* GetResultsForConformations() const
```

Returns the iterator whose elements are *OESingleConfResult* objects for the ensemble of unique conformations.

GetTemperature

```
double GetTemperature() const
```

Returns temperature for which calculations have been performed.

GetTotalEntropy

```
double GetTotalEntropy() const
```

Returns estimated molar solution entropy for the input compound.

3.1.15 OEFreeFormSolvOptions

```
class OEFreeFormSolvOptions : public OESystem::OEOptions
```

Configuration options for *OEEstimateSolvFreeEnergy* solvation free-energy calculations.

Constructors

```
OEFreeFormSolvOptions()
OEFreeFormSolvOptions(const OEFreeFormSolvOptions&)
```

Default and copy constructors.

GetAtomicRadiiType

```
unsigned GetAtomicRadiiType() const
```

Returns the type of atomic radii used for PB calculations. Possible values passed to the method are defined in *OEAtomicRadii* namespace.

GetChargeType

```
unsigned GetChargeType() const
```

Returns the type of partial charges used for Sheffield and PB calculations.

GetConfGenRMS

```
double GetConfGenRMS() const
```

Returns the value of RMSD used by the conformation generator for conformation deduplication. The default values are 0.6 Å for solvation free energy estimation and 0.3 Å for conformation free energy estimation.

GetIonicState

```
unsigned GetIonicState() const
```

Returns the type of charge state. Possible values are defined in the *OEFreeFormIonicState* namespace. The default value is *OEFreeFormIonicState.PH74*.

GetMaxConfGen

```
unsigned GetMaxConfGen() const
```

Returns the maximum conformations number used by the conformation generator.

GetUseInput3D

```
bool GetUseInput3D() const
Determines if the calculations are set to use 3D coordinates of the input molecule.
```

GetUseInputEnsemble

```
bool GetUseInputEnsemble() const
```

Determines if the calculations are set to use the input ensemble of conformations rather than generate the ensemble internally.

SetAtomicRadiiType

```
bool SetAtomicRadiiType(unsigned radii_type)
```

Sets the atomic radii type for all PB calculations. Possible values passed to the method are defined in *OEAtomicRadii* namespace.

SetChargeType

```
bool SetChargeType(unsigned charge_type)
```

Sets the partial charges type. Possible values to pass to the method are defined in a subset of *the `OEProton::OECharges`* namespace (please look at the documentation of The **Quacpac TK**.) The allowed types are: `OECharges.NoOp` (to use existing partial charges), `OECharges.AM1BCCSym`, `OECharges.AM1BCCNoSym`, `OECharges.AM1BCCSymSPt`, `OECharges.AM1BCCNoSymSPt` and `OECharges.MMFF94`. The default values are `OECharges.AM1BCCNoSymSPt` and `OECharges.AM1BCCSym` for solvation free energy estimation and conformation free energy estimation respectively.

SetConfGenRMS

```
bool SetConfGenRMS(double rms)
```

Sets the value of RMSD used by the conformation generator for conformation deduplication. The default values are 0.6 Å for solvation free energy estimation and 0.3 Å for conformation free energy estimation.

SetIonicState

```
bool SetIonicState(unsigned ionic_state)
```

Sets the charge state of the molecule. Possible values are defined in the *OEFreeFormIonicState* namespace. The default value is *OEFreeFormIonicState.PH74*.

SetMaxConfGen

```
bool SetMaxConfGen(unsigned nmax)
```

Sets the maximum number of conformations generated by the conformation generator. The default values are 200 for solvation free energy estimation and 20000 for conformation free energy estimation.

SetUseInput3D

```
bool SetUseInput3D(bool use3D)
```

When parameter `use3D` is set to `true`, solvation free energy is calculated just for the input structure representing specific conformation. This option allows to find what is the difference in solvation energy between different conformations.

SetUseInputEnsemble

```
bool SetUseInputEnsemble(bool use_ensemble)
```

Sets the usage of the input ensemble rather than generating it internally when the parameter `use_ensemble` is set to `true`.

> **Note**
>
> The value set by the *OEFreeFormSolvOptions.SetUseInput3D* method takes precedence over *OEFreeFormSolvOptions.SetUseInputEnsemble*. In other words when the first of those methods passes boolean value `true`, solvation code is looking for a single 3D structure (conformation) provided by the user and determines its solvation free energy. When the value passed by this method is `false` or when it is not used, solvation code is selecting the lowest force field energy conformation either from the user ensemble (when medthod *OEFreeFormSolvOptions.SetUseInputEnsemble* passes boolean `true`) or from the ensemble generated internally and determines its solvation free energy.

3.1.16 OEFreeFormSolvResults

```
class OEFreeFormSolvResults
```

Container for solvation free-energy results returned by *OEEstimateSolvFreeEnergy*.

Constructors

```
OEFreeFormSolvResults()
OEFreeFormSolvResults(const OEFreeFormSolvResults&)
```

Default and copy constructors.

GetAtomicPotentials

```
void GetAtomicPotentials(std::vector<double>& apot) const
```

Fills the vector passed as an arguments with calculated atomic potentials. Each calculated atomic potential is a sum of electrostatic and hydrophobic contributions.

GetElectrostaticSolvationFreeEnergy

```
double GetElectrostaticSolvationFreeEnergy() const
```

Returns electrostatic part of solvation free energy after calling *OEEstimateSolvFreeEnergy* function.

GetHydrophobicSolvationFreeEnergy

```
double GetHydrophobicSolvationFreeEnergy() const
```

Returns hydrophobic part of solvation free energy after calling *OEEstimateSolvFreeEnergy* function.

GetReturnCode

```
unsigned int GetReturnCode() const
```

Returns one of the values defined in the *OEFreeFormReturnCode* namespace.

GetSolvationFreeEnergy

```
double GetSolvationFreeEnergy() const
```

Returns total Gibbs free energy of hydration for the input compound after calling *OEEstimateSolvFreeEnergy* function.

GetWarnings

```
void GetWarnings(std::vector<unsigned int>& warnings) const
```

Fills the vector passed as an argument with warning codes relevant to the calculation performed. When no warnings are generated the returned vector is empty.

3.1.17 OELigandChargeOptions

```
class OELigandChargeOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for assigning ligand charges.

The *OELigandChargeOptions* class defines the following public methods:

- *GetGenerateElfConfs* and *SetGenerateElfConfs*

- *GetLigandCharge*, *GetLigandChargeType*, and *SetLigandCharge*

Constructor

```
OELigandChargeOptions()
OELigandChargeOptions(const OELigandChargeOptions &)
```

Default and copy constructors.

Assignment operator.

GetGenerateElfConfs

```
bool GetGenerateElfConfs() const
```

See *SetGenerateElfConfs* method.

GetLigandCharge

```
const OEProton::OEChargeEngineBase* GetLigandCharge() const
const OEProton::OEChargeEngineBase* GetLigandCharge() const
```

See *SetLigandCharge* method.

GetLigandChargeType

```
unsigned GetLigandChargeType() const
```

See *SetLigandCharge* method.

SetGenerateElfConfs

```
bool SetGenerateElfConfs(const bool)
```

Sets flag if **conformers should be generated** for ELF10 conformer selection. **Default: False.**

SetLigandCharge

```
bool SetLigandCharge(const unsigned)
bool SetLigandCharge(const std::string&)
bool SetLigandCharge(const OEProton::OEChargeEngineBase&)
```

Sets the **charging method** for the ligand. The first overload takes an *unsigned* from the *OELigandChargeType* namespace, and the second overload takes corresponding string values. The final overload allows the use of a user-defined OEChargeEngineBase. Method returns `false` when the invalid selection is chosen, `true` otherwise. **Default:** *OELigandChargeType.CURRENT*

3.1.18 OELigandEntropyOptions: public OELigandChargeOptions

```
class OELigandEntropyOptions : public OESystem::OEOptions
```

Used for entropy estimation of free ligands in solution or in vacuum.

See also

- *OEEntropyResults* class
- *OELigandEntropy* function

The following methods are publicly inherited from *OELigandChargeOptions*:

- *GetGenerateElfConfs* and *SetGenerateElfConfs*
- *GetLigandCharge*, *GetLigandChargeType*, and *SetLigandCharge*

The *OELigandEntropyOptions* class defines the following public methods:

- *GetEnvironment* and *SetEnvironment*
- *GetForceField*, *GetForceFieldType* and *SetForceField*

Constructors

```
OELigandEntropyOptions()
OELigandEntropyOptions(const OELigandEntropyOptions &)
```

Default and copy constructors.

GetEnvironment

```
unsigned int GetEnvironment() const
```

Returns one of 2 types of environement *Gas*, *Solution*

GetForceField

```
const OEMolPotential::OEForceField* GetForceField() const
```

See *SetForceField* method.

GetForceFieldType

```
unsigned GetForceFieldType() const
```

See *SetForceField* method.

SetEnvironment

```
bool SetEnvironment(unsigned int etype)
```

Selects solution (etype = *OEEnvType.Solution*) or gas (etype = *OEEnvType.Gas*) state.

SetForceField

```
bool SetForceField(const unsigned)
bool SetForceField(const std::string&)
bool SetForceField(const OEMolPotential::OEForceField&)
```

Sets the **force field** for optimization. The first overload takes an *unsigned* from the *OESz_OEForceFieldType* constant namespace, and the second overload takes corresponding string values. The final overload allows the use of a user-defined OEForceField. Method returns `false` when the invalid selection is chosen, `true` otherwise. **Default:** *OEForceFieldType.SAGE_OPENFF*

3.1.19 OEProteinFlexOptions

```
class OEProteinFlexOptions : public OESystem::OEOptions
```

This class provides an interface to setup protein flexibility for flexible protein optimization, as used in *OEFlexProteinLigandOptimizer*.

The *OEProteinFlexOptions* class defines the following public methods:

- *GetFlexRange* and *SetFlexRange*
- *GetResidueID* and *SetResidueID*

Constructor

```
OEProteinFlexOptions()
OEProteinFlexOptions(const OEProteinFlexOptions &)
```

Default and copy constructors.

GetFlexRange

```
double GetFlexRange() const
```

See *SetFlexRange* method.

GetResidueID

```
std::string GetResidueID() const
```

See *SetResidueID* method.

SetFlexRange

```
bool SetFlexRange(const double)
```

Sets the **flexibility range** for protein flexibility. All the complete residues of the protein within the specified range (distance) from the ligand are considered flexible. Default: 2.0.

SetResidueID

```
bool SetResidueID(const std::string value)
```

Sets the **residue ID** of the residues to be treated as flexible, using regular expressions. If this is specified, the *FlexRange* is ignored. Default: None.

3.1.20 OEProteinLigandOptOptions

```
class OEProteinLigandOptOptions : public OELigandChargeOptions
```

This class provides an interface to setup options required for optimization with *OEFixedProteinLigandOptimizer* and *OEFlexProteinLigandOptimizer*.

The following methods are publicly inherited from *OELigandChargeOptions*:

- *GetGenerateElfConfs* and *SetGenerateElfConfs*
- *GetLigandCharge*, *GetLigandChargeType*, and *SetLigandCharge*

The *OEProteinLigandOptOptions* class defines the following public methods:

- *GetGradTolerance* and *SetGradTolerance*
- *GetForceField* and *SetForceField*
- *GetMaxIter* and *SetMaxIter*
- *GetNonBondCutoff* and *SetNonBondCutoff*
- *GetOptimizationType* and *SetOptimizationType*
- *GetSolventModel* and *SetSolventModel*

Constructor

```
OEProteinLigandOptOptions()
OEProteinLigandOptOptions(const OEProteinLigandOptOptions &)
```

Default and copy constructors.

Assignment operator.

GetGradTolerance

```
double GetGradTolerance() const
```

See *SetGradTolerance* method.

GetForceField

```
const OEFF::OEComplexFF* GetForceField() const
```

See *SetForceField* method.

GetMaxIter

```
unsigned GetMaxIter() const
```

See *SetMaxIter* method.

GetNonBondCutoff

```
double GetNonBondCutoff() const
```

See *SetNonBondCutoff* method.

GetOptimizationType

```
unsigned GetOptimizationType() const
```

See *SetOptimizationType* method.

GetSolventModel

```
unsigned GetSolventModel() const
```

See *SetSolventModel* method.

SetGradTolerance

```
bool SetGradTolerance(const double)
```

Sets the **gradient tolerance** for optimization. Default: 1.e-6.

SetForceField

```
bool SetForceField(const unsigned)
bool SetForceField(const std::string&)
bool SetForceField(const OEFF::OEComplexFF&)
```

Sets the **force field** for optimization. The first overload takes an *unsigned* from the *OEComplexFFType* namespace, and the second overload takes corresponding string values. The final overload allows the use of a user-defined OEComplexFF. Method returns `false` when the invalid selection is chosen, `true` otherwise. **Default:** *OEComplexFFType.FF14SB_SAGE*

SetMaxIter

```
bool SetMaxIter(const unsigned)
```

Sets the **maximum iterations** for optimization. Default: 1000.

SetNonBondCutoff

```
bool SetNonBondCutoff(const double)
```

Sets the **Non-bonded interactions cutoff** for optimization. Setting a value of `0.0` or less assumes that all interactions should be included without any cutoff. Default: 0.0.

SetOptimizationType

```
bool SetOptimizationType(const unsigned)
```

Sets the **type** for optimization. Allowed methods are defined in the *OEOptimizationType* namespace. Default is *OEOptimizationType.PoseCartesian*.

SetSolventModel

```
bool SetSolventModel(const unsigned)
```

Sets the **solvent model to be used** during optimization. Allowed methods are *OESolventModel.NoSolv* and *OESolventModel.Sheffield*, as defined in the *OESolventModel* namespace. Default is *OESolventModel.NoSolv*.

3.1.21 OEProteinLigandOptResults

```
class OEProteinLigandOptResults
```

This class provides results of optimization of a single conformer of a molecule, obtained from *OEFixedProteinLigandOptimizer* and *OEFlexProteinLigandOptimizer* calculations.

The *OEProteinLigandOptResults* class defines the following public methods:

- *GetReturnCode*
- *GetInitialEnergies*
- *GetFinalEnergies*
- *GetLigandRMSD*
- *GetProteinRMSD*
- *GetTotalRMSD*

Constructors

```
OEProteinLigandOptResults()
OEProteinLigandOptResults(const OEProteinLigandOptResults&)
```

Default and copy constructors.

GetReturnCode

```
unsigned GetReturnCode() const
```

Returns the calculation return code, defined in *OESzybkiReturnCode* namespace.

GetInitialEnergies

```
const OEFF::OEComplexEnergies& GetInitialEnergies() const
```

Returns initial energies of the protein-ligand system, before optimization.

GetFinalEnergies

```
const OEFF::OEComplexEnergies& GetFinalEnergies() const
```

Returns final energies of the protein-ligand system, after optimization.

GetLigandRMSD

```
double GetLigandRMSD() const
```

Returns the RMSD between the initial and final conformations of the ligand.

GetProteinRMSD

```
double GetProteinRMSD() const
```

Returns the RMSD between the initial and final conformations of the protein.

GetTotalRMSD

```
double GetTotalRMSD() const
```

Returns the RMSD between the initial and final conformations of the protein-ligand complex.

3.1.22 OERestrictionEnergyResult

```
class OERestrictionEnergyResult
```

Container for restriction energy results for a single conformer.

Constructors

```
OERestrictionEnergyResult()
OERestrictionEnergyResult(const OEChem::OEMolBase& argConf)
OERestrictionEnergyResult(const OERestrictionEnergyResult&)
```

Default and copy constructors.

GetGlobalStrain

```
double GetGlobalStrain() const
```

Returns the global strain (in kcal/mol) associated with this conformer, due to being restricted.

GetLocalStrain

```
double GetLocalStrain() const
```

Returns the local strain (in kcal/mol) associated with this conformer, due to being restricted.

3.1.23 OESingleConfResult

```
class OESingleConfResult
```

Provides per-conformer access when iterating a *OEFreeFormConfResults* object. Each item gives properties for the conformer identified by *OESingleConfResult.GetConfIdx*.

The *OESingleConfResult* class defines the following public methods:

- *GetConfIdx*
- *GetDeltaE*
- *GetDeltaG*
- *GetLnQrotational*
- *GetLnQvibrational*

- *GetMMFFEnergy*
- *GetProbability*
- *GetRelativeEnergy*
- *GetRelLnQ*
- *GetRotationalEntropy*
- *GetSolvEnergy*
- *GetTotalEnergy*
- *GetVibrationalEnergy*
- *GetVibrationalEntropy*

The *OESingleConfResult* class deprecated public methods:

- *GetRMSD*

Constructors

```
OESingleConfResult()
OESingleConfResult(const OEChem::OEConfBase&)
OESingleConfResult(const OESingleConfResult&)
```

Default and copy constructors.

GetConfIdx

```
unsigned int GetConfIdx() const
```

Returns the conformation id for that conformation.

GetDeltaE

```
double GetDeltaE() const
```

Returns the relative intramolecular energy for that conformation, with respect to the lowest intramolecular energy conformer of the whole ensemble.

GetDeltaG

```
double GetDeltaG() const
```

Returns the free energy of selecting that conformation from the ensemble.

GetLnQrotational

```
double GetLnQrotational() const
```

Returns the natural logarithm of the rotational partition function for that conformation.

GetLnQvibrational

```
double GetLnQvibrational() const
```

Returns the natural logarithm of the vibrational partition function for that conformation.

GetMMFFEnergy

```
double GetMMFFEnergy() const
```

Returns the total intermolecular energy of that conformation.

GetProbability

```
double GetProbability() const
```

Returns the probability of state associated with this conformation.

GetRelativeEnergy

```
double GetRelativeEnergy() const
```

Returns the total relative energy (intramolecular + solvation energy) for that conformation with respect to the lowest energy conformer of the whole ensemble.

GetRelLnQ

```
double GetRelLnQ() const
```

Returns the natural logarithm of the relative vibrational-rotational partition function with respect to the lowest free energy conformer: $\ln q_{rel,i} = \ln q_i - \ln q_{ref}$. In that formula the reference conformation is the one with the lowest value returned by the *OESingleConfResult.GetDeltaG* method. This quantity allows us to calculate conformer relative free energies analogously to conventional relative energies.

GetRotationalEntropy

```
double GetRotationalEntropy() const
```

Returns the rotational entropy (in J/(mol K)) for that conformation.

GetSolvEnergy

```
double GetSolvEnergy() const
```

Returns the solvation energy for that conformation.

GetTotalEnergy

```
double GetTotalEnergy() const
```

Returns the total energy (intramolecular + solvation) for that conformation.

GetVibrationalEnergy

```
double GetVibrationalEnergy() const
```

Returns the relative vibrational energy for that conformation, with respect to the lowest vibrational energy conformer of the whole ensemble.

GetVibrationalEntropy

```
double GetVibrationalEntropy() const
```

Returns the vibrational entropy (in J/(mol K)) for that conformation.

GetRMSD

> **Attention**
>
> This is a deprecated API.
>
> ```
> double GetRMSD() const
> ```

Returns 0.0. This method has been deprecated since RMSD is not a meaningful quantity for a conformer in an ensemble.

3.1.24 OESzybki

```
class OESzybki
```

The primary interface for molecular force field optimization. Supports Cartesian, torsional, and solid-body optimization modes with optional protein-ligand constraints, solvation models, and entropy calculations.

Constructors

```
OESzybki(unsigned int force_field = OEForceFieldType::MMFF94)
OESzybki(const OESzybkiOptions& opts)
```

The default constructor builds *OESzybki* with the MMFF force field. The *OESzybkiOptions* can be provided as argument with specifying many other parameters, including other force fields supported.

operator()

```
OESystem::OEIterBase<OESz::OESzybkiResults>* operator()(OEChem::OEMCMolBase& m)
bool operator()(OEChem::OEMolBase& m, OESz::OESzybkiResults& res)
```

The first operator performs geometry optimization on all conformers in the passed molecule `m` and updates it with optimized coordinates. It returns an iterator whose elements are *OESzybkiResults* objects. *OESzybkiResults* contains the energy values of the relevant potential terms, total energies and gradients. The description of the *OESzybkiResults* structure is given below. The second operator performs the same task on a molecule in a single conformation. A reference to an instance of OESzybkiResults structure has to be passed. If the calculation fails the operator returns `false`.

ClearFixAtoms

```
void ClearFixAtoms()
```

Removes the `fix` constraint on any fixed atoms. After the use of this function, positions of all atoms could change upon optimization.

ClearHarmonicConstraints

```
bool ClearHarmonicConstraints()
```

Removes harmonic constraints. Returns `true` if constraints were removed and `false` when the action was not necessary because a constraining potential was not present.

ClearTorsionConstraint

```
bool ClearTorsionConstraint()
```

Removes torsion constraint. Returns `true` if constraint is removed and `false` when the action was not necessary because a constraining potential was not present.

FixAtoms

```
bool FixAtoms(std::string smarts_pattern)
bool FixAtoms(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase> &)
```

Fixes a set of atoms at their initial positions in space. The first function fixes a set of atoms which belong to a SMARTS pattern determined by the parameter `smarts_pattern`. It returns `true` if the SMARTS string passed is valid, `false` otherwise. The second function fixes only those atoms for which the predicate passed is `true`, and returns `true` if the predicate is valid.

GetEntropy(const OEChem::OEMCMolBase&, unsigned int, unsigned int)

```
double GetEntropy(const OEChem::OEMCMolBase &m,
                  unsigned int method = OEEntropyMethod::QNewton,
                  unsigned int env = OEEnvType::SolutionSPT)
```

Estimates and returns the entropy of a compound in different environments for an ensemble of conformations given in the input molecule `m`. The method does not alter the input set of conformations. The default parameter `method` specifies that the method of entropy evaluation is quasi-Newton. The alternative value for this parameter is *OEEntropyMethod.Analytic*. The last parameter in both methods specifies ligand environment. Its default value refers to the ligand in solution where solvation entropy is handled by scaled-particle theory (SPT). The remaining possible values of the parameter env are: *OEEnvType.Gas*, *OEEnvType.Protein* and *OEEnvType.SolutionSA* which specify gas-phase, protein-bound and solution ligand, respectively. In the case of *OEEnvType.SolutionSA* the solvation entropy is evaluated with the "Surface Area" model, however we recommend the default which uses SPT.

Deprecated since version 1.8.2: Use overloaded *OESzybki.GetEntropy* instead.

GetEntropy

```
double GetEntropy(OEChem::OEMCMolBase& m,
                  OESzybkiEnsembleResults& r,
                  unsigned int method = OEEntropyMethod::QNewton,
                  unsigned int env = OEEnvType::SolutionSPT)
```

Estimates and returns the entropy of a compound in different environments for an ensemble of conformations given in the input molecule `m`. The set of optimized unique conformations is returned in the first parameter `m`. The method takes the reference of the *OESzybkiEnsembleResults* object `r` which carries the entropy results for the ensemble. The default parameter `method` specifies that the method of vibrational entropy evaluation is based on second derivatives matrix (Hessian) which is taken from the final step of quasi-Newton optimization. The alternative value for this parameter is *OEEntropyMethod.Analytic*. When the latter is selected, Hessian matrix is calculated analytically. The last parameter specifies ligand environment. Its default value refers to the ligand in solution where solvation entropy is handled by scaled-particle theory (SPT). The remaining possible values of the parameter `env` are: *OEEnvType.Gas*, *OEEnvType.Protein* and *OEEnvType.SolutionSA* which specify gas-phase, protein-bound and solution ligand respectively. In the case of *OEEnvType.SolutionSA* the solvation entropy is evaluated with the "Surface Area" model, however we recommend the default which uses SPT.

GetFixPattern

```
const char* GetFixPattern() const
```

Returns the SMARTS string used for fixing atoms.

GetFixPredicate

```
const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* GetFixPredicate() const
```

Returns a pointer to the predicate used to determine which atoms are fixed.

GetHarmonicConstraints

```
const OESystem::OEUnaryPredicate<OEChem::OEAtomBase> *
const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* GetHarmonicConstraints(double& kc,
                                                                             double& kd)␣
↪const
```

Provides a method to query for the harmonic constraint used, the force constant `kc`, and distance `kd`. The function returns a pointer to a predicate which determines which atoms are constrained. If the returned pointer is 0, then no harmonic constraints are used.

GetProtein

```
bool GetProtein(OEChem::OEMolBase& p) const
```

Copies the current protein molecule to the passed molecule object `p`. If the *OESzybki* object does not hold a protein molecule, the function will return `false`.

GetSheffieldParameters

```
void GetSheffieldParameters(double& a, double& b, double& dc) const
```

Returns the Sheffield solvation model parameters which are currently used.

LoadPotentialGrid

```
void LoadPotentialGrid(const std::string& fname)
```

Reads a pre-calculated potential grid from the file name `fname`. For a large number of ligands to be optimized inside a single protein with the use of the electrostatic model *OEProteinElectrostatics.GridPB* or *OEProteinElectrostatics.GridCoulomb* (see function *OESzybkiProteinOptions.SetProteinElectrostaticModel*), this function offers a significant saving in CPU time.

SavePotentialGrid

```
void SavePotentialGrid(const std::string& fname)
```

Saves potential grid in the file named `fname`. The potential grid will be generated in protein-ligand calculations in which electrostatic model used is *OEProteinElectrostatics.GridPB* or *OEProteinElectrostatics.GridCoulomb* set with the function *OESzybkiProteinOptions.SetProteinElectrostaticModel*.

SetHarmonicConstraints

```
bool SetHarmonicConstraints(double kc, double kd = 0.0,
                            const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred =␣
↪OEChem::OEIsHeavy())
```

Adds potential term $V = k_c(r - k_d)^2$ for every atom for which the passed predicate operator returns `true`. Default predicate imposes harmonic potential on every heavy atom. Parameter `kc` is the force constant k_c (in $kcal/(mol|A|^2)$) and `kd` is the constraining distance k_d in Å. By default `kd` is set to 0. Function returns `true` if harmonic potential added.

SetProtein

```
bool SetProtein(const OEChem::OEMolBase& protein)
```

Sets the protein molecule for subsequent szybki calculations. The protein molecule may be modified by this operation, if needed. It returns `false` if the operation fails, otherwise `true`.

> **Note**
>
> To specify charging options for protein-ligand coulombic interactions, use the *OESzybkiSolventOptions.SetChargeEngine* method under the *OESzybkiSolventOptions* class. To work with the pre-existing charges, use the OEChargeEngineNoOp as the charging engine.

SetSheffieldParameters

```
void SetSheffieldParameters(double a, double b, double dc = 1.0)
```

Sheffield solvation energy requires two parameters a and b: $E_{solv} = -\frac{f_\epsilon}{8\pi\epsilon_0}\sum_{i,j}\frac{q_i q_j}{\sqrt{(aR_iR_j+br_{ij}^2)}}$ where q_i and q_j are atomic charges, R_i and R_j are vdW radii for atoms i and j, and r_{ij} is the interatomic distance. Parameter f_ϵ depends on dielectric constants of the solute (ϵ) and the solvent (ϵ_{solv}): $f_\epsilon = 1/\epsilon - 1/\epsilon_{solv}$). This function sets both parameters a and b to the required values, as well as the solute dielectric constant (which may not exceed 10). If the function not used, the solute dielectric constant is set to 1.0, and the default values of $a = 1.553149$ and $b = 0.735694$ are adopted ([Grant-2007]).

3.1.25 SetTorsionConstraint

```
bool SetTorsionConstraint(const std::string& tor_pattern, double k,
                          double phi0)
bool SetTorsionConstraint(const OEChem::OETorsion& tor, double k)
```

The method adds a potential term $V = k_c(cos(phi) - cos(phi0))^2$ for a torsion. The first method specifies the torsion with a SMARTS pattern passed as a string `tor pattern`. This must be in the indexed form which show which atoms forms the torsion. For example a valid pattern is `[C:1][N:2][c:3][s:4]`. `k` is the user specified force field in kcal/mol, and `phi0` is the torsion reference dihedral angle in radians. The function returns `true` if a torsion constraint is added successfully.

The second method takes an instance of an OETorsion object, `tor`, which defines a single torsion to be constrained. `k` is the user specified force field in kcal/mol. The function returns `true` if a constrained torsion is added successfully.

3.1.26 UnsetProtein

```
void UnsetProtein() const
```

Removes protein context from the *OESzybki* object.

3.1.27 OESzybkiEnsembleResults

```
class OESzybkiEnsembleResults
```

Container for ensemble geometry optimization results.

Constructors

```
OESzybkiEnsembleResults()
OESzybkiEnsembleResults(const OESzybkiEnsembleResults &)
```

Default and copy constructors.

NumConfs

```
unsigned int NumConfs() const
```

Returns the number of unique conformations in the ensemble.

GetTemperature

```
double GetTemperature() const
```

Returns the temperature in K. This is a convenience method returning the same information as the method of the same name in the OESzybki class.

GetTotalEntropy

```
double GetTotalEntropy() const
```

Returns the total entropy of the ligand in J/(mol K).

GetEntropicEnergy

```
double GetEntropicEnergy() const
```

Returns the product of total entropy and temperature, ST in kcal/mol.

GetChargeType

```
unsigned int GetChargeType() const
```

Returns the type of atomic partial charges used in entropy estimation. The returned type are is either `OECharges.AM1BCCNoSymSPt` or `OECharges.NoOp`.

GetConfigurationalEntropy

```
double GetConfigurationalEntropy() const
```

Returns the ensemble configurational entropy. For gas-phase ligands that value is the same as the value returned by *OESzybkiEnsembleResults.GetTotalEntropy*. For solution ligands the configurational entropy made by translational-vibrational-rotational entropy, and for protein-bound ligands it represents vibrational entropy.

GetEnsembleLigSolvEntropy

```
double GetEnsembleLigSolvEntropy() const
```

Returns the value of the ensemble solvation entropy for a ligand in solution. The method returns zero for gas-phase and protein-bound ligands.

GetEnsembleLigPartialSolvEntropy

```
double GetEnsembleLigPartialSolvEntropy() const
```

Returns the value of solvation entropy for the protein-bound ligand partially exposed to solvent.

GetEnsembleProtDesolvEntropy

```
double GetEnsembleProtDesolvEntropy() const
```

Returns the value of protein desolvation entropy upon ligand binding.

GetResultsForConformations

```
OESystem::OEIterBase<OESzybkiResults>* GetResultsForConformations() const
```

Returns the iterator whose elements are *OESzybkiResults* objects for the ensemble of conformation members optimized during the entropy estimation. Data only for unique conformations are represented in the returned set.

HasEntropy

```
bool HasEntropy() const
```

Returns `true` if the *OESzybkiEnsembleResults* object contains calculated entropy, `false` is returned otherwise.

3.1.28 OESzybkiGeneralOptions

```
class OESzybkiGeneralOptions : public OESystem::OEOptions
```

This class represents general options for szybki.

Constructors

```
OESzybkiGeneralOptions()
OESzybkiGeneralOptions(const OESzybkiGeneralOptions &)
```

Default and copy constructors.

GetCalculateGradients

```
bool GetCalculateGradients() const
```

Returns the state of the calculate gradients flag. If true, OESzybki will calculate and report gradients.

GetForceField

```
const OEMolPotential::OEForceField* GetForceField() const
```

Returns the pointer to the OEForceField object which was used in the method *OESzybkiGeneralOptions.SetForceField*.

GetForceFieldType

```
unsigned int GetForceFieldType() const
```

Returns the force field type to be used for OESzybki calculations, as *unsigned int* from the *OEForceFieldType* namespace.

GetIEFFCluster

```
bool GetIEFFCluster() const
```

Returns the state of the IEFF cluster flag. If true, OESzybki will optimize a cluster of molecules with the MMFF94-IEFF or MMFF94S-IEFF force field.

GetIntramolecularVdWCutoff

```
double GetIntramolecularVdWCutoff() const
```

Returns the value of the intramolecular vdW interactions cutoff. The intramolecular VdW interactions are considered to be zero between atoms that are separated by distances larger than the cutoff.

GetLigandRMSDHeavy

```
bool GetLigandRMSDHeavy() const
```

Returns the state of the Ligand RMSD heavy flag. If true, OESzybki reports RMSD for the optimized molecular systems based on heavy atoms only.

GetRemoveAttractiveVdWForces

```
bool GetRemoveAttractiveVdWForces() const
```

Returns the state of the Remove Attractive VdW Forces terms flag. If true, OESzybki removes the attractive part of the VdW interaction terms from the force field for calculations.

GetRemoveCoulombTerms

```
bool GetRemoveCoulombTerms() const
```

Returns the state of the Remove Coloumb terms flag. If true, OESzybki removes the Coulomb terms from the force field for calculations.

GetSoluteDielectric

```
float GetSoluteDielectric() const
```

Returns the value of ligand dielectric constant used for the optimization of ligand in solution.

GetTemperature

```
double GetTemperature() const
```

Returns the value of temperature to be used for entropy calculation, in units of kelvins.

GetVerbose

```
bool GetVerbose() const
```

Returns the state of the verbose flag. If true, OESzybki is set for reporting optimization progress.

SetCalculateGradients

```
bool SetCalculateGradients(const bool)
```

Sets the state of the calculate gradients flag. If `true` is passed, OESzybki will calculate and report gradients.

SetForceField

```
bool SetForceField(const OEMolPotential::OEForceField& value)
```

Sets the force field to be used for OESzybki calculations, using an instance of OEForceField. This method provids an alternative to setting force field by a predefined type with *OESzybkiGeneralOptions.SetForceFieldType*. Method returns `true` when the setting is successful, `false` otherwise.

SetForceFieldType

```
bool SetForceFieldType(const unsigned int)
```

Sets the force field to be used for OESzybki calculations, using constants defined in the *OEForceFieldType* namespace. This method provids an alternative to setting force field by a force field instance with *OESzybkiGeneralOptions.SetForceField*. Method returns `true` when the setting is successful, `false` otherwise.

SetIEFFCluster

```
bool SetIEFFCluster(const bool)
```

Sets the state of the IEFF cluster flag. If `true` is passed, OESzybki will optimize a cluster of molecules with the MMFF94-IEFF or MMFF94S-IEFF force field.

SetIntramolecularVdWCutoff

```
bool SetIntramolecularVdWCutoff(const double)
```

Sets the value of the intramolecular vdW interactions. The intramolecular VdW interactions are considered to be zero between atoms that are separated by distances larger than the cutoff. By default no cutoff is used.

SetLigandRMSDHeavy

```
bool SetLigandRMSDHeavy(const bool)
```

Sets the state of the Ligand RMSD heavy flag. If `true` is passed, OESzybki reports RMSD for the optimized molecular systems based on heavy atoms only. By default, RMSD is calculated based on all atoms.

SetRemoveAttractiveVdWForces

```
bool SetRemoveAttractiveVdWForces(const bool value)
```

Sets the state of the Remove Attractive VdW Forces terms flag. If true, OESzybki removes the attractive part of the VdW interaction terms from the force field for calculations.

SetRemoveCoulombTerms

```
bool SetRemoveCoulombTerms(const bool)
```

Sets the state of the Remove Coloumb terms flag. If `true` is passed, OESzybki removes the Coulomb terms from the force field for calculations.

SetSoluteDielectric

```
bool SetSoluteDielectric(const double)
```

Sets the value of ligand dielectric constant used for the optimization of ligand in solution.

SetTemperature

```
bool SetTemperature(const double)
```

Returns the value of temperature to be used for entropy calculation, in units of kelvins. The default temperature is set to 298.15K.

SetVerbose

```
bool SetVerbose(const bool)
```

Sets the state of the verbose flag. If `true` is passed, OESzybki is set for reporting optimization progress.

3.1.29 OESzybkiOptions

```
class OESzybkiOptions : public OESystem::OEOptions
```

Configuration options for *OESzybki* optimizations, including force field selection, run type, optimizer algorithm, solvation model, and entropy method.

Constructors

```
OESzybkiOptions()
OESzybkiOptions(const OESzybkiOptions &)
```

Default and copy constructors.

GetGeneralOptions

```
OESzybkiGeneralOptions& GetGeneralOptions()
const OESzybkiGeneralOptions& GetGeneralOptions() const
```

Returns a reference to the *OESzybkiGeneralOptions* instance as currently set. These options are used in defining general environment for all szybki calculations.

GetOptOptions

```
OESzybkiOptOptions& GetOptOptions()
const OESzybkiOptOptions& GetOptOptions() const
```

Returns a reference to the *OESzybkiOptOptions* instance as currently set. These are options related to optimization in OESzybki.

GetProteinOptions

```
OESzybkiProteinOptions& GetProteinOptions()
const OESzybkiProteinOptions& GetProteinOptions() const
```

Returns a reference to the *OESzybkiProteinOptions* instance as currently set. These are options related to proteins in OESzybki.

GetRunType

```
unsigned int GetRunType() const
```

Returns Szybki run type declared in the *OESzybkiOptions* object.

GetSolventOptions

```
OESzybkiSolventOptions& GetSolventOptions()
const OESzybkiSolventOptions& GetSolventOptions() const
```

Returns a reference to the *OESzybkiSolventOptions* instance as currently set. These are options related to solvents in OESzybki.

SetGeneralOptions

```
void SetGeneralOptions(const OESzybkiGeneralOptions&)
```

Sets the general szybki options by passing in an *OESzybkiGeneralOptions* instance. These options are used in defining general environment for all szybki calculations.

SetOptOptions

```
void SetOptOptions(const OESzybkiOptOptions&)
```

Sets the optimization options by passing in *OESzybkiOptOptions* instance. These are options related to optimization in OESzybki.

SetProteinOptions

```
void SetProteinOptions(const OESzybkiProteinOptions&)
```

Sets the protein options by passing in *OESzybkiProteinOptions* instance. These are options related to proteins in OESzybki.

SetRunType

```
bool SetRunType(unsigned int run_type)
```

Sets requested run `type`. Returns `true` if run was successfully set. Possible input choices are given in the namespace *OERunType*. Warning: Setting `run_type` with *OERunType.SolidBodyOpt* implies optimization of a ligand inside rigid protein receptor. An attempt of solid-body optimization inside flexible protein receptor (see method *OESzybkiProteinOptions.SetProteinFlexibilityType*) will result in a message: `Fatal: Solid-body optimization allowed only for ligand in fixed protein`.

SetSolventOptions

```
void SetSolventOptions(const OESzybkiSolventOptions&)
```

Sets the solvent options by passing in *OESzybkiSolventOptions* instance. These are options related to solvents in OESzybki.

3.1.30 OESzybkiOptOptions

```
class OESzybkiOptOptions : public OESystem::OEOptions
```

This class represents optimization related options for szybki.

Constructors

```
OESzybkiOptOptions()
OESzybkiOptOptions(const OESzybkiOptOptions &)
```

Default and copy constructors.

GetCalculateFrozenTerms

```
bool GetCalculateFrozenTerms() const
```

Returns the state of the calculate frozen terms flag. If true, OESzybki will include the constant energy terms in the final optimized energy.

GetGradTolerance

```
double GetGradTolerance() const
```

Returns the optimizer gradient convergence criterion defined as RMS gradient in kcal/(mol Å).

GetMaxIter

```
unsigned int GetMaxIter() const
```

Returns the value of maximum number of optimizer iterations.

GetOptimizerType

```
unsigned int GetOptimizerType() const
```

Returns the optimizer type to be used, as as *unsigned int* from the *OEOptType* namespace.

SetCalculateFrozenTerms

```
bool SetCalculateFrozenTerms(const bool)
```

Allows the omission of the energy calculation of the fixed terms at the end of the optimization. This situation occurs when adaptors are used. By default constant energy terms are calculated for the optimized structure and added to the total energy.

SetGradTolerance

```
bool SetGradTolerance(const double)
```

Sets the optimizer gradient convergence criterion defined as RMS gradient in kcal/(mol Å). If not set by the user a 0.1 value threshold for gradient norm $\sqrt{\sum_i g_i g_i}$, will be used as termination criteria for all optimization types but Newton-Raphson. In the case of the latter the default gradient tolerance is RMS gradient of 1e-5 kcal/(mol Å).

SetMaxIter

```
bool SetMaxIter(const unsigned int)
```

Sets the value of maximum number of optimizer iterations. The default value is set to 1000.

SetOptimizerType

```
bool SetOptimizerType(const unsigned int)
```

Selects the type of the optimization which will be used for optimization. The available values are defined in the *OEOptType* namespace. Method returns `false` when the invalid selection is chosen, `true` otherwise.

3.1.31 OESzybkiProteinOptions

```
class OESzybkiProteinOptions : public OESystem::OEOptions
```

This class represents options related to proteins in szybki.

Constructors

```
OESzybkiProteinOptions()
OESzybkiProteinOptions(const OESzybkiProteinOptions &)
```

Default and copy constructors.

AddFlexibleResidue

```
bool AddFlexibleResidue(const OEChem::OEResidue&)
```

Adds a single residue to the list of residues which will be optimized together with the ligand. Has to be called multiple times in order to make more than one residue flexible.

See also

- Example in *Partially flexible protein optimization*

ClearFlexibleResidues

```
void ClearFlexibleResidues()
```

Clears all of the currently set flexible residues.

GetExactVdWProteinLigand

```
bool GetExactVdWProteinLigand() const
```

Returns the state of the exact VdW protein-ligand flag. If true, OESzybki will be using analytical vdW protein-ligand interaction.

GetIntermolecularVdWCutoff

```
double GetIntermolecularVdWCutoff() const
```

Returns the value of the intermolecular vdW interactions cutoff in Å. The intermolecular VdW interactions are considered to be zero between atoms that are separated by distances larger than the cutoff.

GetProteinDielectric

```
double GetProteinDielectric() const
```

Returns the value of protein dielectric constant.

GetProteinElectrostaticModel

```
unsigned int GetProteinElectrostaticModel() const
```

Returns the protein electrostatics model to be used for OESzybki calculations, as *unsigned int* from the *OEProteinElectrostatics* namespace.

GetProteinFlexibilityRange

```
double GetProteinFlexibilityRange() const
```

Returns the value of partial protein flexibility distance from the ligand.

GetProteinFlexibilityType

```
unsigned int GetProteinFlexibilityType() const
```

Returns the type of partial protein flexibility to be used for OESzybki calculations, as *unsigned int* from the *OEProtFlex* namespace.

HasFlexibleResidues

```
bool HasFlexibleResidues() const
```

Returns the state of flexible residues. If true, some flexible residues are currently set.

SetExactVdWProteinLigand

```
bool SetExactVdWProteinLigand(const bool)
```

Sets the state of the exact VdW protein-ligand flag. If `true` is passed, OESzybki will be using analytical vdW protein-ligand interaction.

SetIntermolecularVdWCutoff

```
bool SetIntermolecularVdWCutoff(const double)
```

Sets the value of the intermolecular vdW interactions cutoff in Å. The intermolecular VdW interactions are considered to be zero between atoms that are separated by distances larger than the cutoff. The default value is 18 Å, however when Newton-Raphson optimization is used (see method *OESzybkiOptOptions.SetOptimizerType*) the default cutoff of 50 Å is used.

SetProteinDielectric

```
bool SetProteinDielectric(const double)
```

Sets the value of protein dielectric constant for the calculation of protein-ligand interaction. The default value of the protein dielectric constant is 1.0.

SetProteinElectrostaticModel

```
bool SetProteinElectrostaticModel(unsigned int)
```

Sets the protein electrostatics model to be used for OESzybki calculations, as *unsigned int* from the *OEProteinElectrostatics* namespace. Method returns `false` when the invalid selection is chosen, `true` otherwise.

SetProteinFlexibilityRange

```
bool SetProteinFlexibilityRange(const double)
```

Sets the value of partial protein flexibility distance from the ligand.

SetProteinFlexibilityType

```
bool SetProteinFlexibilityType(const unsigned int)
```

Sets the type of partial protein flexibility to be used for OESzybki calculations, as *unsigned int* from the *OEProtFlex* namespace. Method returns `false` when the invalid selection is chosen, `true` otherwise.

3.1.32 OESzybkiResults

```
class OESzybkiResults
```

Container for geometry optimization results, including final energies, individual force field component contributions, RMSD from starting geometry, and convergence status.

Constructors

```
OESzybkiResults()
OESzybkiResults(const OESzybkiResults &)
```

Default and copy constructors.

Clear

```
void Clear()
```

Removes all data from the *OESzybkiResults* object.

GetCPUTime

```
float GetCPUTime() const
```

Returns CPU time in seconds for the optimization.

GetConfIdx

```
unsigned int GetConfIdx() const
```

Returns conformer id number.

GetEnergyTerm

```
double GetEnergyTerm(unsigned int) const
```

Returns the current energy value for a specified potential term. The integer values which determine potential terms are defined in the namespace *OEPotentialTerms*

GetFinalRMSGradient

```
double GetFinalRMSGradient() const
```

Returns final RMS of forces, $\sqrt{\frac{\mathbf{g}\cdot\mathbf{g}}{n_v}}$, where $\mathbf{g}$ is the gradient vector and n_v the number of variables.

GetGradients

```
bool GetGradients(double* gradients, const OEChem::OEAtomBase* atom) const
```

Fills the passed double array with gradients for the atom passed as a second argument. Returns `true` if the gradients are successfully exported, `false` otherwise. Export of gradients is available only when the method *OESzybkiGeneralOptions.SetCalculateGradients* is called prior to the calculation performed with one of the *OESzybki.operator()* operators. Currently calculation of gradients is available only for single-point runs.

GetFinalTotalPotential

```
double GetFinalTotalPotential() const
```

Returns the value of the optimized potential for the molecular system.

GetIntramolecularLigandEnergy

```
double GetIntramolecularLigandEnergy() const
```

Returns the value of the value of intramolecular MMFF energy.

GetInitialRMSGradient

```
double GetInitialRMSGradient() const
```

Returns the initial RMS of forces, $\sqrt{\frac{\mathbf{g}\cdot\mathbf{g}}{n_v}}$, where $\mathbf{g}$ is the gradient vector and n_v the number of variables.

GetInitialTotalPotential

```
double GetInitialTotalPotential() const
```

Returns the value of the initial potential for the molecular system to be optimized.

GetInterEnergy

```
double GetInterEnergy() const
```

Returns the interaction energy between protein (or DNA) and the ligand optimized inside the macromolecule. For ligands optimized in vacuum or in solution, this function returns 0.

GetMaxDisplacement

```
double GetMaxDisplacement() const
```

Returns the maximum atomic displacement for a single atom in Å during optimization.

GetNumCycles

```
unsigned int GetNumCycles() const
```

Returns the number of cycles performed by the optimizer.

GetNumFixAtoms

```
unsigned int GetNumFixAtoms() const
```

Returns the number of atoms which will be fixed during optimization.

GetNumRotors

```
unsigned int GetNumRotors() const
```

Returns the number of rotatable bonds in the molecule.

GetProteinRMSD

```
double GetProteinRMSD() const
```

In the case of a partially optimized protein (residues, side chains, polar hydrogens in proximity to the ligand) the function returns the RMS displacement of a protein from its initial structure.

