

OpenEye
SCIENTIFIC

Shape TK – Java

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Theory **1**

1.1 Theory . . . 1
1.1.1 Molecular Shape . . . 1
1.1.2 Shape Characteristics and the Use of Gaussians . . . 2
1.1.3 Shape from Hermite Representation . . . 4
1.1.4 Color Features . . . 6
1.1.5 Similarity Measures . . . 9
1.1.6 Implementation Details and Usage . . . 9

2 Examples **13**

2.1 Calculating Overlap . . . 13
2.1.1 Simple Overlap . . . 13
2.1.2 Adding overlap to molecules . . . 14
2.1.3 Color Overlap . . . 15
2.1.4 Adding color score to molecules . . . 17
2.1.5 User Defined Color . . . 18
2.1.6 Shape and Color Overlap . . . 20
2.2 ROCS . . . 21
2.2.1 Best Hit . . . 21
2.2.2 Top Hits . . . 22
2.2.3 Top Hits with Multiple Conformers . . . 24
2.3 Overlay Optimization . . . 25
2.3.1 Best Optimization Result . . . 25
2.3.2 All Optimization Results . . . 27
2.3.3 Manipulating all results . . . 28
2.4 Working with Shape Query . . . 31
2.4.1 Creating a shape query . . . 31
2.4.2 Overlap with shape query . . . 32
2.4.3 Top hit against shape query . . . 33
2.4.4 Grid-To-Gaussians Fit . . . 34
2.5 Shape from Hermite expansion . . . 35
2.5.1 Hermite expansion . . . 35
2.5.2 Hermite Tanimoto comparison of shape . . . 37
2.5.3 Hermite Grid Overlay . . . 39
2.6 Flexible Overlay with Shape, Color, and Forcefield . . . 40
2.6.1 Flexible Overlay . . . 40
2.6.2 Flexible Best Overlay . . . 40
2.7 Advanced Examples . . . 40
2.7.1 Calculating NxN scores . . . 40
2.7.2 Calculating shape characteristics . . . 42

2.7.3 Calculating excluded volume . . . 44

3 API **47**

3.1 OEShape Classes . . . 47
3.1.1 OEAnalyticColorFunc . . . 47
3.1.2 OEAnalyticOptions . . . 48
3.1.3 OEAnalyticShapeFunc . . . 49
3.1.4 OEAsIsStarts . . . 51
3.1.5 OEAtAtomStarts . . . 51
3.1.6 OEBestOverlayResults . . . 52
3.1.7 OEBestOverlayScore . . . 52
3.1.8 OECartesianStarts . . . 54
3.1.9 OEColorFFParameter . . . 55
3.1.10 OEColorForceField . . . 56
3.1.11 OEColorFunc . . . 58
3.1.12 OEColorOptions . . . 59
3.1.13 OEColorTanimotoCutoff . . . 60
3.1.14 OEExactColorFunc . . . 60
3.1.15 OEExactShapeFunc . . . 62
3.1.16 OEFitColorTverskyCutoff . . . 63
3.1.17 OEFitTverskyComboCutoff . . . 63
3.1.18 OEFitTverskyCutoff . . . 63
3.1.19 OEFlexiOverlapFunc . . . 64
3.1.20 Overlap . . . 64
3.1.21 SetupRef . . . 65
3.1.22 UpdateColorCoords . . . 65
3.1.23 OEFlexiOverlapOptions . . . 65
3.1.24 Validate . . . 65
3.1.25 GetShapeFuncType . . . 66
3.1.26 GetColorFuncType . . . 66
3.1.27 GetForceField . . . 66
3.1.28 GetHarmonic . . . 66
3.1.29 GetShapeOptions . . . 66
3.1.30 GetColorOptions . . . 66
3.1.31 GetOverlapFunc . . . 66
3.1.32 SetShapeFuncType . . . 67
3.1.33 SetColorFuncType . . . 67
3.1.34 SetForceField . . . 67
3.1.35 SetHarmonic . . . 67
3.1.36 SetShapeOptions . . . 67
3.1.37 SetColorOptions . . . 67
3.1.38 SetOverlapFunc . . . 67
3.1.39 OEFlexiOverlapResults . . . 68
3.1.40 OEFlexiOverlay . . . 69
3.1.41 SetupRef . . . 69
3.1.42 Overlay . . . 69
3.1.43 SortedOverlay . . . 69
3.1.44 BestOverlay . . . 70
3.1.45 OEFlexiOverlayOptions . . . 70
3.1.46 GetMaxOptSteps . . . 70
3.1.47 GetRigidOverlay . . . 70
3.1.48 GetFlexiOverlapOptions . . . 71
3.1.49 GetRigidOverlayOptions . . . 71
3.1.50 SetMaxOptSteps . . . 71

3.1.51 SetRigidOverlay . . . 71
3.1.52 SetFlexiOverlapOptions . . . 71
3.1.53 SetRigidOverlayOptions . . . 71
3.1.54 OEGridColorFunc . . . 71
3.1.55 OEGridShapeFunc . . . 73
3.1.56 OEHermite . . . 74
3.1.57 OEHermiteOptions . . . 76
3.1.58 GetLambdaX . . . 76
3.1.59 GetLambdaY . . . 76
3.1.60 GetLambdaZ . . . 76
3.1.61 GetNPolyMax . . . 76
3.1.62 GetUseOptimalLambdas . . . 77
3.1.63 SetLambdaX . . . 77
3.1.64 SetLambdaY . . . 77
3.1.65 SetLambdaZ . . . 77
3.1.66 SetNPolyMax . . . 77
3.1.67 SetUseOptimalLambdas . . . 77
3.1.68 OEHermiteShapeFunc . . . 77
3.1.69 OEHighestColorTanimoto . . . 79
3.1.70 OEHighestFitColorTversky . . . 79
3.1.71 OEHighestFitTversky . . . 80
3.1.72 OEHighestFitTverskyCombo . . . 80
3.1.73 OEHighestOverlap . . . 80
3.1.74 OEHighestRefColorTversky . . . 81
3.1.75 OEHighestRefTversky . . . 81
3.1.76 OEHighestRefTverskyCombo . . . 81
3.1.77 OEHighestScaledColor . . . 82
3.1.78 OEHighestTanimoto . . . 82
3.1.79 OEHighestTanimotoCombo . . . 82
3.1.80 OEHighestTverskyCombo . . . 83
3.1.81 OEInertialStarts . . . 83
3.1.82 OEIsColorAtomPred . . . 85
3.1.83 OEMultiRefOverlay . . . 85
3.1.84 OEOverlapCutoff . . . 86
3.1.85 OEOverlapFunc . . . 87
3.1.86 GetColorFunc . . . 88
3.1.87 GetShapeFunc . . . 88
3.1.88 OEOverlapFuncBase . . . 88
3.1.89 Overlap . . . 90
3.1.90 SetupFlex . . . 90
3.1.91 SetupRef . . . 90
3.1.92 OEOverlapPrep . . . 90
3.1.93 OEOverlapPrepOptions . . . 92
3.1.94 OEOverlapResults . . . 95
3.1.95 OEOverlay . . . 97
3.1.96 BestOverlay . . . 98
3.1.97 GetOverlayOptions . . . 98
3.1.98 Overlay . . . 98
3.1.99 SetupRef . . . 99
3.1.100 SortedOverlay . . . 99
3.1.101 OEOverlayOptions . . . 99
3.1.102 GetColorFuncType . . . 100
3.1.103 GetColorOptions . . . 100
3.1.104 GetMaxOptSteps . . . 100

3.1.105 GetOverlapFunc . . . 100
3.1.106 GetShapeFuncType . . . 100
3.1.107 GetShapeOptions . . . 100
3.1.108 GetStarts . . . 100
3.1.109 SetColorFuncType . . . 100
3.1.110 SetColorOptions . . . 101
3.1.111 SetMaxOptSteps . . . 101
3.1.112 SetOverlapFunc . . . 101
3.1.113 SetShapeFuncType . . . 101
3.1.114 SetShapeOptions . . . 101
3.1.115 SetStarts . . . 101
3.1.116 OEOverlayScoreCutoff . . . 101
3.1.117 OEQuatStarts . . . 102
3.1.118 OERefColorTverskyCutoff . . . 103
3.1.119 OERefTverskyComboCutoff . . . 103
3.1.120 OERefTverskyCutoff . . . 103
3.1.121 OERandomStarts . . . 104
3.1.122 OEROCS . . . 105
3.1.123 OEROCSOptions . . . 106
3.1.124 OEROCSResult . . . 109
3.1.125 OEShapeFunc . . . 111
3.1.126 OEShapeGridOptions . . . 112
3.1.127 OEShapeOptions . . . 113
3.1.128 OEStarts . . . 114
3.1.129 OEShapeQuery . . . 117
3.1.130 OEShapeQueryBase . . . 120
3.1.131 OESubrocsStarts . . . 123
3.1.132 OETanimotoComboCutoff . . . 124
3.1.133 OETanimotoCutoff . . . 124
3.2 OEShape Constants . . . 124
3.2.1 OEColorFFType . . . 124
3.2.2 OEDerivativeType . . . 125
3.2.3 OEExponentialType . . . 126
3.2.4 OEMomentSymmetry . . . 126
3.2.5 OEOverlapRadii . . . 126
3.2.6 OEScoreType . . . 127
3.2.7 OEShapeType . . . 127
3.2.8 OEColorType . . . 128
3.3 OEShape Functions . . . 128
3.3.1 OEAddColorAtom . . . 128
3.3.2 OEAddColorAtoms . . . 129
3.3.3 OEAddCoordsToColorAtom . . . 129
3.3.4 OECalcShapeMultipoles . . . 129
3.3.5 OECalcVolume . . . 130
3.3.6 OEClearCachedSelfColor . . . 130
3.3.7 OEClearCachedSelfShape . . . 130
3.3.8 OECompressColorAtoms . . . 130
3.3.9 OECountColorAtoms . . . 130
3.3.10 OEDeleteCompressedColorAtoms . . . 130
3.3.11 OEGetCachedSelfColor . . . 131
3.3.12 OEGetCachedSelfShape . . . 131
3.3.13 OEGetCenterAndExtents . . . 131
3.3.14 OEGetColorAtoms . . . 131
3.3.15 OEGetColorParents . . . 131

3.3.16 OEGetColorType . . . 132
3.3.17 OEGetMomentsOfInertiaTransform . . . 132
3.3.18 OEHasCachedSelfColor . . . 132
3.3.19 OEHasCachedSelfShape . . . 132
3.3.20 OEHasColorAtoms . . . 132
3.3.21 OEHasCompressedColorAtoms . . . 133
3.3.22 OEHasPerceivedColorAtoms . . . 133
3.3.23 OEIsColorAtom . . . 133
3.3.24 OEIsFastROCSShapeQuery . . . 133
3.3.25 OEMol2Query . . . 133
3.3.26 OEOrientByMomentsOfInertia . . . 133
3.3.27 OEReadShapeQuery . . . 134
3.3.28 OERemoveColorAtoms . . . 134
3.3.29 OEROCSOverlay . . . 134
3.3.30 OESelfColor . . . 134
3.3.31 OESelfShape . . . 135
3.3.32 OESetCachedSelfColor . . . 135
3.3.33 OESetCachedSelfShape . . . 135
3.3.34 OESetColorParents . . . 136
3.3.35 OESetColorType . . . 136
3.3.36 OESetCoordsFromColorParents . . . 136
3.3.37 OEShapeGetArch . . . 136
3.3.38 OEShapeGetLicensee . . . 136
3.3.39 OEShapeGetPlatform . . . 136
3.3.40 OEShapeGetRelease . . . 136
3.3.41 OEShapeGetSite . . . 137
3.3.42 OEShapeGetVersion . . . 137
3.3.43 OEShapeIsLicensed . . . 137
3.3.44 OESortOverlayScores . . . 137
3.3.45 OEUncompressColorAtoms . . . 138
3.3.46 OEWriteShapeQuery . . . 138

4 Preliminary API **139**
4.1 Preliminary OEShape Classes . . . 139
4.1.1 OEShapeQueryOptions . . . 139
4.1.2 OEHermiteOverlayOptions . . . 141
4.1.3 GetHermiteOptions . . . 141
4.1.4 GetMaxOptSteps . . . 141
4.1.5 GetStarts . . . 141
4.1.6 SetHermiteOptions . . . 141
4.1.7 SetMaxOptSteps . . . 142
4.1.8 SetStarts . . . 142
4.1.9 OEHermiteOverlay . . . 142
4.1.10 BestOverlay . . . 142
4.1.11 SetupRef . . . 143
4.2 Preliminary OEShape Constants . . . 143
4.2.1 OEGridScoreType . . . 143

5 Deprecated API **145**
5.1 Deprecated OEShape Classes . . . 145
5.1.1 OEBestOverlay . . . 145
5.1.2 OEColorOverlap . . . 150
5.1.3 OEColorResults . . . 151
5.1.4 OEHighestComboScore . . . 152

5.1.5 OEOverlap . . . 152
5.1.6 OEShapeQueryPublic . . . 155
5.2 Deprecated OEShape Constants . . . 155
5.2.1 OEBOMinType . . . 155
5.2.2 OEBOOrientation . . . 156
5.2.3 OEOverlapMethod . . . 156
5.2.4 OEOverlapRepresentation . . . 157

6 Release History **159**
6.1 Shape TK 3.10.0 . . . 159
6.1.1 New Features . . . 159
6.1.2 Minor Bug Fixes . . . 159
6.2 Shape TK 3.9.0 . . . 159
6.2.1 New Features . . . 159
6.2.2 Minor Bug Fixes . . . 160
6.3 Shape TK 3.8.0 . . . 160
6.3.1 New Features . . . 160
6.4 Shape TK 3.7.0 . . . 160
6.4.1 New features . . . 160
6.4.2 Minor bug fixes . . . 160
6.5 Shape TK 3.6.2 . . . 161
6.5.1 New features . . . 161
6.6 Shape TK 3.6.1 . . . 161
6.6.1 Major bug fixes . . . 161
6.7 Shape TK 3.6.0 . . . 161
6.7.1 New features . . . 161
6.7.2 Major bug fixes . . . 161
6.7.3 Minor bug fixes . . . 161
6.8 Shape TK 3.5.1 . . . 162
6.8.1 New features . . . 162
6.8.2 Major bug fixes . . . 162
6.8.3 Minor bug fixes . . . 162
6.8.4 Documentation changes . . . 162
6.9 Shape TK 3.5.0 . . . 163
6.9.1 New features . . . 163
6.9.2 Minor bug fixes . . . 163
6.10 Shape TK 3.4.3 . . . 163
6.10.1 New features . . . 163
6.10.2 Minor bug fixes . . . 163
6.11 Shape TK 3.4.2 . . . 163
6.11.1 Major bug fixes . . . 163
6.12 Shape TK 3.4.1 . . . 164
6.12.1 New features . . . 164
6.12.2 Major bug fixes . . . 164
6.12.3 Minor bug fixes . . . 164
6.13 Shape TK 3.4.0 . . . 164
6.13.1 Minor bug fixes . . . 164
6.14 Shape TK 2.0.4 . . . 165
6.14.1 New features . . . 165
6.14.2 Minor bug fixes . . . 165
6.15 Shape TK 2.0.3 . . . 165
6.15.1 New features . . . 165
6.15.2 Minor bug fixes . . . 165
6.15.3 Documentation changes . . . 165

6.16 Shape TK 2.0.2 . . . 165
6.16.1 New features . . . 165
6.16.2 Minor bug fixes . . . 165
6.16.3 Java-specific changes . . . 166
6.16.4 C#-specific changes . . . 166
6.16.5 Documentation changes . . . 166
6.17 Shape TK 2.0.1 . . . 166
6.17.1 New features . . . 166
6.17.2 Minor bug fixes . . . 166
6.17.3 Documentation changes . . . 166
6.18 Shape TK 2.0.0 . . . 166
6.18.1 New features . . . 166
6.18.2 Minor Bug Fixes . . . 168
6.18.3 Documentation changes . . . 168
6.19 Shape TK 1.11.0 . . . 168
6.19.1 New features . . . 168
6.19.2 Documentation changes . . . 169
6.20 Shape TK 1.10.4 . . . 169
6.21 Shape TK 1.10.3 . . . 169
6.21.1 Python-specific changes . . . 169
6.21.2 Java-specific changes . . . 169
6.22 Shape TK 1.10.2 . . . 169
6.22.1 Major bug fixes . . . 169
6.22.2 Minor bug fixes . . . 169
6.23 Shape TK 1.10.1 . . . 169
6.23.1 New features . . . 169
6.23.2 Documentation fixes . . . 170
6.24 Shape TK 1.10.0 . . . 170
6.24.1 Minor bug fixes . . . 170
6.25 Shape TK 1.9.7 . . . 170
6.25.1 New features . . . 170
6.25.2 Major bug fixes . . . 170
6.25.3 Minor bug fixes . . . 170
6.25.4 Documentation changes . . . 170
6.26 Shape TK 1.9.6 . . . 170
6.26.1 New features . . . 170
6.26.2 Major bug fixes . . . 170
6.26.3 Minor bug fixes . . . 171
6.27 Shape TK 1.9.5 . . . 171
6.27.1 New features . . . 171
6.27.2 Major bug fixes . . . 171
6.28 Shape TK 1.9.4 . . . 171
6.28.1 Major bug fixes . . . 171
6.28.2 Minor bug fixes . . . 171
6.29 Shape TK 1.9.3 . . . 171
6.30 Shape TK 1.9.2 . . . 171
6.30.1 Minor bug fixes . . . 171
6.31 Shape TK 1.9.1 . . . 171
6.31.1 Bug fixes . . . 171
6.32 Shape TK 1.9.0 . . . 171
6.32.1 New features . . . 171
6.32.2 Major bug fixes . . . 172
6.32.3 Minor bug fixes . . . 172
6.33 Shape TK 1.8.3 . . . 172

6.33.1 New features . . . 172
6.33.2 Minor bug fixes . . . 172
6.34 Shape TK 1.8.2 . . . 172
6.35 Shape TK 1.8.1 . . . 172
6.35.1 Bug fixes . . . 172
6.36 Shape TK 1.8.0 . . . 172
6.36.1 New features . . . 172
6.36.2 Bug fixes . . . 173
6.37 Shape TK 1.7.2 . . . 173
6.37.1 New features . . . 173
6.37.2 Bug fixes . . . 174
6.38 Shape TK 1.7.1 . . . 174
6.38.1 New features . . . 174
6.38.2 Bug fixes . . . 174
6.39 Shape TK 1.7.0 . . . 174
6.39.1 New features . . . 174
6.39.2 Bug fixes . . . 175
6.40 Shape TK 1.6.2 . . . 175
6.40.1 New features . . . 175
6.40.2 Bug fixes . . . 175
6.41 Shape TK 1.6.1 . . . 175
6.41.1 New features . . . 175
6.41.2 Bug fixes . . . 175
6.42 Shape TK 1.6 . . . 175
6.42.1 New features . . . 175
6.42.2 Bug fixes . . . 175
6.43 Shape TK 1.5 . . . 176
6.43.1 New features . . . 176

7 Copyright and Trademarks **177**

8 Sample Code **179**

9 Citation **181**
9.1 Orion® . . . 181
9.2 Orion Floes . . . 181
9.3 Toolkits and Applications . . . 181
9.4 Publications for Bibliographies . . . 183
9.4.1 Orion . . . 183
9.4.2 AFITT and FLYNN Applications . . . 183
9.4.3 OEDocking Application and Toolkit . . . 183
9.4.4 OMEGA Application and Toolkit . . . 183
9.4.5 ROCS Application . . . 184
9.5 OpenEye MMDS Web Service . . . 184

10 Technology Licensing **185**
10.1 GCC . . . 257
10.1.1 GCC RUNTIME LIBRARY EXCEPTION . . . 258
10.1.2 GNU GENERAL PUBLIC LICENSE . . . 259

Index **273**

CHAPTER

ONE

THEORY

1.1 Theory

1.1.1 Molecular Shape

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

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

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

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

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

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

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

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

Multiplying the terms in the integral out gives:

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

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

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

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

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

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

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

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

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

1.1.2 Shape Characteristics and the Use of Gaussians

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1.1.3 Shape from Hermite Representation

So far, discussion has focused on a multi-center expansion of molecular shape based on a Gaussian representation of atoms. An interesting alternative is a single-center expansion of the shape into a complete set of functions. One possibility is to use a product of three Hermite functions, one per space dimension and expand the shape into the resulting complete set of functions. Expansion of an electron-density map into the three-dimensional basis of Hermite polynomials has been previously considered in Ref. [Derevyanko-2014].

Hermites are a class of functions known as "special" functions. They are special because linear combinations of these functions can reproduce any other reasonably behaved function. Just as the Discrete Fourier Transform shows that any periodic function can be represented by a sum of the trigonometric functions $\{sin(n\varphi), cos(n\varphi)\}$, Hermite functions can be used to represent any localized, non-periodic function such as molecular shape. The "special" in special functions refers to the fact that each function in one of these classes is orthogonal to every other function. This means that the integral of the product of any two such functions is 1 if they are the same function, or zero if they are not. This makes much of the mathematics of finding the coefficients of the representational linear sum much simpler: a coefficient is then just the integral of the product of each particular special function with the function being represented.

There are many well-known special functions such as sine and cosine, Hermite polynomials, Legendre polynomials, and Laguerre polynomials. So why work with Hermite functions? The art of special functions is to choose one that fits the purpose. For instance, for a periodic function, sine and cosine make sense. Hermites make sense for shape because of their form. The **Hermite function** of order n (any non-negative integer) is defined as follows:

$$\mathrm{H}_n(x) = H_n(x)e^{-x^2/2}$$

The **Hermite polynomial** is defined as:

$$H_n(x) = (-1)^n e^{x^2} \left(\frac{d}{dx}\right)^n e^{-x^2}.$$

The first few Hermite polynomials are:

$$H_0(x) = 1, \quad H_1(x) = 2x, \quad H_2(x) = 4x^2 - 2, \quad H_3(x) = 8x^3 - 12x.$$

Hermite functions can be thought of as generalizations of Gaussian function to a complete set: this is why we like them as candidates for representing molecular shape! Since its formation, OpenEye has worked with molecular shape as a sum Gaussian ([Grant-1995], [Grant-1996], [Grant-1997]) and Hermites represent an extension of that work. The difference is that Hermites are all centered at the origin, while the "classical" OpenEye representation of molecular shape is to place a Gaussian at each non-hydrogen atom.

So, are there any advantages to using a single-centered Hermite representation rather than a multi-centered set of Gaussians? This is a similar question to the one we first posed at the beginning of OpenEye: Does representing shape by Gaussians give us any advantages over the canonical "sum of spheres" representation? We felt that it did.

Shape became a smooth function that seemed more physical than sharp spherical functions, and they allowed a more efficient molecular overlay of shapes. We postulated that this might make a better dielectric function for Poisson-Boltzmann, which has been largely verified in **ZAP TK** ([Grant-2001]). The concept seemed very general, so we embedded it into an OpenEye toolkit, **Shape TK**, so that we and our customers could explore different uses. This led to applications in crystallography (**AFITT**), bioisostere replacement (**BROOD**), docking (**FRED**), pose prediction using ligand information (**POSIT**), and shape-based alignment (**ROCS**).

So what advantages might Hermites represent? For one, they are very compact (i.e., they have few coefficients). Second, the more coefficients we include (i.e., higher order functions), the more exact the match to a sum of atom-centered Gaussians; conversely, the fewer functions, the more "coarse" the representation becomes, while retaining the smooth properties we like about Gaussians in the first place. Third, the Fourier Transform of a Hermite function is the same function! If we imagine wanting to generate Fourier representations of shape, Hermites make it easy. Fourth, the overlap of two shapes represented by Hermite functions is just the sum of the product of the coefficients: it's that easy! It is not difficult to imagine ways in which we could apply Hermites to the same problems we currently tackle with atom-centered Gaussians.

The first application that has already intrigued us is the representation of protein shapes. Consider that an arbitrary molecular shape can be expanded into the following combination of Hermite functions:

$$f(x,y,z) = \sum_{\ell,m,n}^{\infty} f_{\ell mn}\, \mathrm{H}_\ell(\lambda_x x)\, \mathrm{H}_m(\lambda_y y)\, \mathrm{H}_n(\lambda_z z).$$

The (infinite) set of coefficients for $f_{\ell mn}$ forms the Hermite representation. Due to reflection properties of Hermite polynomials, we can assume without loss of generality that the coefficients $\lambda_x, \lambda_y, \lambda_z$ are all positive. In practice we take a finite sum in the above equation by truncating it with the following condition on ℓ, m, n:

$$\ell + m + n \leq \mathrm{NPolyMax},$$

where NPolyMax is a resolution parameter, and can be varied from 0 (very inaccurate expansion) to infinity (exact Hermite expansion). The recommended value varies depending on the size of the molecule.

Below is an example of the Hermite representation of the protein DHFR.

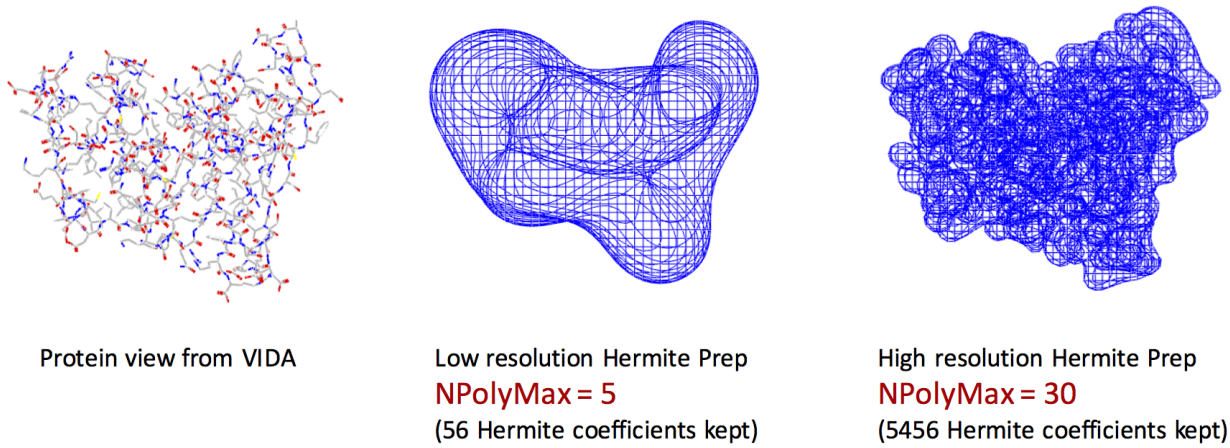


Fig. 1: **Exact shape versus Hermite Representation**

The left figure represents the VIDA view of the exact shape of the protein. The middle figure shows the Hermite representation with NPolyMax = 5. The right figure shows the Hermite representation with NPolyMax = 30. Remarkably, we can encode the main features of the protein shape using Hermite representation with only 56 floating point numbers (middle figure).

The advantages of representing proteins by Hermites include:

- smooth representations that capture any level of detail, from the atomistic (many coefficients, as in the picture on the right) to the very coarse (few coefficients, as in the middle picture),
- the ability to store these representations in a compact manner,
- the ability to transform these representations easily,
- very fast overlap calculations between proteins.

Of course, the mathematics of Hermites is more complex than that of a set of atom-centered Gaussians. We have to know how to make, rotate, translate, and scale such representations. Therefore, we are releasing this toolkit with no immediate application or goal - rather, in the spirit of OpenEye, to make potentially useful tools for our customers.

See also

- *OEHermite* class
- *OEHermiteOptions* class
- *Shape from Hermite expansion* examples

1.1.4 Color Features

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

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

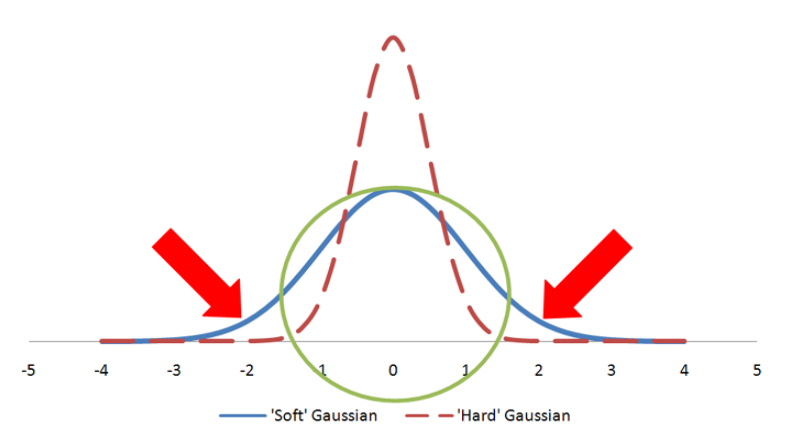


Fig. 2: **Hard vs. Soft Gaussians**
A sphere described by two different Gaussian functions. The 'hard' Gaussian (dashed) is the one employed by **Shape TK** to approximate a color atom sphere.

Shape TK comes pre-loaded with two color force fields, Implicit Mills Dean (default) and Explicit Mills Dean.

These are described in associated color force field files (*.cff). The desired force field file can be supplied to the *OEColorForceField.Init* method. Further information on editing color force field files is given in the below section *Color File Format*.

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

- Color atom types
- **The functional groups to which the color atoms should be applied.**
  Shape TK uses only the heavy atoms of molecules; hydrogens are ignored.
- Whether the interaction between color atoms is attractive or repulsive. Interactions between color atoms of the same type are always attractive. The weight term describes the strength of the interaction relative to the shape gradients and the range term affects the range of the interaction.

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

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

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

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

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

Hydrophobe
Terminal or non-terminal aromatic or aliphatic groups e.g. phenyl

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

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

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

Color File Format

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

```
DEFINE hetero [#7,#8,#15,#16]
DEFINE notNearHetero [!#1;!$($hetero);!$(\*[$hetero])]
#
#
TYPE donor
TYPE acceptor
TYPE rings
TYPE positive
TYPE negative
TYPE structural
```

(continues on next page)

(continued from previous page)

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

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

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

Built-in Color Force Fields

Two color force fields, ImplicitMillsDean and ExplicitMillsDean, are built into the Shape toolkit. Both of these force fields define similar 6 TYPE color force-fields. The types are hydrogen-bond donors, hydrogen-bond acceptors, hydrophobes, anions, cations, and rings. The ImplicitMillsDean force field is recommended.

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

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

1.1.5 Similarity Measures

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

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

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

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

Tanimoto

Formula:

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

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

Tversky

Formula:

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

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

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

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

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

1.1.6 Implementation Details and Usage

Overlap Functions

The overlap functions extend the molecule objective function (OEMolFunc1) interfaces, define in **OEFF TK**, with overlap interactions, through *OEOverlapFuncBase*. Similar to the force fields, a shape or a color is also defined as a collection of pairwise interactions. Implementation of overlaps as extension of the molecule objective functions allows for use of the standard optimizers (OEOptimizer1 and/or OEOptimizer2) for minimization of overlaps and corresponding quantities. Implementation of overlaps as extension of the molecule objective functions also allows for combining

the overlap functions with other molecule objective functions, such as the force fields, for system optimization. An overlap function in **Shape TK** can consist of just shape or color, or a combination of both.

The overlap functions are the simplest objects in **Shape TK**, and can is used to calculate simple, static, overlap between two objects (molecules, grids or shape queries). Note that static means that the two input species (ref and fit) are not moved at all. These objects simply calculate the static overlap, gradients and other corresponding quantities, given the input positions.

Overlap Methods

Three different algorithms are implemented to calculate the overlap and the corresponding gradients, between two gaussians, that provides a balance between accuracy and speed.

The *Exact* option considers all pairs of atoms in the system, and the Gaussian overlaps are calculated exactly.

The *Analytic* option provides multiple options to approximate the calculations. The first option corresponds to using a pre-calculated table for estimating the exponentials vs. doing exact calculation of the exponentials. The second option allows to use a proximity grid to only consider those atom pairs that are within a certain threshold distance (by default 3.0 Å). The average error introduced by either of these approximations is in the order of one part in a thousand. The speed gain from the proximity grid increases with size of the molecules, as well as with the number of overlap calculations performed. The later is related to the increased setup time for the proximity grid for the reference molecule. If both the tabulated exponentials and the proximity grid is turned off, the *Analytic* option simply folds back to the *Exact* option.

The final option, *Grid*, uses a grid representation of the volume of the target molecule. It requires significant set-up time relative to the cost of a single overlap calculation (~0.01s compared to ~0.0001s) but is significantly faster than other methods for the evaluation of each overlap once set. *Grid* suffers a few caveats and drawbacks. First is that, currently, for shape overlap calculation, all atoms are treated as if they have the same radius (that assigned to carbon). The second is that the approximation is slightly worse, typically a few parts in a thousand, at typical grid resolutions.

Both *Analytic* and *Grid* improve performance when the fit molecule is large (>20 atoms) because, if there are **n** atoms in the fit and **m** in the reference, the work per atom in the fit is proportional to a constant not **n**.

Besides the Gaussian based calculations, a fourth option, *Hermite* is available for shape overlap calculation, based on Hermite representation of shape as described above earlier.

Radii approximations

Since we are considering molecular volume overlap as a measure of shape, the radii used for each atom is important. There are essentially two settings. A radii approximation of `OEOverlapRadii.All` means each atom will be treated with the radius as passed in. Alternatively, one can treat all heavy atoms as similar and apply the `OEOverlapRadii.Carbon` radius approximation. With this, all heavy atoms will be assigned the same radius, regardless of the value attached to each atom. As noted above, if the *Grid* method is used for overlap calculation, only the `OEOverlapRadii.Carbon` radius approximation can be used.

Overlap Optimization and Overlay

Two high level objects, *OEROCS* and *OEOverlay* are provided for optimizing a *OEOverlapFuncBase*, to maximize shape (and color) overlap between two objects and obtain the best overlay. Both the *OEROCS* and *OEOverlay* optimizes the overlap between the reference and fit object, but provides different level of usability.

Starting positions for optimization

The starting points are important for any optimization. Various options to start an overlap optimization is provided through the *OEStarts* interface. The reference structure is always aligned by its principal moments of inertia, and the initial alignments of fit structure is determined by the specified options.

In the *Inertial* start, in general, the fit object is aligned in 4 positions with the primary and secondary moments of inertia in both directions. In order to deal with structures with symmetrical moments of inertia, additional starting points are

generated. For a reference or fit where the 2 major moments of inertia are equal (to a user-defined percent, nominally 15%), 4 extra starting positions are generated. In the rare case of a molecule with all 3 moments of inertia essentially equal, 20 random starting translations and rotations are chosen as starting positions. For virtual screening uses, where the reference and fit molecule are similar in size, the inertial starts provide an excellent choice for starting positions that balances quality results with speed.

There are times when there is a large difference in size and a more elaborate search is needed. For these, there are a couple of built-in searches as well a user-defined search. To increase searching, instead of doing the inertial frame rotations with to 2 molecule centers-of-mass (COM) aligned, we can do a set of inertial rotations at many more locations.

The *AtAtom* starts will translate the COM of the fit molecule to the specified atom center (usually the heavy atoms) locations of the reference molecule. At each of these translations, it will perform the 4 basic inertial transforms. Note that this means that instead of 4 (or 8) starting positions, there will be 4 x number of reference molecule heavy atoms (or what other atom centers specified) starts, resulting in 20-30 more starting positions compared to the *Inertial* starts. Obviously this will slow the overall calculation so this is not recommended for high-volume virtual screening, but is very useful for low-volume fragment searching. A slightly less aggressive set of starting positions can be to use the color atom centers instead of the heavy atoms of the reference molecule. For an average reference molecule with ~10 color atoms, this will be faster than using all heavy atom positions, but not as elaborate of a search.

The *Subrocs* starts automatically performs *AtAtom* starts on each heavy atom of the larger molecule, regardless of which molecule is set as the reference.

With the *Cartesian* starts, the user can specify the points at which to translate the fit molecule before doing the 4 basic inertial transforms. This provides more flexibility in terms of how to starts to do for any optimization.

The *Random* starts, will generate **N** random starting positions where **N** is user-defined, as well as the maximum translation allowed between random starts. The fit molecule will be translated to these random starting positions, and the basic inertial transforms will be applied.

The *AsIs* starts gives the option to the user to start the optimization from a single, pre-aligned position.

And finally, the *Quat* starts takes the user specified quaternions as the transformations to be applied to the fit molecule before performing optimizations.

Thus, for any given optimization, there are multiple results returned, usually only one of which is useful.

Preparing Molecules

Tools for preparing molecules are provided separately so users can prepare their molecules once and reuse those molecules for many calculations. However, the high level *OEROCS* object for overlay optimization also provides a convenient option to automatically prepare molecules, if desired.

There are essentially three components to preparing molecules.

The first is to choose either to have hydrogens explicitly or not, and add or remove the hydrogens in the molecules. Shape is a heavy atom property and most OpenEye uses (as well as the default option in *OEOverlapPrep*) is to ignore the hydrogens.

The second is to assign Bondi radii on the atoms, if desired. This is required if a radii approximation of `OEOverlapRadii.All` is to be used for shape overlap calculations.

The third is to choose to have color atoms or not. Even if color is used for the calculations, it is possible to have user defined color atoms on the molecules and direct *OEOverlapPrep* to not make modifications to them (see `OEOverlapPrep.SetAssignColor`).

CHAPTER

TWO

EXAMPLES

2.1 Calculating Overlap

Overlap calculation is the simplest functionality offered through the **Shape TK**. These examples show how to calculate static overlap between two objects (molecules, grids or shape query). Note that static means that the two input species (ref and fit) are not moved at all. These examples simply calculate the overlap and/or other related quantities, given the input positions. Examples for performing calculations that actually optimize the alignment/overlap are considered in a separate section.

2.1.1 Simple Overlap

This example reads in a reference molecule and a few fit molecules and prints out the *Exact* overlap calculated.

Listing 1: Simple overlap using Exact Overlap

```
/*
(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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class SimpleOverlap {

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

(continues on next page)

(continued from previous page)

```
        if (args.length!=2) {
            System.out.println("SimpleOverlap <reffile> <fitfile>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);

        OEGraphMol refmol = new OEGraphMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();

        // Prepare reference molecule for calculation
        // With default options this will remove any explicit hydrogens present
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.Prep(refmol);

        // Get appropriate function to calculate exact shape
        OEExactShapeFunc shapeFunc = new OEExactShapeFunc();
        shapeFunc.SetupRef(refmol);

        OEOverlapResults res = new OEOverlapResults();
        OEGraphMol fitmol = new OEGraphMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            prep.Prep(fitmol);
            shapeFunc.Overlap(fitmol, res);
            System.out.print(fitmol.GetTitle());
            System.out.println(" exact tanimoto = "+res.GetTanimoto());
        }
    }
}
```

2.1.2 Adding overlap to molecules

This next example program uses the *Analytic* overlap method and also uses a little extra OEChem to attach the overlap scores to each molecule as SD data. This can be used to rescore a set of ROCS hits or to measure the overlap of ROCS alignments against an exclusion region in the binding site.

Listing 2: Rescoring pre-aligned structures

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
```

(continues on next page)

(continued from previous page)

```
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class Rescore {

    public static void main(String[] args) {
        if (args.length!=3) {
            System.out.println("Rescore <reffile> <rocs_hits> <output.sdf>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);
        oemolostream outfs = new oemolostream(args[2]);

        OEGraphMol refmol = new OEGraphMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();

        // Get appropriate function to calculate analytic shape
        OEAnalyticShapeFunc shapeFunc = new OEAnalyticShapeFunc();
        shapeFunc.SetupRef(refmol);

        OEOverlapResults res = new OEOverlapResults();
        OEGraphMol fitmol = new OEGraphMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            shapeFunc.Overlap(fitmol, res);
            oechem.OESetSDData(fitmol, "AnalyticTanimoto",
                    String.valueOf(res.GetTanimoto()));
            oechem.OEWriteMolecule(outfs, fitmol);
        }
    }
}
```

2.1.3 Color Overlap

This example reads in a reference molecule and a few fit molecules and prints out the *Exact* color score calculated. By default all color calculations are performed using the ImplicitMillsDean force field.

Listing 3: Calculating color score

```
/*
(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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class ColorOverlap {

    public static void main(String[] args) {
        if (args.length!=2) {
            System.out.println("ColorOverlap <reffile> <overlayfile>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);

        OEGraphMol refmol = new OEGraphMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();

        // Prepare reference molecule for calculation
        // With default options this will remove any explicit
        // hydrogens present, and add required color atoms
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.Prep(refmol);

        // Get appropriate function to calculate exact color
        OEExactColorFunc colorFunc = new OEExactColorFunc();
        colorFunc.SetupRef(refmol);

        OEOverlapResults res = new OEOverlapResults();
        OEGraphMol fitmol = new OEGraphMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            prep.Prep(fitmol);
            colorFunc.Overlap(fitmol, res);
            System.out.print("Fit. Title: "+fitmol.GetTitle()+" ");
            System.out.print("Color Score: "+res.GetColorScore()+" ");
```

(continues on next page)

(continued from previous page)

```
        }
    }
}
```

2.1.4 Adding color score to molecules

This next example uses the ImplicitMillsDean force field and the *Analytic* color to rescore a set of ROCS hits and add the color scores to SD tags.

Listing 4: Using color to add scores to pre-aligned molecules.

```
/*
(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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class ColorScore {

    public static void main(String[] args) {
        if (args.length!=3) {
            System.out.println("ColorScore <reffile> <overlayfile> <outfile>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);
        oemolostream outfs = new oemolostream(args[2]);

        OEGraphMol refmol = new OEGraphMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();

        // Prepare reference molecule for calculation
        // With default options this will add required color atoms
```

(continues on next page)

(continued from previous page)

```
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.Prep(refmol);

        // Get appropriate function to calculate analytic color
        OEAnalyticColorFunc colorFunc = new OEAnalyticColorFunc();
        colorFunc.SetupRef(refmol);

        OEOverlapResults res = new OEOverlapResults();
        OEGraphMol fitmol = new OEGraphMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            prep.Prep(fitmol);
            colorFunc.Overlap(fitmol, res);
            oechem.OESetSDData(fitmol, "MyColorTanimoto", String.valueOf(res.
↪GetColorTanimoto()));
            oechem.OESetSDData(fitmol, "MyColorScore",  String.valueOf(res.
↪GetColorScore()));
            oeshape.OERemoveColorAtoms(fitmol);
            oechem.OEWriteMolecule(outfs, fitmol);

            System.out.print("Fit. Title: "+fitmol.GetTitle()+" ");
            System.out.print("Color Tanimoto: "+res.GetColorTanimoto()+" ");
            System.out.print("Color Score: "+res.GetColorScore()+" ");
        }
    }
}
```

2.1.5 User Defined Color

As a step toward writing a complete color force field, it is possible to combine built-in rules for color atom assignment with user defined interactions. A new *OEColorForceField* object can be created using one of the built-in types, then the interactions can be cleared using *OEColorForceField.ClearInteractions* and subsequent user interactions added with *OEColorForceField.AddInteraction*.

For example, to use the ImplicitMillsDean atom typing rules, but to only consider donor-donor and acceptor-acceptor interactions, one can use the following:

Listing 5: Using ImplicitMillsDean with user interactions.

```
/*
(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
```

(continues on next page)

(continued from previous page)

```
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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class UserColor {

    public static void main(String[] args) {
        if (args.length!=2) {
            System.out.println("UserColor <reffile> <overlayfile>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);

        OEGraphMol refmol = new OEGraphMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();

        // Modify ImplicitMillsDean color force field by
        // adding user defined color interactions
        OEColorForceField cff = new OEColorForceField();
        cff.Init(OEColorFFType.ImplicitMillsDean);
        cff.ClearInteractions();
        int donorType = cff.GetType("donor");
        int accepType = cff.GetType("acceptor");
        cff.AddInteraction(donorType, donorType, "gaussian", -1.0f, 1.0f);
        cff.AddInteraction(accepType, accepType, "gaussian", -1.0f, 1.0f);

        // Prepare reference molecule for calculation
        // With default options this will add required color atoms
        // Set the modified color force field for addignment
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.SetColorForceField(cff);
        prep.Prep(refmol);

        // Get appropriate function to calculate exact color
        // Set appropriate options to use the user defined color
        OEColorOptions options = new OEColorOptions();
        options.SetColorForceField(cff);
        OEExactColorFunc colorFunc = new OEExactColorFunc(options);
        colorFunc.SetupRef(refmol);

        OEOverlapResults res = new OEOverlapResults();
        OEGraphMol fitmol = new OEGraphMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            prep.Prep(fitmol);
```

(continues on next page)

(continued from previous page)

```
            colorFunc.Overlap(fitmol, res);
            System.out.print("Fit. Title: "+fitmol.GetTitle()+" ");
            System.out.println("Color TAnimoto: "+res.GetColorTanimoto());
        }
    }
}
```

2.1.6 Shape and Color Overlap

This example reads in a reference molecule and a few fit molecules and prints out both the shape overlap and the color score calculated. By default the OEOverlapFunc uses the *Grid* shape and *Exact* color.

Listing 6: Calculating both shape and color overlap

```
/*
(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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class OverlapColor {

    public static void main(String[] args) {
        if (args.length!=2) {
            System.out.println("OverlapColor <reffile> <fitfile>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);

        OEGraphMol refmol = new OEGraphMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();
```

(continues on next page)

(continued from previous page)

```
        // Prepare reference molecule for calculation
        // With default options this will remove any explicit hydrogens present
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.Prep(refmol);

        // Get appropriate function to calculate both shape and color
        // By default the OEOverlapFunc contains OEGridShapeFunc for shape
        // and OEExactColorFunc for color
        OEOverlapFunc func = new OEOverlapFunc();
        func.SetupRef(refmol);

        OEOverlapResults res = new OEOverlapResults();
        OEGraphMol fitmol = new OEGraphMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            prep.Prep(fitmol);
            func.Overlap(fitmol, res);
            System.out.print(fitmol.GetTitle());
            System.out.println(" tanimoto combo = "+res.GetTanimotoCombo());
            System.out.println(" shape tanimoto = "+res.GetTanimoto());
            System.out.println(" color tanimoto = "+res.GetColorTanimoto());
        }
    }
}
```

2.2 ROCS

The high-level *OEROCS* provides the simplest way to maximize shape and/or color between two objects. OEROCS uses the same shape and color functions as used for static overlap calculation, but optimizes the overlap between the reference and the fit objects. By default both shape and color optimization is performed, with shape estimated with the *Grid* method and color estimated with the *Exact* method.

The fit object is not moved during the optimization. Part of the results returned from the calculation are the transform required to move the fit object into the final orientation. The results also contain a copy of the fit molecule conformer(s) in its final orientation.

A helper function OEROCSOverlay is created for the most simple usage of the functionality that performs optimizations between a set of conformers and returns the best hit.

2.2.1 Best Hit

This example reads in a reference molecule and a fit molecule, performs overlay optimization, finds the best hit conformer and prints it out in final orientation.

Listing 7: Finding best hit conformer pair

```
/*
(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,
```

(continues on next page)

(continued from previous page)

```
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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class BestHit {

    public static void main(String[] args) {
        if (args.length!=3) {
            System.out.println("BestHit <reffile> <fitfile> <outfile>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);
        oemolostream outfs = new oemolostream(args[2]);

        OEMol refmol = new OEMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();

        OEMol fitmol = new OEMol();
        oechem.OEReadMolecule(fitfs, fitmol);
        fitfs.close();

        OEROCSResult res = new OEROCSResult();
        oeshape.OEROCSOverlay(res, refmol, fitmol);
        OEMolBase outmol = res.GetOverlayConf();

        oechem.OEWriteMolecule(outfs, outmol);
        System.out.println("Tanimoto combo = "+res.GetTanimotoCombo());
    }
}
```

2.2.2 Top Hits

This example reads in a reference molecule and a few fit molecules, performs overlay optimization, finds the few top hits, and prints them out in final orientation.

