NAME

RDKitFilterTorsionStrainEnergyAlerts.py - Filter torsion strain energy library alerts

SYNOPSIS

RDKitFilterTorsionStrainEnergyAlerts.py [--alertsMode <TotalEnergy, MaxSingleEnergy, or TotalOrMaxSingleEnergy>] [--alertsMaxSingleEnergyCutoff <Number>] [--alertsTotalEnergyCutoff <Number>] [--filterTorsionsNotObserved <yes or no>] [--infileParams <Name,Value,...>] [--mode <filter or count>] [--mp <yes or no>] [--mpParams <Name,Value,...>] [--outfileAlerts <yes or no>] [--outfileAlertsMode <All or AlertsOnly>] [--outfileFiltered <yes or no>] [--outfilesFilteredByRules <yes or no>] [--outfilesFilteredByRulesMaxCount <All or number>] [--outfileSummary <yes or no>] [--outfileSDFieldLabels <Type,Label,...>] [--outfileParams <Name,Value,...>] [--overwrite] [--precision <number>] [--rotBondsSMARTSMode <NonStrict, SemiStrict,...>] [--rotBondsSMARTSPattern <SMARTS>] [--torsionEnergyLibraryFile <FileName or auto>] [-w <dir>] -i <infile> -o <outfile>

RDKitFilterTorsionStrainEnergyAlerts.py [--torsionEnergyLibraryFile <FileName or auto>] -l | --list

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

DESCRIPTION

Filter strained molecules from an input file for torsion strain energy library [Ref 153] alerts by matching rotatable bonds against SMARTS patterns specified for torsion rules in a torsion energy library file and write out appropriate molecules to output files. The molecules must have 3D coordinates in input file. The default torsion strain energy library file, TorsionStrainEnergyLibrary.xml, is available under MAYACHEMTOOLS/lib/python/TorsionAlerts directory.

The data in torsion strain energy library file is organized in a hierarchical manner. It consists of one generic class and six specific classes at the highest level. Each class contains multiple subclasses corresponding to named functional groups or substructure patterns. The subclasses consist of torsion rules sorted from specific to generic torsion patterns. The torsion rule, in turn, contains a list of peak values for torsion angles and two tolerance values. A pair of tolerance values define torsion bins around a torsion peak value.

A strain energy calculation method, 'exact' or 'approximate' [Ref 153], is associated with each torsion rule for calculating torsion strain energy. The 'exact' stain energy calculation relies on the energy bins available under the energy histogram consisting of 36 bins covering angles from -180 to 180. The width of each bin is 10 degree. The energy bins are are defined at the right end points. The first and the last energy bins correspond to -170 and 180 respectively. The torsion angle is mapped to a energy bin. An angle offset is calculated for the torsion angle from the the right end point angle of the bin. The strain energy is estimated for the angle offset based on the energy difference between the current and previous bins. The torsion strain energy, in terms of torsion energy units (TEUs), corresponds to the sum of bin strain energy and the angle offset strain energy.

```
Energy = BinEnergyDiff/10.0 * BinAngleOffset + BinEnergy[BinNum]

Where:

BinEnergyDiff = BinEnergy[BinNum] - BinEnergy[PreviousBinNum]
BinAngleOffset = TorsionAngle - BinAngleRightSide
```

The 'approximate' strain energy calculation relies on the angle difference between a torsion angle and the torsion peaks observed for the torsion rules in the torsion energy library. The torsion angle is matched to a torsion peak based on the value of torsion angle difference. It must be less than or equal to the value for the second tolerance 'tolerance2'. Otherwise, the torsion angle is not observed in the torsion energy library and a value of 'NA' is assigned for torsion energy along with the lower and upper bounds on energy at 95% confidence interval. The 'approximate' torsion energy (TEUs) for observed torsion angle is calculated using the following formula:

```
Energy = beta_1 * (AngleDiff ** 2) + beta_2 * (AngleDiff ** 4)
```

The coefficients 'beta_1' and 'beta_2' are available for the observed angles in the torsion strain energy library. The 'AngleDiff' is the difference between the torsion angle and the matched torsion peak.