GetRMSD

```
double GetRMSD() const
```

Returns the RMS displacement of the optimized structure with respect to the initial structure.

GetTotalEnergy

```
double GetTotalEnergy() const
```

Returns the total energy of the optimized system. That includes all intra and inter molecular MMFF terms, solvation energy (Sheffield or PB, if present) and protein-ligand interaction energy if a ligand is optimized inside the protein. Harmonic energy constraint energy is excluded.

GetTotalEnergyWithHarmConstraint

```
double GetTotalEnergyWithHarmConstraint() const
```

Returns total energy of the optimized system plus harmonic energy constraint.

IsActiveTerm

```
bool IsActiveTerm(unsigned int term) const
```

Returns `true` if the potential term specified by the parameter term is included in the potential function of the system. Possible parameters values are defined in the namespace *OEPotentialTerms*.

GetConfFreeEnergyFromEnsemble

```
double GetConfFreeEnergyFromEnsemble() const
```

Returns the free energy of selecting a conformation out of the ensemble calculated as: $-RTln(q/Q)$, where q is the partition function of a ligand conformer and Q is the partition function for the entire ensemble. The method will throw an error unless it is run following the call to *OESzybki.GetEntropy* method which takes the *OESzybkiEnsembleResults* instance as a second parameter, followed by the call to *OESzybkiEnsembleResults.GetResultsForConformations* on the returned *OESzybkiEnsembleResults* object.

GetVibEntropy

```
double GetVibEntropy() const
```

Returns the vibrational entropy of a conformation in e.u. (cal/(mol K)) when entropy has been calculated with one of the *OESzybki.GetEntropy* methods.

GetRotEntropy

```
double GetRotEntropy() const
```

Returns the rotational entropy of a conformation in e.u. (cal/(mol K)) when entropy has been calculated with one of the *OESzybki.GetEntropy* methods.

GetLnQvib

```
double GetLnQvib() const
```

Returns the natural logarithm of a vibrational partition function when entropy has been calculated with one of the *OESzybki.GetEntropy* methods.

GetLnQrot

```
double GetLnQrot() const
```

Returns the natural logarithm of a rotational partition function when entropy has been calculated with one of the *OESzybki.GetEntropy* methods.

IsUnique

```
bool IsUnique() const
```

Returns `true` if the conformation is unique in terms of the structure. Although the method *OESzybki.GetEntropy* which takes a non-const OEMCMolBase object is guaranteed to return an ensemble of unique conformations, the other methods in oeszybki library do not. The method might be therefore useful to make sure if the current conformation is unique or not.

Print

```
void Print(OEPlatform::oeostream &) const
void Print(OESystem::OEErrorHandler &) const
```

Both functions allow the generation of log information on the optimized system. The following information is reported:

- Conformer id
- Number of fixed atoms (if any)
- Number of torsions (if any and if optimization in torsion space is true)
- Initial energy
- Initial rms gradient
- Final energy
- Final rms gradient
- RMS displacement upon optimization
- Maximum displacement during optimization
- Values of potential terms at final geometry

3.1.33 OESzybkiSolventOptions

```
class OESzybkiSolventOptions : public OESystem::OEOptions
```

This class represents options related to solvent modeling in szybki.

Constructors

```
OESzybkiSolventOptions()
OESzybkiSolventOptions(const OESzybkiSolventOptions &)
```

Default and copy constructors.

Assignment operator.

GetAtomicRadiiType

```
unsigned int GetAtomicRadiiType() const
```

Returns the type of atomic radii used for PB calculations, as an *unsigned int* from the *OEAtomicRadii* namespace.

GetCavitySolventParameter

```
double GetCavitySolventParameter() const
```

Returns the microscopic surface tension coefficient (in kcal/(mol A**2)) used in hydrophobic ligand solvation model.

GetSaltConcentration

```
double GetSaltConcentration() const
```

Returns the value of salt concentration in moles/liter (M).

GetSolventDielectric

```
double GetSolventDielectric() const
```

Returns the value of solvent dielectric constant for the protein-ligand system.

GetSolventModel

```
unsigned int GetSolventModel() const
```

Returns the type of solvent model used for the calculations of solution ligands, as an *unsigned int* from the *OESolventModel* namespace.

SetAtomicRadiiType

```
bool SetAtomicRadiiType(const unsigned int)
```

Sets the type of atomic radii to be used for PB calculations, as an *unsigned int* from the *OEAtomicRadii* namespace.

SetCavitySolventParameter

```
bool SetCavitySolventParameter(const double)
```

Sets the microscopic surface tension coefficient (in kcal/(mol A**2)) used in hydrophobic ligand solvation model. The default value is set to be 0.025 kcal/(mol A**2). Smaller values usually in the range 0-10 kcal/(mol A**2) usually reflects a combined hydrophobic solvation effect of cavitation and solute-solvent vdW interactions.

SetSaltConcentration

```
bool SetSaltConcentration(const double)
```

Sets the value of salt concentration in moles/liter (M). By default a zero salt concentration is assumed. Values higher than 0.08 should not be used and are not supported.

SetSolventDielectric

```
bool SetSolventDielectric(const double)
```

Sets the value of solvent dielectric constant for the protein-ligand systems in PB calculations. The default value of solvent dielectric constant is set to 80.0.

SetSolventModel

```
bool SetSolventModel(const unsigned int)
```

Sets the type of solvent model used for the calculations of solution ligands, as an *unsigned int* from the *OESolventModel* namespace. Method returns `false` when the invalid selection is chosen, `true` otherwise.

SetChargeEngine

```
void SetChargeEngine(const OEProton::OEChargeEngineBase&)
```

Sets the charging engine to obtain partial charges used for solvation calculation by passing in an OEChargeEngineBase instance.

SetUseCurrentCharges

```
void SetUseCurrentCharges()
```

Sets the charging engine to work with the existing charges. This command is equivalent to using the *OESzybkiSolventOptions.SetChargeEngine* method with an OEChargeEngineNoOp instance as input.

3.1.34 OETorsionScanOptions

```
class OETorsionScanOptions : public OELigandChargeOptions
```

Configuration options for *OETorsionScan* torsion-energy scans.

See also

- *OETorsionScanResult* class
- *OETorsionScan* function

The following methods are publicly inherited from *OELigandChargeOptions*:

- *GetGenerateElfConfs* and *SetGenerateElfConfs*
- *GetLigandCharge*, *GetLigandChargeType*, and *SetLigandCharge*

The *OETorsionScanOptions* class defines the following public methods:

- *GetDelta* and *SetDelta*
- *GetForceField*, *GetForceFieldType* and *SetForceField*
- *GetMaxIter* and *SetMaxIter*
- *GetPenaltyForceConstant* and *SetPenaltyForceConstant*
- *GetSolvationType* and *SetSolvationType*
- *GetTolerance* and *SetTolerance*
- *GetUseInternalCoord* and *SetUseInternalCoord*
- *GetUserDefinedAngles* and *SetUserDefinedAngles*
- *GetUseSinglePointPB* and *SetUseSinglePointPB*
- *GetNumStarts* and *SetNumStarts*
- *GetOverlapDiv* and *SetOverlapDiv*

Constructors

```
OETorsionScanOptions()
OETorsionScanOptions(const OETorsionScanOptions &)
```

Default and copy constructors.

GetDelta

```
double GetDelta() const
```

Returns the value of dihedral angle increase (in degrees).

GetForceField

```
const OEMolPotential::OEForceField* GetForceField() const
```

Returns pointer to the user defined force field.

GetForceFieldType

```
unsigned int GetForceFieldType() const
```

Returns the type of force field defined.

GetMaxIter

```
unsigned GetMaxIter() const
```

Returns the number of maximum iterations during optimization.

GetPenaltyForceConstant

```
double GetPenaltyForceConstant() const
```

Returns force constant for the dihedral angle constraint used for torsion scan.

GetSolvationType

```
unsigned int GetSolvationType() const
```

Returns the type of solvation model used for torsion scan, see *OETorsionScanOptions.SetSolvationType* for more details.

GetTolerance

```
double GetTolerance() const
```

Returns the tolerance during optimization.

GetUseInternalCoord

```
bool GetUseInternalCoord() const
```

Returns `true` if *SetUseInternalCoord* is set to using internal coordinates for optimization while the torsion selected for scanning is fixed.

GetUserDefinedAngles

```
std::vector<double> GetUserDefinedAngles() const
```

Returns vector of user defined dihedral angles for which torsion scan will be done.

3.1.35 GetUseSinglePointPB

```
bool GetUseSinglePointPB() const
```

Returns `true` if the *OETorsionScanOptions.SetUseSinglePointPB* is set to correcting the torsion scan with a PB solvation energy.

3.1.36 GetNumStarts

```
unsigned GetNumStarts() const
```

Returns the number of starts for the torsion scan. See *OETorsionScanOptions.SetNumStarts* for more details

3.1.37 GetOverlapDiv

```
unsigned GetOverlapDiv() const
```

Returns the divisor used for determining the size of the overlaps between starts. See *OETorsionScanOptions.SetOverlapDiv* for more details.

3.1.38 SetDelta

```
bool SetDelta(double delta)
```

Determines the number of points in that is $(360.0/delta + 1)$ in the $[0.0°, 360.0°]$ range for the scan. Minimum and maximum values are `0.1` and `60.0` degrees respectively. The default value is `5.0` degrees. When the given value of `delta` is outside the range, `false` is returned.

3.1.39 SetForceField

```
bool SetForceField(const OEMolPotential::OEForceField& value)
bool SetForceField(const std::string& value)
bool SetForceField(OEPlatform::oeistream& value)
bool SetForceField(const unsigned value)
```

Sets user defined force field which will be used for torsion scan.

3.1.40 SetForceFieldType

```
bool SetForceFieldType(unsigned int type)
```

Selects the force field for the scan. The allowed values for parameter `type` are *OEForceFieldType.MMFF94S* (default), *OEForceFieldType.MMFF94*, *OEForceFieldType.SMIRNOFF99FROSST*, *OEForceFieldType.PARSLEY_OPENFF100*, *OEForceFieldType.PARSLEY_OPENFF111*, *OEForceFieldType.PARSLEY_OPENFF121*, *OEForceFieldType.PARSLEY_OPENFF131*, *OEForceFieldType.SAGE_OPENFF200*, and *OEForceFieldType.SAGE_OPENFF210* Method returns `false` if other than the above values are passed.

3.1.41 SetMaxIter

```
bool SetMaxIter(const unsigned value)
```

Sets the number of maximum iterations for optimization.

3.1.42 SetSolvationType

```
bool SetSolvationType(unsigned int type)
```

Selects the solvation model used for the scan. The allowed values of `type` are: *OESolventModel.NoSolv* (vacuum) and *OESolventModel.Sheffield* (aqueous solution, default). Method returns `false` if invalid solvation model is passed.

3.1.43 SetPenaltyForceConstant

```
bool SetPenaltyForceConstant(double pfc)
```

Overwrites a default value (`1e4`) for force constant in dihedral angle constraint used for torsion scan. Although for most molecules a default value works fine, in some cases a molecular torsion might be over-constrained or underconstrained. This method allows to avoid those two extremes. Method returns `false` if invalid value of force constant is passed.

3.1.44 SetTolerance

```
bool SetTolerance(const double)
```

Sets the tolerance limit for optimization.

3.1.45 SetUseInternalCoord

```
bool SetUseInternalCoord(bool useInt)
```

Overwrites a default method for torsion scan, which applies a large enough penalty force to keep the value of the dihedral angle constant during optimization. When the parameter `useInt` is set to `true`, optimization is performed in 3N-5 internal coordinates while the selected torsion for which a scan is done is kept frozen.

3.1.46 SetUserDefinedAngles

```
bool SetUserDefinedAngles(const std::vector<double>& angles)
```

Takes a vector of user selected dihedral angles (in deg) for which torsion scan will be done. Vector passed as argument should contain at least 2 values, otherwise `false` is returned.

3.1.47 SetUseSinglePointPB

```
bool SetUseSinglePointPB(bool usePB)
```

Corrects the final scan values with a PB solvation energy, when the solvation model is set to *OESolventModel.Sheffield* (see *OETorsionScanOptions.SetSolvationType*). Method returns `false` if the solvation model is not correctly set.

3.1.48 SetNumStarts

```
bool SetNumStarts(unsigned starts)
```

Sets the number of starts for the torsion scan. The default value is 2, meaning scans will start at 0 and 180 degrees. The scans are performed both forwards and backwards. This setting is not compatible with setting *OETorsionScanOptions.SetUserDefinedAngles*, and will be ignored in case user defined angles are supplied.

3.1.49 SetOverlapDiv

```
bool SetOverlapDiv(unsigned overlapDiv)
```

Sets the divisor used for determining the size of the overlaps between starts. If set to zero or one, will perform full 360 forwards and backwards scans for each start, meaning the number of starts will increase the computational cost linearly. To prevent the linear increase in cost for additional starts, the overlap divisor in combination with a number of starts > 1, will first divide the angle space by the number of starts, then to ensure some overlap between the scanned angles will take the remaining space and divide by this value (default value is 4). For example with the default values of 2 starts, scans would be performed from `0->180`, `0->-180`, `180->360`, and `180->0`. The `180` degree remainder (if a full scan would have been done) is instead divided by 4, which equals `45` degrees, which is then added to each scan to ensure overlaps of the curves. The resulting scan will look like `0->225`, `0->-225`, `180->45` (through `360`), and `180->-45`. The larger the overlap divisor the smaller the overlaps between curves. This setting is not compatible with setting *OETorsionScanOptions.SetUserDefinedAngles*, and will be ignored in case user defined angles are supplied.

3.1.50 OETorsionScanResult

```
class OETorsionScanResult
```

A single data point from a torsion scan, storing a torsion angle (in degrees) and its corresponding force field energy (in kcal/mol).

See also

- *OETorsionScanOptions* class
- *OETorsionScan* function

Constructors

```
OETorsionScanResult(double dihedral_angle, double energy)
OETorsionScanResult(const OETorsionScanResult &)
```

Default and copy constructors. Default constructor takes values of torsion angle (in degrees) and corresponding energy (in kcal/mol).

GetAngle

```
double GetAngle() const
```

Returns the value of dihedral angle (in degrees) for a torsion scan point.

GetEnergy

```
double GetEnergy() const
```

Returns the value of relative energy (in kcal/mol) for a torsion scan point.

3.1.51 OETorsionScanner

```
class OETorsionScanner
```

This class provides an interface to perform potential energy scanning for a selected torsion in the input molecule. Torsion scan is preformed by a series of constrained optimizations in which all internal degrees of freedom but selected torsion, are optimized.

See also

- *OETorsionScan* method
- *OETorsionScanOptions* class
- *OETorsionScanResult* class

The *OETorsionScanner* class defines the following public methods:

- *GetLowestEnergyConfs*
- *GetLowestEneResults*
- *GetScannedAngles*
- *GetSinglePointConfs*
- *Scan*

Constructor

```
OETorsionScanner(const OETorsionScanOptions& opts = OETorsionScanOptions())
OETorsionScanner(const OETorsionScanner&)
```

Default and copy constructors.

Assignment operator.

3.1.52 GetLowestEnergyConfs

```
bool GetLowestEnergyConfs(OEChem::OEMCMolBase& dst) const
```

Returns the optimized lowest energy conformers corresponding to each of the angles that scanning was performed. The *Scan* must be performed before calling this method.

3.1.53 GetLowestEneResults

```
OESystem::OEIterBase<OETorsionScanResult>* GetLowestEneResults() const
```

Returns the lowest energy scan results corresponding to each of the angles that scanning was performed. The *Scan* must be performed before calling this method.

3.1.54 GetScannedAngles

```
std::vector<double> GetScannedAngles() const
```

Returns all the angles at which scanning was performed. The *Scan* must be performed before calling this method.

3.1.55 GetSinglePointConfs

```
bool GetSinglePointConfs(OEChem::OEMCMolBase& mol,
                         const double angle) const
```

Returns all the optimized conformers corresponding to the specified angle. The *Scan* must be performed before calling this method.

3.1.56 Scan

```
unsigned Scan(const OEChem::OEMCMolBase&, const OEChem::OETorsion&)
```

Perform torsion scanning for the specified torsion on the input molecule.

3.2 OESz Constants

3.2.1 OEEntropyMethod

This namespace contains constants describing the type of Hessian used for vibrational entropy calculations.

QNewton

Hessian matrix is obtained from Quasi-Newton optimization.

Analytic

Hessian matrix is calculated analytically.

ExternalHessian

Hessian matrix is imported instead of being calculated.

3.2.2 OEEnvType

This namespace contains constants describing the type of environment for entropy calculations to be used.

Gas

Gas-phase molecule.

SolutionSPT

Molecule in solution for which hydrophobic solvation is described with the Scaled Particle Theory (SPT).

SolutionSA

Molecule in solution for which hydrophobic solvation is described with the Surface Area (SA) model

Protein

Molecule in the protein binding-site.

Solution

Same as *oe:constant::SolutionSPT*

3.2.3 OEForceFieldType

This namespace contains constants describing the force field type

MMFF94

Standard MMFF94 force field. (String value: mmff94)

MMFF94S

A version of MMFF94 with modified out-of-plane and torsional parameters. This force-field describes time-averaged structures mainly for trigonal nitrogen centers. (String value: mmff94s)

MMFF_AMBER

A combined force field which can be used for the optimization of the ligand in the rigid protein only. Can't be applied for free ligand. In this potential ligand internal degrees of freedom are handled by the MMFF94 force field, while protein-ligand interactions (vdW and Coulomb) are described by Amber force field.

MMFFS_AMBER

As above, but with MMFF94S forcefield applied for ligand intramolecular interactions.

MMFF_IEFF

A combined force field which can be used for the optimization of the ligand inside a protein receptor or for the optimization of a cluster of molecules. In this force field ligand internal molecular degrees of freedom are handled by the MMFF94, while all intermolecular interactions are calculated with the IEFF potential described in [Hamaguchi-2012].

MMFFS_IEFF

As above, but with the MMFF94S forcefield applied for all intramolecular interactions.

SMIRNOFF99FROSST

SMIRNOFF99FROSST force field.

PARSLEY_OPENFF100

Parsley OpenFF1.0.0 force field.

PARSLEY_OPENFF111

Parsley OpenFF1.1.1 force field.

PARSLEY_OPENFF121

Parsley OpenFF1.2.1 force field.

PARSLEY_OPENFF131

Parsley OpenFF1.3.1 force field.

PARSLEY_OPENFF

The latest version of Parsley OpenFF force field (String value: parsley)

SAGE_OPENFF200

Sage OpenFF2.0.0 force field.

SAGE_OPENFF210

Sage OpenFF2.1.0 force field.

SAGE_OPENFF

The latest version of Sage OpenFF force field (String value: sage)

FF14SB_SMIRNOFF

Combined Amber/ff14SB - PARSLEY_OPENFF or (Amber/ff14SB - SAGE_OPENFF) force field which might be used to optimize a part of protein including cofactor. Protein part will be described with Amber/ff14SB parameters while cofactor with PARSLEY_OPENFF131 or SAGE_OPENFF200.

> **Note**
>
> Using SMIRNOFF99FROSST, PARSLEY_OPENFF100, PARSLEY_OPENFF111, PARSLEY_OPENFF121, PARSLEY_OPENFF131, SAGE_OPENFF200 and FF14SB_SMIRNOFF force fields requires that the input molecule(s) have assigned atomic partial charges. For small molecules we recommend using AMABCC-ELF10 partial charges while for proteins we suggest using AMBER partial charges.

3.2.4 OEOptType

This namespace contains constants describing the type of optimization method to be used.

BFGS

This flag sets the Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimizer. It is the default optimizer in Szybki.

CG

This flag selects the conjugate gradient optimizer instead of the default *OEOptType.BFGS*.

SD

This flag selects the steepest descent optimizer.

SD_BFGS

This flag selects 5 steps of *OEOptType.SD* pre-optimization followed by complete BFGS optimization.

SD_CG

This flag selects 5 steps of *OEOptType.SD* pre-optimization followed by complete CG optimization.

NEWTON

This flag selects Newton optimizer. It includes 1-20 steps of *OEOptType.SD* preconditioning depending on initial gradients and the size of the input molecule.

3.2.5 OEPotentialTerms

This namespace contains constants describing potential terms which might be calculated in different Szybki runs. Entries: *OEPotentialTerms.VdW*, *OEPotentialTerms.Coulomb*, *OEPotentialTerms.Bond*, *OEPotentialTerms.Bend*, *OEPotentialTerms.StretchBend*, *OEPotentialTerms.Torsion* and *OEPotentialTerms.ImproperTorsion* correspond to the total MMFF94 or Smirnoff potential terms. In the case of partial optimization of

the protein, entries with a prefix `Ligand` refer to the force field terms of the ligand (for example *OEPotentialTerms.LigandVdW*), while entries with a prefix `Protein` (for example *OEPotentialTerms.ProteinVdW*) are for the MMFF94 or Smirnoff terms of the flexible part of the protein. Force field terms describing VdW and Coulomb interactions between the ligand and the flexible part of the protein are defined with the constants *OEPotentialTerms.ProteinLigandVdW* and *OEPotentialTerms.ProteinLigandCoulomb* respectively. Term *OEPotentialTerms.ProteinLigandInteraction* describes complete ligand-protein interaction according to the protein electrostatic model chosen. Values defined by *OEPotentialTerms.SheffieldSolvation*, *OEPotentialTerms.PBSolvation* and *OEPotentialTerms.CavitySolvation* refer to solvation energies used to optimize a ligand in solution, while the term *OEPotentialTerms.HarmonicConstraint* determines the he value of the imposed harmonic constraint. Entries: *OEPotentialTerms.VdWProteinLigand*, *OEPotentialTerms.CoulombProteinLigand*, *OEPotentialTerms.ProteinDesolvation*, *OEPotentialTerms.LigandDesolvation*, *OEPotentialTerms.SolventScreening*, *OEPotentialTerms.GridCoulomb* and *OEPotentialTerms.ExactCoulomb* are used to break down the total protein-ligand interaction energy into components. Value of the *OEPotentialTerms.ProteinPseudoLigandInteraction* represents the interaction of the flexible part of of the optimized protein-ligand system (ligand + flexible part of the protein) with the rest of the protein and as such is not an observable quantity. When Amber force field is applied for protein-ligand interaction, values of *OEPotentialTerms.ProteinLigandAmberVdW* and *OEPotentialTerms.ProteinLigandAmberCoulomb* are used for the corresponding vdW and Coulomb protein-ligand energies. Term *OEPotentialTerms.TorsionHarmonicConstraint* is the value of the constraint torsion harmonic potential. Two terms: *OEPotentialTerms.IEFFInteraction* and *OEPotentialTerms.InterLigandIEFF* are IEFF energies, total and interligand respectively. The latter has nonzero value when more than one ligand are present in the active site of the protein.

VdW

Value of vdW energy

Coulomb

Value of Coulomb energy

Bond

Value of bond stretch energy

Bend

Value of angle bend energy

StretchBend

Value of stretch-bend energy. This potential term appears only when MMFF force field is used

Torsion

Value of torsion energy

ImproperTorsion

Value of improper torsion energy

LigandVdW

Value of ligand vdW energy

LigandCoulomb

Value of ligand Coulomb energy

LigandBond

Value of ligand bond energy

LigandBend

Value of ligand angle bend energy

LigandStretchBend

Value of ligand stretch-bend energy. This potential term appears only when MMFF force field is used

LigandTorsion

Value of ligand torsion energy

LigandImproperTorsion

Value of ligand improper torsion energy

ProteinVdW

Value of protein vdW energy

ProteinCoulomb

Value of protein Coulomb energy

ProteinBond

Value of protein bond energy

ProteinBend

Value of protein angle bend energy

ProteinStretchBend

Value of protein stretch-bend energy. This potential term appears only when MMFF force field is used

ProteinTorsion

Value of protein torsion energy

ProteinImproperTorsion

Value of protein improper torsion energy

ProteinLigandVdW

Value of protein-ligand vdW energy

ProteinLigandCoulomb

Value of protein-ligand Coulomb energy

ProteinLigandInteraction

Value of total protein-ligand interaction

SheffieldSolvation

Value of Sheffield solvation energy

HarmonicConstraint

Value of atom harmonic constraint energy

PBSolvation

Value of PB solvation energy

CavitySolvation

Value of cavity solvation energy

VdWProteinLigand

Value of protein-ligand vdW interaction energy

CoulombProteinLigand

Value of protein-ligand Coulomb interaction energy

ProteinDesolvation

Value of protein desolvation energy

LigandDesolvation

Value of ligand desolvation energy

SolventScreening

Value of PB solvent screening energy

GridCoulomb

Value of protein-ligand Coulomb energy on grid

ExactCoulomb

Value of exact protein-ligand Coulomb energy

ProteinPseudoLigandInteraction

Value of ligand and flexible protein (pseudoligand) interaction with the rest of the protein energy

ProteinLigandAmberVdW

Value of Amber protein-ligand vdW energy

ProteinLigandAmberCoulomb

Value of Amber protein-ligand Coulomb energy

TorsionHarmonicConstraint

Value of torsion harmonic constraint energy

IEFFInteraction

Value of total IEFF intermolecular energy

InterLigandIEFF

Value of interligand IEFF energy

Max

Number of potential terms

3.2.6 OEProteinElectrostatics

This namespace contains constants describing handling of protein-ligand electrostatics.

NoElectrostatics

No protein-ligand electrostatic interaction included.

GridPB

Protein electrostatic potential as a solution of PB equation is stored on a grid and used to calculate its electrostatic interaction with any ligand placed inside the protein.

GridCoulomb

Protein Coulomb electrostatic potential is stored on a grid and used to calculate its electrostatic interaction with any ligand placed inside the protein.

ExactCoulomb

Exact Coulombic potential used to calculate protein-ligand electrostatic interaction.

SolventPBForces

Complete PB protein-ligand interaction in solvent is used. Electrostatic protein-ligand interaction is a sum of Coulomb, solvent screening, protein desolvation and ligand desolvation terms.

3.2.7 OERunType

This namespace contains constants describing the type of optimization run.

SinglePoint

Indicates a single-point geometry run.

TorsionsOpt

Optimization in torsional coordinates. In the case of a ligand in the protein active site, translational and rotational degrees of freedom are included in the optimization. Terms `torsional coordinates` defines a set of rotors in the molecules. Numerical values of `torsional coordinates` are dihedral angles defined by Cartesian coordinates of 4 bonded atoms `a-b-c-d` where the middle bond between atoms b and c is a rotatable bond. Please look at the OEChem documentation for the definition of `rotatable bond`. Bonds a-b and c-d are selected arbitrarily but uniquely, so a single dihedral angle is assigned to each torsional coordinate. `Torsional coordinates` are a subset of `torsional space`, however usually the latter term is used synonymously with the former.

CartesiansOpt

Optimization in atomic Cartesian coordinates.

SolidBodyOpt

This flag applies only for the optimization of a ligand inside the protein. Both translational and rotational degrees of freedom are optimized. Term `solid body optimization` means that only 6 degrees of freedom are allowed to change during optimization. Those are 3 translational and 3 rotational degrees of freedom. In our implementation 3 rotational degrees of freedom are described by 4 quaternions. Please consult any molecular physics or mechanics textbook for the definition of terms `translational-rotational degrees of freedom` and `quaternions`.

3.2.8 OESolventModel

This namespace contains constants describing solvation model of a molecule in solution

NoSolv

No solvation. Gas-phase molecule.

Sheffield

Simplified analytical electrostatic solvation model used.

PB

Poisson–Boltzmann electrostatic solvation model used.

Cavity

Surface Area hydrophobic solvation model used. It might be used in combination with electrostatic PB or Sheffield solvation models.

3.2.9 OEAtomicRadii

This namespace contains constants describing atomic radii used for PB calculations

Bondi

Default set.

ZAP9

Special set of atomic radii described in ([Nicholls-2010]).

ZAP7

Special set of atomic radii described in ([Nicholls-2010]).

3.2.10 OEProtFlex

This namespace contains constants describing partial flexibility of protein receptor which are included during bound ligand optimization.

NoFlex

No protein-flexibility.

PolarH

Polar hydrogens (for example hydroxyl hydrogens in serine) within specified distance from a ligand will be optimized.

SideChains

Residue side chains within specified distance from a bound ligand will be optimized.

Residues

Complete residues within specified distance from a bound ligand will be optimized.

ResidueList

Complete residues specified by the user will be optimized.

SideChainsList

Residue side chains specified by the user will be optimized.

PolarHResList

Polar hydrogens of residues specified by the user will be optimized.

3.2.11 OEFreeFormIonicState

This namespace contains constants describing the charge state of a molecule for solvation free energy estimation or conformations free energies in solution.

Uncharged

Neutral molecule.

PH74

Charge state corresponding to physiological pH = 7.4.

Input

Charge state is determined in the molecular input file.

Default

Same as PH74 above.

3.2.12 OEFreeFormReturnCode

This namespace contains constants describing the status of the FreeForm runs.

Success

Normal execution.

AtomTypeFailure

Failed to assign MMFF94 atom types.

ImplicitHydrogens

Input molecule contains implicit hydrogens.

InputGeometry

3D structure of the input molecule is badly defined.

No3DCoordinates

Input molecule has no 3D coordinates.

ConfGenFailure

Failed to generate conformations.

TooManyRotors

Input molecule contains too many rotors.

ChargeMismatch

Total formal charge of the molecule is different than the sum of partial charges.

NotSupportedCharges

Attempt to use not supported by FreeForm partial charges type.

UnrecognizedChargeState

Wrong charge state of the molecule.

UnsupportedRadiiType

Radii type not supported by FreeForm.

ChargeAssignFailure

Failed to assign atomic partial charges.

GraphMismatch

Cannot use molecules with dissimilar graphs.

UnequalNumConfs

Number of conformers in two MCMols are different.

PartnerConfNotFound

A corresponding conformer does not exist.

MissingPartnerConf

Partner conformer information does not exist.

MissingFFConfEne

FreeFormConf free energies do not exist.

3.2.13 OEFreeFormWarning

This namespace contains constants describing warnings produced by FreeForm runs.

UnspecifiedStereo

Input molecule contains chiral atoms with unspecified stereo configuration.

NotRemovableCharge

Input molecule can't be neutralized.

HSamplingSkipped

Hydrogen atom sampling during conformation generation was not performed.

Missing3DCoordinates

Input ensemble can't be used because input molecular file has no 3D coordinates.

ZeroPartialCharges

Input molecule does not contain any partial charges.

3.3 OESz Functions

3.3.1 OEEstimateConfFreeEnergies

```
bool OEEstimateConfFreeEnergies(OEFreeFormConfResults& results,
                                OEChem::OEMCMolBase& moldst,
                                const OEChem::OEMCMolBase& molsrc,
                                const OEFreeFormConfOptions& opts)
```

> **Warning**
>
> This function should not be used on 32-bit platforms because the memory requirements are too high.

Estimates the Gibbs free energy to convert an ensemble of solution conformations into a state where only a single specific conformation is present as described in the theory section. Function returns `true` if the calculation was successful, `false` otherwise. Results are accessible from the `results` object (see *OEFreeFormConfResults*). The ensemble of unique conformations is returned as a multiconformer molecule `moldst`. The third argument represents the input molecule, and the last one contains user selected options (see *OEFreeFormConfOptions*).

3.3.2 OEEstimateSolvFreeEnergy

```
bool OEEstimateSolvFreeEnergy(OEFreeFormSolvResults& results,
                              OEChem::OEMolBase& dst,
                              const OEChem::OEMCMolBase& src,
                              const OEFreeFormSolvOptions& opts)
```

Estimates the Gibbs energy of solvation, according to the algorithm described in the theory section. Function returns `true` if the calculation was successful, `false` otherwise. Results are accessible from the `results` object (see *OEFreeFormSolvResults*). The second parameter returns the molecule representing the structure selected by the algorithm for solvation calculation. Third and fourth arguments are the input molecule and an object containing user options (see *OEFreeFormSolvOptions*) respectively.

3.3.3 OEGetEnergyTermName

```
const char* OEGetEnergyTermName(unsigned int)
```

3.3.4 OEGetFreeFormError

```
std::string OEGetFreeFormError(const unsigned int)
```

Returns error messages corresponding to return codes defined in *OEFreeFormReturnCode* namespace.

3.3.5 OEGetSzybkiError

```
std::string OEGetSzybkiError(const unsigned)
```

Returns error messages corresponding to return codes defined in *OESzybkiReturnCode* namespace.

3.3.6 OESzybkiGetArch

```
const char* OESzybkiGetArch()
```

3.3.7 OESzybkiGetLicensee

```
bool OESzybkiGetLicensee(std::string& licensee)
```

3.3.8 OESzybkiGetPlatform

```
const char* OESzybkiGetPlatform()
```

3.3.9 OESzybkiGetRelease

```
const char* OESzybkiGetRelease()
```

3.3.10 OESzybkiGetSite

```
bool OESzybkiGetSite(std::string& site)
```

3.3.11 OESzybkiGetVersion

```
unsigned int OESzybkiGetVersion()
```

3.3.12 OETorsionScan

```
OESystem::OEIterBase<OETorsionScanResult>* OETorsionScan(OEChem::OEMCMolBase& dst,
                                                         const OEChem::OEMCMolBase& src,
                                                         const OEChem::OETorsion&␣
↪torsion,
                                                         const OETorsionScanOptions&␣
↪opts)
```

Performs potential energy scan for a selected torsion in the input molecule. Torsion scan is preformed by a series of constrained Newton optimizations in which all internal degrees of freedom but selected torsion, are optimized. Returns an iterator to the *OETorsionScanResult* objects from which the user can obtain actual values of the torsion angle within $[0.0°, 360.0°]$ range and the corresponding relative energies.

dst
: The output multi-conformation molecule that stores conformations corresponding to the points on the scan.

src
: The input molecule of which torsion is scanned.

torsion
: The torsion of the input molecule that is being scanned.

opts
: The option that stores parameters for the scan such as force field, environment (vacuum or solution), and resolution (*OETorsionScanOptions*).

Note

Using multiple conformations for the input molecule (*src*) is not a requirement as the function will work for a single conformation. However using as many as possible conformations is recommended because it increases the probability of finding the minimum energy path for a given torsion. The indication that the torsion profile is not the minimum energy path is the occurrence of sharp drop(s) of potential energy, or/and different energy values obtained for 0.0 and 360.0 degrees. Increasing the number of the input geometries in such a situation might remove those symptoms.

Note

As the function is identifying the minimum energy path the output conformations can come from different input conformations this means the SD data from the input conformations is not passed through to the output molecule.

See also

- *OETorsionScanOptions* class
- *OETorsionScanResult* class

Example to scan all torsions of a molecule:

```
OEMol outmol = new OEMol();

OETorsionScanOptions opts = new OETorsionScanOptions();
opts.SetDelta(30.0);
opts.SetForceFieldType(OEForceFieldType.MMFF94);
opts.SetSolvationType(OESolventModel.NoSolv);

foreach (OETorsion tor in OEChem.OEGetTorsions(mol))
{
    Console.WriteLine("Torsion: {0} {1} {2} {3}",
                      tor.a.GetIdx(), tor.b.GetIdx(),
                      tor.c.GetIdx(), tor.d.GetIdx());

    foreach (OETorsionScanResult res in OESzybkilib.OETorsionScan(outmol, mol, tor,␣
↪opts))
        Console.WriteLine("{0:0.00} {1:0.00}", res.GetAngle(), res.GetEnergy());

}
```

3.3.13 OEBoundLigandEntropy

```
unsigned OEBoundLigandEntropy(OEEntropyResults& res,
                              const OEChem::OEMolBase& protein,
                              const OEChem::OEMCMolBase& poses,
                              const OEBoundEntropyOptions& opts =␣
↪OEBoundEntropyOptions())
unsigned OEBoundLigandEntropy(OEEntropyResults& res,
```

(continues on next page)

(continued from previous page)

```
                                const OEBio::OEDesignUnit& du,
                                const unsigned int proteinMask,
                                const OEChem::OEMCMolBase& poses,
                                const OEBoundEntropyOptions& opts =␣
↪OEBoundEntropyOptions())
```

Estimates entropy of a ligand bound in the rigid protein receptor. Returns error messages corresponding to the return codes defined in *OESzybkiReturnCode* namespace.

res
Instance or the results class from which numerical entropy values can be retrieved. *OEEntropyResults*.

protein
Protein receptor object.

du
In the second overloaded function, instead of a protein molecule, a design unit object is used. The advantage of this alternative method is a possibility of including cofactors, solvent and other molecules which are present in the

poses
Ensemble of poses for which ligand bound entropy will be estimated.

opts
The option that stores parameters for entropy estimation, in particular force field and charging method used to assigned partial charges to a ligand. *OEBoundEntropyOptions*.

3.3.14 OELigandEntropy

```
unsigned OELigandEntropy(OEEntropyResults& res,
                         const OEChem::OEMCMolBase& ensemble,
                         const OELigandEntropyOptions& opts = OELigandEntropyOptions())
```

Estimates entropy of a free ligand in solution or in vacuum. Returns error messages corresponding to the return codes defined in *OESzybkiReturnCode* namespace.

res
Instance or the results class from which numerical entropy values can be retrieved. *OEEntropyResults*.

ensemble
Ensemble of conformations for which free ligand entropy will be estimated

opts
The option that stores parameters for entropy estimation, in particular force field and charging method used to assigned partial charges to a ligand. *OELigandEntropyOptions*.

See also

- *OEEntropyResults* class
- *OEBoundEntropyOptions* class
- *OELigandEntropyOptions* class

CHAPTER

FOUR

PRELIMINARY API

4.1 Preliminary OESz Classes

4.1.1 OELigandEnergy

> **Attention**
>
> This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

```
class OELigandEnergy
```

This class provides a container for the total energy and energy components resulting from a FF energy calculation on a ligand.

> **See also**
>
> - *OELigandOptResults* class

The *OELigandEnergy* class defines the following public methods:

- *GetBondEne*
- *GetBendEne*
- *GetTorsionEne*
- *GetImproperEne*
- *GetVdWEne*
- *GetCoulombEne*
- *GetIntraEne*
- *GetSolvationEne*
- *GetTotalEne*

Constructor

```
OELigandEnergy()
OELigandEnergy(const OELigandEnergy&)
```

Default and copy constructors.

Assignment operator.

GetBondEne

```
double GetBondEne() const
```

Returns the bond energy component of the total energy

GetBendEne

```
double GetBendEne() const
```

Returns the angle bending energy component of the total energy

GetTorsionEne

```
double GetTorsionEne() const
```

Returns the proper torsional energy component of the total energy

GetImproperEne

```
double GetImproperEne() const
```

Returns the improper torsional energy component of the total energy

GetVdWEne

```
double GetVdWEne() const
```

Returns the van der Walls interaction energy component of the total energy

GetCoulombEne

```
double GetCoulombEne() const
```

Returns the coulomb energy component of the total energy

GetIntraEne

```
double GetIntraEne() const
```

Returns the intra energy component of the total energy

GetSolvationEne

```
double GetSolvationEne() const
```

Returns the solvation energy component of the total energy

GetTotalEne

```
double GetTotalEne() const
```

Returns the total energy

4.1.2 OELigandOptOptions

> **Attention**
>
> This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

```
class OELigandOptOptions
```

This class provides an interface to modify the options for FF optimization of a ligand molecule.

> **See also**
>
> - *OELigandOptResults* class

The *OELigandOptOptions* class defines the following public methods:

- *GetForceField*
- *GetForceFieldType*
- *GetLigandCharge*
- *GetLigandChargeType*
- *GetNonBondCutoff*
- *GetGradTolerance*
- *GetMaxIter*
- *GetSolventModel*
- *SetForceField*
- *SetLigandCharge*
- *SetNonBondCutoff*
- *SetGradTolerance*
- *SetMaxIter*
- *SetSolventModel*

Constructor

```
OELigandOptOptions()
OELigandOptOptions(const OELigandOptOptions&)
```

Default and copy constructors.

Assignment operator.

4.1.3 GetForceField

```
const OEMolPotential::OEForceField* GetForceField() const
```

Returns pointer to the force field used in the optimization. ** Default ** *SAGE_OPENFF*.

4.1.4 GetForceFieldType

```
unsigned int GetForceFieldType() const
```

Returns the force field type used for the calculations, as *unsigned int* from the *OEForceFieldType* namespace. ** Default ** *SAGE_OPENFF*.

4.1.5 GetLigandCharge

```
const OEProton::OEChargeEngineBase* GetLigandCharge() const
```

Returns the charge engine used for the calculations. See *SetLigandCharge* method. ** Default ** OEChargeEngineNoOp.

4.1.6 GetLigandChargeType

```
unsigned int GetLigandChargeType() const
```

Returns the charge type to be used for the calculations, as *unsigned int* from the *OELigandChargeType*.
Default: *OELigandChargeType.CURRENT*

4.1.7 GetNonBondCutoff

```
double GetNonBondCutoff() const
```

See *SetNonBondCutoff* method

4.1.8 GetGradTolerance

```
double GetGradTolerance() const
```

See *SetGradTolerance* method

4.1.9 GetMaxIter

```
unsigned GetMaxIter() const
```

See *SetMaxIter* method

4.1.10 GetSolventModel

```
unsigned GetSolventModel() const
```

See *SetSolventModel* method

4.1.11 SetForceField

```
bool SetForceField(const unsigned)
bool SetForceField(const std::string&)
bool SetForceField(const OEMolPotential::OEForceField&)
```

Sets force field to be used. Method takes either an *unsigned* from the *OEForceFieldType* namespace, or the takes corresponding string value, or a custom forcefield. Available predefined values are: *MMFF94*, *MMFF94S*, *PARSLEY_OPENFF*, and *SAGE_OPENFF*. Default: *sage*

4.1.12 SetLigandCharge

```
bool SetLigandCharge(const unsigned)
bool SetLigandCharge(const std::string&)
bool SetLigandCharge(const OEProton::OEChargeEngineBase&)
```

Selects the method for ligand partial charges assigment. Method takes either an *unsigned* from the *OELigandChargeType* namespace, or the takes corresponding string value, or a user-defined instance of OEChargeEngineBase. It returns `false` when the invalid selection is chosen, `true` otherwise. The default engine OEChargeEngineNoOp will use the pre-existing charges.

4.1.13 SetNonBondCutoff

```
bool SetNonBondCutoff(const double)
```

Sets the **Non-bonded interactions cutoff** for optimization. Setting a value of `0.0` or less assumes that all interactions should be included without any cutoff. Default: 0.0.

4.1.14 SetGradTolerance

```
bool SetGradTolerance(const double)
```

Sets the **gradient tolerance** for optimization. Default: 1.e-4.

4.1.15 SetMaxIter

```
bool SetMaxIter(const unsigned)
```

Sets the number of maximum iterations for optimization Default: 400

4.1.16 SetSolventModel

```
bool SetSolventModel(const unsigned)
```

Sets the type of solvent model used for the calculations of solution ligands, as an *unsigned int* from the *OESolventModel* namespace. Method returns `false` when the invalid selection is chosen, `true` otherwise. Default is *OESolventModel.NoSolv*

4.1.17 OELigandOptResults

Attention

This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

```
class OELigandOptResults
```

This class provides a container for the results of a FF optimization of a ligand molecule.

See also

- *OELigandEnergy* class

The *OELigandOptResults* class defines the following public methods:

- *GetInitialEnergies*
- *GetFinalEnergies*
- *GetRMSD*
- *GetReturnCode*

Constructor

```
OELigandOptResults()
OELigandOptResults(const OELigandOptResults&)
```

Default and copy constructors.

Assignment operator.

GetInitialEnergies

```
OELigandEnergy GetInitialEnergies() const
```

Returns an object of the *OELigandEnergy* class corresponding to the FF energy at the initial (input) geometry.

GetFinalEnergies

```
OELigandEnergy GetFinalEnergies() const
```

Returns an object of the *OELigandEnergy* class corresponding to the FF energy at the final (optimized) geometry.

GetRMSD

```
double GetRMSD() const
```

Returns the RMSD between the initial and final (input and optimized) geometries

GetReturnCode

```
unsigned GetReturnCode() const
```

Returns one of the values defined in the *OESzybkiReturnCode* namespace.

4.1.18 OETorConstraintOptOptions

> **Attention**
>
> This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

```
class OETorConstraintOptOptions
```

This class provides an interface to modify the options for an all-torsion-constrained optimization of a ligand molecule.

> **See also**
>
> - *OELigandOptOptions* class

The *OETorConstraintOptOptions* class inherits the following methods from the *OELigandOptOptions* class.

- *GetForceField*
- *GetForceFieldType*
- *GetLigandCharge*
- *GetLigandChargeType*
- *GetNonBondCutoff*
- *GetGradTolerance*
- *GetMaxIter*
- *GetSolventModel*
- *SetForceField*
- *SetLigandCharge*
- *SetNonBondCutoff*
- *SetGradTolerance*
- *SetMaxIter*
- *SetSolventModel*

The *OETorConstraintOptOptions* class defines the following public methods:

- *GetPenaltyForceConstant*
- *GetUseInternalCoord*
- *SetPenaltyForceConstant*
- *SetUseInternalCoord*

Constructor

```
OETorConstraintOptOptions()
OETorConstraintOptOptions(const OETorConstraintOptOptions&)
```

Default and copy constructors.

Assignment operator.

4.1.19 GetPenaltyForceConstant

```
double GetPenaltyForceConstant() const
```

See *SetPenaltyForceConstant* method.

4.1.20 GetUseInternalCoord

```
bool GetUseInternalCoord() const
```

See *SetUseInternalCoord* method.

4.1.21 SetPenaltyForceConstant

```
bool SetPenaltyForceConstant(double pfc)
```

Overwrites the default value (`1e4`) for force constant in dihedral angle constraint used for optimization. Method returns `false` if invalid value of force constant is passed.

4.1.22 SetUseInternalCoord

```
bool SetUseInternalCoord(bool useInt)
```

When the parameter `useInt` is set to `true`, optimization is performed in internal coordinates. Overwrites a default method for torsion constraint, which applies a large enough penalty force to keep the value of the dihedral angle constant during optimization.

4.1.23 OETorConstraintOptimizer

Attention

This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

```
class OETorConstraintOptimizer
```

This class provides an interface to perform an all-torsion-constrained optimization of a ligand molecule.

See also

- *OELigandOptResults* class

The *OETorConstraintOptimizer* class defines the following public methods:

- *Setup*
- *Optimize*

Constructor

```
OETorConstraintOptimizer(const OETorConstraintOptOptions& opts =␣
↪OETorConstraintOptOptions())
OETorConstraintOptimizer(const OETorConstraintOptimizer&)
```

Default and copy constructors.

Assignment operator.