Listing 8: Finding top ROCS hits

```
/*
(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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class TopHits {

    public static void main(String[] args) {
        if (args.length!=4) {
            System.out.println("TopHits <reffile> <fitfile> <outfile> <nhits>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);
        oemolostream outfs = new oemolostream(args[2]);
        int nhits = Integer.parseInt(args[3]);

        OEMol refmol = new OEMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();

        // Setup OEROCS with specified number of best hits
        OEROCSOptions options = new OEROCSOptions();
        options.SetNumBestHits(nhits);
        OEROCS rocs = new OEROCS(options);
        rocs.SetDatabase(fitfs);

        for (OEROCSResult res : rocs.Overlay(refmol))
        {
            OEGraphMol outmol = new OEGraphMol(res.GetOverlayConf());
            oechem.OEWriteMolecule(outfs, outmol);
            System.out.println(" title = "+outmol.GetTitle());
```

(continues on next page)

(continued from previous page)

```
            System.out.println(" tanimoto combo = "+res.GetTanimotoCombo());
        }
    }
}
```

2.2.3 Top Hits with Multiple Conformers

This example repeats the Top Hits example above but fixes the number of hits to 3 and asks for multiple best conformers for each hit reported.

Listing 9: Finding top ROCS hits with multiple 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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class TopHitsConf {

    public static void main(String[] args) {
        if (args.length!=4) {
            System.out.println("TopHitsConf <reffile> <fitfile> <outfile> <confsperhit>
↪");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);
        oemolostream outfs = new oemolostream(args[2]);
        int nconfs = Integer.parseInt(args[3]);

        OEMol refmol = new OEMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();
```

(continues on next page)

(continued from previous page)

```
        // Setup OEROCS with specified number of best hits
        OEROCSOptions options = new OEROCSOptions();
        options.SetNumBestHits(3);
        options.SetConfsPerHit(nconfs);
        OEROCS rocs = new OEROCS(options);
        rocs.SetDatabase(fitfs);

        for (OEROCSResult res : rocs.Overlay(refmol))
        {
            OEMol outmol = new OEMol(res.GetOverlayConfs());
            oechem.OEWriteMolecule(outfs, outmol);
            System.out.println(" title = "+outmol.GetTitle());
            System.out.println(" tanimoto combo = "+res.GetTanimotoCombo());
        }
    }
}
```

2.3 Overlay Optimization

The methods *OEOverlay* and *OEMultiRefOverlay*, compared to *OEROCS*, allow setting up a reference system and process overlay optimization of multiple fit molecules. Another difference is that both *OEOverlay* and *OEMultiRefOverlay*, in addition to providing the best overlay score, also give access to all the results obtained during an overlay optimization. Methods *OEOverlay* and *OEMultiRefOverlay*, which provide all results, should be used only when such a rigorous amount of resulting data is desired.

2.3.1 Best Optimization Result

This example reads in a reference molecule and a few fit molecules, performs overlay optimization, and returns only the best result per reference to fit molecule overlay optimization.

Listing 10: Getting the best scores from OEMultiRefOverlay.

```
/*
(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)

```java
// This example reads in a reference molecule and a few fit
// molecules, performs overlay optimization, sorts the results
// in tanimoto order, and shows the single best result.
// With the default options, OEOverlay optimizes both shape and color.

package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class OverlayBestRes {

    public static void main(String[] args) {
        if (args.length!=2) {
            System.out.println("OverlayBestRes <reffile> <fitfile>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);

        OEMol refmol = new OEMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();
        System.out.print("Ref. title: "+refmol.GetTitle()+" ");
        System.out.println("Num Confs: "+refmol.NumConfs());

        // Prepare reference molecule for calculation
        // With default options this will remove any explicit
        // hydrogens present and add color atoms
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.Prep(refmol);

        OEMultiRefOverlay overlay = new OEMultiRefOverlay();
        overlay.SetupRef(refmol);

        OEMol fitmol = new OEMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            prep.Prep(fitmol);
            OEBestOverlayScore score = new OEBestOverlayScore();
            overlay.BestOverlay(score, fitmol, new OEHighestTanimoto());

            System.out.print("Fit. title: "+fitmol.GetTitle()+" ");
            System.out.print("FitConfIdx: "+score.GetFitConfIdx()+" ");
            System.out.print("RefConfIdx: "+score.GetRefConfIdx()+" ");
            System.out.println("Tanimoto Combo: "+score.GetTanimotoCombo());
        }
        fitfs.close();
    }
}
```

2.3.2 All Optimization Results

This example repeats the above, but uses *Overlay* to obtain all the results.

A single Overlay calculation will return N results, where N is the number of ref conformers times the number of fit conformers times the number of starting positions for each pair. So comparing 2 molecules with 10 conformers each could return 400 or more results.

There are two helper classes designed solely to contain these results and to make it easy to extract all or just the desired subset. *OEBestOverlayResults* holds the results of a single pair of conformers. It contains a set of *OEBestOverlayScore* objects, one for each starting position.

This example uses 2 iterators to show all the results.

Listing 11: Getting all the scores from OEMultiRefOverlay.

```
/*
(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.
*/

// This example reads in a reference molecule and a few fit
// molecules, performs overlay optimization, and uses 2 iterators to
// show all the results.
// With the default options, OEOverlay optimizes both shape and color.

package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class OverlayAllRes {

    public static void main(String[] args) {
        if (args.length!=2) {
            System.out.println("OverlayAllRes <reffile> <fitfile>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);
```

(continues on next page)

(continued from previous page)

```
        OEMol refmol = new OEMol();
        oechem.OEReadMolecule(reffs, refmol);
        reffs.close();
        System.out.print("Ref. title: "+refmol.GetTitle()+" ");
        System.out.println("Num Confs: "+refmol.NumConfs());

        // Prepare reference molecule for calculation
        // With default options this will remove any explicit
        // hydrogens present and add color atoms
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.Prep(refmol);

        OEMultiRefOverlay overlay = new OEMultiRefOverlay();
        overlay.SetupRef(refmol);

        OEMol fitmol = new OEMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            System.out.print("Fit. title: "+fitmol.GetTitle()+" ");
            System.out.println("Num Confs: "+fitmol.NumConfs());

            prep.Prep(fitmol);
            int resCount=0;

            // double loop over results and scores to obtain all scores
            for (OEBestOverlayResults res : overlay.Overlay(fitmol)) {
                for (OEBestOverlayScore score : res.GetScores()) {
                    System.out.print("FitConfIdx: "+score.GetFitConfIdx()+" ");
                    System.out.print("RefConfIdx: "+score.GetRefConfIdx()+" ");
                    System.out.println("Tanimoto Combo: "+score.GetTanimotoCombo());
                    ++resCount;
                }
            }
            System.out.println(resCount+" results returned.");
        }
        fitfs.close();
    }
}
```

2.3.3 Manipulating all results

This example repeats the above to access all overlay results, and shows how to navigate through the results to extract a specific set. Here, the *OESortOverlayScores* function is used to turn the double iterator as shown in the example above into a single iterator of *OEBestOverlayScore*. Note that the third argument is a functor used to sort the list, such that in this next example, we get one *OEBestOverlayScore* for each pair of conformers, and they are returned in Tanimoto order.

OEOverlay or OEMultiRefOverlay does not actually move the fit molecule. Part of OEBestOverlayScore is the rotation matrix and translation matrix necessary to move the fit molecule into the final overlap position. (Note well: the rotation matrix is left-multiplied so can be regarded as left-handed.) This example also shows how to apply these transformations. The OpenEye standard is rotation then translation, but as a convenience, OEBestOverlayScore has *OEBestOverlayScore.Transform* method that will apply the transforms in the proper order.

This example keeps the user specified number of scores, aligns each fit conformer to the ref conformer it was overlaid on, and then write the pair to an output file. If SD or OEB is used as the output file type, then scores will also be stored in SD tags.

Listing 12: Writing few aligned structures from OEMultiRefOverlay.

```
/*
(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.
*/

// This example reads in a reference molecule and a few fit
// molecules, performs overlay optimization, sorts the results
// in tanimoto order, shows the user specified number of
// results, and saves the overlayed structures.
// With the default options, OEOverlay optimizes both shape and color.

package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class OverlayStruct {

    public static void main(String[] args) {
        if (args.length!=4) {
            System.out.println("OverlayStruct <reffile> <fitfile> <out.sdf> <keepsize>");
            System.exit(0);
        }

        oemolistream reffs = new oemolistream(args[0]);
        oemolistream fitfs = new oemolistream(args[1]);
        oemolostream outfs = new oemolostream(args[2]);
        int keepsize = Integer.parseInt(args[3]);

        OEMol refmol = new OEMol();
        oechem.OEReadMolecule(reffs, refmol);
        System.out.print("Ref. title: "+refmol.GetTitle()+" ");
        System.out.println("Num Confs: "+refmol.NumConfs());
```

(continues on next page)

(continued from previous page)

```
        reffs.close();

        // Prepare reference molecule for calculation
        // With default options this will remove any explicit
        // hydrogens present and add color atoms
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.Prep(refmol);

        OEMultiRefOverlay overlay = new OEMultiRefOverlay();
        overlay.SetupRef(refmol);

        OEMol fitmol = new OEMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            System.out.print("Fit. title: "+fitmol.GetTitle()+" ");
            System.out.println("Num Confs: "+fitmol.NumConfs());

            prep.Prep(fitmol);
            int resCount=0;

            // Sort all scores according to highest tanimoto
            OEBestOverlayScoreIter scoreiter = new OEBestOverlayScoreIter();
            oeshape.OESortOverlayScores(scoreiter, overlay.Overlay(fitmol), new␣
↪OEHighestTanimoto());

            for (OEBestOverlayScore score : scoreiter) {
                OEGraphMol outmol = new OEGraphMol(fitmol.GetConf(new OEHasConfIdx(score.
↪GetFitConfIdx())));
                score.Transform(outmol);

                oechem.OESetSDData(outmol, "RefConfIdx", String.valueOf(score.
↪GetRefConfIdx()));
                oechem.OESetSDData(outmol, "Tanimoto Combo", String.valueOf(score.
↪GetTanimotoCombo()));

                oeshape.OERemoveColorAtoms(refmol);
                oechem.OEWriteConstMolecule(outfs, refmol.GetConf(new OEHasConfIdx(score.
↪GetRefConfIdx())));
                oeshape.OERemoveColorAtoms(outmol);
                oechem.OEWriteConstMolecule(outfs, outmol);

                ++resCount;

                // Break at the user specified size
                if (resCount==keepsize)
                    break;
            }
            System.out.println(resCount+" results returned.");
        }
        fitfs.close();
        outfs.close();
    }
}
```

2.4 Working with Shape Query

This section of examples shows the use of a *OEShapeQuery*, and performs calculations where a shape query is used as the reference system instead of a molecule as the reference.

2.4.1 Creating a shape query

This examples shows how to create a simple shape query that combines a molecule with gaussians. In this example, the molecule is added in the query to define shape, and color is added separately in terms to gaussians. The created query is then saved in a file.

Listing 13: Creating a shape query.

```
/*
(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.
*/
package openeye.examples.oeshape;

import java.net.URL;
import openeye.oechem.*;
import openeye.oeshape.*;

public class CreateQuery {

    public static void main(String[] args) {
        OESimpleAppOptions opts = new OESimpleAppOptions("CreateQuery", OEFileStringType.
↪Mol3D, "sq");
        if (oechem.OEConfigureOpts(opts, args, false) == OEOptsConfigureStatus.Help)
            System.exit(0);

        oemolistream ifs = new oemolistream();
        if (!ifs.open(opts.GetInFile()))
            oechem.OEThrow.Fatal("Unable to open " + opts.GetInFile() + " for reading");

        oeofstream ofs = new oeofstream();
        if (!ofs.open(opts.GetOutFile()))
            oechem.OEThrow.Fatal("Unable to open " + opts.GetOutFile() + " for writing");
```

(continues on next page)

(continued from previous page)

```
        OEColorForceField cff = new OEColorForceField();
        cff.Init(OEColorFFType.ImplicitMillsDean);

        OEGraphMol mol = new OEGraphMol();
        while (oechem.OEReadMolecule(ifs, mol))
        {
            OEShapeQuery query = new OEShapeQuery();
            if (oeshape.OEMol2Query(query, mol, cff))
                oeshape.OEWriteShapeQuery(ofs, query);
            else
                oechem.OEThrow.Warning("Failed to create query from: " + mol.GetTitle());
        }

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

2.4.2 Overlap with shape query

This example reads in a reference shape query and a few fit molecules and prints out both the shape overlap and the color score calculated. By default the OEOverlapFunc uses the *Grid* shape and *Exact* color.

Listing 14: Overlap with a shape query

```
/*
(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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class OverlapQuery {

    public static void main(String[] args) {
        if (args.length!=2) {
```

(continues on next page)

(continued from previous page)

```
            System.out.println("OverlapQuery <queryfile> <fitfile>");
            System.exit(0);
        }

        String ext = oechem.OEGetFileExtension(args[0]);
        if (!ext.equals("sq"))
            oechem.OEThrow.Fatal("Requires a shape query .sq input file format");

        OEShapeQuery query = new OEShapeQuery();
        oeshape.OEReadShapeQuery(args[0], query);

        // Get appropriate function to calculate both shape and color
        // By default the OEOverlapFunc contains OEGridShapeFunc for shape
        // and OEExactColorFunc for color
        OEOverlapFunc func = new OEOverlapFunc();
        func.SetupRef(query);

        OEOverlapResults res = new OEOverlapResults();
        oemolistream fitfs = new oemolistream(args[1]);
        OEOverlapPrep prep = new OEOverlapPrep();
        OEGraphMol fitmol = new OEGraphMol();
        while (oechem.OEReadMolecule(fitfs, fitmol)) {
            prep.Prep(fitmol);
            func.Overlap(fitmol, res);
            System.out.print(fitmol.GetTitle());
            System.out.println(" tanimoto combo = "+res.GetTanimotoCombo());
            System.out.println(" shape tanimoto = "+res.GetTanimoto());
            System.out.println(" color tanimoto = "+res.GetColorTanimoto());
        }
    }
}
```

2.4.3 Top hit against shape query

This example reads in a reference query and a few fit molecules, performs overlay optimization, finds the few top hits, and prints them out in final orientation.

Listing 15: Finding top ROCS hits against a query

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
```

(continues on next page)

(continued from previous page)

```
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class TopHitsQuery {

    public static void main(String[] args) {
        if (args.length!=4) {
            System.out.println("TopHitsQuery <queryfile> <fitfile> <outfile> <nhits>");
            System.exit(0);
        }

        String ext = oechem.OEGetFileExtension(args[0]);
        if (!ext.equals("sq"))
            oechem.OEThrow.Fatal("Requires a shape query .sq input file format");

        oemolistream fitfs = new oemolistream(args[1]);
        oemolostream outfs = new oemolostream(args[2]);
        int nhits = Integer.parseInt(args[3]);

        OEShapeQuery query = new OEShapeQuery();
        oeshape.OEReadShapeQuery(args[0], query);

        // Setup OEROCS with specified number of best hits
        OEROCSOptions options = new OEROCSOptions();
        options.SetNumBestHits(nhits);
        OEROCS rocs = new OEROCS(options);
        rocs.SetDatabase(fitfs);

        for (OEROCSResult res : rocs.Overlay(query))
        {
            OEGraphMol outmol = new OEGraphMol(res.GetOverlayConf());
            oechem.OEWriteMolecule(outfs, outmol);
            System.out.println(" title = "+outmol.GetTitle());
            System.out.println(" tanimoto combo = "+res.GetTanimotoCombo());
        }
    }
}
```

2.4.4 Grid-To-Gaussians Fit

In this example, an input molecule is converted to a molecular grid. Next the resulting grid is fitted to shape Gaussians.

2.5 Shape from Hermite expansion

Using Hermite expansion we can obtain two functionalities: expand a molecule into Hermite representation and compare two molecules that are in the Hermite representation by computing their shape-Tanimoto coefficient. We illustrate both functionalities below with two examples.

2.5.1 Hermite expansion

In this example we input a molecule, set the level of Hermite expansion using NPolyMax variable, determine the optimal parameters for $\lambda_x, \lambda_y, \lambda_z$, and perform the computation of the Hermite expansion. At the end we output the resulting Hermite representation of the molecule as a grid, using CreateGrid method of the *OEHermite class*. The user can vary NPolyMax from low numbers, like 4 or 5, to higher numbers like 10–30 and observe convergence of the shape. Which value to use in practice depends on the size of the input molecule. Input molecule can be drug-like as well as a protein.

Listing 16: Expanding a molecular shape into Hermite polynomials

```
/*
(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.
*/

// This example reads in a molecule and performs Hermite expansion of
//that molecule to a given order in the NPolyMax parameter and outputs
//the resulting Hermite approximated shape to a grid.

package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;
import openeye.oegrid.*;
import java.net.URL;
import java.io.IOException;
import java.util.Vector;

public class HermiteExpansion {
    public static void main(String[] args) {
        URL fileURL = HermiteExpansion.class.getResource("HermiteExpansion.txt");
        OEInterface itf = null;
```

(continues on next page)

(continued from previous page)

```
        try {
            itf = new OEInterface(fileURL, "HermiteExpansion", args);
        } catch(IOException e) {
            oechem.OEThrow.Fatal("Unable to open itf file");
        }

        // Get the input parameters from the OEInterface

        int NPolyMax = itf.GetInt("-NPolyMax");
        float gridspacing = itf.GetFloat("-gridspacing");
        String ifname = itf.GetString("-inputfile");
        String ofname = itf.GetString("-outputgrid");

        oemolistream ifs = new oemolistream();

        if (!ifs.open(ifname)) {
            oechem.OEThrow.Fatal("Unable to open " + ifname + " for reading");
        }


        if (!ofname.endsWith(".grd")) {
            oechem.OEThrow.Fatal("Output grid file extension hast to be '.grd' ");
        }

        OEMol mol = new OEMol();

        if (!oechem.OEReadMolecule(ifs, mol)) {
            oechem.OEThrow.Fatal("Unable to read molecule in " + ifname);
        }

        OEOverlapPrep prep = new OEOverlapPrep();
        prep.SetAssignColor(false);
        prep.Prep(mol);


        OETrans transfm = new OETrans();
        oeshape.OEOrientByMomentsOfInertia(mol, transfm);

        OEHermiteOptions hermiteoptions = new OEHermiteOptions();
        hermiteoptions.SetNPolyMax(NPolyMax);
        hermiteoptions.SetUseOptimalLambdas(true);

        OEHermite hermite = new OEHermite(hermiteoptions);

        if (!hermite.Setup(mol)) {
            oechem.OEThrow.Fatal("Was not able to Setup the molecule for the OEHermite␣
↪class.");
        }

        OEHermiteOptions hopts = hermite.GetOptions();

        System.out.println("Best lambdas found X="
```

(continues on next page)

(continued from previous page)

```
                            + String.valueOf(hopts.GetLambdaX()) + "  Y="
                            + String.valueOf(hopts.GetLambdaY()) + "  Z="
                            + String.valueOf(hopts.GetLambdaZ()));

        System.out.println("Hermite self-overlap =" + String.valueOf(hermite.
↪GetSelfOverlap()));

        Vector<Double> coeffs = new  Vector<Double>();
        hermite.GetCoefficients(coeffs);

        String NPolyMaxstring = String.valueOf(NPolyMax);

        System.out.println("Hermite coefficients f_{l,m,n} in the following order l = 0..
↪."
              + NPolyMaxstring + ", m = 0..." + NPolyMaxstring+"-l, n = " +␣
↪NPolyMaxstring + "-l-m :");

        String coeff_str=coeffs.toString();

        System.out.println(coeff_str);

        OEScalarGrid grid = new OEScalarGrid();

        hermite.CreateGrid(grid, gridspacing, transfm);

        if (!oegrid.OEWriteGrid(ofname, grid)){
            oechem.OEThrow.Fatal("Unable to write grid file");
        }

    }


}
```

2.5.2 Hermite Tanimoto comparison of shape

In the second example below, we set up two molecules and compute the Tanimoto shape similarity coefficient, using Hermite expansion. The Hermite prep parameter NPolyMax is used to compute the shape-Tanimoto coefficient. First molecule in the reference input file is used versus all molecules in the fit input file. Results are attached as SD data to the moved fit molecule, as dictated by Hermite overlay, and sent to the output file. One can observe convergence of the Hermite Tanimoto coefficients as the level of expansion becomes more accurate.

Listing 17: Calculating shape Tanimoto using Hermite expansion

```
/*
(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
```

(continues on next page)

(continued from previous page)

```
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.
*/

package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;
import openeye.oegrid.*;
import java.net.URL;
import java.io.IOException;


public class HermiteTanimoto {
    public static void main(String[] args) {

        URL fileURL = HermiteTanimoto.class.getResource("HermiteTanimoto.txt");
        OEInterface itf = null;
        try {
            itf = new OEInterface(fileURL, "HermiteTanimoto", args);
        } catch(IOException e) {
            oechem.OEThrow.Fatal("Unable to open itf file");
        }

        int NPolyMax = itf.GetInt("-NPolyMax");
        String ifrefname = itf.GetString("-inputreffile");
        String iffitname = itf.GetString("-inputfitfile");
        String ofname = itf.GetString("-outputfile");

        oemolistream ifsref = new oemolistream();
        oemolistream ifsfit = new oemolistream();
        oemolostream ofs = new oemolostream();

        if (!ifsref.open(ifrefname)){
            oechem.OEThrow.Fatal("Unable to open " + ifrefname + " file for reading" );
        }

        if (!ifsfit.open(iffitname)){
            oechem.OEThrow.Fatal("Unable to open " + iffitname + " file for reading" );
        }

        OEMol refmol = new OEMol();
        OEMol fitmol = new OEMol();

        if (!oechem.OEReadMolecule(ifsref, refmol)){
```

(continues on next page)

(continued from previous page)

```
            oechem.OEThrow.Fatal("Unable to read molecule in " + ifrefname + " file");
        }

        OEOverlapPrep prep = new OEOverlapPrep();
        prep.SetAssignColor(false);
        prep.Prep(refmol);

        OEHermiteOptions hermiteoptions = new OEHermiteOptions();
        hermiteoptions.SetNPolyMax(NPolyMax);
        hermiteoptions.SetUseOptimalLambdas(true);

        OEHermiteOverlayOptions options = new OEHermiteOverlayOptions();
        options.SetHermiteOptions(hermiteoptions);
        OEHermiteOverlay overlay = new OEHermiteOverlay(options);
        overlay.SetupRef(refmol);

        if (!ofs.open(ofname)){
            oechem.OEThrow.Fatal("Unable to open " + ofname + " file for writing");
        }

        int idx = 0;
        while (oechem.OEReadMolecule(ifsfit, fitmol)){
            prep.Prep(fitmol);

            OEBestOverlayScore score = new OEBestOverlayScore();
            overlay.BestOverlay(score, fitmol);
            System.out.println("Hermite Tanimoto = " + String.valueOf(score.
↪GetTanimoto()));

            oechem.OESetSDData(fitmol, "HermiteTanimoto_" + String.valueOf(NPolyMax),␣
↪String.valueOf(score.GetTanimoto()));
            score.Transform(fitmol);
            oeshape.OERemoveColorAtoms(fitmol);
            oechem.OEWriteMolecule(ofs, fitmol);

            idx += 1;
        }
        ofs.flush();
        ofs.close();
    }
}
```

2.5.3 Hermite Grid Overlay

In this example, we perform grid-to-grid overlay optimization and compute the Tanimoto coefficient between them using Hermite expansion. Hermite overlays between grids and molecules can also be evaluated in a similar way.

2.6 Flexible Overlay with Shape, Color, and Forcefield

By incorporating Forcefield into the overlay optimization alongside Shape and Color, molecular flexibility can be introduced to the input fit molecule conformers. The flexible overlay optimization approach maintains rigidity in the reference molecule conformer while allowing for flexibility in the fit molecules. To illustrate the functionality of flexible overlay, we provide two examples. Note that, the default settings for the *OEFlexiOverlay* perform optimization on both *Shape* and *Color* overlap while also considering *intra-molecular forcefield*.

It is important to note that when running flexible overlay optimization of a molecule against itself, the user should not expect to obtain a tanimotocombo value of 2.0. This is because the fit molecules undergo flexible optimization, which includes intra-molecular force field, and thus may result in some deviation from 2.0.

2.6.1 Flexible Overlay

In this example, Flexible overlay optimization of the fit molecules is performed against a single reference molecule conformer. Results of flexible overlay of all of the fit molecule conformers are reported and saved.

2.6.2 Flexible Best Overlay

In this example, Flexible overlay optimization of the fit molecules is performed against a single reference molecule conformer. The best overlaid conformers and the corresponding results are reported and saved.

2.7 Advanced Examples

2.7.1 Calculating NxN scores

There are times when you want to have all the pairwise scores among a set of molecules. Maintaining this 2D matrix of scores is more complicated than the single set of scores from the *ref vs. set of fit molecules* examples shown above.

This example takes a set of molecules and performs all the pairwise shape optimizations with OEBestOverlay. It outputs a *spreadsheet* of the scores.

Listing 20: Calculating NxN scores.

```
/*
(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.
*/
package openeye.examples.oeshape;
```

(continues on next page)

(continued from previous page)

```
import openeye.oechem.*;
import openeye.oeshape.*;
import java.io.*;
import java.text.DecimalFormat;

public class NxNShape {
    static {
        oechem.OEThrow.SetStrict(true);
    }

    private static FileOutputStream csv;
    private static PrintWriter csvfile;
    private static DecimalFormat df = new DecimalFormat("#.##");

    private String roundOff(float score) {
        return df.format(score);
    }

    private void oneConf(OEConfBase conf, OEOverlapPrep prep, String filename) {
        csvfile.print(conf.GetTitle() + "_" + conf.GetIdx());
        OEMol refmol = new OEMol(conf);

        OEOverlayOptions options = new OEOverlayOptions();
        options.SetOverlapFunc(new OEGridShapeFunc());
        OEOverlay overlay = new OEOverlay();
        overlay.SetupRef(refmol);

        oemolistream bfs = new oemolistream(filename);
        OEMol fitmol = new OEMol();
        while (oechem.OEReadMolecule(bfs, fitmol)) {
            prep.Prep(fitmol);
            OEBestOverlayResultsIter resiter = overlay.Overlay(fitmol);
            OEBestOverlayScoreIter scoreiter = new OEBestOverlayScoreIter();
            oeshape.OESortOverlayScores(scoreiter, resiter, new OEHighestTanimoto(), 1,␣
↪true);
            for (OEBestOverlayScore score : scoreiter) {
                csvfile.print("," + roundOff(score.GetTanimoto()));
            }
        }
    }

    private void genHeader(String filename) {
        csvfile.print("name");
        oemolistream ifs = new oemolistream(filename);
        OEMol mol = new OEMol();
        while (oechem.OEReadMolecule(ifs, mol)) {
            System.out.println(mol.GetTitle());
            for (OEConfBase conf : mol.GetConfs()) {
                csvfile.print("," + conf.GetTitle() + "_" + conf.GetIdx());
            }
        }
        ifs.close();
```

(continues on next page)

(continued from previous page)

```
        csvfile.println();
    }

    private void openCSV(String csvfilename) {
        try {
            csv = new FileOutputStream(csvfilename);
            csvfile = new PrintWriter(csv);
        } catch (FileNotFoundException e) {
            System.err.println("Unable to open output file " + csvfilename);
        }
    }

    public void nxn(String filename, String csvfilename) {
        openCSV(csvfilename);
        genHeader(filename);
        OEOverlapPrep prep = new OEOverlapPrep();
        prep.SetAssignColor(false);
        oemolistream afs = new oemolistream(filename);
        OEMol molA = new OEMol();
        while (oechem.OEReadMolecule(afs, molA)) {
            prep.Prep(molA);
            for (OEConfBase conf : molA.GetConfs()) {
                oneConf(conf, prep, filename);
            }
        }
        csvfile.close();
        afs.close();
    }

    public static void main(String[] args) {
        if (args.length != 2) {
            System.out.println("NxNShape <infile> <csvfile>");
            System.exit(0);
        }
        NxNShape app = new NxNShape();
        app.nxn(args[0], args[1]);
    }
}
```

2.7.2 Calculating shape characteristics

While most functionality in the Shape Toolkit involves comparison of pairs of molecules, there are a few fundamental properties that can be calculated for a single molecule. All of these calculations are done using the same basic Gaussian description of a molecule as described above.

The simplest property to calculate is volume, using *OECalcVolume*.

In addition to simple volume calculations, the steric multipoles of a molecule can also be calculated. See section *OECalcShapeMultipoles*.

This next example demonstrates the calculation of volume and shape multipoles.

Listing 21: Calculating shape properties.

```
/*
(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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;

public class ShapeProps {

    public static void main(String[] args) {
        if (args.length!=1) {
            System.out.println("ShapeProps <infile>");
            System.exit(0);
        }

        oemolistream ifs = new oemolistream(args[0]);

        OEGraphMol mol = new OEGraphMol();

        while (oechem.OEReadMolecule(ifs,mol)) {
            System.out.println("              Title: "+mol.GetTitle());
            System.out.println("             Volume: "+oeshape.OECalcVolume(mol));
            System.out.println("Volume: (without H): "+
                    oeshape.OECalcVolume(mol, false));

            float [] smult = new float[14];
            oeshape.OECalcShapeMultipoles(mol, smult);

            System.out.println("  Steric multipoles:");
            System.out.println("           monopole: "+smult[0]);
            System.out.println("        quadrupoles: "+
                    smult[1]+" "+smult[2]+" "+smult[3]);
            System.out.println("          octopoles:");
            System.out.println("                xxx: "+smult[4]+
                    " yyy: "+smult[5]+" zzz: "+smult[6]);
```

(continues on next page)

(continued from previous page)

```
            System.out.println("                xxy: "+smult[7]+
                    " xxz: "+smult[8]+" yyx: "+smult[9]);
            System.out.println("                yyz: "+smult[10]+
                    " zzx: "+smult[11]+" zzy: "+smult[12]);
            System.out.println("                xyz: "+smult[13]);

            System.out.println("");
        }
        ifs.close();
    }
}
```

2.7.3 Calculating excluded volume

This code demonstrates how to use the Shape toolkit to do an overlay to a crystallographic ligand, then calculate the overlap of the database molecule to the cognate protein. The output includes a new score **Rescore** which is the **ShapeTanimotoCombo** penalized by the fraction overlap.

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

```
prompt> java ExcludeVolume --help
```

will generate the following output:

```
-q : Query file name
-e : Protein to use as exclusion volume
-d : Database file name
-o : Output file name
```

```
/*
(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.
*/
package openeye.examples.oeshape;

import openeye.oechem.*;
import openeye.oeshape.*;
import java.net.URL;
```

(continues on next page)

(continued from previous page)

```
import java.io.IOException;

public class ExcludedVolume {

    public static void main(String[] args) {
        URL fileURL = ExcludedVolume.class.getResource("ExcludedVolume.txt");
        OEInterface itf = null;
        try {
            itf = new OEInterface(fileURL, "ExcludedVolume", args);
        } catch(IOException e) {
            oechem.OEThrow.Fatal("Unable to open itf file");
        }

        // Set up best overlay to the query molecule
        oemolistream qfs = new oemolistream();
        if (!qfs.open(itf.GetString("-q"))) {
            oechem.OEThrow.Fatal("Unable to open " + itf.GetString("-q"));
        }

        OEMol qmol = new OEMol();
        oechem.OEReadMolecule(qfs, qmol);

        // Set up overlap to protein exclusion volume
        oemolistream efs = new oemolistream();
        if (!efs.open(itf.GetString("-e")))
            oechem.OEThrow.Fatal("Unable to open " + itf.GetString("-e"));

        OEMol emol = new OEMol();
        oechem.OEReadMolecule(efs, emol);

        OEExactShapeFunc evol = new OEExactShapeFunc();
        evol.SetupRef(emol);

        // open database and output streams
        oemolistream ifs = new oemolistream();
        if (!ifs.open(itf.GetString("-d"))) {
            oechem.OEThrow.Fatal("Unable to open " + itf.GetString("-d"));
        }

        oemolostream ofs = new oemolostream();
        if (!ofs.open(itf.GetString("-o"))) {
            oechem.OEThrow.Fatal("Unable to open " + itf.GetString("-o"));
        }

        System.out.println("Title                Combo  Rescore");
        OEMol mol = new OEMol();
        while (oechem.OEReadMolecule(ifs, mol)) {
            OEROCSResult res = new OEROCSResult();
            oeshape.OEROCSOverlay(res, qmol, mol);
            OEGraphMol outmol = new OEGraphMol(res.GetOverlayConf());

            // calculate overlap with protein
```

(continues on next page)

(continued from previous page)

```
            OEOverlapResults eres = new OEOverlapResults();
            evol.Overlap(outmol, res);

            float frac = eres.GetOverlap() / eres.GetFitSelfOverlap();
            float rescore = res.GetTanimotoCombo() - frac;

            // attach data to molecule and write it
            oechem.OESetSDData(outmol, "TanimotoCombo", String.format("%5.3f", res.
↪GetTanimotoCombo()));
            oechem.OESetSDData(outmol, "Exclusion Volume", String.format("%5.3f",  eres.
↪GetOverlap()));
            oechem.OESetSDData(outmol, "Fraction Overlap", String.format("%5.3f", frac));
            oechem.OESetSDData(outmol, "Rescore", String.format("%5.3f", rescore));

            oechem.OEWriteMolecule(ofs, outmol);
            String line = String.format("%-20s %5.3f  %5.3f", outmol.GetTitle(), res.
↪GetTanimotoCombo(), rescore);
            System.out.println(line);
        }
        ifs.close();
    }
}
```

with interface file

```
!CATEGORY %(prog)s
  !BRIEF %(prog)s [-q] <query> [-e] <exclusion> [-d] <database> [-o] <output>
  !PARAMETER -q 1
    !TYPE string
    !REQUIRED true
    !BRIEF Query file name
    !KEYLESS 1
  !END
  !PARAMETER -e 2
    !TYPE string
    !REQUIRED true
    !BRIEF Protein to use as exclusion volume
    !KEYLESS 2
  !END
  !PARAMETER -d 3
    !TYPE string
    !REQUIRED true
    !BRIEF Database file name
    !KEYLESS 3
  !END
  !PARAMETER -o 4
    !TYPE string
    !REQUIRED true
    !BRIEF Output file name
    !KEYLESS 4
  !END
!END
```

CHAPTER

THREE

API

3.1 OEShape Classes

3.1.1 OEAnalyticColorFunc

```
class OEAnalyticColorFunc : public OEShapeFunc
```

The *OEAnalyticColorFunc* class defines color overlap between a reference object and a fit molecule, using the *Analytic* method.

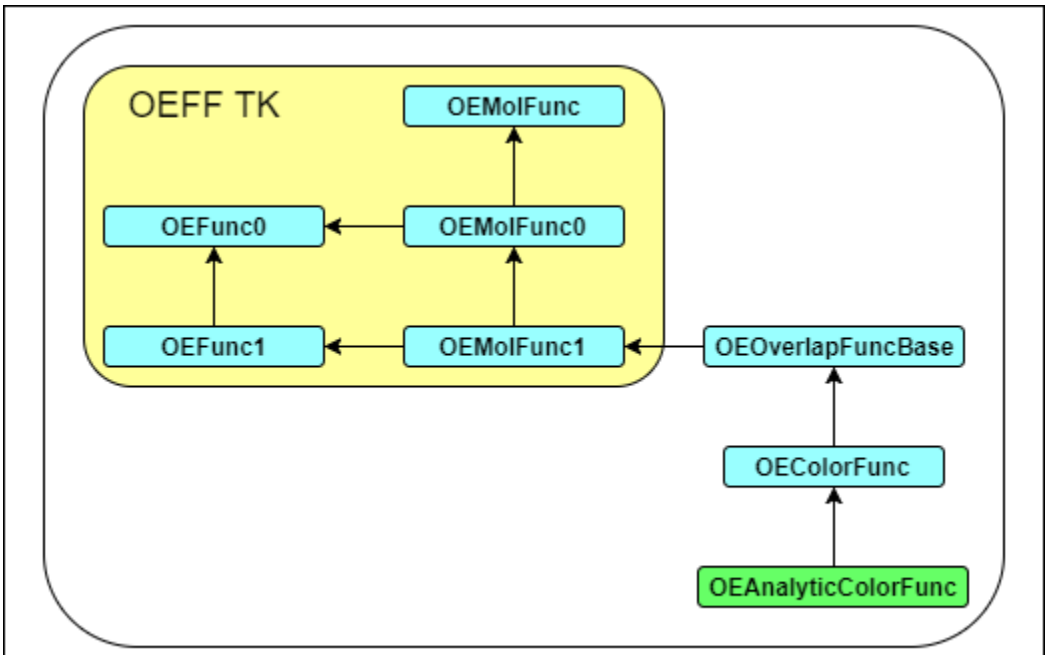


Figure: OEAnalyticColorFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- `operator()`
- `GetFComponents`
- `NumVar`

The following methods are publicly inherited from OEFunc1:

- `operator()`

The following methods are publicly inherited from OEMolFunc:

- GetInteractions
- Setup

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*
- *SetupRef*

The following methods are publicly inherited from *OEColorFunc*:

- *GetColorOptions*

The *OEAnalyticColorFunc* class defines the following public methods:

- *GetAnalyticOptions*

Constructors

```
OEAnalyticColorFunc(const OEColorOptions& argOptions = OEColorOptions())
OEAnalyticColorFunc(const OEAnalyticOptions&)
OEAnalyticColorFunc(const OEColorOptions& argColor,
                    const OEAnalyticOptions& argAnalytic)
OEAnalyticColorFunc(const OEAnalyticColorFunc&)
```

Default and copy constructors.

Constructs an *OEAnalyticColorFunc* instance using the specified set of parameters.

GetAnalyticOptions

```
const OEAnalyticOptions& GetAnalyticOptions() const
```

Returns a reference to the *OEAnalyticOptions* instance as currently set for this *OEAnalyticColorFunc*.

3.1.2 OEAnalyticOptions

```
class OEAnalyticOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for calculations with the *Analytic* method.

The *OEAnalyticOptions* class defines the following public methods:

- *GetExpType*
- *GetProxyGridCutoff*
- *GetUseProxyGrid*
- *SetExpType*
- *SetProxyGridCutoff*
- *SetUseProxyGrid*

Constructors

```
OEAnalyticOptions()
OEAnalyticOptions(const OEAnalyticOptions&)
```

Default and copy constructors.

GetExpType

```
unsigned int GetExpType() const
```

Gets the current value of the exponential type used by the function.

GetProxyGridCutoff

```
double GetProxyGridCutoff() const
```

Gets the proxy grid cutoff currently set.

GetUseProxyGrid

```
bool GetUseProxyGrid() const
```

Returns the state of the use proxy grid flag. If true, proxy grid would be used to decide if an atom pair should be included in calculations.

SetExpType

```
bool SetExpType(const unsigned int)
```

Set the exponential type to be used for calculation. The default is *OEExponentialType.Standard*. Alternatives are defined in the *OEExponentialType* namespace.

SetProxyGridCutoff

```
bool SetProxyGridCutoff(const double)
```

Set the radius to use when using proxy grid is turned on. By default this is set to 3.0 Å.

SetUseProxyGrid

```
bool SetUseProxyGrid(const bool)
```

Sets the state of the use proxy grid flag. If true, proxy grid would be used to decide if an atom pair should be included in calculations.

3.1.3 OEAnalyticShapeFunc

```
class OEAnalyticShapeFunc : public OEShapeFunc
```

The *OEAnalyticShapeFunc* class defines shape overlap between a reference object and a fit molecule, using the *Analytic* method.

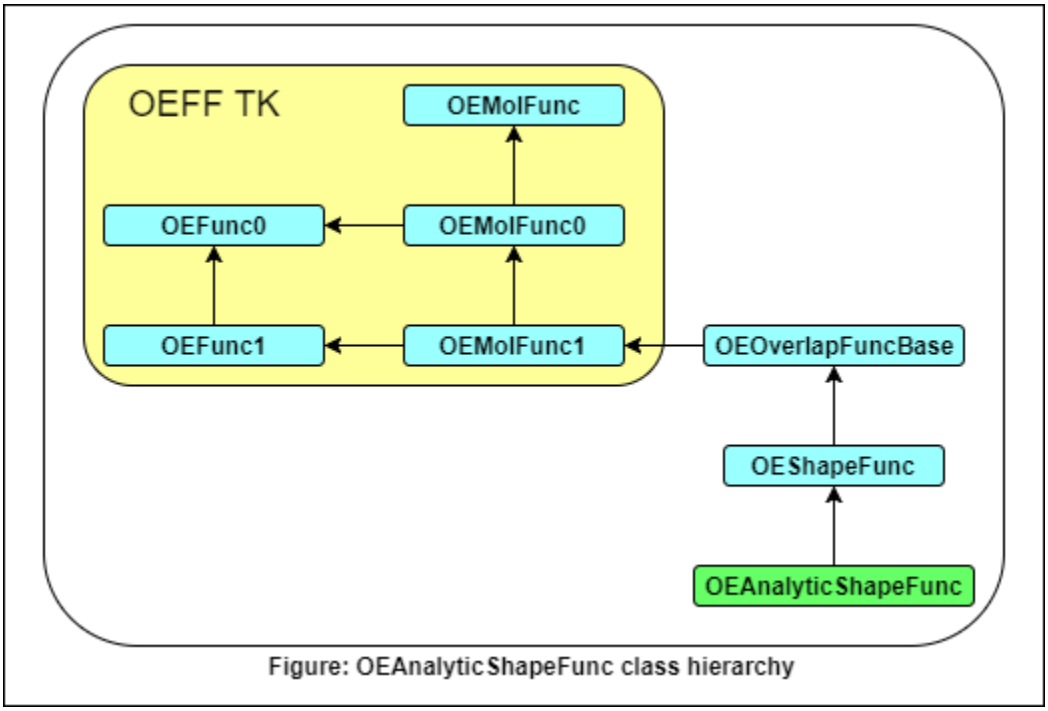


Figure: OEAnalyticShapeFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- operator()
- GetFComponents
- NumVar

The following methods are publicly inherited from OEFunc1:

- operator()

The following methods are publicly inherited from OEMolFunc:

- GetInteractions
- Setup

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*
- *SetupRef*

The following methods are publicly inherited from *OEShapeFunc*:

- *GetShapeOptions*

The *OEAnalyticShapeFunc* class defines the following public methods:

- *GetAnalyticOptions*

Constructors

```
OEAnalyticShapeFunc(const OEShapeOptions& argOptions = OEShapeOptions())
OEAnalyticShapeFunc(const OEAnalyticOptions&)
OEAnalyticShapeFunc(const OEShapeOptions& argShape,
                    const OEAnalyticOptions& argAnalytic)
OEAnalyticShapeFunc(const OEAnalyticShapeFunc&)
```

Default and copy constructors.

Constructs an *OEAnalyticShapeFunc* instance using the specified set of parameters.

GetAnalyticOptions

```
const OEAnalyticOptions& GetAnalyticOptions() const
```

Returns a reference to the *OEAnalyticOptions* instance as currently set for this *OEAnalyticShapeFunc*.

3.1.4 OEAsIsStarts

```
class OEAsIsStarts : public OEStarts
```

The *OEAsIsStarts* represents the AsIs start for overlay optimization. The *OEAsIsStarts* takes the fit conformer as-is, without any rotation or translation.

The following methods are publicly inherited from *OEStarts*:

- *CreateCopy*
- *GetNumOfStarts*
- *GetStarts*
- *Setup*
- *SetupRef*
- *SetupRef*

Constructors

```
OEAsIsStarts()
OEAsIsStarts(const OEAsIsStarts&)
```

Default and copy constructors.

3.1.5 OEAtAtomStarts

```
class OEAtAtomStarts : public OEInertialStarts
```

The *OEAtAtomStarts* represents the AtAtom starts for overlay optimization. The *OEAtAtomStarts* translates the fit molecule to the specified atom center of the reference molecule, followed by the inertial rotations to generate starts.

The following methods are publicly inherited from *OEStarts*:

- *CreateCopy*
- *GetNumOfStarts*
- *GetStarts*

- *Setup*
- *SetupRef*
- *SetupRef*

The following methods are publicly inherited from *OEInertialStarts*:

- *GetSymmetryThreshold*
- *SetSymmetryThreshold*

Constructors

```
OEAtAtomStarts(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& argPredicate =␣
↪OEChem::OEIsHeavy())
OEAtAtomStarts(const OEAtAtomStarts&)
```

Default and copy constructors. Use the second constructor to specify the at center to be picked for AtAtom starts. By default the heavy atoms of reference molecule are picked.

3.1.6 OEBestOverlayResults

```
class OEBestOverlayResults
```

Constructors

```
OEBestOverlayResults()
OEBestOverlayResults(const OEBestOverlayResults& rhs)
```

Default and copy constructors.

AddScore

```
bool AddScore(const OEBestOverlayScore& sc)
```

Clear

```
bool Clear()
```

GetScores

```
OESystem::OEIterBase<OEBestOverlayScore>* GetScores()
OESystem::OEIterBase<OEBestOverlayScore>* GetScores(const OESystem::OEBinaryPredicate
↪<OEBestOverlayScore, OEBestOverlayScore>& sorter,
                                                    int nbest = 1)
```

3.1.7 OEBestOverlayScore

```
class OEBestOverlayScore : public OEOverlapResults
```

This class extends the *OEOverlapResults* to represent the results of a single overlay optimization between a reference and a fit object.

The following methods are publicly inherited from *OEOverlapResults*:

- *GetColorScore*
- *GetColorTanimoto*
- *GetColorTversky*
- *GetFitColorTversky*
- *GetFitSelfColor*
- *GetFitSelfOverlap*
- *GetFitTversky*
- *GetFitTverskyCombo*
- *GetOverlap*
- *GetRefColorTversky*
- *GetRefSelfColor*
- *GetRefSelfOverlap*
- *GetRefTversky*
- *GetRefTverskyCombo*
- *GetTanimoto*
- *GetTanimotoCombo*
- *GetTversky*

The *OEBestOverlayScore* class defines the following public methods:

- *GetFitConfIdx*
- *GetRefConfIdx*
- *GetRotMatrix*
- *GetTranslation*
- *Transform*

Constructors

```
OEBestOverlayScore()
OEBestOverlayScore(const OEBestOverlayScore&)
```

Default and copy constructors.

GetFitConfIdx

```
unsigned int GetFitConfIdx() const
```

Returns the conformer index of the fit conformer.

GetRefConfIdx

```
unsigned int GetRefConfIdx() const
```

Returns the conformer index of the reference conformer.

GetRotMatrix

```
void GetRotMatrix(float*) const
```

Returns the rotation matrix to be applied on the fit conformer to obtain the optimized overlay. Using the OpenEye convention, rotation should be applied before the translation.

GetTranslation

```
void GetTranslation(float*) const
```

Returns the translation vector to be applied on the fit conformer to obtain the optimized overlay. Using the OpenEye convention, rotation should be applied before the translation.

Transform

```
bool Transform(OEChem::OEMolBase& mol) const
bool Transform(OEChem::OEConfBase& conf) const
bool Transform(OESystem::OEScalarGrid& grid) const
```

Transforms the molecule to the optimized overlay position.

3.1.8 OECartesianStarts

```
class OECartesianStarts : public OEInertialStarts
```

The *OECartesianStarts* represents the user defined cartesian starts for overlay optimization. The *OECartesianStarts* translates the fit molecule to the specified starting points, followed by the inertial rotations to generate starts.

The following methods are publicly inherited from *OEStarts*:

- *CreateCopy*
- *GetNumOfStarts*
- *GetStarts*
- *Setup*
- *SetupRef*
- *SetupRef*

The following methods are publicly inherited from *OEInertialStarts*:

- *GetSymmetryThreshold*
- *SetSymmetryThreshold*

The *OECartesianStarts* class defines the following public methods:

- *ClearStartPoints*
- *GetNumStartPoints*
- *GetStartPoints*
- *SetStartPoints*

Constructors

```
OECartesianStarts()
OECartesianStarts(const OECartesianStarts&)
```

Default and copy constructors.

ClearStartPoints

```
void ClearStartPoints()
```

Clears the currently specified cartesian start points.

GetNumStartPoints

```
unsigned int GetNumStartPoints() const
```

Get the number of specified cartesian start points.

GetStartPoints

```
bool GetStartPoints(double* argPoints) const
```

Get the specified cartesian start points. Method expects an array of size 3*NumPoints for the argument, where `NumPoints` can be obtained by calling *GetNumStartPoints*.

SetStartPoints

```
void SetStartPoints(const double* Points, unsigned int NumPoints)
```

Sets the cartesian starting points. Method expects an array of size 3*NumPoints for the first argument.

3.1.9 OEColorFFParameter

```
class OEColorFFParameter : public OESystem::OEMultiParameter<OEShape::OEColorForceField>
```

The *OEColorFFParameter* represents parameter that has a value of type *OEColorForceField*.

Following methods are publicly inherited from OEParameter:

- `AddAlias` and `GetAliases`
- `AddDetail` and `GetDetail`
- `AddIllegalRange` and `GetIllegalRanges`
- `AddIllegalValue` and `GetIllegalValues`
- `AddLegalRange` and `GetLegalRanges`
- `AddStringDefault` , `GetStringDefault` and `GetStringDefaults`
- `AddStringValue` , `GetStringValue` and `GetStringValues`
- `ClearDefaults`
- `ClearValues`
- `CreateCopy`

- GetBrief and SetBrief
- GetHasDefault
- GetHasValue
- GetIsList and SetIsList
- GetKeyless and SetKeyless
- GetName and SetName
- GetOrderPriority and SetOrderPriority
- GetVisibility and SetVisibility
- IsLegalString
- IsSet and IsSetToString

Following methods are publicly inherited from OEMultiParameter:

- AddLegalEntry
- GetDefault and SetDefault
- GetSetting
- GetValue and SetValue

Constructors

```
OEColorFFParameter(const std::string& name, const bool load = true)
OEColorFFParameter()
OEColorFFParameter(const OEColorFFParameter&)
```

Default and copy constructors.

Constructs an *OEColorFFParameter* instance using the specified set of parameters.

3.1.10 OEColorForceField

```
class OEColorForceField
```

Defines chemical feature types and their interaction rules for color overlap calculations.

Constructors

```
OEColorForceField()
OEColorForceField(const OEColorForceField& rhs)
OEColorForceField(const OEColorForceFieldImpl* impl)
```

Default and copy constructors.

AddColorer

```
bool AddColorer(unsigned int type, const char* smarts)
bool AddColorer(unsigned int type, const std::string& smarts)
```

AddInteraction

```
bool AddInteraction(unsigned int type1, unsigned int type2,
                    const std::string& interaction_type,
                    float weight, float range)
```

AddType

```
unsigned int AddType(const char* name)
unsigned int AddType(const std::string& name)
```

Clear

```
void Clear()
```

ClearInteractions

```
void ClearInteractions()
```

GetTitle

```
const char* GetTitle() const
```

GetType

```
unsigned int GetType(std::string k) const
```

GetTypeName

```
std::string GetTypeName(unsigned int type) const
```

GetTypes

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

Init

```
bool Init(unsigned int cffType)
bool Init(OEPlatform::oeistream& is, bool verbose = false)
bool Init(const std::string& filename, bool verbose = false)
```

Ready

```
bool Ready() const
```

SetTitle

```
bool SetTitle(const char* title)
```

Write

```
bool Write(OEPlatform::oeostream& os)
```

3.1.11 OEColorFunc

```
class OEColorFunc : public OEOverlapFuncBase
```

The *OEColorFunc* is an abstract base class. This class defines an interface for color overlap calculation between a reference object and a fit molecule. The *OEColorFunc* is a specialization of The *OEOverlapFuncBase* for color only overlap.

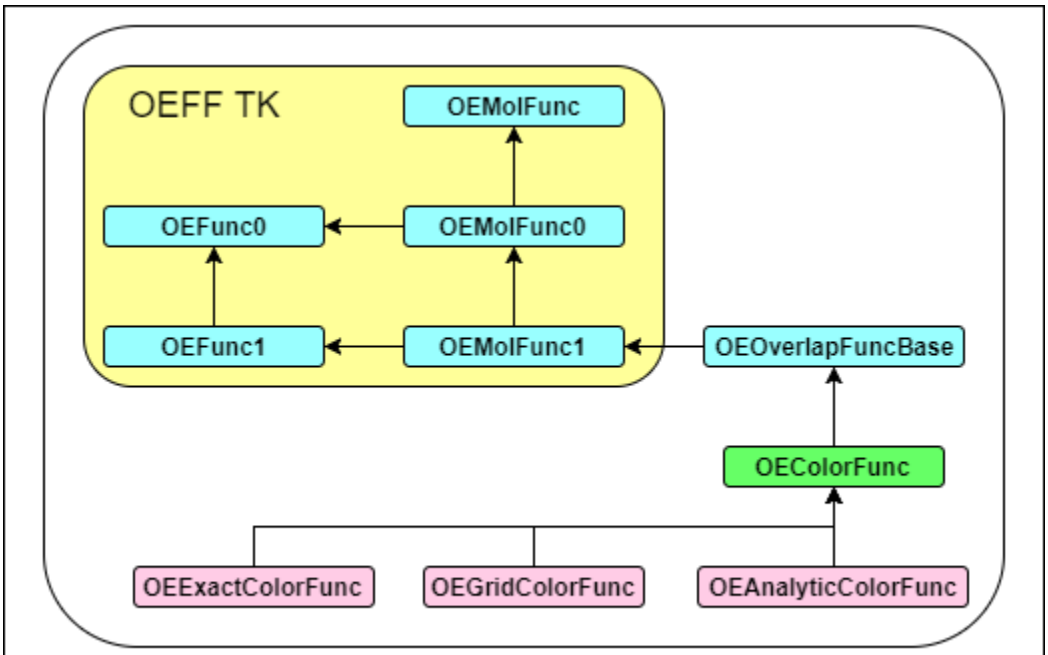


Figure: OEColorFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- operator()
- GetFComponents
- NumVar

The following methods are publicly inherited from OEFunc1:

- operator()

The following methods are publicly inherited from OEMolFunc:

- GetInteractions
- Setup

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*

- *SetupRef*

The *OEColorFunc* class defines the following public methods:

- *GetColorOptions*

The following classes derive from this class:

- *OEAnalyticColorFunc*
- *OEExactColorFunc*
- *OEGridColorFunc*

GetColorOptions

```
const OEColorOptions& GetColorOptions() const
```

Returns a reference to the *OEColorOptions* instance as currently set for this *OEColorFunc*.

3.1.12 OEColorOptions

```
class OEColorOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for *OEColorFunc* calculations.