For example:

```
<library>
    <hierarchyClass id1="G" id2="G" name="GG">
    ...
    </hierarchyClass>
    <hierarchyClass id1="C" id2="O" name="CO">
        <hierarchySubClass name="Ester bond I" smarts="O=[C:2][O:3]">
```

```
                    <torsionRule method="exact" smarts=
                        "[O:1]=[C:2]!@[O:3]~[CH0:4]">
                        <angleList>
                            <angle score="56.52" tolerance1="20.00"
                            tolerance2="25.00" value="0.0"/>
                        </angleList>
                        <histogram>
                            <bin count="1"/>
                            ...
                        </histogram>
                        <histogram_shifted>
                            <bin count="0"/>
                            ...
                        </histogram_shifted>
                        <histogram_converted>
                            <bin energy="4.67... lower="2.14..." upper="Inf"/>
                            ...
                            <bin energy="1.86..." lower="1.58..." upper="2.40..."/>
                            ...
                           </histogram_converted>
                    </torsionRule>
                    <torsionRule method="approximate" smarts=
                        "[cH0:1][c:2]([cH0])!@[O:3][p:4]">
                        <angleList>
                        <angle beta_1="0.002..." beta_2="-7.843...e-07"
                            score="27.14" theta_0="-90.0" tolerance1="30.00"
                            tolerance2="45.00" value="-90.0"/>
                        ...
                        </angleList>
                        <histogram>
                            <bin count="0"/>
                            ...
                        </histogram>
                        <histogram_shifted>
                            <bin count="0"/>
                            ...
                        </histogram_shifted>
                    </torsionRule>
                ...
             ...
            </hierarchyClass>
            <hierarchyClass id1="N" id2="C" name="NC">
            ...
            </hierarchyClass>
            <hierarchyClass id1="S" id2="N" name="SN">
            ...
            </hierarchyClass>
            <hierarchyClass id1="C" id2="S" name="CS">
            ...
            </hierarchyClass>
            <hierarchyClass id1="C" id2="C" name="CC">
            ...
            </hierarchyClass>
            <hierarchyClass id1="S" id2="S" name="SS">
             ...
            </hierarchyClass>
        </library>
```

The rotatable bonds in a 3D molecule are identified using a default SMARTS pattern. A custom SMARTS pattern may be optionally specified to detect rotatable bonds. Each rotatable bond is matched to a torsion rule in the torsion strain energy library. The strain energy is calculated for each rotatable bond using the calculation method, 'exact' or 'approximate', associated with the matched torsion rule.

The total strain energy (TEUs) of a molecule corresponds to the sum of 'exact' and 'approximate' strain energies calculated for all matched rotatable bonds in the molecule. The total strain energy is set to 'NA' for molecules containing a 'approximate' energy estimate for a torsion angle not observed in the torsion energy library. In addition, the lower and upper bounds on energy at 95% confidence interval are set to 'NA'.

The following output files are generated after the filtering:

```
    <OutfileRoot>.sdf
    <OutfileRoot>_Filtered.sdf
    <OutfileRoot>_AlertsSummary.csv
    <OutfileRoot>_Filtered_TopRule*.sdf
```

The last two set of outfile files, <OutfileRoot>_AlertsSummary.csv and <OutfileRoot>_<OutfileRoot>_AlertsSummary.csv, are only generated during filtering by 'MaxSingleEnergy'.

The supported input file formats are: Mol (.mol), SD (.sdf, .sd)

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

OPTIONS

-a, --alertsMode <TotalEnergy,...> [default: TotalEnergy]

Torsion strain energy library alert types to use for filtering molecules containing rotatable bonds based on the calculated values for the total torsion strain energy of a molecule and the maximum single strain energy of a rotatable bond in a molecule.