4.1.24 Setup

```
unsigned Setup(OEChem::OEMolBase& mol)
```

Method to setup molecule for an all-torsions-constrained optimization

4.1.25 Optimize

```
double Optimize(OEChem::OEMolBase& mol,
                const OEChem::OETorsion& torsion) const
double Optimize(OEChem::OEMolBase& mol) const
unsigned Optimize(OELigandOptResults& res, OEChem::OEMolBase& mol,
                  const OEChem::OETorsion& torsion)
unsigned Optimize(OELigandOptResults& res, OEChem::OEMolBase& mol)
```

Optimize the ligand. If an instance ref:*OELigandOptResults <OESz_OELigandOptResults* class is passed, it is populated with the results and a value from the *OESzybkiReturnCode* namespace reporting the success or failure of the calculation is returned. Is not, the final optimized energy is returned.

4.2 Preliminary OESz Constants

4.2.1 OEComplexFFType

Attention

This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

This namespace contains constants describing the force field type

MMFF94

Standard MMFF94 force field. Corresponding string value: mmff94

MMFF94S

A version of MMFF94 with modified out-of-plane and torsional parameters. This force-field describes time-averaged structures mainly for trigonal nitrogen centers. Corresponding string value: mmff94s

MMFF_AMBER

A force field that combines MMFF and Amber force fields. In this potential both the protein and ligand internal degrees of freedom are handled by the MMFF94 force field, while protein-ligand interactions are described by Amber force field. Corresponding string value: mmff_amber

MMFFS_AMBER

As above, but with MMFF94S forcefield applied for intramolecular interactions. Corresponding string value: mmffs_amber

FF14SB_PARSLEY

A force field that combines FF14SB and Parsley force fields. In this potential protein internal degrees of freedom are handled by the FF14SB force field and ligand internal degrees of freedom are handled by the Parsley force field. The inter-molecular protein-ligand interactions are described by use of appropriate mixing rules for the individual atom interaction potentials. Corresponding string value: ff14sb_parsley

FF14SB_SAGE

A force field that combines FF14SB and Sage force fields. In this potential protein internal degrees of freedom are handled by the FF14SB force field and ligand internal degrees of freedom are handled by the Sage force field. The inter-molecular protein-ligand interactions are described by use of appropriate mixing rules for the individual atom interaction potentials. Corresponding string value: ff14sb_sage

4.2.2 OELigandChargeType

> **Attention**
>
> This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

This namespace contains constants describing the ligand charge types associated with the *OELigandChargeOptions.SetLigandCharge* method.

CURRENT

Use the currently existing charges on the ligand. Corresponding string value: current

AM1BCC

Use the `OEAM1BCCCharges` charges. Corresponding string value: am1bcc

AM1BCCELF10

Use the `OEAM1BCCELF10Charges` charges. Corresponding string value: am1bccelf10

MMFF

Use the `OEMMFF94Charges` charges. Corresponding string value: mmff

4.2.3 OEOptimizationType

> **Attention**
>
> This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

This namespace contains constants describing the force field type

Cartesian

Optimization in atomic Cartesian coordinates

PoseCartesian

Constrained, to preserve the pose, optimization in atomic Cartesian coordinates. Optimization is performed in multiple steps, with additional harmonic constraint potentials added to restrict the motion of the heavy atoms of the ligand in the preliminary steps of optimization, followed by regular atomic cartesian coordinates optimization.

4.2.4 OESzybkiReturnCode

> **Attention**
>
> This is a preliminary API and may be improved based on user feedback. It is currently available in C++ and Python.

This namespace contains constants describing the force field type

Success

Success.

ExcessiveOverlap

Failed due to excessive overlap.

AromaticRingDev

Failed due to excessive deviation from planarity for aromatic rings. Excessive deviation is defined if average deviation exceeds 5 degrees or if certain torsion exceeds a maximum of 10 degrees

IncompleteOpt

Optimization did not reach desired gradient tolerance, and finished due to exceeding maximum iterations.

OptimizerFailed

Optimizer failed to find solution.

FFNeedHessiens

Requires a forcefield with second derivative implementation.

InvalidOptions

Invalid Options.

ContainsImplicitH

Contains Implicit Hydrogens.

ProteinFailedFF

Failed due to inability to assign protein force field.

ProteinMissing3D

Failed due to input protein not having 3D coordinates.

ProteinMissingCharge

Failed due to input protein not having charges assigned.

ProteinNotSet

Failed due to input protein missing.

LigandFailedFF

Failed due to inability to assign ligand force field.

LigandMissing3D

Failed due to input ligand not having 3D coordinates.

LigandMissingCharge

Failed due to input ligand not having charges assigned.

LigandFailedCharge

Failed due to inability to assign ligand charges.

Failed

Failed.

CHAPTER

FIVE

EXAMPLES: WORKING WITH SZYBKI TK

The basic *Szybki TK* API is provided to the users in the *OESzybki*, *OESzybkiOptions*, *OESzybkiResults* and *OESzybkiEnsembleResults* classes. In addition, when the purpose is to calculate compound properties, such as solvation free energy or free energy of selecting specific conformations out of the ensemble, the higher level API are defined in the free functions, `OEEstimateSolvFreeEnergy` and `OEEstimateConfFreeEnergies`.

5.1 Ligand Energetics and Optimization

5.1.1 Single ligand in vacuum

The following example illustrates how to optimize a single ligand in vacuum. As one can see assuming a molecule is successfully read, only two objects are needed to perform the optimization: *OESzybki* and *OESzybkiResults*. The latter is passed as the second parameter to the parenthesis operator `OESzybki.operator()`. Molecule with the optimized coordinates is returned as a first parameter. Optimization is done using the default MMFF94. Final energy results are available as SD tags in the returned molecule and optionally with a call of a method `OESzybkiResults.Print`.

Listing 1: Simple Ligand in a Vacuum

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;
```

(continues on next page)

(continued from previous page)

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

        oemolistream ifs = new oemolistream();
        oemolostream ofs = new oemolostream();

        if (!ifs.open(args[0]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[0] + " for reading");
        }
        if (!ofs.open(args[1]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[1] + " for writing");
        }

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);

        OESzybkiOptions opts = new OESzybkiOptions();
        OESzybki sz = new OESzybki(opts);
        OESzybkiResults res = new OESzybkiResults();

        if (!sz.Call(mol, res))
        {
            return 1;
        }

        OEChem.OEWriteMolecule(ofs, mol);
        res.Print(OEChem.oeout);

        return 0;
    }
}
```

5.1.2 Single ligand in vacuum using SMIRNOFF

The following example illustrates how to optimize a single ligand in vacuum using SMIRNOFF (see SMIRNOFF). Only two objects are needed to perform the optimization: *OESzybki* and *OESzybkiResults*. The latter is passed as the second parameter to the parenthesis operator *OESzybki.operator()*. Partial charges have to be preassigned to the input molecule. Molecule with the optimized coordinates is returned as a first parameter. Final energy results are available as SD tags in the returned molecule and optionally with a call of a method *OESzybkiResults.Print*.

Listing 2: Simple Ligand in a Vacuum Using SMIRNOFF

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class SimpleSmirnoff
{
    public static int Main(String[] args)
    {
        if (args.Length!=2)
        {
            OEChem.OEThrow.Usage("SimpleSmirnoff <molfile> <outfile>");
        }

        oemolistream ifs = new oemolistream();
        oemolostream ofs = new oemolostream();

        if (!ifs.open(args[0]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[0] + " for reading");
        }
        if (!ofs.open(args[1]))
        {
            OEChem.OEThrow.Fatal("Unable to open " + args[1] + " for writing");
        }

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);

        OESzybkiOptions opts = new OESzybkiOptions();

        // Selection of SMIRNOFF force field
        opts.GetGeneralOptions().SetForceFieldType(OEForceFieldType.SMIRNOFF99FROSST);
```

(continues on next page)

(continued from previous page)

```
        OESzybki sz = new OESzybki(opts);
        OESzybkiResults res = new OESzybkiResults();

        if (!sz.Call(mol, res))
        {
            return 1;
        }

        OEChem.OEWriteMolecule(ofs, mol);
        res.Print(OEChem.oeout);

        return 0;
    }
}
```

5.1.3 Optimization of a set of ligands

The next example illustrates the usage of Szybki TK to optimize a set of compounds with the MMFF94 force field in vacuum or in solution using Sheffield solvation model. Optionally attractive VdW can be removed. The optimization is done by default in full Cartesian coordinates, however torsion space optimization or single point calculation could be done too. A group of atoms which belong to specified SMARTS pattern might be excluded from optimization so their positions will be fixed at their initial coordinates. Note that the *OESzybki* object is made with a constructor which takes the instance of the *OESzybkiOptions* class.

Listing 3: Optimization of a Set of Ligands

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class LigandMultipleOptions
{
    public static void Main(string[] args)
```

(continues on next page)

(continued from previous page)

```
    {
        OEInterface itf = new OEInterface();
        if (!SetupInterface("LigandMultipleOptions", itf, args))
            return;

        oemolistream ifs = new oemolistream();
        if (!ifs.open(itf.GetString("-in")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-in") + " for reading
↪");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(itf.GetString("-out")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-out") + " for␣
↪writing");

        oeostream logfile = OEChem.oeout;
        if (itf.HasString("-log")) {
            if (!logfile.open(itf.GetString("-log")))
                OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-log") + " for␣
↪writing");
        }

        // Szybki options
        OESzybkiOptions opts = new OESzybkiOptions();

        // select run type
        if (itf.GetBool("-t"))
            opts.SetRunType(OERunType.TorsionsOpt);

        if (itf.GetBool("-n"))
            opts.SetRunType(OERunType.SinglePoint);

        // apply solvent model
        if (itf.GetBool("-s"))
            opts.GetSolventOptions().SetSolventModel(OESolventModel.Sheffield);

        // remove attractive VdW forces
        if (itf.GetBool("-a"))
            opts.GetGeneralOptions().SetRemoveAttractiveVdWForces(true);

        // Szybki object;
        OESzybki sz = new OESzybki(opts);

        // fix atoms
        if (itf.HasString("-f"))
            if (!sz.FixAtoms(itf.GetString("-f")))
                OEChem.OEThrow.Warning("Failed to fix atoms for " + itf.GetString("-f"));

        // process molecules
        OEMol mol = new OEMol();
        while (OEChem.OEReadMolecule(ifs, mol)) {
            logfile.write("\nMolecule " + mol.GetTitle() + "\n");
```

(continues on next page)

(continued from previous page)

```
            bool no_res = true;
            foreach (OESzybkiResults results in sz.Call(mol)) {
                results.Print(logfile);
                no_res = false;
            }

            if (no_res) {
                OEChem.OEThrow.Warning("No results processing molecule: " + mol.
↪GetTitle());
                continue;
            }
            OEChem.OEWriteMolecule(ofs, mol);
        }
    }

    private static bool SetupInterface(string progName, OEInterface itf, string[] args)
    {
        OEChem.OEConfigure(itf, interfaceData);
        if (OEChem.OECheckHelp(itf, args, progName, false))
            return false;
        if (!OEChem.OEParseCommandLine(itf, args, progName))
            return false;
        if (!OEChem.OEIsReadable(OEChem.OEGetFileType(OEChem.OEGetFileExtension(itf.
↪GetString("-in"))))) {
            OEChem.OEThrow.Warning(itf.GetString("-in") + " is not a readable input file
↪");
            return false;
        }
        if (!OEChem.OEIsWriteable(OEChem.OEGetFileType(OEChem.OEGetFileExtension(itf.
↪GetString("-out"))))) {
            OEChem.OEThrow.Warning(itf.GetString("-out") + " is not a writable output␣
↪file");
            return false;
        }
        return true;
    }

    private static string interfaceData = @"
!PARAMETER -in
  !TYPE string
  !REQUIRED true
  !BRIEF Input molecule file name.
!END

!PARAMETER -out
  !TYPE string
  !REQUIRED true
  !BRIEF Output molecule file name.
!END

!PARAMETER -log
  !TYPE string
```

(continues on next page)

(continued from previous page)

```
  !REQUIRED false
  !BRIEF Log file name. Defaults to standard out.
!END

!PARAMETER -s
  !TYPE bool
  !DEFAULT false
  !REQUIRED false
  !BRIEF Optimization in solution.
!END

!PARAMETER -t
  !TYPE bool
  !DEFAULT false
  !REQUIRED false
  !BRIEF Optimization of torsions.
!END

!PARAMETER -n
  !TYPE bool
  !DEFAULT false
  !REQUIRED false
  !BRIEF Single point calculation.
!END

!PARAMETER -a
  !TYPE bool
  !DEFAULT false
  !REQUIRED false
  !BRIEF No attractive VdW forces.
!END

!PARAMETER -f
  !TYPE string
  !REQUIRED false
  !BRIEF SMARTS pattern of fixed atoms.
!END
";
}
```

5.1.4 Optimization of a single ligand with the Newton-Raphson method

The next example in this section shows how to use Newton-Raphson optimization method, rather than the default BFGS:

Listing 4: Optimization of a Single Ligand with the Newton-Raphson Method

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
```

(continues on next page)

(continued from previous page)

```
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class SimpleNewton
{
    public static void Main(string[] args)
    {
        if (args.Length != 2)
            OEChem.OEThrow.Usage("SimpleNewton input_molecule output_molecule");

        string inputFile  = args[0];
        string outputFile = args[1];

        oemolistream ifs = new oemolistream();
        if (!ifs.open(inputFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);
        ifs.close();

        OESzybkiOptions opts = new OESzybkiOptions();
        opts.GetOptOptions().SetOptimizerType(OEOptType.NEWTON);
        opts.GetSolventOptions().SetSolventModel(OESolventModel.Sheffield);

        OESzybki sz = new OESzybki(opts);
        OESzybkiResults res = new OESzybkiResults();
        if (sz.Call(mol, res)) {
            OEChem.OEWriteMolecule(ofs, mol);
            res.Print(OEChem.oeout);
        }
        ofs.close();
    }
```

(continues on next page)

(continued from previous page)

```
}
```

5.1.5 Optimization of all conformers of a ligand

Finally, the last example in this section shows how to use Newton-Raphson optimization method on all conformers of a ligand. The current charges of the ligand are used and will not be changed during the optimization.

Listing 5: Optimization of All Conformers of a Ligand

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OEQuacPac;
using OpenEye.OESzybki;

public class SimpleConformer
{
    public static void Main(string[] args)
    {
        if (args.Length != 2)
            OEChem.OEThrow.Usage("SimpleConformer input_molecule output_molecule");

        string inputFile  = args[0];
        string outputFile = args[1];

        oemolistream ifs = new oemolistream();
        if (!ifs.open(inputFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OESzybkiOptions opts = new OESzybkiOptions();
        opts.GetOptOptions().SetOptimizerType(OEOptType.NEWTON);
```

(continues on next page)

(continued from previous page)

```
        opts.GetSolventOptions().SetSolventModel(OESolventModel.Sheffield);
        opts.GetGeneralOptions().SetForceFieldType(OEForceFieldType.MMFF94S);
        OEChargeEngineNoOp chargeEngine = new OEChargeEngineNoOp();
        opts.GetSolventOptions().SetChargeEngine(chargeEngine);

        OESzybki sz = new OESzybki(opts);
        OESzybkiResults res = new OESzybkiResults();
        OEMol mol = new OEMol();
        while(OEChem.OEReadMolecule(ifs, mol))
        {
            foreach(OEConfBase conf in mol.GetConfs())
            {
                if (sz.Call(conf, res))
                {
                    OEChem.OESetSDData(conf, new OESDDataPair("Total_energy",
                        res.GetTotalEnergy().ToString("0.0000")));
                }
            }
            OEChem.OEWriteMolecule(ofs, mol);
        }

        ifs.close();
        ofs.close();
    }
}
```

5.1.6 Optimization of a single bound ligand

The simplest case is illustrated below. Notice the usage of the method *OESzybki.SetProtein* which tells Szybki that the ligand is placed inside the protein. Since no protein-ligand electrostatics have been specified, neither the coordinates types which should be used by the optimizer, the code below performs the optimization for a rigid ligand using 6 translational-rotational coordinates in the MMFF94 VdW potential field.

Listing 5: Optimization of a Single Bound Ligand

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
```

(continues on next page)

(continued from previous page)

```
*/
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class SimpleProtein
{
    public static void Main(string[] args)
    {
        if (args.Length != 3)
            OEChem.OEThrow.Usage("SimpleProtein <molfile> <protein> <outfile>");

        string inputLFile = args[0];
        string inputPFile = args[1];
        string outputFile = args[2];

        oemolistream lfs = new oemolistream();
        if (!lfs.open(inputLFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputLFile + " for reading");

        oemolistream pfs = new oemolistream();
        if (!pfs.open(inputPFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputPFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OEGraphMol mol = new OEGraphMol();
        OEChem.OEReadMolecule(lfs, mol);
        lfs.close();

        OEGraphMol protein = new OEGraphMol();
        OEChem.OEReadMolecule(pfs, protein);
        pfs.close();

        OESzybkiOptions opts = new OESzybkiOptions();
        OESzybki sz = new OESzybki(opts);
        sz.SetProtein(protein);
        OESzybkiResults res = new OESzybkiResults();
        if (sz.Call(mol, res))
        {
            OEChem.OEWriteMolecule(ofs, mol);
            res.Print(OEChem.oeout);
        }
        ofs.close();
    }
}
```

5.2 Protein-Ligand Energetics and Optimization

Examples in this section show how to optimize a bound ligand.

5.2.1 Optimization of a set of bound ligands in a rigid receptor

The next example illustrates the usage of Szybki TK to optimize a set of ligands with MMFF94 force field inside a protein receptor. By default only VdW protein-ligand interaction is used. Optionally exact or grid Coulomb potential as well as PB solvent screening potentials could be added. When grid potential is selected (either Coulomb or PB) optionally it could be saved or read in when the corresponding grid file is present in the specified directory. Notice that when the exact Coulomb electrostatics is chosen, also the exact VdW potential is chosen (method *OESzybkiProteinOptions.SetExactVdWProteinLigand*) which allows for tight gradients convergence (methods *OESzybkiOptOptions.SetMaxIter* and *OESzybkiOptOptions.SetGradTolerance*). By default ligand is treated as a solid body, that is only its translational and rotational degrees of freedom are optimized. Optionally also torsional degrees could be optimized. In this example protein receptor is rigid. Molecular input file should contain initial 3D coordinates of molecules in any format supported by OEChem TK. Output file is specified with the -out flag. In addition a log file containing energy data terms values is written to stdout or a file specified by -log.

Listing 7: Optimization of a Set of Bound Ligands in a Rigid Receptor

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

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

        oemolistream lfs = new oemolistream();
        if (!lfs.open(itf.GetString("-in")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-in") + " for reading
↪");
```

(continues on next page)

(continued from previous page)

```
        oemolistream pfs = new oemolistream();
        if (!pfs.open(itf.GetString("-p")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-p") + " for reading
↪");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(itf.GetString("-out")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-out") + " for␣
↪writing");

        oeostream logfile = OEChem.oeout;
        if (itf.HasString("-log")) {
            if (!logfile.open(itf.GetString("-log")))
                OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-log") + " for␣
↪writing");
        }

        // Szybki options
        OESzybkiOptions opts = new OESzybkiOptions();

        // select optimization type
        if (itf.GetBool("-t"))
            opts.SetRunType(OERunType.TorsionsOpt);
        else
            opts.SetRunType(OERunType.CartesiansOpt);

        // select protein-electrostatic model
        string emodel = itf.GetString("-e");
        uint elecModel = OEProteinElectrostatics.NoElectrostatics;
        if (emodel.Equals("VdW"))
            elecModel = OEProteinElectrostatics.NoElectrostatics;
        else if (emodel.Equals("PB"))
            elecModel = OEProteinElectrostatics.GridPB;
        else if (emodel.Equals("Coulomb"))
            elecModel = OEProteinElectrostatics.GridCoulomb;
        else if (emodel.Equals("ExactCoulomb"))
            elecModel = OEProteinElectrostatics.ExactCoulomb;
        opts.GetProteinOptions().SetProteinElectrostaticModel(elecModel);

        opts.GetProteinOptions().SetExactVdWProteinLigand(true);
        opts.GetOptOptions().SetMaxIter(1000);
        opts.GetOptOptions().SetGradTolerance(1e-6);

        // Szybki object
        OESzybki sz = new OESzybki(opts);

        // read and setup protein
        OEGraphMol protein = new OEGraphMol();
        OEChem.OEReadMolecule(pfs, protein);
        sz.SetProtein(protein);
```

(continues on next page)

(continued from previous page)

```
        // save or load grid potential
        if (emodel.Equals("PB") || emodel.Equals("Coulomb")) {
            if(itf.HasString("-s"))
                sz.SavePotentialGrid(itf.GetString("-s"));
            if(itf.HasString("-l"))
                sz.LoadPotentialGrid(itf.GetString("-l"));
        }

        // process molecules
        foreach (OEMol mol in lfs.GetOEMols()) {
            logfile.write("\nMolecule " + mol.GetTitle() + "\n");
            bool no_res = true;
            foreach (OESzybkiResults res in sz.Call(mol)) {
                res.Print(logfile);
                no_res = false;
            }

            if (no_res) {
                OEChem.OEThrow.Warning("No results processing molecule: " + mol.
↪GetTitle());
                continue;
            }

            OEChem.OEWriteMolecule(ofs, mol);
        }
    }

    private static string interfaceData = @"
!PARAMETER -in
!TYPE string
!REQUIRED true
!BRIEF Input molecule file name.
!END

!PARAMETER -p
!TYPE string
!REQUIRED true
!BRIEF Input protein file name.
!END

!PARAMETER -out
!TYPE string
!REQUIRED true
!BRIEF Output molecule file name.
!END

!PARAMETER -log
!TYPE string
!REQUIRED false
!BRIEF Log file name. Defaults to standard out.
!END
```

(continues on next page)

(continued from previous page)

```
!PARAMETER -e
!TYPE string
!DEFAULT VdW
!LEGAL_VALUE VdW
!LEGAL_VALUE PB
!LEGAL_VALUE Coulomb
!LEGAL_VALUE ExactCoulomb
!BRIEF Protein ligand electrostatic model.
!END

!PARAMETER -t
!TYPE bool
!DEFAULT false
!REQUIRED false
!BRIEF Torsions added to the optimized variables.
!END

!PARAMETER -l
!TYPE string
!REQUIRED false
!BRIEF File name of the potential grid to be read.
!END

!PARAMETER -s
!TYPE string
!REQUIRED false
!BRIEF File name of the potential grid to be saved.
!END
";

}
```

5.2.2 Optimization of a set of bound ligands in a partially flexible receptor

The next example is very similar with respect to the previous one, but the side chains of the protein which are within the specified range from the ligand, are made flexible during optimization (methods *OESzybkiProteinOptions.SetProteinFlexibilityType* and *OESzybkiProteinOptions.SetProteinFlexibilityRange*). Optionally, partially optimized structure can be saved to a file.

Listing 8: Optimization of a Set of Bound Ligands in a Partially Flexible Receptor

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
```

(continues on next page)

(continued from previous page)

```
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

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

        oemolistream lfs = new oemolistream();
        if (!lfs.open(itf.GetString("-in")))
          OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-in") + " for reading
↪");

        oemolistream pfs = new oemolistream();
        if (!pfs.open(itf.GetString("-p")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-p") + " for reading
↪");

        oemolostream olfs = new oemolostream();
        if (!olfs.open(itf.GetString("-out")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-out") + " for␣
↪writing");

        oemolostream opfs = new oemolostream();
        if (itf.HasString("-s")) {
            if (!opfs.open(itf.GetString("-s")))
                OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-s") + " for␣
↪writing");
        }

        oeostream logfile = OEChem.oeout;
        if (itf.HasString("-log")) {
            if (!logfile.open(itf.GetString("-log")))
              OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-log") + " for␣
↪writing");
        }

        // Szybki options
        OESzybkiOptions opts = new OESzybkiOptions();

        // select optimization type
        string opt = itf.GetString("-opt");
```

(continues on next page)

(continued from previous page)

```
        if (opt.Equals("Cartesian"))
            opts.SetRunType(OERunType.CartesiansOpt);
        else if (opt.Equals("Torsion"))
            opts.SetRunType(OERunType.TorsionsOpt);
        else if (opt.Equals("SolidBody"))
            opts.SetRunType(OERunType.SolidBodyOpt);

        // select protein-electrostatic model
        string emodel = itf.GetString("-e");
        uint elecModel = OEProteinElectrostatics.NoElectrostatics;
        if (emodel.Equals("VdW"))
            elecModel = OEProteinElectrostatics.NoElectrostatics;
        else if (emodel.Equals("PB"))
            elecModel = OEProteinElectrostatics.GridPB;
        else if (emodel.Equals("Coulomb"))
            elecModel = OEProteinElectrostatics.GridCoulomb;
        else if (emodel.Equals("ExactCoulomb"))
            elecModel = OEProteinElectrostatics.ExactCoulomb;
        opts.GetProteinOptions().SetProteinElectrostaticModel(elecModel);

        // use smooth potential and tight convergence
        if (emodel.Equals("VdW") || emodel.Equals("ExactCoulomb")) {
            opts.GetProteinOptions().SetExactVdWProteinLigand(true);
            opts.GetOptOptions().SetMaxIter(1000);
            opts.GetOptOptions().SetGradTolerance(1e-6);
        }

        // protein flexibility
        opts.GetProteinOptions().SetProteinFlexibilityType(OEProtFlex.SideChains);
        opts.GetProteinOptions().SetProteinFlexibilityRange(itf.GetDouble("-d"));

        // Szybki object
        OESzybki sz = new OESzybki(opts);

        // read and setup protein
        OEGraphMol protein = new OEGraphMol();
        OEGraphMol oprotein = new OEGraphMol();  // optimized protein
        OEChem.OEReadMolecule(pfs, protein);
        sz.SetProtein(protein);
        pfs.close();

        // process molecules
        foreach (OEMol mol in lfs.GetOEMols()) {
            logfile.write("\nMolecule " + mol.GetTitle() + "\n");
            foreach (OESzybkiResults res in sz.Call(mol))
                res.Print(logfile);

            OEChem.OEWriteMolecule(olfs, mol);
            if (itf.HasString("-s")) {
                sz.GetProtein(oprotein);
                OEChem.OEWriteMolecule(opfs, oprotein);
            }
```

(continues on next page)

(continued from previous page)

```
        }
        opfs.close();
    }

    private static string interfaceData = @"
!BRIEF -in input_molecule -p protein -out output_molecule
!PARAMETER -in
!TYPE string
!REQUIRED true
!BRIEF Input molecule file name.
!END

!PARAMETER -p
!TYPE string
!REQUIRED true
!BRIEF Input protein file name.
!END

!PARAMETER -out
!TYPE string
!REQUIRED true
!BRIEF Output molecule file name.
!END

!PARAMETER -log
!TYPE string
!REQUIRED false
!BRIEF Log file name. Defaults to standard out.
!END

!PARAMETER -e
!TYPE string
!DEFAULT VdW
!LEGAL_VALUE VdW
!LEGAL_VALUE PB
!LEGAL_VALUE Coulomb
!LEGAL_VALUE ExactCoulomb
!BRIEF Protein ligand electrostatic model.
!END

!PARAMETER -opt
!TYPE string
!DEFAULT Cartesian
!LEGAL_VALUE Cartesian
!LEGAL_VALUE Torsion
!LEGAL_VALUE SolidBody
!BRIEF Optimization method
!END

!PARAMETER -d
!TYPE double
!DEFAULT 5.0
```

(continues on next page)

(continued from previous page)

```
!BRIEF Distance criteria from protein side-chains flexibility.
!END

!PARAMETER -s
!TYPE string
!REQUIRED false
!BRIEF File name the partially optimized protein will be saved.
!END
";
}
```

5.2.3 Optimization of a bound ligand in a partially flexible receptor

The next example is very similar with respect to the previous one, but the list of residue numbers of the protein that are made flexible during optimization is specified by the `-residues` flag (see method *OESzybkiProteinOptions.AddFlexibleResidue*). Partially optimized ligand and protein structures are saved to files specified by the `-outl` and `-outp` flags, respectively.

Listing 9: Optimization of a Bound Ligand in a Partially Flexible Feceptor

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

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

        oemolistream ifs = new oemolistream();
        if (!ifs.open(itf.GetString("-in")))
          OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-in") + " for reading
```

(continues on next page)

(continued from previous page)

```
↪");

        oemolistream pfs = new oemolistream();
        if (!pfs.open(itf.GetString("-protein")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-protein") + " for␣
↪reading");

        oemolostream olfs = new oemolostream();
        if (!olfs.open(itf.GetString("-outl")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-outl") + " for␣
↪writing");

        oemolostream opfs = new oemolostream();
        if (!opfs.open(itf.GetString("-outp")))
            OEChem.OEThrow.Fatal("Unable to open " + itf.GetString("-outp") + " for␣
↪writing");

        OEGraphMol protein = new OEGraphMol();
        OEGraphMol ligand = new OEGraphMol();
        OEChem.OEReadMolecule(pfs, protein);
        OEChem.OEReadMolecule(ifs, ligand);
        pfs.close();
        ifs.close();

        // Szybki options
        OESzybkiOptions opts = new OESzybkiOptions();
        opts.GetGeneralOptions().SetForceFieldType(OEForceFieldType.MMFF94S);
        opts.GetProteinOptions().SetProteinElectrostaticModel(OEProteinElectrostatics.
↪ExactCoulomb);
        opts.GetProteinOptions().SetProteinFlexibilityType(OEProtFlex.SideChainsList);
        opts.SetRunType(OERunType.CartesiansOpt);
        opts.GetOptOptions().SetMaxIter(2000);
        opts.GetOptOptions().SetGradTolerance(1.0e-6);

        OEStringIter res_num = itf.GetStringList("-residues");
        res_num.ToFirst();
        foreach (string rn in res_num) {
            foreach (OEAtomBase atom in protein.GetAtoms()) {
                OEResidue res = OEChem.OEAtomGetResidue(atom);
                if(res.GetResidueNumber().ToString().Equals(rn)) {
                    opts.AddFlexibleResidue(res);
                    break;
                }
            }
        }

        //optimization
        OESzybki sz = new OESzybki(opts);
        sz.SetProtein(protein);
        OESzybkiResults result = new OESzybkiResults();
        sz.Call(ligand, result);
```

(continues on next page)

(continued from previous page)

```
        OEChem.OEWriteMolecule(olfs, ligand);
        sz.GetProtein(protein);
        OEChem.OEWriteMolecule(opfs, protein);

        olfs.close();
        opfs.close();
    }

    private static string interfaceData = @"
!PARAMETER -in
!TYPE string
!REQUIRED true
!BRIEF Input molecule file name.
!END

!PARAMETER -protein
!TYPE string
!REQUIRED true
!BRIEF Input protein file name
!END

!PARAMETER -outl
!TYPE string
!REQUIRED true
!BRIEF Output ligand file name.
!END

!PARAMETER -outp
!TYPE string
!REQUIRED true
!BRIEF Output protein file name.
!END

!PARAMETER -residues
!TYPE int
!LIST true
!REQUIRED true
!BRIEF List of residues numbers to be optimized along with the ligand
!END
";
}
```

5.2.4 Estimation of PB binding for a set of ligands

This example shows how to fast estimate binding for a set of ligands using PB electrostatics. Two *OESzybki* objects are instantiated: one for the optimization of bound ligands in VdW-Coulomb potential, and the second one which performs single-point PB calculations. Final results are attached as SD tags to the output molecules.

Listing 10: Estimation of PB Binding for a Set of Ligands

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class ProteinLigandPB
{
    public static void Main(string[] args)
    {
        if (args.Length != 3)
            OEChem.OEThrow.Usage("ProteinLigandPB ligand_file protein_file output_file␣
↪(SDF or OEB)");

        string ligandFile  = args[0];
        string proteinFile = args[1];
        string outputFile  = args[2];

        oemolistream ligfs = new oemolistream();
        if (!ligfs.open(ligandFile))
            OEChem.OEThrow.Fatal("Unable to open " + ligandFile + " for reading");

        oemolistream prtfs = new oemolistream();
        if (!prtfs.open(proteinFile))
            OEChem.OEThrow.Fatal("Unable to open " + proteinFile + " for reading");

        oemolostream outfs = new oemolostream();
        if (!outfs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        if (!OEChem.OEIsSDDataFormat(outfs.GetFormat()))
            OEChem.OEThrow.Fatal("Output file does not support SD data used by this␣
↪example");
```

(continues on next page)

(continued from previous page)

```
        // Szybki options for VdW-Coulomb calculations
        OESzybkiOptions optsC = new OESzybkiOptions();
        optsC.GetProteinOptions().SetProteinElectrostaticModel(OEProteinElectrostatics.
↪ExactCoulomb);
        optsC.SetRunType(OERunType.CartesiansOpt);

        // Szybki options for PB calculations
        OESzybkiOptions optsPB = new OESzybkiOptions();
        optsPB.GetProteinOptions().SetProteinElectrostaticModel(OEProteinElectrostatics.
↪SolventPBForces);
        optsPB.SetRunType(OERunType.SinglePoint);

        // Szybki objects
        OESzybki szC = new OESzybki(optsC);
        OESzybki szPB = new OESzybki(optsPB);

        // read and setup protein
        OEGraphMol protein = new OEGraphMol();
        OEChem.OEReadMolecule(prtfs, protein);
        prtfs.close();
        szC.SetProtein(protein);
        szPB.SetProtein(protein);

        // process molecules
        OEMol mol = new OEMol();
        while(OEChem.OEReadMolecule(ligfs, mol)) {

            OESzybkiResultsIter results = szC.Call(mol); // optimize mol
            if (!results.IsValid()) {
                OEChem.OEThrow.Warning("No results processing molecule: " + mol.
↪GetTitle());
                continue;
            }

            results = szPB.Call(mol);
            OEConfBaseIter conf = mol.GetConfs();
            if (results.IsValid()) {
                foreach (OESzybkiResults res in results) {
                    if (!conf.IsValid())
                        continue;

                    for (uint i = 0; i < OEPotentialTerms.Max; ++i) {
                        if (i == OEPotentialTerms.ProteinLigandInteraction || i ==␣
↪OEPotentialTerms.VdWProteinLigand ||
                            i == OEPotentialTerms.CoulombProteinLigand     || i ==␣
↪OEPotentialTerms.ProteinDesolvation ||
                            i == OEPotentialTerms.LigandDesolvation        || i ==␣
↪OEPotentialTerms.SolventScreening) {
                            String energy = String.Format("{0,9:F4}", res.
↪GetEnergyTerm(i));
                            OEChem.OEAddSDData(conf.Target(), OESzybkilib.
↪OEGetEnergyTermName(i), energy);
```

(continues on next page)

(continued from previous page)

```
                    }
                }
                conf.Increment();
            }
        }
        OEChem.OEWriteMolecule(outfs, mol);
    }
    ligfs.close();
    outfs.close();
  }
}
```

5.2.5 Optimization of a bound ligand using Newton-Raphson method

The last example in this section illustrates how to use SzybkiTK to optimize a ligand in partially flexible protein with Newton-Raphson optimization method.

Listing 11: Optimization of a Bound Ligand Using Newton-Raphson Method

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class LigandFlexProteinNewton
{
    public static void Main(string[] args)
    {
        if (args.Length != 3)
            OEChem.OEThrow.Usage("LigandFlexProteinNewton protein input_ligand output_
↪ligand");

        string proteinFile = args[0];
        string ligandFile  = args[1];
```

(continues on next page)

(continued from previous page)

```
        string outligandFile = args[2];

        oemolistream pfs = new oemolistream();
        if (!pfs.open(proteinFile))
            OEChem.OEThrow.Fatal("Unable to open " + proteinFile + " for reading");

        oemolistream lfs = new oemolistream();
        if (!lfs.open(ligandFile))
            OEChem.OEThrow.Fatal("Unable to open " + ligandFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outligandFile))
            OEChem.OEThrow.Fatal("Unable to open " + outligandFile + " for writing");

        OEGraphMol mol = new OEGraphMol();
        OEGraphMol protein = new OEGraphMol();
        OEChem.OEReadMolecule(lfs, mol);
        lfs.close();
        OEChem.OEReadMolecule(pfs, protein);
        pfs.close();

        OESzybkiOptions opts = new OESzybkiOptions();
        opts.GetOptOptions().SetOptimizerType(OEOptType.NEWTON);
        opts.GetProteinOptions().SetProteinElectrostaticModel(OEProteinElectrostatics.
↪ExactCoulomb);
        opts.GetProteinOptions().SetProteinFlexibilityType(OEProtFlex.Residues);
        opts.GetProteinOptions().SetProteinFlexibilityRange(2.0f);

        OESzybki sz = new OESzybki(opts);
        sz.SetProtein(protein);

        OESzybkiResults res = new OESzybkiResults();
        if (sz.Call(mol, res)) {
            OEChem.OEWriteMolecule(ofs, mol);
            res.Print(OEChem.oeout);
        }
        ofs.close();
    }
}
```

5.3 DU Protein-Ligand Optimization with FF14SB-Parsley

Examples in this section show how to optimize a ligand in the protein active site using the *OEFixedProteinLigandOptimizer* and the *OEFlexProteinLigandOptimizer* classes. Both of these classes allow use of the FF14SB-Parsley forcefield, along with the MMFF and the MMFF-AMBER forcefields. These examples also demonstrate how to use design units as a source for the protein or the ligand. The first example optimizes an external ligand in a protein active site from a design unit, and the second example uses both the ligand and the protein from the same design unit.

5.3.1 Optimization of ligand in a rigid active site

This examples shows how to optimize a series of ligands in a rigid active site. The active site input is taken from a OEDesignUnit.

5.3.2 Optimization of ligand in a partially flexible active site

This examples shows how to optimize a ligand in a partially flexible active site. In this example, both the ligand and the active site is taken from a single OEDesignUnit.

5.4 Entropy estimation

5.4.1 Estimation of solution ligand entropy

The following code illustrates how to estimate compound entropy in solution with Szybki TK.

Listing 14: Ligand Entropy

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class LigandEntropy
{
    public static void Main(string[] args)
    {
        OELigandEntropyOptions szOpts = new OELigandEntropyOptions();
        OESimpleAppOptions opts = new OESimpleAppOptions(szOpts, "LigandEntropy",␣
↪OEFileStringType.Mol3D);
        if(OEChem.OEConfigureOpts(opts, args, false) == OEOptsConfigureStatus.Help)
            Environment.Exit(1);
        szOpts.UpdateValues(opts);

        oemolistream ifs = new oemolistream();
```

(continues on next page)

(continued from previous page)

```
        if(!ifs.open(opts.GetInFile()))
            OEChem.OEThrow.Fatal("Unable to open " + opts.GetInFile() + " for reading");

        OEMol mol = new OEMol();
        while (OEChem.OEReadMolecule(ifs, mol))
        {
            OEChem.OEThrow.Info("Title: " + mol.GetTitle());
            OEEntropyResults res = new OEEntropyResults();
            uint ret_code = OESzybkilib.OELigandEntropy(res, mol, szOpts);
            if(ret_code == OESzybkiReturnCode.Success)
            {
                Console.WriteLine(String.Format("Configurational entropy {0,10:F2}", res.
↪GetConfigurationalEntropy()));
                Console.WriteLine(String.Format("Solvation entropy       {0,10:F2}", res.
↪GetSolvationEntropy()));
                Console.WriteLine(String.Format("Total entropy  {0,10:F2}", res.
↪GetTotalEntropy()));
            }
            else
                OEChem.OEThrow.Warning("Entropy Calculation Failed");
        }
    }
}
```

5.4.2 Estimation of bound ligand entropy

The next example below shows how to calculate entropy of a bound ligand.

Listing 15: Estimation of Bound Ligand Entropy

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;
```

(continues on next page)

(continued from previous page)

```
public class BoundEntropy
{
    public static void Main(string[] args)
    {
        OEBoundEntropyOptions szOpts = new OEBoundEntropyOptions();
        OERefInputAppOptions opts = new OERefInputAppOptions(szOpts, "BoundEntropy",␣
↪OEFileStringType.Mol3D,
            OEFileStringType.Mol3D, "-protein");
        if(OEChem.OEConfigureOpts(opts, args, false) == OEOptsConfigureStatus.Help)
            Environment.Exit(1);
        szOpts.UpdateValues(opts);

        oemolistream ifs = new oemolistream();
        if(!ifs.open(opts.GetInFile()))
            OEChem.OEThrow.Fatal("Unable to open " + opts.GetInFile() + " for reading");
        oemolistream pfs = new oemolistream();
        if (!pfs.open(opts.GetRefFile()))
            OEChem.OEThrow.Fatal("Unable to open " + opts.GetRefFile() + " for reading");

        OEMol protein = new OEMol();
        if (!OEChem.OEReadMolecule(pfs, protein))
            OEChem.OEThrow.Fatal("Failed to read protein");

        OEMol mol = new OEMol();
        while (OEChem.OEReadMolecule(ifs, mol))
        {
            OEChem.OEThrow.Info("Title: " + mol.GetTitle());
            OEEntropyResults res = new OEEntropyResults();
            uint ret_code = OESzybkilib.OEBoundLigandEntropy(res, protein, mol, szOpts);
            if(ret_code == OESzybkiReturnCode.Success)
            {
                Console.WriteLine(String.Format("Configurational entropy {0,10:F2}", res.
↪GetConfigurationalEntropy()));
                Console.WriteLine(String.Format("Solvation entropy       {0,10:F2}", res.
↪GetSolvationEntropy()));
                Console.WriteLine(String.Format("Total entropy  {0,10:F2}", res.
↪GetTotalEntropy()));
            }
            else
                OEChem.OEThrow.Warning("Entropy Calculation Failed");
        }
    }
}
```

5.5 Solvation free energy estimation

Listing 16: Solvation Free Energy Estimation

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

(continues on next page)

(continued from previous page)

```
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class FreeFormSolv
{
    public static void Main(string[] args)
    {
        if (args.Length != 2)
            OEChem.OEThrow.Usage("FreeFormSolv <input> <output>");

        string inputFile  = args[0];
        string outputFile = args[1];

        oemolistream ifs = new oemolistream();
        if (!ifs.open(inputFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);
        ifs.close();

        OEFreeFormSolvOptions opts = new OEFreeFormSolvOptions();
        opts.SetIonicState(OEFreeFormIonicState.Uncharged);
        OEFreeFormSolvResults res = new OEFreeFormSolvResults();

        OEGraphMol omol = new OEGraphMol();
        if (!OESzybkilib.OEEstimateSolvFreeEnergy(res, omol, mol, opts))
            OEChem.OEThrow.Error("Failed to calculate solvation free energy for molecule
↪" + mol.GetTitle());
```

(continues on next page)

(continued from previous page)

```
        double solvenergy = res.GetSolvationFreeEnergy();
        OEChem.OEThrow.Info(String.Format("Solvation free energy for compound {0} is {1,
↪6:F2} kcal/mol",
                                          mol.GetTitle(), solvenergy));

        OEChem.OEWriteMolecule(ofs, omol);
        ofs.close();
    }
}
```

5.6 Conformations free energy estimation

> **Warning**
>
> This capability should not be used on 32-bit platforms because the memory requirements are too high.

5.6.1 Simple free energy estimation

The following code illustrates how to use the high level commands from *OEFreeFormConf* to estimate conformer free energies in solution with Szybki TK.