The *OEColorOptions* class defines the following public methods:

- *GetColorForceField* and *SetColorForceField*
- *GetMultiplier* and *SetMultiplier*
- *GetScoreType* and *SetScoreType*

Constructors

```
OEColorOptions(const bool loadColorFF = true)
OEColorOptions(const OEColorOptions&)
```

Default and copy constructors.

GetColorForceField

```
const OEColorForceField* GetColorForceField() const
```

See *SetColorForceField* method.

GetMultiplier

```
double GetMultiplier() const
```

See *SetMultiplier* method.

GetScoreType

```
unsigned int GetScoreType() const
```

See *SetScoreType* method.

SetColorForceField

```
bool SetColorForceField(unsigned int type)
bool SetColorForceField(OEPlatform::oeistream& is)
bool SetColorForceField(const std::string& filename)
bool SetColorForceField(const OEColorForceField& cff)
```

Sets the color force field to be used. By default the ImplicitMillsDean color force field is used.

SetMultiplier

```
bool SetMultiplier(const double)
```

Sets the **multiplier** to the color score. THe multiplier only effects the *color score* and has no effect on the *color tanimoto*. **Default: 1.0**

SetScoreType

```
bool SetScoreType(const unsigned)
```

Sets the **score type** to be reported by this function. Possible values are defined in the *OEScoreType* namespace. **Default:** *OEScoreType.Overlap*

3.1.13 OEColorTanimotoCutoff

```
class OEColorTanimotoCutoff : public OEOverlayScoreCutoff
```

The *OEColorTanimotoCutoff* defines cutoff based on the color tanimoto.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OEColorTanimotoCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.14 OEExactColorFunc

```
class OEExactColorFunc : public OEShapeFunc
```

The *OEExactColorFunc* class defines shape overlap between a reference object and a fit molecule, using the *Exact* method.

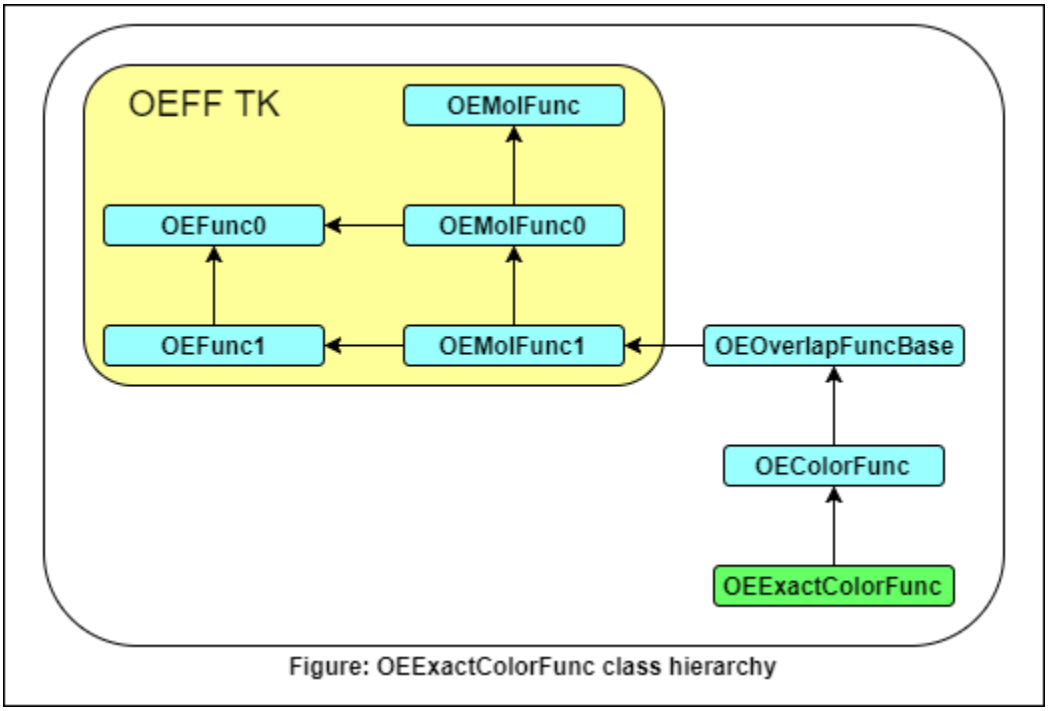


Figure: OEExactColorFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- `operator()`
- `GetFComponents`
- `NumVar`

The following methods are publicly inherited from OEFunc1:

- `operator()`

The following methods are publicly inherited from OEMolFunc:

- `GetInteractions`
- `Setup`

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*
- *SetupRef*

The following methods are publicly inherited from *OEColorFunc*:

- *GetColorOptions*

Constructors

```
OEExactColorFunc(const OEColorOptions& argOpts = OEColorOptions())
OEExactColorFunc(const OEExactColorFunc&)
```

Default and copy constructors.

Constructs an *OEExactColorFunc* instance using the specified set of parameters.

3.1.15 OEExactShapeFunc

```
class OEExactShapeFunc : public OEShapeFunc
```

The *OEExactShapeFunc* class defines shape overlap between a reference object and a fit molecule, using the *Exact* method.

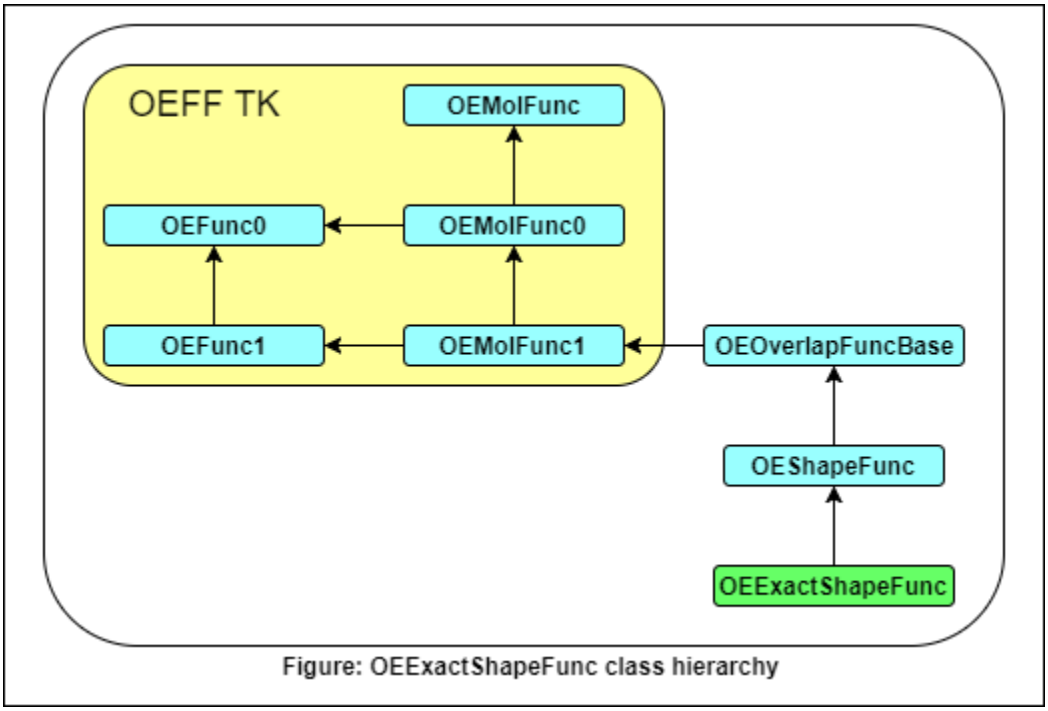


Figure: OEExactShapeFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- `operator()`
- `GetFComponents`
- `NumVar`

The following methods are publicly inherited from OEFunc1:

- `operator()`

The following methods are publicly inherited from OEMolFunc:

- `GetInteractions`
- `Setup`

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*
- *SetupRef*

The following methods are publicly inherited from *OEShapeFunc*:

- *GetShapeOptions*

Constructors

```
OEExactShapeFunc(const OEShapeOptions& argOptions = OEShapeOptions())
OEExactShapeFunc(const OEExactShapeFunc&)
```

Default and copy constructors.

Constructs an *OEExactShapeFunc* instance using the specified set of parameters.

3.1.16 OEFitColorTverskyCutoff

```
class OEFitColorTverskyCutoff : public OEOverlayScoreCutoff
```

The *OEFitColorTverskyCutoff* defines cutoff based on the fit molecule color tversky.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OEFitColorTverskyCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.17 OEFitTverskyComboCutoff

```
class OEFitTverskyComboCutoff : public OEOverlayScoreCutoff
```

The *OEFitTverskyComboCutoff* defines cutoff based on the fit molecule combined shape and color tversky.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OEFitTverskyComboCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.18 OEFitTverskyCutoff

```
class OEFitTverskyCutoff : public OEOverlayScoreCutoff
```

The *OEFitTverskyCutoff* defines cutoff based on the fit molecule shape tversky.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OEFitTverskyCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.19 OEFlexiOverlapFunc

```
class OEFlexiOverlapFunc : public OEMolPotential::OEMolFunc1
```

The *OEFlexiOverlapFunc* class defines an interface for static overlap calculation between a reference and a fit molecule. The *OEFlexiOverlapFunc* extends the OEMolFunc1 class to allow setting up a reference molecule and the underlying functions defining interactions based on shape and color overlaps along with intra-molecular force field of the fit molecule.

The following methods are publicly inherited from OEFunc0:

- operator()
- GetFComponents
- NumVar

The following methods are publicly inherited from OEFunc1:

- operator()

The following methods are publicly inherited from OEMolFunc:

- GetInteractions
- Setup

The *OEFlexiOverlapFunc* class defines the following public methods:

- *Overlap*
- *SetupRef*
- *UpdateColorCoords*

3.1.20 Overlap

```
bool Overlap(const double*, OEFlexiOverlapResults&)
bool Overlap(const OEChem::OEMolBase& fitMol,
             OEFlexiOverlapResults& results)
```

These methods calculate the staitc overlap between the reference and the fit molecule using shape, color, and force field. The reference object must be set using the *SetupRef* method. The first overload assumes that the fit molecule has been set using the Setup method and uses the specified coordinates for the fit molecule. The second overlap takes the specified fit molecule and its current coordinates.

3.1.21 SetupRef

```
bool SetupRef(const OEChem::OEMolBase&)
bool SetupRef(const OEShape::OEShapeQueryBase&)
```

This method defines the interface for setting up the reference system for the *OEFlexiOverlapFunc* instance with a molecule or a shape query.

3.1.22 UpdateColorCoords

```
bool UpdateColorCoords(const double* coords,
                       std::vector<double>& updatedCoord)
```

This method takes the coordinate of fit molecule and updates the coordinate of color atoms based on the modified coordinate of their parent atoms. The fit molecule must be set using the `Setup` method which assigns color atoms to the passed moleucle.

3.1.23 OEFlexiOverlapOptions

```
class OEFlexiOverlapOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for *OEFlexiOverlapFunc*.

The *OEFlexiOverlapOptions* class defines the following public methods:

- *Validate*
- *GetShapeFuncType* and *SetShapeFuncType*
- *GetColorFuncType* and *SetColorFuncType*
- *GetForceField* and *SetForceField*
- *GetHarmonic* and *SetHarmonic*
- *GetShapeOptions* and *SetShapeOptions*
- *GetColorOptions* and *SetColorOptions*
- *GetOverlapFunc* and *SetOverlapFunc*

Constructor

```
OEFlexiOverlapOptions()
OEFlexiOverlapOptions(const OEFlexiOverlapOptions&)
```

Default and copy constructors.

Assignment operator.

3.1.24 Validate

```
bool Validate() const
```

Returns **True** if the provided options corresponding to *OEFlexiOverlapOptions* are valid.

3.1.25 GetShapeFuncType

```
unsigned GetShapeFuncType() const
```

Gets the current type of *OEShapeFunc* used for shape overlap calculations. See also *SetShapeFuncType* method.

3.1.26 GetColorFuncType

```
unsigned GetColorFuncType() const
```

Gets the current type of *OEColorFunc* used for color overlap calculations. See also *SetColorFuncType* method.

3.1.27 GetForceField

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

Gets the current type of ligand force field used. See also *SetForceField* method.

3.1.28 GetHarmonic

```
bool GetHarmonic() const
```

Returns True if the harmonic constraints are added to the force field for the flexible overlap optimizations.

3.1.29 GetShapeOptions

```
OEShapeOptions& GetShapeOptions()
const OEShapeOptions& GetShapeOptions() const
```

Returns a reference to the *OEShapeOptions* instance as currently set. These options are used in flexible overlap optimization of conformers. See also *SetShapeOptions* method.

3.1.30 GetColorOptions

```
OEColorOptions& GetColorOptions()
const OEColorOptions& GetColorOptions() const
```

Returns a reference to the *OEColorOptions* instance as currently set. These options are used in flexible overlap optimization of conformers. See also *SetColorOptions* method.

3.1.31 GetOverlapFunc

```
const OEOverlapFuncBase& GetOverlapFunc() const
```

Returns a reference to the *OEOverlapFuncBase* instance as currently set. This defines the shape/color interactions to be optimized for overlay.

3.1.32 SetShapeFuncType

```
bool SetShapeFuncType(const unsigned)
```

Sets the type of *OEShape::OEShapeFunc* to be used for shape overlap calculations. The default is *OEShapeType.Grid*. Alternatives are defined in the *OEShapeType* namespace.

3.1.33 SetColorFuncType

```
bool SetColorFuncType(const unsigned)
```

Sets the type of *OEShape::OEColorFunc* to be used for color overlap calculations. The default is *OEColorType.Exact*. Alternatives are defined in the *OEColorType* namespace.

3.1.34 SetForceField

```
bool SetForceField(const unsigned)
```

Sets the type of ligand intra-molecular force field to be used for flxible optimizations. The default is `OELigandFFType.SAGE`. Alternatives are defined in the `OELigandFFType` namespace.

3.1.35 SetHarmonic

```
bool SetHarmonic(const bool)
```

This flag defines if the harmonic constraints should be added to the force field for the flexible overlap optimizations. Default *False*.

3.1.36 SetShapeOptions

```
void SetShapeOptions(const OEShapeOptions&)
```

Sets the options related to *OEShapeFunc* calculations used in flexible overlap optimization by passing in *OEShapeOptions* instance.

3.1.37 SetColorOptions

```
void SetColorOptions(const OEColorOptions&)
```

Sets the options related to *OEColorFunc* calculations used in flexible overlap optimization by passing in *OEColorOptions* instance.

3.1.38 SetOverlapFunc

```
bool SetOverlapFunc(const OEOverlapFuncBase&)
```

Sets the overlap function by passing in an *OEOverlapFuncBase* instance. This defines the shape/color interactions to be optimized for overlay.

3.1.39 OEFlexiOverlapResults

```
class OEFlexiOverlapResults : public OEShape::OEOverlapResults
```

This class extends the *OEOverlapResults* to represent the results of *OEFlexiOverlapFunc* calculation.

See also

- *OEOverlapResults* class

The following methods are publicly inherited from *OEOverlapResults*:

- *GetColorScore*
- *GetColorTanimoto*
- *GetColorTversky*
- *GetFitColorTversky*
- *GetFitSelfColor*
- *GetFitSelfOverlap*
- *GetFitTversky*
- *GetFitTverskyCombo*
- *GetOverlap*
- *GetRefColorTversky*
- *GetRefSelfColor*
- *GetRefSelfOverlap*
- *GetRefTversky*
- *GetRefTverskyCombo*
- *GetTanimoto*
- *GetTanimotoCombo*
- *GetTversky*

The *OEFlexiOverlapResults* class defines the following public method:

- *GetInternalEnergy*

Constructor

```
OEFlexiOverlapResults()
OEFlexiOverlapResults(const OEFlexiOverlapResults&)
```

Default and copy constructors.

GetInternalEnergy

```
double GetInternalEnergy() const
```

Returns the computed **internal energy** of the conformer.

3.1.40 OEFlexiOverlay

```
class OEFlexiOverlay
```

The *OEFlexiOverlay* class provides an interface for flexible overlay optimization between the reference molecule and a fit molecule conformers.

The *OEFlexiOverlay* class defines the following public methods:

- *SetupRef*
- *Overlay*
- *SortedOverlay*
- *BestOverlay*

Constructor

```
OEFlexiOverlay(const OEFlexiOverlayOptions& options = OEFlexiOverlayOptions())
OEFlexiOverlay(const OEFlexiOverlay& func)
```

Default and copy constructors.

Assignment operator.

3.1.41 SetupRef

```
bool SetupRef(const OEChem::OEMolBase&)
bool SetupRef(const OEShape::OEShapeQueryBase&)
```

This method defines the interface for setting up the reference system for the *OEFlexiOverlay* using a molecule or a shape query for flexible overlay optimization process.

3.1.42 Overlay

```
OESystem::OEIterBase<OEFlexiOverlapResults>* Overlay(OEChem::OEMCMolBase&)
```

This method optimizes the shape and chemical similarity between the fit molecule conformers and a reference molecule, along with intra-molecular forcefield energies of the fit molecule. The reference molecule must be set using *SetupRef* method.

3.1.43 SortedOverlay

```
OESystem::OEIterBase<OEFlexiOverlapResults>* SortedOverlay(OEChem::OEMCMolBase&
↪)
```

This method optimizes the shape and chemical similarity between the fit molecule conformers and a reference molecule, along with intra-molecular forcefield energies of the fit molecule. The reference molecule must be set using *SetupRef* method. This method returns an iterator over *OEFlexiOverlapResults* which are sorted based on the TanimotoCombo similarity of that conformer and the reference molecule.

3.1.44 BestOverlay

```
bool BestOverlay(OEFlexiOverlapResults&, OEChem::OEMCMolBase&)
```

This method optimizes the shape and chemical similarity between the fit molecule conformers and a reference molecule, along with intra-molecular forcefield energies of the fit molecule. This method returns the best result only which is the conformer with the highest TanimotoCombo similarity to the reference molecule. The reference molecule must be set using *SetupRef* method.

3.1.45 OEFlexiOverlayOptions

```
class OEFlexiOverlayOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required to perform **flexible fitting** between the fit molecule conformers and a reference molecule using *OEFlexiOverlay*.

See also

- *OEFlexiOverlapOptions* class

The *OEFlexiOverlayOptions* class defines the following public methods:

- *GetMaxOptSteps* and *SetMaxOptSteps*
- *GetRigidOverlay* and *SetRigidOverlay*
- *GetFlexiOverlapOptions* and *SetFlexiOverlapOptions*
- *GetRigidOverlayOptions* and *SetRigidOverlayOptions*

Constructor

```
OEFlexiOverlayOptions()
OEFlexiOverlayOptions(const OEFlexiOverlayOptions&)
```

Default and copy constructors.

Assignment operator.

3.1.46 GetMaxOptSteps

```
unsigned GetMaxOptSteps() const
```

See *SetMaxOptSteps* method.

3.1.47 GetRigidOverlay

```
bool GetRigidOverlay() const
```

See *SetRigidOverlay* method.

3.1.48 GetFlexiOverlapOptions

```
OEFlexiOverlapOptions& GetFlexiOverlapOptions()
const OEFlexiOverlapOptions& GetFlexiOverlapOptions() const
```

Returns a reference to the *OEFlexiOverlapOptions* instance as currently set. These are options related to flexible overlap optimization using shape, color, and force field. See also *SetFlexiOverlapOptions* method.

3.1.49 GetRigidOverlayOptions

```
OEOverlayOptions& GetRigidOverlayOptions()
const OEOverlayOptions& GetRigidOverlayOptions() const
```

Returns a reference to the *OEOverlayOptions* instance as currently set. These are options related to rigid overlay optimization using shape and color. See also *SetRigidOverlayOptions* method.

3.1.50 SetMaxOptSteps

```
bool SetMaxOptSteps(const unsigned)
```

Sets the maximum number of optimization iteration steps in flexible overlay optimization. Default: *2000* steps.

3.1.51 SetRigidOverlay

```
bool SetRigidOverlay(const bool)
```

This control flag determines whether an initial rigid overlay optimization between the fit molecule conformers and the reference system should be performed prior to flexible optimization. Default: *True*.

3.1.52 SetFlexiOverlapOptions

```
void SetFlexiOverlapOptions(const OEFlexiOverlapOptions&)
```

Sets the options related to flexible overlap calculations by passing in *OEFlexiOverlapOptions* instance.

3.1.53 SetRigidOverlayOptions

```
void SetRigidOverlayOptions(const OEOverlayOptions&)
```

Sets the options related to overlay optimization using shape and color by passing in *OEFlexiOverlapOptions* instance.

3.1.54 OEGridColorFunc

```
class OEGridColorFunc : public OEShapeFunc
```

The *OEGridColorFunc* class defines color overlap between a reference object and a fit molecule, using the *Grid* method.

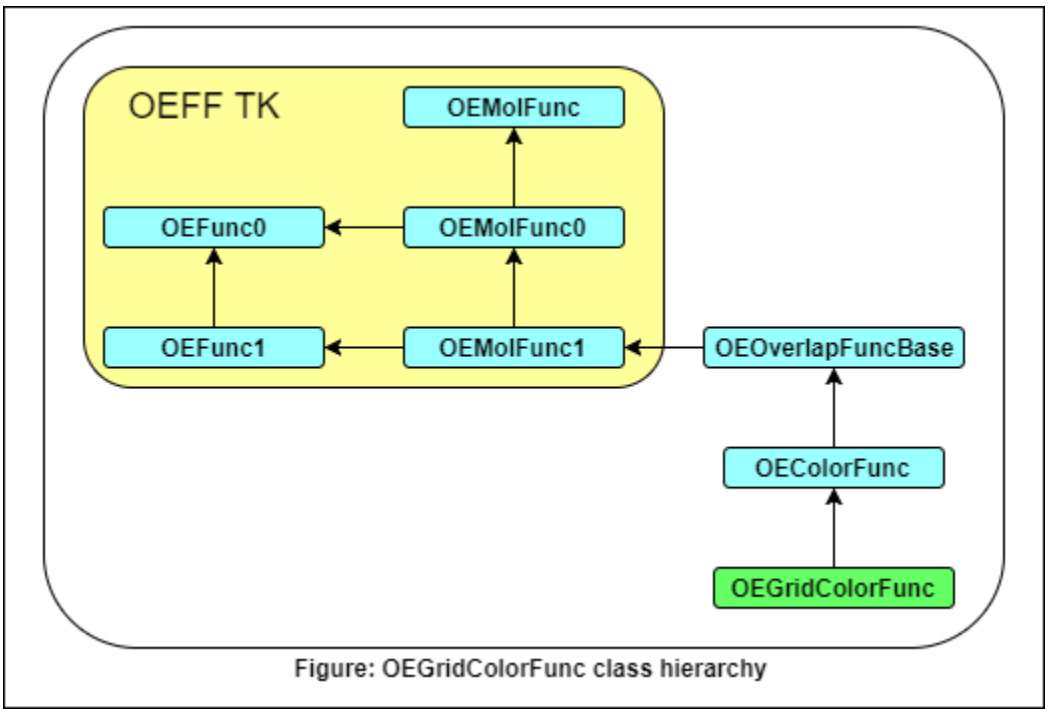


Figure: OEGridColorFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- `operator()`
- `GetFComponents`
- `NumVar`

The following methods are publicly inherited from OEFunc1:

- `operator()`

The following methods are publicly inherited from OEMolFunc:

- `GetInteractions`
- `Setup`

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*
- *SetupRef*

The following methods are publicly inherited from *OEColorFunc*:

- *GetColorOptions*

The *OEGridColorFunc* class defines the following public methods:

- *GetGridOptions*

Constructors

```
OEGridColorFunc(const OEColorOptions& argOptions = OEColorOptions())
OEGridColorFunc(const OEShapeGridOptions&)
OEGridColorFunc(const OEColorOptions& argColor,
                const OEShapeGridOptions& argGrid)
OEGridColorFunc(const OEGridColorFunc&)
```

Default and copy constructors.

Constructs an *OEGridColorFunc* instance using the specified set of parameters.

GetGridOptions

```
const OEShapeGridOptions& GetGridOptions() const
```

Returns a reference to the *OEShapeGridOptions* instance as currently set for this *OEGridColorFunc*.

3.1.55 OEGridShapeFunc

```
class OEGridShapeFunc : public OEShapeFunc
```

The *OEGridShapeFunc* class defines shape overlap between a reference object and a fit molecule, using the *Grid* method.

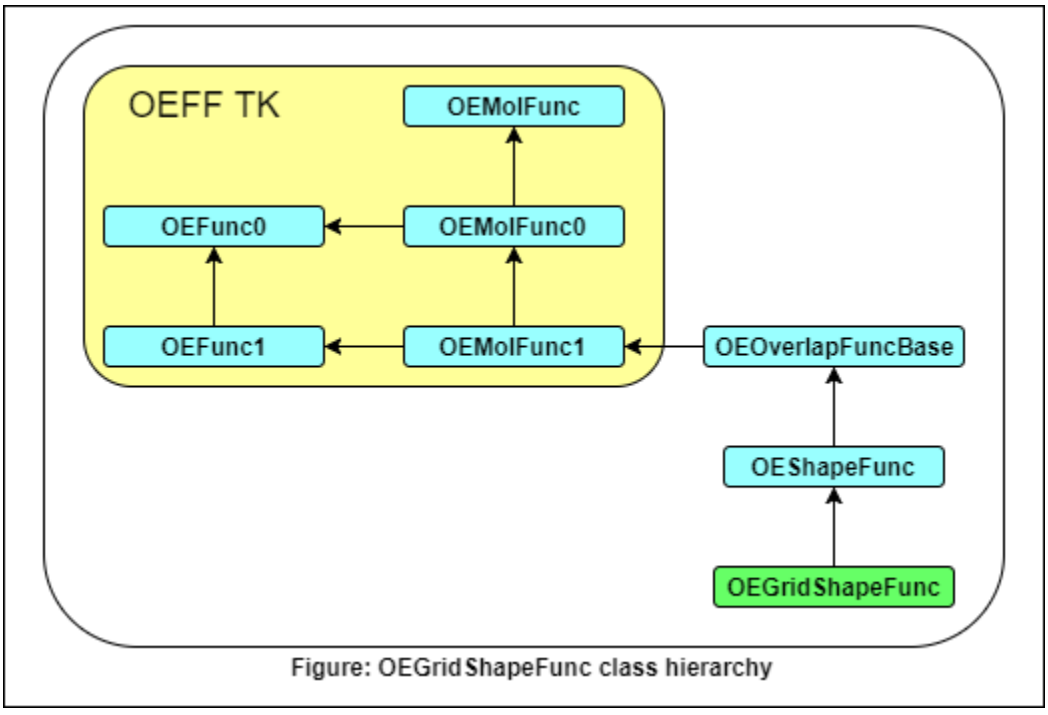


Figure: OEGridShapeFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- `operator()`
- `GetFComponents`
- `NumVar`

The following methods are publicly inherited from OEFunc1:

- operator()

The following methods are publicly inherited from OEMolFunc:

- GetInteractions
- Setup

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*
- *SetupRef*

The following methods are publicly inherited from *OEShapeFunc*:

- *GetShapeOptions*

The *OEGridShapeFunc* class defines the following public methods:

- *GetGridOptions*

Constructors

```
OEGridShapeFunc(const OEShapeOptions& argOptions = OEShapeOptions())
OEGridShapeFunc(const OEShapeGridOptions&)
OEGridShapeFunc(const OEShapeOptions& argShape,
                const OEShapeGridOptions& argGrid)
OEGridShapeFunc(const OEGridShapeFunc&)
```

Default and copy constructors.

Constructs an *OEGridShapeFunc* instance using the specified set of parameters.

GetGridOptions

```
const OEShapeGridOptions& GetGridOptions() const
```

Returns a reference to the *OEShapeGridOptions* instance as currently set for this *OEGridShapeFunc*.

3.1.56 OEHermite

```
class OEHermite
```

This class performs Hermite expansion for a given molecule or a grid representation of a molecule, then returns the coefficients of expansion and creates a grid representation of Hermite shape.

See also

- *Shape from Hermite Representation* section
- *Shape from Hermite expansion* examples

Constructors

```
OEHermite(const OEHermite &)
OEHermite(const OEHermiteOptions& hermOpts = OEHermiteOptions())
```

This class has three constructors. The first two are default and copy constructors. The third one is a constructor that takes an option class *OEHermiteOptions*.

Assignment operator.

CreateGrid

```
void CreateGrid(OESystem::OEScalarGrid&, const float spacing)
void CreateGrid(OESystem::OEScalarGrid&, const float spacing,
                const OEChem::OETrans)
```

This method creates a grid representation of the obtained Hermite expansion. Depending on the grid `spacing` parameter, the number of grid points to cover the entire molecule will be calculated. If this number exceeds 500 points per space dimension, the user will be warned that the maximum allowed value of 500 will be used instead. The second overload uses the transform that is applied to transform the generated grid.

GetCoefficients

```
bool GetCoefficients(std::vector <double>& coeffs) const
const std::vector<double>& GetCoefficients() const
```

Returns the values of Hermite expansion coefficients in the form of an STL vector. These are precisely the coefficients $f_{\ell mn}$ introduced in the *Shape from Hermite Representation*, in the following order: $N = 0 \ldots \text{NPolyMax}$, $\ell = 0 \ldots \text{N}$, $m = 0 \ldots \text{N} - \ell$, $n = \text{N} - \ell - m$.

GetOptions

```
const OEHermiteOptions& GetOptions() const
```

Returns the option class *OEHermiteOptions* that the *OEHermite* class is currently using.

GetSelfOverlap

```
double GetSelfOverlap() const
```

Returns the self-overlap of the molecule corresponding to Hermite expansion.

NumCoefficients

```
unsigned int NumCoefficients() const
```

Returns the size of the Hermite expansion representation. This number is given by $(\text{NPolyMax} + 1)(\text{NPolyMax} + 2)(\text{NPolyMax} + 3)/6$, which is the number of possible sets (ℓ, m, n) such that $\ell + m + n \leq NPolyMax$.

Setup

```
bool Setup(const OEShapeQuery& argQuery)
bool Setup(const OEChem::OEMolBase& mol)
bool Setup(const OESystem::OEScalarGrid& grid)
```

This overloaded method allows a user to input into the *OEHermite* class a molecule as a shape query object or a molecule object, or by using a grid representation. The method returns *True* if the setup was successful and *False* otherwise.

3.1.57 OEHermiteOptions

```
class OEHermiteOptions
```

This is an option class for the *OEHermite* class. It allows the user to input the level of the *Hermite expansion* NPolyMax, the parameters $\lambda_x, \lambda_y, \lambda_z$, and whether or not the parameters need to be optimized inside the *OEHermite* class.

See also

- *Shape from Hermite Representation* section
- *Shape from Hermite expansion* examples

Constructors

```
OEHermiteOptions()
OEHermiteOptions(const OEHermiteOptions &)
```

Default and copy constructors.

Assignment operator.

3.1.58 GetLambdaX

```
double GetLambdaX() const
```

Returns the value of the λ_x parameter.

3.1.59 GetLambdaY

```
double GetLambdaY() const
```

Returns the value of the λ_y parameter.

3.1.60 GetLambdaZ

```
double GetLambdaZ() const
```

Returns the value of the λ_z parameter.

3.1.61 GetNPolyMax

```
unsigned int GetNPolyMax() const
```

Returns the value of the NPolyMax parameter, which controls the level of Hermite expansion. In particular, the lowest value *0* corresponds to a very inaccurate expansion, approximating the entire molecule by a single Gaussian. In the limit NPolyMax $\rightarrow \infty$, we obtain exact equivalence of the Hermite expansion to the original representation of the molecule.

3.1.62 GetUseOptimalLambdas

```
bool GetUseOptimalLambdas() const
```

When *UseOptimalLambdas* is *True*, the parameters $\lambda_x, \lambda_y, \lambda_z$ of the Hermite expansion will be chosen automatically based on the molecule input and resolution of expansion. Alternatively, when *UseOptimalLambdas* is *False*, the current values of the $\lambda_{x,y,z}$ parameters will be used instead. Note that the Hermite expansion will converge regardless of the choice of $\lambda_{x,y,z}$ as long as NPolyMax is high enough, but in practice, at a given low resolution of Hermite expansion, it is useful to use optimal $\lambda_{x,y,z}$ parameters for faster convergence.

3.1.63 SetLambdaX

```
bool SetLambdaX(double)
```

Sets the value of the λ_x parameter. Default: *0.5*.

3.1.64 SetLambdaY

```
bool SetLambdaY(double)
```

Sets the value of the λ_y parameter. Default: *0.5*.

3.1.65 SetLambdaZ

```
bool SetLambdaZ(double)
```

Sets the value of the λ_z parameter. Default: *0.5*.

3.1.66 SetNPolyMax

```
bool SetNPolyMax(unsigned int)
```

Sets the value of the NPolyMax parameter. Default: *4*.

3.1.67 SetUseOptimalLambdas

```
bool SetUseOptimalLambdas(bool)
```

Sets the value of the *UseOptimalLambdas* parameter. Default: *True*.

3.1.68 OEHermiteShapeFunc

```
class OEHermiteShapeFunc : public OEShapeFunc
```

This class is derived from *OEShapeFunc* and is used to approximate Tanimoto shape-similarity coefficient based on Hermite representation of molecules.

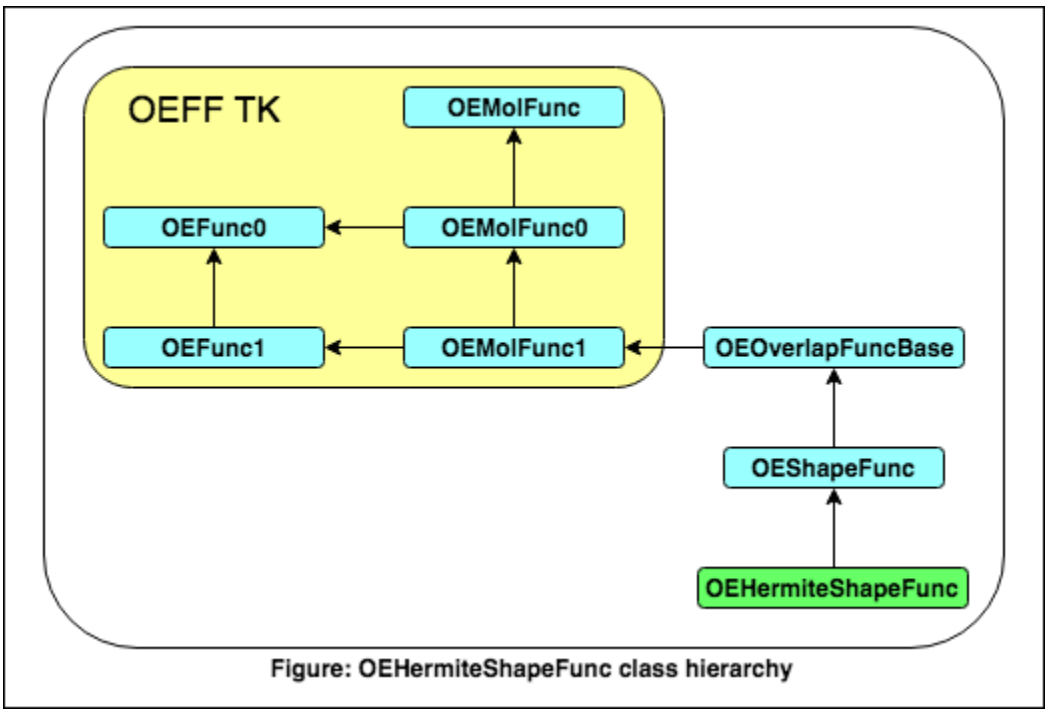


Figure: OEHermiteShapeFunc class hierarchy

See also

- *Shape from Hermite Representation* section
- *OEHermite* class
- *OEHermiteOptions* class
- *Shape from Hermite expansion* examples

The following methods are publicly inherited from OEFunc0:

- `operator()`
- `GetFComponents`
- `NumVar`

The following methods are publicly inherited from OEFunc1:

- `operator()`

The following methods are publicly inherited from OEMolFunc:

- `GetInteractions`
- `Setup`

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupRef*

The following methods are publicly inherited from *OEShapeFunc*:

- *GetShapeOptions*

The *OEHermiteShapeFunc* class defines the following public methods:

- *GetFitHermiteOptions*
- *GetRefHermiteOptions*

Constructors

```
OEHermiteShapeFunc(const OEHermiteShapeFunc &)
OEHermiteShapeFunc(const OEShapeOptions& argOptions = OEShapeOptions())
OEHermiteShapeFunc(const OEHermiteOptions& HermOptRef,
                   const OEHermiteOptions& HermOptFit)
OEHermiteShapeFunc(const OEShapeOptions& argOptions,
                   const OEHermiteOptions& HermOptRef,
                   const OEHermiteOptions& HermOptFit)
```

The four constructors are in the respective order: a standard copy constructor, a constructor that inputs a *OEShapeOptions* object, a constructor that inputs two *OEHermiteOptions* objects, one for reference and one for fit molecules, and finally a constructor that inputs *OEShapeOptions* object along with two *OEHermiteOptions* objects, one for reference and one for fit molecules.

GetFitHermiteOptions

```
const OEHermiteOptions& GetFitHermiteOptions() const
```

This method returns *OEHermiteOptions* class currently being used for the fit molecule.

GetRefHermiteOptions

```
const OEHermiteOptions& GetRefHermiteOptions() const
```

This method returns *OEHermiteOptions* class currently being used for the reference molecule.

3.1.69 OEHighestColorTanimoto

```
struct OEHighestColorTanimoto : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on color Tanimoto.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.70 OEHighestFitColorTversky

```
struct OEHighestFitColorTversky : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on the Tversky color score of the fit molecule.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.71 OEHighestFitTversky

```
struct OEHighestFitTversky : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on fit molecule shape Tversky.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.72 OEHighestFitTverskyCombo

```
struct OEHighestFitTverskyCombo : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on fit molecule shape plus color Tversky.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.73 OEHighestOverlap

```
struct OEHighestOverlap : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on shape overlap.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.74 OEHighestRefColorTversky

```
struct OEHighestRefColorTversky : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on reference molecule color Tversky.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.75 OEHighestRefTversky

```
struct OEHighestRefTversky : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on reference molecule shape Tversky.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.76 OEHighestRefTverskyCombo

```
struct OEHighestRefTverskyCombo : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on reference molecule shape plus color Tversky.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.77 OEHighestScaledColor

```
struct OEHighestScaledColor : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on scaled color.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.78 OEHighestTanimoto

```
struct OEHighestTanimoto : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on shape Tanimoto.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.79 OEHighestTanimotoCombo

```
struct OEHighestTanimotoCombo : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on shape plus color Tanimoto.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.80 OEHighestTverskyCombo

```
class OEHighestTverskyCombo : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Predicate to compare *OEBestOverlayScore* based on shape plus color Tversky.

Constructors

```
OEHighestTverskyCombo(const float alpha)
```

Default constructor. The argument requires the desired alpha value for Tversky calculation.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

3.1.81 OEInertialStarts

```
class OEInertialStarts : public OEStarts
```

The *OEInertialStarts* represents the inertial starts for overlay optimization.

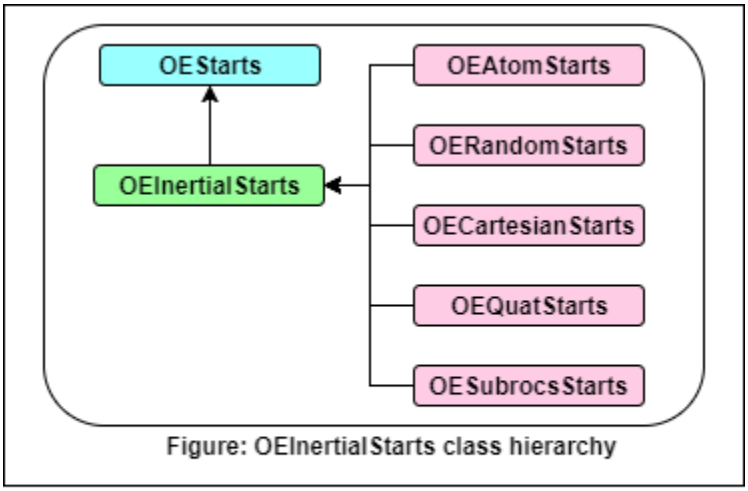


Figure: OEInertialStarts class hierarchy

The following methods are publicly inherited from *OEStarts*:

- *CreateCopy*
- *GetNumOfStarts*
- *GetStarts*
- *Setup*
- *SetupRef*
- *SetupRef*

The *OEInertialStarts* class defines the following public methods:

- *GetSymmetryThreshold*
- *SetSymmetryThreshold*

The following classes derive from this class:

- *OEAtAtomStarts*
- *OECartesianStarts*
- *OEQuatStarts*
- *OERandomStarts*
- *OESubrocsStarts*

Constructors

```
OEInertialStarts()
OEInertialStarts(const OEInertialStarts&)
```

Default and copy constructors.

GetSymmetryThreshold

```
double GetSymmetryThreshold() const
```

Returns the symmetry threshold value currently set.

SetSymmetryThreshold

```
void SetSymmetryThreshold(const double)
```

Sets the symmetry threshold value to be used to determine symmetry of objects. Default: 0.15.

3.1.82 OEIsColorAtomPred

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

Unary predicate that returns true if an atom is a color atom.

Constructors

```
OEIsColorAtomPred()
```

Default and copy constructors.

operator()

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

CreateCopy

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

3.1.83 OEMultiRefOverlay

```
class OEMultiRefOverlay
```

> **Note**
>
> With the default settings, the *OEMultiRefOverlay* optimizes both shape and color.

The *OEMultiRefOverlay* provides an extension to *OEOverlay* to enable setting up a multi-conformer molecule as the reference system for overlay optimization between a reference and a fit molecule conformers.

The *OEMultiRefOverlay* class defines the following public methods:

- *BestOverlay*
- *GetOverlayOptions*
- *Overlay*
- *SetupRef*

Constructors

```
OEMultiRefOverlay(const OEOverlayOptions& argOptions = OEOverlayOptions())
OEMultiRefOverlay(const OEMultiRefOverlay&)
```

Default and copy constructors.

