NAME

RDKitGenerateConformers.py - Generate molecular conformations

SYNOPSIS

RDKitGenerateConformers.py [--alignConformers <yes or no>] [--addHydrogens <yes or no>] [--conformerGenerator <SDG, ETDG, KDG, ETKDG>] [--embedRMSDCutoff <number>] [--enforceChirality <yes or no>] [--energyRMSDCutoff <number>] [--energyWindow <number>] [--forceField <UFF, MMFF, None>] [--infileParams <Name,Value,...>] [--maxConfs <number>] [--maxIters <number>] [--outfileParams <Name,Value,...>] [--overwrite] [--removeHydrogens <yes or no>] [--randomSeed <number>] [-w <dir>] -i <infile> -o <outfile>

RDKitGenerateConformers.py -h | --help | -e | --examples

DESCRIPTION

Generate molecular conformations using a combination of distance geometry and forcefield minimization. The forcefield minimization may be skipped to only generate conformations by available distance geometry based methodologies.

The supported input file formats are: Mol (.mol), SD (.sdf, .sd), SMILES (.smi, .csv, .tcsv .txt)

The supported output file format are: SD (.sdf, .sd)

OPTIONS

-a, --addHydrogens <yes or no> [default: yes]

Add hydrogens before minimization.

--alignConformers <yes or no> [default: yes]

Align conformers for each molecule.

-c, --conformerGenerator <SDG, ETDG, KDG, ETKDG> [default: ETKDG]

Conformation generation methodology for generating initial 3D coordinates of a molecule. Possible values: Standard Distance Geometry, (SDG), Experimental Torsion-angle preference with Distance Geometry (ETDG) [Ref 129] , basic Knowledge-terms with Distance Geometry (KDG), and Experimental Torsion-angle preference along with basic Knowledge-terms and Distance Geometry (ETKDG).

--embedRMSDCutoff <number> [default: none]

RMSD cutoff for retaining conformations after embedding and before energy minimization. All embedded conformations are kept by default. Otherwise, only those conformations which are different from each other by the specified RMSD cutoff are kept. The first embedded conformation is always retained.

--enforceChirality <yes or no> [default: Yes]

Enforce chirality for defined chiral centers.

--energyRMSDCutoff <number> [default: none]

RMSD cutoff for retaining conformations after energy minimization. By default, all minimized conformations with in the specified energy window from the lowest energy conformation are kept. Otherwise, only those conformations which are different from the lowest energy conformation by the specified RMSD cutoff and are with in the specified energy window are kept. The lowest energy conformation is always retained.

--energyWindow <number> [default: 20]

Energy window in kcal/mol for selecting conformers. This option is ignored during 'None' value of '-f, --forcefield' option.

-e, --examples

Print examples.

-f, --forceField <UFF, MMFF, None> [default: MMFF]

Forcefield method to use for energy minimization. Possible values: Universal Force Field (UFF) [Ref 81], Merck Molecular Mechanics Force Field (MMFF) [Ref 83-87] or None.

-h, --help

Print this help message.

-i, --infile <infile>

Input file name.

--infileParams <Name,Value,...> [default: auto]

A comma delimited list of parameter name and value pairs for reading molecules from files. The supported parameter names for different file formats, along with their default values, are shown below:

```
SD, MOL: removeHydrogens,yes,sanitize,yes,strictParsing,yes
SMILES: smilesColumn,1,smilesNameColumn,2,smilesDelimiter,space,
    smilesTitleLine,auto,sanitize,yes
```

Possible values for smilesDelimiter: space, comma or tab.

--maxConfs <number> [default: auto]

Maximum number of conformations to generate for each molecule by conformation generation methodology. The conformations are minimized using the specified forcefield as needed and written to the output file. The default value for maximum number of conformations is dependent on the number of rotatable bonds in molecules: RotBonds <= 5, maxConfs = 100; RotBonds >=6 and <= 10, MaxConfs = 200; RotBonds >= 11, maxConfs = 300

--maxIters <number> [default: 250]

Maximum number of iterations to perform for each molecule during forcefield minimization.

-o, --outfile <outfile>

Output file name.

--outfileParams <Name,Value,...> [default: auto]

A comma delimited list of parameter name and value pairs for writing molecules to files. The supported parameter names for different file formats, along with their default values, are shown below:

```
SD: kekulize,no
```

--overwrite

Overwrite existing files.

-r, --removeHydrogens <yes or no> [default: Yes]

Remove hydrogens after minimization.

--randomSeed <number> [default: auto]

Seed for the random number generator for reproducing 3D coordinates. Default is to use a random seed.

-w, --workingdir <dir>

Location of working directory which defaults to the current directory.

EXAMPLES

To generate conformers using Experimental Torsion-angle preference along with basic Knowledge-terms and Distance Geometry (ETKDG) followed by MMFF minimization with automatic determination of maximum number of conformers for each molecule and write out a SD file, type:

```
% RDKitGenerateConformers.py  -i Sample.smi -o SampleOut.sdf
```

To generate up to 150 conformers for each molecule using ETKDG and UFF forcefield minimization along with conformers within 25 kcal/mol energy window and write out a SD file, type:

```
% RDKitGenerateConformers.py  --energyWindow 25 -f UFF --maxConfs 150
  -i Sample.smi -o SampleOut.sdf
```

To generate up to 50 conformers for each molecule using KDG without any forcefield minimization and alignment of conformers and write out a SD file, type:

```
% RDKitGenerateConformers.py  -f none --maxConfs 50 --alignConformers no
  -i Sample.sdf -o SampleOut.sdf
```

To generate up to 50 conformers using SDG without any forcefield minimization and alignment of conformers for molecules in a CSV SMILES file, SMILES strings in column 1, name in column 2, and write out a SD file, type:

```
% RDKitGenerateConformers.py  --maxConfs 50  --maxIters 50 -c SDG
  --alignConformers no -f none --infileParams "smilesDelimiter,comma,
  smilesTitleLine,yes, smilesColumn,1,smilesNameColumn,2"
  -i SampleSMILES.csv -o SampleOut.sdf
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

RDKitCalculateRMSD.py, RDKitCalculateMolecularDescriptors.py, RDKitCompareMoleculeShapes.py, RDKitConvertFileFormat.py, RDKitPerformMinimization.py

COPYRIGHT

The functionality available in this script is implemented using RDKit, an open source toolkit for cheminformatics developed by Greg Landrum.

This file is part of MayaChemTools.

MayaChemTools is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.