Listing 17: Simple Free Energy Estimation

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class FreeFormConf
{
    public static void Main(string[] args)
```

(continues on next page)

(continued from previous page)

```
    {
        if (args.Length != 2)
            OEChem.OEThrow.Usage("FreeFormConf <input> <output>");

        string inputFile  = args[0];
        string outputFile = args[1];

        oemolistream ifs = new oemolistream();
        if (!ifs.open(inputFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);
        ifs.close();

        OEFreeFormConfOptions opts = new OEFreeFormConfOptions();
        OEFreeFormConf ffconf = new OEFreeFormConf(opts);

        OEMol omol = new OEMol(mol);
        if (!(ffconf.EstimateFreeEnergies(omol) == OEFreeFormReturnCode.Success))
            OEChem.OEThrow.Error("Failed to estimate conformational free energies for␣
↪molecule " + mol.GetTitle());

        OEFreeFormConfResults res = new OEFreeFormConfResults(omol);
        OEChem.OEThrow.Info(String.Format("Number of unique conformations: {0}", res.
↪GetNumUniqueConfs()));
        OEChem.OEThrow.Info("Conf.  Delta_G   Vibrational_Entropy");
        OEChem.OEThrow.Info("      [kcal/mol]     [J/(mol K)]");
        foreach (OESingleConfResult r in res.GetResultsForConformations()) {
            OEChem.OEThrow.Info(String.Format("{0,2} {1,10:F2} {2,14:F2}",
                                              r.GetConfIdx(), r.GetDeltaG(), r.
↪GetVibrationalEntropy()));
        }
        OEChem.OEWriteMolecule(ofs, omol);
        ofs.close();
    }
}
```

5.6.2 Simple restriction energy estimation

The following code illustrates how to use the high level commands from *OEFreeFormConf* to estimate restriction energies on conformers, along with conformer free energies in solution, with Szybki TK.

Listing 18: Simple Restriction Energy Estimation

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

(continues on next page)

(continued from previous page)

```
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class FreeFormConfRstr
{
    public static void Main(string[] args)
    {
        if (args.Length != 2)
            OEChem.OEThrow.Usage("FreeFormConfRstr <input> <output>");

        string inputFile  = args[0];
        string outputFile = args[1];

        oemolistream ifs = new oemolistream();
        if (!ifs.open(inputFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);
        ifs.close();

        OEFreeFormConfOptions opts = new OEFreeFormConfOptions();
        OEFreeFormConf ffconf = new OEFreeFormConf(opts);

        OEMol omol = new OEMol(mol);
        OEMol rmol = new OEMol(mol);
        if (ffconf.EstimateFreeEnergies(omol, rmol) != OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Failed to estimate conformational free energies for␣
↪molecule " + mol.GetTitle());

        OEFreeFormConfResults res = new OEFreeFormConfResults(omol);
```

(continues on next page)

(continued from previous page)

```
        OEChem.OEThrow.Info(String.Format("Number of unique conformations: {0}", res.
↪GetNumUniqueConfs()));
        OEChem.OEThrow.Info("Conf.  Delta_G   Vibrational_Entropy");
        OEChem.OEThrow.Info("      [kcal/mol]     [J/(mol K)]");
        foreach (OESingleConfResult r in res.GetResultsForConformations()) {
            OEChem.OEThrow.Info(String.Format("{0,2} {1,10:F2} {2,14:F2}",
                                              r.GetConfIdx(), r.GetDeltaG(), r.
↪GetVibrationalEntropy()));
        }

        OERestrictionEnergyResult rstrRes = new OERestrictionEnergyResult(rmol);
        OEChem.OEThrow.Info(String.Format("Global strain: {0}", rstrRes.
↪GetGlobalStrain()));
        OEChem.OEThrow.Info(String.Format("Local strain: {0}", rstrRes.
↪GetLocalStrain()));
        OEChem.OEWriteMolecule(ofs, omol);
        ofs.close();
    }
}
```

5.6.3 Advanced free energy estimation

The following code illustrates how to use the low level commands from *OEFreeFormConfAdvanced* to estimate the conformer free energies in solution with Szybki TK. These low level methods of estimation gives an advantage over the high level methods of *OEFreeFormConf* in that these gives the user control over better managing certain expensive parts of the calculation.

Listing 19: Advanced Free Energy Estimation

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;
```

(continues on next page)

(continued from previous page)

```
public class FreeFormConfAdvanced
{
    public static void Main(string[] args)
    {
        if (args.Length != 2)
            OEChem.OEThrow.Usage("FreeFormConfAdvanced <input> <output>");

        string inputFile  = args[0];
        string outputFile = args[1];

        oemolistream ifs = new oemolistream();
        if (!ifs.open(inputFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);
        ifs.close();

        OEFreeFormConfOptions opts = new OEFreeFormConfOptions();
        OEFreeFormConfAdvanced ffconf = new OEFreeFormConfAdvanced(opts);

        // Make a copy of our MCMol.  We will execute the FreeFormConf commands on
        // the copied molecule so that our original molecule stays intact.
        OEMol omol = new OEMol(mol);

        // Prepare a comprehensive ensemble of molecule conformers. This will
        // generate a comprehensive set of conformers, assign solvent charges on the␣
↪molecule
        // and check that the ensemble is otherwise ready for FreeFormConf calculations.
        if (ffconf.PrepareEnsemble(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Failed to prepare ensemble for FreeFormConf␣
↪calculations");

        // Perform loose optimization of the ensemble conformers.  We will remove
        // duplicates based on the loose optimization, to reduce the time needed for
        // tighter, more stricter optimization
        if(ffconf.PreOptimizeEnsemble(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Pre-optimization of the ensembles failed");

        // Remove duplicates from the pre-optimized ensemble
        if(ffconf.RemoveDuplicates(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Duplicate removal from the ensembles failed");

        // Perform the desired optimization.  This uses a stricter convergence
        // criteria in the default settings.
        if(ffconf.Optimize(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Optimization of the ensembles failed");
```

(continues on next page)

(continued from previous page)

```
        // Remove duplicates to obtain the set of minimum energy conformers
        if(ffconf.RemoveDuplicates(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Duplicate removal from the ensembles failed");

        // Perform FreeFormConf free energy calculations.  When all the above steps
        // have already been performed on the ensemble, this energy calculation
        // step is fast.
        if(ffconf.EstimateEnergies(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Estimation of FreeFormConf energies failed");

        // Gather results of calculation into a results object for ease of viewing, etc.
        OEFreeFormConfResults res = new OEFreeFormConfResults(omol);
        OEChem.OEThrow.Info(String.Format("Number of unique conformations: {0}", res.
↪GetNumUniqueConfs()));
        OEChem.OEThrow.Info("Conf.  Delta_G   Vibrational_Entropy");
        OEChem.OEThrow.Info("      [kcal/mol]     [J/(mol K)]");
        foreach (OESingleConfResult r in res.GetResultsForConformations()) {
            OEChem.OEThrow.Info(String.Format("{0,2} {1,10:F2} {2,14:F2}",
                                              r.GetConfIdx(), r.GetDeltaG(), r.
↪GetVibrationalEntropy()));
        }

        OEChem.OEWriteMolecule(ofs, omol);
        ofs.close();
    }
}
```

5.6.4 Advanced restriction energy estimation

The following code illustrates how to use the low level commands from *OEFreeFormConfAdvanced* to estimate the restriction energies of conformers, along with conformer free energies in solution, with Szybki TK. These low level methods of estimation gives an advantage over the high level methods of *OEFreeFormConf* in that these gives the user control over better managing certain expensive parts of the calculation.

Listing 20: Advanced Restriction Energy Estimation

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
```

(continues on next page)

(continued from previous page)

```
or its use.
*/
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class FreeFormConfAdvRstr
{
    public static void Main(string[] args)
    {
        if (args.Length != 2)
            OEChem.OEThrow.Usage("FreeFormConfAdvRstr <input> <output>");

        string inputFile  = args[0];
        string outputFile = args[1];

        oemolistream ifs = new oemolistream();
        if (!ifs.open(inputFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);
        ifs.close();

        OEFreeFormConfOptions opts = new OEFreeFormConfOptions();
        OEFreeFormConfAdvanced ffconf = new OEFreeFormConfAdvanced(opts);

        // Make a copy of our MCMol.  We will execute the FreeFormConf commands on
        // the copied molecule so that our original molecule stays intact.
        OEMol omol = new OEMol(mol);

        // Make further copies of our original molecule.  The copied molecule(s) would␣
↪be used
        // as source on which retriction energies would be calculated
        OEMol rmol = new OEMol(mol);
        OEMol fmol = new OEMol(mol);

        // Prepare a comprehensive ensemble of molecule conformers.  For calculation
        // of restriction energies we want to make sure that all the corresponding free
        // conformers are also part of the comprehensive ensemble.  This will also
        // assign solvent charges on the molecule and check that the ensemble is
        // otherwise ready for FreeFormConf calculations. The resulting `fmol`
        // contains the correspondig free conformers.
        if (ffconf.PrepareEnsemble(omol, rmol, fmol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Failed to prepare ensemble for FreeFormConf␣
↪calculations");
```

(continues on next page)

(continued from previous page)

```
        // Perform loose optimization of the ensemble conformers.  We will remove
        // duplicates based on the loose optimization, to reduce the time needed for
        // tighter, more stricter optimization
        if(ffconf.PreOptimizeEnsemble(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Pre-optimization of the ensembles failed");

        // Remove duplicates from the pre-optimized ensemble
        if(ffconf.RemoveDuplicates(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Duplicate removal from the ensembles failed");

        // Perform the desired optimization.  This uses a stricter convergence
        // criteria in the default settings.
        if(ffconf.Optimize(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Optimization of the ensembles failed");

        // Remove duplicates to obtain the set of minimum energy conformers
        if(ffconf.RemoveDuplicates(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Duplicate removal from the ensembles failed");

        // Perform FreeFormConf free energy calculations.  When all the above steps
        // have already been performed on the ensemble, this energy calculation
        // step is fast.
        if(ffconf.EstimateEnergies(omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Estimation of FreeFormConf energies failed");

        // Gather results of calculation into a results object for ease of viewing, etc.
        OEFreeFormConfResults res = new OEFreeFormConfResults(omol);
        OEChem.OEThrow.Info(String.Format("Number of unique conformations: {0}", res.
↪GetNumUniqueConfs()));
        OEChem.OEThrow.Info("Conf.  Delta_G   Vibrational_Entropy");
        OEChem.OEThrow.Info("      [kcal/mol]     [J/(mol K)]");
        foreach (OESingleConfResult r in res.GetResultsForConformations()) {
            OEChem.OEThrow.Info(String.Format("{0,2} {1,10:F2} {2,14:F2}",
                                              r.GetConfIdx(), r.GetDeltaG(), r.
↪GetVibrationalEntropy()));
        }

        // Identify the corresponding conformer(s) to the free minimized conformer(s).
        // If identified, the corresponding (Conf)Free energy information is also
        // copied to the free conformers
        if(ffconf.IdentifyConformer(fmol, omol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Identification of free conformer(s) failed");

        // Estimate restriction energies. Since both restricted and free conformer
        // energy components are already available, this operation is fast.
        if(ffconf.EstimateRestrictionEnergy(fmol, rmol)!=OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Restriction energy estimation failed");

        // Gather restriction energies into a results object for ease of viewing, etc.
        OERestrictionEnergyResult rstrRes = new OERestrictionEnergyResult(fmol);
        OEChem.OEThrow.Info(String.Format("Global strain: {0}", rstrRes.
↪GetGlobalStrain()));
```

(continues on next page)

(continued from previous page)

```
        OEChem.OEThrow.Info(String.Format("Local strain: {0}", rstrRes.
↪GetLocalStrain()));

        // Optionally it is desired to perform a restrained optimization of the
        // restricted conformer(s) to brush out any energy differences due to
        // force field constaints or the sources of coonformer coordinates.  Note: The
        // high level EstimateFreeEnergy method does not perform this opertion.
        if (ffconf.OptimizeRestraint(rmol) != OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Restraint optimization of the conformer(s) failed");

        // Estimate restriction energies. Since both restricted and free conformer
        // energy components are already available, this operation is fast.
        if (ffconf.EstimateRestrictionEnergy(fmol, rmol) != OEFreeFormReturnCode.Success)
            OEChem.OEThrow.Error("Restriction energy estimation failed");

        // Gather restriction energies into a results object for ease of viewing, etc.
        OERestrictionEnergyResult rstrRes2 = new OERestrictionEnergyResult(fmol);
        OEChem.OEThrow.Info(String.Format("Global strain: {0}", rstrRes2.
↪GetGlobalStrain()));
        OEChem.OEThrow.Info(String.Format("Local strain: {0}", rstrRes2.
↪GetLocalStrain()));

        OEChem.OEWriteMolecule(ofs, omol);
        ofs.close();
    }
}
```

5.6.5 Finding similar conformers

The following code illustrates how to find similar conformers to the ones at hand, from a pool of minimum energy conformers.

Listing 21: Finding Similar Conformers

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
```

(continues on next page)

(continued from previous page)

```
using System;
using OpenEye.OEChem;
using OpenEye.OESzybki;
using OpenEye.OEQuacPac;

public class FindSimilarConfs
{
    public static void Main(string[] args)
    {
        if (args.Length != 2)
            OEChem.OEThrow.Usage("FindSimilarConfs <input> <output>");

        string inputFile  = args[0];
        string outputFile = args[1];

        oemolistream ifs = new oemolistream();
        if (!ifs.open(inputFile))
            OEChem.OEThrow.Fatal("Unable to open " + inputFile + " for reading");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(outputFile))
            OEChem.OEThrow.Fatal("Unable to open " + outputFile + " for writing");

        OEMol mol = new OEMol();
        OEChem.OEReadMolecule(ifs, mol);
        ifs.close();

        OEFreeFormConfOptions opts = new OEFreeFormConfOptions();
        OEFreeFormConf ffconf = new OEFreeFormConf(opts);

        // Estimate free energies to ontain the minimum energy conformers
        OEMol omol = new OEMol(mol);
        if (!(ffconf.EstimateFreeEnergies(omol) == OEFreeFormReturnCode.Success))
            OEChem.OEThrow.Error("Failed to estimate conformational free energies for␣
↪molecule " + mol.GetTitle());

        // Find similar conformers to the ones we started with, from the
        // pool of minimum energy conformers
        OEMol fmol = new OEMol(mol);
        foreach (OEConfBase conf in mol.GetConfs())
            ffconf.FindSimilarConfs(fmol, omol, conf, new OESimilarByRMSD(0.05));

        OEChem.OEWriteMolecule(ofs, fmol);
        ofs.close();
    }
}
```

5.6.6 Torsion scan

The following code shows how to scan all torsions in the input molecule

CHAPTER

SIX

RELEASE HISTORY

6.1 Szybki TK 2.9.2

Minor internal improvements have been made.

6.2 Szybki TK 2.9.1

Minor internal improvements have been made.

6.3 Szybki TK 2.9.0

6.3.1 New Features

- A new class, *OEOELigandChargeOptions*, has been added that provides options related to ligand charge assignment. The following classes now derive from *OEOELigandChargeOptions*, enabling greater flexibility in ligand charge assignment:
  - *OEProteinLigandOptOptions*
  - *OETorsionScanOptions*
  - *OEBoundEntropyOptions*
  - *OELigandEntropyOptions*
- The default optimization type has been changed to *PoseCartesian* in both *OEFixedProteinLigandOptimizer* and *OEFlexProteinLigandOptimizer*.
- The following two new constants have been added to the *OESzybkiReturnCode* namespace:
  - *AromaticRingDev*: refers to failure due to excessive aromatic distortion.
  - *OptimizerFailed*: refers to the optimizer failing to find a solution.
- The following class methods now take an optional predicate describing residues that should be considered as part of the protein target during optimization:
  - *SetProtein* in *OEFixedProteinLigandOptimizer*
  - *Optimize* in *OEFlexProteinLigandOptimizer*
  - *Energy* in *OEFlexProteinLigandOptimizer*
- The following new preliminary API classes have been added, enabling optimization of a ligand with torsion constraints:
  - *OELigandEnergy*

  - *OELigandOptOptions*
  - *OELigandOptResults*
  - *OETorConstraintOptimizer*
  - *OETorConstraintOptOptions*

6.3.2 Major Bug Fixes

- *OEFixedProteinLigandOptimizer* and *OEFlexProteinLigandOptimizer* no longer fail to optimize with the *ExcessiveOverlap* error code prematurely.

6.3.3 Minor Bug Fixes

- *OEFreeFormConfOptions* now sets the *latest Sage force field* as the default.
- An issue has been fixed in *OETorsionScan* where scanning using the option *SetUseInternalCoord* set to `true` would occasionally produce large unphysical energies at certain angles.

6.4 Szybki TK 2.8.0

6.4.1 New features

- The constant *SAGE_OPENFF* refers to the latest version of the Sage OpenFF force field.

6.5 Szybki TK 2.7.1.3

- The constant *OESz_OEForceFieldType_SAGE_OPENFF* refers to the 2.2.0 version of the Sage-OpenFF force field.

6.6 Szybki TK 2.7.1.0

- *OEFixedProteinLigandOptimizer* and *OEFlexProteinLigandOptimizer* now propagate error better when optimization fails.

6.7 Szybki TK 2.7.0

6.7.1 New features

- The ability has been added to perform optimization in internal coordinates, with fixed torsions, during a torsion scan. Accordingly, the following additional commands have been added to *OETorsionScanOptions*:
  - *SetUseInternalCoord*
  - *GetUseInternalCoord*
  - *SetTolerance*
  - *GetTolerance*
  - *SetMaxIter*
  - *GetMaxIter*
- The ability to apply a nonbonded interactions cutoff in *OEProteinLigandOptOptions* has been added. The following two methods are also available in the class:

  - *SetNonBondCutoff*
  - *GetNonBondCutoff*
- Both *OEFixedProteinLigandOptimizer* and *OEFlexProteinLigandOptimizer* now apply a nonbonded interactions cutoff of *40.0* by default.
- A new class, *OETorsionScanner*, has been added to provide greater flexibility over the existing function *OETorsionScan* in extracting intermediate results from the scanning.

6.7.2 Major bug fixes

- An issue where the overloaded method *SetTorsionConstraint* would occasionally fail has been fixed.

6.7.3 Minor bug fixes

- Validation of user-selected torsion in torsion scan has been added. If a torsion passed to function *OETorsionScan* or method *Scan* does not exist, the torsion scan code stops with a message that the torsion does not exist in the input molecule.

6.8 Szybki TK 2.6.0.1

- New constant *SAGE_OPENFF210* has been introduced that refers to the Sage 2.1.0 force field. The constant *SAGE_OPENFF* now refers to this latest version of Sage force field.

6.9 Szybki TK 2.6.0.0

6.9.1 New features

- Three classes have been added to facilitate using ff14SB/OpenFF potential for bound ligands and to bypass using OESzybki:
  - *OEBoundEntropyOptions*
  - *OELigandEntropyOptions*
  - *OEEntropyResults*
- A new function, *OEBoundLigandEntropy*, has been added to calculate the entropy of a bound ligand.
- A new function, *OELigandEntropy*, has been added to calculate the entropy of a free ligand.
- A new pair of options has been added to the *OETorsionScanOptions* to address hysteresis in torsion scanning. These provide the ability to increase the number of starts (*SetNumStarts*) and address the overlaps between scans (*SetOverlapDiv*) to avoid doing full 360-degree scans for many starts unless desired.

6.9.2 Major bug fixes

- An issue with the torsion scan (*OETorsionScan*) in solution with PB solvation has been fixed. It no longer incorrectly uses the MMFF94S potential even when scanning is done with the OpenFF force field.
- An issue with hysteresis that in certain cases caused asymmetric torsion scans has been addressed by resetting to the starting angle before starting reverse scans.

6.9.3 Minor bug fixes

- Better diagnostics have been added to *OEFixedProteinLigandOptimizer* when the optimization of a ligand in a fixed protein receptor fails due to an initial clash. An appropriate error code is now returned for such cases.
- *OEFlexProteinLigandOptimizer* now provides an appropriate error code when asked to optimize a design unit that does not contain a ligand.

6.10 Szybki TK 2.5.1.2

- Possible crashes in *OEFixedProteinLigandOptimizer_Optimize* and *OEFlexProteinLigandOptimizer_Optimize* methods have been fixed.

6.11 Szybki TK 2.5.1.1

6.11.1 New features

- Optimization of bound ligands with ff14SB/Sage and ff14SB/Parsley force fields with PB solvation forces in rigid and partially flexible receptors is now available.
- Setting the force field to *PARSLEY_OPENFF131* or *SAGE_OPENFF200* now references the most recent official versions of those two parameterizations which are OpenFF_1.3.1 and OpenFF_2.0.0 respectively. Older versions are still available from the *OEForceFieldType* namespace.
- Final ligand geometry is now checked when large initial clashes occur while using *OEFixedProteinLigandOptimizer* and *OESz_OEFlexProteinLigandOptimizer* for optimization of a ligand inside a protein receptor. In most cases, clashes are successfully relaxed; however, in some cases they can lead to a nonphysical minimum which is characterized with distorted geometry. Such results can now be detected by checking the return values of methods *OEFixedProteinLigandOptimizer_Optimize* and *OEFlexProteinLigandOptimizer_Optimize*.

6.11.2 Major bug fixes

- An issue with torsion scan for alkanes using MMFF has been fixed.
- *OEFixedProteinLigandOptimizer_Optimize* and *OEFlexProteinLigandOptimizer_Optimize* now fails properly when optimization leads to a nonphysical state of a molecule, characterized with distorted geometry. Such nonphysical optimization can sometimes happen when the starting configuration of a protein-ligand complex contains high overlaps.

6.11.3 Minor bug fixes

- A warning regarding 'invalidation of receptor' upon optimization using the *OEFlexProteinLigandOptimizer* class has been removed.

6.12 Szybki TK 2.5.0

6.12.1 New Features

- The ability to use the Sheffield Solvation Model has been added to *OEFlexProteinLigandOptimizer* and *OEFixedProteinLigandOptimizer*.
- Two additional methods have been added to the *OEProteinLigandOptOptions* class:
  - *GetSolventModel*
  - *SetSolventModel*

6.13 Szybki TK 2.4.0

Fall 2021

6.13.1 New features

- A new function, *OEGetFreeFormError*, is provided that returns error messages corresponding to return codes defined in *OEFreeFormReturnCode* namespace.
- The latest version of the Open Force Field Initiative, Sage 2.0.0, has been added to *OEForceFieldType* as built-in force field *OEForceFieldType.SAGE_OPENFF200*.
- Freeform for estimating free energy of conformations in the ensemble, *OEFreeFormConf*, now defaults to using the most recent version of the Open Force Field (OESage). Previously it was using *MMFF94S* by default.
- Protein-ligand optimization classes *OEFixedProteinLigandOptimizer* and *OEFlexProteinLigandOptimizer* now defaults to using the OEFF14SBSageComplex force field.

6.13.2 Minor bug fixes

- *OETorsionScanOptions.SetForceField* now properly defaults to *OEForceFieldType.MMFF94S*.

6.14 Szybki TK 2.3.1

July 2021

6.14.1 New features

- The latest version of the Open Force Field Initiative, Parsley 1.3.1, has been added to the *OEForceFieldType* namespace as built-in force field *PARSLEY_OPENFF131*.
- Support for user-defined force fields has been added to the *OETorsionScan* function. Two new API points, *SetForceField* and *GetForceField*, have been added for setting and getting this value.
- Support for user defined dihedral angles has been added to *OETorsionScan*. Two new API points, *SetUserDefinedAngles* and *GetUserDefinedAngles*, have been added for setting and getting this value.

6.15 Szybki TK 2.3.0

Fall 2020

6.15.1 New features

- A new SMIRNOFF force field from the Open Force Field Initiative, Parsley 1.2.1, has been added as a built-in force field. It is defined as *PARSLEY_OPENFF121*.
- A protein or part of a protein can now be optimized with the FF14SB force field. Cofactors with the exception if HEME are handled by the OpenFF/Parsley force field (for example Parsley 1.2.1), while parameters for HEME are taken from Bryce Group.
- The following preliminary classes have been added that enable optimizing protein-ligand complexes, with or without partial flexibility of the proteins, using the new OEFF14SBParsleyComplex force fields, among others:
  - *OEFixedProteinLigandOptimizer*
  - *OEFlexProteinLigandOptimizer*

- A new preliminary class, *OEProteinLigandOptResults*, has been added that represents results of a protein-ligand complex optimization.
- Two new preliminary classes, *OEProteinLigandOptOptions* and *OEProteinFlexOptions*, have been added that provide choices for protein-ligand optimization and protein flexibility.
- A new namespace, *OESzybkiReturnCode*, and a corresponding function, *OEGetSzybkiError*, have been added that correspond to return codes from the new optimizer classes.

6.15.2 Minor bug fixes

- Protein-ligand optimization with OpenFF/Parsley force fields using *OESzybki* no longer fails, even when the initial gradient norm is 1e^15 kcal/(mol A) or larger.

6.16 Szybki TK 2.2.0

6.16.1 New features

- Two new methods, *OESzybkiGeneralOptions.SetForceField* and *OESzybkiGeneralOptions.GetForceField*, have been added that provide support for custom force fields in **Szybki TK**. These methods provide users with the ability to use SMIRNOFF force fields that are not built into **Szybki TK**.
- Two new SMIRNOFF force fields from the Open Force Field Initiative, namely `Parsley (OpenFF-1.0.0)` and `Parsley (OpenFF-1.1.1)`, have been added as built-in force fields. They are defined as *OEForceFieldType.PARSLEY_OPENFF100* and *OEForceFieldType.PARSLEY_OPENFF111*, respectively.
- Potential energy terms in the namespace *OEPotentialTerms* are now independent of the force field names. Exceptions are combined force fields in which intramolecular terms are taken from MMFF94 as well as intermolecular terms from either AMBER or OEFF.

6.16.2 Major bug fixes

- A bug in the entropy calculation for a ligand inside a receptor binding site has been fixed.

6.16.3 Minor bug fixes

- Conformer free energy calculation using *OEFreeFormConf.EstimateFreeEnergies* now produces results identical to performing the calculation in steps using the methods from *OEFreeFormConfAdvanced*.
- The following methods now return `bool`, instead of a `void`:
  - *OESzybkiGeneralOptions.SetCalculateGradients*
  - *OESzybkiGeneralOptions.SetIEFFCluster*
  - *OESzybkiGeneralOptions.SetIntramolecularVdWCutoff*
  - *OESzybkiGeneralOptions.SetLigandRMSDHeavy*
  - *OESzybkiGeneralOptions.SetRemoveAttractiveVdWForces*
  - *OESzybkiGeneralOptions.SetRemoveCoulombTerms*
  - *OESzybkiGeneralOptions.SetSoluteDielectric*
  - *OESzybkiGeneralOptions.SetTemperature*
  - *OESzybkiGeneralOptions.SetVerbose*
  - *OESzybkiOptOptions.SetCalculateFrozenTerms*
  - *OESzybkiOptOptions.SetGradTolerance*

  - *OESzybkiOptOptions.SetMaxIter*
  - *OESzybkiSolventOptions.SetAtomicRadiiType*
  - *OESzybkiSolventOptions.SetCavitySolventParameter*
  - *OESzybkiSolventOptions.SetSaltConcentration*
  - *OESzybkiSolventOptions.SetSolventDielectric*
  - *OESzybkiProteinOptions.SetExactVdWProteinLigand*
  - *OESzybkiProteinOptions.SetIntermolecularVdWCutoff*
  - *OESzybkiProteinOptions.SetProteinDielectric*
  - *OESzybkiProteinOptions.SetProteinFlexibilityRange*
  - *OEFreeFormConfOptions.SetUseInputEnsemble*
  - *OEFreeFormConfOptions.SetUseSolventCorr*
  - *OEFreeFormSolvOptions.SetAtomicRadiiType*
  - *OEFreeFormSolvOptions.SetChargeType*
  - *OEFreeFormSolvOptions.SetConfGenRMS*
  - *OEFreeFormSolvOptions.SetIonicState*
  - *OEFreeFormSolvOptions.SetMaxConfGen*
  - *OEFreeFormSolvOptions.SetUseInput3D*
  - *OEFreeFormSolvOptions.SetUseInputEnsemble*
- `OEForceFieldType::SMIRNOFF` has been renamed *OEForceFieldType.SMIRNOFF99FROSST* for clarity.
- Perceiving residues in the input proteins are now done only when the input protein does not have them.
- **Szybki TK** now properly handles a hard clash between ligand and protein atoms when a SMIRNOFF force field is used to optimize a ligand inside a receptor (with a gradient norm larger than 1e15).

6.17 Szybki TK 2.1.0

6.17.1 New features

- *OESzybki.SetProtein* has been updated to return a Boolean, allowing failures to be handled in code.
- **Szybki** can now perform energy optimization using a SMIRNOFF force field (https://openforcefield.org/).

6.17.2 Minor bug fixes

- The default `EnergyWindow` for conformer generation in *OEFreeFormConfOptions* is now consistent with that in OEOmegaOptions in the `OEOmegaSampling.Dense` mode.
- Two methods in *OEFreeFormConfResults*, *PrintTxt* and *PrintCsv*, have been modified to widen the scope of the associated streams.

6.17.3 Documentation changes

- A new example program, *Optimization of a bound ligand in a partially flexible receptor*, has been added.
- A new example program, *Single ligand in vacuum using SMIRNOFF*, has been added.

6.18 Szybki TK 2.0.4

6.18.1 Major bug fixes

- *OEEstimateSolvFreeEnergy* has been fixed to properly calculate solvation energy when used with input molecule charges.

6.18.2 Minor bug fixes

- *OETorsionScan* no longer requires an `am1bcc` license when *OETorsionScanOptions.SetUseSinglePointPB* is set to `true`.

6.19 Szybki TK 2.0.3

6.19.1 New features

- New *OETorsionScanOptions.SetPenaltyForceConstant* and *OETorsionScanOptions.GetPenaltyForceConstant* methods have been added.

6.19.2 Documentation changes

- The example code in the *Estimation of solution ligand entropy* section has been adjusted for solution environment by using the *OESolventModel.Sheffield* constant.

6.20 Szybki TK 2.0.2

6.20.1 New features

- A new method, *OESzybkiSolventOptions.SetUseCurrentCharges*, has been added as a convenience function.
- Entropy calculation of a bound ligand can now be computed when an IEFF force field is selected (*OEForceFieldType.MMFF_IEFF* or *OEForceFieldType.MMFFS_IEFF*). It uses an accurate numerical Hessian.

6.20.2 Minor bug fixes

- An issue related to **Szybki TK** losing information about already perceived protein residues has been fixed.
- Previously, highly clashed protein-ligand systems that were often a result of docking were not automatically relaxed when using the default *OEOptType.BFGS* optimization method. Users had to explicitly use the *OEOptType.SD_BFGS* method. Now, a clash that exceeds 1e6 (kcal/mol)/A on a gradient norm is detected and 5 steps of steepest descent optimization are automatically applied before the BFGS optimizer is run.

6.21 Szybki TK 2.0.1

6.21.1 Minor bug fixes

- A crash that occurred when adding harmonic constraint with MMFF_IEFF has been fixed.
- A problem where the *OESzybkiSolventOptions.SetChargeEngine* option was not obeyed if using *OESzybki.SetProtein* has been fixed.

6.22 Szybki TK 2.0.0

6.22.1 Minor bug fixes

- The *OESzybki.GetEntropy* method now checks for consistency between the input argument for environment type and the already defined solvent method in *OESzybkiOptions*. This check ensures that the wrong entropy calculation is not silently performed.

6.22.2 Documentation changes

- The Advanced API section has been removed. The low-level functionality offered through the advanced API has been improved and is now offered as a new toolkit named **OEFF TK**. The documentation is available at OEFF toolkits.

6.23 Szybki TK 1.9.2

6.23.1 New features

- Two new classes, *OEFreeFormConf* and *OEFreeFormConfAdvanced*, have been added to provide conformer free energy functionality. The methods in *OEFreeFormConf* provide high-level functionality; those in *OEFreeFormConfAdvanced* provide finer control of the functionality.

  In general, the new conformer free energy functionality provides 2 new capabilities: (1) methods to estimate restriction free energy and (2) methods to find conformers similar to the one specified from a pool of minimum energy conformers.

- The *OEFreeFormConfOptions* API methods have been modified and enhanced with OEOmegaOptions and various *OESzybkiOptions* sub-options. The enhanced API allows greater control over the details of free energy calculation.

- The *OESzybkiOptions* class has been simplified and now consists of various sub-options: *OESzybkiGeneralOptions*, *OESzybkiOptOptions*, *OESzybkiSolventOptions*, and *OESzybkiProteinOptions*. The specific methods to set or get option values are now provided through these sub-option classes.

  The following table shows the older, deprecated option functions and their new sub-option replacements:

Deprecated Option Method	New Sub-option Method
`OESz::OESzybki::SetIEFFCluster`	*OESzybkiGeneralOptions.SetIEFFCluster*
`OESz::OESzybkiOptions::SetIEFFCluster`	*OESzybkiGeneralOptions.SetIEFFCluster*
`OESz::OESzybki::GetIEFFCluster`	*OESzybkiGeneralOptions.GetIEFFCluster*
`OESz::OESzybkiOptions::GetIEFFCluster`	*OESzybkiGeneralOptions.GetIEFFCluster*
`OESz::OESzybki::SetRemoveCoulombTerms`	*OESzybkiGeneralOptions.SetRemoveCoulombTerms*
`OESz::OESzybkiOptions::SetRemoveCoulombTe`	*OESzybkiGeneralOptions.SetRemoveCoulombTerms*
`OESz::OESzybki::GetRemoveCoulombTerms`	*OESzybkiGeneralOptions.GetRemoveCoulombTerms*
`OESz::OESzybkiOptions::GetRemoveCoulombTe`	*OESzybkiGeneralOptions.GetRemoveCoulombTerms*
`OESz::OESzybki::SetRemoveAttractiveVdWFor`	*OESzybkiGeneralOptions.SetRemoveAttractiveVdWForces*
`OESz::OESzybkiOptions::SetRemoveAttractiv`	*OESzybkiGeneralOptions.SetRemoveAttractiveVdWForces*

continues on next page

Table 1 – continued from previous page

OESz::OESzybki::GetRemoveAttractiveVdWFor	*OESzybkiGeneralOptions.GetRemoveAttractiveVdWForces*
OESz::OESzybkiOptions::GetRemoveAttractiv	*OESzybkiGeneralOptions.GetRemoveAttractiveVdWForces*
OESz::OESzybki::SetLigandRMSDHeavy	*OESzybkiGeneralOptions.SetLigandRMSDHeavy*
OESz::OESzybkiOptions::SetLigandRMSDHeavy	*OESzybkiGeneralOptions.SetLigandRMSDHeavy*
OESz::OESzybki::GetLigandRMSDHeavy	*OESzybkiGeneralOptions.GetLigandRMSDHeavy*
OESz::OESzybkiOptions::GetLigandRMSDHeavy	*OESzybkiGeneralOptions.GetLigandRMSDHeavy*
OESz::OESzybki::SetCalculateGradients	*OESzybkiGeneralOptions.SetCalculateGradients*
OESz::OESzybkiOptions::SetCalculateGradie	*OESzybkiGeneralOptions.SetCalculateGradients*
OESz::OESzybki::GetCalculateGradients	*OESzybkiGeneralOptions.GetCalculateGradients*
OESz::OESzybkiOptions::GetCalculateGradie	*OESzybkiGeneralOptions.GetCalculateGradients*
OESz::OESzybki::SetVerbose	*OESzybkiGeneralOptions.SetVerbose*
OESz::OESzybkiOptions::SetVerbose	*OESzybkiGeneralOptions.SetVerbose*
OESz::OESzybki::GetVerbose	*OESzybkiGeneralOptions.GetVerbose*
OESz::OESzybkiOptions::GetVerbose	*OESzybkiGeneralOptions.GetVerbose*
OESz::OESzybki::SetIntramolecularVdWCutof	*OESzybkiGeneralOptions.SetIntramolecularVdWCutoff*
OESz::OESzybkiOptions::SetIntramolecularV	*OESzybkiGeneralOptions.SetIntramolecularVdWCutoff*
OESz::OESzybki::GetIntramolecularVdWCutof	*OESzybkiGeneralOptions.GetIntramolecularVdWCutoff*
OESz::OESzybkiOptions::GetIntramolecularV	*OESzybkiGeneralOptions.GetIntramolecularVdWCutoff*
OESz::OESzybki::SetTemperature	*OESzybkiGeneralOptions.SetTemperature*
OESz::OESzybkiOptions::SetTemperature	*OESzybkiGeneralOptions.SetTemperature*
OESz::OESzybki::GetTemperature	*OESzybkiGeneralOptions.GetTemperature*
OESz::OESzybkiOptions::GetTemperature	*OESzybkiGeneralOptions.GetTemperature*
OESz::OESzybki::SetSoluteDielectric	*OESzybkiGeneralOptions.SetSoluteDielectric*
OESz::OESzybkiOptions::SetSoluteDielectri	*OESzybkiGeneralOptions.SetSoluteDielectric*
OESz::OESzybki::GetSoluteDielectric	*OESzybkiGeneralOptions.GetSoluteDielectric*
OESz::OESzybkiOptions::GetSoluteDielectri	*OESzybkiGeneralOptions.GetSoluteDielectric*
OESz::OESzybki::SetForceFieldType	*OESzybkiGeneralOptions.SetForceFieldType*
OESz::OESzybkiOptions::SetForceFieldType	*OESzybkiGeneralOptions.SetForceFieldType*
OESz::OESzybki::GetForceFieldType	*OESzybkiGeneralOptions.GetForceFieldType*
OESz::OESzybkiOptions::GetForceFieldType	*OESzybkiGeneralOptions.GetForceFieldType*
OESz::OESzybki::SetCalculateFrozenTerms	*OESzybkiOptOptions.SetCalculateFrozenTerms*
OESz::OESzybkiOptions::SetCalculateFrozen	*OESzybkiOptOptions.SetCalculateFrozenTerms*

continues on next page

Table 1 – continued from previous page

OESz::OESzybki::GetCalculateFrozenTerms	*OESzybkiOptOptions.GetCalculateFrozenTerms*
OESz::OESzybkiOptions::GetCalculateFrozen'	*OESzybkiOptOptions.GetCalculateFrozenTerms*
OESz::OESzybki::SetGradTolerance	*OESzybkiOptOptions.SetGradTolerance*
OESz::OESzybkiOptions::SetGradTolerance	*OESzybkiOptOptions.SetGradTolerance*
OESz::OESzybki::GetGradTolerance	*OESzybkiOptOptions.GetGradTolerance*
OESz::OESzybkiOptions::GetGradTolerance	*OESzybkiOptOptions.GetGradTolerance*
OESz::OESzybki::SetOptimizerType	*OESzybkiOptOptions.SetOptimizerType*
OESz::OESzybkiOptions::SetOptimizerType	*OESzybkiOptOptions.SetOptimizerType*
OESz::OESzybki::GetOptimizerType	*OESzybkiOptOptions.GetOptimizerType*
OESz::OESzybkiOptions::GetOptimizerType	*OESzybkiOptOptions.GetOptimizerType*
OESz::OESzybki::SetMaxIter	*OESzybkiOptOptions.SetMaxIter*
OESz::OESzybkiOptions::SetMaxIter	*OESzybkiOptOptions.SetMaxIter*
OESz::OESzybki::GetMaxIter	*OESzybkiOptOptions.GetMaxIter*
OESz::OESzybkiOptions::GetMaxIter	*OESzybkiOptOptions.GetMaxIter*
OESz::OESzybki::SetSolventDielectric	*OESzybkiSolventOptions.SetSolventDielectric*
OESz::OESzybkiOptions::SetSolventDielectr	*OESzybkiSolventOptions.SetSolventDielectric*
OESz::OESzybki::GetSolventDielectric	*OESzybkiSolventOptions.GetSolventDielectric*
OESz::OESzybkiOptions::GetSolventDielectr	*OESzybkiSolventOptions.GetSolventDielectric*
OESz::OESzybki::SetSaltConcentration	*OESzybkiSolventOptions.SetSaltConcentration*
OESz::OESzybkiOptions::SetSaltConcentrati	*OESzybkiSolventOptions.SetSaltConcentration*
OESz::OESzybki::GetSaltConcentration	*OESzybkiSolventOptions.GetSaltConcentration*
OESz::OESzybkiOptions::GetSaltConcentrati	*OESzybkiSolventOptions.GetSaltConcentration*
OESz::OESzybki::SetCavitySolventParameter	*OESzybkiSolventOptions.SetCavitySolventParameter*
OESz::OESzybkiOptions::SetCavitySolventPa	*OESzybkiSolventOptions.SetCavitySolventParameter*
OESz::OESzybki::GetCavitySolventParameter	*OESzybkiSolventOptions.GetCavitySolventParameter*
OESz::OESzybkiOptions::GetCavitySolventPa	*OESzybkiSolventOptions.GetCavitySolventParameter*
OESz::OESzybki::SetSolventModel	*OESzybkiSolventOptions.SetSolventModel*
OESz::OESzybkiOptions::SetSolventModel	*OESzybkiSolventOptions.SetSolventModel*
OESz::OESzybki::GetSolventModel	*OESzybkiSolventOptions.GetSolventModel*
OESz::OESzybkiOptions::GetSolventModel	*OESzybkiSolventOptions.GetSolventModel*
OESz::OESzybki::SetAtomicRadii	*OESzybkiSolventOptions.SetAtomicRadiiType*
OESz::OESzybkiOptions::SetAtomicRadii	*OESzybkiSolventOptions.SetAtomicRadiiType*
OESz::OESzybki::GetAtomicRadiiType	*OESzybkiSolventOptions.GetAtomicRadiiType*
OESz::OESzybkiOptions::GetAtomicRadiiType	*OESzybkiSolventOptions.GetAtomicRadiiType*

continues on next page

Table 1 – continued from previous page

OESz::OESzybki::SetUseCurrentCharges	*OESzybkiSolventOptions.SetChargeEngine*
OESz::OESzybkiOptions::SetUseCurrentCharg	*OESzybkiSolventOptions.SetChargeEngine*
OESz::OESzybki::GetUseCurrentCharges	*OESzybkiSolventOptions.SetChargeEngine*
OESz::OESzybkiOptions::GetUseCurrentCharg	*OESzybkiSolventOptions.SetChargeEngine*
OESz::OESzybki::SetEveryConfAM1BCCCharges	*OESzybkiSolventOptions.SetChargeEngine*
OESz::OESzybkiOptions::SetEveryConfAM1BCC	*OESzybkiSolventOptions.SetChargeEngine*
OESz::OESzybki::GetEveryConfAM1BCCCharges	*OESzybkiSolventOptions.SetChargeEngine*
OESz::OESzybkiOptions::GetEveryConfAM1BCC	*OESzybkiSolventOptions.SetChargeEngine*
OESz::OESzybki::SetExactVdWProteinLigand	*OESzybkiProteinOptions.SetExactVdWProteinLigand*
OESz::OESzybkiOptions::SetExactVdWProtein	*OESzybkiProteinOptions.SetExactVdWProteinLigand*
OESz::OESzybki::GetExactVdWProteinLigand	*OESzybkiProteinOptions.GetExactVdWProteinLigand*
OESz::OESzybkiOptions::GetExactVdWProtein	*OESzybkiProteinOptions.GetExactVdWProteinLigand*
OESz::OESzybki::SetProteinDielectric	*OESzybkiProteinOptions.SetProteinDielectric*
OESz::OESzybkiOptions::SetProteinDielectr	*OESzybkiProteinOptions.SetProteinDielectric*
OESz::OESzybki::GetProteinDielectric	*OESzybkiProteinOptions.GetProteinDielectric*
OESz::OESzybkiOptions::GetProteinDielectr	*OESzybkiProteinOptions.GetProteinDielectric*
OESz::OESzybki::SetProteinFlexibilityRang	*OESzybkiProteinOptions.SetProteinFlexibilityRange*
OESz::OESzybkiOptions::SetProteinFlexibil	*OESzybkiProteinOptions.SetProteinFlexibilityRange*
OESz::OESzybki::GetProteinFlexibilityRang	*OESzybkiProteinOptions.GetProteinFlexibilityRange*
OESz::OESzybkiOptions::GetProteinFlexibil	*OESzybkiProteinOptions.GetProteinFlexibilityRange*
OESz::OESzybki::SetIntermolecularVdWCutof	*OESzybkiProteinOptions.SetIntermolecularVdWCutoff*
OESz::OESzybkiOptions::SetIntermolecularV	*OESzybkiProteinOptions.SetIntermolecularVdWCutoff*
OESz::OESzybki::GetIntermolecularVdWCutof	*OESzybkiProteinOptions.GetIntermolecularVdWCutoff*
OESz::OESzybkiOptions::GetIntermolecularV	*OESzybkiProteinOptions.GetIntermolecularVdWCutoff*
OESz::OESzybki::SetProteinElectrostaticMo	*OESzybkiProteinOptions.SetProteinElectrostaticModel*
OESz::OESzybkiOptions::SetProteinElectros	*OESzybkiProteinOptions.SetProteinElectrostaticModel*
OESz::OESzybki::GetProteinElectrostaticMo	*OESzybkiProteinOptions.GetProteinElectrostaticModel*
OESz::OESzybkiOptions::GetProteinElectros	*OESzybkiProteinOptions.GetProteinElectrostaticModel*
OESz::OESzybki::SetProteinFlexibilityType	*OESzybkiProteinOptions.SetProteinFlexibilityType*
OESz::OESzybkiOptions::SetProteinFlexibil	*OESzybkiProteinOptions.SetProteinFlexibilityType*

continues on next page

Table 1 – continued from previous page

OESz::OESzybki::GetProteinFlexibilityType	*OESzybkiProteinOptions.GetProteinFlexibilityType*
OESz::OESzybkiOptions::GetProteinFlexibil	*OESzybkiProteinOptions.GetProteinFlexibilityType*

6.23.2 Minor bug fixes

- User-defined partial charges set for the MMFFS_AMBER or MMFF_AMBER force field calculations had previously not been preserved. They are now passed to the optimized molecule.
- Changing the protein flexibility type from range-defined to a list of residues for an *OESzybkiOptions* object previously required creating a copy of the object. This issue has been fixed.
- Atomic radii for IEFF/PB calculations were previously restricted to the Bondi type. Although the default is still Bondi, ZAP7 and ZAP9 radii are now available.

6.24 Szybki TK 1.9.1

6.24.1 New features

- Zero point energy is now included in the free energy estimation of conformations.
- The accuracy of the solvation energy calculated in low dielectric solvents (< 20) has been increased by doubling the size of the ZAP grid used in the calculation (from 4 to 8 Angstroms).

6.24.2 Minor bug fixes

- The API for the method `OESz::OESzybkiOptions::SetForceFieldType` has been changed. Previously, the method returned `void`. Now it validates the input parameter and returns `true` if the force field selection is valid and `false` otherwise.
- When performing a constrained minimization using the *OEProteinElectrostatics.SolventPBForces* electrostatic model, a clash between a protein and a ligand where the square distance exceeds 0.1^2 Angstrom^2 will be detected and a warning will be issued.

6.24.3 Documentation fixes

- Units of the returned values for the methods *OESzybkiResults.GetVibEntropy* and *OESzybkiResults.GetRotEntropy* had been incorrectly described as J/(mol K). In fact, both methods return entropy in cal/(mol K). This error has been fixed.

6.25 Szybki TK 1.9.0

6.25.1 New features

- Analytical second order derivatives in AMBER intermolecular interactions have been added. This new feature allows fast Newton optimization when *OEForceFieldType.MMFF_AMBER* or *OEForceFieldType.MMFFS_AMBER* force field is selected with method `OESz::OESzybkiOptions::SetForceFieldType` and a variable *OEOptType.NEWTON* is passed to `OESz::OESzybkiOptions::SetOptimizerType` method.
- APIs for torsion scanning is available, including free function *OETorsionScan* and classes *OETorsionScanOptions* and *OETorsionScanResult*.

6.25.2 Minor bug fixes

- API for method `OESz::OESzybkiOptions::SetIEFFCluster` has been changed. Previously, the method had returned `void`; it now returns `bool`.

6.26 Szybki TK 1.8.7

6.26.1 New features

- *OESzybkiEnsembleResults.HasEntropy* has been added. This new method returns `false` whenever a requested entropy calculation fails.
- The default behavior of protein-ligand systems handled with MMFF or MMFF_AMBER force fields has changed from *OEProteinElectrostatics.NoElectrostatics* to *OEProteinElectrostatics.ExactCoulomb*. In other words, Coulomb protein-ligand interactions are included by default. Beginning with this release, removing Coulomb terms from the protein-ligand interactions requires using the `OESz::OESzybkiOptions::SetProteinElectrostaticModel` method with the *OEProteinElectrostatics.NoElectrostatics* argument.
- Previously, the combination of IEFF force field and PB desolvation was restricted to using ESP charges calculated along with atomic multipoles. This restriction has been removed and IEFF for protein-ligand interaction can now be used together with the method `OESz::OESzybkiOptions::SetUseCurrentCharges`.

6.26.2 Major bug fixes

- Return values of the methods `OESz::OESzybkiOptions::SetExactVdWProteinLigand` and `OESz::OESzybkiOptions::SetProteinFlexibilityRange` have been changed from `void` to `bool`. This change allows for verifying whether the options have been successfully set.
- When IEFF force field is selected, implying that molecular input files with pre-calculated atomic multipoles have been used, the addition of explicit hydrogens is now disabled. In very rare cases (for example, tetracyanoquinodimethane ligand), perception of bonding might not correspond to the actual ab-initio pre-calculated structure, causing the IEFF calculated energies to be incorrect.
- Using external charges (see method `OESz::OESzybkiOptions::SetUseCurrentCharges`) in the *OEForceFieldType.MMFF_AMBER* force field for protein-ligand Coulomb interactions was failing unless the protein was eligible for correct AMBER charging. This unnecessary condition has been removed: any atomic preassigned atomic charges can now be used in combination with AMBER vdW protein-ligand interactions.
- Entropy calculation at very low temperatures is now handled properly: at 0 Kelvin, a value of zero is always returned. At temperatures close to 0 Kelvin, calculation is attempted and a warning is issued if the calculation fails. *OESzybkiEnsembleResults.HasEntropy* (see above) might be used to query if the *OESzybkiResults* object contains valid calculated entropy.
- Using IEFF/PB potential for bound ligand optimization in torsion space when the ligand receptor is partially flexible did not always work. This issue has been fixed.