BestOverlay

```
bool BestOverlay(OEBestOverlayScore& argScore, const OEChem::OEMCMolBase& argFitMol,
const OESystem::OEBinaryPredicate<OEBestOverlayScore, OEBestOverlayScore>& argPred =␣
↪OEHighestTanimotoCombo())
```

Method optimizes the overlay between the reference object and the fit molecule conformers, and returns the best result. The reference object must be set using the *SetupRef* method.

GetOverlayOptions

```
const OEOverlayOptions& GetOverlayOptions() const
```

Returns a reference to the *OEOverlayOptions* instance as currently set. This defines options to calculate Overlay.

Overlay

```
OESystem::OEIterBase<OEBestOverlayResults>* Overlay(const OEChem::OEMCMolBase&)
```

Method optimizes the Overlay between the reference molecule and the fit molecule conformers. The reference molecule must be set using the *SetupRef* method.

SetupRef

```
bool SetupRef(const OEChem::OEMCMolBase&)
```

Method defines the interface for setting up the reference system for the *OEMultiRefOverlay* instance with a multi-conformer molecule.

3.1.84 OEOverlapCutoff

```
class OEOverlapCutoff : public OEOverlayScoreCutoff
```

The *OEOverlapCutoff* defines cutoff based on the shape overlap.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OEOverlapCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.85 OEOverlapFunc

```
class OEOverlapFunc : public OEOverlapFuncBase
```

The *OEOverlapFunc* class defines an interface for combined shape and color overlap calculation between a reference object and a fit molecule. The *OEOverlapFunc* is a specialization of The *OEOverlapFuncBase* for combined shape and color overlap.

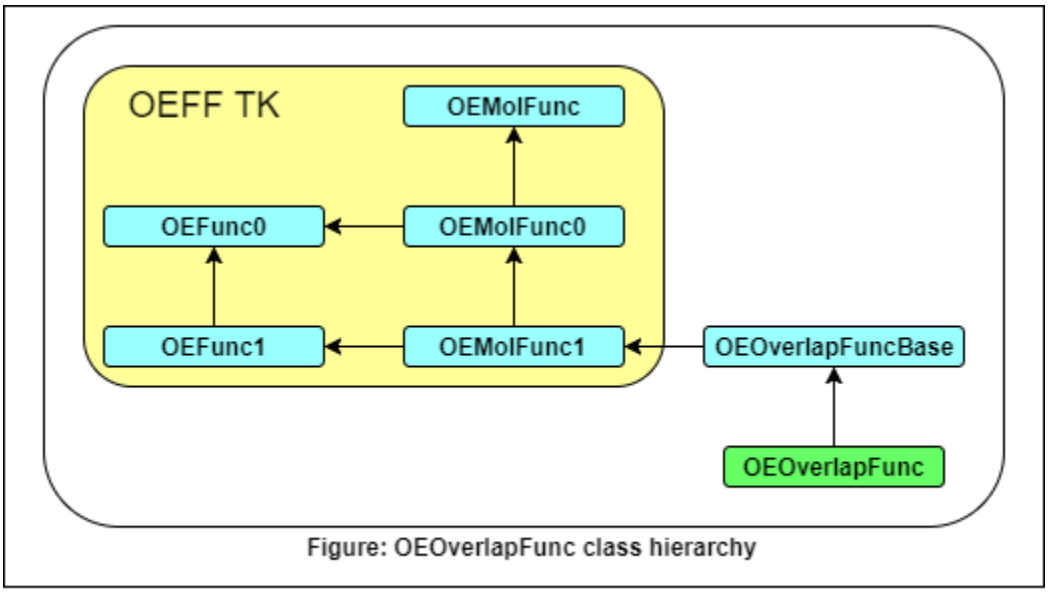


Figure: OEOverlapFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- operator()
- GetFComponents
- NumVar

The following methods are publicly inherited from OEFunc1:

- operator()

The following methods are publicly inherited from OEMolFunc:

- GetInteractions
- Setup

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*

- *SetupRef*

The *OEOverlapFunc* class defines the following public methods:

- *GetColorFunc*
- *GetShapeFunc*

Constructors

```
OEOverlapFunc(const OEShapeFunc& argShape = OEGridShapeFunc(),
              const OEColorFunc& argColor = OEExactColorFunc())
OEOverlapFunc(const OEColorFunc&)
OEOverlapFunc(const OEOverlapFunc&)
```

Default and copy constructors.

Constructs an *OEOverlapFunc* instance using the specified set of parameters.

Assignment operator.

3.1.86 GetColorFunc

```
const OEColorFunc& GetColorFunc() const
```

Returns a reference to the *OEColorFunc* instance as currently set for this *OEOverlapFunc*.

3.1.87 GetShapeFunc

```
const OEShapeFunc& GetShapeFunc() const
```

Returns a reference to the *OEShapeFunc* instance as currently set for this *OEOverlapFunc*.

3.1.88 OEOverlapFuncBase

```
class OEOverlapFuncBase : public OEMolPotential::OEMolFunc1
```

The *OEOverlapFuncBase* is an abstract base class. This class defines an interface for static overlap calculation between a reference object and a fit molecule. The *OEOverlapFuncBase* extends the OEMolFunc1 class to allow setting up a reference object (which could be a molecule, grid or a shape query) and the underlying functions defining interactions based on shape or color overlaps.

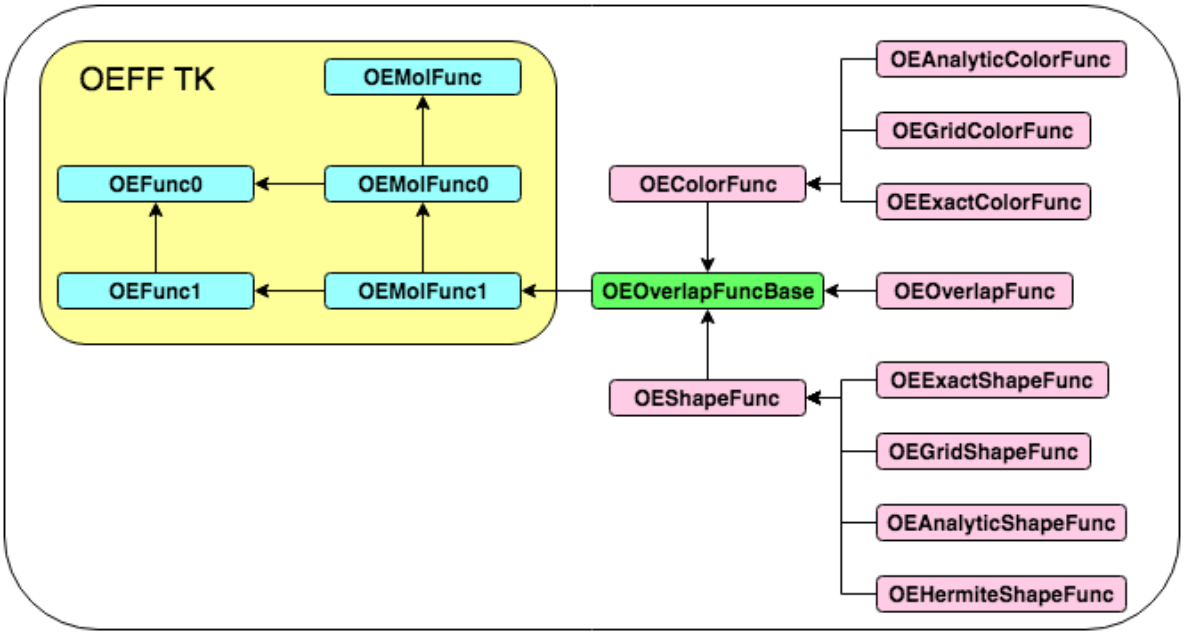


Figure: OEOverlapFuncBase class hierarchy

The following methods are publicly inherited from OEFunc0:

- `operator()`
- `GetFComponents`
- `NumVar`

The following methods are publicly inherited from OEFunc1:

- `operator()`

The following methods are publicly inherited from OEMolFunc:

- `GetInteractions`
- `Setup`

The *OEOverlapFuncBase* class defines the following public methods:

- *Overlap*
- *SetupFlex*
- *SetupRef*

The following classes derive from this class:

- *OEAnalyticColorFunc*
- *OEAnalyticShapeFunc*
- *OEColorFunc*
- *OEExactColorFunc*
- *OEExactShapeFunc*
- *OEGridColorFunc*
- *OEGridShapeFunc*

- *OEOverlapFunc*
- *OEShapeFunc*

3.1.89 Overlap

```
bool Overlap(const double* argCoords, OEOverlapResults &)
bool Overlap(const OEChem::OEMolBase& argFitMol, OEOverlapResults &)
```

These methods calculate the overlap between the reference object and the fit molecule. The reference object must be set using the *SetupRef* method. The first overload assumes that the fit molecule has been set using the Setup method and uses the specified coordinates for the fit molecule. The second overlap takes the specified fit molecule and its current coordinates.

3.1.90 SetupFlex

```
bool SetupFlex(const OEChem::OEMolBase&)
```

This method is a substitute for the Setup method to setup the fit molecule. This method should be called instead of the Setup, when the fit molecule is expected to be flexible and that the shape of the molecule is expected to change.

3.1.91 SetupRef

```
bool SetupRef(const OEChem::OEMolBase&)
bool SetupRef(const OESystem::OEScalarGrid&)
bool SetupRef(const OEShapeQuery&)
```

These method defines the interface for setting up the reference system for the *OEOverlapFuncBase* derived instance with a molecule, grid or a shape query.

3.1.92 OEOverlapPrep

```
class OEOverlapPrep
```

This class provides facilities to prepare molecules for overlap calculations. Preparing a molecule for overlap calculation consists of any or all of the following: adding or removing explicit hydrogens; adding color atoms; and assigning Bondi radii to all the atoms. When color atoms are added to molecules for overlap calculations, it is often required to remove them before further modeling work involving those molecules. Color atoms from molecules can be removed using the *OERemoveColorAtoms* function.

The *OEOverlapPrep* class defines the following public methods:

- *GetAssignBondiRadii*
- *GetAssignColor*
- *GetColorForceField*
- *GetUseHydrogens*
- *Prep*
- *SetAssignBondiRadii*
- *SetAssignColor*
- *SetColorForceField*

- *SetUseHydrogens*

Constructors

```
OEOverlapPrep(const OEOverlapPrepOptions& opts = OEOverlapPrepOptions())
OEOverlapPrep(const OEOverlapPrep&)
```

Default and copy constructors.

Assignment operator.

GetAssignBondiRadii

```
bool GetAssignBondiRadii() const
```

Returns the state of the assign Bondi radii flag. If true, Bondi radii would be assigned to all atoms during the *Prep*.

GetAssignColor

```
bool GetAssignColor() const
```

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

GetColorForceField

```
const OEColorForceField& GetColorForceField() const
```

Gets the instance of *OEColorForceField* currently associated with the *OEOverlapPrep*.

GetUseHydrogens

```
bool GetUseHydrogens() const
```

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

Prep

```
bool Prep(OEChem::OEMCMolBase&)
bool Prep(OEChem::OEMolBase&)
```

Prepares the specified molecule for overlap calculation based on the current set of options set on *Prep*.

SetAssignBondiRadii

```
void SetAssignBondiRadii(const bool)
```

Sets the state of the assign Bondi radii flag. If true, Bondi radii would be assigned to all atoms during the *Prep*.

SetAssignColor

```
void SetAssignColor(const bool)
```

Sets the state of the assign color flag. If true, color atoms would be assigned during the *Prep*. Color atoms from molecules can be removed using the *OERemoveColorAtoms* function for further modeling work involving a molecule.

SetColorForceField

```
bool SetColorForceField(const OEColorForceField& cff)
```

Sets the color force field to be used. By default, the ImplicitMillsDean color force field is used.

SetUseHydrogens

```
void SetUseHydrogens(const bool)
```

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

3.1.93 OEOverlapPrepOptions

```
class OEOverlapPrepOptions : public OESystem::OEOptions
```

This class provides an interface to set up options required for *OEOverlapPrep*. In addition to preparing options available in *OEOverlapPrep*, this option class provides two more additional options to control the *Prep* process regarding the removal of color atoms which are closer than a threshold distance.

The *OEOverlapPrepOptions* class defines the following public methods:

- *GetAssignBondiRadii* and *SetAssignBondiRadii*
- *GetAssignColor* and *SetAssignColor*
- *GetDuplicateDistance* and *SetDuplicateDistance*
- *AddDuplicateSmarts*, *GetDuplicateSmarts*, and *ClearDuplicateSmarts*
- *GetRemoveColor* and *SetRemoveColor*
- *GetRemoveDuplicate* and *SetRemoveDuplicate*
- *GetUseHydrogens* and *SetUseHydrogens*
- *GetColorForceField* and *SetColorForceField*

Constructor

```
OEOverlapPrepOptions(const bool loadColorFF = true)
OEOverlapPrepOptions(const OEOverlapPrepOptions&)
```

Default and copy constructors.

Assignment operator.

GetAssignBondiRadii

```
bool GetAssignBondiRadii() const
```

See *SetAssignBondiRadii* method.

GetAssignColor

```
bool GetAssignColor() const
```

See *SetAssignColor* method.

GetRemoveColor

```
bool GetRemoveColor() const
```

See *SetRemoveColor* method.

GetDuplicateDistance

```
float GetDuplicateDistance() const
```

See *SetDuplicateDistance* method.

GetRemoveDuplicate

```
bool GetRemoveDuplicate() const
```

See *SetRemoveDuplicate* method.

GetUseHydrogens

```
bool GetUseHydrogens() const
```

See *SetUseHydrogens* method.

GetColorForceField

```
const OEColorForceField& GetColorForceField() const
```

See *SetColorForceField* method.

SetAssignBondiRadii

```
void SetAssignBondiRadii(const bool)
```

Sets if **Bondi Radii** should be assigned. If true, Bondi radii would be assigned to all atoms during the *Prep*. Default: *False*.

SetAssignColor

```
void SetAssignColor(const bool)
```

Sets if **color atoms** should be assigned. If true, color atoms would be assigned during the *Prep*. Color atoms are only assigned to molecules that currently do not have any color atoms. To re-assign color atoms to all molecules, use *SetRemoveColor* to *True*. Default: *True*.

SetRemoveColor

```
bool SetRemoveColor(const bool)
```

Sets if existing color atoms should be removed. If true, existing color atoms are removed during *Prep*. Default: *False*.

SetDuplicateDistance

```
bool SetDuplicateDistance(const float)
```

Sets the threshold distance for duplicate removal of color atoms. The acceptable range is between *0.0* and *1.0*. Default *0.0*.

SetRemoveDuplicate

```
bool SetRemoveDuplicate(const bool)
```

Sets if duplicate color atoms should be removed. Default *True*.

SetUseHydrogens

```
bool SetUseHydrogens(const bool)
```

Sets if explicit hydrogen atoms should be used. If true, explicit hydrogen atoms would be created during the *Prep*. If false, all hydrogen atoms would be removed. Default *False*.

SetColorForceField

```
bool SetColorForceField(const OEColorForceField&)
```

Sets the color force field to be used. By default, the *ImplicitMillsDean* color force field is used.

AddDuplicateSmarts

```
bool AddDuplicateSmarts(const std::string& smarts)
```

Adds a SMARTS pattern used to identify duplicate color atoms. Returns *false* if the SMARTS pattern already exists in the list, *true* otherwise. By default, SMARTS patterns for Cubane and Adamantane are included.

GetDuplicateSmarts

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

Returns the list of SMARTS patterns used for identifying duplicate color atoms. See *AddDuplicateSmarts* method.

ClearDuplicateSmarts

```
bool ClearDuplicateSmarts()
```

Removes all SMARTS patterns from the duplicate detection list. See *AddDuplicateSmarts* method.

3.1.94 OEOverlapResults

```
struct OEOverlapResults
```

This class represents overlap calculation results between a reference and a fit object.

The *OEOverlapResults* class defines the following public methods:

- *GetColorScore*
- *GetColorTanimoto*
- *GetColorTversky*
- *GetFitColorTversky*
- *GetFitSelfColor*
- *GetFitSelfOverlap*
- *GetFitTversky*
- *GetFitTverskyCombo*
- *GetOverlap*
- *GetRefColorTversky*
- *GetRefSelfColor*
- *GetRefSelfOverlap*
- *GetRefTversky*
- *GetRefTverskyCombo*
- *GetTanimoto*
- *GetTanimotoCombo*
- *GetTversky*

Constructors

```
OEOverlapResults()
OEOverlapResults(const OEOverlapResults&)
```

Default and copy constructors.

GetColorScore

```
float GetColorScore() const
```

Returns the color score (overlap) between the reference and the fit.

GetColorTanimoto

```
float GetColorTanimoto() const
```

Returns the color tanimoto.

GetColorTversky

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

Returns the color tversky with the specified alpha and beta.

GetFitColorTversky

```
float GetFitColorTversky() const
```

Returns the color tversky with beta = 0.95.

GetFitSelfColor

```
float GetFitSelfColor() const
```

Returns the self color score for the fit molecule.

GetFitSelfOverlap

```
float GetFitSelfOverlap() const
```

Returns the self shape overlap for the fit molecule.

GetFitTversky

```
float GetFitTversky() const
```

Returns the shape tversky with beta = 0.95.

GetFitTverskyCombo

```
float GetFitTverskyCombo() const
```

Returns the sum of shape tversky with beta = 0.95, and the color tversky with beta = 0.95.

GetOverlap

```
float GetOverlap() const
```

Returns the shape overlap between the reference and the fit.

GetRefColorTversky

```
float GetRefColorTversky() const
Returns the color tversky with alpha = 0.95.
```

GetRefSelfColor

```
float GetRefSelfColor() const
```

Returns the self color score for the reference object.

GetRefSelfOverlap

```
float GetRefSelfOverlap() const
```

Returns the self shape overlap for the reference object.

GetRefTversky

```
float GetRefTversky() const
```

Returns the shape tversky with alpha = 0.95.

GetRefTverskyCombo

```
float GetRefTverskyCombo() const
```

Returns the sum of shape tversky with alpha = 0.95, and color tversky with alpha = 0.95.

GetTanimoto

```
float GetTanimoto() const
```

Returns the shape tanimoto.

GetTanimotoCombo

```
float GetTanimotoCombo() const
```

Returns the sum of shape tanimoto and color tanimoto.

GetTversky

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

Returns the shape tversky with the specified alpha and beta.

3.1.95 OEOverlay

```
class OEOverlay
```

> **Note**
>
> With the default settings, the *OEOverlay* optimizes both shape and color.

The *OEOverlay* defined an interface for overlap optimization between a reference object (a molecule, grid or a shape query) and a fit molecule conformers.

The *OEOverlay* class defines the following public methods:

- *BestOverlay*
- *GetOverlayOptions*
- *Overlay*
- *SetupRef*
- *SortedOverlay*

Constructors

```
OEOverlay(const OEOverlayOptions& opts = OEOverlayOptions())
OEOverlay(const OEOverlay&)
```

Default and copy constructors.

Assignment operator.

3.1.96 BestOverlay

```
bool BestOverlay(OEBestOverlayScore& score, const OEChem::OEMCMolBase& fitmol,
const OESystem::OEBinaryPredicate<OEBestOverlayScore, OEBestOverlayScore>&␣
↪pred = OEHighestTanimotoCombo())
```

Method optimizes the overlay between the reference object and the fit molecule conformers, and returns the best result, sorted based on the binary predicate `pred`. The predicate has a default setting of *OEHighestTanimotoCombo*. The reference object must be set using the *SetupRef* method, prior to calling this method.

3.1.97 GetOverlayOptions

```
const OEOverlayOptions& GetOverlayOptions() const
```

Returns a reference to the *OEOverlayOptions* instance as currently set. This defines options to calculate overlay.

3.1.98 Overlay

```
OESystem::OEIterBase<OEBestOverlayResults>* Overlay(const OEChem::OEMCMolBase&)
```

Method optimizes the overlay between the reference object and the fit molecule conformers. The reference object must be set using the *SetupRef* method, prior to calling this method.

See also

- *BestOverlay*
- *SortedOverlay*

3.1.99 SetupRef

```
bool SetupRef(const OEChem::OEMolBase&)
bool SetupRef(const OESystem::OEScalarGrid&)
bool SetupRef(const OEShapeQuery&)
```

These method defines the interface for setting up the reference system for the *OEOverlay* instance with a molecule, grid or a shape query.

3.1.100 SortedOverlay

```
OESystem::OEIterBase<OEBestOverlayScore>* SortedOverlay(const␣
↪OEChem::OEMCMolBase& fitmol,
                                                        const unsigned maxHits,
                                                        const␣
↪OESystem::OEBinaryPredicate<OEBestOverlayScore, OEBestOverlayScore>& pred =␣
↪OEHighestTanimotoCombo())
```

Method optimizes the overlay between the reference object and the fit molecule conformers, and returns the desired number of top hits defined by `maxHits`, sorted based on the binary predicate `pred`. The predicate has a default setting of *OEHighestTanimotoCombo*. The reference object must be set using the *SetupRef* method, prior to calling this method.

This method is specifically preferred over the *Overlay* method when working with molecules containing a large number of conformers, as the burden to store all the results associated with all the conformers can stress the memory limits of a machine.

3.1.101 OEOverlayOptions

```
class OEOverlayOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for *OEOverlay*.

The *OEOverlayOptions* class defines the following public methods:

- *GetColorFuncType* and *SetColorFuncType*
- *GetColorOptions* and *SetColorOptions*
- *GetMaxOptSteps* and *SetMaxOptSteps*
- *GetOverlapFunc* and *SetOverlapFunc*
- *GetShapeFuncType* and *SetShapeFuncType*
- *GetShapeOptions* and *SetShapeOptions*
- *GetStarts* and *SetStarts*

Constructors

```
OEOverlayOptions(const bool loadColorFF = true)
OEOverlayOptions(const OEOverlayOptions&)
```

Default and copy constructors.

Assignment operator.

3.1.102 GetColorFuncType

```
unsigned GetColorFuncType() const
```

See *SetColorFuncType* method.

3.1.103 GetColorOptions

```
OEColorOptions& GetColorOptions()
const OEColorOptions& GetColorOptions() const
```

See *SetColorOptions* method.

3.1.104 GetMaxOptSteps

```
unsigned GetMaxOptSteps() const
```

See *SetMaxOptSteps* method.

3.1.105 GetOverlapFunc

```
const OEOverlapFuncBase& GetOverlapFunc() const
```

See *SetOverlapFunc* method.

3.1.106 GetShapeFuncType

```
unsigned GetShapeFuncType() const
```

See *SetShapeFuncType* method.

3.1.107 GetShapeOptions

```
OEShapeOptions& GetShapeOptions()
const OEShapeOptions& GetShapeOptions() const
```

See *SetShapeOptions* method.

3.1.108 GetStarts

```
const OEStarts& GetStarts() const
```

See *SetStarts* method.

3.1.109 SetColorFuncType

```
bool SetColorFuncType(const unsigned)
```

Set the **type of color func** to be used for estimating color scores, as defined in the *OEColorType* namespace. **Default:** *OEColorType.Exact*.

3.1.110 SetColorOptions

```
void SetColorOptions(const OEColorOptions&)
```

Sets *options* related to color functions used.

3.1.111 SetMaxOptSteps

```
bool SetMaxOptSteps(const unsigned int)
```

Set the **maximum number of optimization** iteration steps. **Default: 200**

3.1.112 SetOverlapFunc

```
bool SetOverlapFunc(const OEOverlapFuncBase&)
```

Set the overlap function by by passing in an *OEOverlapFuncBase* instance. This defines the shape/color interactions to be optimized for overlay.

3.1.113 SetShapeFuncType

```
bool SetShapeFuncType(const unsigned)
```

Set the **type of shape func** to be used for estimating shape overlap, as defined in the *OEShapeType* namespace. **Default:** *OEShapeType.Grid*.

3.1.114 SetShapeOptions

```
void SetShapeOptions(const OEShapeOptions&)
```

Sets *options* related to shape functions used.

3.1.115 SetStarts

```
void SetStarts(const OEStarts&)
```

Set the starts by by passing in an *OEStarts* instance. This defines the starts to be used for optimization.

3.1.116 OEOverlayScoreCutoff

```
class OEOverlayScoreCutoff : public OESystem::OEUnaryPredicate<OEBestOverlayScore>
```

The *OEOverlayScoreCutoff* is an abstract base class. This class defines an interface specifying cutoff criteria for finding *OEROCS* hits.

The following methods are publicly inherited from OEUnaryPredicate:

- `CreatePredicateCopy`

The *OEOverlayScoreCutoff* class defines the following public methods:

- *operator()*
- *CreateCopy*

operator()

```
bool operator()(const OEBestOverlayScore&) const
```

Method returns *True* if the specified score satisfies the cutoff criteria.

CreateCopy

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

Returns a new copied instance of this unary function.

3.1.117 OEQuatStarts

```
class OEQuatStarts : public OEInertialStarts
```

The *OEQuatStarts* represents the user defined quaternion starts for overlay optimization. The *OEQuatStarts* takes the user defined quaternions as the starts, with reference to the inertial frame of the reference system.

The following methods are publicly inherited from *OEStarts*:

- *CreateCopy*
- *GetNumOfStarts*
- *GetStarts*
- *Setup*
- *SetupRef*
- *SetupRef*

The following methods are publicly inherited from *OEInertialStarts*:

- *GetSymmetryThreshold*
- *SetSymmetryThreshold*

The *OEQuatStarts* class defines the following public methods:

- *ClearStarts*
- *SetStarts*

Constructors

```
OEQuatStarts()
OEQuatStarts(const OEQuatStarts&)
```

Default and copy constructors.

ClearStarts

```
void ClearStarts()
```

Clears the currently specified starts.

SetStarts

```
void SetStarts(const double* Quats, unsigned int NumQuats)
```

Sets the starting quaternions. Method expects an array of size 7*NumQuats for the first argument.

3.1.118 OERefColorTverskyCutoff

```
class OERefColorTverskyCutoff : public OEOverlayScoreCutoff
```

The *OERefColorTverskyCutoff* defines cutoff based on the reference system color tversky.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OERefColorTverskyCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.119 OERefTverskyComboCutoff

```
class OERefTverskyComboCutoff : public OEOverlayScoreCutoff
```

The *OERefTverskyComboCutoff* defines cutoff based on the reference system combined shape and color tversky.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OERefTverskyComboCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.120 OERefTverskyCutoff

```
class OERefTverskyCutoff : public OEOverlayScoreCutoff
```

The *OERefTverskyCutoff* defines cutoff based on the reference system shape tversky.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OERefTverskyCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.121 OERandomStarts

```
class OERandomStarts : public OEInertialStarts
```

The *OERandomStarts* represents the random starts for overlay optimization. The *OERandomStarts* translates the fit molecule to specified number of random starting points, followed by the inertial rotations to generate starts.

The following methods are publicly inherited from *OEStarts*:

- *CreateCopy*
- *GetNumOfStarts*
- *GetStarts*
- *Setup*
- *SetupRef*
- *SetupRef*

The following methods are publicly inherited from *OEInertialStarts*:

- *GetSymmetryThreshold*
- *SetSymmetryThreshold*

The *OERandomStarts* class defines the following public methods:

- *GetMaxRandomTranslation*
- *GetNumRandomStarts*
- *GetRandomSeed*
- *SetMaxRandomTranslation*
- *SetNumRandomStarts*
- *SetRandomSeed*

Constructors

```
OERandomStarts()
OERandomStarts(const OERandomStarts&)
```

Default and copy constructors.

GetMaxRandomTranslation

```
double GetMaxRandomTranslation() const
```

Returns the maximum allowed translation value currently set.

GetNumRandomStarts

```
unsigned int GetNumRandomStarts() const
```

Returns the current value of the number of random starting points.

GetRandomSeed

```
unsigned int GetRandomSeed() const
```

Returns the current value of the seed for random number generation.

SetMaxRandomTranslation

```
void SetMaxRandomTranslation(const double)
```

Sets the maximum allowed translation for generating the starting points. Default: 2.0.

SetNumRandomStarts

```
unsigned int GetNumRandomStarts() const
```

Sets the number of random starting points to be generated. Default: 4.

SetRandomSeed

```
void SetRandomSeed(const unsigned int argSeed)
```

Sets the seed for random number generation. Default: 0.

3.1.122 OEROCS

```
class OEROCS
```

The *OEROCS* defines an interface for finding ROCS hits from a search database, by overlaying a reference object (a molecule, grid or a shape query) .

The *OEROCS* class defines the following public methods:

- *AddMolecule*
- *ClearMolecules*
- *GetROCSOptions*
- *SetDatabase*
- *Overlay*

Constructors

```
OEROCS(const OEROCSOptions& argOptions = OEROCSOptions())
OEROCS(const OEROCS&)
```

Default and copy constructors.

AddMolecule

```
unsigned int AddMolecule(const OEChem::OEMCMolBase& argMol)
```

Adds a copy of the specified molecule to the search database. Returns the index of the molecule in the search database.

ClearMolecules

```
void ClearMolecules()
```

Remove all molecules and clear the search database.

GetROCSOptions

```
const OEROCSOptions& GetROCSOptions() const
```

Returns a reference to the *OEROCSOptions* instance as currently set. This defines options to perform *OEROCS* calculations.

SetDatabase

```
bool SetDatabase(const OEChem::OEMolDatabase&)
bool SetDatabase(OEChem::oemolistream&)
```

Sets the search database with molecules from specified database or file stream. These methods cleans up any previous existing content of the search database.

Overlay

```
OESystem::OEIterBase<OEROCSResult>* Overlay(const OEChem::OEMolBase&)
OESystem::OEIterBase<OEROCSResult>* Overlay(const OEChem::OEMCMolBase&)
OESystem::OEIterBase<OEROCSResult>* Overlay(const OESystem::OEScalarGrid&)
OESystem::OEIterBase<OEROCSResult>* Overlay(const OEShapeQuery&)
```

Method optimizes the overlay between the reference object and the molecule conformers in the search database.

3.1.123 OEROCSOptions

```
class OEROCSOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for *OEROCS*.

The *OEROCSOptions* class defines the following public methods:

- *AddCutoff*
- *ClearCutoffs*

- *GetConfsPerHit*
- *GetMaxHits*
- *GetNumBestHits*
- *GetOverlayOptions*
- *GetPerformPrep*
- *GetRankPredicate*
- *GetUseMaxHits*
- *HasCutoff*
- *MeetsCutoff*
- *SetConfsPerHit*
- *SetMaxHits*
- *SetNumBestHits*
- *SetOverlayOptions*
- *SetPerformPrep*
- *SetRankPredicate*
- *SetUseMaxHits*

Constructors

```
OEROCSOptions()
OEROCSOptions(const OEROCSOptions&)
```

Default and copy constructors.

AddCutoff

```
void AddCutoff(const OESystem::OEUnaryPredicate<OEBestOverlayScore>&)
```

Add a new cutoff criteria for scanning if a hit is acceptable. Multiple cutoff criteria can used for a single *OEROCS* calculation.

ClearCutoffs

```
void ClearCutoffs()
```

Remove all currently defined cutoffs.

GetConfsPerHit

```
unsigned int GetConfsPerHit() const
```

Returns the current value for the desired number of conformers to be kept per hit in the *OEROCSResult* in a *OEROCS* calculation.

GetMaxHits

```
unsigned int GetMaxHits() const
```

Returns the current value of the maximum number of hits to be looked for in a *OEROCS* calculation, based on cutoffs.

GetNumBestHits

```
unsigned int GetNumBestHits() const
```

Returns the current value of the maximum number of best hit results to be returned from a *OEROCS* calculation.

GetOverlayOptions

```
const OEOverlayOptions& GetOverlayOptions() const
```

Returns an instance of the *OEOverlayOptions* as currently set for overlay optimization. These options are used in overlay optimization of conformers.

GetPerformPrep

```
bool GetPerformPrep() const
```

Returns the current state of the perform prep flag. This flag defines if the reference and fit molecules should be prepared (or used as is), before *OEROCS* calculations.

GetRankPredicate

```
const OESystem::OEBinaryPredicate<OEBestOverlayScore, OEBestOverlayScore>&␣
↪GetRankPredicate()
```

Returns an instance of the binary predicate as currently set for ranking the obtained hits.

GetUseMaxHits

```
bool GetUseMaxHits() const
```

Returns the current state of the use max hits flag. This flag defines if a *OEROCS* calculation should be terminated when the defined maximum hits is reached.

HasCutoff

```
bool HasCutoff() const
```

Returns *True* if any cutoffs are currently set, *False* otherwise.

MeetsCutoff

```
bool MeetsCutoff(const OEBestOverlayScore&) const
```

Returns *True* if the specified score meets the cutoffs criterion currently set, *False* otherwise.

SetConfsPerHit

```
bool SetConfsPerHit(const unsigned int)
```

Sets the value for the desired number of conformers to be kept per hit in the *OEROCSResult* in a *OEROCS* calculation. Default: 1.

SetMaxHits

```
bool SetMaxHits(const unsigned int)
```

Sets the value of the maximum number of hits to be looked for in a *OEROCS* calculation, based on cutoffs. Default: 500.

SetNumBestHits

```
bool SetNumBestHits(const unsigned int)
```

Sets the value of the maximum number of best hit results to be returned from a *OEROCS* calculation. Default: 500.

SetOverlayOptions

```
void SetOverlayOptions(const OEOverlayOptions&)
```

Sets the *OEOverlayOptions* for overlay optimization. These options are used in overlay optimization of conformers.

SetPerformPrep

```
bool SetPerformPrep(const bool)
```

Sets the state of the perform prep flag. This flag defines if the reference and fit molecules should be prepared (or used as is), before *OEROCS* calculations. Default: *True*.

SetRankPredicate

```
void SetRankPredicate(const OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>&)
```

Sets the binary predicate to be used for ranking the obtained hits.

SetUseMaxHits

```
bool SetUseMaxHits(const bool)
```

Sets the state of the use max hits flag. This flag defines if a *OEROCS* calculation should be terminated when the defined maximum hits is reached. Default: *False*.

3.1.124 OEROCSResult

```
class OEROCSResult : public OEOverlapResults
```

This class extends the *OEBestOverlayScore* to represent the results of a single *OEROCS* calculation.

The following methods are publicly inherited from *OEOverlapResults*:

- *GetColorScore*
- *GetColorTanimoto*
- *GetColorTversky*
- *GetFitColorTversky*
- *GetFitSelfColor*
- *GetFitSelfOverlap*
- *GetFitTversky*
- *GetFitTverskyCombo*
- *GetOverlap*
- *GetRefColorTversky*
- *GetRefSelfColor*
- *GetRefSelfOverlap*
- *GetRefTversky*
- *GetRefTverskyCombo*
- *GetTanimoto*
- *GetTanimotoCombo*
- *GetTversky*

The following methods are publicly inherited from *OEBestOverlayScore*:

- *GetFitConfIdx*
- *GetRefConfIdx*
- *GetRotMatrix*
- *GetTranslation*
- *Transform*

The *OEROCSResult* class defines the following public methods:

- *AddScore*
- *GetOverlayConf*
- *GetOverlayConfs*

Constructors

```
OEROCSResult()
OEROCSResult(const OEBestOverlayScore&, const OEChem::OEMCMolBase&)
OEROCSResult(const OEROCSResult&)
```

Default and copy constructors. The second constructor takes in an instance of a *OEBestOverlayScore* and the corresponding fit molecule.

AddScore

```
void AddScore(const OEBestOverlayScore&, const OEChem::OEMCMolBase&)
```

Add a new conformer to the score. This method is useful for generating a multi-conformer *OEROCSResult*.

GetOverlayConf

```
const OEChem::OEMolBase& GetOverlayConf() const
```

Returns the best hit conformer in the overlaid state.

GetOverlayConfs

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

Returns the best hit conformers in the overlaid state, for a multi-conformer result.

3.1.125 OEShapeFunc

```
class OEShapeFunc : public OEOverlapFuncBase
```

The *OEShapeFunc* is an abstract base class. This class defines an interface for shape overlap calculation between a reference object and a fit molecule. The *OEShapeFunc* is a specialization of The *OEOverlapFuncBase* for shape only overlap.

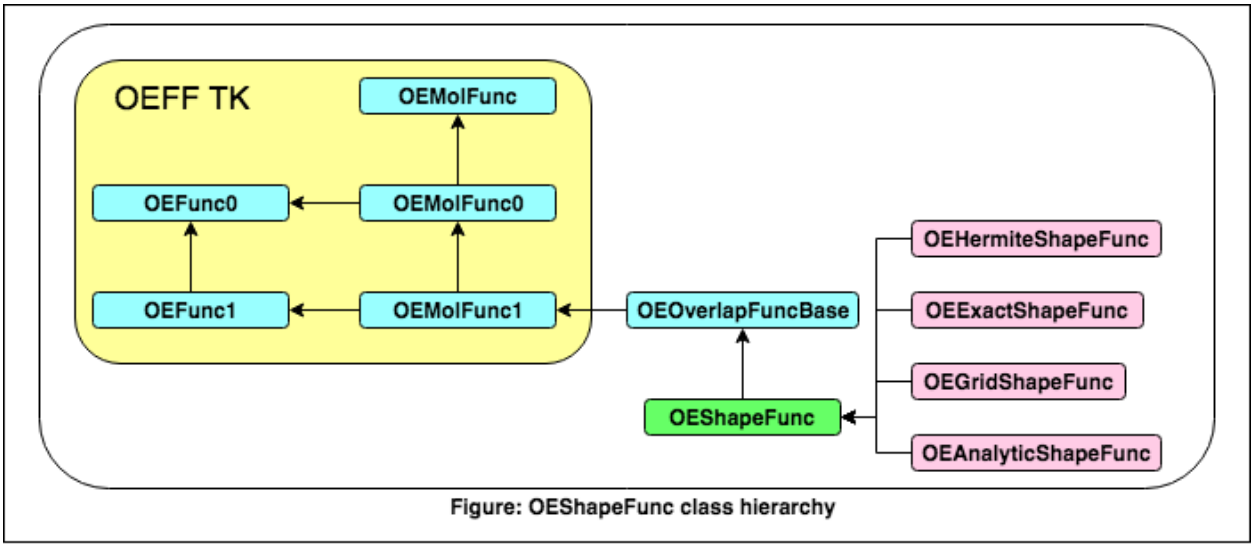


Figure: OEShapeFunc class hierarchy

The following methods are publicly inherited from OEFunc0:

- operator()
- GetFComponents
- NumVar

The following methods are publicly inherited from OEFunc1:

- operator()

The following methods are publicly inherited from OEMolFunc:

- GetInteractions
- Setup

The following methods are publicly inherited from *OEOverlapFuncBase*:

- *Overlap*
- *SetupFlex*
- *SetupRef*

The *OEShapeFunc* class defines the following public methods:

- *GetShapeOptions*

The following classes derive from this class:

- *OEAnalyticShapeFunc*
- *OEExactShapeFunc*
- *OEGridShapeFunc*

GetShapeOptions

```
const OEShapeOptions& GetShapeOptions() const
```

Returns a reference to the *OEShapeOptions* instance as currently set for this *OEShapeFunc*.

3.1.126 OEShapeGridOptions

```
class OEShapeGridOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for calculations using the *Grid* method.

The *OEShapeGridOptions* class defines the following public methods:

- *GetDerivativeType*
- *GetGridSpacing*
- *SetDerivativeType*
- *SetGridSpacing*

Constructors

```
OEShapeGridOptions()
OEShapeGridOptions(const OEShapeGridOptions&)
```

Default and copy constructors.

GetDerivativeType

```
unsigned int GetDerivativeType() const
```

Gets the current value of the derivative type to be used on *Grid* method.

GetGridSpacing

```
double GetGridSpacing() const
```

Gets the grid spacing to be used on *Grid* method.

SetDerivativeType

```
bool SetDerivativeType(const unsigned int)
```

Set the derivative type to be used on *Grid* method. The default is *OEDerivativeType.NearestValue* for shape overlap and *OEDerivativeType.Interpolated* for color overlap. Alternatives are defined in the *OEDerivativeType* namespace.

SetGridSpacing

```
bool SetGridSpacing(const double)
```

Set the grid spacing to be used on *Grid* method. By default this is set to 0.25 Å.

3.1.127 OEShapeOptions

```
class OEShapeOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for *OEShapeFunc* calculations.

The *OEShapeOptions* class defines the following public methods:

- *GetCarbonRadius*
- *GetRadiiApproximation*
- *GetScoreType*
- *SetCarbonRadius*
- *SetRadiiApproximation*
- *SetScoreType*

Constructors

```
OEShapeOptions()
OEShapeOptions(const OEShapeOptions&)
```

Default and copy constructors.

GetCarbonRadius

```
double GetCarbonRadius() const
```

Gets the radius value for the carbon radius approximation, associated with *OEOverlapRadii.Carbon*.

GetRadiiApproximation

```
unsigned int GetRadiiApproximation() const
```

Gets the current value of the radii approximation.

GetScoreType

```
unsigned int GetScoreType() const
```

Gets the current value of the score type of the function.

SetCarbonRadius

```
bool SetCarbonRadius(double cradius)
```

Set the radius to use when using *OEOverlapRadii.Carbon*. By default this is set to 1.7 Å.

SetRadiiApproximation

```
bool SetRadiiApproximation(unsigned int type)
```

Set the radius approximation used to calculate overlap. The default is *OEOverlapRadii.Carbon*. Alternatives are defined in the *OEOverlapRadii* namespace.

SetScoreType

```
bool SetScoreType(unsigned int type)
```

Set the score type to be reported by this function. The default is *OEScoreType.Tanimoto*. Alternatives are defined in the *OEScoreType* namespace.

3.1.128 OEStarts

```
class OEStarts
```

The *OEStarts* is an abstract base class. This class defines an interface for providing various kinds of starts for overlay optimization calculations. Most methods in this interface are useful for advanced users who want to create their own optimization workflow using these starts. To specify starts for *OEOverlay* or *OEROCS* calculations, a user merely needs to instantiate the appropriate *OEStarts*.

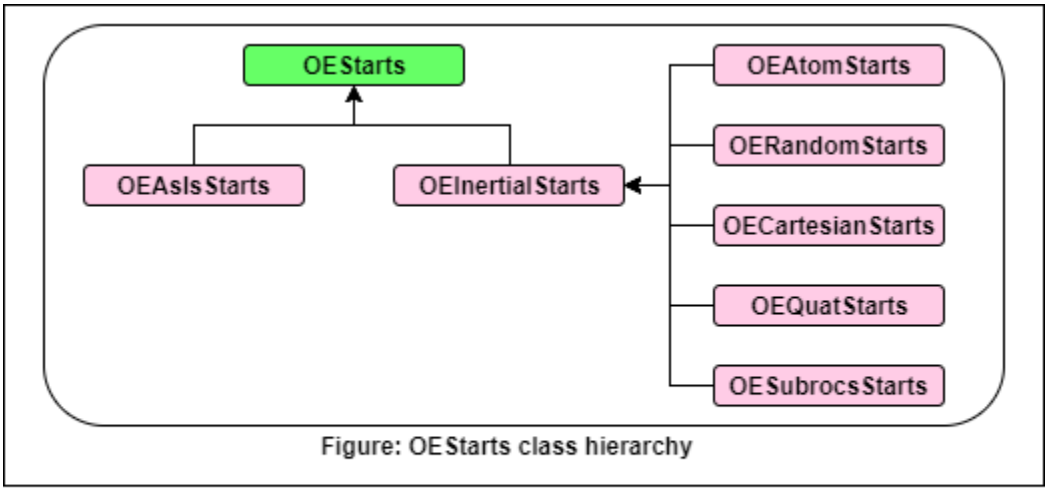


Figure: OEStarts class hierarchy

The *OEStarts* class defines the following public methods:

- *CreateCopy*
- *GetNumOfStarts*
- *GetStarts*
- *Setup*
- *SetupRef*
- *SetupRef*

The following classes derive from this class:

- *OEAsIsStarts*
- *OEAtAtomStarts*
- *OECartesianStarts*
- *OEInertialStarts*
- *OEQuatStarts*
- *OERandomStarts*
- *OESubrocsStarts*

CreateCopy

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

Returns a new instance of an *OEStarts* with the same parameters as the inherent starts.

GetNumOfStarts

```
unsigned int GetNumOfStarts() const
```

Returns the number of starts currently set. In most cases, the number of starts depends on the reference and the fit object to be used for optimization. Both *SetupRef* and *Setup* should be called to set up the reference and fit information prior to accessing the number of starts for any specific optimization.

GetStarts

```
bool GetStarts(double* argQuats) const
```

Returns the starting quaternions currently set. In most cases, the number of starts depends on the reference and the fit object to be used for optimization. Both *SetupRef* and *Setup* should be called to set up the reference and fit information prior to accessing the number of starts for any specific optimization.

The array passed in for obtaining the starting quaternions must have a length of at least 7*(NumStarts).

Setup

```
unsigned int Setup(const OEChem::OEMolBase& argFitMol,
                   OEChem::OETrans& argTrans)
bool Setup(const unsigned int argFitSymmetry)
```

The first *Setup* defines the interface for setting up the fit system for the *OEStarts* derived instance with a molecule. The second argument returns the transformation required to be applied on the fit molecule prior to starting any optimization. The return value of the method corresponds to the calculated symmetry type of the fit molecule. The second *Setup* can be used for setting up the fit system when the symmetry type is already known. The reference system information must be set by calling *SetupRef* prior to calling *Setup*.

SetupFrame

```
bool SetupFrame(OEChem::OEMolBase& mol)
bool SetupFrame(OESystem::OEScalarGrid& grid)
bool SetupFrame(OEShapeQuery& query)
```

These methods define the interface for setting up the reference system for the *OEStarts* derived instance. These methods provide simpler interface to setting up the reference system, over the *SetupRef* methods.

SetupRef

```
bool SetupRef(const unsigned int symmetry)
bool SetupRef(const OEChem::OEMolBase& argRefMol,
              const unsigned int symmetry)
```

These methods define the interface for setting up the reference system for the *OEStarts* derived instance. The knowledge of the reference molecule is required for some variants of the *OEStarts*, whereas the others simply require the `symmetry` information for the reference system. The corresponding `symmetry` values are defined in the *OEMomentSymmetry* namespace and can be calculated using the *OEOrientByMomentsOfInertia* method.

See also

- *SetupFrame* method

3.1.129 OEShapeQuery

```
class OEShapeQuery : public OEShapeQueryBase
```

The *OEShapeQuery* class defines an interface for creating a shape query, which is useful for defining a reference system for shape and color overlap calculations.

The following methods are publicly inherited from OEBase:

- operator=
- operator+=
- AddBaseData
- AddData
- CreateCopy
- DeleteData
- GetBoolData
- GetData
- GetDataIter
- GetDataType
- GetDoubleData
- GetFloatData
- GetIntData
- GetStringData
- HasData
- IsDataType
- SetBaseData
- SetBoolData
- SetData
- SetDoubleData
- SetFloatData
- SetIntData
- SetStringData

The following methods are publicly inherited from *OEShapeQueryBase*:

- *GetAtomStrength*
- *GetColorForceField*
- *GetColorGaussians*
- *GetCompositeMolecule*
- *GetMolecule*
- *GetShapeGaussians*
- *GetShapeGrid*

- *GetTitle*
- *HasColor*
- *HasMolecule*
- *HasShape*
- *HasShapeGrid*
- *IsEmpty*
- *SetTitle*

The *OEShapeQuery* class defines the following public methods:

- *AddColorGaussian*
- *AddColorGaussians*
- *AddShapeGaussian*
- *AddShapeGaussians*
- *Clear*
- *ClearColorGaussians*
- *ClearMolecule*
- *ClearShapeGaussians*
- *ClearShapeGrid*
- *DeleteColorGaussian*
- *DeleteShapeGaussian*
- *SetAtomStrength*
- *SetMolecule*
- *SetShapeGrid*

Constructors

```
OEShapeQuery(const OEShapeQueryOptions& argOptions = OEShapeQueryOptions())
OEShapeQuery(const OEShapeQuery&)
OEShapeQuery(const OEShapeQueryBase&);
```

Default and copy constructors.

Assignment operator.