6.26.3 Minor bug fixes

- When attempting to use Newton optimization and IEFF force field results, an error is now thrown with the message "Second derivatives not available for IEFF ff".

6.27 Szybki TK 1.8.6

6.27.1 Major bug fixes

- A memory issue that caused periodic crash of JVM in a situation when torsion optimization was performed and Coulomb terms were removed from the MMFF94 or MMFF94S force field in Java-wrapped code has been fixed.
- Possible memory corruption in the IEFF force field has been fixed.
- Previously, making the protein partially flexible and selecting a run type of either *OERunType.SinglePoint* or *OERunType.TorsionsOpt* had been causing crashes. These bugs have both been fixed.
- The order of the residue names generated in the logging output for flexible polar hydrogens for the Protein-Ligand IEFF-MMFF calculation will now be consistent between runs. Previously, the order of the residue names could change in subsequent runs.
- A discontinuity in the flat-bottom harmonic potential has been fixed. The potential function generated by the method *OESzybki.SetHarmonicConstraints* had a discontinuity in the primary function and its first derivative that caused the optimizer to over-constrain optimizations with harmonic constraints with flat-bottom. The class OEHarmonicPotential has been reimplemented.

6.27.2 Minor bug fixes

- Language in the warning regarding the input of atoms with unspecified chirality is improved. It now reads: `Input molecule contains chiral atoms with unspecified chirality`. This warning may be generated when using the FreeForm part of the **Szybki TK**.
- *OEEstimateConfFreeEnergies* now accepts an input ensemble containing a molecule with 20 or more rotatable bonds. Previously, encountering such a molecule resulted in a warning and then that molecule was skipped.

6.27.3 Documentation fixes

- Documentation for constants describing optimization in torsional coordinates and translational-rotational coordinates (*OERunType.TorsionsOpt* and *OERunType.SolidBodyOpt* has been expanded. Specifically, short descriptions of the terms `torsional coordinates`, and `solid-body optimization` have been added.
- Missing documentation for the *OEOptType.NEWTON* constant, which sets Newton type optimization, has been added.

6.28 Szybki TK 1.8.5

6.28.1 New features

- *OEForceFieldType.MMFF_IEFF* and *OEForceFieldType.MMFFS_IEFF* protein-ligand potentials can be now combined with the PB solvent energetics and forces calculated with ESP partial charges that are read from the same molecular input as atomic multipoles.
- The upper limit of the ligand size that might be charged with the AM1BCC partial charges when protein-ligand potential is set using *OEForceFieldType.MMFF_AMBER* or *OEForceFieldType.MMFFS_AMBER* types has increased from 100 to 150 heavy atoms.
- `OESetForceFieldDummyAtom` can be used to ignore dummy atoms in forcefield calculations if only van der Waals and Coulombic components are included. Dummy atoms are ignored in MMFF force fields and cannot be used in intramolecular components.
- *OESzybkiResults.Clear* has been added to reset the state of the *OESzybkiResults* object.

6.28.2 Major bug fixes

- *OESzybki* will no longer crash when `OESz::OESzybkiOptions::SetForceFieldType` is called with the argument *OEForceFieldType.MMFFS_AMBER* and then followed by a call of `OESz::OESzybkiOptions::SetUseCurrentCharges` with `true`.

6.28.3 Minor bug fixes

- *OESzybki.operator()* will no longer lose SD data when operating on a single conformer OEMolBase.
- *OESzybkiResults* is now cleared from whenever *OESzybki.operator()* fails. Clearing is done by the new method *OESzybkiResults.Clear*. A cleared *OESzybkiResults* will make *OESzybkiResults.Print* output: `OESzybkiResults object does not have data to report`.
- `OESz::OESzybki::GetForceFieldType` now properly returns *OEForceFieldType.MMFF_AMBER* or *OEForceFieldType.MMFF_IEFF*. Previously, they could only be returned after a calculation was performed with the *OESzybki* object.
- A CYS residue bonded by the sulfur to something other than another CYS residue is no longer treated as anionic by the AMBER charging system.
- Final geometry of a ligand optimized with the MMFF_IEFF or MMFFS_IEFF potential has been updated with the transformed multipoles attached to atoms as generic data.
- A potential memory leak when performing repeated protein-ligand minimizations has been fixed.

6.28.4 Documentation fixes

- `OEFuncType` has been documented.
- The *Optimization of all conformers of a ligand* section has a new example that demonstrates how to minimize all conformers in a ligand.

6.29 Szybki TK 1.8.4

6.29.1 New features

- A new *OESingleConfResult.GetRelLnQ* method has been added for retrieving the logarithm of the relative vibrational-rotational partition function with respect to the minimum free-energy conformation.

6.29.2 Major bug fixes

- Previously, when protein-ligand systems where ligands were made of two or more molecules (for example, an inhibitor + water molecule), interligand IEFF interaction energy was not correctly calculated in two cases: (1) for single point calculations and (2) for solid-body ligand pose optimization. In both cases, the total protein-ligand interaction energy was not affected. In the second case, optimization itself was not affected. This issue has been fixed.

6.29.3 Minor bug fixes

- Internal refactoring has been performed to improve stability.
- Memory leaks related to protein-ligand interaction and handling protein flexibility have been fixed.

6.29.4 API Changes

- The unused `OEMMFFSubsetVdw` class has been removed.

6.29.5 Documentation fixes

- Cross-references have been fixed in several sections.

6.29.6 Java- and C#-specific changes

- The following examples were added for Java and C#:
  - *Ligand in vacuum optimization*
  - *Optimize multiple ligands in Sheffield solvation model*
  - *Partially flexible protein optimization*
  - *Partially flexible protein with Newton-Raphson optimization*
  - *FreeForm Solvation*
  - *FreeForm Conformer*
  - *Entropy of a bound ligand*
  - *Bound ligand optimization*
  - *Rigid receptor optimization*
  - *Newton-Raphson optimization*
  - *Bound and solution entropy*
  - *PB Protein-ligand binding*

6.30 Szybki TK 1.8.3

6.30.1 New features

- New forcefield for ligand-ligand and protein-ligand interactions has been added. It is a combination of MMFF94S (or MMFF94) with intermolecular potential called IEFF developed at OpenEye. In this combined force field, MMFF94S is used to describe the intramolecular interactions while IEFF which has 3 components (vdW, Coulomb and Fermi repulsion) handles intermolecular interactions. Coulomb terms in IEFF require precalculated and assigned atomic multipoles to ligand and protein molecules. This task can be achieved with two OpenEye codes called MOLQM and PROTQM. IEFF has been described in the following publication:

```
N. Hamaguchi, L. Fusti-Molnar and S. Wlodek
Force-field and quantum-mechanical binding study of selected SAMPL3 host-guest␣
↪complexes
J. Comput. Aided Mol. Des. Vol. 26, pp. 577-582, 2012
```

  Please contact support@eyesopen.com if you are interested in testing this new forcefield.

- New overloaded method *`OESzybki.SetTorsionConstraint`* has been added to the public API. It allows selectively constraining a specific torsion in the optimized molecule.

- A set of flexible residues in ligand optimization with partially flexible protein is no longer restricted to the specified distance from the ligand. This goal has been achieved by the expansion of the *`OEProtFlex`* namespace for new constants which might be passed to the `OESz::OESzybkiOptions::SetProteinFlexibilityType` method, and the addition of the new method `OESz::OESzybkiOptions::AddFlexibleResidue` which can be used multiple times to make a set of residues flexible.

- Added new virtual method `OEOpt::OEFunc1::IsRedundant`. It allows to check if two formally numerically different sets of coordinates indeed represent two different points in the coordinate space. This method has no trivial implementation in `OEQuatAdaptor.IsRedundant`.
- MMFF calculations now allow for a "DUMMY" atom type, `OEMMFFType.DUMMY` to specify that an atom should be avoided during an MMFF energy calculation.
- Added a DUMMY atom type to the AMBER force field to specify that an atom should be avoided during an AMBER energy calculation.

6.30.2 Major bug fixes

- Fixed a rare crash when using the AMBER force field on some platforms.
- A combination of Newton optimization with the use of the following two methods: `OESz::OESzybkiOptions::SetProteinElectrostaticModel` and `OESz::OESzybkiOptions::SetExactVdWProteinLigand` with parameters *OEProteinElectrostatics.ExactCoulomb* and `true` respectively caused that Coulomb potential was missing from the intermolecular interaction. That bug has been fixed.
- *OESzybkiResults.GetLnQvib* and *OESingleConfResult.GetLnQvibrational* will no longer return negative infinity on Windows for large ligands.

6.30.3 Minor bug fixes

- Fixed the name of the saved electrostatic grid file when the name of the molecule is used as part of the file name, by replacing all non alpha-numeric characters with an underscore character.
- An attempt to use Newton optimization with a potential for which second derivatives are not implemented is terminated with a proper error message.
- When AM1BCC charges are selected in Sheffield or PB solvation models for multiply charged species, SCF procedure in AM1 might not converge within 20 cycles (this has been observed for methotrexate only so far). To make sure that the electron density is converged, and calculated atomic charges stable, we allow up to 1000 SCF iterations for multiply charged molecules.
- Return value of the method *OESzybkiResults.GetTotalEnergy* for protein-ligand system with partially flexible protein does not contain intrinsic MMFF energy of the protein.
- An attempt to use protein flexibility type which is not a member of the *OEProtFlex* namespace in the method `OESz::OESzybkiOptions::SetProteinFlexibilityType` issues an error message ``Unrecognized protein flexibility`.
- An error message generated upon the attempt of using partially flexible protein in solid body optimization type is generated instead of warning.
- A more informative error message is generated upon the attempt of free ligand entropy estimation in the situation when protein environment is set.
- An error message is generated when trying to use Newton optimization for protein-ligand systems with protein electrostatic models different than *OEProteinElectrostatics.NoElectrostatics* or *OEProteinElectrostatics.ExactCoulomb*
- Flexible polar hydrogens are printed when residue polar hydrogens are asked to be optimized along with the ligand.

6.30.4 Documentation fixes

- Documentation for the `OEMolMMFF::OEMMFFSubsetVdw` class has been removed.

6.31 Szybki TK 1.8.2

6.31.1 New features

- *OEEstimateSolvFreeEnergy* and *OEEstimateConfFreeEnergies* added to perform FreeForm calculations. Their usage requires passing instances of the following classes:
  - *OEFreeFormSolvOptions*
  - *OEFreeFormConfOptions*
  - *OEFreeFormSolvResults*
  - *OEFreeFormConfResults*
- `OEOpt::OEFunc1::CheckVariablesValidity` added for checking the validity of arguments for the function to be optimized. The users of the all classes derived from OEFunc1 can implement it according to specific features of their objective functions.
- `OESz::OESzybki::GetSolutionEntropy` and `OESz::OESzybki::GetProteinBoundLigandEntropy` have been officially deprecated. *OESzybki.GetEntropy* that returns `double` values by reference has also been deprecated. Users should switch to using the *OESzybkiEnsembleResults* object instead.

6.31.2 Major bug fixes

- Dangling pointer in the implementation of the *OESzybki.UnsetProtein* has been found and fixed.
- Setting up a Newton-Raphson optimization on large systems can easily exhaust the amount of memory needed because of the O(n^2) Hessian matrix. If memory exhaustion is detected, the calculation will now error out instead of crashing.
- An exception has been added to catch the memory exhaustion upon Hessian allocation used for Newton-Raphson optimization of large systems. That could happen on some platforms with limited memory (for example Win32).
- *OESzybkiOptions* can now be passed to the *OESzybki* constructor.
- *OEProtFlex* namespace constants are now available in Python, Java, and C#.

6.31.3 Minor bug fixes

- Removed unbounded stack allocations.
- Verbose mode of some protein-ligand optimization could not report the number of cycles at the very end of optimization.
- Warning is issued upon the attempt to remove Coulomb terms from the combined force field Amber-MMFF (and Amber-MMFFS) with methods `OESz::OESzybki::SetRemoveCoulombTerms` and `OESz::OESzybkiOptions::SetRemoveCoulombTerms`.
- Unnecessary and confusing warning `Warning: Solid-body optimization allowed only for ligand in fixed protein` is eliminated when *OESzybkiOptions.SetRunType* is used with the *OERunType.SolidBodyOpt* argument.

6.31.4 Documentation fixes

- Examples added to demonstrate how to do an entropy estimation.
- Examples have been updated to use non-deprecated APIs.

6.32 Szybki TK 1.8.1

6.32.1 New features

- OENewtonOpt added for Newton-Raphson optimization.
- *OESzybkiOptions* class added for setting all options of the *OESzybki* object during construction. All new functionality, starting from the mentioned above Newton-Raphson optimization, will only be available through the *OESzybkiOptions* class. All the *OESzybki* methods with the same name as their *OESzybkiOptions* counterpart are deprecated and will be removed in Szybki TK.
- OEOptimizer2 abstract base class added to allow optimizers that use analytical second derivatives.
- Newton-Raphson optimization method added as an option for minimization. For this release it is restricted however to the MMFF94/MMFF94S force field and Sheffield solvation model. It is available only for optimization in Cartesian coordinates.
- The string representation of the gradient norm by the OECheckpoint1 object and by the *OESzybkiResults.Print* method will be in scientific notation to provide better precision.
- The following methods have been added to *OESzybkiResults*:
  - *OESzybkiResults.GetVibEntropy*
  - *OESzybkiResults.GetRotEntropy*
  - *OESzybkiResults.GetLnQvib*
  - *OESzybkiResults.GetLnQrot*
- Efficiency of diagonalization of Hessian matrix is improved.
- Added two new example programs showing how to use Newton-Raphson optimization.

6.32.2 Major bug fixes

- Calculation of PB solvent forces will no longer run out of memory and crash due to violent moves of the optimizer.
- *OESzybki.LoadPotentialGrid* and *OESzybki.SavePotentialGrid* will no longer crash due to the length of the file name passed.
- *OESzybki* will no longer allow a solid-body optimization in partially flexible protein as this was never an intended feature.

6.32.3 Minor bug fixes

- *OESzybki.SetProtein* will now fail and throw a warning if all the protein coordinates are zero.
- *OESzybkiResults.Print* output will now include MMFF terms for a protein-ligand complex single-point calculation when *OESzybki.SetProtein* was called before `OESz::OESzybki::SetRunType`.

6.32.4 Documentation fixes

- *OESzybkiEnsembleResults.GetChargeType* return type is now properly documented.

6.33 Szybki TK 1.7.5

6.33.1 New features

- User partial charges can be used in the combined AMBER-MMFF force field introduced in the version 1.7.3. This is done with the method `OESz::OESzybki::SetUseCurrentCharges` which until now could be applied only to MMFF94 force field for solvation and protein-ligand interaction.
- *OEOptType.SD_BFGS* and *OEOptType.SD_CG* can now be passed to `OESz::OESzybki::SetOptimizerType` to perform pre-optimization with 5 steps of steepest descent before either BFGS or CG optimization.
- *OESzybkiEnsembleResults.GetChargeType* added to return what type of partial charges were used for entropy estimation.
- *OESzybkiResults.GetTotalEnergy* no longer includes the harmonic constraint energy. *OESzybkiResults.GetTotalEnergyWithHarmConstraint* has been added to return the total energy with the harmonic constraint energy included.

6.33.2 Minor bug fixes

- `OESz::OESzybki::SetEveryConfAM1BCCCharges` will now preserve the existing partial charges from the input molecule when using AM1BCC charges for PB, Sheffield and Coulomb protein-ligand interactions. In previous versions MMFF94 charges were attached to the output molecule.
- When calculating ligand entropy, analytical calculation of second order derivatives in solution no longer misses the Hessian components coming from solvent forces.
- *OESzybki.GetEntropy* will no longer crash whenever a protein was set with *OESzybki.SetProtein* and the environment was not *OEEnvType.Protein*.
- Method `OESz::OESzybki::SetUseCurrentCharges` was ignored in Sheffield solvent second derivatives (default MMFF94 partial charges were used). This issue has been fixed.

6.34 Szybki TK 1.7.4

6.34.1 Bug fixes

- *OESzybkiEnsembleResults* is now usable with *OESzybki.GetEntropy* from Python, Java, and C#
- `OEOpt::OEMaxStepLineMinimize::SetMaxStep` had its arguments in the incorrect order for inheriting from OELineMinimize.
- OEBFGSOpt now properly handles gradients approaching zero.

6.35 Szybki TK 1.7.3

6.35.1 New features

- A new forcefield for protein-ligand interactions has been added. It is a combination of MMFF94S (or MMFF94) with Amber. In this combined force field, MMFF94S is used to describe the intramolecular interactions of the ligand and the Amber force field is used for the VdW and Coulomb interactions between ligand and protein. Currently, this force field can be used only for ligands inside rigid proteins.

- Two new methods were added to the API: *OESzybki.SetTorsionConstraint* and *OESzybki.ClearTorsionConstraint*. The first one adds a single-minimum harmonic torsional constraint for a torsion which is defined by a SMARTS pattern. It uses the functional form: $V = k_c(cos(phi) - cos(phi0))^2$, where k_c is the user specified force constant and $phi0$ is the reference torsion dihedral angle. The second method removes the torsional constraint if it exists.
- Estimation of ligand entropy using analytical MMFF Hessian is extended for ligands bound in a rigid protein. Previously the entropy of only free ligands in vacuum or in solution could be estimated with analytical second derivatives.

6.35.2 Major bug fixes

- A few numerical errors in the calculation of analytical second derivatives (Hessian matrix) in MMFF torsion and out-of plane terms have been fixed.

6.35.3 Minor bug fixes

- Constraint energy was reported twice by the method *OESzybkiResults.Print*. The problem is fixed.
- The method `OESz::OESzybki::GetExactVdWProteinLigand` was misspelled (`GetExactVdEProteinLigand` instead of `GetExactVdWProteinLigand`).
- When preoptimized ligands are used for entropy estimation it is necessary to gently move the ligand out of the minimum. The gradient normal below which such a perturbation was done was raised from 1 kcal/(mol Å) to 10 kcal/(mol Å). Research showed that for some conformations the above limit was too low which resulted in overestimation of its vibrational entropy.

6.36 Szybki TK 1.7.2

6.36.1 New features

- Two new API methods added: *SetIntramolecularVdWCutoff* and *GetIntramolecularVdWCutoff*
- Methods `OESz::OESzybki::SetProteinVdWCutoff` and `OESz::OESzybki::GetProteinVdWCutoff` have become deprecated. They have been replaced by `OESz::OESzybki::SetIntermolecularVdWCutoff` and `OESz::OESzybki::GetIntermolecularVdWCutoff` respectively.
- Steepest descent (*OEOptType.SD*) method added as an optional optimizer instead of BFGS.

6.36.2 Major bug fixes

- *OESzybkiEnsembleResults* now accessible from C#.

6.36.3 Minor bug fixes

- Reusing the *OESzybki* object after ligand entropy estimation in the protein environment, for other, non-entropy runs of a ligand inside the same protein, required an extra call to *OESzybki.SetProtein* in order to set correctly the new protein-electrostatic model correctly. This is not needed anymore.
- In the rare case when ligand is moved out from the protein binding site during entropy estimation the run is terminated with the proper warning. Previously, in such a case the attempts were done to reoptimize the ligand and rebuild the Hessian. Such a strategy was wrong because even if formally successful, it would lead to physically meaningless results. The situation described here can happen when the ligand is poorly docked, or if the X-ray structure used as input is very inaccurate.
- When the method `OESz::OESzybki::SetUseCurrentCharges` was used which causes the use of input charges for PB or Sheffield calculations, the molecule which was processed with the *OESzybki.operator()* methods,

on exit received the mmff94 charges. That behavior has been replaced by retaining the initial charges, and zeroing the molecule's partial charges when the input molecule had none.

- Error message for the case when the input structure generates infinite initial gradients is more informative.

6.37 Szybki TK 1.7.1

6.37.1 New features

- *OESzybki.GetEntropy* added which takes a non-const multiconformer molecule and a reference to a new object *OESzybkiEnsembleResults*. The method evaluates ligand entropy in different environments as other existing methods, however in addition provides essential data on the ensemble of ligand conformations. Specifically, it returns the optimized ensemble of unique ligand conformations in its first parameter, and populates the *OESzybkiEnsembleResults* objects with entropy components data for the ensemble. The *OESzybkiEnsembleResults* also provides the data for individual unique optimized conformations in the ensemble; in particular the free energy of isolating a specific conformation from the entire ensemble. An example code explaining the usage of the new *OESzybki.GetEntropy* method and the *OESzybkiEnsembleResults* class has been added.
- *OESzybkiEnsembleResults* class added that is mentioned above has been added.
- Two new API methods have been added which control the temperature in the entropy estimation. These are: *SetTemperature* and *GetTemperature*. The default temperature is 298K.

6.37.2 Bug fixes

- *OESzybki.SetHarmonicConstraints* did not impose harmonic constraints on flexible part of the protein. The problem has been fixed.
- Trying to reuse an *OESzybki* instance after an entropy calculation sometimes (depending on the subsequent methods call order) caused a crash. To work around that problem it was necessary to use a new instance of the *OESzybki* class. This bug has been fixed.
- Method *OESzybki.ClearFixAtoms* did not clear the unnecessary evaluation of the complete ligand energy at the very end of optimization with a subset of atoms frozen. The problem has been fixed.
- Protection against exhausting memory upon the usage of too many Coulomb atom pairs (for example in the attempt to optimize a large protein) has been added. Also, a confusing message upon exhausting the memory with too many vdW interactions has been removed.
- A warning is now issued in the case where one of the three methods setting the protein flexibility is used (*SetSideChainsFlexibility*, *SetPolarHFlexibility* or *SetResidueFlexibility*) when protein is not set. Also, a request of single point calculation (using *SetRunType*) properly clears any previous settings of partial protein flexibility.
- In the default optimization of a ligand inside the protein receptor the convergence criteria set by the method *SetOptimizerType* were not consistent with the intrinsic criteria of one the optimizers used. This inconsistency has been fixed.
- Using conjugate gradient optimizer (set with the *SetOptimizerType*) in combination with the method *SetExactVdWProteinLigand* did not report the number of cycles made by the optimizer. This problem has been fixed.

6.38 Szybki TK 1.7.0

6.38.1 New features

- An extension of the MMFF94 force field for tricoordinate boron compounds is offered in this release. Most compounds containing B-X bonds where X=C,N,O,S, and H are covered with the following exceptions: X=N(imine),N(sulfonamide), N(pyridinium) and N(quaternary). Also not supported are compounds in which boron is bonded to X=F,Cl,Br,I,B and Si, or makes a bond angle BYX. Compounds in which boron is a part

of four-membered rings of B1CCC1 type are also not available in the current parameterization because their existence is questionable: Ab initio calculations at the MP2/6-31G** level failed to identify stable structures for them (highly polar structures in which boron is four-coordinated are formed).

- NOTE WELL: Because of the partial parameterization for boron-containing compounds, users of the OEMMFF class (derived from OEGenericFF) need to pay attention to the return value of the `OEMolPotential::OEGenericFF::Setup` method: In the case where the parameters for a specific boron molecule are not available, this function returns `false`. Checking the return value of `OEMolPotential::OEGenericFF::PrepMol` is not enough, because it does not catch the case of missing force field parameters for a specific compound.
- Two new API methods have been added which control the salt concentration for all PB calculations with Szybki TK. These are: *SetSaltConcentration* and *GetSaltConcentration*. The first method takes salt concentration in M as a float number. The valid range is 0 - 0.08M; this method should not be used for higher salt concentrations. The second method returns the current salt concentration as a float. Default value is zero. Previous versions of Szybki TK assumed zero salt concentration.
- All PB calculations carried out with the previous Szybki TK versions used Bondi atomic radii. The current release offers two additional sets of atomic radii, called ZAP7 and ZAP9 as alternatives. They are described by Nicholls et. al. The new method: *SetAtomicRadii* allows the use of one of these two sets. The new method *GetAtomicRadiiType* returns the type of current atomic radii set.
- Better control of dielectric constants is provided. This includes a new method: *SetSolventDielectric* which allows change from the default value of 80. Method *GetSolventDielectric* returns the current value of the solvent dielectric constant. Method: *SetSolventModel* which determines the solvation model for a free ligand in solution, now takes an additional default parameter which sets the intrinsic dielectric constant of the ligand.
- A method which obtains gradients from the single point calculations is added. Specifically, a new method *OESzybkiResults.GetGradients* is added to the szybki_results.h header, and two new methods to the szybki.h header file. There are: *SetCalculateGradients* and *GetCalculateGradients*.
- Enforcement of proper behavior by the *OESzybki.operator()* for molecules without 3D coordinates. Such molecules are not processed and a warning is issued.

6.38.2 Bug fixes

- Method *GetProteinBoundLigandEntropy* is protected from selection of an inappropriate protein-ligand electrostatic model (entropy calculations for protein-bound ligands are done with the *OEProteinElectrostatics.ExactCoulomb* model).
- Calling *SetRunType* with an argument *OERunType.CartesiansOpt* immediately after a call to same method but with an argument *OERunType.TorsionsOpt* caused missing potential values for bonds and bond angles. The problem is fixed.
- Method *OESzybkiResults.Print* was not reporting MMFF terms in the case when *OESzybki.SetProtein* was used. The bug is fixed. Also, some minor improvements to this method is made, so it is clear which terms contribute to the total protein-ligand interactions and which are reported only for comparison. For example the usage of *OEProteinElectrostatics* results in reporting the exact and approximated Coulomb terms, however only the latter is used in the reported protein-ligand interaction.
- Fixed a minor bug where unnecessary warnings regarding missing protein parameters were issued in an entropy calculation where the protein is held rigid. This happens when mistakes or inaccuracies in the input protein structures lead to undefined force field parameters.
- Previously, processed molecules were output with aromaticity specified according to the MMFF94 aromaticity model. Now, output molecule are converted to the OE aromaticity model.
- Entropy calculations using the quasi-Newton method now give significantly more accurate answers in the two extreme cases where the input ligand structure is a) already optimized, or b) has a very high-energy (i.e. poor) structure.

6.39 Szybki TK 1.6.0

6.39.1 New features

- The algorithm for entropy estimation has been improved, specifically the calculation of vibrational entropy. It can now handle molecules with up to 2000 atoms. The previous limit was about 200 atoms.
- Entropy calculation of protein-bound ligands has been modified. This change refers to the partial desolvation entropy of the protein receptor. In the previous version, eq. 26 in *Wlodek-2010* had been used to evaluate the fraction of ligand exposed to the solvent. Recent unpublished data has shown that a much more accurate estimation can be achieved by using the direct surface calculation with Spicoli TK. This change was shown to be better at reproducing binding entropies.
- Method *SetSolventModel* now takes a dielectric constant as a second argument. Its default value is 80. In the previous versions dielectric constant was assumed to be 80 and the user could not change its value.
- Badly defined input molecules with very small distances between atoms (< 1e-3 A) that could generate infinite initial forces are now detected and no attempt is made to optimize such molecules. A warning "Geometry of the input protein or conformation is badly defined!" is issued so the user is forced to correct the geometry of the input molecules.
- A capability of calculating solution entropy for monoatomic ions like Na::math`^+` or F::math`^-` and noble gases is added.
- Documentation for three libraries: opt, mmff and molpotential, which in the past were called CASE, was added.

6.39.2 Bug fixes

- Estimation of entropy with the quasi-Newton method returned differing results (up to 6%) on different platforms. The problem was due to numerical-instability in the calculation of second derivatives. This problem has been eliminated in the current release with the use of a more stable method for line minimization (a backtrack line search).

6.40 Szybki TK 1.5.0

6.40.1 New features

- Entropy calculations of a ligand in different environments can be performed based on the methods described in the recent paper of Wlodek *et. al.* ([Wlodek-2010]).

  New methods added to the public API are:

  - `OESzybki.GetEntropy`
  - *GetSolutionEntropy*
  - *GetProteinBoundLigandEntropy*

- Default VdW protein-ligand potential used for the optimization of a ligand inside the protein binding site is precalculated in the form of a lookup table. This is done for the purpose of speed. The exact VdW protein-ligand potential can be used with the new method:

  - *SetExactVdWProteinLigand*

  when boolean parameter passed is `true`. Passing `false` causes switching to the default form of the protein-ligand VdW. The function:

  - *GetExactVdWProteinLigand*

  allows the querying of which form of the VdW potential is used.

- Two functions have been added to the public API in szybki.h. The first new function allows the use of conformation-dependent AM1BCC charges for every conformation in the calculation of PB and Sheffield solvation energies, for protein-ligand Coulomb interactions, and in entropy calculations.
  - *SetEveryConfAM1BCCCharges*

  The second function returns `true` if AM1BCC are being calculated for every conformation, `false` otherwise.
  - *GetEveryConfAM1BCCCharges*

6.40.2 Bug fixes

- In a special situation of ligand optimization in the protein binding site with the use of the electrostatics model determined by the *OEProteinElectrostatics.ExactCoulomb*, constant, when the optimization was preceded by the Zap binding calculations with the same protein and the same ligand, Coulombic interaction resulted in zero value. This bug has been fixed.

6.41 Szybki TK 1.4.0

6.41.1 New features

- Starting from the current release, certain classes of divalent selenium compounds can be handled by SZYBKI. Currently only some compounds in which selenium is bonded to the hydrogen or carbon atoms can be processed. Those compounds include selenols (RSeH) and selenides (RSeR1) in which Se is bonded to alkyl or aromatic carbon atoms, with the following exceptions:
  - Se makes a bond with alkyl C which belongs to the 3- or 4-membered ring
  - Se makes a bond with aromatic C which belongs to 5-membered ring

  In the restriction above the MMFF94 aromaticity model is adopted, so for example a pyridine derivative, pyridine-3-selenol can be processed, but a pyran derivative 2H-pyran-3-selenol cannot.
- Previous SZYBKI releases attempted to assign the "closest" MMFF94 atom type to those atoms for which a unique MMFF94 type could not be found. Starting from the 1.3.5 SZYBKI Application release this feature is removed, so consequently the input molecules containing such atoms will not be processed.
- Ligand RMSD which might be extracted from OESzybkiResults object, can optionally refer to heavy atoms only. A new public function:

  `void SetLigandRMSDHeavy(bool)`

  in OESzybki class has been added for that purpose.

6.41.2 Bug fixes

- A bug causing memory corruption when the function `SetSideChainsFlexibility()` was called after `SetProten()` was fixed.
- Memory leak in ligand torsion optimization inside the partially flexible protein receptor was removed.

6.42 Szybki TK 1.3.4

6.42.1 New features

- Calculation of the constant potential terms at the end of adapted optimization can be optionally eliminated, by the usage of the function `void SetCalculateFrozenTerms(bool)`. This feature reduces memory and cpu usage in the case of partial optimization of large molecules.

6.42.2 Bug fixes

- Member function `ClearFixAtoms()` did not clear properly previously fixed atoms. As a result, in spite of its use some atoms were still fixed. This bug has been fixed.
- A bug which occasionally caused errors in fixing a subset of atoms was fixed.

CHAPTER

SEVEN

COPYRIGHT AND TRADEMARKS

© 1997-2024 Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA. All rights reserved.

OpenEye, Lexichem, ROCS, Grapheme, and Orion are registered trademarks of Cadence Design Systems, Inc.

This material contains proprietary information of Cadence Design Systems, Inc. Use of copyright notice is precautionary only and does not imply publication or disclosure.

The information supplied in this document is believed to be true, but no liability is assumed for its use or the infringement of the rights of others resulting from its use. Information in this document is subject to change without notice and does not represent a commitment on the part of Cadence Design Systems, Inc.

This package is sold/licensed/distributed subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out, or otherwise circulated without Cadence Design Systems' prior consent, in any form of packaging or cover other than that in which it was produced. No part of this manual or accompanying documentation, may be reproduced, stored in a retrieval system on optical or magnetic disk, tape, CD, DVD or other medium, or transmitted in any form or by any means, electronic, mechanical, photocopying recording, or otherwise for any purpose other than for the purchaser's personal use without a legal agreement or other written permission granted by Cadence.

This product should not be used in the planning, construction, maintenance, operation or use of any nuclear facility nor the flight, navigation, or communication of aircraft or ground support equipment. Cadence Design Systems, Inc., shall not be liable, in whole or in part, for any claims arising from such use, including death, bankruptcy or outbreak of war.

Windows is a registered trademark of Microsoft Corporation. Apple, OS X, macOS, and Macintosh are registered trademarks of Apple Inc. UNIX is a registered trademark of the Open Group. Linux is a registered trademark of Linus Torvalds. Red Hat is a registered trademark of Red Hat, Inc. SUSE, SLED, and SLES are registered trademarks of Novell, Inc. Ubuntu is a registered trademark of Canonical Ltd.

SYBYL is a registered trademark of Certara, L.P. MDL, BIOVIA and ISIS are registered trademarks of Dassault Systemes and/or its affiliates. SMARTS and SMIRKS are registered trademarks of Daylight Chemical Information Systems, Inc. MacroModel is a registered trademark of Schrodinger, LLC.

Python is a registered trademark of the Python Software Foundation. Django is a registered trademark of the Django Software Foundation. Java is a registered trademark of Oracle and/or its affiliates.

Other products and software packages referenced in this document are trademarks and registered trademarks of their respective vendors or manufacturers.

CHAPTER

EIGHT

SAMPLE CODE

For all of the code examples and samples within documents published by OpenEye, Cadence Molecular Sciences, the following terms of use apply.

TERMS FOR USE OF SAMPLE CODE

The software below ("Sample Code") is provided to current licensees or subscribers of Cadence products or SaaS offerings (each a "Customer"). Customer is hereby permitted to use, copy, and modify the Sample Code, subject to these terms. Cadence claims no rights to Customer's modifications. Modification of Sample Code is at Customer's sole and exclusive risk. Sample Code may require Customer to have a then current license or subscription to the applicable Cadence offering.

THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. CADENCE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be liable for any damages or liability in connection with the Sample Code or its use.

CHAPTER

NINE

CITATION

Please follow the guidelines in this document to cite use of our products in research publications and other documents.

For more information, please contact OpenEye Support.

9.1 Orion®

To cite use of Orion, please use the following:

```
Orion <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

For example:

```
Orion 2025.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.eyesopen.com.
```

The version number for Orion appears in the product's release notes, for example, https://docs.eyesopen.com/orion-user-interface/releasenotes/index.html.

9.2 Orion Floes

To cite use of an Orion-based floe package, please use the following:

```
OpenEye <package-name> <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe,␣
↪NM. http://www.eyesopen.com.
```

For example:

```
OpenEye Large-Scale Floes 4.2.3. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

The version number for a floe package is displayed on the first page of the package's release notes. For example: https://docs.eyesopen.com/floe/modules/large_scale_floes/docs/source/releasenotes_2025_1.html.

9.3 Toolkits and Applications

To cite OpenEye toolkits or applications used in your work, please use the following:

```
OpenEye Toolkits [or Applications] <version-number>. OpenEye, Cadence Molecular Sciences,
↪ Santa Fe, NM. http://www.eyesopen.com.
```

For example:

```
OpenEye Toolkits 2024.2.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

The Applications and Toolkits are released as a package, multiple times per year. The first part of the version number designates the calendar year. The version number appears on the summary document for the package.

If you want to cite an individual application or toolkit, you can use the syntax:

```
<product name> <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

where <*product name*> is:

- Bioisostere TK
- OEChem TK
- OEDepict TK
- OEDocking TK
- OEFF TK
- Eon TK
- FastROCS TK
- Grapheme TK
- GraphSim TK
- Lexichem TK
- OEMedChem TK
- MolProp TK
- Omega TK
- Quacpac TK
- Shape TK
- Spicoli TK
- Spruce TK
- Szmap TK
- Szybki TK
- Zap TK
- AFITT Application
- BROOD Application
- OEDocking Application
- EON Application
- OMEGA Application
- PICTO Application
- pKa-Prospector Application
- QUACPAC Application

- ROCS Application
- SiteHopper Application
- SPRUCE Application
- SZMAP Application
- SZYBKI Application
- VIDA Application

You can use the version number of the individual Application or Toolkit. For example:

```
SZYBKI Application 2.8.0.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://
↪www.eyesopen.com.

Szybki TK 2.8.0.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

The first documentation pages give the product names and version numbers, as in https://docs.eyesopen.com/applications/szybki/index.html and https://docs.eyesopen.com/toolkits/python/szybkitk/index.html.

9.4 Publications for Bibliographies

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

9.4.1 Orion

Sørensen, J., Bannan, C.C., Calabrò, G., Jain, V., Ovanesyan, G., Smith, A., Zhang, S., Bayly, C.I., Darden, T.A., Geballe, M.T., LeBard, D.N., McGann, M., Moon, J.B., Muddana, H.S., Shewmaker, A., LaFon, J., Tolbert, R.W., Skillman, A.G., Nicholls, A. Orion® A Cloud-Native Molecular Design Platform. In *Computational Drug Discovery*. V. Poongavanam and V. Ramaswamy, Eds. Wiley-VCH, 2024; pp. 579–615. DOI: 10.1002/9783527840748.ch24

9.4.2 AFITT and FLYNN Applications

Wlodek, S.; Skillman, A.G.; Nicholls, A. Automated Ligand Placement and Refinement with a Combined Force Field and Shape Potential. *Acta Crystallogr., Sect. D: Struct. Biol.* **2006**, *62* (7), 741–749. DOI: 10.1107/s0907444906016076

9.4.3 OEDocking Application and Toolkit

Kelley, B.P.; Brown, S.P.; Warren, G.L.; Muchmore, S.W. POSIT: Flexible Shape-Guided Docking For Pose Prediction. *J. Chem. Inf. Model.* **2015**, *55* (8), 1771-1780. DOI: 10.1021/acs.jcim.5b00142

McGann, M. FRED Pose Prediction and Virtual Screening Accuracy. *J. Chem. Inf. Model.* **2011**, *51* (3), 578-596. DOI: 10.1021/ci100436p

McGann, M. FRED and HYBRID Docking Performance on Standardized Datasets. *J. Comput.-Aided Mol. Des.* **2012**, *26* (8), 897-906. DOI: 10.1007/s10822-012-9584-8

9.4.4 OMEGA Application and Toolkit

If you are using OMEGA in torsion driving mode, please cite the following two publications.

Hawkins, P.C.D.; Skillman, A.G.; Warren, G.L.; Ellingson, B.A.; Stahl, M.T. Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and the Cambridge Structural Database. *J. Chem. Inf. Model.* **2010**, *50* (4), 572-584. DOI: 10.1021/ci100031x

Hawkins, P.C.D. and Nicholls, A. Conformer Generation with OMEGA: Learning from the Data Set and the Analysis of Failures. *J. Chem. Inf. Model.* **2012**, *52* (11), 2919–2936. DOI: /10.1021/ci300314k

If you are using OMEGA in macrocyle mode, please cite the following.

Hawkins, P.C.D. and Wlodek, S. Decisions with Confidence: Application to the Conformation Sampling of Molecules in the Solid State. *J. Chem. Inf. Model.* **2020**, *60* (7), 3518–3533. DOI: /10.1021/acs.jcim.0c00358

9.4.5 ROCS Application

Hawkins, P.C.D.; Skillman, A.G.; Nicholls, A. Comparison of Shape-Matching and Docking as Virtual Screening Tools. *J. Med. Chem.* **2007**, *50* (1), 74-82. DOI: 10.1021/jm0603365

9.5 OpenEye MMDS Web Service

To cite use of the Macromolecular Data Service (MMDS) web service, please use the syntax:

```
Macromolecular Data Service <version-number>. OpenEye, Cadence Molecular Sciences, Santa␣
↪Fe, NM. http://www.eyesopen.com.
```

For example:

```
Macromolecular Data Service 1.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

The MMDS version number appears on the web service's release notes.

CHAPTER

TEN

TECHNOLOGY LICENSING

OpenEye products use the following technology under license.

Some of the open source technologies we use are licensed under the GNU Public License (GPL) version 2 or 3. In all instances, these technologies are communicated with using the "at arms length" principle, using pipes and command line arguments.

While the software is in some instances assembled into a container, that assembly does not invoke the GPL copyleft scope. For each Floe package where these are used, the environment log clearly details the installed programs and their version numbers. With the links below for each technology, the source for each of those programs is available.

Table 1: Licensed Technology

Name of Project	Website	License URL
_libgcc_mutex	https://github.com/conda-forge/_libgcc_mutex-feedstock	https://github.com/conda-forge/_libgcc_mutex-feedstock/blob/master/LICENSE.txt
_openmp_mutex	https://github.com/conda-forge/_openmp_mutex-feedstock	https://github.com/conda-forge/_openmp_mutex-feedstock/blob/main/LICENSE.txt
ablang2	https://github.com/TobiasHeOl/AbLang2	https://github.com/TobiasHeOl/AbLang2/blob/main/LICENSE
abseil-cpp	https://github.com/abseil/abseil-cpp	https://github.com/abseil/abseil-cpp/blob/master/LICENSE
absl-py	https://github.com/abseil/abseil-py	https://github.com/abseil/abseil-py/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
aiohappyeyeballs	https://github.com/aio-libs/aiohappyeyeballs	https://github.com/aio-libs/aiohappyeyeballs/blob/main/LICENSE
aiohttp	https://docs.aiohttp.org/en/stable/	https://github.com/aio-libs/aiohttp/blob/master/LICENSE.txt
aiosignal	https://github.com/aio-libs/aiosignal	https://github.com/aio-libs/aiosignal/blob/master/LICENSE
Amazon Linux 2	https://aws.amazon.com/amazon-linux-2	N/A
AmberUtils	http://ambermd.org	N/A
ambit	https://github.com/khimaros/ambit	https://github.com/khimaros/ambit/blob/master/LICENSE
amqp	https://github.com/celery/py-amqp	https://github.com/celery/py-amqp/blob/master/LICENSE
anaconda-anon-usage	https://github.com/anaconda/anaconda-anon-usage	https://github.com/anaconda/anaconda-anon-usage/blob/main/LICENSE
angular	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md
angular-animate	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md
angular-cache	https://github.com/jmdobry/angular-cache	https://github.com/jmdobry/angular-cache/raw/master/LICENSE
angular-cookies	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
angular-loggly-logger	https://github.com/ajbrown/angular-loggly-logger	https://github.com/ajbrown/angular-loggly-logger/raw/master/LICENSE.md
angular-mocks	https://github.com/angular/angular.js	https://raw.githubusercontent.com/angular/angular.js/master/LICENSE
angular-resource	https://github.com/angular/angular.js	https://raw.githubusercontent.com/angular/angular.js/master/LICENSE
angular-toggle-switch	https://github.com/cgarvis/angular-toggle-switch	https://github.com/cgarvis/angular-toggle-switch/blob/master/LICENSE
angular-ui-grid	https://github.com/angular-ui/ui-grid	https://github.com/angular-ui/ui-grid
angular-ui-router	https://github.com/angular-ui/ui-router	https://github.com/angular-ui/ui-router/raw/master/LICENSE
angular-ui-tree	https://github.com/angular-ui-tree/angular-ui-tree	https://github.com/angular-ui-tree/angular-ui-tree/raw/master/LICENSE
angular-vs-repeat	https://github.com/kamilkp/angular-vs-repeat	https://raw.githubusercontent.com/kamilkp/angular-vs-repeat/master/LICENSE
aniso8601	https://bitbucket.org/nielsenb/aniso8601	https://bitbucket.org/nielsenb/aniso8601/src/master/LICENSE
annotated-types	https://github.com/annotated-types/annotated-types	https://github.com/annotated-types/annotated-types/blob/main/LICENSE
antlr4-python3-runtime	http://www.antlr.org OR https://pypi.org/project/antlr4-python3-runtin	https://github.com/conda-forge/antlr4-python3-runtin/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
anyio	https://github.com/agronholm/anyio	https://github.com/agronholm/anyio/blob/master/LICENSE
appdirs	http://github.com/ActiveState/appdirs	http://github.com/ActiveState/appdirs/blob/master/LICENSE.txt
appengine	https://google.golang.org/appengine	https://github.com/golang/appengine/blob/master/LICENSE
arabic-reshaper	https://github.com/mpcabd/python-arabic-reshape	https://github.com/mpcabd/python-arabic-reshape/blob/master/LICENSE
archspec	https://github.com/archspec/archspec/blob/master/README.md	https://github.com/archspec/archspec/blob/master/COPYRIGHT
argon2-cffi	https://github.com/hynek/argon2-cffi	https://github.com/hynek/argon2-cffi/blob/master/LICENSE
argon2-cffi-bindings	https://github.com/hynek/argon2-cffi-bindings	https://github.com/hynek/argon2-cffi-bindings/blob/master/LICENSE
arpack	https://www.caam.rice.edu/software/ARPACK/	https://www.caam.rice.edu/software/ARPACK/
arrow	https://github.com/arrow-py/arrow	https://github.com/arrow-py/arrow?tab=Apache-2.0-1-ov-file#readme
ascii85	https://github.com/huandu/node-ascii85	https://github.com/huandu/node-ascii85/raw/master/LICENSE
ase	https://wiki.fysik.dtu.dk/ase/	https://wiki.fysik.dtu.dk/ase/development/licenseinfo.html#license-info
asgiref	https://github.com/django/asgiref/	https://github.com/django/asgiref/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
asn1crypto	https://github.com/wbond/asn1crypto	https://github.com/wbond/asn1crypto/blob/master/LICENSE
assertions go-render	https://github.com/smartystreets/assertions/internal/go-render/render	https://github.com/smartystreets/assertions/blob/master/internal/go-render/LICENSE
assertions oglmatchers	https://github.com/smartystreets/assertions/internal/oglematchers	https://github.com/smartystreets/assertions/blob/master/internal/oglematchers/LICENSE
assertions	https://github.com/smartystreets/assertions	https://github.com/smartystreets/assertions/blob/master/LICENSE.md
asttokens	https://github.com/gristlabs/asttokens	https://github.com/gristlabs/asttokens/blob/master/LICENSE
astunparse	https://github.com/simonpercivall/astunparse	https://github.com/simonpercivall/astunparse/blob/master/LICENSE
async-lru	https://github.com/aio-libs/async-lru	https://github.com/aio-libs/async-lru/blob/master/LICENSE
async-timeout	https://github.com/aio-libs/async-timeout	https://github.com/aio-libs/async-timeout/blob/master/LICENSE
atk-1.0	https://docs.gtk.org/atk/	https://gitlab.gnome.org/GNOME/atk/-/blob/master/COPYING
atomic	https://github.com/uber-go/atomic	https://github.com/uber-go/atomic/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
atomicwrites	https://github.com/untitaker/python-atomicwrites	https://github.com/untitaker/python-atomicwrites/blob/master/LICENSE
attrs	https://www.attrs.org/	https://github.com/python-attrs/attrs/blob/main/LICENSE
aws-sdk-go	https://github.com/aws/aws-sdk-go	https://github.com/aws/aws-sdk-go/blob/master/LICENSE.txt
babel	https://babel.pocoo.org/ OR https://github.com/python-babel/babel	https://github.com/python-babel/babel/blob/master/LICENSE OR https://babel.pocoo.org/en/latest/license.html#babel-license
Babel	https://github.com/python-babel/babel	https://github.com/python-babel/babel/blob/master/LICENSE
backcall	https://github.com/takluyver/backcall	https://github.com/takluyver/backcall/blob/master/LICENSE
backports	https://github.com/brandon-rhodes/backports	https://github.com/brandon-rhodes/backports/blob/master/LICENSE.TXT
backports.functools-lru-cache	https://github.com/jaraco/backports.functools_lru_cache	https://github.com/jaraco/backports.functools_lru_cache/blob/main/LICENSE
backports.zstd	https://github.com/rogdham/backports.zstd	https://github.com/rogdham/backports.zstd?tab=License-1-ov-file#readme
base62	https://github.com/kare/base62	https://github.com/kare/base62/blob/main/LICENSE
beautifulsoup4	https://www.crummy.com/software/BeautifulSoup/	https://www.crummy.com/software/BeautifulSoup/

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
billiard	https://github.com/celery/billiard	https://github.com/celery/billiard/blob/master/LICENSE.txt
biopython	https://biopython.org	https://github.com/biopython/biopython/blob/master/LICENSE.rst
biotite	https://github.com/biotite-dev/biotite/blob/master/README.rst	https://github.com/biotite-dev/biotite/blob/master/LICENSE.rst
bitarray	https://github.com/ilanschnell/bitarray	https://github.com/ilanschnell/bitarray/blob/master/LICENSE
bitset	https://github.com/willf/bitset	https://github.com/willf/bitset/blob/master/LICENSE
blas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
bleach	https://github.com/mozilla/bleach	https://github.com/mozilla/bleach/blob/main/LICENSE
blessings	https://github.com/erikrose/blessings	https://github.com/erikrose/blessings/blob/master/LICENSE
blinker	https://pythonhosted.org/blinker/	https://github.com/pallets-eco/blinker/blob/master/LICENSE
bokeh	https://github.com/bokeh/bokeh	https://github.com/bokeh/bokeh/blob/branch-4.0/LICENSE.txt
blosc	https://github.com/Blosc/python-blosc	https://github.com/Blosc/python-blosc/blob/master/LICENSE.txt
blosc2	https://github.com/Blosc/python-blosc2	https://github.com/Blosc/c-blosc2/blob/main/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
boltons	https://github.com/mahmoud/boltons	https://github.com/mahmoud/boltons/blob/master/LICENSE
Boltz	https://boltz.bio	https://github.com/jwohlwend/boltz/blob/main/LICENSE
boost	https://www.boost.org/doc/libs/1_71_0/libs/python/doc/html/index.html	https://www.boost.org/LICENSE_1_0.txt
boost-cpp	https://www.boost.org/doc/libs/1_71_0/libs/python/doc/html/index.html	https://www.boost.org/LICENSE_1_0.txt
bootstrap-vue	https://github.com/bootstrap-vue/bootstrap-vue	https://github.com/bootstrap-vue/bootstrap-vue/raw/master/LICENSE
boto3	https://github.com/boto/boto3	https://github.com/boto/boto3/blob/develop/LICENSE
botocore	https://github.com/boto/botocore	https://github.com/boto/botocore/blob/develop/LICENSE.txt
Bottleneck	https://bottleneck.readthedocs.io/en/latest/index.html	https://bottleneck.readthedocs.io/en/latest/license.html
Brotli	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
brotli-bin	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
brotli-python	http://python-hyper.org/projects/brotlipy/en/latest/	https://github.com/python-hyper/brotlicffi/blob/main/LICENSE
brotlicffi	https://github.com/python-hyper/brotlicffi	https://github.com/python-hyper/brotlicffi?tab=MIT-1-ov-file#readme
brotlipy	https://github.com/python-hyper/brotlicffi	https://github.com/python-hyper/brotlicffi/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
bson	http://github.com/py-bson/bson	https://github.com/py-bson/bson/blob/master/LICENSE_APACHE
bytefmt	https://code.cloudfoundry.org/bytefmt	https://github.com/cloudfoundry/bytefmt/blob/master/LICENSE
bzip2	https://www.bzip.org	https://spdx.org/licenses/bzip2-1.0.6.html
c-ares	https://github.com/c-ares/c-ares	https://github.com/c-ares/c-ares/blob/main/LICENSE.md
ca-certificates	https://github.com/conda-forge/ca-certificates-feedsto	https://wiki.mozilla.org/CA/Included_Certificates
cached-property	https://github.com/pydanny/cached-property	https://github.com/pydanny/cached-property/blob/master/LICENSE
cachetools	https://github.com/tkem/cachetools/	https://github.com/tkem/cachetools/blob/master/LICENSE
cairo	https://pycairo.readthedocs.io/en/latest/	https://github.com/pygobject/pycairo/blob/master/COPYING
canvas	https://github.com/Automattic/node-canvas	https://github.com/Automattic/node-canvas
cctbx	https://github.com/cctbx/cctbx_project	https://github.com/cctbx/cctbx_project/blob/master/cctbx/LICENSE_2_0.txt
celery	https://github.com/celery/celery	https://github.com/celery/celery/blob/master/LICENSE
Cerberus	https://docs.python-cerberus.org/en/stable/	https://docs.python-cerberus.org/en/stable/license.html
certifi	https://certifiio.readthedocs.io/en/latest/	https://wiki.mozilla.org/index.php?title=CA/IncludedCertificates&redirect=no

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cffi	https://github.com/python-cffi	https://github.com/cffi/cffi/blob/master/COPYRIGHT
cftime	https://pypi.org/project/cftime/	https://github.com/Unidata/cftime/blob/master/LICENSE
chardet	https://github.com/chardet/chardet	https://github.com/chardet/chardet/blob/master/LICENSE
charset-normalizer	https://github.com/ousret/charset_normalizer	https://github.com/Ousret/charset_normalizer/blob/master/LICENSE
ChEMBL Structure Pipeline	https://github.com/chembl/ChEMBL_Structure_Pipeline/tree/master	https://github.com/chembl/ChEMBL_Structure_Pipeline/blob/master/LICENSE
choreographer	https://github.com/mazpie/choreographer	https://github.com/mazpie/choreographer/blob/main/LICENSE
chunkreader	https://github.com/jackc/chunkreader/v2	https://github.com/jackc/chunkreader/blob/master/LICENSE
click	https://palletsprojects.com/p/click/	https://github.com/pallets/click/blob/main/LICENSE.rst
click-completion	https://github.com/click-contrib/click-completion	https://github.com/click-contrib/click-completion/blob/master/LICENSE
click-didyoumean	https://github.com/click-contrib/click-didyoumean	https://github.com/click-contrib/click-didyoumean/blob/master/LICENSE
click-plugins	https://github.com/click-contrib/click-plugins	https://github.com/click-contrib/click-plugins/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
click-repl	https://github.com/untitaker/click-repl	https://github.com/click-contrib/click-repl/blob/master/LICENSE
cloudpickle	https://github.com/cloudpipe/cloudpickle	https://github.com/cloudpipe/cloudpickle/blob/master/LICENSE
cmake	https://cmake.org/	https://cmake.org/licensing/
colorama	https://github.com/tartley/colorama	https://github.com/tartley/colorama/blob/master/LICENSE.txt
comm	https://github.com/ipython/comm	https://github.com/ipython/comm/blob/master/LICENSE
compose	https://github.com/docker/compose	https://github.com/docker/compose/blob/v2/LICENSE
conda-content-trust	https://github.com/conda/conda-content-trust	https://github.com/conda/conda-content-trust/blob/main/LICENSE
conda-libmamba-solver	https://github.com/conda/conda-libmamba-solv	https://github.com/conda/conda-libmamba-solv blob/main/LICENSE
conda-pack	https://conda.github.io/conda-pack/	https://github.com/conda/conda-pack/blob/main/LICENSE.txt
conda-package-handling	https://github.com/conda/conda-package-handli	https://github.com/conda/conda-package-handli blob/main/LICENSE
conda-package-streaming	https://anaconda.org/conda-forge/conda-package-strean	https://github.com/conda/conda-package-strean blob/main/LICENSE
conda-token	https://anaconda.org/anaconda/conda-token	https://anaconda.org/anaconda/conda-token

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
conda_package_strear	https://github.com/conda/conda-package-streamtree/main/conda_package_streaming	https://github.com/conda/conda-package-streamtab=License-1-ov-file
confuse	https://github.com/beetbox/confuse	https://github.com/beetbox/confuse/blob/main/LICENSE
contourpy	https://github.com/contourpy/contourpy	https://github.com/contourpy/contourpy/blob/main/LICENSE
cpp-expected	https://github.com/TartanLlama/expected	https://github.com/TartanLlama/expected?tab=CC0-1.0-1-ov-file#readme
cpp-peglib	https://github.com/yhirose/cpp-peglib	https://github.com/yhirose/cpp-peglib/blob/master/LICENSE
cryptography	https://github.com/pyca/cryptography	https://github.com/pyca/cryptography/blob/main/LICENSE
cssselect2	https://github.com/Kozea/cssselect2/	https://github.com/Kozea/cssselect2/blob/master/LICENSE
cuda-bindings	https://nvidia.github.io/cuda-python/cuda-bindings/latest/install.html	https://github.com/NVIDIA/cuda-python/blob/main/cuda_bindings/LICENSE
cuda-pathfinder	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/license.html
cudatoolkit	https://developer.nvidia.com/cuda-toolkit	https://docs.nvidia.com/cuda/eula/index.html
cuequivariance-ops-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cuequivariance-ops-torch-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cuequivariance-torch	https://docs.nvidia.com/cuda/cuequivariance/	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cupy-cuda113	https://cupy.dev/	https://github.com/cupy/cupy/blob/main/LICENSE
cupy-cuda11x	https://cupy.dev/	https://github.com/cupy/cupy/?tab=MIT-1-ov-file#readme
curl	https://curl.se	https://curl.se/docs/copyright.html
cycler	https://github.com/matplotlib/cycler	https://github.com/matplotlib/cycler/blob/main/LICENSE
cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl/blob/master/COPYING
Cython	https://cython.org	https://www.apache.org/licenses/LICENSE-2.0.html
d3	https://github.com/mbostock/d3	https://github.com/mbostock/d3/raw/master/LICENSE
dataclasses	https://github.com/ericvsmith/dataclasses	https://github.com/ericvsmith/dataclasses/blob/master/LICENSE.txt
ddsketch	http://github.com/datadog/sketches-py	https://github.com/DataDog/sketches-py/blob/master/LICENSE
debugpy	https://aka.ms/debugpy	https://github.com/microsoft/debugpy/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
decimal	https://github.com/shopspring/decimal	https://github.com/shopspring/decimal/blob/master/LICENSE
decorator	https://github.com/micheles/decorator	https://github.com/micheles/decorator/blob/master/LICENSE.txt
deepdiff	https://github.com/seperman/deepdiff	https://github.com/seperman/deepdiff/blob/master/LICENSE
deeptime	https://github.com/deeptime-ml/deeptime	https://github.com/deeptime-ml/deeptime/blob/main/LICENSE.txt
defusedxml	https://github.com/tiran/defusedxml	https://github.com/tiran/defusedxml/blob/main/LICENSE
dftd3	https://github.com/dftd3/tad-dftd3	https://github.com/dftd3/tad-dftd3/blob/main/LICENSE
dill	https://github.com/uqfoundation/dill	https://github.com/uqfoundation/dill/blob/master/LICENSE
distlib	https://distlib.readthedocs.io/en/latest/	https://github.com/pypa/distlib/blob/master/LICENSE.txt
distro	https://github.com/python-distro/distro	https://github.com/python-distro/distro/blob/master/LICENSE
Django	https://www.djangoproject.com/	https://github.com/django/django/blob/main/LICENSE
django-classy-tags	https://github.com/django-cms/django-classy-tags	https://github.com/django-cms/django-classy-tags/blob/master/LICENSE
django-cors-headers	https://github.com/adamchainz/django-cors-headers	https://github.com/adamchainz/django-cors-headers/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
django-csp	https://github.com/mozilla/django-csp	https://github.com/mozilla/django-csp/blob/main/LICENSE
django-extensions	https://github.com/django-extensions/django-extensions	https://github.com/django-extensions/django-extensions/blob/main/LICENSE
django-filter	https://github.com/carltongibson/django-filter/tree/master	https://github.com/carltongibson/django-filter/blob/main/LICENSE
django-redis	https://github.com/jazzband/django-redis	https://github.com/jazzband/django-redis/blob/master/LICENSE
django-taggit	https://github.com/jazzband/django-taggit	https://github.com/jazzband/django-taggit/blob/master/LICENSE
django-taggit-serializer	https://github.com/glemmaPaul/django-taggit-serializ	https://github.com/glemmaPaul/django-taggit-serializ blob/master/LICENSE
django-taggit-templatetags2	https://github.com/fizista/django-taggit-templat	https://github.com/fizista/django-taggit-templat blob/master/LICENSE.txt
djangorestframework	https://www.django-rest-framewor org/	https://github.com/encode/django-rest-framewor blob/master/LICENSE.md
dkh	https://psicode.org/psi4manual/master/dkh.html	https://github.com/psi4/dkh/blob/master/COPYING.LESSER
dm-tree	https://github.com/deepmind/tree	https://github.com/deepmind/tree/blob/master/LICENSE
docker-py	https://github.com/docker/docker-py/	https://github.com/docker/docker-py/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
docker-pycreds	https://github.com/shin-/dockerpy-creds	https://github.com/shin-/dockerpy-creds/blob/master/LICENSE
docopt	https://docopt.org	https://github.com/docopt/docopt/blob/master/LICENSE-MIT
docutils	https://docutils.sourceforge.io	https://docutils.sourceforge.io/COPYING.html
drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields/blob/master/LICENSE
DuckDB	https://duckdb.org/	https://github.com/duckdb/duckdb?tab=MIT-1-ov-file#readme
edgembar	https://rutgerslbsr.gitlab.io/fe-toolkit/edgembar/index.html	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
editdistance	https://github.com/roy-ht/editdistance	https://github.com/roy-ht/editdistance/blob/master/LICENSE
eigen	https://eigen.tuxfamily.org/index.php?title=Main_Page	https://eigen.tuxfamily.org/index.php?title=Main_Page#License
einops	https://github.com/arogozhnikov/einops	https://github.com/arogozhnikov/einops/blob/master/LICENSE
einx	https://github.com/fferflo/einx	https://github.com/fferflo/einx/blob/master/LICENSE
entrypoints	https://github.com/takluyver/entrypoints	https://github.com/takluyver/entrypoints/blob/master/LICENSE
errors	https://github.com/pkg/errors	https://github.com/pkg/errors/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
eslint-plugin	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile/-/blob/branch/default/LICENCE.rst
exceptiongroup	https://github.com/agronholm/exceptiongroup	https://github.com/agronholm/exceptiongroup/blob/main/LICENSE
execnet	https://execnet.readthedocs.io/en/latest/	https://github.com/pytest-dev/execnet/blob/master/LICENSE
executing	https://github.com/alexmojaki/executing	https://github.com/alexmojaki/executing/blob/master/LICENSE.txt
expat	https://libexpat.github.io	https://github.com/libexpat/libexpat/blob/master/expat/COPYING
fairscale	https://github.com/facebookresearch/fairscale	https://github.com/facebookresearch/fairscale/blob/main/LICENSE
fairseq	https://github.com/pytorch/fairseq	https://github.com/facebookresearch/fairseq/blob/main/LICENSE
fastjsonschema	https://github.com/horejsek/python-fastjsonschem	https://github.com/horejsek/python-fastjsonschem/blob/master/LICENSE
fastrlock	https://github.com/scoder/fastrlock	https://github.com/scoder/fastrlock/blob/master/LICENSE
fftw	https://www.fftw.org	commercial license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
filebuffer	https://github.com/mattetti/filebuffer	https://github.com/mattetti/filebuffer/blob/master/LICENSE
filelock	https://py-filelock.readthedocs.io/en/latest/index.html	https://github.com/tox-dev/py-filelock/blob/main/LICENSE
finufft	https://github.com/flatironinstitute/finufft	https://github.com/flatironinstitute/finufft/blob/master/LICENSE
Flask	https://flask.palletsprojects.com/en/2.1.x/	https://flask.palletsprojects.com/en/2.1.x/license/
flatbuffers	https://google.github.io/flatbuffers/	https://github.com/google/flatbuffers/blob/master/LICENSE.txt
flexcache	https://github.com/hgrecco/flexcache	https://github.com/hgrecco/flexcache/blob/main/LICENSE
flexparser	https://github.com/hgrecco/flexparser	https://github.com/hgrecco/flexparser/blob/main/LICENSE
flit-core	https://github.com/pypa/flit	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
FLTK	https://www.fltk.org/index.php	https://www.fltk.org/COPYING.php
fmt	https://fmt.dev/latest/index.html	https://github.com/fmtlib/fmt/blob/master/LICENSE.rst
font-awesome	https://github.com/FortAwesome/Font-Awesome	https://github.com/FortAwesome/Font-Awesome
font-ttf-dejavu-sans-mono	https://dejavu-fonts.github.io	https://dejavu-fonts.github.io/License.html
font-ttf-inconsolata	https://fonts.google.com/specimen/Inconsolata	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
font-ttf-source-code-pro	https://fonts.google.com/specimen/Source+Code+Pro	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL
font-ttf-ubuntu	https://fonts.google.com/specimen/Ubuntu	https://ubuntu.com/legal/font-licence
fontconfig	https://www.freedesktop.org/wiki/Software/fontconfig/	https://gitlab.freedesktop.org/fontconfig/fontconfig/-/blob/main/COPYING
fonts-conda-ecosystem	https://anaconda.org/conda-forge/fonts-conda-ecosyster	https://docs.conda.io/en/latest/license.html
fonts-conda-forge	https://anaconda.org/conda-forge/fonts-conda-forge/	https://creativecommons.org/licenses/by/4.0/
fonttools	https://github.com/fonttools/fonttools	https://github.com/fonttools/fonttools/blob/main/LICENSE
forcebalance	https://simtk.org/home/forcebalance OR https://github.com/leeping/forcebalance	https://github.com/leeping/forcebalance/blob/master/LICENSE
fqdn	https://github.com/ypcrts/fqdn	https://github.com/ypcrts/fqdn?tab=MPL-2.0-1-ov-file#readme
freetype	https://freetype.org	https://freetype.org/license.html
freetype-py	https://github.com/rougier/freetype-py	https://github.com/rougier/freetype-py?tab=License-1-ov-file#readme
fribidi	https://github.com/fribidi/fribidi	https://github.com/fribidi/fribidi/blob/master/COPYING
frozendict	https://github.com/Marco-Sulla/python-frozendict	https://github.com/Marco-Sulla/python-frozendict/blob/master/LICENSE.txt
frozenlist	https://github.com/aio-libs/frozenlist	https://github.com/aio-libs/frozenlist/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
fsmlite	https://github.com/tkem/fsmlite	https://github.com/tkem/fsmlite/blob/master/LICENSE
fsspec	https://github.com/fsspec/filesystem_spec	https://github.com/fsspec/filesystem_spec/blob/master/LICENSE
funcy	https://github.com/Suor/funcy	https://github.com/Suor/funcy/blob/master/LICENSE
gast	https://github.com/serge-sans-paille/gast/	https://github.com/serge-sans-paille/gast/blob/master/LICENSE
gau2grid	https://github.com/dgasmith/gau2grid	https://github.com/dgasmith/gau2grid/blob/master/LICENSE
gax-go	https://github.com/googleapis/gax-go/v2	https://github.com/googleapis/gax-go/blob/main/LICENSE
gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf/-/blob/master/COPYING
gemmi	https://gemmi.readthedocs.io/en/latest/	https://github.com/project-gemmi/gemmi/blob/master/LICENSE.txt
genproto	https://google.golang.org/genproto/googleapis	https://github.com/googleapis/go-genproto/blob/main/LICENSE
geometric	https://openbase.com/python/geometric	https://openbase.com/python/geometric
giflib	https://giflib.sourceforge.net	https://github.com/mirrorer/giflib/blob/master/COPYING
gitdb	https://github.com/gitpython-developers/gitdb	https://github.com/gitpython-developers/gitdb/blob/master/LICENSE
GitPython	https://github.com/gitpython-developers/GitPython	https://github.com/gitpython-developers/GitPython/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
glib	https://docs.gtk.org/glib/	https://gitlab.gnome.org/GNOME/glib/-/blob/main/LICENSES/LGPL-2.1-or-later.txt
GLM Library	https://github.com/g-truc/glm	https://github.com/g-truc/glm/blob/master/manual.md#section0
gls	https://github.com/jtolds/gls	https://github.com/jtolds/gls/blob/master/LICENSE
gmpy2	https://github.com/aleaxit/gmpy	https://github.com/gmpy2/gmpy2/blob/master/COPYING.LESSER
go-cleanhttp	https://github.com/hashicorp/go-cleanhttp	https://github.com/hashicorp/go-cleanhttp/blob/master/LICENSE
go-connections	https://github.com/docker/go-connections	https://github.com/docker/go-connections/blob/master/LICENSE
go-cpio	https://github.com/cavaliercoder/go-cpio	https://github.com/cavaliercoder/go-cpio/blob/master/LICENSE
go-getter	https://github.com/hashicorp/go-getter	https://github.com/hashicorp/go-getter/blob/master/LICENSE
go-homedir	https://github.com/mitchellh/go-homedir	https://github.com/mitchellh/go-homedir/blob/master/LICENSE
go-ini	https://github.com/go-ini/ini	https://github.com/go-ini/ini/blob/master/LICENSE
go-jmespath	https://github.com/jmespath/go-jmespath	https://github.com/jmespath/go-jmespath/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go-junit-report	https://github.com/jstemmer/go-junit-report	https://github.com/jstemmer/go-junit-report/blob/master/LICENSE
go-netrc	https://github.com/bgentry/go-netrc/netrc	https://github.com/bgentry/go-netrc/blob/master/LICENSE
go-ole	https://github.com/go-ole/go-ole	https://github.com/go-ole/go-ole/blob/master/LICENSE
go-pg	https://github.com/go-pg/pg	https://github.com/go-pg/pg/blob/master/LICENSE
go-redis	https://github.com/go-redis/redis/v8	https://github.com/go-redis/redis/blob/master/LICENSE
go-rendezvous	https://github.com/dgryski/go-rendezvous	https://github.com/dgryski/go-rendezvous/blob/master/LICENSE
go-safetemp	https://github.com/hashicorp/go-safetemp	https://github.com/hashicorp/go-safetemp/blob/master/LICENSE
go-sysconf	https://github.com/tklauser/go-sysconf	https://github.com/tklauser/go-sysconf/blob/master/LICENSE
go-testing-interface	https://github.com/mitchellh/go-testing-interface	https://github.com/mitchellh/go-testing-interface/blob/master/LICENSE
go-units	https://github.com/docker/go-units	https://github.com/docker/go-units/blob/master/LICENSE
go-version	https://github.com/hashicorp/go-version	https://github.com/hashicorp/go-version/blob/master/LICENSE
go-zglob	https://github.com/mattn/go-zglob	https://github.com/mattn/go-zglob/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go.opencensus	https://go.opencensus.io	https://github.com/census-instrumentatio opencensus-go/blob/master/LICENSE
gobrake.v2	https://gopkg.in/airbrake/gobrake.v2	https://github.com/airbrake/gobrake/blob/v2.0.9/LICENSE
goconvey	https://github.com/smartystreets/goconvey	https://github.com/smartystreets/goconvey/blob/master/LICENSE.md
golden-layout	https://github.com/deepstreamIO/golden-layout	https://github.com/deepstreamIO/golden-layout/raw/master/LICENSE
google-auth	https://google-auth.readthedocs.io/en/master/	https://github.com/GoogleCloudPlatform google-auth-library-p blob/master/LICENSE
google-auth-oauthlib	https://github.com/googleapis/google-auth-library-p	https://github.com/googleapis/google-auth-library-p blob/main/LICENSE
google-cloud-go	https://cloud.google.com/go	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-cloud-go/storage	https://cloud.google.com/go/storage	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-pasta	https://github.com/google/pasta	https://github.com/google/pasta/blob/master/LICENSE
google.golang.org/api	https://google.golang.org/api	https://github.com/googleapis/google-api-go-client/blob/main/LICENSE
gopsutil	https://github.com/shirou/gopsutil	https://github.com/shirou/gopsutil/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
gorilla websocket	https://github.com/gorilla/websocket	https://github.com/gorilla/websocket/blob/master/LICENSE
gpytorch	https://gpytorch.ai	https://github.com/cornellius-gp/gpytorch?tab=MIT-1-ov-file#readme
graphite2	https://github.com/silnrsi/graphite	https://github.com/silnrsi/graphite/blob/master/LICENSE
graphviz	https://graphviz.org/	https://graphviz.org/license/
greenlet	https://greenlet.readthedocs.io/en/latest/	https://github.com/python-greenlet/greenlet/blob/master/LICENSE
groupcache	https://github.com/golang/groupcache	https://github.com/golang/groupcache/blob/master/LICENSE
grpc	https://google.golang.org/grpc	https://github.com/grpc/grpc-go/blob/master/LICENSE
grpc-cpp	https://github.com/grpc/grpc	https://github.com/grpc/grpc/blob/master/LICENSE
grpcio	https://github.com/grpc/grpc.io/blob/main/LICENSE	https://github.com/grpc/grpc.io/blob/main/LICENSE
gtk2	https://gitlab.gnome.org/GNOME/gtk	https://gitlab.gnome.org/GNOME/gtk/-/blob/main/COPYING
gts	https://sourceforge.net/projects/gts/	https://www.gnu.org/licenses/lgpl-3.0.html
h11	https://github.com/python-hyper/h11	https://github.com/python-hyper/h11/blob/master/LICENSE.txt
h2	https://github.com/python-hyper/h2	https://github.com/python-hyper/h2?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
h5py	https://www.h5py.org	https://docs.h5py.org/en/stable/licenses.html
harfbuzz	https://github.com/harfbuzz/harfbuzz	https://github.com/harfbuzz/harfbuzz/blob/main/COPYING
hdbscan	https://hdbscan.readthedocs.io/en/latest/	https://github.com/scikit-learn-contrib/hdbscan/blob/master/LICENSE
hdf4	https://www.hdfgroup.org/solutions/hdf4/	https://github.com/HDFGroup/hdf4/blob/master/COPYING
hdf5	https://www.hdfgroup.org/solutions/hdf5/	https://github.com/HDFGroup/hdf5/blob/master/COPYING
he	https://github.com/mathiasbynens/he	https://github.com/mathiasbynens/he/raw/master/LICENSE-MIT.txt
hpack	https://github.com/python-hyper/hpack	https://github.com/python-hyper/hpack?tab=MIT-1-ov-file#readme
html-loader	https://github.com/webpack-contrib/html-loader	https://github.com/webpack-contrib/html-loader/raw/master/LICENSE
html5lib	https://github.com/html5lib/html5lib-python	https://github.com/html5lib/html5lib-python/blob/master/LICENSE
htslib	https://www.htslib.org	https://github.com/samtools/htslib/blob/develop/LICENSE
httpcore	https://www.encode.io/httpcore/	https://github.com/encode/httpcore/blob/master/LICENSE.md
httpx	https://github.com/encode/httpx	https://github.com/encode/httpx/blob/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
humanize	https://github.com/jmoiron/humanize	https://github.com/jmoiron/humanize/blob/master/LICENCE
hydra-core	https://github.com/facebookresearch/hydra	https://github.com/facebookresearch/hydra/blob/main/LICENSE
hyperframe	https://github.com/python-hyper/hyperframe	https://github.com/python-hyper/hyperframe?tab=MIT-1-ov-file#readme
icu	https://github.com/unicode-org/icu	https://github.com/unicode-org/icu/blob/main/icu4c/LICENSE
idna	https://github.com/kjd/idna	https://github.com/kjd/idna/blob/master/LICENSE.md
ihm	https://github.com/ihmwg/python-ihm	https://github.com/ihmwg/python-ihm/blob/main/LICENSE
imageio	https://github.com/imageio/imageio	https://github.com/imageio/imageio/blob/master/LICENSE
imagesize	https://github.com/shibukawa/imagesize_py	https://github.com/shibukawa/imagesize_py/blob/master/LICENSE.rst
ImmuneBuilder	https://github.com/oxpig/ImmuneBuilder/tree/main	https://github.com/oxpig/ImmuneBuilder/blob/main/LICENSE
importlib-metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE
importlib_metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
importlib_resources	https://github.com/python/importlib_resources	https://github.com/python/importlib_resources/blob/main/LICENSE
InChI	https://iupac.org/who-we-are/divisions/division-details/inchi/	https://iupac.org/wp-content/uploads/2021/06/iupac-inchi-license_2020.pdf
inflection	https://github.com/jinzhu/inflection	https://github.com/jinzhu/inflection/blob/master/LICENSE
ini.v1	https://gopkg.in/ini.v1	https://github.com/go-ini/ini/blob/v1.66.2/LICENSE
iniconfig	https://github.com/pytest-dev/iniconfig	https://github.com/pytest-dev/iniconfig/blob/main/LICENSE
innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill/raw/master/LICENSE
intel-openmp	https://github.com/hermitcore/libomp_oss	https://github.com/hermitcore/libomp_oss/blob/hermit/LICENSE.txt
ipykernel	https://ipython.org	https://ipython.org
ipython	https://ipython.org	https://github.com/ipython/ipython/blob/master/LICENSE
ipython-genutils	http://ipython.org	https://github.com/ipython/ipython_genutils/blob/master/COPYING.md
ipython_pygments_le	https://github.com/ipython/ipython-pygments-lex	https://github.com/ipython/ipython-pygments-lex blob/main/LICENSE
ipywidgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
isodate	https://github.com/gweis/isodate/	https://github.com/gweis/isodate/blob/master/LICENSE
isoduration	https://github.com/bolsote/isoduration	https://github.com/bolsote/isoduration?tab=ISC-1-ov-file#readme
itsdangerous	https://palletsprojects.com/p/itsdangerous/	https://itsdangerous.palletsprojects.com/en/2.1.x/license/
jax	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxlib	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxtyping	https://github.com/patrick-kidger/jaxtyping	https://github.com/patrick-kidger/jaxtyping?tab=License-1-ov-file#readme
jedi	https://jedi.readthedocs.io/en/latest/	https://github.com/davidhalter/jedi/blob/master/LICENSE.txt
Jinja2	https://palletsprojects.com/p/jinja/	https://github.com/pallets/jinja/blob/main/LICENSE.rst
jmespath	https://github.com/jmespath/jmespath.py	https://github.com/jmespath/jmespath.py/blob/develop/LICENSE.txt
joblib	https://joblib.readthedocs.io/en/latest/	https://github.com/joblib/joblib/blob/master/LICENSE.txt
jpeg	https://www.ijg.org	https://www.ijg.org/files/README
json5	https://github.com/dpranke/pyjson5	https://github.com/dpranke/pyjson5/blob/main/LICENSE
jsonfield	https://github.com/dmkoch/django-jsonfield/	https://github.com/rpkilby/jsonfield/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jsonpatch	https://github.com/stefankoegl/python-json-patch	https://github.com/stefankoegl/python-json-patch/blob/master/LICENSE
jsonpickle	https://github.com/jsonpickle/jsonpickle	https://github.com/jsonpickle/jsonpickle/blob/main/LICENSE
jsonpointer	https://github.com/stefankoegl/python-json-pointer	https://github.com/stefankoegl/python-json-pointer/blob/master/LICENSE.txt
jsonschema	https://github.com/python-jsonschema/jsonschema	https://github.com/python-jsonschema/jsonschema/blob/master/COPYING
jsonschema-specifications	https://github.com/python-jsonschema/jsonschema-specificat	https://github.com/python-jsonschema/jsonschema-specificat blob/main/COPYING
jstat	https://github.com/jstat/jstat	https://www.opensource.org/licenses/mit-license.php
jupyter-client	https://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/COPYING.md
jupyter-core	https://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/COPYING.md
jupyter-events	https://github.com/jupyter/jupyter_events	https://github.com/jupyter/jupyter_events/blob/main/LICENSE
jupyter-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp
jupyter-serverhttp://jupyter.or	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE	
jupyter_client	http://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jupyter_core	http://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/LICENSE
jupyter_server	https://github.com/jupyter-server/jupyter_server	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE
jupyter_server_termin	https://github.com/jupyter-server/jupyter_server_terminals	https://github.com/jupyter-server/jupyter_server_terminals/blob/main/LICENSE
jupyterlab	https://github.com/jupyterlab/jupyterlab	https://github.com/jupyterlab/jupyterlab/blob/main/LICENSE
jupyterlab-pygments	http://jupyter.org	https://github.com/jupyterlab/jupyterlab_pygments/blob/main/LICENSE
jupyterlab-widgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/master/LICENSE
jupyter-lab_pygments	https://github.com/jupyterlab/jupyterlab_pygments	https://github.com/jupyterlab/jupyterlab_pygments?tab=BSD-3-Clause-1-ov-file#readme
jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server/blob/main/LICENSE
jupyterlab_widgets	https://github.com/jupyter-widgets/ipywidgets	https://github.com/jupyter-widgets/ipywidgets?tab=BSD-3-Clause-1-ov-file#readme
kaleido	https://github.com/plotly/Kaleido	https://github.com/plotly/Kaleido/blob/master/LICENSE.txt
keras	https://github.com/keras-team/keras	https://github.com/keras-team/keras/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
Keras-Preprocessing	https://github.com/keras-team/keras-preprocessing	https://github.com/keras-team/keras-preprocessing/blob/master/LICENSE
keras-tuner	https://github.com/keras-team/keras-tuner	https://github.com/keras-team/keras-tuner/blob/master/LICENSE
keyring	https://github.com/jaraco/keyring	https://github.com/jaraco/keyring/blob/main/LICENSE
keyutils	https://github.com/sassoftware/python-keyutils	https://github.com/sassoftware/python-keyutils/blob/master/LICENSE
kiwisolver	https://kiwisolver.readthedocs.io/en/latest/	https://github.com/nucleic/kiwi/blob/main/LICENSE
kombu	https://kombu.readthedocs.io	https://github.com/celery/kombu/blob/master/LICENSE
krb5	https://web.mit.edu/kerberos/	https://github.com/krb5/krb5/blob/master/src/lib/gssapi/LICENSE
kt-legacy	https://github.com/haifeng-jin/kt-legacy	https://github.com/haifeng-jin/kt-legacy/blob/main/LICENSE
lark	https://github.com/lark-parser/lark	https://github.com/lark-parser/lark?tab=MIT-1-ov-file#readme
lazy_loader	https://github.com/scientific-python/lazy_loader	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
lcms2	https://www.littlecms.com	https://github.com/sk1project/python-lcms2/blob/master/LICENSE
legacy-cgi	https://github.com/jackrosenthal/legacy-cgi	https://github.com/jackrosenthal/legacy-cgi?tab=License-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lerc	https://github.com/Esri/lerc	https://github.com/Esri/lerc/blob/master/LICENSE
libarchive	http://www.libarchive.org	https://raw.githubusercontent.com/libarchive/libarchive/master/COPYING
libblas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
libbrotlicommon	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlidec	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlienc	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libcblas	https://anaconda.org/conda-forge/libcblas	N/A
libclang	https://github.com/sighingnow/libclang	https://github.com/sighingnow/libclang/blob/master/LICENSE.TXT
libcurl	https://curl.se/libcurl/	https://curl.se/docs/copyright.html
libcxx	https://github.com/llvm-mirror/libcxx	https://github.com/llvm-mirror/libcxx/blob/master/LICENSE.TXT
libdb	https://www.oracle.com/technology/software/products/berkeley-db/index.html	https://github.com/berkeleydb/libdb/blob/master/LEGAL
libdeflate	https://github.com/ebiggers/libdeflate	https://github.com/ebiggers/libdeflate/blob/master/COPYING
libedit	https://thrysoee.dk/editline/	http://www.netbsd.org/Goals/redistribution.html
libev	https://software.schmorp.de/pkg/libev.html	https://software.schmorp.de/pkg/libev.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libexpat	https://libexpat.github.io/	https://github.com/libexpat/libexpat?tab=MIT-1-ov-file#readme
libffi	https://github.com/libffi/libffi	https://github.com/libffi/libffi/blob/master/LICENSE
libgcc	https://gcc.gnu.org/onlinedocs/gccint/Libgcc.html	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libgcrypt	https://gnupg.org/software/index.html	https://git.gnupg.org/cgi-bin/gitweb.cgi?p=libgcrypt.git;a=blob;f=COPYING.LIB;hb=HEAD
libgd	https://libgd.github.io	https://libgd.github.io/manuals/2.2.1/files/license-txt.html
libglib	https://github.com/PolMine/libglib	https://github.com/PolMine/libglib/blob/master/include/glib-2.0/glib.h
libiconv	https://www.gnu.org/software/libiconv/	https://www.gnu.org/software/libiconv/#downloading
libint	https://tinyurl.com/yvw97wbw	https://tinyurl.com/yvw97wbw
liblapack	http://www.netlib.org/lapack/	https://www.netlib.org/lapack/LICENSE.txt
liblapacke	https://anaconda.org/conda-forge/liblapacke	https://www.netlib.org/lapack/LICENSE.txt
liblzma	https://tukaani.org/xz/	https://git.tukaani.org/?p=xz.git;a=blob;f=COPYING.0BSD
libmamba	https://medium.com/@QuantStack/open-software-packag	https://github.com/mamba-org/mamba/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libmambapy	https://www.anaconda.com/blog/a-faster-conda-for-a-g	https://github.com/conda/conda-libmamba-solv blob/main/LICENSE
libnetcdf	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
libnghttp2	https://github.com/nghttp2/nghttp2	https://github.com/nghttp2/nghttp2/blob/master/COPYING
libnsl	https://github.com/thkukuk/libnsl	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libopenblas	https://www.openblas.net/	https://github.com/xianyi/OpenBLAS/blob/develop/LICENSE
libpng	https://www.libpng.org/pub/png/libpng.html	https://www.libpng.org/pub/png/src/libpng-LICENSE.txt
libpq	https://www.postgresql.org/	https://www.postgresql.org/about/licence/
librsvg	https://gitlab.gnome.org/GNOME/librsvg	https://gitlab.gnome.org/GNOME/librsvg/-/blob/main/COPYING.LIB
libsolv	https://github.com/openSUSE/libsolv	https://github.com/openSUSE/libsolv/blob/master/LICENSE.BSD
libsqlite	https://github.com/mackyle/sqlite	https://github.com/mackyle/sqlite?tab=License-1-ov-file#readme
libssh2	https://github.com/libssh2/libssh2	https://github.com/libssh2/libssh2/blob/master/COPYING
libstdcxx	https://gcc.gnu.org/	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libtiff	https://www.libtiff.org/	https://www.libtiff.org/misc.html
libtrust	https://github.com/docker/libtrust	https://github.com/docker/libtrust/blob/master/LICENSE
libuuid	https://sourceforge.net/projects/libuuid/	https://sourceforge.net/p/libuuid/code/ci/master/tree/COPYING
libuv	https://github.com/libuv/libuv	https://github.com/libuv/libuv/blob/v1.x/LICENSE
libwebp	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libwebp-base	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libxc	https://www.tddft.org/programs/libxc/	https://gitlab.com/libxc/libxc/-/blob/master/COPYING
libxcb	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-libxcb/blob/master/COPYING
libxcrypt	https://github.com/besser82/libxcrypt	https://github.com/besser82/libxcrypt?tab=LGPL-2.1-1-ov-file#readme
libxml2	https://git.gnome.org/browse/libxml2/	https://gitlab.gnome.org/GNOME/libxml2/-/blob/master/Copyright
libxmlsec1	https://github.com/lsh123/xmlsec	https://github.com/lsh123/xmlsec/blob/master/README
libxslt	https://gitlab.gnome.org/GNOME/libxslt/-/wikis/home	https://gitlab.gnome.org/GNOME/libxslt/-/blob/master/Copyright

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libzlib	https://zlib.net	https://zlib.net/zlib_license.html
lightning-utilities	https://github.com/Lightning-AI/utilities	https://github.com/Lightning-AI/utilities?tab=Apache-2.0-1-ov-file#readme
lime	https://github.com/marcotcr/lime	https://github.com/marcotcr/lime/blob/master/LICENSE
linear-operator	https://github.com/cornellius-gp/linear_operator	https://github.com/cornellius-gp/linear_operator?tab=MIT-1-ov-file#readme
lit	http://llvm.org	https://releases.llvm.org/14.0.0/LICENSE.TXT
llvm-openmp	https://github.com/llvm-mirror/openmp	https://github.com/llvm-mirror/openmp/blob/master/LICENSE.txt
llvmlite	http://llvmlite.readthedocs.io	https://github.com/numba/llvmlite/blob/main/LICENSE
loader-utils	https://github.com/webpack/loader-utils	https://github.com/webpack/loader-utils/raw/master/LICENSE
logistro	https://github.com/geopozo/logistro	https://github.com/geopozo/logistro?tab=MIT-1-ov-file#readme
logomaker	https://logomaker.readthedocs.io/en/latest/	https://github.com/jbkinney/logomaker/blob/master/LICENSE
logrus	https://github.com/sirupsen/logrus	https://github.com/sirupsen/logrus/blob/master/LICENSE
logrus-airbrake-hook.v2	https://gopkg.in/gemnasium/logrus-airbrake-hook.v2	https://github.com/gemnasium/logrus-airbrake-hook/blob/v2.1.2/LICENSE
lxml	https://lxml.de	https://lxml.de/index.html#license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lz4-c	https://www.lz4.org/	https://github.com/lz4/lz4/blob/dev/LICENSE
lzo	https://www.oberhumer.com/opensource/lzo/	https://www.oberhumer.com/opensource/gpl.html
mamba	https://github.com/mamba-org/mamba	https://github.com/mamba-org/mamba?tab=BSD-3-Clause-1-ov-f readme
markdown	https://github.com/evilstreak/markdown-js	https://www.opensource.org/licenses/mit-license.php
markdown-it-py	https://github.com/executablebooks/markdown-it-py	https://github.com/executablebooks/markdown-it-py/blob/master/LICENSE
markupsafe	https://github.com/pallets/markupsafe/tree/main/src/markupsafe	https://github.com/pallets/markupsafe/tree/main?tab=BSD-3-Clause-1-ov-f
MarkupSafe	https://palletsprojects.com/p/markupsafe/	https://github.com/pallets/markupsafe/blob/main/LICENSE.rst
mashumaro	https://github.com/Fatal1ty/mashumaro	https://github.com/Fatal1ty/mashumaro
matplotlib	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-base	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-inline	https://github.com/ipython/matplotlib-inline	https://github.com/ipython/matplotlib-inline/blob/master/LICENSE
mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mda_xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib?tab=License-1-ov-file#readme
mdtraj	https://www.mdtraj.org/	https://www.mdtraj.org/1.9.8.dev0/index.html#license
mdurl	https://github.com/markdown-it/mdurl	https://github.com/markdown-it/mdurl/blob/master/LICENSE
menuinst	https://github.com/conda/menuinst	https://github.com/conda/menuinst/blob/main/LICENSE.txt
mergo	https://github.com/imdario/mergo	https://github.com/imdario/mergo/blob/master/LICENSE
mistune	https://github.com/lepture/mistune	https://github.com/lepture/mistune/blob/master/LICENSE
mkl	https://github.com/rust-math/intel-mkl-src	https://github.com/rust-math/intel-mkl-src/blob/master/License.txt
mkl-fft	https://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft/blob/master/LICENSE.txt
mkl-random	https://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random/blob/master/LICENSE.txt
mkl-service	https://github.com/IntelPython/mkl-service	https://github.com/IntelPython/mkl-service/blob/master/LICENSE.txt
mkl_fft	http://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft?tab=BSD-3-Clause-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mkl_random	http://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random?tab=BSD-3-Clause-1-ov-f readme
ml-dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
ml_dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
modelcif	https://github.com/ihmwg/python-modelcif	https://github.com/ihmwg/python-modelcif/blob/main/LICENSE
molecupy	https://molecupy.readthedocs.io/en/latest/	https://github.com/samirelanduk/atomium/blob/master/LICENSE
moment	https://github.com/moment/moment	https://github.com/moment/moment/raw/master/LICENSE
moment-precise-range-plugin	https://github.com/codebox/moment-precise-range	https://github.com/codebox/moment-precise-range raw/master/LICENSE.md
more-itertools	https://github.com/more-itertools/more-itertools	https://github.com/more-itertools/more-itertools/blob/master/LICENSE
mpiplus	https://github.com/choderalab/mpiplus	https://github.com/choderalab/mpiplus/blob/main/LICENSE
mpmath	http://mpmath.org/	https://github.com/mpmath/mpmath/blob/master/LICENSE
mrcfile	https://github.com/ccpem/mrcfile	https://github.com/ccpem/mrcfile/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
msgpack	https://msgpack.org/	https://github.com/msgpack/msgpack-python/blob/main/COPYING
multidict	https://github.com/aio-libs/multidict	https://github.com/aio-libs/multidict/blob/master/LICENSE
multierr	https://go.uber.org/multierr	https://github.com/uber-go/multierr/blob/master/LICENSE.txt
multiprocess	https://github.com/uqfoundation/multiprocess	https://github.com/uqfoundation/multiprocess/blob/master/LICENSE
munkres	https://software.clapper.org/munkres/	https://github.com/bmc/munkres/blob/master/LICENSE.md
myesui uuid	https://github.com/myesui/uuid	https://pkg.go.dev/github.com/myesui/uuid?tab=licenses
namex	https://github.com/bcgov/namex	https://github.com/bcgov/namex/blob/main/LICENSE
narwhals	https://github.com/narwhals-dev/narwhals	https://github.com/narwhals-dev/narwhals?tab=MIT-1-ov-file#readme
nbclassic	http://jupyter.org	https://github.com/jupyter/nbclassic/blob/main/LICENSE
nbclient	https://jupyter.org	https://github.com/jupyter/nbclient/blob/main/LICENSE
nbconvert	https://jupyter.org	https://github.com/jupyter/nbconvert/blob/main/LICENSE
nbformat	http://jupyter.org	https://github.com/jupyter/nbformat/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ncurses	https://invisible-island.net/ncurses/	https://invisible-island.net/ncurses/ncurses-license.html
ndfes	https://rutgerslbsr.gitlab.io/fe-toolkit/	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
ndindex	https://quansight-labs.github.io/ndindex/	https://github.com/Quansight-Labs/ndindex?tab=MIT-1-ov-file#readme
nest-asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio/blob/master/LICENSE
nest_asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio?tab=BSD-2-Clause-1-ov-f readme
netcdf-fortran	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
netCDF4	http://github.com/Unidata/netcdf4-python	https://github.com/Unidata/netcdf4-python/blob/master/LICENSE
nettle	https://git.lysator.liu.se/nettle/nettle	https://git.lysator.liu.se/nettle/nettle/-/blob/master/AUTHORS?ref_type=heads
networkx	https://networkx.org	https://github.com/networkx/networkx/blob/main/LICENSE.txt
nfpm	https://github.com/goreleaser/nfpm	https://github.com/goreleaser/nfpm/blob/master/LICENSE.md
ng-tags-input	https://github.com/mbenford/ngTagsInput	https://github.com/mbenford/ngTagsInput
ng-toast	https://github.com/tameraydin/ngToast	https://github.com/tameraydin/ngToast