AddColorGaussian

```
void AddColorGaussian(const OESystem::OEGaussianBase&)
```

Adds the specified color Gaussian to the query.

AddColorGaussians

```
unsigned AddColorGaussians(const OESystem::OEScalarGrid& grid,
                           const unsigned colorType)
unsigned AddColorGaussians(const OESystem::OEScalarGrid& grid,
                           const unsigned colorType,
                           const double relGridContour)
```

Adds Gaussians of the specified color type to the query, based on the input grid. Gaussians are placed to reproduce the shape of the input grid. The optional argument `relGridContour` refers to the grid values relative to the maximum, below which a Gaussian should not be placed.

AddShapeGaussian

```
void AddShapeGaussian(const OESystem::OEGaussianBase&)
```

Adds the specified shape Gaussian to the query.

AddShapeGaussians

```
unsigned AddShapeGaussians(const OESystem::OEScalarGrid& grid)
```

Adds shape Gaussians to the query, based on the input grid. Gaussians are placed to reproduce the shape of the input grid. The optional argument `options` refers to the options class that controls the grid to Gaussian algorithm.

Clear

```
void Clear()
```

Removes all the current contents from the query.

ClearColorGaussians

```
void ClearColorGaussians()
```

Removes all the color Gaussians from the query.

ClearMolecule

```
void ClearMolecule()
```

Removes the molecule from the query.

ClearShapeGaussians

```
void ClearShapeGaussians()
```

Removes all the shape Gaussians from the query.

ClearShapeGrid

```
void ClearShapeGrid()
```

Removes the shape grid from the query.

DeleteColorGaussian

```
bool DeleteColorGaussian(OESystem::OEGaussianBase*)
```

Removes the specified color Gaussian from the query.

DeleteShapeGaussian

```
bool DeleteShapeGaussian(OESystem::OEGaussianBase*)
```

Removes the specified shape Gaussian from the query.

SetAtomStrength

```
bool SetAtomStrength(OEChem::OEAtomBase& atom,
                     const unsigned strength)
```

Sets the strength of the specified atom. A strength of larger than 1 means that the atom should be represented by multiple atoms at the same location. The atom must be a part of the molecule contained in the query and may not be a color atom. The method returns `False` if the strength cannot be set.

SetMolecule

```
void SetMolecule( const OEChem::OEMolBase&)
```

Adds the specified molecule to the query.

SetShapeGrid

```
void SetShapeGrid(const OESystem::OEScalarGrid&)
```

Adds the specified shape grid to the query.

3.1.130 OEShapeQueryBase

```
class OEShapeQueryBase : public OEBase
```

The *OEShapeQueryBase* is an abstract base class. This class defines an interface for creating a shape query, useful for defining a reference system for shape/color overlap calculations.

The following methods are publicly inherited from OEBase:

- operator=
- operator+=
- AddBaseData
- AddData
- CreateCopy
- DeleteData
- GetBoolData
- GetData
- GetDataIter

- GetDataType
- GetDoubleData
- GetFloatData
- GetIntData
- GetStringData
- HasData
- IsDataType
- SetBaseData
- SetBoolData
- SetData
- SetDoubleData
- SetFloatData
- SetIntData
- SetStringData

The *OEShapeQueryBase* class defines the following public methods:

- *GetAtomStrength*
- *GetColorForceField*
- *GetColorGaussians*
- *GetCompositeMolecule*
- *GetMolecule*
- *GetShapeGaussians*
- *GetShapeGrid*
- *GetTitle*
- *HasColor*
- *HasMolecule*
- *HasShape*
- *HasShapeGrid*
- *IsEmpty*
- *SetTitle*

The following classes derive from this class:

- *OEShapeQuery*

GetAtomStrength

```
unsigned GetAtomStrength(const OEChem::OEAtomBase&) const
```

Returns the strength of the specified atom. The default strength of any atom that is part of the molecule within the query is 1.

GetColorForceField

```
const OEColorForceField* GetColorForceField() const
```

Returns a pointer to the color force field associated with this query.

GetColorGaussians

```
OESystem::OEIterBase<const OESystem::OEGaussianBase>* GetColorGaussians() const
```

Returns an iterator over the color gaussians currently contained in this query.

GetCompositeMolecule

```
bool GetCompositeMolecule(OEChem::OEMolBase& mol) const
```

This method is useful in extracting the molecule contained in the query with atom strengths explicitly taken care of. Atoms with strength larger than 1 are explicitly represented by multiple atoms in the composite molecule.

GetMolecule

```
const OEChem::OEMolBase& GetMolecule() const
```

Returns the molecule currently contained in this query. In contrast to the *GetCompositeMolecule* method, this method returns the molecule as is.

GetShapeGaussians

```
OESystem::OEIterBase<const OESystem::OEGaussianBase>* GetShapeGaussians() const
```

Returns an iterator over the shape gaussians currently contained in this query.

GetShapeGrid

```
const OESystem::OEScalarGrid& GetShapeGrid() const
```

Returns the shape grid currently contained in this query.

GetTitle

```
const char* GetTitle() const
```

Returns the title of the query.

HasColor

```
bool HasColor() const
```

Returns *True* if the query currently contains any color atoms or color gaussians.

HasMolecule

```
bool HasMolecule() const
```

Returns *True* if the query currently contains any molecule.

HasShape

```
bool HasShape() const
```

Returns *True* if the query currently contains any molecule, shape grid, or shape gaussians.

HasShapeGrid

```
bool HasShapeGrid() const
```

Returns *True* if the query currently contains any shape grid.

IsEmpty

```
bool IsEmpty() const
```

Returns *True* if the query currently contains no shape or color.

SetTitle

```
bool SetTitle(const char *)
bool SetTitle(const std::string&)
```

Sets the title of the query.

3.1.131 OESubrocsStarts

```
class OESubrocsStarts : public OEInertialStarts
```

The *OESubrocsStarts* represents the Subrocs starts for overlay optimization.

The following methods are publicly inherited from *OEStarts*:

- *CreateCopy*
- *GetNumOfStarts*
- *GetStarts*
- *Setup*
- *SetupRef*
- *SetupRef*

The following methods are publicly inherited from *OEInertialStarts*:

- *GetSymmetryThreshold*
- *SetSymmetryThreshold*

Constructors

```
OESubrocsStarts()
OESubrocsStarts(const OESubrocsStarts&)
```

Default and copy constructors.

3.1.132 OETanimotoComboCutoff

```
class OETanimotoComboCutoff : public OEOverlayScoreCutoff
```

The *OETanimotoComboCutoff* defines cutoff based on the combined shape and color tanimoto.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OETanimotoComboCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.1.133 OETanimotoCutoff

```
class OETanimotoCutoff : public OEOverlayScoreCutoff
```

The *OETanimotoCutoff* defines cutoff based on the shape tanimoto.

The following methods are publicly inherited from OEUnaryPredicate:

- CreatePredicateCopy

The following methods are publicly inherited from *OEOverlayScoreCutoff*:

- *operator()*
- *CreateCopy*

Constructors

```
OETanimotoCutoff(const double)
```

Default and copy constructors. The argument expects the desired cutoff.

3.2 OEShape Constants

3.2.1 OEColorFFType

This namespace contains constants defining color force fields.

Undefined

OEDefault

The default approximation is ImplicitMillsDean

ImplicitMillsDean

The `ImplicitMillsDean` color forcefield.

ExplicitMillsDean

The `ExplicitMillsDean` color forcefield.

ImplicitMillsDeanNoRings

The `ImplicitMillsDean` color forcefield without rings.

ExplicitMillsDeanNoRings

The `ExplicitMillsDean` color forcefield without rings.

Custom

Custom color forcefield.

Max

The largest numerical value that can be contained in this namespace. Useful for being able to iterate over all the constants in this namespace.

3.2.2 OEDerivativeType

This namespace controls how to extract derivatives from a shape grid, associated with the *Grid* based overlap calculation.

Interpolated

Derivatives obtained from interpolation.

NearestValue

Derivatives of the nearest grid point are picked.

3.2.3 OEExponentialType

This namespace controls how to obtain exponentials during calculation of Gaussian overlaps and their derivatives.

Standard

Evaluate the exponentials with standard methods every time.

Tabulated

Interpolate from tabulated values.

3.2.4 OEMomentSymmetry

This namespace defines constants corresponding to various possible Moments symmetry.

NoSymmetry

No symmetry around any of the inertials axes

Oblate

Oblate symmetry.

Prolate

Prolate symmetry.

Spherical

Spherical symmetry.

Unknown

Unknown symmetry.

3.2.5 OEOverlapRadii

This namespace controls how radii of atoms should be handled by the **Shape TK** when doing calculations. The default value of using the carbon radii approximation, *OEOverlapRadii.Carbon*, was found to have the best performance in virtual screening experiments.

Carbon

Treat all atoms, as carbon, with the same radius.

All

Use the radius of the atom as returned by `OEAtomBase.GetRadius`.

Default

The default approximation is *`OEOverlapRadii.Carbon`*, to treat all atoms as carbon.

Max

The largest numerical value that can be contained in this namespace. Useful for being able to iterate over all the constants in this namespace.

3.2.6 OEScoreType

This namespace controls the type of score to be returned by the operator() methods of a *OEOverlapFuncBase* derived instance.

Overlap

Calculate overlap

Tanimoto

Calculate tanimoto

3.2.7 OEShapeType

This namespace contains constants that defines shape type appropriate for overlap calculations.

Grid

Calculates shape overlap using *OEGridShapeFunc*

Exact

Calculates shape overlap using *OEExactShapeFunc*

Analytic

Calculates shape overlap using *OEAnalyticShapeFunc*

Hermite

Calculates shape overlap using *OEHermiteShapeFunc*

None

Ignore shape overlap calculation

3.2.8 OEColorType

This namespace contains constants that defines color type appropriate for color scores calculations.

Grid

Calculates color scores using *OEGridColorFunc*

Exact

Calculates color scores using *OEExactColorFunc*

Analytic

Calculates color scores using *OEAnalyticColorFunc*

None

Ignore color scores calculation

3.3 OEShape Functions

3.3.1 OEAddColorAtom

```
OEChem::OEAtomBase* OEAddColorAtom(OEChem::OEMolBase& mol,
                                   const float* coords, unsigned type,
                                   const std::string& typeName,
                                   const std::string& parentName = std::string())
OEChem::OEAtomBase* OEAddColorAtom(OEChem::OEMolBase& mol,
                                   const OEChem::OEAtomBase** parents,
                                   unsigned num, unsigned type,
                                   const std::string& typeName)
OEChem::OEAtomBase* OEAddColorAtom(OEChem::OEMCMolBase& mol,
                                   const OEChem::OEAtomBase** parents,
                                   unsigned num, unsigned type,
                                   const std::string& typeName)
```

These functions allow adding color atoms manually, as opposed to being assigned via SMARTS matches from a color force field. Note that the color atom will either be placed at the coordinates supplied in the first version, or at the center of mass of the *parents* atoms. num is the number of atoms in the parent array.

3.3.2 OEAddColorAtoms

```
unsigned int OEAddColorAtoms(OEChem::OEMolBase& mol,
                             const OEColorForceField& cff,
                             bool setCachedSelfColor = false)
unsigned int OEAddColorAtoms(OEChem::OEMCMolBase& mol,
                             const OEColorForceField& cff,
                             bool setCachedSelfColor = false)
```

Add color atoms to the passed in molecule. Note that this is not normally required as both OEColorOverlap and OEBestOverlay make internal copies of molecules and add color atoms automatically, but if you continually re-use the same molecule, this can be an optimization. Additionally, setting *setCachedSelfColor* to true will pre-calculate and cache the self color on the molecule. This cached value will then be used by OEColorOverlap and OEBestOverlay. The value can be accessed via *OEGetCachedSelfColor*.

3.3.3 OEAddCoordsToColorAtom

```
bool OEAddCoordsToColorAtom(OEChem::OEMolBase& mol,
                            OEChem::OEAtomBase* coloratom, float* coords)
```

Set the 3D coordinates of the given color atom.

3.3.4 OECalcShapeMultipoles

```
bool OECalcShapeMultipoles(const OEChem::OEMolBase& mol, float* smult,
                           unsigned int type = OEOverlapRepresentation::Atoms,
                           bool useHydrogens = false)
```

Calculate the shape (steric) multipoles of a molecule.

The molecule must have valid 3D coordinates assigned. Multipole coefficients are written to the caller-supplied array *smult*. The required array size depends on the *type* parameter: for *OEOverlapRepresentation.Atoms* the array must be large enough to hold one entry per atom.

mol

The molecule to calculate multipoles for.

smult

Output array that receives the computed multipole coefficients.

type

The overlap representation to use, chosen from the *OEOverlapRepresentation* namespace.

useHydrogens

If true, hydrogen atoms are included. The default is false.

Returns true on success.

3.3.5 OECalcVolume

```
float OECalcVolume(const OEChem::OEMolBase& mol,
                   bool useHydrogens = true)
```

Calculate the Gaussian shape volume of a molecule.

The volume is computed as the first-order Gaussian overlap integral over all atoms in the molecule. The molecule must have valid 3D coordinates assigned.

mol

The molecule to calculate the volume for.

useHydrogens

If true (the default), hydrogen atoms are included in the volume calculation. If false, only heavy atoms contribute.

Returns the Gaussian shape volume in cubic angstroms.

3.3.6 OEClearCachedSelfColor

```
void OEClearCachedSelfColor(OEChem::OEMCMolBase& mol)
void OEClearCachedSelfColor(OEChem::OEMolBase& mol)
```

Clear cached self color from molecule. See *OESetCachedSelfColor* for more info.

3.3.7 OEClearCachedSelfShape

```
void OEClearCachedSelfShape(OEChem::OEMCMolBase& mol)
void OEClearCachedSelfShape(OEChem::OEMolBase& mol)
```

Clear cached self shape from molecule. See *OESetCachedSelfShape* for more info.

3.3.8 OECompressColorAtoms

```
unsigned int OECompressColorAtoms(OEChem::OEMolBase& mol)
```

Convert the color atoms on a molecule to an internal compressed form. Useful for saving space when storing pre-colored molecules. Use *OEUncompressColorAtoms* to restore the actual color atoms.

3.3.9 OECountColorAtoms

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

Return the number of color atoms attached to a molecule.

3.3.10 OEDeleteCompressedColorAtoms

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

Removes information created from the *OECompressColorAtoms* function.

3.3.11 OEGetCachedSelfColor

```
float OEGetCachedSelfColor(const OEChem::OEConfBase& mol,
                           bool allcolor = false)
float OEGetCachedSelfColor(const OEChem::OEMolBase& mol,
                           bool allcolor = false)
```

Return the value of cached self color on the molecule or conformer. This function doesn't work for OEMCMolBase directly, so call it on each individual conformer. Returns 0.0 if there is no cached value. Use *OEHasCachedSelfColor* to determine if a real value is attached.

3.3.12 OEGetCachedSelfShape

```
float OEGetCachedSelfShape(const OEChem::OEConfBase& conf,
                           unsigned int radiiApprox = OEOverlapRadii::Carbon,
                           float cradius = 1.7f)
float OEGetCachedSelfShape(const OEChem::OEMolBase& mol,
                           unsigned int radiiApprox = OEOverlapRadii::Carbon,
                           float cradius = 1.7f)
```

Return the value of cached self shape on the molecule or conformer. The value for radii approximation and cradius need to match those used to set this value initially. This function doesn't work for OEMCMolBase directly, so call it on each individual conformer. Returns 0.0 if there is no cached value. Use *OEHasCachedSelfShape* to determine if a real value is attached.

3.3.13 OEGetCenterAndExtents

```
void OEGetCenterAndExtents(const OEShapeQueryPublic& sq, float* ctr,
                           float* ext)
```

Fills the first three values of *ctr* with the location of the center of *sq* and fills the first three values of *ext* with the extents.

3.3.14 OEGetColorAtoms

```
OESystem::OEIterBase<OEChem::OEAtomBase>* OEGetColorAtoms(OEChem::OEMolBase& mol)
OESystem::OEIterBase<const OEChem::OEAtomBase>* OEGetColorAtoms(const OEChem::OEMolBase&␣
↪mol)
```

Return an iterator of all the color atoms attached to a molecule.

3.3.15 OEGetColorParents

```
OESystem::OEIterBase<OEChem::OEAtomBase>* OEGetColorParents(OEChem::OEAtomBase* atom)
OESystem::OEIterBase<const OEChem::OEAtomBase>* OEGetColorParents(const␣
↪OEChem::OEAtomBase* atom)
```

Color atoms set via a SMARTS match store the matching atoms as *parents*. This method returns the parent atoms of a given color atom.

3.3.16 OEGetColorType

```
unsigned int OEGetColorType(const OEChem::OEAtomBase* coloratom)
```

Returns the type of a color atom as an unsigned int.

3.3.17 OEGetMomentsOfInertiaTransform

```
unsigned int OEGetMomentsOfInertiaTransform(OEChem::OETrans& argTransform,
                                            const OEChem::OEMolBase & argMol,
                                            const double argThreshold = 0.15)
unsigned int OEGetMomentsOfInertiaTransform(OEChem::OETrans& argTransform,
                                            const OESystem::OEScalarGrid& argGrid,
                                            const double argThreshold = 0.15)
unsigned int OEGetMomentsOfInertiaTransform(OEChem::OETrans& argTransform,
                                            const OEShapeQuery& argQuery,
                                            const double argThreshold = 0.15)
```

Calculate the transform required to move the object (the specified molecule, grid or the shape query) to the center of its inertial frame. Method returns the moments symmetry of the object defined in the *OEMomentSymmetry* namespace. The required transform is returned in the first argument.

3.3.18 OEHasCachedSelfColor

```
bool OEHasCachedSelfColor(const OEChem::OEMCMolBase& mol)
bool OEHasCachedSelfColor(const OEChem::OEMolBase& mol)
```

Returns true if the molecule has cached self color.

3.3.19 OEHasCachedSelfShape

```
bool OEHasCachedSelfShape(const OEChem::OEMCMolBase& mcmol,
                          unsigned int radiiApprox = OEOverlapRadii::Carbon,
                          float cradius = 1.7f)
bool OEHasCachedSelfShape(const OEChem::OEMolBase& mol,
                          unsigned int radiiApprox = OEOverlapRadii::Carbon,
                          float cradius = 1.7f)
```

Returns true if the molecule has cached self shape. The value for radii approximation and cradius need to match those used to set this value initially.

3.3.20 OEHasColorAtoms

```
bool OEHasColorAtoms(const OEChem::OEMolBase& mol)
```

Returns `true` if the molecule has any color atoms attached. Note, *OEHasColorAtoms* only checks for the existence of color atoms on the molecule, even if *OEAddColorAtoms* did not match any substructures to create color atoms. Use *OEHasPerceivedColorAtoms* if trying to determine whether *OEAddColorAtoms* still needs to be called on the molecule.

3.3.21 OEHasCompressedColorAtoms

```
bool OEHasCompressedColorAtoms(const OEChem::OEMolBase& mol)
```

Returns `true` if the molecule contains compressed color atom information created from the *OECompressColorAtoms* function.

3.3.22 OEHasPerceivedColorAtoms

```
bool OEHasPerceivedColorAtoms(const OEChem::OEMolBase& mol)
```

Returns `true` if *OEAddColorAtoms* has already been called on `mol`. Note, *OEHasColorAtoms* only checks for the existence of color atoms on the molecule, even if *OEAddColorAtoms* did not match any substructures to create color atoms.

3.3.23 OEIsColorAtom

```
bool OEIsColorAtom(const OEChem::OEAtomBase* atom)
```

Returns `true` if the given atom is a color atom.

3.3.24 OEIsFastROCSShapeQuery

```
bool OEIsFastROCSShapeQuery(const OEShapeQueryPublic& query)
bool OEIsFastROCSShapeQuery(const OEShapeQuery& query)
```

Determines if a shape query object is compatible with the *OEShapeQuery* or a *OEShapeQueryPublic* overload of the `OEShapeDatabase.GetScores` and `OEShapeDatabase.GetSortedScores` in **FastROCS TK**.

3.3.25 OEMol2Query

```
bool OEMol2Query(OEShape::OEShapeQuery& query,
                 const OEChem::OEMolBase& mol,
                 const OEShape::OEColorForceField& cff)
```

Creates a *OEShapeQuery* from the input OEMolBase. The molecule must have valid 3D coordinates assigned. The specified *OEColorForceField* is used to generate color atoms during the query generation process.

Returns true on successful query generation.

3.3.26 OEOrientByMomentsOfInertia

```
unsigned int OEOrientByMomentsOfInertia(OEChem::OEMolBase & argMol,
                                        OEChem::OETrans& argTransform,
                                        const double argThreshold = 0.15)
unsigned int OEOrientByMomentsOfInertia(OESystem::OEScalarGrid& argGrid,
                                        OEChem::OETrans& argTransform,
                                        const double argThreshold = 0.15)
unsigned int OEOrientByMomentsOfInertia(OEShapeQuery& argQuery,
                                        OEChem::OETrans& argTransform,
                                        const double argThreshold = 0.15)
```

Calculate the transform required to move the object (the specified molecule, grid or the shape query) to the center of its inertial frame, and apply the transform to move the object to the inertial frame. Method returns the moments symmetry

of the object defined in the *OEMomentSymmetry* namespace. The transform required to get the object back into the original frame is returned in the second argument of the method.

3.3.27 OEReadShapeQuery

```
bool OEReadShapeQuery(const char* filename, OEShapeQuery& query)
bool OEReadShapeQuery(const char* filename, OEShapeQueryPublic& query)
bool OEReadShapeQuery(OEPlatform::oeistream& ifs, OEShapeQuery& query)
bool OEReadShapeQuery(OEPlatform::oeistream& ifs,
                      OEShapeQueryPublic& query)
```

The first two overloads reads from a file given in the first parameter into the shape query given in the second parameter. The later two overloads will read from the stream given in the first parameter into the shape query given in the second parameter.

3.3.28 OERemoveColorAtoms

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

Remove all color atoms from the molecule.

3.3.29 OEROCSOverlay

```
bool OEROCSOverlay(OEROCSResult& argResult,
                   const OEChem::OEMCMolBase& argQuery,
                   const OEChem::OEMCMolBase& argFitMol,
                   const OEROCSOptions& argOptions = OEROCSOptions())
bool OEROCSOverlay(OEROCSResult& argResult,
                   const OEChem::OEMolBase& argQuery,
                   const OEChem::OEMCMolBase& argFitMol,
                   const OEROCSOptions& argOptions = OEROCSOptions())
bool OEROCSOverlay(OEROCSResult& argResult,
                   const OEChem::OEMolBase& argQuery,
                   const OEChem::OEMolBase& argFitMol,
                   const OEROCSOptions& argOptions = OEROCSOptions())
```

Calculate the overlay between the query and the fit molecule. Returns *True* if calculation completed successfully. The first argument returns the results of the single best hit.

> **Note**
>
> This function sets up the reference system in *OEROCS* every time it is called, even if it is called with the same reference object. This can have unexpected performance issues if the function is used in a loop.

3.3.30 OESelfColor

```
float OESelfColor(const OEChem::OEMolBase& mol,
                  const OEColorForceField& cff, bool allcolor = false)
```

Calculate the self color score for the given molecule, using the passed in color force field.

3.3.31 OESelfShape

```
float OESelfShape(const OEChem::OEMolBase& mol,
                  unsigned int radiiApprox = OEOverlapRadii::Carbon,
                  float cradius = 1.7f)
```

Return the self-volume term that can be used in molecular similarity measures. The volume is calculated as the first-order-gaussian overlap of all the atoms in the molecule to itself.

mol

The molecule to calculate the self-volume for. This molecule is expected to have valid 3D coordinates.

radiiApprox

The approximation to use when calculating the volume. The constant should be pulled from the *OEOverlapRadii* namespace. The default is to approximate all atom types as carbon with the same radius. If *OEOverlapRadii.All* is given, the radius of the atom as returned by `OEAtomBase.GetRadius` will be used.

cradius

The radius to use when the carbon radii approximation is being used. The default of 1.7 is based upon the original [Grant-1995] paper to reproduce hard sphere volumes.

See also

- *Molecular Shape*

3.3.32 OESetCachedSelfColor

```
bool OESetCachedSelfColor(OEChem::OEMCMolBase& mol,
                          const OEColorForceField& cff)
bool OESetCachedSelfColor(OEChem::OEMolBase& mol,
                          const OEColorForceField& cff)
```

Using the passed-in OEColorForceField, calculate and store the self color on each molecule. For the OEMCMolBase, each conformer gets its own stored value. To retrieve the value, use *OEGetCachedSelfColor*. OEColorOverlap and OEBestOverlay will use the cached value instead of re-calculating the self terms.

3.3.33 OESetCachedSelfShape

```
bool OESetCachedSelfShape(OEChem::OEMCMolBase& mcmol,
                          unsigned int radiiApprox = OEOverlapRadii::Carbon,
                          float cradius = 1.7f)
bool OESetCachedSelfShape(OEChem::OEMolBase& mol,
                          unsigned int radiiApprox = OEOverlapRadii::Carbon,
                          float cradius = 1.7f)
```

Store the self shape term on the molecule. For the OEMCMolBase, each conformer stores its own value. To retrieve the value, use *OEGetCachedSelfShape*. OEOverlap and OEBestOverlay will use the cached value instead of re-calculating the self terms.

3.3.34 OESetColorParents

```
bool OESetColorParents(OEChem::OEAtomBase* atom,
                       const OEChem::OEAtomBase** parents,
                       unsigned num)
```

Set the parent atoms of a given color atom. Not normally used as parents are set for color atoms added via a SMARTS match in the color force field. But if you are manually creating color atoms, this is a way to set its parents.

3.3.35 OESetColorType

```
bool OESetColorType(OEChem::OEAtomBase* coloratom, unsigned type,
                    const std::string& typeName)
Set/changes the type of a given color atom.
```

3.3.36 OESetCoordsFromColorParents

```
bool OESetCoordsFromColorParents(OEChem::OEMCMolBase& mol)
bool OESetCoordsFromColorParents(OEChem::OEMolBase& mol)
```

Generates 3D coordinates for the given color atom based on the center of mass of that color atom's parents.

3.3.37 OEShapeGetArch

```
const char* OEShapeGetArch()
```

Returns the architecture of the current version of the Shape toolkit. Examples include:

- microsoft-win32-x86
- redhat-RHEL5-x86_64

3.3.38 OEShapeGetLicensee

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

Returns the LICENSEE from the current valid Shape TK license.

3.3.39 OEShapeGetPlatform

```
const char* OEShapeGetPlatform()
```

Returns the platform, including compiler version, of the current version of the Shape toolkit. Examples include:

- microsoft-win32-msvc9-MD-x86
- redhat-RHEL5-g++4.1-x86_64

3.3.40 OEShapeGetRelease

```
const char* OEShapeGetRelease()
```

Returns the version of this toolkit release. For example: `1.6.1`.

3.3.41 OEShapeGetSite

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

Returns the SITE from the current valid Shape TK license.

3.3.42 OEShapeGetVersion

```
unsigned int OEShapeGetVersion()
```

Returns the build date of the toolkit as an unsigned int. For example: `20090227`.

3.3.43 OEShapeIsLicensed

```
bool OEShapeIsLicensed(const char* feature = 0,
                       unsigned int* expdate = 0)
```

Determines whether a valid license file is present. This function may be called without a legitimate run-time license to determine whether it is safe to call any **Shape TK** functionality.

The 'feature' argument can be used to check for a valid license to **Shape TK** along with the following features: 'python', 'java', or 'clr' (for CSharp).

The 'expdate' argument can be used to obtain the expiration date of the current **Shape TK** license.

See also

- `OEChemIsLicensed` function

3.3.44 OESortOverlayScores

```
void OESortOverlayScores(OESystem::OEIter<OEBestOverlayScore>& dst,
                         OESystem::OEIter<OEBestOverlayResults>& scores,
                         const OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>& sorter,
                         unsigned nbest = 1, bool conforder = false)
```

Sort the scores from multiple sets of *OEBestOverlayResults* into a single iterator of *OEBestOverlayScore*. The sorting function can be one of the pre-defined functors or a user-defined version.

The `nbest` default value of **1** implies that on the best overlay for each ref-fit conformer pair is returned. A value greater than one will mean multiple overlays for each ref-fit conformer pair will be kept.

Setting `conforder` to **true** forces the results to come out in conformer index order. If `nbest` is greater than **1** and `conforder` is **true**, then for each ref-fit conformer pair, the scores will be sorted by the sorting function, but each set of ref-fit results will come out together.

Several of the OEBestOverlay examples show variations of these values. Perhaps the easiest way to understand them is to modify one of the examples and observe the change in the number and order of the scores.

3.3.45 OEUncompressColorAtoms

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

Convert the internal compressed color atoms to actual color atoms. Has no effect unless the color atoms were previously compressed by *OECompressColorAtoms*

> **Note**
>
> This function was formerly named OEStringToColorAtoms.

3.3.46 OEWriteShapeQuery

```
bool OEWriteShapeQuery(const char* filename,
                       const OEShapeQueryBase& query)
bool OEWriteShapeQuery(OEPlatform::oeostream& ofs,
                       const OEShapeQueryBase& query)
```

Writes a query given in the second parameter. The first overload writes the query in the specified file and the second overload writes it to a oeostream. Method returns `True` if write is successful.

CHAPTER

FOUR

PRELIMINARY API

4.1 Preliminary OEShape Classes

4.1.1 OEShapeQueryOptions

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

```
class OEShapeQueryOptions : public OESystem::OEOptions
```

This class provides an interface to setup options required for *OEShapeQuery* class.

The *OEShapeQueryOptions* class defines the following public methods:

- *GetRelativeGridContourVal*
- *GetShapeGaussianRadius*
- *GetColorGaussianRadius*
- *GetScoreType*
- *SetRelativeGridContourVal*
- *SetShapeGaussianRadius*
- *SetColorGaussianRadius*
- *SetScoreType*

Constructors

```
OEShapeQueryOptions()
OEShapeQueryOptions(const OEShapeQueryOptions&)
```

Default and copy constructors.

Assignment operator.

GetRelativeGridContourVal

```
double GetRelativeGridContourVal() const
```

Returns the maximum relative (to maximum grid value of the input grid) grid contour value allowed for the Gaussians in the final result.

GetShapeGaussianRadius

```
double GetShapeGaussianRadius() const
```

Returns the radius of shape Gaussians used in the grid-to-Gaussians fit. The number 1.70 is the default for the carbon radius.

GetColorGaussianRadius

```
double GetColorGaussianRadius() const
```

Returns the radius of color Gaussians used in the grid-to-Gaussians fit. The number 1.08 corresponds to the color Gaussian radius and is the default.

GetScoreType

```
unsigned GetScoreType() const
```

Returns the grid score type via the namespace *OEGridScoreType*, with possible values *OEGridScoreType.CosAngle* and *OEGridScoreType.Tanimoto*.

SetRelativeGridContourVal

```
bool SetRelativeGridContourVal(double)
```

Sets the maximum relative (to maximum grid value of the input grid) grid contour value allowed for the Gaussians in the final result.

SetShapeGaussianRadius

```
bool SetShapeGaussianRadius(double)
```

Sets the radius of shape Gaussians used in the grid-to-Gaussians fit.

SetColorGaussianRadius

```
bool SetColorGaussianRadius(double)
```

Sets the radius of color Gaussians used in the grid-to-Gaussians fit.

SetScoreType

```
bool SetScoreType(unsigned)
```

Sets the grid score type via the namespace *OEGridScoreType*, with possible values *OEGridScoreType.CosAngle* and *OEGridScoreType.Tanimoto*.

4.1.2 OEHermiteOverlayOptions

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

```
class OEHermiteOverlayOptions : public OESystem::OEOptions
```

This class provides an interface to set up options required for *OEHermiteOverlay*.

The *OEHermiteOverlayOptions* class defines the following public methods:

- *GetHermiteOptions* and *SetHermiteOptions*
- *GetMaxOptSteps* and *SetMaxOptSteps*
- *GetStarts* and *SetStarts*

Constructors

```
OEHermiteOverlayOptions()
OEHermiteOverlayOptions(const OEHermiteOverlayOptions&)
```

Default and copy constructors.

Assignment operator.

4.1.3 GetHermiteOptions

```
OEHermiteOptions& GetHermiteOptions()
const OEHermiteOptions& GetHermiteOptions() const
```

See the *SetHermiteOptions* method.

4.1.4 GetMaxOptSteps

```
unsigned GetMaxOptSteps() const
```

See the *SetMaxOptSteps* method.

4.1.5 GetStarts

```
const OEStarts& GetStarts() const
```

See the *SetStarts* method.

4.1.6 SetHermiteOptions

```
void SetHermiteOptions(const OEHermiteOptions&)
```

Sets the Hermite parameters that control the *Hermite expansion* of reference and fit objects via the *OEHermiteOptions* class.

4.1.7 SetMaxOptSteps

```
bool SetMaxOptSteps(const unsigned int)
```

Sets the maximum number of optimization iteration steps. Default: *200*.

4.1.8 SetStarts

```
void SetStarts(const OEStarts&)
```

Set the starts by passing in an *OEStarts* instance. This defines the starts to be used for optimization.

4.1.9 OEHermiteOverlay

Attention

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

```
class OEHermiteOverlay
```

The *OEHermiteOverlay* defines an interface for overlap optimization between a reference object (a molecule, grid, shape query, or Hermite) and a fit object (molecule conformers, grid, or Hermite).

Note

Currently *OEHermiteOverlay* optimizes shape only.

The *OEHermiteOverlay* class defines the following public methods:

- *BestOverlay*
- *SetupRef*

Constructors

```
OEHermiteOverlay(const OEHermiteOverlayOptions& hermOverlayOpts =␣
↪OEHermiteOverlayOptions())
OEHermiteOverlay(const OEHermiteOverlay&)
```

Default and copy constructors.

Assignment operator.

4.1.10 BestOverlay

```
bool BestOverlay(OEBestOverlayScore& score,
                 const OEHermite& fitherm,
                 const OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>& sorter = OEHighestTanimoto())
bool BestOverlay(OEBestOverlayScore& score,
                 const OEChem::OEMCMolBase& fitmol,
                 const OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
```

(continues on next page)

(continued from previous page)

```
↪OEBestOverlayScore>& sorter = OEHighestTanimoto())
bool BestOverlay(OEBestOverlayScore& score,
                 const OESystem::OEScalarGrid& fitgrid,
                 const OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>& sorter = OEHighestTanimoto())
```

This method optimizes the overlay between the reference object and the fit object (molecule conformers, grid, or Hermite), and returns the best result, sorted based on the binary predicate `sorter`. The predicate has a default setting of *OEHighestTanimoto*. The reference object must be set using the *SetupRef* method prior to calling this method.

4.1.11 SetupRef

```
bool SetupRef(const OEHermite& refherm)
bool SetupRef(const OEChem::OEMolBase& refmol)
bool SetupRef(const OESystem::OEScalarGrid& refgrid)
bool SetupRef(const OEShapeQuery& query)
```

This method defines the interface for setting up the reference system for the *OEHermiteOverlay* instance with a molecule, grid, shape query, or Hermite object.

4.2 Preliminary OEShape Constants

4.2.1 OEGridScoreType

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

This namespace controls the type of score to be used in the fit and returned by the *AddShapeGaussians* and *AddColorGaussians* methods of *OEShapeQuery*. This is processed via the options class *OEShapeQueryOptions*.

CosAngle

Maximizes the cosine angle between two vectorized grids: the first from the input grid and the second from the one that follows from fitted Gaussians. This measure is insensitive to overall normalization of the input grid.

Tanimoto

Maximizes the Tanimoto between the input grid and the grid that follows from fitted Gaussians. This measure depends on self-overlaps of the input grid and the grid that follows from fitted Gaussians. Consequently, it is sensitive to the overall coefficient of the input grid.

CHAPTER

FIVE

DEPRECATED API

5.1 Deprecated OEShape Classes

5.1.1 OEBestOverlay

```
class OEBestOverlay
```

> **Note**
>
> This deprecated class will not be developed further. It is being replaced by *OEOverlay* and *OEROCS* which provides more control and greater expandability.

This class is used to optimize the overlap between 2 molecules. Both molecules can contain one or more conformers. The reference molecule is held rigid and the fit molecule orientation is calculated to maximize overlap. Each conformer of the fit molecule is optimized against each conformer of the reference molecule, resulting in a large number of results that can be returned. The *OEBestOverlayResults* and *OEBestOverlayScore* classes are used to make handling this large amount of results easier.

Additionally, the reference or fit shape can be a grid instead of a molecule.

Since an OEBestOverlay optimization can result in numerous final results, one for each starting position, the *OEBestOverlayResults* class is a container to hold all the results for a fit conformer against a reference conformer. This class is essentially a container for one or more *OEBestOverlayScore* instances. The actual number of scores depends on the *OEBOOrientation* used in the OEBestOverlay.

For *OEBOOrientation.Inertial*, there will be 4, 8 or 20 scores per ref-fit conformer pair, depending on the symmetry of each conformer. For *OEBOOrientation.AsIs*, there will be just 1 *OEBestOverlayScore* inside each *OEBestOverlayResults* instance. And for *OEBOOrientation.Random*, then there will be 1 OEBestOverlayScore for each of the **N** random starts.

When dealing with the results from OEBestOverlay, the user can either use a double loop, *i.e.* loop over all the OEBestOverlayResults instances for each ref-fit conformer pair then loop over each OEBestOverlayScore inside each OEBestOverlayResults instance. In this case, the results also come out in conformer order.

```
OEBestOverlayResultsIter resiter = best.Overlay(fitmol);
for (OEBestOverlayResults res : best.Overlay(fitmol)) {
    for (OEBestOverlayScore score : res.GetScores()) {
        // do something with score
    }
}
```

Or, there is a free function *OESortOverlayScores*, that takes an iterator of OEBestOverlayResults and returns a single, sorted iterator of OEBestOverlayScores that can be used in a single loop.

```
OEBestOverlayResultsIter resiter = best.Overlay(fitmol);
OEBestOverlayScoreIter scoreiter = new OEBestOverlayScoreIter();
oeshape.OESortOverlayScores(scoreiter, resiter, new OEHighestTanimotoCombo());
for (OEBestOverlayScore score : scoreiter) {
    // do something with score
}
```

Constructors

```
OEBestOverlay()
OEBestOverlay(const OEBestOverlay &)
OEBestOverlay(const OEChem::OEMCMolBase& refmol)
OEBestOverlay(const OESystem::OEScalarGrid& refgrid,
              float interpolate = 0.5f)
```

An empty OEBestOverlay class instance can be created, or a new instance can take the reference molecule or grid as an argument. Note that by default, grids that have a resolution larger that 0.5 Å will be interpolated to that resolution.

ClearColorForceField

```
void ClearColorForceField()
```

Clear out any color force field. No color scores will be calculated.

ClearUserStarts

```
void ClearUserStarts()
```

Set the number of User Starts back to zero.

GetCarbonRadius

```
float GetCarbonRadius() const
```

Return the current value for the carbon radius approximation.

GetInitialOrientation

```
unsigned int GetInitialOrientation() const
```

Get the current value for initial orientation. Possible values are from the *OEBOOrientation* namespace.

GetMaxRandomTranslation

```
float GetMaxRandomTranslation() const
```

Get the value in Å of maximum random translation.

GetMethod

```
unsigned int GetMethod() const
```

Return the current value of overlap method.

GetMinimizeType

```
unsigned int GetMinimizeType() const
```

Get the current value for minimize type. Possible values are from the *OEBOMinType* namespace.

GetNumRandomStarts

```
unsigned int GetNumRandomStarts() const
```

Get the current number of random starts.

GetNumUserStarts

```
unsigned int GetNumUserStarts() const
```

Get the number of user starts currently set. Note that these are not actually used unless *OEBOOrientation.UserInertialStarts* is also set.

GetRadiiApproximation

```
unsigned int GetRadiiApproximation() const
```

Return the current value of the radii approximation.

GetRandomSeed

```
unsigned int GetRandomSeed() const
```

Get the current value of the random seed.

GetRefGrid

```
const OESystem::OEScalarGrid* GetRefGrid() const
```

GetRefMol

```
const OEChem::OEMCMolBase* GetRefMol() const
```

GetRefSelfColor

```
float GetRefSelfColor()
```

Return the self color score of the reference molecule.

GetRefSymmetry

```
unsigned int GetRefSymmetry(unsigned int confIdx) const
```

Return the current representation level.

GetSymmetryThreshold

```
float GetSymmetryThreshold() const
```

GetUseHydrogens

```
bool GetUseHydrogens() const
```

Return the status of hydrogen use in OEBestOverlay.

GetUserStarts

```
bool GetUserStarts(float* xyz) const
```

Extract the coordinates of the user starts. `xyz` should be sized to 3 * GetNumUserStarts().

Overlay

```
OESystem::OEIterBase<OEBestOverlayResults>* Overlay(const OEChem::OEMCMolBase& fitmol)
```

Perform the calculation and return the an iterator of the results.

SetCarbonRadius

```
void SetCarbonRadius(float cradius)
```

Set the radius to use when using *OEOverlapRadii.Carbon*. By default this is set to 1.7 Å.

SetColorForceField

```
bool SetColorForceField(unsigned int type)
bool SetColorForceField(OEPlatform::oeistream& is)
bool SetColorForceField(const std::string& filename)
bool SetColorForceField(const OEColorForceField& cff)
```

Set the color force field to be used. Once set, color scores will be calculated and included in the results. Color gradients will not be included in the optimization unless specifically set using SetColorOptimize(true).

SetColorOptimize

```
void SetColorOptimize(bool state)
```

Add color gradients to shape gradients in the optimization. Has no effect unless a color force field is also set via *OEBestOverlay.SetColorForceField*.

SetInitialOrientation

```
void SetInitialOrientation(unsigned int orient)
```

Determines the initial orientation (starting position) for each optimization. The default is *OEBOOrientation.Inertial*. Alternatives are defined in the *OEBOOrientation* namespace.

SetMaxRandomTranslation

```
void SetMaxRandomTranslation(float trans)
```

If using random starts, this set the maximum distance (in Å) that the center of mass of the fit molecule will be moved away from the center calculated for inertial frame alignment.

SetMethod

```
void SetMethod(unsigned int method)
```

Set the method used to calculate overlap. The default for OEBestOverlay is *OEOverlapMethod.Grid*. Alternatives are defined in the *OEOverlapMethod* namespace.

SetMinimizeType

```
void SetMinimizeType(unsigned int type)
```

Set the score to use in the optimization. Options are in the *OEBOMinType* namespace.

SetNumRandomStarts

```
void SetNumRandomStarts(unsigned int n)
```

If SetInitialOrientation is set to *OEBOOrientation.Random*, this method sets the number of random starting positions that will be used.

SetRadiiApproximation

```
void SetRadiiApproximation(unsigned int type)
```

Set the radius approximation used to calculate overlap. The default for OEBestOverlay is *OEOverlapRadii.Carbon*. Alternatives are defined in the *OEOverlapRadii* namespace.

SetRandomSeed

```
void SetRandomSeed(unsigned int seed)
```

Set a random seed value to allow reproducible random searches.

SetRefGrid

```
bool SetRefGrid(const OESystem::OEScalarGrid& refgrid,
                float interpolate = 0.5f)
```

Set a reference grid for the calculation. OEBestOverlay makes an internal copy. Pre-existing reference molecules or grids are replaced.

SetRefMol

```
bool SetRefMol(const OEChem::OEMCMolBase& refmol)
```

Set a reference molecule for the calculation. OEBestOverlay makes an internal copy. Pre-existing reference molecules or grids are replaced.

SetSymmetryThreshold

```
void SetSymmetryThreshold(float threshold)
```

SetUseHydrogens

```
void SetUseHydrogens(bool state)
```

Boolean to determine whether hydrogens are included in the shape calculation. By default this is false and hydrogens are ignored.

SetUserStarts

```
void SetUserStarts(const float* xyz, unsigned int nstarts)
```

Set a set of 3D coordinates for doing separate inertial starts. `xyz` should be 3 * nstarts. These are not used unless *OEBOOrientation.UserInertialStarts* is also set.

5.1.2 OEColorOverlap

```
class OEColorOverlap
```

> **Note**
>
> This deprecated class will not be developed further. It is being replaced by derivatives of *OEColorFunc* which provides more control and greater expandability.

Calculates chemical feature color overlap between a reference and a fit molecule.

Constructors

```
OEColorOverlap()
OEColorOverlap(const OEColorOverlap& rhs)
```

Default and copy constructors.

ColorScore

```
bool ColorScore(OEColorResults& res)
bool ColorScore(const OEChem::OEMolBase& fitmol,
                OEColorResults& res)
```

GetAllColor

```
bool GetAllColor() const
```

GetSelfColor

```
float GetSelfColor() const
```

SetAllColor

```
void SetAllColor(bool state)
```

SetColorForceField

```
bool SetColorForceField(unsigned int cffType)
bool SetColorForceField(OEPlatform::oeistream& is)
bool SetColorForceField(const std::string& filename)
bool SetColorForceField(const OEColorForceField& cff)
```

SetFitMol

```
bool SetFitMol(const OEChem::OEMolBase& fitmol)
```

SetRefMol

```
bool SetRefMol(const OEChem::OEMolBase& refmol)
```

5.1.3 OEColorResults

```
struct OEColorResults
```

> **Note**
>
> This deprecated class will not be developed further. It is being replaced by *OEOverlapResults* which provides more control and greater expandability.

Stores the result of a color overlap calculation.

Constructors

```
OEColorResults()
```

Default and copy constructors.

GetFitTversky

```
float GetFitTversky() const
```

GetRefTversky

```
float GetRefTversky() const
```

GetTanimoto

```
float GetTanimoto() const
```

GetTversky

```
float GetTversky(float alpha, float beta) const
```

5.1.4 OEHighestComboScore

```
struct OEHighestComboScore : public OESystem::OEBinaryPredicate<OEBestOverlayScore,␣
↪OEBestOverlayScore>
```

Note

This deprecated class will not be developed further, as it was developed based on a nonstandard score.

Predicate to compare *OEBestOverlayScore* based on a combination of shape Tanimoto and color score.

operator()

```
bool operator()(const OEBestOverlayScore& s1, const OEBestOverlayScore& s2) const
```

CreateCopy

```
OESystem::OEBinaryFunction<OEBestOverlayScore , OEBestOverlayScore , bool>* CreateCopy()␣
↪const
```

5.1.5 OEOverlap

```
class OEOverlap
```

Note

This deprecated class will not be developed further. It is being replaced by derivatives of *OEShapeFunc* which provides more control and greater expandability.

OEOverlap calculates the static shape overlap between a reference molecule or grid and a fit molecule or grid. Note that this does not move the fit molecule(grid) nor does it optimize the overlap. It simply calculates the score for the provided orientation.

Constructors

```
OEOverlap()
OEOverlap(const OEOverlap& rhs)
OEOverlap(const OEChem::OEMolBase& refmol)
OEOverlap(const OESystem::OEScalarGrid& refmol)
```

Default and copy constructors.

GetCarbonRadius

```
float GetCarbonRadius() const
```

Return the current value for the carbon radius approximation.

GetGridSpacing

```
float GetGridSpacing() const
```

Return the current value for the grid spacing to use for the *OEOverlapMethod.Grid* method of calculating overlaps. Is not used by any other overlap method. Defaults to `0.25`.

GetMethod

```
unsigned int GetMethod() const
```

Return the current value of overlap method.

GetRadiiApproximation

```
unsigned int GetRadiiApproximation() const
```

Return the current value of the radii approximation.

GetUseHydrogens

```
bool GetUseHydrogens() const
```

Return the status of hydrogen use in OEOverlap.

Overlap

```
bool Overlap(OEOverlapResults& res, float* atomOverlaps = 0)
bool Overlap(const OEChem::OEMolBase& fitmol,
             OEOverlapResults& res, float* atomOverlaps = 0)
```

Calculate the overlap of the passed in fit molecule or fit grid and place the results into the passed instance of *OEOverlapResults*.

SetCarbonRadius

```
bool SetCarbonRadius(float cradius)
```

Set the radius to use when using *OEOverlapRadii.Carbon*. By default this is set to 1.7 Angstroms. See *OEBestOverlay.GetCarbonRadius*.

SetFitMol

```
bool SetFitMol(const OEChem::OEMolBase& fitmol)
```

Set molecule to be used as fit object.

SetGridSpacing

```
bool SetGridSpacing(float spacing)
```

Set the grid spacing to use for the *OEOverlapMethod.Grid* method of calculating overlaps. Smaller values typically lead to more accurate results when considering large numbers of compounds, though not necessarily more accurate results in individual cases. This method will only accept values between 0.125 and 1.0 inclusive. The default is 0.25 angstroms for a balance of performance and reproducibility. Larger values may lead to better performance on some platforms.

SetMethod

```
bool SetMethod(unsigned int m)
```

Set the method used to calculate overlap. The default for OEOverlap is *OEOverlapMethod.Exact*. Alternatives are defined in the *OEOverlapMethod* namespace.

SetRadiiApproximation

```
bool SetRadiiApproximation(unsigned int type)
```

Set the radius approximation used to calculate overlap. The default for OEOverlap is *OEOverlapRadii.Carbon*. Alternatives are defined in the *OEOverlapRadii* namespace.

SetRefGrid

```
bool SetRefGrid(const OESystem::OEScalarGrid& refgrid)
```

Set a reference grid for the calculation. An internal copy is made. Any previous reference molecule or grid is cleared.

SetRefMol

```
bool SetRefMol(const OEChem::OEMolBase& refmol)
```

Set a reference molecule for the calculation. An internal copy is made. Any previous reference molecule or grid is cleared.

SetUseHydrogens

```
bool SetUseHydrogens(bool state)
```

Boolean to determine whether hydrogens are included in the shape calculation. By default this is false and hydrogens are ignored.

5.1.6 OEShapeQueryPublic

```
class OEShapeQueryPublic
```

Initial public implementation of a shape query with limited functionality.

Constructors

```
OEShapeQueryPublic()
OEShapeQueryPublic(const OEShapeQueryPublic &)
```

Default and copy constructors.

operator bool

```
operator bool() const
```

Returns *true* if the instance is valid. Otherwise returns false.

GetColorForceField

```
const OEColorForceField* GetColorForceField() const
```

Returns a pointer to its *OEColorForceField*

GetTitle

```
const char* GetTitle() const
```

Returns the title.

5.2 Deprecated OEShape Constants

5.2.1 OEBOMinType

> **Note**
>
> This is a deprecated namespace.

This namespace contains constants.

Tanimoto

Overlap

Autoscale

Default

5.2.2 OEBOOrientation

> **Note**
>
> This is a deprecated namespace.

This namespace contains constants.

Inertial

Asls

Random

InertialAtHeavyAtoms

InertialAtColorAtoms

UserInertialStarts

Subrocs

Default

5.2.3 OEOverlapMethod

> **Note**
>
> This is a deprecated namespace.

This namespace contains constants.

Grid

Analytic

Analytic2

Exact

Max

5.2.4 OEOverlapRepresentation

> **Note**
>
> This is a deprecated namespace.

This namespace contains constants.

Atomic

Atoms

Bonding

Bonds

Full

All

Gaussian

Default

Max

CHAPTER

SIX

RELEASE HISTORY

6.1 Shape TK 3.10.0

6.1.1 New Features

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

6.1.2 Minor Bug Fixes

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

6.2 Shape TK 3.9.0

6.2.1 New Features

- The *SetupRef* method in *OEOverlay* now accepts molecules containing only color atoms. Accordingly, any molecule containing only color atoms or an *OEShapeQuery* containing only color Gaussians can now be used as a reference for overlay optimization.
- The *SetupRef* method in *OEOverlapFunc* now accepts molecules and *OEShapeQuery* containing just color components.

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

6.2.2 Minor Bug Fixes

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

6.3 Shape TK 3.8.0

6.3.1 New Features

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

6.4 Shape TK 3.7.0

6.4.1 New features

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

6.4.2 Minor bug fixes

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

6.5 Shape TK 3.6.2

6.5.1 New features

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

6.6 Shape TK 3.6.1

6.6.1 Major bug fixes

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

6.7 Shape TK 3.6.0

6.7.1 New features

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

6.7.2 Major bug fixes

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

6.7.3 Minor bug fixes

- The default value for the `color` function in *OEFlexiOverlapOptions* during *OEFlexiOverlay* has been changed from `none` to `exact`. This ensures that color overlaps are reported when flexible optimization is performed with default parameters.
- An issue has been fixed with *OEFlexiOverlay* that was causing shape Tanimoto to be occasionally reported as larger than 1.0 after a flexible optimization.

- The legal range of values for the color score multiplier has been added to the *OEFlexiOverlapOptions* and *OEOverlayOptions* option classes.
- *OEColorFFParameter* now allows users to select *explicitmillsdeannorings* properly as the color force field.
- The acceptable ranges for the number of optimization iteration steps have been defined for *OEFlexiOverlayOptions* and *OEOverlayOptions*.
- Newer implementation of Hermite code had limits on lambda parameters of the expansion. In rare cases, these were affecting the lambda parameter search (Newton optimization stepping outside the limits). Removing the limits fixed the previous problems with finding lambda parameters.

6.8 Shape TK 3.5.1

6.8.1 New features

- The following new methods have been added to *OEOverlayOptions* to improve flexibility in defining parameters:
  - *SetShapeFuncType*
  - *GetShapeFuncType*
  - *SetColorFuncType*
  - *GetColorFuncType*
  - *SetShapeOptions*
  - *GetShapeOptions*
  - *SetColorOptions*
  - *GetColorOptions*
- A new overload for the method *CreateGrid* has been added that takes a transform object as an additional argument that is applied after the grid is generated.

6.8.2 Major bug fixes

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

6.8.3 Minor bug fixes

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

6.8.4 Documentation changes

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

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

6.9 Shape TK 3.5.0

6.9.1 New features

- The new preliminary API classes,
  - *OEFlexiOverlapFunc*
  - *OEFlexiOverlapOptions*
  - *OEFlexiOverlapResults*

  have been added which provide an interface for overlap calculation between a reference molecule and a fit molecule using Shape and Color similarity along with intra-molecular force field of a fit molecule.
- The new preliminary API classes,
  - *OEFlexiOverlay*
  - *OEFlexiOverlayOptions*

have been added which provide an interface for flexible overlay optimization between a reference molecule and a fit molecule conformers using Shape and Color similarity along with intra-molecular force field of a fit molecule.

6.9.2 Minor bug fixes

- Swig wrapping issue for `GetShapeGaussians` and `GetColorGaussians` have been fixed.

6.10 Shape TK 3.4.3

Fall 2021

6.10.1 New features

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

6.10.2 Minor bug fixes

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

6.11 Shape TK 3.4.2

July 2021

6.11.1 Major bug fixes

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

6.12 Shape TK 3.4.1

Fall 2020

6.12.1 New features

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

6.12.2 Major bug fixes

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

6.12.3 Minor bug fixes

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

6.13 Shape TK 3.4.0

As part of the integration of **OEToolkits** and **OEApplications** into a single release, the version number of **Shape TK** has been upgraded to 3.4.0 to make it consistent with that of the **ROCS** application.

6.13.1 Minor bug fixes

- The following methods now return `bool` instead of a `void`:
  - *OEShapeOptions.SetCarbonRadius*
  - *OEShapeOptions.SetRadiiApproximation*
  - *OEShapeOptions.SetScoreType*
  - *OEColorOptions.SetMultiplier*
  - *OEColorOptions.SetScoreType*
  - *OEAnalyticOptions.SetExpType*
  - *OEAnalyticOptions.SetProxyGridCutoff*
  - *OEAnalyticOptions.SetUseProxyGrid*
  - *OEShapeGridOptions.SetDerivativeType*
  - *OEShapeGridOptions.SetGridSpacing*
  - *OEOverlayOptions.SetMaxOptSteps*
  - *OEROCSOptions.SetConfsPerHit*
  - *OEROCSOptions.SetMaxHits*
  - *OEROCSOptions.SetNumBestHits*
  - *OEROCSOptions.SetPerformPrep*
  - *OEROCSOptions.SetUseMaxHits*
- The explicit Mills-Dean color force fields, *ExplicitMillsDean* and *ExplicitMillsDean*, now accurately treat a tertiary amine as a donor and not an acceptor.

- The *OEColorForceField.AddColorer* method now checks to ensure that the SMARTS patterns added as color do not have a mix of explicit and implicit hydrogen expressions.
- The *OEOverlapPrep.Prep* and *OEAddColorAtoms* methods now properly handle color atom assignments for molecules with explicit hydrogens and color SMARTS patterns containing hybridization.

6.14 Shape TK 2.0.4

6.14.1 New features

- Two new methods, *SetMultiplier* and *GetMultiplier*, have been added to the *OEColorOptions* class that enable use of a multiplier to the calculated color scores.

6.14.2 Minor bug fixes

- The Python example in the section, *Calculating NxN scores*, has been fixed.
- The multithread version of the Python example in *Best Optimization Result* has been fixed.

6.15 Shape TK 2.0.3

6.15.1 New features

- A new overload of *OEWriteShapeQuery* has been added that writes queries to an oeostream object.

6.15.2 Minor bug fixes

- *OEAnalyticShapeFunc* and *OEGridShapeFunc* now produce a value of `1.0` for self-shape Tanimoto.
- *OEOverlay* now properly handles empty query objects and queries consisting of only color atoms.
- *OEOverlay* now properly handles molecules without coordinates.
- The *OEColorFunc* assignment operator has been fixed.

6.15.3 Documentation changes

- Code examples in the *Examples* section have been revised.

6.16 Shape TK 2.0.2

6.16.1 New features

- Hermite classes *OEHermiteOptions*, *OEHermite*, and *OEHermiteShapeFunc* have graduated from preliminary API to stable API status for the 2018.Oct release.
- The *OEHermiteShapeFunc* class now derives from *OEShapeFunc*. As a result, Hermite overlay can be done more efficiently using the *OEOverlay* class.

6.16.2 Minor bug fixes

- The *OEHermite.Setup* method now takes an OEMolBase rather than a OEMCMolBase parameter.
- The `OEShape::OEColorFunc::operator=` method has been fixed.

6.16.3 Java-specific changes

- *OEHermiteOptions*, *OEHermite*, and *OEHermiteShapeFunc* are now available in Java.

6.16.4 C#-specific changes

- *OEHermiteOptions*, *OEHermite*, and *OEHermiteShapeFunc* are now available in C#.

6.16.5 Documentation changes

- Java and C# examples have been added to the *Shape from Hermite expansion* examples.

6.17 Shape TK 2.0.1

6.17.1 New features

- A new class, *OEMultiRefOverlay*, has been added to provide overlay optimization with a multi-conformer molecule as the reference.
- A new method, `OEOverlay.BestOverlay`, has been added to provide the single best *OEBestOverlayScore* as output from an overlay optimization calculation.
- Two new methods, `OEColorOptions.SetScoreType` and `OEColorOptions.GetScoreType`, have been added to control color optimization options.
- A new class, *OEHighestTverskyCombo*, has been added to sort *OEBestOverlayScore* based on TverskyCombo.
- Instantiation of the old API classes *OEBestOverlay*, *OEOverlap*, and *OEColorOverlap* now produces a warning message stating that the class is deprecated.

6.17.2 Minor bug fixes

- An issue related to **Shape TK** requiring a case license has been fixed.
- An issue related to overlay optimization when an *OEShapeQuery* class with only Gaussians as the reference has been fixed.
- An unnecessary warning that occurred when an *OEColorForceField* was initialized with `OEColorFFType.Custom` has been removed.
- A SMARTS definition in `ImplicitMillsDean` that incorrectly defined an anion on the central carbon of a malate-like molecule with no hydrogen has been fixed.

6.17.3 Documentation changes

- The `hermite-tanimoto` example in both C++ and Python has been modified to move the fit molecule to the overlaid position.
- A Python example for performing best overlay calculations using multithreading has been added.

6.18 Shape TK 2.0.0

6.18.1 New features

- The **Shape TK** has been completely redesigned to make the library thread-safe, to make the library more flexible to use, and to allow the API to be easily extended with additional functionality.
- Three new classes, *OEExactShapeFunc*, *OEGridShapeFunc*, and *OEAnalyticShapeFunc*, have been added to provide static shape overlap calculation based on Exact, Grid, and Analytic methods.

- Three new classes, *OEExactColorFunc*, *OEGridColorFunc*, and *OEAnalyticColorFunc*, have been added to provide static color score calculation based on Exact, Grid, and Analytic methods.
- A new class, *OEShapeOptions*, has been added to define options related to all *OEShapeFunc* evaluations.
- A new class, *OEColorOptions*, has been added to define options related to all *OEColorFunc* evaluations.
- A new class, *OEShapeGridOptions*, has been added to define options related to grid-based method evaluations.
- A new class, *OEAnalyticOptions*, has been added to define options related to analytic method evaluations.
- A new class, *OEOverlapPrep*, has been added that can be used to prepare molecules for shape and/or color calculations.
- A new class, *OEOverlay*, has been added that can be used to overlay conformers of an OEMol on a reference system that can be a molecule, grid, or shape query.
- A new class, *OEOverlayOptions*, has been added that sets all options for overlay optimization using *OEOverlay*. *OEOverlayOptions* provides flexibility to optimize based on shape or color or both, to choose any method for shape or color calculation, and to optimize overlap or Tanimoto individually for shape and color.
- A new class, *OEROCS*, has been added that can be used to generate hits from a database on a reference system that can be a molecule, grid, or shape query. *OEROCS* provides top hits based on flexible scoring criterion.
- A new class, *OEROCSOptions*, has been added to define options for generating hits using *OEROCS*.
- A new class, *OEROCSResult*, has been added that extends the *OEBestOverlayScore* with overlaid conformer structures.
- Three overloads of a new function, `OEROCSOverlay`, have been added to provide the best hit between a pair of molecules.
- New unary predicates have been added to terminate a hit search calculation using *OEROCS*:
  - *OEColorTanimotoCutoff*
  - *OEFitColorTverskyCutoff*
  - *OEFitTverskyComboCutoff*
  - *OEFitTverskyCutoff*
  - *OEOverlapCutoff*
  - *OERefColorTverskyCutoff*
  - *OERefTverskyComboCutoff*
  - *OERefTverskyCutoff*
- The *OEShapeQueryPublic* class has been replaced by *OEShapeQuery* and is complete with functionality to create and edit a shape query with combinations of molecules, grids, and Gaussians.
- With the introduction of the new API classes in **Shape TK**, several previously existing classes have been deprecated. The following table shows the deprecated classes and their replacements:

Deprecated Class	New Class
`OEShape::OEOverlap`	*OEExactShapeFunc OEGridShapeFunc OEAnalyticShapeFunc*
`OEShape::OEColorOverlap`	*OEExactColorFunc OEGridColorFunc OEAnalyticColorFunc*
`OEShape::OEBestOverlay`	*OEOverlay OEROCS*
`OEShape::OEColorResults`	*OEOverlapResults*
`OEShape::OEShapeQueryPublic`	*OEShapeQuery*

- With the introduction of the new API, several obsolete functionalities have been removed. These functionalities were added in the early development days of **Shape TK** and are no longer deemed useful. Methods related to such functionality, contained within the deprecated classes, are now ignored. The list of methods that fall into this category, with their current behavior, is as follows:
  - `OEShape::OEOverlap::Overlap(const OESystem::OEScalarGrid&, OEOverlapResults&)`: The method does nothing and returns `false`.
  - `OEShape::OEOverlap::SetFitGrid(const OESystem::OEScalarGrid&)`: The method does nothing and returns `false`.
  - `OEShape::OEOverlap::SetRepresentationLevel(unsigned int)`: The method does nothing and returns `false`. The default representation level is `OEShape::OEOverlapRepresentation::Atomic`.
  - `OEShape::OEColorOverlap::SetAllColor(bool)`: The method does nothing and returns `false`. The default `AllColor` state is `true`.
  - `OEShape::OEBestOverlay::SetAllColor(bool)`: The method does nothing and returns `false`. The default `AllColor` state is `true`.
  - `OEShape::OEBestOverlay::SetInertialAxialDivisions(unsigned int)`: The method does nothing and returns `false`.
  - `OEShape::OEBestOverlay::SetRepresentationLevel(unsigned int)`: The method does nothing and returns `false`. The default representation level is `OEShape::OEOverlapRepresentation::Atomic`.
  - `OEShape::OEBestOverlay::Overlay(const OESystem::OEScalarGrid&)`: The method does nothing and returns an empty iterator.

6.18.2 Minor Bug Fixes

- The Hermite grid generated by *OEHermite* was previously offset from the original frame of the molecule being expanded. This has been corrected.
- Previously, using a molecule without 3D coordinates as a reference for an overlay had caused a crash. This has been fixed.

6.18.3 Documentation changes

- The **ShapeTK** examples have been rewritten to illustrate the use of the new API. The various `bestOverlay` examples now have more descriptive names.

6.19 Shape TK 1.11.0

6.19.1 New features

- A new API has been added to perform Hermite expansion and store the approximate shape of drug-size molecules and proteins.
  - The class *OEHermite* and associated options class *OEHermiteOptions* allows the user to perform Hermite expansion of a molecule and output the results to a grid.
  - The class *OEHermiteShapeFunc* can be used to obtain a best shape overlap of two molecules by approximating each molecule with its Hermite representation.

6.19.2 Documentation changes

- An overview of using Hermite expansion for molecular shape can be found here *Shape from Hermite Representation*.
- A new example section, *Shape from Hermite expansion*, has been added to illustrate the capabilities of the new API for Hermite expansion.

6.20 Shape TK 1.10.4

- Minor internal improvements have been made.

6.21 Shape TK 1.10.3

6.21.1 Python-specific changes

- The *OESortOverlayScores* function now accepts a custom scoring function derived from the `OEBestOverlayScoreSorterPred` Python class.

6.21.2 Java-specific changes

- The *OESortOverlayScores* function now accepts a custom scoring function derived from the `OEBestOverlayScoreSorterPred` Java class.

6.22 Shape TK 1.10.2

6.22.1 Major bug fixes

- `OEShape::OEBestOverlay::SetAllColor(true)` with custom color atoms will no longer result in Tanimoto values larger than 1.0 when custom color atoms are placed on the molecule. The bug was located in *OESelfColor* function when the `allColor` parameter was set to `true` and caused the function to arbitrarily remove the custom color atoms placed on the molecule and recalculate a new set of color atoms based on the color force field set. This meant *OEBestOverlay* was calculating the self term and the overlap terms with an entirely different set of color atoms. *OESelfColor* will now only incur a molecule copy and *OEAddColorAtoms* when *OEHasColorAtoms* is `false`.

6.22.2 Minor bug fixes

- *OEBestOverlay.SetColorForceField* will now throw a warning and return `false` if *OEColorForceField* is not valid, as determined by *OEColorForceField.Ready* method.

6.23 Shape TK 1.10.1

6.23.1 New features

- *OESelfShape* is now nearly two times faster for the default case of the carbon radii approximation with the *OEOverlapRadii.Carbon* constant.
- *OEUncompressColorAtoms* performance has been improved.

6.23.2 Documentation fixes

- A correction has been made to the Tversky equation in the *Molecular Shape* section.

6.24 Shape TK 1.10.0

6.24.1 Minor bug fixes

- A crash that occurred when attempting to use the Subrocs initial orientation method with an *OEBestOverlay* object to overlay molecules onto a reference grid has been fixed. A warning is now issued stating that Subrocs matching against grids is not implemented, and that inertial frame alignment will be used instead.

6.25 Shape TK 1.9.7

6.25.1 New features

- *OEReadShapeQuery* overload has been added to work on oeistream objects instead of a direct file name.
- *OEOverlap.SetGridSpacing* has been added to allow adjusting the grid spacing used by the *OEOverlapMethod.Grid* shape overlap method. Lower values typically yield overlaps that agree more closely with the *OEOverlapMethod.Exact* method, while higher values can sometimes be faster and will diverge more.

6.25.2 Major bug fixes

- *OEBOOrientation.Subrocs* will sometimes crash when given a grid as the reference. Some of these crashes have been fixed, but others are more difficult to pin down and will persist till the next release.

6.25.3 Minor bug fixes

- `OEShape::OEBestOverlayScore::GetTversky` can now accept alpha and beta of `0.0` provided both parameters are not `0.0`.
- *OEHasCompressedColorAtoms* return type has changed from `unsigned int` to `bool` as the return value was always Boolean.

6.25.4 Documentation changes

- *OEHasCompressedColorAtoms* and *OEDeleteCompressedColorAtoms* have been documented and are now fully supported.

6.26 Shape TK 1.9.6

6.26.1 New features

- A new *OEHasPerceivedColorAtoms* function has been added for determining whether *OEAddColorAtoms* has already been called on a molecule.

6.26.2 Major bug fixes

- *OEBestOverlay* will no longer cause undefined behavior and frequent crashes, particularly in Java, when more than one *OEBestOverlay* object is created and destroyed.
- Dummy atoms (atoms with an atomic number of 0) with non-zero radius will no longer cause shape Tanimoto values to exceed 1.0. These dummy atoms are now properly ignored for shape Tanimoto calculations.

- Undefined behavior can now be avoided when handling 32 different types of color atoms.
- Internal refactoring has been performed to improve stability.

6.26.3 Minor bug fixes

Examples of using exclusion volumes in a ROCS search were added for C++, Java, and C#.

6.27 Shape TK 1.9.5

6.27.1 New features

6.27.2 Major bug fixes

- *OEBestOverlay* constructor that takes a molecule will not terminate the process when given a molecule with only hydrogen atoms.

6.28 Shape TK 1.9.4

6.28.1 Major bug fixes

- *OECalcVolume* will no longer allocate excessive amounts of memory and usually crash for molecules without coordinates.

6.28.2 Minor bug fixes

- Removed unbounded stack allocations.

6.29 Shape TK 1.9.3

- Minor internal improvements.

6.30 Shape TK 1.9.2

6.30.1 Minor bug fixes

- A bug has been fixed where *OEBOOrientation.Subrocs* was not properly setting starting positions when the database molecule was larger than the query molecule.
- Fixed a rare invalid memory access from the *OEOverlapMethod.Analytic2* method.

6.31 Shape TK 1.9.1

6.31.1 Bug fixes

- *OEUncompressColorAtoms* no longer throws a warning when no color atoms are present.

6.32 Shape TK 1.9.0

6.32.1 New features

- *OEBOOrientation.Subrocs* has been added as a new starting position method. This method is very similar to *OEBOOrientation.InertialAtHeavyAtoms*, except that it automatically uses the heavy atoms and center-of-

mass of the larger molecule when determining starting positions.

- *OEShapeQueryPublic* has been added for reading shape queries. This is an initial public implementation with limited functionality.

6.32.2 Major bug fixes

6.32.3 Minor bug fixes

- *OEBOOrientation.Subrocs* will now be used for highly symmetric molecules instead of random starts.
- *OEBOOrientation.Subrocs* and *OEBOOrientation.InertialAtHeavyAtoms* use the center-of-mass in addition to the heavy atoms when determining starting positions.

6.33 Shape TK 1.8.3

6.33.1 New features

- *OEColorForceField.GetTypes* method added to return an iterator of the integer color forcefield types.

6.33.2 Minor bug fixes

- *OEAddColorAtoms* was setting an internal flag incorrectly such that *OEHasColorAtoms* would return `true` even if no color atoms were added. This has been fixed.
- *OEAddColorAtom* corrected to also set internal flags regarding presence of color atoms.
- *OEBOOrientation.InertialAtHeavyAtoms* will now use the center-of-mass as a startpoint in addition to all heavy atoms. The constant is an option for *OEBestOverlay.SetInitialOrientation*.
- *OEBOOrientation.UserInertialStarts* will now work with the reference being a grid.

6.34 Shape TK 1.8.2

- Internal build system improvements.

6.35 Shape TK 1.8.1

6.35.1 Bug fixes

- Fixing a crash in *OECalcVolume* when dealing with Zinc and dummy atoms.
- *OECalcVolume* will now default to using appropriate Bondi VdW radii when no radii are set on the molecule. This effectively changes the value returned this function most molecules.

6.36 Shape TK 1.8.0

6.36.1 New features

- Added new functions that will cache the self shape score on a molecule. Analogous to the caching of self color, introduced in 1.7.2, this can speed up repeat shape calculations on the same molecule by not recalculating the self terms.
  - *OESetCachedSelfShape*
  - *OEHasCachedSelfShape*

  - *OEGetCachedSelfShape*
  - *OEClearCachedSelfShape*

- Updated the functions that cache self color to store work with multi-conformer molecules, storing a self color for each conformer.
- Added two new color force fields, ImplicitMillsDeanNoRings and ExplicitMillsDeanNoRings. These are identical to the existing similarly named force fields, but with no "rings" atom types or interactions.
- Added Has and Delete functions to the API for dealing with compressed color atoms.
- Several of the examples have been updated to count the output better and to use a better sort functor when color is included.
- Add new methods to *OEBestOverlayScore* to allow access to the transform.
  - *OEBestOverlayScore.GetRotMatrix*
  - *OEBestOverlayScore.GetTranslation*

6.36.2 Bug fixes

- Added OEGetColorAtoms and OEGetColorParents to Python, Java and C#.
- Fixed a bug where non-deterministic results could sometime occur when using OEBestOverlay and a highly symmetric molecule.
- Fixed calculation of multipoles. Previous versions used a non-irreducible representation of multipoles that is nonstandard. This method now calculates standard multipoles and allow better comparison between the values of two molecules.
- Fixed a missing type in ImplicitMillsDean to match an aldehyde acceptor with or without the explicit proton.
- Fixed a bug where OECalcVolume could crash using a molecule with deleted atoms.

6.37 Shape TK 1.7.2

6.37.1 New features

- OEAddColorAtoms now has an optional third argument to cause precalculation and caching of the self color. When re-using molecules in multiple comparisons (for example NxN studies), use OEAddColorAtoms once and then OEBestOverlay and OEColorOverlap will use the pre-stored color atoms and self color to speed up the calculations.
- Added a set of functions to manipulate cached self color.
  - *OESetCachedSelfColor*
  - *OEHasCachedSelfColor*
  - *OEGetCachedSelfColor*
  - *OEClearCachedSelfColor*
- The pair of functions (`OEColorAtomsToString` and `OEStringToColorAtoms`) added in v1.7.1 have been renamed to more appropriate names. The new functions are *OECompressColorAtoms* and *OEUncompressColorAtoms*.

6.37.2 Bug fixes

- Fixed a bug where dummy atoms could cause an erroneous result for OECalcVolume.

6.38 Shape TK 1.7.1

6.38.1 New features

- Added new types in *OEBOOrientation*. All of these are designed to provide a more deterministic search over the reference molecule for those case where the size of the fit molecule is much smaller than the reference, for example, when trying to match a fragment into part of a reference molecule.
  - *OEBOOrientation.InertialAtHeavyAtoms* moves the center of mass of the fit molecule to each reference molecule heavy atom and performs 4 inertial starts at the position. This results in many more starting positions, but provides a more direct way to search over an entire reference molecule, without resorting to random starts.
  - *OEBOOrientation.InertialAtColorAtoms* performs a similar search as above, but just moves to the location of each reference molecule color atom.
  - *OEBOOrientation.UserInertialStarts*, used in conjunction with *OEBestOverlay.SetUserStarts* allows the user to pick specific points in space to perform the 4 inertial starts.
- Fixed a bug when calculating Tanimoto while using a grid as the reference object.
- Added more functions to manipulate the color atoms on a molecule. These include the ability to add color atoms one at a time (*OEAddColorAtom*) and the ability to get an iterator of color atoms from a molecule (*OEGetColorAtoms*).
- Added a pair of functions (`OEShape::OEColorAtomsToString` and `OEShape::OEStringToColorAtoms`) that allow converting the color atoms of a molecule into a compressed string representation (that is attached to the molecule) and then to restore the actual color atoms from that string.

6.38.2 Bug fixes

- Fixed a bug that could cause a crash when passing an empty molecule into *OECalcVolume* or *OECalcShapeMultipoles*.
- Fixed a bug that could cause a crash when passing large molecules to *OECalcVolume*.

6.39 Shape TK 1.7.0

6.39.1 New features

- Color Tanimoto is calculated just like Shape Tanimoto by using the self color scores of each molecule combined with the actual color score. The end result is to better match molecules of the same color, penalizing a molecule for having too much as well as too little color. Color Tversky scores can be calculated as well. We now calculate these extra color scores in *OEColorOverlap* and *OEBestOverlay*.
- *OEColorResults* now has Tanimoto, Tversky, RefTversky and FitTversky.
- *OEBestOverlayScore* now has ColorTanimoto, ColorTversky, RefColorTversky, FitColorTversky and adds three new combo scores based on these:
  - TanimotoCombo - the sum of Shape Tanimoto and Color Tanimoto
  - RefTverskyCombo - the sum of Ref Shape Tversky and Ref Color Tversky
  - FitTverskyCombo - the sum of Fit Shape Tversky and Fit Color Tversky

- *OEOverlapResults* now has Tversky, RefTversky and FitTversky.

6.39.2 Bug fixes

- Fixed an internal memory leak in OEColorOverlap/OEBestOverlap color objects.
- Fixed a bug where in some cases the user-supplied value for carbon radius in OEOverlap was ignored.

6.40 Shape TK 1.6.2

6.40.1 New features

- Added the ability to get the expiration date from *OEShapeIsLicensed* function.

6.40.2 Bug fixes

- Fixed a memory leak in the calculation of self color in OEColorFunc.
- Fixed a bug that prevented pre-colored molecules from being used in OEColorFunc.
- Replace old error messages that just went to stderr and routed all errors and warnings via OEThrow.

6.41 Shape TK 1.6.1

This is the first release as part of the consolidated toolkits release.

6.41.1 New features

- Added functions to retrieve the version and build date of the library.

6.41.2 Bug fixes

- Changed to a stable sort inside *OESortOverlayScores* to prevent crashes with a very large number of results.
- Fixes to *OEOverlap* that in some cases could result in slow behavior when internal grids were rebuilt instead of being cached.
- Fix a subtle bug that could result in erroneous behavior if reusing the same *OEOverlap* object while changing the fit object from a grid to a molecule or from a molecule to a grid.

6.42 Shape TK 1.6

6.42.1 New features

- Added new functions to calculate steric volume (*OECalcVolume*) and steric multipoles (*OECalcShapeMultipoles*).

6.42.2 Bug fixes

- Fixed a bug where *OEAddColorAtoms* would not correctly add color atoms to a molecule that had the IntType field on atoms filled from a previous calculation. IntType is scratch space on atoms, so *OEAddColorAtoms* now clears the IntType before adding color atoms to a molecule.
- Changed one of the cation definitions in ImplicitMillsDean force field from the too general (“n:1anaa1”) to the more specific (“n:1cncc1”).
- Fixed a bug that caused *OEBOOrientation.AsIs* to not always start from the passed in positions.

6.43 Shape TK 1.5

6.43.1 New features

- First release in C++, Java and Python.

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-stream	https://github.com/conda/conda-package-stream/blob/main/LICENSE
conda-token	https://anaconda.org/anaconda/conda-token	https://anaconda.org/anaconda/conda-token

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
conda_package_strear	https://github.com/conda/conda-package-streamtree/main/conda_package_streaming	https://github.com/conda/conda-package-streamtab=License-1-ov-file
confuse	https://github.com/beetbox/confuse	https://github.com/beetbox/confuse/blob/main/LICENSE
contourpy	https://github.com/contourpy/contourpy	https://github.com/contourpy/contourpy/blob/main/LICENSE
cpp-expected	https://github.com/TartanLlama/expected	https://github.com/TartanLlama/expected?tab=CC0-1.0-1-ov-file#readme
cpp-peglib	https://github.com/yhirose/cpp-peglib	https://github.com/yhirose/cpp-peglib/blob/master/LICENSE
cryptography	https://github.com/pyca/cryptography	https://github.com/pyca/cryptography/blob/main/LICENSE
cssselect2	https://github.com/Kozea/cssselect2/	https://github.com/Kozea/cssselect2/blob/master/LICENSE
cuda-bindings	https://nvidia.github.io/cuda-python/cuda-bindings/latest/install.html	https://github.com/NVIDIA/cuda-python/blob/main/cuda_bindings/LICENSE
cuda-pathfinder	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/license.html
cudatoolkit	https://developer.nvidia.com/cuda-toolkit	https://docs.nvidia.com/cuda/eula/index.html
cuequivariance-ops-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cuequivariance-ops-torch-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cuequivariance-torch	https://docs.nvidia.com/cuda/cuequivariance/	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cupy-cuda113	https://cupy.dev/	https://github.com/cupy/cupy/blob/main/LICENSE
cupy-cuda11x	https://cupy.dev/	https://github.com/cupy/cupy/?tab=MIT-1-ov-file#readme
curl	https://curl.se	https://curl.se/docs/copyright.html
cycler	https://github.com/matplotlib/cycler	https://github.com/matplotlib/cycler/blob/main/LICENSE
cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl/blob/master/COPYING
Cython	https://cython.org	https://www.apache.org/licenses/LICENSE-2.0.html
d3	https://github.com/mbostock/d3	https://github.com/mbostock/d3/raw/master/LICENSE
dataclasses	https://github.com/ericvsmith/dataclasses	https://github.com/ericvsmith/dataclasses/blob/master/LICENSE.txt
ddsketch	http://github.com/datadog/sketches-py	https://github.com/DataDog/sketches-py/blob/master/LICENSE
debugpy	https://aka.ms/debugpy	https://github.com/microsoft/debugpy/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
decimal	https://github.com/shopspring/decimal	https://github.com/shopspring/decimal/blob/master/LICENSE
decorator	https://github.com/micheles/decorator	https://github.com/micheles/decorator/blob/master/LICENSE.txt
deepdiff	https://github.com/seperman/deepdiff	https://github.com/seperman/deepdiff/blob/master/LICENSE
deeptime	https://github.com/deeptime-ml/deeptime	https://github.com/deeptime-ml/deeptime/blob/main/LICENSE.txt
defusedxml	https://github.com/tiran/defusedxml	https://github.com/tiran/defusedxml/blob/main/LICENSE
dftd3	https://github.com/dftd3/tad-dftd3	https://github.com/dftd3/tad-dftd3/blob/main/LICENSE
dill	https://github.com/uqfoundation/dill	https://github.com/uqfoundation/dill/blob/master/LICENSE
distlib	https://distlib.readthedocs.io/en/latest/	https://github.com/pypa/distlib/blob/master/LICENSE.txt
distro	https://github.com/python-distro/distro	https://github.com/python-distro/distro/blob/master/LICENSE
Django	https://www.djangoproject.com/	https://github.com/django/django/blob/main/LICENSE
django-classy-tags	https://github.com/django-cms/django-classy-tags	https://github.com/django-cms/django-classy-tags/blob/master/LICENSE
django-cors-headers	https://github.com/adamchainz/django-cors-headers	https://github.com/adamchainz/django-cors-headers/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
django-csp	https://github.com/mozilla/django-csp	https://github.com/mozilla/django-csp/blob/main/LICENSE
django-extensions	https://github.com/django-extensions/django-extensions	https://github.com/django-extensions/django-extensions/blob/main/LICENSE
django-filter	https://github.com/carltongibson/django-filter/tree/master	https://github.com/carltongibson/django-filter/blob/main/LICENSE
django-redis	https://github.com/jazzband/django-redis	https://github.com/jazzband/django-redis/blob/master/LICENSE
django-taggit	https://github.com/jazzband/django-taggit	https://github.com/jazzband/django-taggit/blob/master/LICENSE
django-taggit-serializer	https://github.com/glemmaPaul/django-taggit-serializ	https://github.com/glemmaPaul/django-taggit-serializ blob/master/LICENSE
django-taggit-templatetags2	https://github.com/fizista/django-taggit-templat	https://github.com/fizista/django-taggit-templat blob/master/LICENSE.txt
djangorestframework	https://www.django-rest-framewor org/	https://github.com/encode/django-rest-framewor blob/master/LICENSE.md
dkh	https://psicode.org/psi4manual/master/dkh.html	https://github.com/psi4/dkh/blob/master/COPYING.LESSER
dm-tree	https://github.com/deepmind/tree	https://github.com/deepmind/tree/blob/master/LICENSE
docker-py	https://github.com/docker/docker-py/	https://github.com/docker/docker-py/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
docker-pycreds	https://github.com/shin-/dockerpy-creds	https://github.com/shin-/dockerpy-creds/blob/master/LICENSE
docopt	https://docopt.org	https://github.com/docopt/docopt/blob/master/LICENSE-MIT
docutils	https://docutils.sourceforge.io	https://docutils.sourceforge.io/COPYING.html
drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields/blob/master/LICENSE
DuckDB	https://duckdb.org/	https://github.com/duckdb/duckdb?tab=MIT-1-ov-file#readme
edgembar	https://rutgerslbsr.gitlab.io/fe-toolkit/edgembar/index.html	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
editdistance	https://github.com/roy-ht/editdistance	https://github.com/roy-ht/editdistance/blob/master/LICENSE
eigen	https://eigen.tuxfamily.org/index.php?title=Main_Page	https://eigen.tuxfamily.org/index.php?title=Main_Page#License
einops	https://github.com/arogozhnikov/einops	https://github.com/arogozhnikov/einops/blob/master/LICENSE
einx	https://github.com/fferflo/einx	https://github.com/fferflo/einx/blob/master/LICENSE
entrypoints	https://github.com/takluyver/entrypoints	https://github.com/takluyver/entrypoints/blob/master/LICENSE
errors	https://github.com/pkg/errors	https://github.com/pkg/errors/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
eslint-plugin	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile/-/blob/branch/default/LICENCE.rst
exceptiongroup	https://github.com/agronholm/exceptiongroup	https://github.com/agronholm/exceptiongroup/blob/main/LICENSE
execnet	https://execnet.readthedocs.io/en/latest/	https://github.com/pytest-dev/execnet/blob/master/LICENSE
executing	https://github.com/alexmojaki/executing	https://github.com/alexmojaki/executing/blob/master/LICENSE.txt
expat	https://libexpat.github.io	https://github.com/libexpat/libexpat/blob/master/expat/COPYING
fairscale	https://github.com/facebookresearch/fairscale	https://github.com/facebookresearch/fairscale/blob/main/LICENSE
fairseq	https://github.com/pytorch/fairseq	https://github.com/facebookresearch/fairseq/blob/main/LICENSE
fastjsonschema	https://github.com/horejsek/python-fastjsonschem	https://github.com/horejsek/python-fastjsonschem/blob/master/LICENSE
fastrlock	https://github.com/scoder/fastrlock	https://github.com/scoder/fastrlock/blob/master/LICENSE
fftw	https://www.fftw.org	commercial license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
filebuffer	https://github.com/mattetti/filebuffer	https://github.com/mattetti/filebuffer/blob/master/LICENSE
filelock	https://py-filelock.readthedocs.io/en/latest/index.html	https://github.com/tox-dev/py-filelock/blob/main/LICENSE
finufft	https://github.com/flatironinstitute/finufft	https://github.com/flatironinstitute/finufft/blob/master/LICENSE
Flask	https://flask.palletsprojects.com/en/2.1.x/	https://flask.palletsprojects.com/en/2.1.x/license/
flatbuffers	https://google.github.io/flatbuffers/	https://github.com/google/flatbuffers/blob/master/LICENSE.txt
flexcache	https://github.com/hgrecco/flexcache	https://github.com/hgrecco/flexcache/blob/main/LICENSE
flexparser	https://github.com/hgrecco/flexparser	https://github.com/hgrecco/flexparser/blob/main/LICENSE
flit-core	https://github.com/pypa/flit	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
FLTK	https://www.fltk.org/index.php	https://www.fltk.org/COPYING.php
fmt	https://fmt.dev/latest/index.html	https://github.com/fmtlib/fmt/blob/master/LICENSE.rst
font-awesome	https://github.com/FortAwesome/Font-Awesome	https://github.com/FortAwesome/Font-Awesome
font-ttf-dejavu-sans-mono	https://dejavu-fonts.github.io	https://dejavu-fonts.github.io/License.html
font-ttf-inconsolata	https://fonts.google.com/specimen/Inconsolata	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
font-ttf-source-code-pro	https://fonts.google.com/specimen/Source+Code+Pro	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL
font-ttf-ubuntu	https://fonts.google.com/specimen/Ubuntu	https://ubuntu.com/legal/font-licence
fontconfig	https://www.freedesktop.org/wiki/Software/fontconfig/	https://gitlab.freedesktop.org/fontconfig/fontconfig/-/blob/main/COPYING
fonts-conda-ecosystem	https://anaconda.org/conda-forge/fonts-conda-ecosyster	https://docs.conda.io/en/latest/license.html
fonts-conda-forge	https://anaconda.org/conda-forge/fonts-conda-forge/	https://creativecommons.org/licenses/by/4.0/
fonttools	https://github.com/fonttools/fonttools	https://github.com/fonttools/fonttools/blob/main/LICENSE
forcebalance	https://simtk.org/home/forcebalance OR https://github.com/leeping/forcebalance	https://github.com/leeping/forcebalance/blob/master/LICENSE
fqdn	https://github.com/ypcrts/fqdn	https://github.com/ypcrts/fqdn?tab=MPL-2.0-1-ov-file#readme
freetype	https://freetype.org	https://freetype.org/license.html
freetype-py	https://github.com/rougier/freetype-py	https://github.com/rougier/freetype-py?tab=License-1-ov-file#readme
fribidi	https://github.com/fribidi/fribidi	https://github.com/fribidi/fribidi/blob/master/COPYING
frozendict	https://github.com/Marco-Sulla/python-frozendict	https://github.com/Marco-Sulla/python-frozendict/blob/master/LICENSE.txt
frozenlist	https://github.com/aio-libs/frozenlist	https://github.com/aio-libs/frozenlist/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
fsmlite	https://github.com/tkem/fsmlite	https://github.com/tkem/fsmlite/blob/master/LICENSE
fsspec	https://github.com/fsspec/filesystem_spec	https://github.com/fsspec/filesystem_spec/blob/master/LICENSE
funcy	https://github.com/Suor/funcy	https://github.com/Suor/funcy/blob/master/LICENSE
gast	https://github.com/serge-sans-paille/gast/	https://github.com/serge-sans-paille/gast/blob/master/LICENSE
gau2grid	https://github.com/dgasmith/gau2grid	https://github.com/dgasmith/gau2grid/blob/master/LICENSE
gax-go	https://github.com/googleapis/gax-go/v2	https://github.com/googleapis/gax-go/blob/main/LICENSE
gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf/-/blob/master/COPYING
gemmi	https://gemmi.readthedocs.io/en/latest/	https://github.com/project-gemmi/gemmi/blob/master/LICENSE.txt
genproto	https://google.golang.org/genproto/googleapis	https://github.com/googleapis/go-genproto/blob/main/LICENSE
geometric	https://openbase.com/python/geometric	https://openbase.com/python/geometric
giflib	https://giflib.sourceforge.net	https://github.com/mirrorer/giflib/blob/master/COPYING
gitdb	https://github.com/gitpython-developers/gitdb	https://github.com/gitpython-developers/gitdb/blob/master/LICENSE
GitPython	https://github.com/gitpython-developers/GitPython	https://github.com/gitpython-developers/GitPython/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
glib	https://docs.gtk.org/glib/	https://gitlab.gnome.org/GNOME/glib/-/blob/main/LICENSES/LGPL-2.1-or-later.txt
GLM Library	https://github.com/g-truc/glm	https://github.com/g-truc/glm/blob/master/manual.md#section0
gls	https://github.com/jtolds/gls	https://github.com/jtolds/gls/blob/master/LICENSE
gmpy2	https://github.com/aleaxit/gmpy	https://github.com/gmpy2/gmpy2/blob/master/COPYING.LESSER
go-cleanhttp	https://github.com/hashicorp/go-cleanhttp	https://github.com/hashicorp/go-cleanhttp/blob/master/LICENSE
go-connections	https://github.com/docker/go-connections	https://github.com/docker/go-connections/blob/master/LICENSE
go-cpio	https://github.com/cavaliercoder/go-cpio	https://github.com/cavaliercoder/go-cpio/blob/master/LICENSE
go-getter	https://github.com/hashicorp/go-getter	https://github.com/hashicorp/go-getter/blob/master/LICENSE
go-homedir	https://github.com/mitchellh/go-homedir	https://github.com/mitchellh/go-homedir/blob/master/LICENSE
go-ini	https://github.com/go-ini/ini	https://github.com/go-ini/ini/blob/master/LICENSE
go-jmespath	https://github.com/jmespath/go-jmespath	https://github.com/jmespath/go-jmespath/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go-junit-report	https://github.com/jstemmer/go-junit-report	https://github.com/jstemmer/go-junit-report/blob/master/LICENSE
go-netrc	https://github.com/bgentry/go-netrc/netrc	https://github.com/bgentry/go-netrc/blob/master/LICENSE
go-ole	https://github.com/go-ole/go-ole	https://github.com/go-ole/go-ole/blob/master/LICENSE
go-pg	https://github.com/go-pg/pg	https://github.com/go-pg/pg/blob/master/LICENSE
go-redis	https://github.com/go-redis/redis/v8	https://github.com/go-redis/redis/blob/master/LICENSE
go-rendezvous	https://github.com/dgryski/go-rendezvous	https://github.com/dgryski/go-rendezvous/blob/master/LICENSE
go-safetemp	https://github.com/hashicorp/go-safetemp	https://github.com/hashicorp/go-safetemp/blob/master/LICENSE
go-sysconf	https://github.com/tklauser/go-sysconf	https://github.com/tklauser/go-sysconf/blob/master/LICENSE
go-testing-interface	https://github.com/mitchellh/go-testing-interface	https://github.com/mitchellh/go-testing-interface/blob/master/LICENSE
go-units	https://github.com/docker/go-units	https://github.com/docker/go-units/blob/master/LICENSE
go-version	https://github.com/hashicorp/go-version	https://github.com/hashicorp/go-version/blob/master/LICENSE
go-zglob	https://github.com/mattn/go-zglob	https://github.com/mattn/go-zglob/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go.opencensus	https://go.opencensus.io	https://github.com/census-instrumentatio opencensus-go/blob/master/LICENSE
gobrake.v2	https://gopkg.in/airbrake/gobrake.v2	https://github.com/airbrake/gobrake/blob/v2.0.9/LICENSE
goconvey	https://github.com/smartystreets/goconvey	https://github.com/smartystreets/goconvey/blob/master/LICENSE.md
golden-layout	https://github.com/deepstreamIO/golden-layout	https://github.com/deepstreamIO/golden-layout/raw/master/LICENSE
google-auth	https://google-auth.readthedocs.io/en/master/	https://github.com/GoogleCloudPlatform google-auth-library-p blob/master/LICENSE
google-auth-oauthlib	https://github.com/googleapis/google-auth-library-p	https://github.com/googleapis/google-auth-library-p blob/main/LICENSE
google-cloud-go	https://cloud.google.com/go	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-cloud-go/storage	https://cloud.google.com/go/storage	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-pasta	https://github.com/google/pasta	https://github.com/google/pasta/blob/master/LICENSE
google.golang.org/api	https://google.golang.org/api	https://github.com/googleapis/google-api-go-client/blob/main/LICENSE
gopsutil	https://github.com/shirou/gopsutil	https://github.com/shirou/gopsutil/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
gorilla websocket	https://github.com/gorilla/websocket	https://github.com/gorilla/websocket/blob/master/LICENSE
gpytorch	https://gpytorch.ai	https://github.com/cornellius-gp/gpytorch?tab=MIT-1-ov-file#readme
graphite2	https://github.com/silnrsi/graphite	https://github.com/silnrsi/graphite/blob/master/LICENSE
graphviz	https://graphviz.org/	https://graphviz.org/license/
greenlet	https://greenlet.readthedocs.io/en/latest/	https://github.com/python-greenlet/greenlet/blob/master/LICENSE
groupcache	https://github.com/golang/groupcache	https://github.com/golang/groupcache/blob/master/LICENSE
grpc	https://google.golang.org/grpc	https://github.com/grpc/grpc-go/blob/master/LICENSE
grpc-cpp	https://github.com/grpc/grpc	https://github.com/grpc/grpc/blob/master/LICENSE
grpcio	https://github.com/grpc/grpc.io/blob/main/LICENSE	https://github.com/grpc/grpc.io/blob/main/LICENSE
gtk2	https://gitlab.gnome.org/GNOME/gtk	https://gitlab.gnome.org/GNOME/gtk/-/blob/main/COPYING
gts	https://sourceforge.net/projects/gts/	https://www.gnu.org/licenses/lgpl-3.0.html
h11	https://github.com/python-hyper/h11	https://github.com/python-hyper/h11/blob/master/LICENSE.txt
h2	https://github.com/python-hyper/h2	https://github.com/python-hyper/h2?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
h5py	https://www.h5py.org	https://docs.h5py.org/en/stable/licenses.html
harfbuzz	https://github.com/harfbuzz/harfbuzz	https://github.com/harfbuzz/harfbuzz/blob/main/COPYING
hdbscan	https://hdbscan.readthedocs.io/en/latest/	https://github.com/scikit-learn-contrib/hdbscan/blob/master/LICENSE
hdf4	https://www.hdfgroup.org/solutions/hdf4/	https://github.com/HDFGroup/hdf4/blob/master/COPYING
hdf5	https://www.hdfgroup.org/solutions/hdf5/	https://github.com/HDFGroup/hdf5/blob/master/COPYING
he	https://github.com/mathiasbynens/he	https://github.com/mathiasbynens/he/raw/master/LICENSE-MIT.txt
hpack	https://github.com/python-hyper/hpack	https://github.com/python-hyper/hpack?tab=MIT-1-ov-file#readme
html-loader	https://github.com/webpack-contrib/html-loader	https://github.com/webpack-contrib/html-loader/raw/master/LICENSE
html5lib	https://github.com/html5lib/html5lib-python	https://github.com/html5lib/html5lib-python/blob/master/LICENSE
htslib	https://www.htslib.org	https://github.com/samtools/htslib/blob/develop/LICENSE
httpcore	https://www.encode.io/httpcore/	https://github.com/encode/httpcore/blob/master/LICENSE.md
httpx	https://github.com/encode/httpx	https://github.com/encode/httpx/blob/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
humanize	https://github.com/jmoiron/humanize	https://github.com/jmoiron/humanize/blob/master/LICENCE
hydra-core	https://github.com/facebookresearch/hydra	https://github.com/facebookresearch/hydra/blob/main/LICENSE
hyperframe	https://github.com/python-hyper/hyperframe	https://github.com/python-hyper/hyperframe?tab=MIT-1-ov-file#readme
icu	https://github.com/unicode-org/icu	https://github.com/unicode-org/icu/blob/main/icu4c/LICENSE
idna	https://github.com/kjd/idna	https://github.com/kjd/idna/blob/master/LICENSE.md
ihm	https://github.com/ihmwg/python-ihm	https://github.com/ihmwg/python-ihm/blob/main/LICENSE
imageio	https://github.com/imageio/imageio	https://github.com/imageio/imageio/blob/master/LICENSE
imagesize	https://github.com/shibukawa/imagesize_py	https://github.com/shibukawa/imagesize_py/blob/master/LICENSE.rst
ImmuneBuilder	https://github.com/oxpig/ImmuneBuilder/tree/main	https://github.com/oxpig/ImmuneBuilder/blob/main/LICENSE
importlib-metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE
importlib_metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
importlib_resources	https://github.com/python/importlib_resources	https://github.com/python/importlib_resources/blob/main/LICENSE
InChI	https://iupac.org/who-we-are/divisions/division-details/inchi/	https://iupac.org/wp-content/uploads/2021/06/iupac-inchi-license_2020.pdf
inflection	https://github.com/jinzhu/inflection	https://github.com/jinzhu/inflection/blob/master/LICENSE
ini.v1	https://gopkg.in/ini.v1	https://github.com/go-ini/ini/blob/v1.66.2/LICENSE
iniconfig	https://github.com/pytest-dev/iniconfig	https://github.com/pytest-dev/iniconfig/blob/main/LICENSE
innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill/raw/master/LICENSE
intel-openmp	https://github.com/hermitcore/libomp_oss	https://github.com/hermitcore/libomp_oss/blob/hermit/LICENSE.txt
ipykernel	https://ipython.org	https://ipython.org
ipython	https://ipython.org	https://github.com/ipython/ipython/blob/master/LICENSE
ipython-genutils	http://ipython.org	https://github.com/ipython/ipython_genutils/blob/master/COPYING.md
ipython_pygments_le	https://github.com/ipython/ipython-pygments-lex	https://github.com/ipython/ipython-pygments-lex blob/main/LICENSE
ipywidgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
isodate	https://github.com/gweis/isodate/	https://github.com/gweis/isodate/blob/master/LICENSE
isoduration	https://github.com/bolsote/isoduration	https://github.com/bolsote/isoduration?tab=ISC-1-ov-file#readme
itsdangerous	https://palletsprojects.com/p/itsdangerous/	https://itsdangerous.palletsprojects.com/en/2.1.x/license/
jax	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxlib	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxtyping	https://github.com/patrick-kidger/jaxtyping	https://github.com/patrick-kidger/jaxtyping?tab=License-1-ov-file#readme
jedi	https://jedi.readthedocs.io/en/latest/	https://github.com/davidhalter/jedi/blob/master/LICENSE.txt
Jinja2	https://palletsprojects.com/p/jinja/	https://github.com/pallets/jinja/blob/main/LICENSE.rst
jmespath	https://github.com/jmespath/jmespath.py	https://github.com/jmespath/jmespath.py/blob/develop/LICENSE.txt
joblib	https://joblib.readthedocs.io/en/latest/	https://github.com/joblib/joblib/blob/master/LICENSE.txt
jpeg	https://www.ijg.org	https://www.ijg.org/files/README
json5	https://github.com/dpranke/pyjson5	https://github.com/dpranke/pyjson5/blob/main/LICENSE
jsonfield	https://github.com/dmkoch/django-jsonfield/	https://github.com/rpkilby/jsonfield/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jsonpatch	https://github.com/stefankoegl/python-json-patch	https://github.com/stefankoegl/python-json-patch/blob/master/LICENSE
jsonpickle	https://github.com/jsonpickle/jsonpickle	https://github.com/jsonpickle/jsonpickle/blob/main/LICENSE
jsonpointer	https://github.com/stefankoegl/python-json-pointer	https://github.com/stefankoegl/python-json-pointer/blob/master/LICENSE.txt
jsonschema	https://github.com/python-jsonschema/jsonschema	https://github.com/python-jsonschema/jsonschema/blob/master/COPYING
jsonschema-specifications	https://github.com/python-jsonschema/jsonschema-specificat	https://github.com/python-jsonschema/jsonschema-specificat blob/main/COPYING
jstat	https://github.com/jstat/jstat	https://www.opensource.org/licenses/mit-license.php
jupyter-client	https://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/COPYING.md
jupyter-core	https://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/COPYING.md
jupyter-events	https://github.com/jupyter/jupyter_events	https://github.com/jupyter/jupyter_events/blob/main/LICENSE
jupyter-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp
jupyter-serverhttp://jupyter.or	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE	
jupyter_client	http://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jupyter_core	http://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/LICENSE
jupyter_server	https://github.com/jupyter-server/jupyter_server	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE
jupyter_server_termin	https://github.com/jupyter-server/jupyter_server_terminals	https://github.com/jupyter-server/jupyter_server_terminals/blob/main/LICENSE
jupyterlab	https://github.com/jupyterlab/jupyterlab	https://github.com/jupyterlab/jupyterlab/blob/main/LICENSE
jupyterlab-pygments	http://jupyter.org	https://github.com/jupyterlab/jupyterlab_pygments/blob/main/LICENSE
jupyterlab-widgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/master/LICENSE
jupyter-lab_pygments	https://github.com/jupyterlab/jupyterlab_pygments	https://github.com/jupyterlab/jupyterlab_pygments?tab=BSD-3-Clause-1-ov-file#readme
jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server/blob/main/LICENSE
jupyterlab_widgets	https://github.com/jupyter-widgets/ipywidgets	https://github.com/jupyter-widgets/ipywidgets?tab=BSD-3-Clause-1-ov-file#readme
kaleido	https://github.com/plotly/Kaleido	https://github.com/plotly/Kaleido/blob/master/LICENSE.txt
keras	https://github.com/keras-team/keras	https://github.com/keras-team/keras/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
Keras-Preprocessing	https://github.com/keras-team/keras-preprocessing	https://github.com/keras-team/keras-preprocessing/blob/master/LICENSE
keras-tuner	https://github.com/keras-team/keras-tuner	https://github.com/keras-team/keras-tuner/blob/master/LICENSE
keyring	https://github.com/jaraco/keyring	https://github.com/jaraco/keyring/blob/main/LICENSE
keyutils	https://github.com/sassoftware/python-keyutils	https://github.com/sassoftware/python-keyutils/blob/master/LICENSE
kiwisolver	https://kiwisolver.readthedocs.io/en/latest/	https://github.com/nucleic/kiwi/blob/main/LICENSE
kombu	https://kombu.readthedocs.io	https://github.com/celery/kombu/blob/master/LICENSE
krb5	https://web.mit.edu/kerberos/	https://github.com/krb5/krb5/blob/master/src/lib/gssapi/LICENSE
kt-legacy	https://github.com/haifeng-jin/kt-legacy	https://github.com/haifeng-jin/kt-legacy/blob/main/LICENSE
lark	https://github.com/lark-parser/lark	https://github.com/lark-parser/lark?tab=MIT-1-ov-file#readme
lazy_loader	https://github.com/scientific-python/lazy_loader	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
lcms2	https://www.littlecms.com	https://github.com/sk1project/python-lcms2/blob/master/LICENSE
legacy-cgi	https://github.com/jackrosenthal/legacy-cgi	https://github.com/jackrosenthal/legacy-cgi?tab=License-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lerc	https://github.com/Esri/lerc	https://github.com/Esri/lerc/blob/master/LICENSE
libarchive	http://www.libarchive.org	https://raw.githubusercontent.com/libarchive/libarchive/master/COPYING
libblas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
libbrotlicommon	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlidec	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlienc	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libcblas	https://anaconda.org/conda-forge/libcblas	N/A
libclang	https://github.com/sighingnow/libclang	https://github.com/sighingnow/libclang/blob/master/LICENSE.TXT
libcurl	https://curl.se/libcurl/	https://curl.se/docs/copyright.html
libcxx	https://github.com/llvm-mirror/libcxx	https://github.com/llvm-mirror/libcxx/blob/master/LICENSE.TXT
libdb	https://www.oracle.com/technology/software/products/berkeley-db/index.html	https://github.com/berkeleydb/libdb/blob/master/LEGAL
libdeflate	https://github.com/ebiggers/libdeflate	https://github.com/ebiggers/libdeflate/blob/master/COPYING
libedit	https://thrysoee.dk/editline/	http://www.netbsd.org/Goals/redistribution.html
libev	https://software.schmorp.de/pkg/libev.html	https://software.schmorp.de/pkg/libev.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libexpat	https://libexpat.github.io/	https://github.com/libexpat/libexpat?tab=MIT-1-ov-file#readme
libffi	https://github.com/libffi/libffi	https://github.com/libffi/libffi/blob/master/LICENSE
libgcc	https://gcc.gnu.org/onlinedocs/gccint/Libgcc.html	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libgcrypt	https://gnupg.org/software/index.html	https://git.gnupg.org/cgi-bin/gitweb.cgi?p=libgcrypt.git;a=blob;f=COPYING.LIB;hb=HEAD
libgd	https://libgd.github.io	https://libgd.github.io/manuals/2.2.1/files/license-txt.html
libglib	https://github.com/PolMine/libglib	https://github.com/PolMine/libglib/blob/master/include/glib-2.0/glib.h
libiconv	https://www.gnu.org/software/libiconv/	https://www.gnu.org/software/libiconv/#downloading
libint	https://tinyurl.com/yvw97wbw	https://tinyurl.com/yvw97wbw
liblapack	http://www.netlib.org/lapack/	https://www.netlib.org/lapack/LICENSE.txt
liblapacke	https://anaconda.org/conda-forge/liblapacke	https://www.netlib.org/lapack/LICENSE.txt
liblzma	https://tukaani.org/xz/	https://git.tukaani.org/?p=xz.git;a=blob;f=COPYING.0BSD
libmamba	https://medium.com/@QuantStack/open-software-packag	https://github.com/mamba-org/mamba/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libmambapy	https://www.anaconda.com/blog/a-faster-conda-for-a-g	https://github.com/conda/conda-libmamba-solv/blob/main/LICENSE
libnetcdf	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
libnghttp2	https://github.com/nghttp2/nghttp2	https://github.com/nghttp2/nghttp2/blob/master/COPYING
libnsl	https://github.com/thkukuk/libnsl	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libopenblas	https://www.openblas.net/	https://github.com/xianyi/OpenBLAS/blob/develop/LICENSE
libpng	https://www.libpng.org/pub/png/libpng.html	https://www.libpng.org/pub/png/src/libpng-LICENSE.txt
libpq	https://www.postgresql.org/	https://www.postgresql.org/about/licence/
librsvg	https://gitlab.gnome.org/GNOME/librsvg	https://gitlab.gnome.org/GNOME/librsvg/-/blob/main/COPYING.LIB
libsolv	https://github.com/openSUSE/libsolv	https://github.com/openSUSE/libsolv/blob/master/LICENSE.BSD
libsqlite	https://github.com/mackyle/sqlite	https://github.com/mackyle/sqlite?tab=License-1-ov-file#readme
libssh2	https://github.com/libssh2/libssh2	https://github.com/libssh2/libssh2/blob/master/COPYING
libstdcxx	https://gcc.gnu.org/	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libtiff	https://www.libtiff.org/	https://www.libtiff.org/misc.html
libtrust	https://github.com/docker/libtrust	https://github.com/docker/libtrust/blob/master/LICENSE
libuuid	https://sourceforge.net/projects/libuuid/	https://sourceforge.net/p/libuuid/code/ci/master/tree/COPYING
libuv	https://github.com/libuv/libuv	https://github.com/libuv/libuv/blob/v1.x/LICENSE
libwebp	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libwebp-base	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libxc	https://www.tddft.org/programs/libxc/	https://gitlab.com/libxc/libxc/-/blob/master/COPYING
libxcb	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-libxcb/blob/master/COPYING
libxcrypt	https://github.com/besser82/libxcrypt	https://github.com/besser82/libxcrypt?tab=LGPL-2.1-1-ov-file#readme
libxml2	https://git.gnome.org/browse/libxml2/	https://gitlab.gnome.org/GNOME/libxml2/-/blob/master/Copyright
libxmlsec1	https://github.com/lsh123/xmlsec	https://github.com/lsh123/xmlsec/blob/master/README
libxslt	https://gitlab.gnome.org/GNOME/libxslt/-/wikis/home	https://gitlab.gnome.org/GNOME/libxslt/-/blob/master/Copyright