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ngdraggable	https://github.com/fatlinesofcode/ngDraggable	https://github.com/fatlinesofcode/ngDraggable/raw/master/LICENSE.txt
ngVue	https://github.com/ngVue/ngVue	https://github.com/ngVue/ngVue/raw/master/LICENSE
nh3	https://nh3.readthedocs.io/en/latest/	https://github.com/messense/nh3/blob/main/LICENSE
nlohmann_json	https://github.com/nlohmann/json	https://github.com/nlohmann/json?tab=MIT-1-ov-file#readme
nlopt	https://nlopt.readthedocs.io/en/latest/NLopt_Python_Reference/	https://nlopt.readthedocs.io/en/latest/NLopt_License_and_Copyright/
nodejs	https://nodejs.org/en/	https://raw.githubusercontent.com/nodejs/node/master/LICENSE
nomkl	https://github.com/conda-forge/nomkl-feedstock	https://github.com/conda-forge/nomkl-feedstock/blob/main/LICENSE.txt
notebook	http://jupyter.org	https://github.com/jupyter/notebook/blob/main/LICENSE
notebook-shim	https://github.com/jupyter/notebook_shim	https://github.com/jupyter/notebook_shim
notebook_shim	http://jupyter.org	https://github.com/jupyter/notebook_shim/blob/main/LICENSE
numba	https://numba.pydata.org	https://github.com/numba/numba/blob/main/LICENSE
numcpus	https://github.com/tklauser/numcpus	https://github.com/tklauser/numcpus/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
numexpr	https://github.com/pydata/numexpr	https://github.com/pydata/numexpr/blob/master/LICENSE.txt
numpy	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpy-base	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpydoc	https://numpydoc.readthedocs.io/en/latest/index.html	https://github.com/numpy/numpydoc/blob/main/LICENSE.txt
nvidia-cublas-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cublas-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cudnn-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html
nvidia-cudnn-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-cuEquivariance	https://github.com/NVIDIA/cuEquivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
nvidia-cufft-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufft-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufile-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html#
nvidia-curand-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-curand-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparselt-cu12	https://docs.nvidia.com/cuda/cusparselt/index.html	https://docs.nvidia.com/cuda/cusparselt/license.html
nvidia-ml-py	https://pypi.org/project/nvidia-ml-py/	https://pypi.org/project/nvidia-ml-py/
nvidia-nccl-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nccl-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nvjitlink-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-nvshmem-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
Oat++	https://oatpp.io/	https://github.com/oatpp/oatpp/blob/master/LICENSE
oauthlib	https://github.com/oauthlib/oauthlib	https://github.com/oauthlib/oauthlib/blob/master/LICENSE
ocl-icd	https://github.com/OCL-dev/ocl-icd	https://github.com/OCL-dev/ocl-icd/blob/master/COPYING
ocl-icd-system	https://github.com/conda-forge/ocl-icd-system-feedst	https://github.com/conda-forge/ocl-icd-system-feedst blob/main/LICENSE.txt
olefile	https://www.decalage.info/python/olefileio	https://olefile.readthedocs.io/en/latest/License.html
omegaconf	https://github.com/omry/omegaconf	https://github.com/omry/omegaconf/blob/master/LICENSE
OmegaFold	https://github.com/HeliXonProtein/OmegaFold/tree/main	https://github.com/HeliXonProtein/OmegaFold/blob/main/LICENSE
omnicanvas	https://omnicanvas.readthedocs.io/en/latest/	https://github.com/samirelanduk/omnicanvas/blob/master/LICENSE
OpenFF	https://openforcefield.org/	https://github.com/openforcefield/openff-forcefields
openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openff-forcefields	https://openforcefield.org	https://github.com/openforcefield/openff-forcefields/blob/master/LICENSE
openff-fragmenter	https://github.com/openforcefield/openff-fragmenter	https://github.com/openforcefield/openff-fragmenter/blob/main/LICENSE
openff-interchange	https://github.com/openforcefield/openff-interchange	https://github.com/openforcefield/openff-interchange/blob/main/LICENSE
openff-models	https://github.com/openforcefield/openff-models	https://github.com/openforcefield/openff-models/blob/main/LICENSE
openff-nagl	https://docs.openforcefield.org/projects/nagl/en/latest/?badge=latest	https://github.com/openforcefield/openff-nagl/blob/main/LICENSE
openff-nagl-models	https://github.com/openforcefield/openff-nagl-models	https://github.com/openforcefield/openff-nagl-models/blob/main/LICENSE
openff-toolkit	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-toolkit-base	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-units	https://github.com/openforcefield/openff-units	https://github.com/openforcefield/openff-units/blob/main/LICENSE
openff-utilities	https://github.com/openforcefield/openff-utilities	https://github.com/openforcefield/openff-utilities/blob/main/LICENSE
openjpeg	https://github.com/uclouvain/openjpeg	https://github.com/uclouvain/openjpeg/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openldap	https://www.openldap.org/software/repo.html	https://git.openldap.org/openldap/openldap/-/blob/master/LICENSE
OpenMM	https://openmm.org	https://docs.openmm.org/7.1.0/userguide/library.html#license
openmmtools	https://github.com/choderalab/openmmtools	https://github.com/choderalab/openmmtools/blob/main/LICENSE
openmoltools	https://github.com/choderalab/openmoltools	https://github.com/choderalab/openmoltools/blob/master/LICENSE
openpyxl	https://openpyxl.readthedocs.io/en/stable/	https://foss.heptapod.net/openpyxl/openpyxl/-/blob/branch/3.0/LICENCE.rst
openssl	https://www.openssl.org	https://github.com/openssl/openssl/blob/master/LICENSE.txt
opt-einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/master/LICENSE
opt_einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/main/LICENSE
OptKing	https://github.com/psi-rking/optking	https://github.com/psi-rking/optking/blob/master/LICENSE
optree	https://optree.readthedocs.io/en/latest/	https://github.com/metaopt/optree/blob/main/LICENSE
oscrypto	https://github.com/wbond/oscrypto	https://github.com/wbond/oscrypto/blob/master/LICENSE
overrides	https://github.com/mkorpela/overrides	https://github.com/mkorpela/overrides/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
packaging	https://github.com/pypa/packaging	https://github.com/pypa/packaging/blob/main/LICENSE
packmol	https://leandro.iqm.unicamp.br/m3g/packmol/home.shtml	https://github.com/m3g/packmol/blob/master/LICENSE
packmol_memgen	http://cpclab.uni-duesseldorf.de/	https://gitlab.com/cbjh/memgen/py-memgen/-/blob/main/LICENSE
pandas	https://pandas.pydata.org	https://github.com/pandas-dev/pandas/blob/main/LICENSE
pandocfilters	http://github.com/jgm/pandocfilters	https://github.com/jgm/pandocfilters/blob/master/LICENSE
panedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pango	https://pango.gnome.org	https://gitlab.gnome.org/GNOME/pango/-/blob/main/COPYING
ParmEd	https://parmed.github.io/ParmEd/html/index.html	https://github.com/ParmEd/ParmEd#license
parser	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
parso	https://parso.readthedocs.io/en/latest/	https://github.com/davidhalter/parso/blob/master/LICENSE.txt
pathos	https://github.com/uqfoundation/pathos	https://github.com/uqfoundation/pathos/blob/master/LICENSE
patsy	https://patsy.readthedocs.io/en/latest/	https://github.com/pydata/patsy/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pbkdf2	https://golang.org/x/crypto/pbkdf2	https://cs.opensource.google/go/x/crypto/+/master:LICENSE
pbr	https://docs.openstack.org/pbr/latest/	https://opendev.org/openstack/pbr/src/branch/master/LICENSE
pcmsolver	https://pcmsolver.readthedocs.io/en/stable/	https://github.com/PCMSolver/pcmsolver/blob/master/LICENSE
pcre	https://www.pcre.org	https://www.pcre.org/licence.txt
pcre2	https://www.pcre.org	https://github.com/PCRE2Project/pcre2/blob/master/LICENCE
pdb4amber	https://github.com/Amber-MD/pdb4amber	https://github.com/Amber-MD/pdb4amber/blob/master/LICENSE
pdbfixer	https://github.com/openmm/pdbfixer	https://github.com/openmm/pdbfixer/blob/master/LICENSE
pexpect	https://pexpect.readthedocs.io/	https://github.com/pexpect/pexpect/blob/master/LICENSE
pgconn	https://github.com/jackc/pgconn	https://github.com/jackc/pgconn/blob/master/LICENSE
pgio	https://github.com/jackc/pgio	https://github.com/jackc/pgio/blob/master/LICENSE
pgpassfile	https://github.com/jackc/pgpassfile	https://github.com/jackc/pgpassfile/blob/master/LICENSE
pgproto3	https://github.com/jackc/pgproto3/v2	https://github.com/jackc/pgproto3/blob/master/LICENSE
pgtype	https://github.com/jackc/pgtype	https://github.com/jackc/pgtype/blob/master/LICENSE
pgx	https://github.com/jackc/pgx/v4	https://github.com/jackc/pgx/blob/v4.0.1/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
phonopy	https://github.com/phonopy/phono3py	https://github.com/phonopy/phono3py/blob/develop/LICENSE
pickleshare	https://github.com/pickleshare/pickleshare	https://github.com/pickleshare/pickleshare/blob/master/LICENSE
Pillow	https://python-pillow.org	https://github.com/python-pillow/Pillow/blob/main/LICENSE
Pint	https://pint.readthedocs.io/en/stable/	https://github.com/hgrecco/pint/blob/master/LICENSE
pip	https://pip.pypa.io/	https://github.com/pypa/pip/blob/main/LICENSE.txt
pip-licenses	https://github.com/raimon49/pip-licenses	https://github.com/raimon49/pip-licenses/blob/master/LICENSE
pixman	https://pixman.org	https://cgit.freedesktop.org/pixman/tree/COPYING
pkginfo	https://launchpad.net/pkginfo	https://bazaar.launchpad.net/~tseaver/pkginfo/trunk/view/head:/LICENSE.txt
pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
pkgutil_resolve_name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
platformdirs	https://github.com/platformdirs/platformdirs	https://github.com/platformdirs/platformdirs/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
plotly	https://plotly.com/python/	https://github.com/plotly/plotly.py/blob/master/LICENSE.txt
plotly-orca	https://github.com/plotly/orca	https://github.com/plotly/orca/blob/master/LICENSE
plotly.js	https://github.com/plotly/plotly.js	https://github.com/plotly/plotly.js/raw/master/LICENSE
pluggy	https://pluggy.readthedocs.io/en/stable/index.html	https://github.com/pytest-dev/pluggy/blob/main/LICENSE
pooch	https://github.com/fatiando/pooch	https://github.com/fatiando/pooch/blob/main/LICENSE.txt
portalocker	https://github.com/WoLpH/portalocker	https://github.com/wolph/portalocker/blob/develop/LICENSE
pox	https://github.com/uqfoundation/pox	https://github.com/uqfoundation/pox/blob/master/LICENSE
ppft	https://github.com/uqfoundation/ppft	https://github.com/uqfoundation/ppft/blob/master/LICENSE
pq	https://github.com/lib/pq	https://github.com/lib/pq/blob/master/LICENSE.md
ProDy	http://www.csb.pitt.edu/ProDy	https://github.com/prody/ProDy/blob/master/LICENSE.rst
prometheus-client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prometheus_client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prompt-toolkit	https://python-prompt-toolkit.readthedocs.io/en/stable/	https://github.com/prompt-toolkit/python-prompt-toolkit/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
prompt_toolkit	https://github.com/prompt-toolkit/python-prompt-toolki	https://github.com/prompt-toolkit/python-prompt-toolki tab=BSD-3-Clause-1-ov-f readme
propcache	https://github.com/aio-libs/propcache	https://github.com/aio-libs/propcache?tab=Apache-2.0-1-ov-file#readme
protobuf	https://google.golang.org/protobuf	https://github.com/protocolbuffers/protobuf-go/blob/master/LICENSE
psi4	https://psicode.org	https://github.com/psi4/psi4/blob/master/COPYING.LESSER
psutil	https://psutil.readthedocs.io/en/latest/	https://github.com/giampaolo/psutil/blob/master/LICENSE
psycopg2	https://psycopg.org/	https://github.com/psycopg/psycopg2/blob/master/LICENSE
PTable	https://github.com/kxxoling/PTable	https://github.com/kxxoling/PTable/blob/master/COPYING
pthread-stubs	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-pthread-stubs/blob/master/COPYING
ptyprocess	https://ptyprocess.readthedocs.io/en/latest/	https://github.com/pexpect/ptyprocess/blob/master/LICENSE
pure-eval	http://github.com/alexmojaki/pure_eval	http://github.com/alexmojaki/pure_eval/blob/master/LICENSE.txt
pure_eval	http://github.com/alexmojaki/pure_eval	https://github.com/alexmojaki/pure_eval?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
py	https://py.readthedocs.io/en/latest/	https://github.com/pytest-dev/py/blob/master/LICENSE
py-cpuinfo	https://github.com/workhorsy/py-cpuinfo	https://github.com/workhorsy/py-cpuinfo/blob/master/LICENSE
pyasn1	https://github.com/etingof/pyasn1	https://github.com/etingof/pyasn1/blob/master/LICENSE.rst
pyasn1-modules	https://github.com/etingof/pyasn1-modules	https://github.com/etingof/pyasn1-modules/blob/master/LICENSE.txt
pyasn1_modules	https://pyasn1.readthedocs.io/en/latest/contents.html	https://github.com/pyasn1/pyasn1-modules?tab=BSD-2-Clause-1-ov-file#readme
pybind11	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11-abi	https://github.com/pybind/pybind11	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11_global	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pycairo	https://pycairo.readthedocs.io/en/latest/index.html	https://github.com/pygobject/pycairo/blob/master/COPYING
pycosat	https://github.com/conda/pycosat	https://github.com/conda/pycosat/blob/main/LICENSE
pycparser	https://github.com/eliben/pycparser	https://github.com/eliben/pycparser/blob/master/LICENSE
pydantic	https://pydantic-docs.helpmanual.io	https://github.com/samuelcolvin/pydantic/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pydantic-core	https://github.com/pydantic/pydantic-core	https://github.com/pydantic/pydantic-core
pyedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pyEMMA	http://github.com/markovmodel/PyEMMA	https://github.com/markovmodel/PyEMMA/blob/devel/LICENSE.txt
Pygments	https://pygments.org	https://github.com/pygments/pygments/blob/master/LICENSE
pygments	https://pygments.org/	https://github.com/pygments/pygments?tab=BSD-2-Clause-1-ov-file#readme
pygraphviz	https://pygraphviz.github.io	https://github.com/pygraphviz/pygraphviz/blob/main/LICENSE
pygtop	https://pygtop.readthedocs.io/en/latest/	https://github.com/samirelanduk/pygtop/blob/master/LICENSE
pyHanko	https://github.com/MatthiasValvekens/pyHanko	https://github.com/MatthiasValvekens/pyHanko/blob/master/LICENSE
pyhanko-certvalidator	https://github.com/MatthiasValvekens/certvalidator	https://github.com/MatthiasValvekens/certvalidator/blob/master/LICENSE
PyJWT	https://github.com/jpadilla/pyjwt	https://github.com/jpadilla/pyjwt/blob/master/LICENSE
pymbar	https://pymbar.org	https://github.com/choderalab/pymbar/blob/master/LICENSE
pynvml	https://github.com/gpuopenanalytics/pynvml	https://github.com/gpuopenanalytics/pynvml/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pyOpenSSL	https://pyopenssl.org/	https://github.com/pyca/pyopenssl/blob/main/LICENSE
pyparsing	https://pyparsing-docs.readthedocs.io/en/latest/	https://github.com/pyparsing/pyparsing/blob/master/LICENSE
PyPDF3	https://github.com/sfneal/PyPDF3	https://github.com/sfneal/PyPDF3/blob/master/LICENSE
pyrsistent	http://github.com/tobgu/pyrsistent/	https://github.com/tobgu/pyrsistent/blob/master/LICENSE.mit
pysam	https://github.com/pysam-developers/pysam	https://github.com/pysam-developers/pysam/blob/master/COPYING
PySocks	https://github.com/Anorov/PySocks	https://github.com/Anorov/PySocks/blob/master/LICENSE
pytables	https://www.pytables.org	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
python	https://www.python.org/	https://docs.python.org/3/license.html
python-bidi	https://github.com/MeirKriheli/python-bidi	https://github.com/MeirKriheli/python-bidi/blob/master/COPYING
python-constraint	https://github.com/python-constraint/python-constraint	https://github.com/python-constraint/python-constraint/blob/master/LICENSE
python-dateutil	https://dateutil.readthedocs.io	https://github.com/dateutil/dateutil/blob/master/LICENSE
python-json-logger	http://github.com/madzak/python-json-logger	https://github.com/madzak/python-json-logger/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
python-pptx	https://github.com/scanny/python-pptx	https://github.com/scanny/python-pptx/blob/master/LICENSE
python-ldap	https://www.python-ldap.org/	https://github.com/python-ldap/python-ldap/blob/python-ldap-3.4.0/LICENCE
python3-saml	https://github.com/onelogin/python3-saml	https://github.com/onelogin/python3-saml/blob/master/LICENSE
python_abi	https://github.com/conda-forge/python_abi-feedstock	https://github.com/conda-forge/python_abi-feedstock/blob/main/LICENSE.txt
pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning/blob/master/LICENSE
pytz	https://pythonhosted.org/pytz	https://git.launchpad.net/pytz/tree/LICENSE.txt
pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shin	https://github.com/pganssle/pytz-deprecation-shin blob/master/LICENSE
PyWavelets	https://github.com/PyWavelets/pywt	https://github.com/PyWavelets/pywt/blob/master/LICENSE
PyYAML	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
pyyml	No longer available	No longer available
pyzmq	https://pyzmq.readthedocs.io/en/latest/	https://github.com/zeromq/pyzmq/blob/main/COPYING.BSD
qcelemental	https://github.com/MolSSI/QCElemental	https://github.com/MolSSI/QCElemental/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
qcengine	https://github.com/MolSSI/QCEngine	https://github.com/MolSSI/QCEngine/blob/master/LICENSE
qrcode	https://github.com/lincolnloop/python-qrcode	https://github.com/lincolnloop/python-qrcode/blob/main/LICENSE
ramda	https://github.com/ramda/ramda	https://github.com/ramda/ramda/raw/master/LICENSE.txt
rapidjson	https://rapidjson.org/	https://github.com/Tencent/rapidjson/blob/master/license.txt
rdkit	https://www.rdkit.org	https://github.com/rdkit/rdkit/blob/master/license.txt
re2	https://github.com/google/re2	https://github.com/google/re2/blob/main/LICENSE
readme-renderer	https://github.com/pypa/readme_renderer	https://github.com/pypa/readme_renderer/blob/main/LICENSE
redis-py	https://github.com/andymccurdy/redis-py	https://github.com/redis/redis-py/blob/master/LICENSE
referencing	https://github.com/python-jsonschema/referencing	https://github.com/python-jsonschema/referencing
regex	https://github.com/mrabarnett/mrab-regex	https://github.com/mrabarnett/mrab-regex/blob/hg/LICENSE.txt
reportlab	https://www.reportlab.com	https://www.reportlab.com/dev/docs/faq/#1.3
reproc	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE
reproc-cpp	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
requests	https://requests.readthedocs.io	https://github.com/psf/requests/blob/main/LICENSE
requests-oauthlib	https://github.com/requests/requests-oauthlib	https://github.com/requests/requests-oauthlib/blob/master/LICENSE
requests-toolbelt	https://toolbelt.readthedocs.org	https://github.com/requests/toolbelt/blob/master/LICENSE
resumable	https://github.com/stevvooe/resumable	https://github.com/stevvooe/resumable/blob/master/LICENSE
retrying	https://github.com/rholder/retrying	https://github.com/rholder/retrying/blob/master/LICENSE
rfc3339-validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator/blob/master/LICENSE
rfc3339_validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator?tab=MIT-1-ov-file#readme
rfc3986	https://rfc3986.readthedocs.io/en/latest/	https://github.com/python-hyper/rfc3986/blob/main/LICENSE
rfc3986-validator	https://github.com/naimetti/rfc3986-validator	https://github.com/naimetti/rfc3986-validator/blob/master/LICENSE
rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax?tab=MIT-1-ov-file#readme
rich	https://github.com/Textualize/rich	https://github.com/Textualize/rich/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
rich-argparse	https://github.com/hamdanal/rich-argparse	https://github.com/hamdanal/rich-argparse/blob/main/LICENSE
rlPyCairo	https://pypi.org/project/rlPyCairo/	https://hg.reportlab.com/hg-public/rlPyCairo/file/tip/rlPyCairo/LICENSE.txt
RocksDict	https://github.com/rocksdict/RocksDict	https://github.com/rocksdict/RocksDict?tab=MIT-1-ov-file#readme
roman-numerals	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
roman-numerals-py	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
rotary-embedding-torch	https://github.com/lucidrains/rotary-embedding-tor	https://github.com/lucidrains/rotary-embedding-tor blob/main/LICENSE
rpds-py	https://github.com/crate-py/rpds	https://github.com/crate-py/rpds
rpmpack	https://github.com/google/rpmpack	https://github.com/google/rpmpack/blob/master/LICENSE
rsa	https://stuvel.eu/rsa	https://github.com/sybrenstuvel/python-rsa/blob/main/LICENSE
ruamel-yaml	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/LICENSE
ruamel.yaml.clib	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
s3transfer	https://github.com/boto/s3transfer	https://github.com/boto/s3transfer/blob/develop/LICENSE.txt
sacrebleu	https://github.com/mjpost/sacrebleu	https://github.com/mjpost/sacrebleu/blob/master/LICENSE.txt
sapiens	https://github.com/Merck/Sapiens	https://github.com/Merck/Sapiens/blob/main/LICENSE
sasl	https://mellium.im/sasl	https://github.com/mellium/sasl/blob/main/LICENSE
scikit-gstat	https://mmaelicke.github.io/scikit-gstat/	https://github.com/mmaelicke/scikit-gstat/blob/main/LICENSE
scikit-image	https://scikit-image.org	https://scikit-image.org/docs/dev/license.html
scikit-learn	https://scikit-learn.org/stable/	https://github.com/scikit-learn/scikit-learn/blob/main/COPYING
scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra/blob/main/LICENSE
scikit-network	https://github.com/sknetwork-team/scikit-network	https://github.com/sknetwork-team/scikit-network?tab=License-1-ov-file#readme
scipy	https://scipy.org	https://github.com/scipy/scipy/blob/main/LICENSE.txt
seaborn	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE
seaborn-base	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
semver	https://github.com/Masterminds/semver/v3	https://github.com/Masterminds/semver/blob/v3.0.3/LICENSE.txt
Send2Trash	https://github.com/arsenetar/send2trash	https://github.com/arsenetar/send2trash/blob/master/LICENSE
sentry-sdk	https://github.com/getsentry/sentry-python	https://github.com/getsentry/sentry-python/blob/master/LICENSE
setproctitle	https://github.com/dvarrazzo/py-setproctitle	https://github.com/dvarrazzo/py-setproctitle/blob/master/LICENSE
setuptools	https://github.com/pypa/setuptools	https://github.com/pypa/setuptools/blob/main/LICENSE
setuptools-scm	https://github.com/pypa/setuptools_scm/	https://github.com/pypa/setuptools_scm/blob/main/LICENSE
sh	https://github.com/amoffat/sh	https://github.com/amoffat/sh/blob/develop/LICENSE.txt
shellingham	https://github.com/sarugaku/shellingham	https://github.com/sarugaku/shellingham/blob/master/LICENSE
simdjson	https://github.com/simdjson/simdjson/tree/master	https://github.com/simdjson/simdjson/blob/master/LICENSE
simint	https://www.bennyp.org/research/simint/	https://github.com/simint-chem/simint-generator/blob/master/LICENSE
simplejson	https://github.com/simplejson/simplejson	https://github.com/simplejson/simplejson/blob/master/LICENSE.txt
six	https://github.com/benjaminp/six	https://github.com/benjaminp/six/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
smirnoff99frosst	https://github.com/openforcefield/smirnoff99frosst	https://github.com/openforcefield/smirnoff99Frosst/blob/master/LICENSE
smmap	https://github.com/gitpython-developers/smmap	https://github.com/gitpython-developers/smmap/blob/master/LICENSE
snappy	https://github.com/google/snappy	https://github.com/google/snappy/blob/main/COPYING
sniffio	https://github.com/python-trio/sniffio	https://github.com/python-trio/sniffio/blob/master/LICENSE.MIT
snowballstemmer	https://github.com/snowballstem/snowball	https://github.com/snowballstem/snowball/blob/master/COPYING
soupsieve	https://github.com/facelessuser/soupsieve	https://github.com/facelessuser/soupsieve/blob/main/LICENSE.md
spdlog	https://github.com/gabime/spdlog	https://github.com/gabime/spdlog?tab=License-1-ov-file#readme
spglib	https://spglib.readthedocs.io/	https://github.com/spglib/spglib/blob/develop/COPYING
sphinx	https://github.com/sphinx-doc/sphinx	https://github.com/sphinx-doc/sphinx/blob/5.x/LICENSE
sphinxcontrib-applehelp	https://github.com/sphinx-doc/sphinxcontrib-applehe	https://github.com/sphinx-doc/sphinxcontrib-applehe blob/master/LICENSE
sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp blob/master/LICENSE
sphinxcontrib-htmlhelp	https://github.com/sphinx-doc/sphinxcontrib-htmlhel	https://github.com/sphinx-doc/sphinxcontrib-htmlhel blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath/blob/master/LICENSE
sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp/blob/master/LICENSE
sphinxcontrib-serializinghtml	https://github.com/sphinx-doc/sphinxcontrib-serializ	https://github.com/sphinx-doc/sphinxcontrib-serializ blob/master/LICENSE
SQLAlchemy	https://www.sqlalchemy.org	https://docs.sqlalchemy.org/en/14/copyright.html
SQLAlchemy	https://www.sqlalchemy.org/	https://github.com/sqlalchemy/sqlalchemy?tab=MIT-1-ov-file#readme
sqlite	https://sqlite.org/index.html	https://sqlite.org/copyright.html
sqlparse	https://github.com/andialbrecht/sqlparse	https://github.com/andialbrecht/sqlparse/blob/master/LICENSE
stack-data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data/blob/master/LICENSE.txt
stack_data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data?tab=MIT-1-ov-file#readme
starfile	https://github.com/alisterburt/starfile	https://github.com/teamtomo/starfile/blob/main/LICENSE
statsmodels	https://github.com/statsmodels/statsmodels	https://github.com/statsmodels/statsmodels/blob/main/LICENSE.txt
structlog	https://www.structlog.org/	https://github.com/hynek/structlog/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
streamlit	https://github.com/streamlit/streamlit	https://github.com/streamlit/streamlit/blob/develop/LICENSE
svglib	https://github.com/deeplook/svglib	https://github.com/deeplook/svglib/blob/master/LICENSE.txt
sympy	https://sympy.org	https://github.com/sympy/sympy/blob/master/LICENSE
tables	https://www.pytables.org/	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
tabulate	https://github.com/astanin/python-tabulate	https://github.com/astanin/python-tabulate/blob/master/LICENSE
tbb	https://github.com/oneapi-src/oneTBB	https://github.com/oneapi-src/oneTBB/blob/master/LICENSE.txt
tenacity	https://github.com/jd/tenacity	https://github.com/jd/tenacity/blob/master/LICENSE
tensorboard	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-data-server	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-plugin-wit	https://github.com/pair-code/what-if-tool	https://github.com/PAIR-code/what-if-tool/blob/master/LICENSE
tensorboardX	https://github.com/lanpa/tensorboardX	https://github.com/lanpa/tensorboardX/blob/master/LICENSE
tensorflow	https://github.com/tensorflow/tensorflow	https://github.com/tensorflow/tensorflow/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tensorflow-estimator	https://github.com/tensorflow/estimator	https://github.com/tensorflow/estimator/blob/master/LICENSE
tensorflow-io-gcs-filesystem	https://github.com/tensorflow/io	https://github.com/tensorflow/io/blob/master/LICENSE
tensorflow-probability	https://github.com/tensorflow/probability	https://github.com/tensorflow/probability/blob/main/LICENSE
termcolor	https://github.com/hugovk/termcolor	https://github.com/hugovk/termcolor/blob/main/COPYING.txt
terminado	https://github.com/jupyter/terminado	https://github.com/jupyter/terminado/blob/main/LICENSE
testpath	https://github.com/jupyter/testpath	https://github.com/jupyter/testpath/blob/master/LICENSE
textangular	https://github.com/fraywing/textAngular	https://github.com/fraywing/textAngular
texttable	https://github.com/foutaise/texttable/	https://github.com/foutaise/texttable/?tab=MIT-1-ov-file#readme
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
threadpoolctl	https://github.com/joblib/threadpoolctl	https://github.com/joblib/threadpoolctl/blob/master/LICENSE
three	https://github.com/mrdoob/three.js	https://github.com/mrdoob/three.js/raw/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tiffile	https://github.com/cgohlke/tiffile/	https://github.com/cgohlke/tiffile/blob/master/LICENSE
tinycss2	https://github.com/Kozea/tinycss2/	https://github.com/Kozea/tinycss2/blob/master/LICENSE
tinyxml2	https://github.com/leethomason/tinyxml2	https://github.com/leethomason/tinyxml2/blob/master/LICENSE.txt
tk	https://www.tcl.tk/	https://www.tcl.tk/software/tcltk/license.html
toml	https://github.com/toml-lang/toml	https://github.com/toml-lang/toml/blob/main/LICENSE
tomli	https://github.com/hukkin/tomli	https://github.com/hukkin/tomli/blob/master/LICENSE
toolz	https://github.com/pytoolz/toolz	https://github.com/pytoolz/toolz/blob/master/LICENSE.txt
torch	https://pytorch.org/	https://pypi.org/project/torch/#license
torch-geometric	https://pyg.org/	https://github.com/pyg-team/pytorch_geometric?tab=MIT-1-ov-file#readme
torcheval	https://github.com/pytorch/torcheval	https://github.com/pytorch/torcheval?tab=License-1-ov-file#readme
torchmetrics	https://github.com/Lightning-AI/torchmetrics	https://github.com/Lightning-AI/torchmetrics?tab=Apache-2.0-1-ov-file#readme
tornado	https://www.tornadoweb.org	https://github.com/tornadoweb/tornado/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL		
tqdm	https://github.com/tqdm/tqdm	https://github.com/tqdm/tqdm/blob/master/LICENCE		
tracy	https://github.com/gear-genomics/tracy	https://github.com/gear-genomics/tracy/blob/main/LICENSE		
traitlets	https://github.com/ipython/traitlets	https://github.com/ipython/traitlets/blob/main/COPYING.md		
triton	https://github.com/openai/triton/	https://github.com/openai/triton/blob/main/LICENSE		
truststore	https://github.com/sethmlarson/trus	https://github.com/sethmlarson/truststore/blob/main/LICENSE		
truststore	https://github.com/sethmlarson/truststore	https://github.com/sethmlarson/truststore/blob/main/LICENSEtruststore	https://github.com/sethmlarson/truststore	https://github.com/sethmlarson/truststore/blob/main/LICENSE
ts-jest	https://github.com/kulshekhar/ts-jest	https://github.com/kulshekhar/ts-jest/raw/master/LICENSE.md		
ts-loader	https://github.com/TypeStrong/ts-loader	https://github.com/TypeStrong/ts-loader/raw/master/LICENSE		
twine	https://github.com/pypa/twine	https://github.com/pypa/twine/blob/main/LICENSE		
twinj uuid	https://github.com/twinj/uuid	https://github.com/twinj/uuid/blob/master/LICENSE		
types	https://github.com/babel/babel	https://github.com/babel/babel/raw/master/LICENSE		
types-python-dateutil	https://github.com/python/typeshed	https://github.com/python/typeshed?tab=License-1-ov-file#readme		
types-requests	https://pypi.org/project/types-requests/	https://github.com/python/typeshed/blob/main/LICENSE		

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
typescript	https://github.com/Microsoft/TypeScript	https://github.com/Microsoft/TypeScript/raw/master/LICENSE.txt
typing-inspection	https://github.com/pydantic/typing-inspection	https://github.com/pydantic/typing-inspection?tab=MIT-1-ov-file#readme
typing-utils	https://github.com/bojiang/typing_utils	https://github.com/bojiang/typing_utils?tab=Apache-2.0-1-ov-file#readme
typing_extensions	https://github.com/python/typing	https://github.com/python/typing/blob/master/LICENSE
typing_utils	https://github.com/bojiang/typing_utils	https://github.com/bojiang/typing_utils?tab=Apache-2.0-1-ov-file#readme
tzdata	https://github.com/python/tzdata	https://github.com/python/tzdata/blob/master/LICENSE
tzlocal	https://github.com/regebro/tzlocal	https://github.com/regebro/tzlocal/blob/master/LICENSE.txt
umf	https://github.com/oneapi-src/unified-memory-fram	https://github.com/oneapi-src/unified-memory-fram tab=License-1-ov-file#readme
umi-tools	https://github.com/CGATOxford/UMI-tools	https://github.com/CGATOxford/UMI-tools/blob/master/LICENSE
unicodedata2	https://github.com/fonttools/unicodedata2	https://github.com/fonttools/unicodedata2/blob/master/LICENSE
uri-template	https://gitlab.linss.com/open-source/python/uri-template	https://gitlab.linss.com/open-source/python/uri-template/-/blob/main/LICENSE
uritools	https://github.com/tkem/uritools/	https://github.com/tkem/uritools/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL		
urllib3	https://urllib3.readthedocs.io/	https://github.com/urllib3/urllib3/blob/main/LICENSE.txt		
vine	https://github.com/celery/vine	https://github.com/celery/vine/blob/master/LICENSE		
vue	https://github.com/vuejs/vue	https://github.com/vuejs/vue/raw/master/LICENSE		
wadler_lindig	https://github.com/patrick-kidger/wadler_lindig	https://github.com/patrick-kidger/wadler_lindig?tab=Apache-2.0-1-ov-file#readme		
wadler_lindig	https://github.com/patrick-kidger/wadler_lindig	https://github.com/patrick-kidger/wadler_lindig?tab=Apache-2.0-1-ov-file#readmewcwidth	https://github.com/jquast/wcwidth	https://github.com/jquast/wcwidth/blob/master/LICENSE
wandb	https://github.com/wandb/wandb	https://github.com/wandb/wandb/blob/main/LICENSE		
webcolors	https://github.com/ubernostrum/webcolors	https://github.com/ubernostrum/webcolors?tab=BSD-3-Clause-1-ov-file#readme		
webencodings	https://github.com/gsnedders/python-webencodings	https://github.com/gsnedders/python-webencodings/blob/master/LICENSE		
websocket-client	https://github.com/websocket-client/websocket-client.git	https://github.com/websocket-client/websocket-client/blob/master/LICENSE		
Werkzeug	https://palletsprojects.com/p/werkzeug/	https://werkzeug.palletsprojects.com/en/2.1.x/license/		
westpa	http://github.com/westpa/westpa	https://github.com/westpa/westpa/blob/westpa2/LICENSE		
wheel	https://github.com/pypa/wheel	https://github.com/pypa/wheel/blob/main/LICENSE.txt		

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
widgetsnbextension	https://github.com/jupyter-widgets/ipywidgets#readme	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE
wrapt	https://github.com/GrahamDumpleton/wrapt	https://github.com/GrahamDumpleton/wrapt/blob/develop/LICENSE
wsutil	https://github.com/yhat/wsutil	https://github.com/yhat/wsutil/blob/master/LICENSE
x/lint	https://golang.org/x/lint	https://github.com/golang/lint/blob/master/LICENSE
x/mod	https://golang.org/x/mod/semver	https://cs.opensource.google/go/x/mod/+/master:LICENSE
x/net	https://golang.org/x/net	https://cs.opensource.google/go/x/net/+/master:LICENSE
x/oauth2	https://golang.org/x/oauth2	https://cs.opensource.google/go/x/oauth2/+/master:LICENSE
x/sys	https://golang.org/x/sys	https://cs.opensource.google/go/x/sys/+/master:LICENSE
x/text	https://golang.org/x/text	https://cs.opensource.google/go/x/text/+/master:LICENSE
x/tools	https://golang.org/x/tools	https://cs.opensource.google/go/x/tools/+/master:LICENSE
x/xerrors	https://golang.org/x/xerrors	https://cs.opensource.google/go/x/xerrors/+/master:LICENSE
xgboost	https://github.com/dmlc/xgboost	https://github.com/dmlc/xgboost/blob/master/LICENSE
xhtml2pdf	http://github.com/xhtml2pdf/xhtml2pdf	https://github.com/xhtml2pdf/xhtml2pdf/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
xlrd	https://github.com/python-excel/xlrd	https://github.com/python-excel/xlrd/blob/master/LICENSE
xlsxwriter	https://github.com/jmcnamara/XlsxWriter	https://github.com/jmcnamara/XlsxWriter/blob/main/LICENSE.txt
xmlsec	https://github.com/mehcode/python-xmlsec	https://github.com/mehcode/python-xmlsec/blob/master/LICENSE
xmltodict	https://github.com/martinblech/xmltodict	https://github.com/martinblech/xmltodict/blob/master/LICENSE
xorg-kbproto	https://gitlab.freedesktop.org/xorg/proto/kbproto	https://gitlab.freedesktop.org/xorg/proto/kbproto/-/blob/master/COPYING
xorg-libice	https://gitlab.freedesktop.org/xorg/lib/libice	https://gitlab.freedesktop.org/xorg/lib/libice/-/blob/master/COPYING
xorg-libsm	https://gitlab.freedesktop.org/xorg/lib/libsm	https://gitlab.freedesktop.org/xorg/lib/libsm/-/blob/master/COPYING
xorg-libx11	https://gitlab.freedesktop.org/xorg/lib/libx11	https://gitlab.freedesktop.org/xorg/lib/libx11/-/blob/master/COPYING
xorg-libxau	https://gitlab.freedesktop.org/xorg/lib/libxau	https://gitlab.freedesktop.org/xorg/lib/libxau/-/blob/master/COPYING
xorg-libxdmcp	https://gitlab.freedesktop.org/xorg/lib/libxdmcp	https://gitlab.freedesktop.org/xorg/lib/libxdmcp/-/blob/master/COPYING

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
xorg-libxext	https://gitlab.freedesktop.org/xorg/lib/libxext	https://gitlab.freedesktop.org/xorg/lib/libxext/-/blob/master/COPYING
xorg-libxrender	https://gitlab.freedesktop.org/xorg/lib/libxrender	https://gitlab.freedesktop.org/xorg/lib/libxrender/-/blob/master/COPYING
xorg-libxt	https://gitlab.freedesktop.org/xorg/lib/libxt	https://gitlab.freedesktop.org/xorg/lib/libxt/-/blob/master/COPYING
xorg-renderproto	https://gitlab.freedesktop.org/xorg/proto/renderproto	https://gitlab.freedesktop.org/xorg/proto/renderproto/-/blob/master/COPYING
xorg-xextproto	https://gitlab.freedesktop.org/xorg/proto/xextproto	https://gitlab.freedesktop.org/xorg/proto/xextproto/-/blob/master/COPYING
xorg-xproto	https://gitlab.freedesktop.org/xorg/proto/xproto	https://gitlab.freedesktop.org/xorg/proto/xproto/-/blob/master/COPYING
xxhash	https://github.com/cespare/xxhash/v2	https://github.com/cespare/xxhash/blob/main/LICENSE.txt
xz	https://github.com/ulikunitz/xz	https://github.com/ulikunitz/xz/blob/master/LICENSE
yaml	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
yaml-cpp	https://github.com/jbeder/yaml-cpp	https://github.com/jbeder/yaml-cpp/blob/master/LICENSE
yaml.v2	https://gopkg.in/yaml.v2	https://github.com/go-yaml/yaml/blob/v2.4.0/LICENSE
yaml.v3	https://gopkg.in/yaml.v3	https://github.com/go-yaml/yaml/blob/v3/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
yarl	https://github.com/aio-libs/yarl/	https://github.com/aio-libs/yarl/blob/master/LICENSE
yaspin	https://github.com/pavdmyt/yaspin	https://github.com/pavdmyt/yaspin/blob/master/LICENSE
yfiles	https://www.yworks.com/products/yfiles	commercial license
yml	https://pypi.org/project/yml/	N/A
zap	https://go.uber.org/zap	https://github.com/uber-go/zap/blob/master/LICENSE.txt
zipp	https://github.com/jaraco/zipp	https://github.com/jaraco/zipp/blob/main/LICENSE
zlib	https://zlib.net/	https://zlib.net/zlib_license.html
zstandard	https://github.com/indygreg/python-zstandard	https://github.com/indygreg/python-zstandard/blob/main/LICENSE
zstd	https://facebook.github.io/zstd/	https://github.com/facebook/zstd/blob/dev/LICENSE
click-option-group	https://github.com/click-contrib/click-option-group	https://github.com/click-contrib/click-option-group/blob/master/LICENSE

10.1 GCC

On the Linux platform, The OpenEye toolkits are linked with the GCC libraries using an eligible compilation process under the terms of the GCC RUNTIME LIBRARY EXCEPTION (see below).

The OpenEye toolkits are NOT open source software and are NOT governed by GPL or other open source licenses. Your right to use the toolkits are governed by the OpenEye license which is binding to You.

For certain Linux distributions, GCC library components are included as object files. Your rights to use GCC library components (libstdc++, libgcc, and libgomp) included with these distributions are governed by the GPLv3 license as follows:

The components libstdc++, libgcc, and libgomp are part of GCC.

- GCC is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
- GCC is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the

implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with GCC (see below). If not, see Licenses GNU Project .

Per GPLv3, section 6d, the source for GCC is available for download from the following location: GNU. The GCC library components included here are from version gcc-4.9.3 and are built using the standard instructions at Installing GCC.

10.1.1 GCC RUNTIME LIBRARY EXCEPTION

```
Version 3.1, 31 March 2009

Copyright © 2009 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document,␣
↪but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional permission under␣
↪section 7 of the GNU General Public License,
version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a␣
↪notice placed by the copyright holder
of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header␣
↪files and runtime libraries with
the compiled program. The purpose of this Exception is to allow compilation of non-GPL␣
↪(including proprietary) programs
to use, in this way, the header files and runtime libraries covered by this Exception.

0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for␣
↪execution after a Compilation Process,
or makes use of an interface provided by the Runtime Library, but is not otherwise based␣
↪on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications,␣
↪governed by version 3 (or a specified
later version) of the GNU General Public License (GPL) with the option of using any␣
↪subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and␣
↪use would permit combination with
GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor␣
↪architecture, in executable
form or suitable for input to an assembler, loader, linker and/or execution phase.␣
↪Notwithstanding that, Target Code
does not include data in any format that is used as a compiler intermediate␣
↪representation, or used for producing a
compiler intermediate representation.
```

(continues on next page)

(continued from previous page)

```
The "Compilation Process" transforms code entirely represented in non-intermediate␣
↪languages designed for human-written
code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use␣
↪of source code generators and
preprocessors need not be considered part of the Compilation Process, since the␣
↪Compilation Process can be understood
as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-
↪compatible software, or if it is
done without using any work based on GCC. For example, using non-GPL-compatible Software␣
↪to optimize any GCC intermediate
representations would not qualify as an Eligible Compilation Process.

1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by combining the Runtime␣
↪Library with Independent Modules,
even if such propagation would otherwise violate the terms of GPLv3, provided that all␣
↪Target Code was generated by
Eligible Compilation Processes. You may then convey such a combination under terms of␣
↪your choice, consistent with the
licensing of the Independent Modules.

2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general presumption that third-
↪party software is unaffected by
the copyleft requirements of the license of GCC.
```

See also

- http://www.gnu.org/licenses/gcc-exception.html

10.1.2 GNU GENERAL PUBLIC LICENSE

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

 The GNU General Public License is a free, copyleft license for
 software and other kinds of works.

   The licenses for most software and other practical works are designed
```

(continues on next page)

(continued from previous page)

```
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

  Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

  For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

  Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
```

(continues on next page)

(continued from previous page)

```
patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and
modification follow.

                    TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.
```

(continues on next page)

(continued from previous page)

```
  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
```

(continues on next page)

(continued from previous page)

```
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.