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libzlib	https://zlib.net	https://zlib.net/zlib_license.html
lightning-utilities	https://github.com/Lightning-AI/utilities	https://github.com/Lightning-AI/utilities?tab=Apache-2.0-1-ov-file#readme
lime	https://github.com/marcotcr/lime	https://github.com/marcotcr/lime/blob/master/LICENSE
linear-operator	https://github.com/cornellius-gp/linear_operator	https://github.com/cornellius-gp/linear_operator?tab=MIT-1-ov-file#readme
lit	http://llvm.org	https://releases.llvm.org/14.0.0/LICENSE.TXT
llvm-openmp	https://github.com/llvm-mirror/openmp	https://github.com/llvm-mirror/openmp/blob/master/LICENSE.txt
llvmlite	http://llvmlite.readthedocs.io	https://github.com/numba/llvmlite/blob/main/LICENSE
loader-utils	https://github.com/webpack/loader-utils	https://github.com/webpack/loader-utils/raw/master/LICENSE
logistro	https://github.com/geopozo/logistro	https://github.com/geopozo/logistro?tab=MIT-1-ov-file#readme
logomaker	https://logomaker.readthedocs.io/en/latest/	https://github.com/jbkinney/logomaker/blob/master/LICENSE
logrus	https://github.com/sirupsen/logrus	https://github.com/sirupsen/logrus/blob/master/LICENSE
logrus-airbrake-hook.v2	https://gopkg.in/gemnasium/logrus-airbrake-hook.v2	https://github.com/gemnasium/logrus-airbrake-hook/blob/v2.1.2/LICENSE
lxml	https://lxml.de	https://lxml.de/index.html#license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lz4-c	https://www.lz4.org/	https://github.com/lz4/lz4/blob/dev/LICENSE
lzo	https://www.oberhumer.com/opensource/lzo/	https://www.oberhumer.com/opensource/gpl.html
mamba	https://github.com/mamba-org/mamba	https://github.com/mamba-org/mamba?tab=BSD-3-Clause-1-ov-f readme
markdown	https://github.com/evilstreak/markdown-js	https://www.opensource.org/licenses/mit-license.php
markdown-it-py	https://github.com/executablebooks/markdown-it-py	https://github.com/executablebooks/markdown-it-py/blob/master/LICENSE
markupsafe	https://github.com/pallets/markupsafe/tree/main/src/markupsafe	https://github.com/pallets/markupsafe/tree/main?tab=BSD-3-Clause-1-ov-f
MarkupSafe	https://palletsprojects.com/p/markupsafe/	https://github.com/pallets/markupsafe/blob/main/LICENSE.rst
mashumaro	https://github.com/Fatal1ty/mashumaro	https://github.com/Fatal1ty/mashumaro
matplotlib	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-base	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-inline	https://github.com/ipython/matplotlib-inline	https://github.com/ipython/matplotlib-inline/blob/master/LICENSE
mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mda_xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib?tab=License-1-ov-file#readme
mdtraj	https://www.mdtraj.org/	https://www.mdtraj.org/1.9.8.dev0/index.html#license
mdurl	https://github.com/markdown-it/mdurl	https://github.com/markdown-it/mdurl/blob/master/LICENSE
menuinst	https://github.com/conda/menuinst	https://github.com/conda/menuinst/blob/main/LICENSE.txt
mergo	https://github.com/imdario/mergo	https://github.com/imdario/mergo/blob/master/LICENSE
mistune	https://github.com/lepture/mistune	https://github.com/lepture/mistune/blob/master/LICENSE
mkl	https://github.com/rust-math/intel-mkl-src	https://github.com/rust-math/intel-mkl-src/blob/master/License.txt
mkl-fft	https://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft/blob/master/LICENSE.txt
mkl-random	https://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random/blob/master/LICENSE.txt
mkl-service	https://github.com/IntelPython/mkl-service	https://github.com/IntelPython/mkl-service/blob/master/LICENSE.txt
mkl_fft	http://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft?tab=BSD-3-Clause-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mkl_random	http://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random?tab=BSD-3-Clause-1-ov-f readme
ml-dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
ml_dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
modelcif	https://github.com/ihmwg/python-modelcif	https://github.com/ihmwg/python-modelcif/blob/main/LICENSE
molecupy	https://molecupy.readthedocs.io/en/latest/	https://github.com/samirelanduk/atomium/blob/master/LICENSE
moment	https://github.com/moment/moment	https://github.com/moment/moment/raw/master/LICENSE
moment-precise-range-plugin	https://github.com/codebox/moment-precise-range	https://github.com/codebox/moment-precise-range raw/master/LICENSE.md
more-itertools	https://github.com/more-itertools/more-itertools	https://github.com/more-itertools/more-itertools/blob/master/LICENSE
mpiplus	https://github.com/choderalab/mpiplus	https://github.com/choderalab/mpiplus/blob/main/LICENSE
mpmath	http://mpmath.org/	https://github.com/mpmath/mpmath/blob/master/LICENSE
mrcfile	https://github.com/ccpem/mrcfile	https://github.com/ccpem/mrcfile/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
msgpack	https://msgpack.org/	https://github.com/msgpack/msgpack-python/blob/main/COPYING
multidict	https://github.com/aio-libs/multidict	https://github.com/aio-libs/multidict/blob/master/LICENSE
multierr	https://go.uber.org/multierr	https://github.com/uber-go/multierr/blob/master/LICENSE.txt
multiprocess	https://github.com/uqfoundation/multiprocess	https://github.com/uqfoundation/multiprocess/blob/master/LICENSE
munkres	https://software.clapper.org/munkres/	https://github.com/bmc/munkres/blob/master/LICENSE.md
myesui uuid	https://github.com/myesui/uuid	https://pkg.go.dev/github.com/myesui/uuid?tab=licenses
namex	https://github.com/bcgov/namex	https://github.com/bcgov/namex/blob/main/LICENSE
narwhals	https://github.com/narwhals-dev/narwhals	https://github.com/narwhals-dev/narwhals?tab=MIT-1-ov-file#readme
nbclassic	http://jupyter.org	https://github.com/jupyter/nbclassic/blob/main/LICENSE
nbclient	https://jupyter.org	https://github.com/jupyter/nbclient/blob/main/LICENSE
nbconvert	https://jupyter.org	https://github.com/jupyter/nbconvert/blob/main/LICENSE
nbformat	http://jupyter.org	https://github.com/jupyter/nbformat/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ncurses	https://invisible-island.net/ncurses/	https://invisible-island.net/ncurses/ncurses-license.html
ndfes	https://rutgerslbsr.gitlab.io/fe-toolkit/	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
ndindex	https://quansight-labs.github.io/ndindex/	https://github.com/Quansight-Labs/ndindex?tab=MIT-1-ov-file#readme
nest-asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio/blob/master/LICENSE
nest_asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio?tab=BSD-2-Clause-1-ov-f readme
netcdf-fortran	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
netCDF4	http://github.com/Unidata/netcdf4-python	https://github.com/Unidata/netcdf4-python/blob/master/LICENSE
nettle	https://git.lysator.liu.se/nettle/nettle	https://git.lysator.liu.se/nettle/nettle/-/blob/master/AUTHORS?ref_type=heads
networkx	https://networkx.org	https://github.com/networkx/networkx/blob/main/LICENSE.txt
nfpm	https://github.com/goreleaser/nfpm	https://github.com/goreleaser/nfpm/blob/master/LICENSE.md
ng-tags-input	https://github.com/mbenford/ngTagsInput	https://github.com/mbenford/ngTagsInput
ng-toast	https://github.com/tameraydin/ngToast	https://github.com/tameraydin/ngToast

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ngdraggable	https://github.com/fatlinesofcode/ngDraggable	https://github.com/fatlinesofcode/ngDraggable/raw/master/LICENSE.txt
ngVue	https://github.com/ngVue/ngVue	https://github.com/ngVue/ngVue/raw/master/LICENSE
nh3	https://nh3.readthedocs.io/en/latest/	https://github.com/messense/nh3/blob/main/LICENSE
nlohmann_json	https://github.com/nlohmann/json	https://github.com/nlohmann/json?tab=MIT-1-ov-file#readme
nlopt	https://nlopt.readthedocs.io/en/latest/NLopt_Python_Reference/	https://nlopt.readthedocs.io/en/latest/NLopt_License_and_Copyright/
nodejs	https://nodejs.org/en/	https://raw.githubusercontent.com/nodejs/node/master/LICENSE
nomkl	https://github.com/conda-forge/nomkl-feedstock	https://github.com/conda-forge/nomkl-feedstock/blob/main/LICENSE.txt
notebook	http://jupyter.org	https://github.com/jupyter/notebook/blob/main/LICENSE
notebook-shim	https://github.com/jupyter/notebook_shim	https://github.com/jupyter/notebook_shim
notebook_shim	http://jupyter.org	https://github.com/jupyter/notebook_shim/blob/main/LICENSE
numba	https://numba.pydata.org	https://github.com/numba/numba/blob/main/LICENSE
numcpus	https://github.com/tklauser/numcpus	https://github.com/tklauser/numcpus/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
numexpr	https://github.com/pydata/numexpr	https://github.com/pydata/numexpr/blob/master/LICENSE.txt
numpy	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpy-base	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpydoc	https://numpydoc.readthedocs.io/en/latest/index.html	https://github.com/numpy/numpydoc/blob/main/LICENSE.txt
nvidia-cublas-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cublas-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cudnn-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html
nvidia-cudnn-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-cuEquivariance	https://github.com/NVIDIA/cuEquivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
nvidia-cufft-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufft-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufile-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html#
nvidia-curand-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-curand-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparselt-cu12	https://docs.nvidia.com/cuda/cusparselt/index.html	https://docs.nvidia.com/cuda/cusparselt/license.html
nvidia-ml-py	https://pypi.org/project/nvidia-ml-py/	https://pypi.org/project/nvidia-ml-py/
nvidia-nccl-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nccl-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nvjitlink-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-nvshmem-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
Oat++	https://oatpp.io/	https://github.com/oatpp/oatpp/blob/master/LICENSE
oauthlib	https://github.com/oauthlib/oauthlib	https://github.com/oauthlib/oauthlib/blob/master/LICENSE
ocl-icd	https://github.com/OCL-dev/ocl-icd	https://github.com/OCL-dev/ocl-icd/blob/master/COPYING
ocl-icd-system	https://github.com/conda-forge/ocl-icd-system-feedst	https://github.com/conda-forge/ocl-icd-system-feedst blob/main/LICENSE.txt
olefile	https://www.decalage.info/python/olefileio	https://olefile.readthedocs.io/en/latest/License.html
omegaconf	https://github.com/omry/omegaconf	https://github.com/omry/omegaconf/blob/master/LICENSE
OmegaFold	https://github.com/HeliXonProtein/OmegaFold/tree/main	https://github.com/HeliXonProtein/OmegaFold/blob/main/LICENSE
omnicanvas	https://omnicanvas.readthedocs.io/en/latest/	https://github.com/samirelanduk/omnicanvas/blob/master/LICENSE
OpenFF	https://openforcefield.org/	https://github.com/openforcefield/openff-forcefields
openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openff-forcefields	https://openforcefield.org	https://github.com/openforcefield/openff-forcefields/blob/master/LICENSE
openff-fragmenter	https://github.com/openforcefield/openff-fragmenter	https://github.com/openforcefield/openff-fragmenter/blob/main/LICENSE
openff-interchange	https://github.com/openforcefield/openff-interchange	https://github.com/openforcefield/openff-interchange/blob/main/LICENSE
openff-models	https://github.com/openforcefield/openff-models	https://github.com/openforcefield/openff-models/blob/main/LICENSE
openff-nagl	https://docs.openforcefield.org/projects/nagl/en/latest/?badge=latest	https://github.com/openforcefield/openff-nagl/blob/main/LICENSE
openff-nagl-models	https://github.com/openforcefield/openff-nagl-models	https://github.com/openforcefield/openff-nagl-models/blob/main/LICENSE
openff-toolkit	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-toolkit-base	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-units	https://github.com/openforcefield/openff-units	https://github.com/openforcefield/openff-units/blob/main/LICENSE
openff-utilities	https://github.com/openforcefield/openff-utilities	https://github.com/openforcefield/openff-utilities/blob/main/LICENSE
openjpeg	https://github.com/uclouvain/openjpeg	https://github.com/uclouvain/openjpeg/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openldap	https://www.openldap.org/software/repo.html	https://git.openldap.org/openldap/openldap/-/blob/master/LICENSE
OpenMM	https://openmm.org	https://docs.openmm.org/7.1.0/userguide/library.html#license
openmmtools	https://github.com/choderalab/openmmtools	https://github.com/choderalab/openmmtools/blob/main/LICENSE
openmoltools	https://github.com/choderalab/openmoltools	https://github.com/choderalab/openmoltools/blob/master/LICENSE
openpyxl	https://openpyxl.readthedocs.io/en/stable/	https://foss.heptapod.net/openpyxl/openpyxl/-/blob/branch/3.0/LICENCE.rst
openssl	https://www.openssl.org	https://github.com/openssl/openssl/blob/master/LICENSE.txt
opt-einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/master/LICENSE
opt_einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/main/LICENSE
OptKing	https://github.com/psi-rking/optking	https://github.com/psi-rking/optking/blob/master/LICENSE
optree	https://optree.readthedocs.io/en/latest/	https://github.com/metaopt/optree/blob/main/LICENSE
oscrypto	https://github.com/wbond/oscrypto	https://github.com/wbond/oscrypto/blob/master/LICENSE
overrides	https://github.com/mkorpela/overrides	https://github.com/mkorpela/overrides/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
packaging	https://github.com/pypa/packaging	https://github.com/pypa/packaging/blob/main/LICENSE
packmol	https://leandro.iqm.unicamp.br/m3g/packmol/home.shtml	https://github.com/m3g/packmol/blob/master/LICENSE
packmol_memgen	http://cpclab.uni-duesseldorf.de/	https://gitlab.com/cbjh/memgen/py-memgen/-/blob/main/LICENSE
pandas	https://pandas.pydata.org	https://github.com/pandas-dev/pandas/blob/main/LICENSE
pandocfilters	http://github.com/jgm/pandocfilters	https://github.com/jgm/pandocfilters/blob/master/LICENSE
panedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pango	https://pango.gnome.org	https://gitlab.gnome.org/GNOME/pango/-/blob/main/COPYING
ParmEd	https://parmed.github.io/ParmEd/html/index.html	https://github.com/ParmEd/ParmEd#license
parser	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
parso	https://parso.readthedocs.io/en/latest/	https://github.com/davidhalter/parso/blob/master/LICENSE.txt
pathos	https://github.com/uqfoundation/pathos	https://github.com/uqfoundation/pathos/blob/master/LICENSE
patsy	https://patsy.readthedocs.io/en/latest/	https://github.com/pydata/patsy/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pbkdf2	https://golang.org/x/crypto/pbkdf2	https://cs.opensource.google/go/x/crypto/+/master:LICENSE
pbr	https://docs.openstack.org/pbr/latest/	https://opendev.org/openstack/pbr/src/branch/master/LICENSE
pcmsolver	https://pcmsolver.readthedocs.io/en/stable/	https://github.com/PCMSolver/pcmsolver/blob/master/LICENSE
pcre	https://www.pcre.org	https://www.pcre.org/licence.txt
pcre2	https://www.pcre.org	https://github.com/PCRE2Project/pcre2/blob/master/LICENCE
pdb4amber	https://github.com/Amber-MD/pdb4amber	https://github.com/Amber-MD/pdb4amber/blob/master/LICENSE
pdbfixer	https://github.com/openmm/pdbfixer	https://github.com/openmm/pdbfixer/blob/master/LICENSE
pexpect	https://pexpect.readthedocs.io/	https://github.com/pexpect/pexpect/blob/master/LICENSE
pgconn	https://github.com/jackc/pgconn	https://github.com/jackc/pgconn/blob/master/LICENSE
pgio	https://github.com/jackc/pgio	https://github.com/jackc/pgio/blob/master/LICENSE
pgpassfile	https://github.com/jackc/pgpassfile	https://github.com/jackc/pgpassfile/blob/master/LICENSE
pgproto3	https://github.com/jackc/pgproto3/v2	https://github.com/jackc/pgproto3/blob/master/LICENSE
pgtype	https://github.com/jackc/pgtype	https://github.com/jackc/pgtype/blob/master/LICENSE
pgx	https://github.com/jackc/pgx/v4	https://github.com/jackc/pgx/blob/v4.0.1/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
phonopy	https://github.com/phonopy/phono3py	https://github.com/phonopy/phono3py/blob/develop/LICENSE
pickleshare	https://github.com/pickleshare/pickleshare	https://github.com/pickleshare/pickleshare/blob/master/LICENSE
Pillow	https://python-pillow.org	https://github.com/python-pillow/Pillow/blob/main/LICENSE
Pint	https://pint.readthedocs.io/en/stable/	https://github.com/hgrecco/pint/blob/master/LICENSE
pip	https://pip.pypa.io/	https://github.com/pypa/pip/blob/main/LICENSE.txt
pip-licenses	https://github.com/raimon49/pip-licenses	https://github.com/raimon49/pip-licenses/blob/master/LICENSE
pixman	https://pixman.org	https://cgit.freedesktop.org/pixman/tree/COPYING
pkginfo	https://launchpad.net/pkginfo	https://bazaar.launchpad.net/~tseaver/pkginfo/trunk/view/head:/LICENSE.txt
pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
pkgutil_resolve_name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
platformdirs	https://github.com/platformdirs/platformdirs	https://github.com/platformdirs/platformdirs/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
plotly	https://plotly.com/python/	https://github.com/plotly/plotly.py/blob/master/LICENSE.txt
plotly-orca	https://github.com/plotly/orca	https://github.com/plotly/orca/blob/master/LICENSE
plotly.js	https://github.com/plotly/plotly.js	https://github.com/plotly/plotly.js/raw/master/LICENSE
pluggy	https://pluggy.readthedocs.io/en/stable/index.html	https://github.com/pytest-dev/pluggy/blob/main/LICENSE
pooch	https://github.com/fatiando/pooch	https://github.com/fatiando/pooch/blob/main/LICENSE.txt
portalocker	https://github.com/WoLpH/portalocker	https://github.com/wolph/portalocker/blob/develop/LICENSE
pox	https://github.com/uqfoundation/pox	https://github.com/uqfoundation/pox/blob/master/LICENSE
ppft	https://github.com/uqfoundation/ppft	https://github.com/uqfoundation/ppft/blob/master/LICENSE
pq	https://github.com/lib/pq	https://github.com/lib/pq/blob/master/LICENSE.md
ProDy	http://www.csb.pitt.edu/ProDy	https://github.com/prody/ProDy/blob/master/LICENSE.rst
prometheus-client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prometheus_client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prompt-toolkit	https://python-prompt-toolkit.readthedocs.io/en/stable/	https://github.com/prompt-toolkit/python-prompt-toolkit/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
prompt_toolkit	https://github.com/prompt-toolkit/python-prompt-toolki	https://github.com/prompt-toolkit/python-prompt-toolki tab=BSD-3-Clause-1-ov-f readme
propcache	https://github.com/aio-libs/propcache	https://github.com/aio-libs/propcache?tab=Apache-2.0-1-ov-file#readme
protobuf	https://google.golang.org/protobuf	https://github.com/protocolbuffers/protobuf-go/blob/master/LICENSE
psi4	https://psicode.org	https://github.com/psi4/psi4/blob/master/COPYING.LESSER
psutil	https://psutil.readthedocs.io/en/latest/	https://github.com/giampaolo/psutil/blob/master/LICENSE
psycopg2	https://psycopg.org/	https://github.com/psycopg/psycopg2/blob/master/LICENSE
PTable	https://github.com/kxxoling/PTable	https://github.com/kxxoling/PTable/blob/master/COPYING
pthread-stubs	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-pthread-stubs/blob/master/COPYING
ptyprocess	https://ptyprocess.readthedocs.io/en/latest/	https://github.com/pexpect/ptyprocess/blob/master/LICENSE
pure-eval	http://github.com/alexmojaki/pure_eval	http://github.com/alexmojaki/pure_eval/blob/master/LICENSE.txt
pure_eval	http://github.com/alexmojaki/pure_eval	https://github.com/alexmojaki/pure_eval?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
py	https://py.readthedocs.io/en/latest/	https://github.com/pytest-dev/py/blob/master/LICENSE
py-cpuinfo	https://github.com/workhorsy/py-cpuinfo	https://github.com/workhorsy/py-cpuinfo/blob/master/LICENSE
pyasn1	https://github.com/etingof/pyasn1	https://github.com/etingof/pyasn1/blob/master/LICENSE.rst
pyasn1-modules	https://github.com/etingof/pyasn1-modules	https://github.com/etingof/pyasn1-modules/blob/master/LICENSE.txt
pyasn1_modules	https://pyasn1.readthedocs.io/en/latest/contents.html	https://github.com/pyasn1/pyasn1-modules?tab=BSD-2-Clause-1-ov-file#readme
pybind11	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11-abi	https://github.com/pybind/pybind11	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11_global	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pycairo	https://pycairo.readthedocs.io/en/latest/index.html	https://github.com/pygobject/pycairo/blob/master/COPYING
pycosat	https://github.com/conda/pycosat	https://github.com/conda/pycosat/blob/main/LICENSE
pycparser	https://github.com/eliben/pycparser	https://github.com/eliben/pycparser/blob/master/LICENSE
pydantic	https://pydantic-docs.helpmanual.io	https://github.com/samuelcolvin/pydantic/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pydantic-core	https://github.com/pydantic/pydantic-core	https://github.com/pydantic/pydantic-core
pyedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pyEMMA	http://github.com/markovmodel/PyEMMA	https://github.com/markovmodel/PyEMMA/blob/devel/LICENSE.txt
Pygments	https://pygments.org	https://github.com/pygments/pygments/blob/master/LICENSE
pygments	https://pygments.org/	https://github.com/pygments/pygments?tab=BSD-2-Clause-1-ov-file#readme
pygraphviz	https://pygraphviz.github.io	https://github.com/pygraphviz/pygraphviz/blob/main/LICENSE
pygtop	https://pygtop.readthedocs.io/en/latest/	https://github.com/samirelanduk/pygtop/blob/master/LICENSE
pyHanko	https://github.com/MatthiasValvekens/pyHanko	https://github.com/MatthiasValvekens/pyHanko/blob/master/LICENSE
pyhanko-certvalidator	https://github.com/MatthiasValvekens/certvalidator	https://github.com/MatthiasValvekens/certvalidator/blob/master/LICENSE
PyJWT	https://github.com/jpadilla/pyjwt	https://github.com/jpadilla/pyjwt/blob/master/LICENSE
pymbar	https://pymbar.org	https://github.com/choderalab/pymbar/blob/master/LICENSE
pynvml	https://github.com/gpuopenanalytics/pynvml	https://github.com/gpuopenanalytics/pynvml/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pyOpenSSL	https://pyopenssl.org/	https://github.com/pyca/pyopenssl/blob/main/LICENSE
pyparsing	https://pyparsing-docs.readthedocs.io/en/latest/	https://github.com/pyparsing/pyparsing/blob/master/LICENSE
PyPDF3	https://github.com/sfneal/PyPDF3	https://github.com/sfneal/PyPDF3/blob/master/LICENSE
pyrsistent	http://github.com/tobgu/pyrsistent/	https://github.com/tobgu/pyrsistent/blob/master/LICENSE.mit
pysam	https://github.com/pysam-developers/pysam	https://github.com/pysam-developers/pysam/blob/master/COPYING
PySocks	https://github.com/Anorov/PySocks	https://github.com/Anorov/PySocks/blob/master/LICENSE
pytables	https://www.pytables.org	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
python	https://www.python.org/	https://docs.python.org/3/license.html
python-bidi	https://github.com/MeirKriheli/python-bidi	https://github.com/MeirKriheli/python-bidi/blob/master/COPYING
python-constraint	https://github.com/python-constraint/python-constraint	https://github.com/python-constraint/python-constraint/blob/master/LICENSE
python-dateutil	https://dateutil.readthedocs.io	https://github.com/dateutil/dateutil/blob/master/LICENSE
python-json-logger	http://github.com/madzak/python-json-logger	https://github.com/madzak/python-json-logger/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
python-pptx	https://github.com/scanny/python-pptx	https://github.com/scanny/python-pptx/blob/master/LICENSE
python-ldap	https://www.python-ldap.org/	https://github.com/python-ldap/python-ldap/blob/python-ldap-3.4.0/LICENCE
python3-saml	https://github.com/onelogin/python3-saml	https://github.com/onelogin/python3-saml/blob/master/LICENSE
python_abi	https://github.com/conda-forge/python_abi-feedstock	https://github.com/conda-forge/python_abi-feedstock/blob/main/LICENSE.txt
pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning/blob/master/LICENSE
pytz	https://pythonhosted.org/pytz	https://git.launchpad.net/pytz/tree/LICENSE.txt
pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shim/blob/master/LICENSE
PyWavelets	https://github.com/PyWavelets/pywt	https://github.com/PyWavelets/pywt/blob/master/LICENSE
PyYAML	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
pyyml	No longer available	No longer available
pyzmq	https://pyzmq.readthedocs.io/en/latest/	https://github.com/zeromq/pyzmq/blob/main/COPYING.BSD
qcelemental	https://github.com/MolSSI/QCElemental	https://github.com/MolSSI/QCElemental/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
qcengine	https://github.com/MolSSI/QCEngine	https://github.com/MolSSI/QCEngine/blob/master/LICENSE
qrcode	https://github.com/lincolnloop/python-qrcode	https://github.com/lincolnloop/python-qrcode/blob/main/LICENSE
ramda	https://github.com/ramda/ramda	https://github.com/ramda/ramda/raw/master/LICENSE.txt
rapidjson	https://rapidjson.org/	https://github.com/Tencent/rapidjson/blob/master/license.txt
rdkit	https://www.rdkit.org	https://github.com/rdkit/rdkit/blob/master/license.txt
re2	https://github.com/google/re2	https://github.com/google/re2/blob/main/LICENSE
readme-renderer	https://github.com/pypa/readme_renderer	https://github.com/pypa/readme_renderer/blob/main/LICENSE
redis-py	https://github.com/andymccurdy/redis-py	https://github.com/redis/redis-py/blob/master/LICENSE
referencing	https://github.com/python-jsonschema/referencing	https://github.com/python-jsonschema/referencing
regex	https://github.com/mrabarnett/mrab-regex	https://github.com/mrabarnett/mrab-regex/blob/hg/LICENSE.txt
reportlab	https://www.reportlab.com	https://www.reportlab.com/dev/docs/faq/#1.3
reproc	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE
reproc-cpp	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
requests	https://requests.readthedocs.io	https://github.com/psf/requests/blob/main/LICENSE
requests-oauthlib	https://github.com/requests/requests-oauthlib	https://github.com/requests/requests-oauthlib/blob/master/LICENSE
requests-toolbelt	https://toolbelt.readthedocs.org	https://github.com/requests/toolbelt/blob/master/LICENSE
resumable	https://github.com/stevvooe/resumable	https://github.com/stevvooe/resumable/blob/master/LICENSE
retrying	https://github.com/rholder/retrying	https://github.com/rholder/retrying/blob/master/LICENSE
rfc3339-validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator/blob/master/LICENSE
rfc3339_validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator?tab=MIT-1-ov-file#readme
rfc3986	https://rfc3986.readthedocs.io/en/latest/	https://github.com/python-hyper/rfc3986/blob/main/LICENSE
rfc3986-validator	https://github.com/naimetti/rfc3986-validator	https://github.com/naimetti/rfc3986-validator/blob/master/LICENSE
rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax?tab=MIT-1-ov-file#readme
rich	https://github.com/Textualize/rich	https://github.com/Textualize/rich/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
rich-argparse	https://github.com/hamdanal/rich-argparse	https://github.com/hamdanal/rich-argparse/blob/main/LICENSE
rlPyCairo	https://pypi.org/project/rlPyCairo/	https://hg.reportlab.com/hg-public/rlPyCairo/file/tip/rlPyCairo/LICENSE.txt
RocksDict	https://github.com/rocksdict/RocksDict	https://github.com/rocksdict/RocksDict?tab=MIT-1-ov-file#readme
roman-numerals	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
roman-numerals-py	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
rotary-embedding-torch	https://github.com/lucidrains/rotary-embedding-tor	https://github.com/lucidrains/rotary-embedding-tor blob/main/LICENSE
rpds-py	https://github.com/crate-py/rpds	https://github.com/crate-py/rpds
rpmpack	https://github.com/google/rpmpack	https://github.com/google/rpmpack/blob/master/LICENSE
rsa	https://stuvel.eu/rsa	https://github.com/sybrenstuvel/python-rsa/blob/main/LICENSE
ruamel-yaml	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/LICENSE
ruamel.yaml.clib	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
s3transfer	https://github.com/boto/s3transfer	https://github.com/boto/s3transfer/blob/develop/LICENSE.txt
sacrebleu	https://github.com/mjpost/sacrebleu	https://github.com/mjpost/sacrebleu/blob/master/LICENSE.txt
sapiens	https://github.com/Merck/Sapiens	https://github.com/Merck/Sapiens/blob/main/LICENSE
sasl	https://mellium.im/sasl	https://github.com/mellium/sasl/blob/main/LICENSE
scikit-gstat	https://mmaelicke.github.io/scikit-gstat/	https://github.com/mmaelicke/scikit-gstat/blob/main/LICENSE
scikit-image	https://scikit-image.org	https://scikit-image.org/docs/dev/license.html
scikit-learn	https://scikit-learn.org/stable/	https://github.com/scikit-learn/scikit-learn/blob/main/COPYING
scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra/blob/main/LICENSE
scikit-network	https://github.com/sknetwork-team/scikit-network	https://github.com/sknetwork-team/scikit-network?tab=License-1-ov-file#readme
scipy	https://scipy.org	https://github.com/scipy/scipy/blob/main/LICENSE.txt
seaborn	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE
seaborn-base	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
semver	https://github.com/Masterminds/semver/v3	https://github.com/Masterminds/semver/blob/v3.0.3/LICENSE.txt
Send2Trash	https://github.com/arsenetar/send2trash	https://github.com/arsenetar/send2trash/blob/master/LICENSE
sentry-sdk	https://github.com/getsentry/sentry-python	https://github.com/getsentry/sentry-python/blob/master/LICENSE
setproctitle	https://github.com/dvarrazzo/py-setproctitle	https://github.com/dvarrazzo/py-setproctitle/blob/master/LICENSE
setuptools	https://github.com/pypa/setuptools	https://github.com/pypa/setuptools/blob/main/LICENSE
setuptools-scm	https://github.com/pypa/setuptools_scm/	https://github.com/pypa/setuptools_scm/blob/main/LICENSE
sh	https://github.com/amoffat/sh	https://github.com/amoffat/sh/blob/develop/LICENSE.txt
shellingham	https://github.com/sarugaku/shellingham	https://github.com/sarugaku/shellingham/blob/master/LICENSE
simdjson	https://github.com/simdjson/simdjson/tree/master	https://github.com/simdjson/simdjson/blob/master/LICENSE
simint	https://www.bennyp.org/research/simint/	https://github.com/simint-chem/simint-generator/blob/master/LICENSE
simplejson	https://github.com/simplejson/simplejson	https://github.com/simplejson/simplejson/blob/master/LICENSE.txt
six	https://github.com/benjaminp/six	https://github.com/benjaminp/six/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
smirnoff99frosst	https://github.com/openforcefield/smirnoff99frosst	https://github.com/openforcefield/smirnoff99Frosst/blob/master/LICENSE
smmap	https://github.com/gitpython-developers/smmap	https://github.com/gitpython-developers/smmap/blob/master/LICENSE
snappy	https://github.com/google/snappy	https://github.com/google/snappy/blob/main/COPYING
sniffio	https://github.com/python-trio/sniffio	https://github.com/python-trio/sniffio/blob/master/LICENSE.MIT
snowballstemmer	https://github.com/snowballstem/snowball	https://github.com/snowballstem/snowball/blob/master/COPYING
soupsieve	https://github.com/facelessuser/soupsieve	https://github.com/facelessuser/soupsieve/blob/main/LICENSE.md
spdlog	https://github.com/gabime/spdlog	https://github.com/gabime/spdlog?tab=License-1-ov-file#readme
spglib	https://spglib.readthedocs.io/	https://github.com/spglib/spglib/blob/develop/COPYING
sphinx	https://github.com/sphinx-doc/sphinx	https://github.com/sphinx-doc/sphinx/blob/5.x/LICENSE
sphinxcontrib-applehelp	https://github.com/sphinx-doc/sphinxcontrib-applehe	https://github.com/sphinx-doc/sphinxcontrib-applehe blob/master/LICENSE
sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp blob/master/LICENSE
sphinxcontrib-htmlhelp	https://github.com/sphinx-doc/sphinxcontrib-htmlhel	https://github.com/sphinx-doc/sphinxcontrib-htmlhel blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath/blob/master/LICENSE
sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp/blob/master/LICENSE
sphinxcontrib-serializinghtml	https://github.com/sphinx-doc/sphinxcontrib-serializ	https://github.com/sphinx-doc/sphinxcontrib-serializ blob/master/LICENSE
SQLAlchemy	https://www.sqlalchemy.org	https://docs.sqlalchemy.org/en/14/copyright.html
SQLAlchemy	https://www.sqlalchemy.org/	https://github.com/sqlalchemy/sqlalchemy?tab=MIT-1-ov-file#readme
sqlite	https://sqlite.org/index.html	https://sqlite.org/copyright.html
sqlparse	https://github.com/andialbrecht/sqlparse	https://github.com/andialbrecht/sqlparse/blob/master/LICENSE
stack-data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data/blob/master/LICENSE.txt
stack_data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data?tab=MIT-1-ov-file#readme
starfile	https://github.com/alisterburt/starfile	https://github.com/teamtomo/starfile/blob/main/LICENSE
statsmodels	https://github.com/statsmodels/statsmodels	https://github.com/statsmodels/statsmodels/blob/main/LICENSE.txt
structlog	https://www.structlog.org/	https://github.com/hynek/structlog/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
streamlit	https://github.com/streamlit/streamlit	https://github.com/streamlit/streamlit/blob/develop/LICENSE
svglib	https://github.com/deeplook/svglib	https://github.com/deeplook/svglib/blob/master/LICENSE.txt
sympy	https://sympy.org	https://github.com/sympy/sympy/blob/master/LICENSE
tables	https://www.pytables.org/	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
tabulate	https://github.com/astanin/python-tabulate	https://github.com/astanin/python-tabulate/blob/master/LICENSE
tbb	https://github.com/oneapi-src/oneTBB	https://github.com/oneapi-src/oneTBB/blob/master/LICENSE.txt
tenacity	https://github.com/jd/tenacity	https://github.com/jd/tenacity/blob/master/LICENSE
tensorboard	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-data-server	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-plugin-wit	https://github.com/pair-code/what-if-tool	https://github.com/PAIR-code/what-if-tool/blob/master/LICENSE
tensorboardX	https://github.com/lanpa/tensorboardX	https://github.com/lanpa/tensorboardX/blob/master/LICENSE
tensorflow	https://github.com/tensorflow/tensorflow	https://github.com/tensorflow/tensorflow/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tensorflow-estimator	https://github.com/tensorflow/estimator	https://github.com/tensorflow/estimator/blob/master/LICENSE
tensorflow-io-gcs-filesystem	https://github.com/tensorflow/io	https://github.com/tensorflow/io/blob/master/LICENSE
tensorflow-probability	https://github.com/tensorflow/probability	https://github.com/tensorflow/probability/blob/main/LICENSE
termcolor	https://github.com/hugovk/termcolor	https://github.com/hugovk/termcolor/blob/main/COPYING.txt
terminado	https://github.com/jupyter/terminado	https://github.com/jupyter/terminado/blob/main/LICENSE
testpath	https://github.com/jupyter/testpath	https://github.com/jupyter/testpath/blob/master/LICENSE
textangular	https://github.com/fraywing/textAngular	https://github.com/fraywing/textAngular
texttable	https://github.com/foutaise/texttable/	https://github.com/foutaise/texttable/?tab=MIT-1-ov-file#readme
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
threadpoolctl	https://github.com/joblib/threadpoolctl	https://github.com/joblib/threadpoolctl/blob/master/LICENSE
three	https://github.com/mrdoob/three.js	https://github.com/mrdoob/three.js/raw/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tifffile	https://github.com/cgohlke/tifffile/	https://github.com/cgohlke/tifffile/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