  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

   a) The work must carry prominent notices stating that you modified
   it, and giving a relevant date.

   b) The work must carry prominent notices stating that it is
   released under this License and any conditions added under section
   7.  This requirement modifies the requirement in section 4 to
   "keep intact all notices".
```

(continues on next page)

(continued from previous page)

```
    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

  a) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by the
  Corresponding Source fixed on a durable physical medium
  customarily used for software interchange.

  b) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by a
  written offer, valid for at least three years and valid for as
  long as you offer spare parts or customer support for that product
  model, to give anyone who possesses the object code either (1) a
  copy of the Corresponding Source for all the software in the
  product that is covered by this License, on a durable physical
  medium customarily used for software interchange, for a price no
  more than your reasonable cost of physically performing this
  conveying of source, or (2) access to copy the
  Corresponding Source from a network server at no charge.

  c) Convey individual copies of the object code with a copy of the
  written offer to provide the Corresponding Source.  This
  alternative is allowed only occasionally and noncommercially, and
  only if you received the object code with such an offer, in accord
```

(continues on next page)

(continued from previous page)

```
   with subsection 6b.

   d) Convey the object code by offering access from a designated
   place (gratis or for a charge), and offer equivalent access to the
   Corresponding Source in the same way through the same place at no
   further charge.  You need not require recipients to copy the
   Corresponding Source along with the object code.  If the place to
   copy the object code is a network server, the Corresponding Source
   may be on a different server (operated by you or a third party)
   that supports equivalent copying facilities, provided you maintain
   clear directions next to the object code saying where to find the
   Corresponding Source.  Regardless of what server hosts the
   Corresponding Source, you remain obligated to ensure that it is
   available for as long as needed to satisfy these requirements.

   e) Convey the object code using peer-to-peer transmission, provided
   you inform other peers where the object code and Corresponding
   Source of the work are being offered to the general public at no
   charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
```

(continues on next page)

(continued from previous page)

```
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
```

(continues on next page)

(continued from previous page)

```
   authors of the material; or

   e) Declining to grant rights under trademark law for use of some
   trade names, trademarks, or service marks; or

   f) Requiring indemnification of licensors and authors of that
   material by anyone who conveys the material (or modified versions of
   it) with contractual assumptions of liability to the recipient, for
   any liability that these contractual assumptions directly impose on
   those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.
```

(continues on next page)

(continued from previous page)

```
  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
```

(continues on next page)

```
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
```

(continues on next page)

(continued from previous page)

```
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
```

(continues on next page)

(continued from previous page)

```
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
```

(continues on next page)

(continued from previous page)

```
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

See also

- http://www.gnu.org/licenses/gpl.txt

INDEX

A

AddFlexibleResidue
    OESz::OESzybkiProteinOptions, 50

B

boundentropy.cs
    Example Code, 123

C

Clear
    OESz::OESzybkiResults, 52
ClearFixAtoms
    OESz::OESzybki, 38
ClearFlexibleResidues
    OESz::OESzybkiProteinOptions, 50
ClearHarmonicConstraints
    OESz::OESzybki, 39
ClearTorsionConstraint
    OESz::OESzybki, 39
Constructor
    OESz::OEFixedProteinLigandOptimizer, 15
    OESz::OEFlexProteinLigandOptimizer, 16
    OESz::OELigandChargeOptions, 29
    OESz::OELigandEnergy, 83
    OESz::OELigandOptOptions, 85
    OESz::OELigandOptResults, 88
    OESz::OEProteinFlexOptions, 31
    OESz::OEProteinLigandOptOptions, 32
    OESz::OETorConstraintOptimizer, 90
    OESz::OETorConstraintOptOptions, 89
    OESz::OETorsionScanner, 63
Constructors
    OESz::OEBoundEntropyOptions: public OELigandChargeOptions, 13
    OESz::OEEntropyResults, 14
    OESz::OEFreeFormConf, 18
    OESz::OEFreeFormConfAdvanced, 19
    OESz::OEFreeFormConfOptions, 21
    OESz::OEFreeFormConfResults, 24
    OESz::OEFreeFormSolvOptions, 25
    OESz::OEFreeFormSolvResults, 27
    OESz::OELigandEntropyOptions: public OELigandChargeOptions, 30
    OESz::OEProteinLigandOptResults, 34
    OESz::OERestrictionEnergyResult, 35
    OESz::OESingleConfResult, 36
    OESz::OESzybki, 38
    OESz::OESzybkiEnsembleResults, 42
    OESz::OESzybkiGeneralOptions, 44
    OESz::OESzybkiOptions, 47
    OESz::OESzybkiOptOptions, 48
    OESz::OESzybkiProteinOptions, 50
    OESz::OESzybkiResults, 52
    OESz::OESzybkiSolventOptions, 56
    OESz::OETorsionScanOptions, 58
    OESz::OETorsionScanResult, 62

E

Energy
    OESz::OEFixedProteinLigandOptimizer, 15
    OESz::OEFlexProteinLigandOptimizer, 16
EstimateEnergies
    OESz::OEFreeFormConfAdvanced, 19
EstimateFreeEnergies
    OESz::OEFreeFormConf, 18
EstimateRestrictionEnergy
    OESz::OEFreeFormConfAdvanced, 19
Example Code
    boundentropy.cs, 123
    findsimilarconfs.cs, 134
    flexibleprotein.cs, 111
    flexresidues.cs, 115
    freeformconf.cs, 126
    freeformconfadvanced.cs, 129
    freeformconfadvrstr.cs, 131
    freeformconfrstr.cs, 127
    freeformsolv.cs, 124
    ligandentropy.cs, 122
    ligandflexproteinnewton.cs, 120
    ligandmultipleoptions.cs, 100
    proteinligandpb.cs, 117
    proteinmultipleoptions.cs, 108
    simpleconformer.cs, 105

simplenewton.cs, 103
simpleprotein.cs, 106
simplesmirnoff.cs, 98
simplevacuum.cs, 97

F

FindSimilarConfs
OESz::OEFreeFormConf, 18
findsimilarconfs.cs
Example Code, 134
FixAtoms
OESz::OESzybki, 39
flexibleprotein.cs
Example Code, 111
flexresidues.cs
Example Code, 115
freeformconf.cs
Example Code, 126
freeformconfadvanced.cs
Example Code, 129
freeformconfadvrstr.cs
Example Code, 131
freeformconfrstr.cs
Example Code, 127
freeformsolv.cs
Example Code, 124

G

GetAngle
OESz::OETorsionScanResult, 62
GetAtomicPotentials
OESz::OEFreeFormSolvResults, 28
GetAtomicRadiiType
OESz::OEFreeFormSolvOptions, 25
OESz::OESzybkiSolventOptions, 56
GetBendEne
OESz::OELigandEnergy, 84
GetBondEne
OESz::OELigandEnergy, 84
GetCalculateFrozenTerms
OESz::OESzybkiOptOptions, 48
GetCalculateGradients
OESz::OESzybkiGeneralOptions, 44
GetCavitySolventParameter
OESz::OESzybkiSolventOptions, 56
GetChargeType
OESz::OEFreeFormSolvOptions, 25
OESz::OESzybkiEnsembleResults, 43
GetConfFreeEnergyFromEnsemble
OESz::OESzybkiResults, 55
GetConfGenRMS
OESz::OEFreeFormSolvOptions, 25
GetConfIdx
OESz::OESingleConfResult, 36
OESz::OESzybkiResults, 52
GetConfigurationalEntropy
OESz::OEEntropyResults, 14
OESz::OESzybkiEnsembleResults, 43
GetConfResult
OESz::OEFreeFormConfResults, 24
GetCorrSolventOptions
OESz::OEFreeFormConfOptions, 21
GetCoulombEne
OESz::OELigandEnergy, 84
GetCPUTime
OESz::OESzybkiResults, 52
GetDelta
OESz::OETorsionScanOptions, 58
GetDeltaE
OESz::OESingleConfResult, 36
GetDeltaG
OESz::OESingleConfResult, 36
GetElectrostaticSolvationFreeEnergy
OESz::OEFreeFormSolvResults, 28
GetEnergy
OESz::OETorsionScanResult, 62
GetEnergyTerm
OESz::OESzybkiResults, 52
GetEnsembleLigPartialSolvEntropy
OESz::OESzybkiEnsembleResults, 43
GetEnsembleLigSolvEntropy
OESz::OESzybkiEnsembleResults, 43
GetEnsembleProtDesolvEntropy
OESz::OESzybkiEnsembleResults, 43
GetEntropicEnergy
OESz::OESzybkiEnsembleResults, 42
GetEntropy
OESz::OESzybki, 39
GetEntropy(const OEChem::OEMCMolBase&,
unsigned int, unsigned int)
OESz::OESzybki, 39
GetEnvironment
OESz::OELigandEntropyOptions: public
OELigandChargeOptions, 30
GetExactVdWProteinLigand
OESz::OESzybkiProteinOptions, 50
GetFinalEnergies
OESz::OELigandOptResults, 88
OESz::OEProteinLigandOptResults, 34
GetFinalRMSGradient
OESz::OESzybkiResults, 53
GetFinalTotalPotential
OESz::OESzybkiResults, 53
GetFixPattern
OESz::OESzybki, 40
GetFixPredicate
OESz::OESzybki, 40
GetFlexRange

OESz::OEProteinFlexOptions, 31
GetForceField
OESz::OEBoundEntropyOptions: public OELigandChargeOptions, 13
OESz::OELigandEntropyOptions: public OELigandChargeOptions, 30
OESz::OELigandOptOptions, 85
OESz::OEProteinLigandOptOptions, 32
OESz::OESzybkiGeneralOptions, 44
OESz::OETorsionScanOptions, 58
GetForceFieldType
OESz::OEBoundEntropyOptions: public OELigandChargeOptions, 13
OESz::OELigandEntropyOptions: public OELigandChargeOptions, 30
OESz::OELigandOptOptions, 86
OESz::OESzybkiGeneralOptions, 44
OESz::OETorsionScanOptions, 59
GetGeneralOptions
OESz::OESzybkiOptions, 47
GetGenerateElfConfs
OESz::OELigandChargeOptions, 29
GetGlobalStrain
OESz::OERestrictionEnergyResult, 35
GetGradients
OESz::OESzybkiResults, 53
GetGradTolerance
OESz::OELigandOptOptions, 86
OESz::OEProteinLigandOptOptions, 32
OESz::OESzybkiOptOptions, 49
GetHarmonicConstraints
OESz::OESzybki, 40
GetHydrophobicSolvationFreeEnergy
OESz::OEFreeFormSolvResults, 28
GetIEFFCluster
OESz::OESzybkiGeneralOptions, 44
GetImproperEne
OESz::OELigandEnergy, 84
GetInitialEnergies
OESz::OELigandOptResults, 88
OESz::OEProteinLigandOptResults, 34
GetInitialRMSGradient
OESz::OESzybkiResults, 53
GetInitialTotalPotential
OESz::OESzybkiResults, 53
GetInterEnergy
OESz::OESzybkiResults, 53
GetIntermolecularVdWCutoff
OESz::OESzybkiProteinOptions, 50
GetIntraEne
OESz::OELigandEnergy, 84
GetIntramolecularLigandEnergy
OESz::OESzybkiResults, 53
GetIntramolecularVdWCutoff
OESz::OESzybkiGeneralOptions, 44
GetIonicState
OESz::OEFreeFormConfOptions, 21
OESz::OEFreeFormSolvOptions, 26
GetLigandCharge
OESz::OELigandChargeOptions, 29
OESz::OELigandOptOptions, 86
GetLigandChargeType
OESz::OELigandChargeOptions, 29
OESz::OELigandOptOptions, 86
GetLigandRMSD
OESz::OEProteinLigandOptResults, 34
GetLigandRMSDHeavy
OESz::OESzybkiGeneralOptions, 44
GetLnQrot
OESz::OESzybkiResults, 55
GetLnQrotational
OESz::OESingleConfResult, 36
GetLnQvib
OESz::OESzybkiResults, 55
GetLnQvibrational
OESz::OESingleConfResult, 36
GetLocalStrain
OESz::OERestrictionEnergyResult, 35
GetLowestdGConfIdx
OESz::OEFreeFormConfResults, 24
GetLowestEneResults
OESz::OETorsionScanner, 63
GetLowestEnergyConfs
OESz::OETorsionScanner, 63
GetLowestRelEneConfIdx
OESz::OEFreeFormConfResults, 24
GetMaxConfGen
OESz::OEFreeFormSolvOptions, 26
GetMaxDisplacement
OESz::OESzybkiResults, 53
GetMaxIter
OESz::OELigandOptOptions, 86
OESz::OEProteinLigandOptOptions, 32
OESz::OESzybkiOptOptions, 49
OESz::OETorsionScanOptions, 59
GetMMFFEnergy
OESz::OESingleConfResult, 37
GetNonBondCutoff
OESz::OELigandOptOptions, 86
OESz::OEProteinLigandOptOptions, 32
GetNumCycles
OESz::OESzybkiResults, 54
GetNumFixAtoms
OESz::OESzybkiResults, 54
GetNumRotors
OESz::OESzybkiResults, 54
GetNumStarts
OESz::OETorsionScanOptions, 60

GetNumUniqueConfs
    OESz::OEFreeFormConfResults, 24
GetOmegaOptions
    OESz::OEFreeFormConfOptions, 21
GetOptimizationType
    OESz::OEProteinLigandOptOptions, 32
GetOptimizeOptions
    OESz::OEFreeFormConfOptions, 21
GetOptimizerType
    OESz::OESzybkiOptOptions, 49
GetOptOptions
    OESz::OESzybkiOptions, 47
GetOverlapDiv
    OESz::OETorsionScanOptions, 60
GetPenaltyForceConstant
    OESz::OETorConstraintOptOptions, 89
    OESz::OETorsionScanOptions, 59
GetPreOptimizeOptions
    OESz::OEFreeFormConfOptions, 22
GetProbability
    OESz::OESingleConfResult, 37
GetProtein
    OESz::OESzybki, 40
GetProteinDielectric
    OESz::OESzybkiProteinOptions, 50
GetProteinElectrostaticModel
    OESz::OESzybkiProteinOptions, 50
GetProteinFlexibilityRange
    OESz::OESzybkiProteinOptions, 51
GetProteinFlexibilityType
    OESz::OESzybkiProteinOptions, 51
GetProteinOptions
    OESz::OESzybkiOptions, 47
GetProteinRMSD
    OESz::OEProteinLigandOptResults, 34
    OESz::OESzybkiResults, 54
GetRelativeEnergy
    OESz::OESingleConfResult, 37
GetRelLnQ
    OESz::OESingleConfResult, 37
GetRemoveAttractiveVdWForces
    OESz::OESzybkiGeneralOptions, 44
GetRemoveCoulombTerms
    OESz::OESzybkiGeneralOptions, 45
GetResidueID
    OESz::OEProteinFlexOptions, 31
GetResultsForConformations
    OESz::OEFreeFormConfResults, 25
    OESz::OESzybkiEnsembleResults, 43
GetReturnCode
    OESz::OEFreeFormSolvResults, 28
    OESz::OELigandOptResults, 88
    OESz::OEProteinLigandOptResults, 34
GetRMSD
    OESz::OELigandOptResults, 88
    OESz::OESingleConfResult, 38
    OESz::OESzybkiResults, 54
GetRotationalEntropy
    OESz::OESingleConfResult, 37
GetRotEntropy
    OESz::OESzybkiResults, 55
GetRunType
    OESz::OESzybkiOptions, 47
GetSaltConcentration
    OESz::OESzybkiSolventOptions, 56
GetScannedAngles
    OESz::OETorsionScanner, 63
GetSheffieldParameters
    OESz::OESzybki, 40
GetSinglePointConfs
    OESz::OETorsionScanner, 64
GetSoluteDielectric
    OESz::OESzybkiGeneralOptions, 45
GetSolvationEne
    OESz::OELigandEnergy, 84
GetSolvationEntropy
    OESz::OEEntropyResults, 15
GetSolvationFreeEnergy
    OESz::OEFreeFormSolvResults, 28
GetSolvationType
    OESz::OETorsionScanOptions, 59
GetSolvEnergy
    OESz::OESingleConfResult, 37
GetSolventDielectric
    OESz::OESzybkiSolventOptions, 56
GetSolventModel
    OESz::OELigandOptOptions, 86
    OESz::OEProteinLigandOptOptions, 33
    OESz::OESzybkiSolventOptions, 57
GetSolventOptions
    OESz::OEFreeFormConfOptions, 22
    OESz::OESzybkiOptions, 47
GetSzybkiGeneralOptions
    OESz::OEFreeFormConfOptions, 22
GetTemperature
    OESz::OEFreeFormConfResults, 25
    OESz::OESzybkiEnsembleResults, 42
    OESz::OESzybkiGeneralOptions, 45
GetTolerance
    OESz::OETorsionScanOptions, 59
GetTorsionEne
    OESz::OELigandEnergy, 84
GetTotalEne
    OESz::OELigandEnergy, 84
GetTotalEnergy
    OESz::OESingleConfResult, 37
    OESz::OESzybkiResults, 54
GetTotalEnergyWithHarmConstraint

OESz::OESzybkiResults, 54
GetTotalEntropy
OESz::OEFreeFormConfResults, 25
OESz::OESzybkiEnsembleResults, 42
GetTotalEntropy in J/(mol K).
OESz::OEEntropyResults, 14
GetTotalRMSD
OESz::OEProteinLigandOptResults, 35
GetTranslationalEntropy
OESz::OEEntropyResults, 14
GetUseInput3D
OESz::OEFreeFormSolvOptions, 26
GetUseInputEnsemble
OESz::OEFreeFormConfOptions, 22
OESz::OEFreeFormSolvOptions, 26
GetUseInternalCoord
OESz::OETorConstraintOptOptions, 90
OESz::OETorsionScanOptions, 59
GetUserDefinedAngles
OESz::OETorsionScanOptions, 59
GetUseSinglePointPB
OESz::OETorsionScanOptions, 59
GetUseSolventCorr
OESz::OEFreeFormConfOptions, 22
GetVdWEne
OESz::OELigandEnergy, 84
GetVerbose
OESz::OESzybkiGeneralOptions, 45
GetVibEntropy
OESz::OESzybkiResults, 55
GetVibrationalEnergy
OESz::OESingleConfResult, 37
GetVibrationalEntropy
OESz::OESingleConfResult, 38
GetVibrationalRotationalEntropy
OESz::OEEntropyResults, 14
GetWarnings
OESz::OEFreeFormSolvResults, 28

H

HasEntropy
OESz::OESzybkiEnsembleResults, 43
HasFlexibleResidues
OESz::OESzybkiProteinOptions, 51

I

IdentifyConformer
OESz::OEFreeFormConfAdvanced, 19
IsActiveTerm
OESz::OESzybkiResults, 54
IsUnique
OESz::OESzybkiResults, 55

L

ligandentropy.cs
Example Code, 122
ligandflexproteinnewton.cs
Example Code, 120
ligandmultipleoptions.cs
Example Code, 100
LoadPotentialGrid
OESz::OESzybki, 40

N

NumConfs
OESz::OESzybkiEnsembleResults, 42

O

OESz::OEAtomicRadii, 74
OESz::OEAtomicRadii::Bondi, 74
OESz::OEAtomicRadii::ZAP7, 74
OESz::OEAtomicRadii::ZAP9, 74
OESz::OEBoundEntropyOptions: public OELigandChargeOptions, 13
Constructors, 13
GetForceField, 13
GetForceFieldType, 13
SetForceField, 13
OESz::OEBoundLigandEntropy, 80
OESz::OEComplexFFType, 91
OESz::OEComplexFFType::FF14SB_PARSLEY, 92
OESz::OEComplexFFType::FF14SB_SAGE, 92
OESz::OEComplexFFType::MMFF94, 91
OESz::OEComplexFFType::MMFF94S, 91
OESz::OEComplexFFType::MMFF_AMBER, 91
OESz::OEComplexFFType::MMFFS_AMBER, 92
OESz::OEEntropyMethod, 64
OESz::OEEntropyMethod::Analytic, 64
OESz::OEEntropyMethod::ExternalHessian, 64
OESz::OEEntropyMethod::QNewton, 64
OESz::OEEntropyResults, 14
Constructors, 14
GetConfigurationalEntropy, 14
GetSolvationEntropy, 15
GetTotalEntropy in J/(mol K)., 14
GetTranslationalEntropy, 14
GetVibrationalRotationalEntropy, 14
SetResults, 15
OESz::OEEnvType, 64
OESz::OEEnvType::Gas, 64
OESz::OEEnvType::Protein, 65
OESz::OEEnvType::Solution, 65
OESz::OEEnvType::SolutionSA, 65
OESz::OEEnvType::SolutionSPT, 64
OESz::OEEstimateConfFreeEnergies, 78
OESz::OEEstimateSolvFreeEnergy, 78

OESz::OEFixedProteinLigandOptimizer, 15
    Constructor, 15
    Energy, 15
    Optimize, 15
    SetProtein, 16
OESz::OEFlexProteinLigandOptimizer, 16
    Constructor, 16
    Energy, 16
    Optimize, 17
OESz::OEForceFieldType, 65
OESz::OEForceFieldType::FF14SB_SMIRNOFF, 66
OESz::OEForceFieldType::MMFF94, 65
OESz::OEForceFieldType::MMFF94S, 65
OESz::OEForceFieldType::MMFF_AMBER, 65
OESz::OEForceFieldType::MMFF_IEFF, 65
OESz::OEForceFieldType::MMFFS_AMBER, 65
OESz::OEForceFieldType::MMFFS_IEFF, 65
OESz::OEForceFieldType::PARSLEY_OPENFF, 66
OESz::OEForceFieldType::PARSLEY_OPENFF100, 66
OESz::OEForceFieldType::PARSLEY_OPENFF111, 66
OESz::OEForceFieldType::PARSLEY_OPENFF121, 66
OESz::OEForceFieldType::PARSLEY_OPENFF131, 66
OESz::OEForceFieldType::SAGE_OPENFF, 66
OESz::OEForceFieldType::SAGE_OPENFF200, 66
OESz::OEForceFieldType::SAGE_OPENFF210, 66
OESz::OEForceFieldType::SMIRNOFF99FROSST, 66
OESz::OEFreeFormConf, 18
    Constructors, 18
    EstimateFreeEnergies, 18
    FindSimilarConfs, 18
OESz::OEFreeFormConfAdvanced, 19
    Constructors, 19
    EstimateEnergies, 19
    EstimateRestrictionEnergy, 19
    IdentifyConformer, 19
    Optimize, 20
    OptimizeRestraint, 20
    PreOptimizeEnsemble, 20
    PrepareEnsemble, 20
    RemoveDuplicates, 21
OESz::OEFreeFormConfOptions, 21
    Constructors, 21
    GetCorrSolventOptions, 21
    GetIonicState, 21
    GetOmegaOptions, 21
    GetOptimizeOptions, 21
    GetPreOptimizeOptions, 22
    GetSolventOptions, 22
    GetSzybkiGeneralOptions, 22
    GetUseInputEnsemble, 22
    GetUseSolventCorr, 22
    SetCorrSolventOptions, 22
    SetIonicState, 22
    SetOmegaOptions, 23
    SetOptimizeOptions, 23
    SetPreOptimizeOptions, 23
    SetSolventOptions, 23
    SetSzybkiGeneralOptions, 23
    SetUseInputEnsemble, 23
    SetUseSolventCorr, 23
OESz::OEFreeFormConfResults, 24
    Constructors, 24
    GetConfResult, 24
    GetLowestdGConfIdx, 24
    GetLowestRelEneConfIdx, 24
    GetNumUniqueConfs, 24
    GetResultsForConformations, 25
    GetTemperature, 25
    GetTotalEntropy, 25
    PrintCsv, 24
    PrintTxt, 24
OESz::OEFreeFormIonicState, 75
OESz::OEFreeFormIonicState::Default, 75
OESz::OEFreeFormIonicState::Input, 75
OESz::OEFreeFormIonicState::PH74, 75
OESz::OEFreeFormIonicState::Uncharged, 75
OESz::OEFreeFormReturnCode, 75
OESz::OEFreeFormReturnCode::AtomTypeFailure, 75
OESz::OEFreeFormReturnCode::ChargeAssignFailure, 76
OESz::OEFreeFormReturnCode::ChargeMismatch, 76
OESz::OEFreeFormReturnCode::ConfGenFailure, 76
OESz::OEFreeFormReturnCode::GraphMismatch, 77
OESz::OEFreeFormReturnCode::ImplicitHydrogens, 76
OESz::OEFreeFormReturnCode::InputGeometry, 76
OESz::OEFreeFormReturnCode::MissingFFConfEne, 77
OESz::OEFreeFormReturnCode::MissingPartnerConf, 77
OESz::OEFreeFormReturnCode::No3DCoordinates, 76
OESz::OEFreeFormReturnCode::NotSupportedCharges, 76
OESz::OEFreeFormReturnCode::PartnerConfNotFound, 77
OESz::OEFreeFormReturnCode::Success, 75
OESz::OEFreeFormReturnCode::TooManyRotors, 76
OESz::OEFreeFormReturnCode::UnequalNumConfs, 77
OESz::OEFreeFormReturnCode::UnrecognizedChargeState, 76
OESz::OEFreeFormReturnCode::UnsupportedRadiiType, 76
OESz::OEFreeFormSolvOptions, 25

Constructors, 25
GetAtomicRadiiType, 25
GetChargeType, 25
GetConfGenRMS, 25
GetIonicState, 26
GetMaxConfGen, 26
GetUseInput3D, 26
GetUseInputEnsemble, 26
SetAtomicRadiiType, 26
SetChargeType, 26
SetConfGenRMS, 26
SetIonicState, 27
SetMaxConfGen, 27
SetUseInput3D, 27
SetUseInputEnsemble, 27
OESz::OEFreeFormSolvResults, 27
Constructors, 27
GetAtomicPotentials, 28
GetElectrostaticSolvationFreeEnergy, 28
GetHydrophobicSolvationFreeEnergy, 28
GetReturnCode, 28
GetSolvationFreeEnergy, 28
GetWarnings, 28
OESz::OEFreeFormWarning, 77
OESz::OEFreeFormWarning::HSamplingSkipped, 77
OESz::OEFreeFormWarning::Missing3DCoordinates, 77
OESz::OEFreeFormWarning::NotRemovableCharge, 77
OESz::OEFreeFormWarning::UnspecifiedStereo, 77
OESz::OEFreeFormWarning::ZeroPartialCharges, 77
OESz::OEGetEnergyTermName, 78
OESz::OEGetFreeFormError, 78
OESz::OEGetSzybkiError, 78
OESz::OELigandChargeOptions, 28
Constructor, 29
GetGenerateElfConfs, 29
GetLigandCharge, 29
GetLigandChargeType, 29
SetGenerateElfConfs, 29
SetLigandCharge, 29
OESz::OELigandChargeType, 92
OESz::OELigandChargeType::AM1BCC, 92
OESz::OELigandChargeType::AM1BCCELF10, 92
OESz::OELigandChargeType::CURRENT, 92
OESz::OELigandChargeType::MMFF, 92
OESz::OELigandEnergy, 83
Constructor, 83
GetBendEne, 84
GetBondEne, 84
GetCoulombEne, 84
GetImproperEne, 84
GetIntraEne, 84
GetSolvationEne, 84
GetTorsionEne, 84
GetTotalEne, 84
GetVdWEne, 84
OESz::OELigandEntropy, 81
OESz::OELigandEntropyOptions: public OELigandChargeOptions, 29
Constructors, 30
GetEnvironment, 30
GetForceField, 30
GetForceFieldType, 30
SetEnvironment, 30
SetForceField, 30
OESz::OELigandOptOptions, 85
Constructor, 85
GetForceField, 85
GetForceFieldType, 86
GetGradTolerance, 86
GetLigandCharge, 86
GetLigandChargeType, 86
GetMaxIter, 86
GetNonBondCutoff, 86
GetSolventModel, 86
SetForceField, 86
SetGradTolerance, 87
SetLigandCharge, 87
SetMaxIter, 87
SetNonBondCutoff, 87
SetSolventModel, 87
OESz::OELigandOptResults, 87
Constructor, 88
GetFinalEnergies, 88
GetInitialEnergies, 88
GetReturnCode, 88
GetRMSD, 88
OESz::OEOptimizationType, 93
OESz::OEOptimizationType::Cartesian, 93
OESz::OEOptimizationType::PoseCartesian, 93
OESz::OEOptType, 67
OESz::OEOptType::BFGS, 67
OESz::OEOptType::CG, 67
OESz::OEOptType::NEWTON, 67
OESz::OEOptType::SD, 67
OESz::OEOptType::SD_BFGS, 67
OESz::OEOptType::SD_CG, 67
OESz::OEPotentialTerms, 67
OESz::OEPotentialTerms::Bend, 68
OESz::OEPotentialTerms::Bond, 68
OESz::OEPotentialTerms::CavitySolvation, 71
OESz::OEPotentialTerms::Coulomb, 68
OESz::OEPotentialTerms::CoulombProteinLigand, 71
OESz::OEPotentialTerms::ExactCoulomb, 71

OESz::OEPotentialTerms::GridCoulomb, 71
OESz::OEPotentialTerms::HarmonicConstraint, 70
OESz::OEPotentialTerms::IEFFInteraction, 72
OESz::OEPotentialTerms::ImproperTorsion, 68
OESz::OEPotentialTerms::InterLigandIEFF, 72
OESz::OEPotentialTerms::LigandBend, 69
OESz::OEPotentialTerms::LigandBond, 69
OESz::OEPotentialTerms::LigandCoulomb, 69
OESz::OEPotentialTerms::LigandDesolvation, 71
OESz::OEPotentialTerms::LigandImproperTorsion, 69
OESz::OEPotentialTerms::LigandStretchBend, 69
OESz::OEPotentialTerms::LigandTorsion, 69
OESz::OEPotentialTerms::LigandVdW, 69
OESz::OEPotentialTerms::Max, 72
OESz::OEPotentialTerms::PBSolvation, 70
OESz::OEPotentialTerms::ProteinBend, 70
OESz::OEPotentialTerms::ProteinBond, 69
OESz::OEPotentialTerms::ProteinCoulomb, 69
OESz::OEPotentialTerms::ProteinDesolvation, 71
OESz::OEPotentialTerms::ProteinImproperTorsion, 70
OESz::OEPotentialTerms::ProteinLigandAmberCoulomb, 72
OESz::OEPotentialTerms::ProteinLigandAmberVdW, 71
OESz::OEPotentialTerms::ProteinLigandCoulomb, 70
OESz::OEPotentialTerms::ProteinLigandInteraction, 70
OESz::OEPotentialTerms::ProteinLigandVdW, 70
OESz::OEPotentialTerms::ProteinPseudoLigandInteraction, 71
OESz::OEPotentialTerms::ProteinStretchBend, 70
OESz::OEPotentialTerms::ProteinTorsion, 70
OESz::OEPotentialTerms::ProteinVdW, 69
OESz::OEPotentialTerms::SheffieldSolvation, 70
OESz::OEPotentialTerms::SolventScreening, 71
OESz::OEPotentialTerms::StretchBend, 68
OESz::OEPotentialTerms::Torsion, 68
OESz::OEPotentialTerms::TorsionHarmonicConstraint, 72
OESz::OEPotentialTerms::VdW, 68
OESz::OEPotentialTerms::VdWProteinLigand, 71
OESz::OEProteinElectrostatics, 72
OESz::OEProteinElectrostatics::ExactCoulomb, 72
OESz::OEProteinElectrostatics::GridCoulomb, 72
OESz::OEProteinElectrostatics::GridPB, 72
OESz::OEProteinElectrostatics::NoElectrostatics, 72
OESz::OEProteinElectrostatics::SolventPBForces, 73
OESz::OEProteinFlexOptions, 31
    Constructor, 31
    GetFlexRange, 31
    GetResidueID, 31
    SetFlexRange, 31
    SetResidueID, 31
OESz::OEProteinLigandOptOptions, 31
    Constructor, 32
    GetForceField, 32
    GetGradTolerance, 32
    GetMaxIter, 32
    GetNonBondCutoff, 32
    GetOptimizationType, 32
    GetSolventModel, 33
    SetForceField, 33
    SetGradTolerance, 33
    SetMaxIter, 33
    SetNonBondCutoff, 33
    SetOptimizationType, 33
    SetSolventModel, 33
OESz::OEProteinLigandOptResults, 34
    Constructors, 34
    GetFinalEnergies, 34
    GetInitialEnergies, 34
    GetLigandRMSD, 34
    GetProteinRMSD, 34
    GetReturnCode, 34
    GetTotalRMSD, 35
OESz::OEProtFlex, 74
OESz::OEProtFlex::NoFlex, 74
OESz::OEProtFlex::PolarH, 74
OESz::OEProtFlex::PolarHResList, 75
OESz::OEProtFlex::ResidueList, 75
OESz::OEProtFlex::Residues, 74
OESz::OEProtFlex::SideChains, 74
OESz::OEProtFlex::SideChainsList, 75
OESz::OERestrictionEnergyResult, 35
    Constructors, 35
    GetGlobalStrain, 35
    GetLocalStrain, 35
OESz::OERunType, 73
OESz::OERunType::CartesiansOpt, 73
OESz::OERunType::SinglePoint, 73
OESz::OERunType::SolidBodyOpt, 73
OESz::OERunType::TorsionsOpt, 73
OESz::OESingleConfResult, 35
    Constructors, 36
    GetConfIdx, 36
    GetDeltaE, 36
    GetDeltaG, 36

GetLnQrotational, 36
GetLnQvibrational, 36
GetMMFFEnergy, 37
GetProbability, 37
GetRelativeEnergy, 37
GetRelLnQ, 37
GetRMSD, 38
GetRotationalEntropy, 37
GetSolvEnergy, 37
GetTotalEnergy, 37
GetVibrationalEnergy, 37
GetVibrationalEntropy, 38
OESz::OESolventModel, 73
OESz::OESolventModel::Cavity, 74
OESz::OESolventModel::NoSolv, 73
OESz::OESolventModel::PB, 74
OESz::OESolventModel::Sheffield, 73
OESz::OESzybki, 38
ClearFixAtoms, 38
ClearHarmonicConstraints, 39
ClearTorsionConstraint, 39
Constructors, 38
FixAtoms, 39
GetEntropy, 39
GetEntropy(const OEChem::OEMCMolBase&, unsigned int, unsigned int), 39
GetFixPattern, 40
GetFixPredicate, 40
GetHarmonicConstraints, 40
GetProtein, 40
GetSheffieldParameters, 40
LoadPotentialGrid, 40
operator(), 38
SavePotentialGrid, 41
SetHarmonicConstraints, 41
SetProtein, 41
SetSheffieldParameters, 41
SetTorsionConstraint, 41
UnsetProtein, 42
OESz::OESzybkiEnsembleResults, 42
Constructors, 42
GetChargeType, 43
GetConfigurationalEntropy, 43
GetEnsembleLigPartialSolvEntropy, 43
GetEnsembleLigSolvEntropy, 43
GetEnsembleProtDesolvEntropy, 43
GetEntropicEnergy, 42
GetResultsForConformations, 43
GetTemperature, 42
GetTotalEntropy, 42
HasEntropy, 43
NumConfs, 42
OESz::OESzybkiGeneralOptions, 43
Constructors, 44
GetCalculateGradients, 44
GetForceField, 44
GetForceFieldType, 44
GetIEFFCluster, 44
GetIntramolecularVdWCutoff, 44
GetLigandRMSDHeavy, 44
GetRemoveAttractiveVdWForces, 44
GetRemoveCoulombTerms, 45
GetSoluteDielectric, 45
GetTemperature, 45
GetVerbose, 45
SetCalculateGradients, 45
SetForceField, 45
SetForceFieldType, 45
SetIEFFCluster, 45
SetIntramolecularVdWCutoff, 46
SetLigandRMSDHeavy, 46
SetRemoveAttractiveVdWForces, 46
SetRemoveCoulombTerms, 46
SetSoluteDielectric, 46
SetTemperature, 46
SetVerbose, 46
OESz::OESzybkiGetArch, 78
OESz::OESzybkiGetLicensee, 79
OESz::OESzybkiGetPlatform, 79
OESz::OESzybkiGetRelease, 79
OESz::OESzybkiGetSite, 79
OESz::OESzybkiGetVersion, 79
OESz::OESzybkiOptions, 46
Constructors, 47
GetGeneralOptions, 47
GetOptOptions, 47
GetProteinOptions, 47
GetRunType, 47
GetSolventOptions, 47
SetGeneralOptions, 47
SetOptOptions, 48
SetProteinOptions, 48
SetRunType, 48
SetSolventOptions, 48
OESz::OESzybkiOptOptions, 48
Constructors, 48
GetCalculateFrozenTerms, 48
GetGradTolerance, 49
GetMaxIter, 49
GetOptimizerType, 49
SetCalculateFrozenTerms, 49
SetGradTolerance, 49
SetMaxIter, 49
SetOptimizerType, 49
OESz::OESzybkiProteinOptions, 49
AddFlexibleResidue, 50
ClearFlexibleResidues, 50
Constructors, 50

GetExactVdWProteinLigand, 50
GetIntermolecularVdWCutoff, 50
GetProteinDielectric, 50
GetProteinElectrostaticModel, 50
GetProteinFlexibilityRange, 51
GetProteinFlexibilityType, 51
HasFlexibleResidues, 51
SetExactVdWProteinLigand, 51
SetIntermolecularVdWCutoff, 51
SetProteinDielectric, 51
SetProteinElectrostaticModel, 51
SetProteinFlexibilityRange, 52
SetProteinFlexibilityType, 52
OESz::OESzybkiResults, 52
Clear, 52
Constructors, 52
GetConfFreeEnergyFromEnsemble, 55
GetConfIdx, 52
GetCPUTime, 52
GetEnergyTerm, 52
GetFinalRMSGradient, 53
GetFinalTotalPotential, 53
GetGradients, 53
GetInitialRMSGradient, 53
GetInitialTotalPotential, 53
GetInterEnergy, 53
GetIntramolecularLigandEnergy, 53
GetLnQrot, 55
GetLnQvib, 55
GetMaxDisplacement, 53
GetNumCycles, 54
GetNumFixAtoms, 54
GetNumRotors, 54
GetProteinRMSD, 54
GetRMSD, 54
GetRotEntropy, 55
GetTotalEnergy, 54
GetTotalEnergyWithHarmConstraint, 54
GetVibEntropy, 55
IsActiveTerm, 54
IsUnique, 55
Print, 55
OESz::OESzybkiReturnCode, 93
OESz::OESzybkiReturnCode::AromaticRingDev, 93
OESz::OESzybkiReturnCode::ContainsImplicitH, 94
OESz::OESzybkiReturnCode::ExcessiveOverlap, 93
OESz::OESzybkiReturnCode::Failed, 95
OESz::OESzybkiReturnCode::FFNeedHessiens, 94
OESz::OESzybkiReturnCode::IncompleteOpt, 93
OESz::OESzybkiReturnCode::InvalidOptions, 94
OESz::OESzybkiReturnCode::LigandFailedCharge, 95
OESz::OESzybkiReturnCode::LigandFailedFF, 94
OESz::OESzybkiReturnCode::LigandMissing3D, 94
OESz::OESzybkiReturnCode::LigandMissingCharge, 95
OESz::OESzybkiReturnCode::OptimizerFailed, 94
OESz::OESzybkiReturnCode::ProteinFailedFF, 94
OESz::OESzybkiReturnCode::ProteinMissing3D, 94
OESz::OESzybkiReturnCode::ProteinMissingCharge, 94
OESz::OESzybkiReturnCode::ProteinNotSet, 94
OESz::OESzybkiReturnCode::Success, 93
OESz::OESzybkiSolventOptions, 56
Constructors, 56
GetAtomicRadiiType, 56
GetCavitySolventParameter, 56
GetSaltConcentration, 56
GetSolventDielectric, 56
GetSolventModel, 57
SetAtomicRadiiType, 57
SetCavitySolventParameter, 57
SetChargeEngine, 57
SetSaltConcentration, 57
SetSolventDielectric, 57
SetSolventModel, 57
SetUseCurrentCharges, 57
OESz::OETorConstraintOptimizer, 90
Constructor, 90
Optimize, 91
Setup, 91
OESz::OETorConstraintOptOptions, 88
Constructor, 89
GetPenaltyForceConstant, 89
GetUseInternalCoord, 90
SetPenaltyForceConstant, 90
SetUseInternalCoord, 90
OESz::OETorsionScan, 79
OESz::OETorsionScanner, 63
Constructor, 63
GetLowestEneResults, 63
GetLowestEnergyConfs, 63
GetScannedAngles, 63
GetSinglePointConfs, 64
Scan, 64
OESz::OETorsionScanOptions, 58
Constructors, 58
GetDelta, 58
GetForceField, 58
GetForceFieldType, 59
GetMaxIter, 59
GetNumStarts, 60
GetOverlapDiv, 60
GetPenaltyForceConstant, 59
GetSolvationType, 59

GetTolerance, 59
GetUseInternalCoord, 59
GetUserDefinedAngles, 59
GetUseSinglePointPB, 59
SetDelta, 60
SetForceField, 60
SetForceFieldType, 60
SetMaxIter, 60
SetNumStarts, 61
SetOverlapDiv, 62
SetPenaltyForceConstant, 61
SetSolvationType, 61
SetTolerance, 61
SetUseInternalCoord, 61
SetUserDefinedAngles, 61
SetUseSinglePointPB, 61
OESz::OETorsionScanResult, 62
Constructors, 62
GetAngle, 62
GetEnergy, 62
operator()
OESz::OESzybki, 38
Optimize
OESz::OEFixedProteinLigandOptimizer, 15
OESz::OEFlexProteinLigandOptimizer, 17
OESz::OEFreeFormConfAdvanced, 20
OESz::OETorConstraintOptimizer, 91
OptimizeRestraint
OESz::OEFreeFormConfAdvanced, 20

P

PreOptimizeEnsemble
OESz::OEFreeFormConfAdvanced, 20
PrepareEnsemble
OESz::OEFreeFormConfAdvanced, 20
Print
OESz::OESzybkiResults, 55
PrintCsv
OESz::OEFreeFormConfResults, 24
PrintTxt
OESz::OEFreeFormConfResults, 24
proteinligandpb.cs
Example Code, 117
proteinmultipleoptions.cs
Example Code, 108

R

RemoveDuplicates
OESz::OEFreeFormConfAdvanced, 21

S

SavePotentialGrid
OESz::OESzybki, 41
Scan
OESz::OETorsionScanner, 64
SetAtomicRadiiType
OESz::OEFreeFormSolvOptions, 26
OESz::OESzybkiSolventOptions, 57
SetCalculateFrozenTerms
OESz::OESzybkiOptOptions, 49
SetCalculateGradients
OESz::OESzybkiGeneralOptions, 45
SetCavitySolventParameter
OESz::OESzybkiSolventOptions, 57
SetChargeEngine
OESz::OESzybkiSolventOptions, 57
SetChargeType
OESz::OEFreeFormSolvOptions, 26
SetConfGenRMS
OESz::OEFreeFormSolvOptions, 26
SetCorrSolventOptions
OESz::OEFreeFormConfOptions, 22
SetDelta
OESz::OETorsionScanOptions, 60
SetEnvironment
OESz::OELigandEntropyOptions: public OELigandChargeOptions, 30
SetExactVdWProteinLigand
OESz::OESzybkiProteinOptions, 51
SetFlexRange
OESz::OEProteinFlexOptions, 31
SetForceField
OESz::OEBoundEntropyOptions: public OELigandChargeOptions, 13
OESz::OELigandEntropyOptions: public OELigandChargeOptions, 30
OESz::OELigandOptOptions, 86
OESz::OEProteinLigandOptOptions, 33
OESz::OESzybkiGeneralOptions, 45
OESz::OETorsionScanOptions, 60
SetForceFieldType
OESz::OESzybkiGeneralOptions, 45
OESz::OETorsionScanOptions, 60
SetGeneralOptions
OESz::OESzybkiOptions, 47
SetGenerateElfConfs
OESz::OELigandChargeOptions, 29
SetGradTolerance
OESz::OELigandOptOptions, 87
OESz::OEProteinLigandOptOptions, 33
OESz::OESzybkiOptOptions, 49
SetHarmonicConstraints
OESz::OESzybki, 41
SetIEFFCluster
OESz::OESzybkiGeneralOptions, 45
SetIntermolecularVdWCutoff
OESz::OESzybkiProteinOptions, 51
SetIntramolecularVdWCutoff

OESz::OESzybkiGeneralOptions, 46
SetIonicState
OESz::OEFreeFormConfOptions, 22
OESz::OEFreeFormSolvOptions, 27
SetLigandCharge
OESz::OELigandChargeOptions, 29
OESz::OELigandOptOptions, 87
SetLigandRMSDHeavy
OESz::OESzybkiGeneralOptions, 46
SetMaxConfGen
OESz::OEFreeFormSolvOptions, 27
SetMaxIter
OESz::OELigandOptOptions, 87
OESz::OEProteinLigandOptOptions, 33
OESz::OESzybkiOptOptions, 49
OESz::OETorsionScanOptions, 60
SetNonBondCutoff
OESz::OELigandOptOptions, 87
OESz::OEProteinLigandOptOptions, 33
SetNumStarts
OESz::OETorsionScanOptions, 61
SetOmegaOptions
OESz::OEFreeFormConfOptions, 23
SetOptimizationType
OESz::OEProteinLigandOptOptions, 33
SetOptimizeOptions
OESz::OEFreeFormConfOptions, 23
SetOptimizerType
OESz::OESzybkiOptOptions, 49
SetOptOptions
OESz::OESzybkiOptions, 48
SetOverlapDiv
OESz::OETorsionScanOptions, 62
SetPenaltyForceConstant
OESz::OETorConstraintOptOptions, 90
OESz::OETorsionScanOptions, 61
SetPreOptimizeOptions
OESz::OEFreeFormConfOptions, 23
SetProtein
OESz::OEFixedProteinLigandOptimizer, 16
OESz::OESzybki, 41
SetProteinDielectric
OESz::OESzybkiProteinOptions, 51
SetProteinElectrostaticModel
OESz::OESzybkiProteinOptions, 51
SetProteinFlexibilityRange
OESz::OESzybkiProteinOptions, 52
SetProteinFlexibilityType
OESz::OESzybkiProteinOptions, 52
SetProteinOptions
OESz::OESzybkiOptions, 48
SetRemoveAttractiveVdWForces
OESz::OESzybkiGeneralOptions, 46
SetRemoveCoulombTerms
OESz::OESzybkiGeneralOptions, 46
SetResidueID
OESz::OEProteinFlexOptions, 31
SetResults
OESz::OEEntropyResults, 15
SetRunType
OESz::OESzybkiOptions, 48
SetSaltConcentration
OESz::OESzybkiSolventOptions, 57
SetSheffieldParameters
OESz::OESzybki, 41
SetSoluteDielectric
OESz::OESzybkiGeneralOptions, 46
SetSolvationType
OESz::OETorsionScanOptions, 61
SetSolventDielectric
OESz::OESzybkiSolventOptions, 57
SetSolventModel
OESz::OELigandOptOptions, 87
OESz::OEProteinLigandOptOptions, 33
OESz::OESzybkiSolventOptions, 57
SetSolventOptions
OESz::OEFreeFormConfOptions, 23
OESz::OESzybkiOptions, 48
SetSzybkiGeneralOptions
OESz::OEFreeFormConfOptions, 23
SetTemperature
OESz::OESzybkiGeneralOptions, 46
SetTolerance
OESz::OETorsionScanOptions, 61
SetTorsionConstraint
OESz::OESzybki, 41
Setup
OESz::OETorConstraintOptimizer, 91
SetUseCurrentCharges
OESz::OESzybkiSolventOptions, 57
SetUseInput3D
OESz::OEFreeFormSolvOptions, 27
SetUseInputEnsemble
OESz::OEFreeFormConfOptions, 23
OESz::OEFreeFormSolvOptions, 27
SetUseInternalCoord
OESz::OETorConstraintOptOptions, 90
OESz::OETorsionScanOptions, 61
SetUserDefinedAngles
OESz::OETorsionScanOptions, 61
SetUseSinglePointPB
OESz::OETorsionScanOptions, 61
SetUseSolventCorr
OESz::OEFreeFormConfOptions, 23
SetVerbose
OESz::OESzybkiGeneralOptions, 46
simpleconformer.cs
Example Code, 105

simplenewton.cs
    Example Code, 103
simpleprotein.cs
    Example Code, 106
simplesmirnoff.cs
    Example Code, 98
simplevacuum.cs
    Example Code, 97

U

UnsetProtein
    OESz::OESzybki, 42