AddColorer
    OEShape::OEColorForceField, 56
AddColorGaussian
    OEShape::OEShapeQuery, 118
AddColorGaussians
    OEShape::OEShapeQuery, 118
AddCutoff
    OEShape::OEROCSOptions, 107
AddDuplicateSmarts
    OEShape::OEOverlapPrepOptions, 94
AddInteraction
    OEShape::OEColorForceField, 56
AddMolecule
    OEShape::OEROCS, 106
AddScore
    OEShape::OEBestOverlayResults, 52
    OEShape::OEROCSResult, 110
AddShapeGaussian
    OEShape::OEShapeQuery, 119
AddShapeGaussians
    OEShape::OEShapeQuery, 119
AddType
    OEShape::OEColorForceField, 57

B

besthit.java
    Example Code, 21
BestOverlay
    OEShape::OEFlexiOverlay, 69
    OEShape::OEHermiteOverlay, 142
    OEShape::OEMultiRefOverlay, 86
    OEShape::OEOverlay, 98

C

Clear
    OEShape::OEBestOverlayResults, 52
    OEShape::OEColorForceField, 57
    OEShape::OEShapeQuery, 119
ClearColorForceField
    OEShape::OEBestOverlay, 146
ClearColorGaussians
    OEShape::OEShapeQuery, 119
ClearCutoffs
    OEShape::OEROCSOptions, 107
ClearDuplicateSmarts
    OEShape::OEOverlapPrepOptions, 94
ClearInteractions
    OEShape::OEColorForceField, 57
ClearMolecule
    OEShape::OEShapeQuery, 119
ClearMolecules
    OEShape::OEROCS, 106
ClearShapeGaussians
    OEShape::OEShapeQuery, 119
ClearShapeGrid
    OEShape::OEShapeQuery, 119
ClearStartPoints
    OEShape::OECartesianStarts, 55
ClearStarts
    OEShape::OEQuatStarts, 102
ClearUserStarts
    OEShape::OEBestOverlay, 146
coloroverlap.java
    Example Code, 15
ColorScore
    OEShape::OEColorOverlap, 150
colorscore.java
    Example Code, 17
Constructor
    OEShape::OEFlexiOverlapOptions, 65
    OEShape::OEFlexiOverlapResults, 68
    OEShape::OEFlexiOverlay, 69
    OEShape::OEFlexiOverlayOptions, 70
    OEShape::OEOverlapPrepOptions, 92
Constructors
    OEShape::OEAnalyticColorFunc, 48
    OEShape::OEAnalyticOptions, 48
    OEShape::OEAnalyticShapeFunc, 50
    OEShape::OEAsIsStarts, 51
    OEShape::OEAtAtomStarts, 52
    OEShape::OEBestOverlay, 146
    OEShape::OEBestOverlayResults, 52
    OEShape::OEBestOverlayScore, 53

OEShape::OECartesianStarts, 54
OEShape::OEColorFFParameter, 56
OEShape::OEColorForceField, 56
OEShape::OEColorOptions, 59
OEShape::OEColorOverlap, 150
OEShape::OEColorResults, 151
OEShape::OEColorTanimotoCutoff, 60
OEShape::OEExactColorFunc, 61
OEShape::OEExactShapeFunc, 62
OEShape::OEFitColorTverskyCutoff, 63
OEShape::OEFitTverskyComboCutoff, 63
OEShape::OEFitTverskyCutoff, 64
OEShape::OEGridColorFunc, 72
OEShape::OEGridShapeFunc, 74
OEShape::OEHermite, 74
OEShape::OEHermiteOptions, 76
OEShape::OEHermiteOverlay, 142
OEShape::OEHermiteOverlayOptions, 141
OEShape::OEHermiteShapeFunc, 79
OEShape::OEHighestTverskyCombo, 83
OEShape::OEInertialStarts, 84
OEShape::OEIsColorAtomPred, 85
OEShape::OEMultiRefOverlay, 85
OEShape::OEOverlap, 152
OEShape::OEOverlapCutoff, 86
OEShape::OEOverlapFunc, 88
OEShape::OEOverlapPrep, 91
OEShape::OEOverlapResults, 95
OEShape::OEOverlay, 98
OEShape::OEOverlayOptions, 99
OEShape::OEQuatStarts, 102
OEShape::ORandomStarts, 104
OEShape::OERefColorTverskyCutoff, 103
OEShape::OERefTverskyComboCutoff, 103
OEShape::OERefTverskyCutoff, 104
OEShape::OEROCS, 105
OEShape::OEROCSOptions, 107
OEShape::OEROCSResult, 110
OEShape::OEShapeGridOptions, 112
OEShape::OEShapeOptions, 113
OEShape::OEShapeQuery, 118
OEShape::OEShapeQueryOptions, 139
OEShape::OEShapeQueryPublic, 155
OEShape::OESubrocsStarts, 123
OEShape::OETanimotoComboCutoff, 124
OEShape::OETanimotoCutoff, 124
CreateCopy
OEShape::OEHighestColorTanimoto, 79
OEShape::OEHighestComboScore, 152
OEShape::OEHighestFitColorTversky, 80
OEShape::OEHighestFitTversky, 80
OEShape::OEHighestFitTverskyCombo, 80
OEShape::OEHighestOverlap, 81
OEShape::OEHighestRefColorTversky, 81
OEShape::OEHighestRefTversky, 81
OEShape::OEHighestRefTverskyCombo, 82
OEShape::OEHighestScaledColor, 82
OEShape::OEHighestTanimoto, 82
OEShape::OEHighestTanimotoCombo, 83
OEShape::OEHighestTverskyCombo, 83
OEShape::OEIsColorAtomPred, 85
OEShape::OEOverlayScoreCutoff, 102
OEShape::OEStarts, 115
CreateGrid
OEShape::OEHermite, 75
createquery.java
Example Code, 31

D

DeleteColorGaussian
OEShape::OEShapeQuery, 119
DeleteShapeGaussian
OEShape::OEShapeQuery, 120

E

Example Code
besthit.java, 21
coloroverlap.java, 15
colorscore.java, 17
createquery.java, 31
excludedvolume.java, 44
excludedvolume.txt, 46
hermiteexpansion.java, 35
hermitetanimoto.java, 37
nxnshape.java, 40
overlapcolor.java, 20
overlapquery.java, 32
overlayallres.java, 27
overlaybestres.java, 25
overlaystruct.java, 29
rescore.java, 14
shapeprops.java, 42
simpleoverlap.java, 13
tophits.java, 22
tophitsconf.java, 24
tophitsquery.java, 33
usercolor.java, 18
excludedvolume.java
Example Code, 44
excludedvolume.txt
Example Code, 46

G

GetAllColor
OEShape::OEColorOverlap, 150
GetAnalyticOptions
OEShape::OEAnalyticColorFunc, 48
OEShape::OEAnalyticShapeFunc, 51

GetAssignBondiRadii
    OEShape::OEOverlapPrep, 91
    OEShape::OEOverlapPrepOptions, 92
GetAssignColor
    OEShape::OEOverlapPrep, 91
    OEShape::OEOverlapPrepOptions, 93
GetAtomStrength
    OEShape::OEShapeQueryBase, 121
GetCarbonRadius
    OEShape::OEBestOverlay, 146
    OEShape::OEOverlap, 153
    OEShape::OEShapeOptions, 113
GetCoefficients
    OEShape::OEHermite, 75
GetColorForceField
    OEShape::OEColorOptions, 59
    OEShape::OEOverlapPrep, 91
    OEShape::OEOverlapPrepOptions, 93
    OEShape::OEShapeQueryBase, 121
    OEShape::OEShapeQueryPublic, 155
GetColorFunc
    OEShape::OEOverlapFunc, 88
GetColorFuncType
    OEShape::OEFlexiOverlapOptions, 66
    OEShape::OEOverlayOptions, 99
GetColorGaussianRadius
    OEShape::OEShapeQueryOptions, 140
GetColorGaussians
    OEShape::OEShapeQueryBase, 122
GetColorOptions
    OEShape::OEColorFunc, 59
    OEShape::OEFlexiOverlapOptions, 66
    OEShape::OEOverlayOptions, 100
GetColorScore
    OEShape::OEOverlapResults, 95
GetColorTanimoto
    OEShape::OEOverlapResults, 95
GetColorTversky
    OEShape::OEOverlapResults, 96
GetCompositeMolecule
    OEShape::OEShapeQueryBase, 122
GetConfsPerHit
    OEShape::OEROCSOptions, 107
GetDerivativeType
    OEShape::OEShapeGridOptions, 112
GetDuplicateDistance
    OEShape::OEOverlapPrepOptions, 93
GetDuplicateSmarts
    OEShape::OEOverlapPrepOptions, 94
GetExpType
    OEShape::OEAnalyticOptions, 49
GetFitColorTversky
    OEShape::OEOverlapResults, 96
GetFitConfIdx
    OEShape::OEBestOverlayScore, 53
GetFitHermiteOptions
    OEShape::OEHermiteShapeFunc, 79
GetFitSelfColor
    OEShape::OEOverlapResults, 96
GetFitSelfOverlap
    OEShape::OEOverlapResults, 96
GetFitTversky
    OEShape::OEColorResults, 151
    OEShape::OEOverlapResults, 96
GetFitTverskyCombo
    OEShape::OEOverlapResults, 96
GetFlexiOverlapOptions
    OEShape::OEFlexiOverlayOptions, 70
GetForceField
    OEShape::OEFlexiOverlapOptions, 66
GetGridOptions
    OEShape::OEGridColorFunc, 73
    OEShape::OEGridShapeFunc, 74
GetGridSpacing
    OEShape::OEOverlap, 153
    OEShape::OEShapeGridOptions, 112
GetHarmonic
    OEShape::OEFlexiOverlapOptions, 66
GetHermiteOptions
    OEShape::OEHermiteOverlayOptions, 141
GetInitialOrientation
    OEShape::OEBestOverlay, 146
GetInternalEnergy
    OEShape::OEFlexiOverlapResults, 68
GetLambdaX
    OEShape::OEHermiteOptions, 76
GetLambdaY
    OEShape::OEHermiteOptions, 76
GetLambdaZ
    OEShape::OEHermiteOptions, 76
GetMaxHits
    OEShape::OEROCSOptions, 107
GetMaxOptSteps
    OEShape::OEFlexiOverlayOptions, 70
    OEShape::OEHermiteOverlayOptions, 141
    OEShape::OEOverlayOptions, 100
GetMaxRandomTranslation
    OEShape::OEBestOverlay, 146
    OEShape::OERandomStarts, 104
GetMethod
    OEShape::OEBestOverlay, 146
    OEShape::OEOverlap, 153
GetMinimizeType
    OEShape::OEBestOverlay, 147
GetMolecule
    OEShape::OEShapeQueryBase, 122
GetMultiplier
    OEShape::OEColorOptions, 59

GetNPolyMax
    OEShape::OEHermiteOptions, 76
GetNumBestHits
    OEShape::OEROCSOptions, 108
GetNumOfStarts
    OEShape::OEStarts, 115
GetNumRandomStarts
    OEShape::OEBestOverlay, 147
    OEShape::OERandomStarts, 105
GetNumStartPoints
    OEShape::OECartesianStarts, 55
GetNumUserStarts
    OEShape::OEBestOverlay, 147
GetOptions
    OEShape::OEHermite, 75
GetOverlap
    OEShape::OEOverlapResults, 96
GetOverlapFunc
    OEShape::OEFlexiOverlapOptions, 66
    OEShape::OEOverlayOptions, 100
GetOverlayConf
    OEShape::OEROCSResult, 111
GetOverlayConfs
    OEShape::OEROCSResult, 111
GetOverlayOptions
    OEShape::OEMultiRefOverlay, 86
    OEShape::OEOverlay, 98
    OEShape::OEROCSOptions, 108
GetPerformPrep
    OEShape::OEROCSOptions, 108
GetProxyGridCutoff
    OEShape::OEAnalyticOptions, 49
GetRadiiApproximation
    OEShape::OEBestOverlay, 147
    OEShape::OEOverlap, 153
    OEShape::OEShapeOptions, 113
GetRandomSeed
    OEShape::OEBestOverlay, 147
    OEShape::OERandomStarts, 105
GetRankPredicate
    OEShape::OEROCSOptions, 108
GetRefColorTversky
    OEShape::OEOverlapResults, 96
GetRefConfIdx
    OEShape::OEBestOverlayScore, 53
GetRefGrid
    OEShape::OEBestOverlay, 147
GetRefHermiteOptions
    OEShape::OEHermiteShapeFunc, 79
GetRefMol
    OEShape::OEBestOverlay, 147
GetRefSelfColor
    OEShape::OEBestOverlay, 147
    OEShape::OEOverlapResults, 96
GetRefSelfOverlap
    OEShape::OEOverlapResults, 97
GetRefSymmetry
    OEShape::OEBestOverlay, 147
GetRefTversky
    OEShape::OEColorResults, 151
    OEShape::OEOverlapResults, 97
GetRefTverskyCombo
    OEShape::OEOverlapResults, 97
GetRelativeGridContourVal
    OEShape::OEShapeQueryOptions, 139
GetRemoveColor
    OEShape::OEOverlapPrepOptions, 93
GetRemoveDuplicate
    OEShape::OEOverlapPrepOptions, 93
GetRigidOverlay
    OEShape::OEFlexiOverlayOptions, 70
GetRigidOverlayOptions
    OEShape::OEFlexiOverlayOptions, 71
GetROCSOptions
    OEShape::OEROCS, 106
GetRotMatrix
    OEShape::OEBestOverlayScore, 53
GetScores
    OEShape::OEBestOverlayResults, 52
GetScoreType
    OEShape::OEColorOptions, 59
    OEShape::OEShapeOptions, 114
    OEShape::OEShapeQueryOptions, 140
GetSelfColor
    OEShape::OEColorOverlap, 151
GetSelfOverlap
    OEShape::OEHermite, 75
GetShapeFunc
    OEShape::OEOverlapFunc, 88
GetShapeFuncType
    OEShape::OEFlexiOverlapOptions, 65
    OEShape::OEOverlayOptions, 100
GetShapeGaussianRadius
    OEShape::OEShapeQueryOptions, 140
GetShapeGaussians
    OEShape::OEShapeQueryBase, 122
GetShapeGrid
    OEShape::OEShapeQueryBase, 122
GetShapeOptions
    OEShape::OEFlexiOverlapOptions, 66
    OEShape::OEOverlayOptions, 100
    OEShape::OEShapeFunc, 112
GetStartPoints
    OEShape::OECartesianStarts, 55
GetStarts
    OEShape::OEHermiteOverlayOptions, 141
    OEShape::OEOverlayOptions, 100
    OEShape::OEStarts, 116

GetSymmetryThreshold
    OEShape::OEBestOverlay, 148
    OEShape::OEInertialStarts, 84
GetTanimoto
    OEShape::OEColorResults, 152
    OEShape::OEOverlapResults, 97
GetTanimotoCombo
    OEShape::OEOverlapResults, 97
GetTitle
    OEShape::OEColorForceField, 57
    OEShape::OEShapeQueryBase, 122
    OEShape::OEShapeQueryPublic, 155
GetTranslation
    OEShape::OEBestOverlayScore, 54
GetTversky
    OEShape::OEColorResults, 152
    OEShape::OEOverlapResults, 97
GetType
    OEShape::OEColorForceField, 57
GetTypeName
    OEShape::OEColorForceField, 57
GetTypes
    OEShape::OEColorForceField, 57
GetUseHydrogens
    OEShape::OEBestOverlay, 148
    OEShape::OEOverlap, 153
    OEShape::OEOverlapPrep, 91
    OEShape::OEOverlapPrepOptions, 93
GetUseMaxHits
    OEShape::OEROCSOptions, 108
GetUseOptimalLambdas
    OEShape::OEHermiteOptions, 76
GetUseProxyGrid
    OEShape::OEAnalyticOptions, 49
GetUserStarts
    OEShape::OEBestOverlay, 148

H

HasColor
    OEShape::OEShapeQueryBase, 122
HasCutoff
    OEShape::OEROCSOptions, 108
HasMolecule
    OEShape::OEShapeQueryBase, 122
HasShape
    OEShape::OEShapeQueryBase, 123
HasShapeGrid
    OEShape::OEShapeQueryBase, 123
hermiteexpansion.java
    Example Code, 35
hermitetanimoto.java
    Example Code, 37

I

Init
    OEShape::OEColorForceField, 57
IsEmpty
    OEShape::OEShapeQueryBase, 123

M

MeetsCutoff
    OEShape::OEROCSOptions, 108

N

NumCoefficients
    OEShape::OEHermite, 75
nxnshape.java
    Example Code, 40

O

OEShape::OEAddColorAtom, 128
OEShape::OEAddColorAtoms, 129
OEShape::OEAddCoordsToColorAtom, 129
OEShape::OEAnalyticColorFunc, 47
    Constructors, 48
    GetAnalyticOptions, 48
OEShape::OEAnalyticOptions, 48
    Constructors, 48
    GetExpType, 49
    GetProxyGridCutoff, 49
    GetUseProxyGrid, 49
    SetExpType, 49
    SetProxyGridCutoff, 49
    SetUseProxyGrid, 49
OEShape::OEAnalyticShapeFunc, 49
    Constructors, 50
    GetAnalyticOptions, 51
OEShape::OEAsIsStarts, 51
    Constructors, 51
OEShape::OEAtAtomStarts, 51
    Constructors, 52
OEShape::OEBestOverlay, 145
    ClearColorForceField, 146
    ClearUserStarts, 146
    Constructors, 146
    GetCarbonRadius, 146
    GetInitialOrientation, 146
    GetMaxRandomTranslation, 146
    GetMethod, 146
    GetMinimizeType, 147
    GetNumRandomStarts, 147
    GetNumUserStarts, 147
    GetRadiiApproximation, 147
    GetRandomSeed, 147
    GetRefGrid, 147
    GetRefMol, 147

GetRefSelfColor, 147
GetRefSymmetry, 147
GetSymmetryThreshold, 148
GetUseHydrogens, 148
GetUserStarts, 148
Overlay, 148
SetCarbonRadius, 148
SetColorForceField, 148
SetColorOptimize, 148
SetInitialOrientation, 148
SetMaxRandomTranslation, 149
SetMethod, 149
SetMinimizeType, 149
SetNumRandomStarts, 149
SetRadiiApproximation, 149
SetRandomSeed, 149
SetRefGrid, 149
SetRefMol, 149
SetSymmetryThreshold, 150
SetUseHydrogens, 150
SetUserStarts, 150
OEShape::OEBestOverlayResults, 52
AddScore, 52
Clear, 52
Constructors, 52
GetScores, 52
OEShape::OEBestOverlayScore, 52
Constructors, 53
GetFitConfIdx, 53
GetRefConfIdx, 53
GetRotMatrix, 53
GetTranslation, 54
Transform, 54
OEShape::OEBOMinType, 155
OEShape::OEBOMinType::Autoscale, 155
OEShape::OEBOMinType::Default, 156
OEShape::OEBOMinType::Overlap, 155
OEShape::OEBOMinType::Tanimoto, 155
OEShape::OEBOOrientation, 156
OEShape::OEBOOrientation::AsIs, 156
OEShape::OEBOOrientation::Default, 156
OEShape::OEBOOrientation::Inertial, 156
OEShape::OEBOOrientation::InertialAtColorAtoms, 156
OEShape::OEBOOrientation::InertialAtHeavyAtoms, 156
OEShape::OEBOOrientation::Random, 156
OEShape::OEBOOrientation::Subrocs, 156
OEShape::OEBOOrientation::UserInertialStarts, 156
OEShape::OECalcShapeMultipoles, 129
OEShape::OECalcVolume, 129
OEShape::OECartesianStarts, 54
ClearStartPoints, 55
Constructors, 54
GetNumStartPoints, 55
GetStartPoints, 55
SetStartPoints, 55
OEShape::OEClearCachedSelfColor, 130
OEShape::OEClearCachedSelfShape, 130
OEShape::OEColorFFParameter, 55
Constructors, 56
OEShape::OEColorFFType, 124
OEShape::OEColorFFType::Custom, 125
OEShape::OEColorFFType::ExplicitMillsDean, 125
OEShape::OEColorFFType::ExplicitMillsDeanNoRings, 125
OEShape::OEColorFFType::ImplicitMillsDean, 125
OEShape::OEColorFFType::ImplicitMillsDeanNoRings, 125
OEShape::OEColorFFType::Max, 125
OEShape::OEColorFFType::OEDefault, 125
OEShape::OEColorFFType::Undefined, 124
OEShape::OEColorForceField, 56
AddColorer, 56
AddInteraction, 56
AddType, 57
Clear, 57
ClearInteractions, 57
Constructors, 56
GetTitle, 57
GetType, 57
GetTypeName, 57
GetTypes, 57
Init, 57
Ready, 57
SetTitle, 57
Write, 57
OEShape::OEColorFunc, 58
GetColorOptions, 59
OEShape::OEColorOptions, 59
Constructors, 59
GetColorForceField, 59
GetMultiplier, 59
GetScoreType, 59
SetColorForceField, 59
SetMultiplier, 60
SetScoreType, 60
OEShape::OEColorOverlap, 150
ColorScore, 150
Constructors, 150
GetAllColor, 150
GetSelfColor, 151
SetAllColor, 151
SetColorForceField, 151
SetFitMol, 151

SetRefMol, 151
OEShape::OEColorResults, 151
Constructors, 151
GetFitTversky, 151
GetRefTversky, 151
GetTanimoto, 152
GetTversky, 152
OEShape::OEColorTanimotoCutoff, 60
Constructors, 60
OEShape::OEColorType, 128
OEShape::OEColorType::Analytic, 128
OEShape::OEColorType::Exact, 128
OEShape::OEColorType::Grid, 128
OEShape::OEColorType::None, 128
OEShape::OECompressColorAtoms, 130
OEShape::OECountColorAtoms, 130
OEShape::OEDeleteCompressedColorAtoms, 130
OEShape::OEDerivativeType, 125
OEShape::OEDerivativeType::Interpolated, 125
OEShape::OEDerivativeType::NearestValue, 125
OEShape::OEExactColorFunc, 60
Constructors, 61
OEShape::OEExactShapeFunc, 61
Constructors, 62
OEShape::OEExponentialType, 126
OEShape::OEExponentialType::Standard, 126
OEShape::OEExponentialType::Tabulated, 126
OEShape::OEFitColorTverskyCutoff, 63
Constructors, 63
OEShape::OEFitTverskyComboCutoff, 63
Constructors, 63
OEShape::OEFitTverskyCutoff, 63
Constructors, 64
OEShape::OEFlexiOverlapFunc, 64
Overlap, 64
SetupRef, 64
UpdateColorCoords, 65
OEShape::OEFlexiOverlapOptions, 65
Constructor, 65
GetColorFuncType, 66
GetColorOptions, 66
GetForceField, 66
GetHarmonic, 66
GetOverlapFunc, 66
GetShapeFuncType, 65
GetShapeOptions, 66
SetColorFuncType, 67
SetColorOptions, 67
SetForceField, 67
SetHarmonic, 67
SetOverlapFunc, 67
SetShapeFuncType, 66
SetShapeOptions, 67
Validate, 65
OEShape::OEFlexiOverlapResults, 67
Constructor, 68
GetInternalEnergy, 68
OEShape::OEFlexiOverlay, 68
BestOverlay, 69
Constructor, 69
Overlay, 69
SetupRef, 69
SortedOverlay, 69
OEShape::OEFlexiOverlayOptions, 70
Constructor, 70
GetFlexiOverlapOptions, 70
GetMaxOptSteps, 70
GetRigidOverlay, 70
GetRigidOverlayOptions, 71
SetFlexiOverlapOptions, 71
SetMaxOptSteps, 71
SetRigidOverlay, 71
SetRigidOverlayOptions, 71
OEShape::OEGetCachedSelfColor, 130
OEShape::OEGetCachedSelfShape, 131
OEShape::OEGetCenterAndExtents, 131
OEShape::OEGetColorAtoms, 131
OEShape::OEGetColorParents, 131
OEShape::OEGetColorType, 131
OEShape::OEGetMomentsOfInertiaTransform, 132
OEShape::OEGridColorFunc, 71
Constructors, 72
GetGridOptions, 73
OEShape::OEGridScoreType, 143
OEShape::OEGridScoreType::CosAngle, 143
OEShape::OEGridScoreType::Tanimoto, 143
OEShape::OEGridShapeFunc, 73
Constructors, 74
GetGridOptions, 74
OEShape::OEHasCachedSelfColor, 132
OEShape::OEHasCachedSelfShape, 132
OEShape::OEHasColorAtoms, 132
OEShape::OEHasCompressedColorAtoms, 132
OEShape::OEHasPerceivedColorAtoms, 133
OEShape::OEHermite, 74
Constructors, 74
CreateGrid, 75
GetCoefficients, 75
GetOptions, 75
GetSelfOverlap, 75
NumCoefficients, 75
Setup, 75
OEShape::OEHermiteOptions, 76
Constructors, 76
GetLambdaX, 76
GetLambdaY, 76
GetLambdaZ, 76
GetNPolyMax, 76

GetUseOptimalLambdas, 76
SetLambdaX, 77
SetLambdaY, 77
SetLambdaZ, 77
SetNPolyMax, 77
SetUseOptimalLambdas, 77
OEShape::OEHermiteOverlay, 142
BestOverlay, 142
Constructors, 142
SetupRef, 143
OEShape::OEHermiteOverlayOptions, 140
Constructors, 141
GetHermiteOptions, 141
GetMaxOptSteps, 141
GetStarts, 141
SetHermiteOptions, 141
SetMaxOptSteps, 141
SetStarts, 142
OEShape::OEHermiteShapeFunc, 77
Constructors, 79
GetFitHermiteOptions, 79
GetRefHermiteOptions, 79
OEShape::OEHighestColorTanimoto, 79
CreateCopy, 79
operator(), 79
OEShape::OEHighestComboScore, 152
CreateCopy, 152
operator(), 152
OEShape::OEHighestFitColorTversky, 79
CreateCopy, 80
operator(), 80
OEShape::OEHighestFitTversky, 80
CreateCopy, 80
operator(), 80
OEShape::OEHighestFitTverskyCombo, 80
CreateCopy, 80
operator(), 80
OEShape::OEHighestOverlap, 80
CreateCopy, 81
operator(), 80
OEShape::OEHighestRefColorTversky, 81
CreateCopy, 81
operator(), 81
OEShape::OEHighestRefTversky, 81
CreateCopy, 81
operator(), 81
OEShape::OEHighestRefTverskyCombo, 81
CreateCopy, 82
operator(), 81
OEShape::OEHighestScaledColor, 82
CreateCopy, 82
operator(), 82
OEShape::OEHighestTanimoto, 82
CreateCopy, 82
operator(), 82
OEShape::OEHighestTanimotoCombo, 82
CreateCopy, 83
operator(), 82
OEShape::OEHighestTverskyCombo, 83
Constructors, 83
CreateCopy, 83
operator(), 83
OEShape::OEInertialStarts, 83
Constructors, 84
GetSymmetryThreshold, 84
SetSymmetryThreshold, 85
OEShape::OEIsColorAtom, 133
OEShape::OEIsColorAtomPred, 85
Constructors, 85
CreateCopy, 85
operator(), 85
OEShape::OEIsFastROCSShapeQuery, 133
OEShape::OEMol2Query, 133
OEShape::OEMomentSymmetry, 126
OEShape::OEMomentSymmetry::NoSymmetry, 126
OEShape::OEMomentSymmetry::Oblate, 126
OEShape::OEMomentSymmetry::Prolate, 126
OEShape::OEMomentSymmetry::Spherical, 126
OEShape::OEMomentSymmetry::Unknown, 126
OEShape::OEMultiRefOverlay, 85
BestOverlay, 86
Constructors, 85
GetOverlayOptions, 86
Overlay, 86
SetupRef, 86
OEShape::OEOrientByMomentsOfInertia, 133
OEShape::OEOverlap, 152
Constructors, 152
GetCarbonRadius, 153
GetGridSpacing, 153
GetMethod, 153
GetRadiiApproximation, 153
GetUseHydrogens, 153
Overlap, 153
SetCarbonRadius, 153
SetFitMol, 153
SetGridSpacing, 154
SetMethod, 154
SetRadiiApproximation, 154
SetRefGrid, 154
SetRefMol, 154
SetUseHydrogens, 154
OEShape::OEOverlapCutoff, 86
Constructors, 86
OEShape::OEOverlapFunc, 87
Constructors, 88
GetColorFunc, 88
GetShapeFunc, 88

OEShape::OEOverlapFuncBase, 88
Overlap, 90
SetupFlex, 90
SetupRef, 90
OEShape::OEOverlapMethod, 156
OEShape::OEOverlapMethod::Analytic, 157
OEShape::OEOverlapMethod::Analytic2, 157
OEShape::OEOverlapMethod::Exact, 157
OEShape::OEOverlapMethod::Grid, 156
OEShape::OEOverlapMethod::Max, 157
OEShape::OEOverlapPrep, 90
Constructors, 91
GetAssignBondiRadii, 91
GetAssignColor, 91
GetColorForceField, 91
GetUseHydrogens, 91
Prep, 91
SetAssignBondiRadii, 91
SetAssignColor, 91
SetColorForceField, 92
SetUseHydrogens, 92
OEShape::OEOverlapPrepOptions, 92
AddDuplicateSmarts, 94
ClearDuplicateSmarts, 94
Constructor, 92
GetAssignBondiRadii, 92
GetAssignColor, 93
GetColorForceField, 93
GetDuplicateDistance, 93
GetDuplicateSmarts, 94
GetRemoveColor, 93
GetRemoveDuplicate, 93
GetUseHydrogens, 93
SetAssignBondiRadii, 93
SetAssignColor, 93
SetColorForceField, 94
SetDuplicateDistance, 94
SetRemoveColor, 94
SetRemoveDuplicate, 94
SetUseHydrogens, 94
OEShape::OEOverlapRadii, 126
OEShape::OEOverlapRadii::All, 127
OEShape::OEOverlapRadii::Carbon, 126
OEShape::OEOverlapRadii::Default, 127
OEShape::OEOverlapRadii::Max, 127
OEShape::OEOverlapRepresentation, 157
OEShape::OEOverlapRepresentation::All, 157
OEShape::OEOverlapRepresentation::Atomic, 157
OEShape::OEOverlapRepresentation::Atoms, 157
OEShape::OEOverlapRepresentation::Bonding, 157
OEShape::OEOverlapRepresentation::Bonds, 157
OEShape::OEOverlapRepresentation::Default, 157
OEShape::OEOverlapRepresentation::Full, 157
OEShape::OEOverlapRepresentation::Gaussian, 157
OEShape::OEOverlapRepresentation::Max, 158
OEShape::OEOverlapResults, 95
Constructors, 95
GetColorScore, 95
GetColorTanimoto, 95
GetColorTversky, 96
GetFitColorTversky, 96
GetFitSelfColor, 96
GetFitSelfOverlap, 96
GetFitTversky, 96
GetFitTverskyCombo, 96
GetOverlap, 96
GetRefColorTversky, 96
GetRefSelfColor, 96
GetRefSelfOverlap, 97
GetRefTversky, 97
GetRefTverskyCombo, 97
GetTanimoto, 97
GetTanimotoCombo, 97
GetTversky, 97
OEShape::OEOverlay, 97
BestOverlay, 98
Constructors, 98
GetOverlayOptions, 98
Overlay, 98
SetupRef, 98
SortedOverlay, 99
OEShape::OEOverlayOptions, 99
Constructors, 99
GetColorFuncType, 99
GetColorOptions, 100
GetMaxOptSteps, 100
GetOverlapFunc, 100
GetShapeFuncType, 100
GetShapeOptions, 100
GetStarts, 100
SetColorFuncType, 100
SetColorOptions, 100
SetMaxOptSteps, 101
SetOverlapFunc, 101
SetShapeFuncType, 101
SetShapeOptions, 101
SetStarts, 101
OEShape::OEOverlayScoreCutoff, 101
CreateCopy, 102
operator(), 101
OEShape::OEQuatStarts, 102
ClearStarts, 102
Constructors, 102
SetStarts, 102
OEShape::OERandomStarts, 104

Constructors, 104
GetMaxRandomTranslation, 104
GetNumRandomStarts, 105
GetRandomSeed, 105
SetMaxRandomTranslation, 105
SetNumRandomStarts, 105
SetRandomSeed, 105
OEShape::OEReadShapeQuery, 134
OEShape::OERefColorTverskyCutoff, 103
Constructors, 103
OEShape::OERefTverskyComboCutoff, 103
Constructors, 103
OEShape::OERefTverskyCutoff, 103
Constructors, 104
OEShape::OERemoveColorAtoms, 134
OEShape::OEROCS, 105
AddMolecule, 106
ClearMolecules, 106
Constructors, 105
GetROCSOptions, 106
Overlay, 106
SetDatabase, 106
OEShape::OEROCSOptions, 106
AddCutoff, 107
ClearCutoffs, 107
Constructors, 107
GetConfsPerHit, 107
GetMaxHits, 107
GetNumBestHits, 108
GetOverlayOptions, 108
GetPerformPrep, 108
GetRankPredicate, 108
GetUseMaxHits, 108
HasCutoff, 108
MeetsCutoff, 108
SetConfsPerHit, 108
SetMaxHits, 109
SetNumBestHits, 109
SetOverlayOptions, 109
SetPerformPrep, 109
SetRankPredicate, 109
SetUseMaxHits, 109
OEShape::OEROCSOverlay, 134
OEShape::OEROCSResult, 109
AddScore, 110
Constructors, 110
GetOverlayConf, 111
GetOverlayConfs, 111
OEShape::OEScoreType, 127
OEShape::OEScoreType::Overlap, 127
OEShape::OEScoreType::Tanimoto, 127
OEShape::OESelfColor, 134
OEShape::OESelfShape, 134
OEShape::OESetCachedSelfColor, 135
OEShape::OESetCachedSelfShape, 135
OEShape::OESetColorParents, 135
OEShape::OESetColorType, 136
OEShape::OESetCoordsFromColorParents, 136
OEShape::OEShapeFunc, 111
GetShapeOptions, 112
OEShape::OEShapeGetArch, 136
OEShape::OEShapeGetLicensee, 136
OEShape::OEShapeGetPlatform, 136
OEShape::OEShapeGetRelease, 136
OEShape::OEShapeGetSite, 136
OEShape::OEShapeGetVersion, 137
OEShape::OEShapeGridOptions, 112
Constructors, 112
GetDerivativeType, 112
GetGridSpacing, 112
SetDerivativeType, 113
SetGridSpacing, 113
OEShape::OEShapeIsLicensed, 137
OEShape::OEShapeOptions, 113
Constructors, 113
GetCarbonRadius, 113
GetRadiiApproximation, 113
GetScoreType, 114
SetCarbonRadius, 114
SetRadiiApproximation, 114
SetScoreType, 114
OEShape::OEShapeQuery, 116
AddColorGaussian, 118
AddColorGaussians, 118
AddShapeGaussian, 119
AddShapeGaussians, 119
Clear, 119
ClearColorGaussians, 119
ClearMolecule, 119
ClearShapeGaussians, 119
ClearShapeGrid, 119
Constructors, 118
DeleteColorGaussian, 119
DeleteShapeGaussian, 120
SetAtomStrength, 120
SetMolecule, 120
SetShapeGrid, 120
OEShape::OEShapeQueryBase, 120
GetAtomStrength, 121
GetColorForceField, 121
GetColorGaussians, 122
GetCompositeMolecule, 122
GetMolecule, 122
GetShapeGaussians, 122
GetShapeGrid, 122
GetTitle, 122
HasColor, 122
HasMolecule, 122

HasShape, 123
HasShapeGrid, 123
IsEmpty, 123
SetTitle, 123
OEShape::OEShapeQueryOptions, 139
Constructors, 139
GetColorGaussianRadius, 140
GetRelativeGridContourVal, 139
GetScoreType, 140
GetShapeGaussianRadius, 140
SetColorGaussianRadius, 140
SetRelativeGridContourVal, 140
SetScoreType, 140
SetShapeGaussianRadius, 140
OEShape::OEShapeQueryPublic, 154
Constructors, 155
GetColorForceField, 155
GetTitle, 155
operator bool, 155
OEShape::OEShapeType, 127
OEShape::OEShapeType::Analytic, 127
OEShape::OEShapeType::Exact, 127
OEShape::OEShapeType::Grid, 127
OEShape::OEShapeType::Hermite, 128
OEShape::OEShapeType::None, 128
OEShape::OESortOverlayScores, 137
OEShape::OEStarts, 114
CreateCopy, 115
GetNumOfStarts, 115
GetStarts, 116
Setup, 116
SetupFrame, 116
SetupRef, 116
OEShape::OESubrocsStarts, 123
Constructors, 123
OEShape::OETanimotoComboCutoff, 124
Constructors, 124
OEShape::OETanimotoCutoff, 124
Constructors, 124
OEShape::OEUncompressColorAtoms, 137
OEShape::OEWriteShapeQuery, 138
operator bool
OEShape::OEShapeQueryPublic, 155
operator()
OEShape::OEHighestColorTanimoto, 79
OEShape::OEHighestComboScore, 152
OEShape::OEHighestFitColorTversky, 80
OEShape::OEHighestFitTversky, 80
OEShape::OEHighestFitTverskyCombo, 80
OEShape::OEHighestOverlap, 80
OEShape::OEHighestRefColorTversky, 81
OEShape::OEHighestRefTversky, 81
OEShape::OEHighestRefTverskyCombo, 81
OEShape::OEHighestScaledColor, 82
OEShape::OEHighestTanimoto, 82
OEShape::OEHighestTanimotoCombo, 82
OEShape::OEHighestTverskyCombo, 83
OEShape::OEIsColorAtomPred, 85
OEShape::OEOverlayScoreCutoff, 101
Overlap
OEShape::OEFlexiOverlapFunc, 64
OEShape::OEOverlap, 153
OEShape::OEOverlapFuncBase, 90
overlapcolor.java
Example Code, 20
overlapquery.java
Example Code, 32
Overlay
OEShape::OEBestOverlay, 148
OEShape::OEFlexiOverlay, 69
OEShape::OEMultiRefOverlay, 86
OEShape::OEOverlay, 98
OEShape::OEROCS, 106
overlayallres.java
Example Code, 27
overlaybestres.java
Example Code, 25
overlaystruct.java
Example Code, 29

P

Prep
OEShape::OEOverlapPrep, 91

R

Ready
OEShape::OEColorForceField, 57
rescore.java
Example Code, 14

S

SetAllColor
OEShape::OEColorOverlap, 151
SetAssignBondiRadii
OEShape::OEOverlapPrep, 91
OEShape::OEOverlapPrepOptions, 93
SetAssignColor
OEShape::OEOverlapPrep, 91
OEShape::OEOverlapPrepOptions, 93
SetAtomStrength
OEShape::OEShapeQuery, 120
SetCarbonRadius
OEShape::OEBestOverlay, 148
OEShape::OEOverlap, 153
OEShape::OEShapeOptions, 114
SetColorForceField
OEShape::OEBestOverlay, 148
OEShape::OEColorOptions, 59

OEShape::OEColorOverlap, 151
OEShape::OEOverlapPrep, 92
OEShape::OEOverlapPrepOptions, 94
SetColorFuncType
OEShape::OEFlexiOverlapOptions, 67
OEShape::OEOverlayOptions, 100
SetColorGaussianRadius
OEShape::OEShapeQueryOptions, 140
SetColorOptimize
OEShape::OEBestOverlay, 148
SetColorOptions
OEShape::OEFlexiOverlapOptions, 67
OEShape::OEOverlayOptions, 100
SetConfsPerHit
OEShape::OEROCSOptions, 108
SetDatabase
OEShape::OEROCS, 106
SetDerivativeType
OEShape::OEShapeGridOptions, 113
SetDuplicateDistance
OEShape::OEOverlapPrepOptions, 94
SetExpType
OEShape::OEAnalyticOptions, 49
SetFitMol
OEShape::OEColorOverlap, 151
OEShape::OEOverlap, 153
SetFlexiOverlapOptions
OEShape::OEFlexiOverlayOptions, 71
SetForceField
OEShape::OEFlexiOverlapOptions, 67
SetGridSpacing
OEShape::OEOverlap, 154
OEShape::OEShapeGridOptions, 113
SetHarmonic
OEShape::OEFlexiOverlapOptions, 67
SetHermiteOptions
OEShape::OEHermiteOverlayOptions, 141
SetInitialOrientation
OEShape::OEBestOverlay, 148
SetLambdaX
OEShape::OEHermiteOptions, 77
SetLambdaY
OEShape::OEHermiteOptions, 77
SetLambdaZ
OEShape::OEHermiteOptions, 77
SetMaxHits
OEShape::OEROCSOptions, 109
SetMaxOptSteps
OEShape::OEFlexiOverlayOptions, 71
OEShape::OEHermiteOverlayOptions, 141
OEShape::OEOverlayOptions, 101
SetMaxRandomTranslation
OEShape::OEBestOverlay, 149
OEShape::OERandomStarts, 105
SetMethod
OEShape::OEBestOverlay, 149
OEShape::OEOverlap, 154
SetMinimizeType
OEShape::OEBestOverlay, 149
SetMolecule
OEShape::OEShapeQuery, 120
SetMultiplier
OEShape::OEColorOptions, 60
SetNPolyMax
OEShape::OEHermiteOptions, 77
SetNumBestHits
OEShape::OEROCSOptions, 109
SetNumRandomStarts
OEShape::OEBestOverlay, 149
OEShape::OERandomStarts, 105
SetOverlapFunc
OEShape::OEFlexiOverlapOptions, 67
OEShape::OEOverlayOptions, 101
SetOverlayOptions
OEShape::OEROCSOptions, 109
SetPerformPrep
OEShape::OEROCSOptions, 109
SetProxyGridCutoff
OEShape::OEAnalyticOptions, 49
SetRadiiApproximation
OEShape::OEBestOverlay, 149
OEShape::OEOverlap, 154
OEShape::OEShapeOptions, 114
SetRandomSeed
OEShape::OEBestOverlay, 149
OEShape::OERandomStarts, 105
SetRankPredicate
OEShape::OEROCSOptions, 109
SetRefGrid
OEShape::OEBestOverlay, 149
OEShape::OEOverlap, 154
SetRefMol
OEShape::OEBestOverlay, 149
OEShape::OEColorOverlap, 151
OEShape::OEOverlap, 154
SetRelativeGridContourVal
OEShape::OEShapeQueryOptions, 140
SetRemoveColor
OEShape::OEOverlapPrepOptions, 94
SetRemoveDuplicate
OEShape::OEOverlapPrepOptions, 94
SetRigidOverlay
OEShape::OEFlexiOverlayOptions, 71
SetRigidOverlayOptions
OEShape::OEFlexiOverlayOptions, 71
SetScoreType
OEShape::OEColorOptions, 60
OEShape::OEShapeOptions, 114

OEShape::OEShapeQueryOptions, 140
SetShapeFuncType
OEShape::OEFlexiOverlapOptions, 66
OEShape::OEOverlayOptions, 101
SetShapeGaussianRadius
OEShape::OEShapeQueryOptions, 140
SetShapeGrid
OEShape::OEShapeQuery, 120
SetShapeOptions
OEShape::OEFlexiOverlapOptions, 67
OEShape::OEOverlayOptions, 101
SetStartPoints
OEShape::OECartesianStarts, 55
SetStarts
OEShape::OEHermiteOverlayOptions, 142
OEShape::OEOverlayOptions, 101
OEShape::OEQuatStarts, 102
SetSymmetryThreshold
OEShape::OEBestOverlay, 150
OEShape::OEInertialStarts, 85
SetTitle
OEShape::OEColorForceField, 57
OEShape::OEShapeQueryBase, 123
Setup
OEShape::OEHermite, 75
OEShape::OEStarts, 116
SetupFlex
OEShape::OEOverlapFuncBase, 90
SetupFrame
OEShape::OEStarts, 116
SetupRef
OEShape::OEFlexiOverlapFunc, 64
OEShape::OEFlexiOverlay, 69
OEShape::OEHermiteOverlay, 143
OEShape::OEMultiRefOverlay, 86
OEShape::OEOverlapFuncBase, 90
OEShape::OEOverlay, 98
OEShape::OEStarts, 116
SetUseHydrogens
OEShape::OEBestOverlay, 150
OEShape::OEOverlap, 154
OEShape::OEOverlapPrep, 92
OEShape::OEOverlapPrepOptions, 94
SetUseMaxHits
OEShape::OEROCSOptions, 109
SetUseOptimalLambdas
OEShape::OEHermiteOptions, 77
SetUseProxyGrid
OEShape::OEAnalyticOptions, 49
SetUserStarts
OEShape::OEBestOverlay, 150
shapeprops.java
Example Code, 42
simpleoverlap.java
Example Code, 13
SortedOverlay
OEShape::OEFlexiOverlay, 69
OEShape::OEOverlay, 99

T

tophits.java
Example Code, 22
tophitsconf.java
Example Code, 24
tophitsquery.java
Example Code, 33
Transform
OEShape::OEBestOverlayScore, 54

U

UpdateColorCoords
OEShape::OEFlexiOverlapFunc, 65
usercolor.java
Example Code, 18

V

Validate
OEShape::OEFlexiOverlapOptions, 65

W

Write
OEShape::OEColorForceField, 57