Possible values: TotalEnergy, MaxSingleEnergy, or TotalOrMaxSingleEnergy

The strain energy cutoff values in terms of torsion energy units (TEUs) are used to filter molecules as shown below:

```
    AlertsMode                AlertsEnergyCutoffs (TEUs)

    TotalEnergy               >= TotalEnergyCutoff

    MaxSingleEnergy           >= MaxSingleEnergyCutoff

    TotalOrMaxSingleEnergy    >= TotalEnergyCutoff
                              or >= MaxSingleEnergyCutoff
```

--alertsMaxSingleEnergyCutoff <Number> [default: 1.8]

Maximum single strain energy (TEUs) cutoff [Ref 153] for filtering molecules based on the maximum value of a single strain energy of a rotatable bond in a molecule. This option is used during 'MaxSingleEnergy' or 'TotalOrMaxSingleEnergy' values of '-a, --alertsMode' option.

The maximum single strain energy must be greater than or equal to the specified cutoff value for filtering molecules.

--alertsTotalEnergyCutoff <Number> [default: 6.0]

Total strain strain energy (TEUs) cutoff [Ref 153] for filtering molecules based on total strain energy for all rotatable bonds in a molecule. This option is used during 'TotalEnergy' or 'TotalOrMaxSingleEnergy' values of '-a, --alertsMode' option.

The total strain energy must be greater than or equal to the specified cutoff value for filtering molecules.

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

Filter molecules containing torsion angles not observed in torsion strain energy library. It's not possible to calculate torsion strain energies for these torsions during 'approximate' match to a specified torsion in the library.

The 'approximate' strain energy calculation relies on the angle difference between a torsion angle and the torsion peaks observed for the torsion rules in the torsion energy library. The torsion angle is matched to a torsion peak based on the value of torsion angle difference. It must be less than or equal to the value for the second tolerance 'tolerance2'. Otherwise, the torsion angle is not observed in the torsion energy library and a value of 'NA' is assigned for torsion energy along with the lower and upper bounds on energy at 95% confidence interval.

-e, --examples

Print examples.

-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,no,sanitize,yes,strictParsing,yes
```

-l, --list

List torsion library information without performing any filtering.

-m, --mode <filter or count> [default: filter]

Specify whether to filter molecules for torsion strain energy library [Ref 153] alerts by matching rotatable bonds against SMARTS patterns specified for torsion rules to calculate torsion strain energies and write out the rest of the molecules to an outfile or simply count the number of matched molecules marked for filtering.

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

Use multiprocessing.

By default, input data is retrieved in a lazy manner via mp.Pool.imap() function employing lazy RDKit data iterable. This allows processing of arbitrary large data sets without any additional requirements memory.

All input data may be optionally loaded into memory by mp.Pool.map() before starting worker processes in a process pool by setting the value of 'inputDataMode' to 'InMemory' in '--mpParams' option.

A word to the wise: The default 'chunkSize' value of 1 during 'Lazy' input data mode may adversely impact the performance. The '--mpParams' section provides additional information to tune the value of 'chunkSize'.

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

A comma delimited list of parameter name and value pairs to configure multiprocessing.

The supported parameter names along with their default and possible values are shown below:

```
chunkSize, auto
inputDataMode, Lazy   [ Possible values: InMemory or Lazy ]
numProcesses, auto   [ Default: mp.cpu_count() ]
```

These parameters are used by the following functions to configure and control the behavior of multiprocessing: mp.Pool(), mp.Pool.map(), and mp.Pool.imap().

The chunkSize determines chunks of input data passed to each worker process in a process pool by mp.Pool.map() and mp.Pool.imap() functions. The default value of chunkSize is dependent on the value of 'inputDataMode'.

The mp.Pool.map() function, invoked during 'InMemory' input data mode, automatically converts RDKit data iterable into a list, loads all data into memory, and calculates the default chunkSize using the following method as shown in its code:

```
chunkSize, extra = divmod(len(dataIterable), len(numProcesses) * 4)
if extra: chunkSize += 1
```

For example, the default chunkSize will be 7 for a pool of 4 worker processes and 100 data items.

The mp.Pool.imap() function, invoked during 'Lazy' input data mode, employs 'lazy' RDKit data iterable to retrieve data as needed, without loading all the data into memory. Consequently, the size of input data is not known a priori. It's not possible to estimate an optimal value for the chunkSize. The default chunkSize is set to 1.

The default value for the chunkSize during 'Lazy' data mode may adversely impact the performance due to the overhead associated with exchanging small chunks of data. It is generally a good idea to explicitly set chunkSize to a larger value during 'Lazy' input data mode, based on the size of your input data and number of processes in the process pool.

The mp.Pool.map() function waits for all worker processes to process all the data and return the results. The mp.Pool.imap() function, however, returns the the results obtained from worker processes as soon as the results become available for specified chunks of data.

The order of data in the results returned by both mp.Pool.map() and mp.Pool.imap() functions always corresponds to the input data.

-o, --outfile <outfile>

Output file name.

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

Write out alerts information to SD output files.

--outfileAlertsMode <All or AlertsOnly> [default: AlertsOnly]

Write alerts information to SD output files for all alerts or only for alerts specified by '--AlertsMode' option. Possible values: All or AlertsOnly This option is only valid for 'Yes' value of '--outfileAlerts' option.

The following alerts information is added to SD output files using 'TorsionAlerts' data field:

```
RotBondIndices TorsionIndices TorsionAngle
Energy EnergyLowerBoundCI EnergyUpperBoundCI
HierarchyClass HierarchySubClass TorsionRule
EnergyMethod AngleNotObserved MaxSingleEnergyAlert
```

The following data filelds are added to SD output files based on the value of '--AlertsMode' option:

```
TotalEnergy
TotalEnergyLowerBoundCI
TotalEnergyUpperBoundCI

MaxSingleEnergy
MaxSingleEnergyAlertsCount

AnglesNotObservedCount
```

The 'RotBondsCount' is always added to SD output files containing both remaining and filtered molecules.

Format:

```
> <RotBondsCount>
Number

> <TotalEnergy>
Number

> <TotalEnergyLowerBoundCI>
Number

> <TotalEnergyUpperBoundCI>
Number

> <MaxSingleEnergy>
Number

> <MaxSingleEnergyAlertsCount>
Number

> <AnglesNotObservedCount>
Number

> <TorsionAlerts (RotBondIndices TorsionIndices TorsionAngle
    Energy EnergyLowerBoundCI EnergyUpperBoundCI
    HierarchyClass HierarchySubClass TorsionRule
    EnergyMethod AngleNotObserved MaxSingleEnergyAlert)>
AtomIndex2,AtomIndex3  AtomIndex1,AtomIndex2,AtomIndex3,AtomIndex4
Angle  Energy EnergyLowerBoundCI EnergyUpperBoundCI
ClassName SubClassName SMARTS EnergyMethod Yes|No|NA Yes|No|NA
... ... ...
... ... ...
```

A set of 12 values is written out as value of 'TorsionAlerts' data field for each torsion in a molecule. The space character is used as a delimiter to separate values with in a set and across set. The comma character is used to delimit multiple values for each value in a set.

The 'RotBondIndices' and 'TorsionIndices' contain 2 and 4 comma delimited values representing atom indices for a rotatable bond and the matched torsion.

The 'Energy' value is the estimated strain energy for the matched torsion. The 'EnergyLowerBoundCI' and 'EnergyUpperBoundCI' represent lower and bound energy estimates at 95% confidence interval. The 'EnergyMethod', exact or approximate, corresponds to the method employed to estimate torsion strain energy.

The 'AngleNotObserved' is only valid for 'approximate' value of 'EnergyMethod'. It has three possible values: Yes, No, or NA. The 'Yes' value indicates that the 'TorsionAngle' is outside the 'tolerance2' of all peaks for the matched torsion rule in the torsion library.

The 'MaxSingleEnergyAlert' is valid for the following values of '-a, --alertsMode' option: 'MaxSingleEnergy' or 'TotalOrMaxSingleEnergy'. It has three possible values: Yes, No, or NA. It's set to 'NA' for 'Yes' or 'NA' values of 'AngleNotObserved'. The 'Yes' value indicates that the estimated torsion energy is greater than the specified value for '--alertsMaxSingleEnergyCutoff' option.

For example:

```
>  <RotBondsCount>  (1)
14

>  <TotalEnergy>  (1)
6.8065

>  <TotalEnergyLowerBoundCI>  (1)
5.9340

>  <TotalEnergyUpperBoundCI>  (1)
NA

>  <MaxSingleEnergy>  (1)
1.7108

>  <MaxSingleEnergyAlertsCount>  (1)
0

>  <AnglesNotObservedCount>  (1)
0

>  <TorsionAlerts(RotBondIndices TorsionIndices TorsionAngle Energy
    EnergyLowerBoundCI EnergyUpperBoundCI HierarchyClass
    HierarchySubClass TorsionRule EnergyMethod AngleNotObserved
    MaxSingleEnergyAlert)>  (1)
2,1 48,2,1,0 61.90 0.0159 -0.0320 0.0674 CO Ether [O:1][CX4:2]!
@[O:3][CX4:4] Exact NA No 2,3 1,2,3,4 109.12 1.5640 1.1175 NA CC
None/[CX4][CX3] [O:1][CX4:2]!@[CX3:3]=[O:4] Exact NA No
... ... ...
```

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

Write out a file containing filtered molecules. Its name is automatically generated from the specified output file. Default: <OutfileRoot>_ Filtered.<OutfileExt>.

--outfilesFilteredByRules <yes or no> [default: auto]

Write out SD files containing filtered molecules for individual torsion rules triggering alerts in molecules. The name of SD files are automatically generated from the specified output file. Default file names: <OutfileRoot>_ Filtered_TopRule*.sdf.

Default value: 'yes' for 'MaxSingleEnergy' of '-a, --alertsMode' option'; otherwise, 'no'.

The output files are only generated for 'MaxSingleEnergy' of '-a, --alertsMode' option.

The following alerts information is added to SD output files:

```
> <RotBondsCount>
Number

> <TotalEnergy>
Number

> <TotalEnergyLowerBoundCI>
Number

> <TotalEnergyUpperBoundCI>
Number

> <MaxSingleEnergy>
Number

> <MaxSingleEnergyAlertsCount>
```

```
Number

> <AnglesNotObservedCount>
Number

>  <TorsionRule (HierarchyClass HierarchySubClass TorsionRule
    EnergyMethod)>
ClassName SubClassName EnergyMethod SMARTS
 ... ... ...

> <TorsionRuleMaxSingleEnergyAlertsCount>
Number

> <TorsionRuleAnglesNotObservedCount>
Number

>  <TorsionRuleAlerts (RotBondIndices TorsionIndices TorsionAngle
    Energy EnergyLowerBoundCI EnergyUpperBoundCI
    AngleNotObserved MaxSingleEnergyAlert)>
AtomIndex2,AtomIndex3  AtomIndex1,AtomIndex2,AtomIndex3,AtomIndex4
Angle Energy EnergyLowerBoundCI EnergyUpperBoundCI EnergyMethod
Yes|No|NA Yes|No|NA
 ... ... ...
```

For example:

```
>  <RotBondsCount>  (1)
8

>  <TotalEnergy>  (1)
6.1889

>  <TotalEnergyLowerBoundCI>  (1)
5.1940

>  <TotalEnergyUpperBoundCI>  (1)
NA

>  <MaxSingleEnergy>  (1)
1.9576

>  <MaxSingleEnergyAlertsCount>  (1)
1

>  <AnglesNotObservedCount>  (1)
0

>  <TorsionRule (HierarchyClass HierarchySubClass TorsionRule
    EnergyMethod)>  (1)
CC None/[CX4:2][CX4:3] [!#1:1][CX4:2]!@[CX4:3][!#1:4] Exact

>  <TorsionRuleMaxSingleEnergyAlertsCount>  (1)
0

>  <TorsionRuleAnglesNotObservedCount>  (1)
0

>  <TorsionRuleAlerts (RotBondIndices TorsionIndices TorsionAngle
    Energy EnergyLowerBoundCI EnergyUpperBoundCI AngleNotObserved
   MaxSingleEnergyAlert)>  (1)
1,3 0,1,3,4 72.63 0.8946 0.8756 0.9145 NA No
```

--outfilesFilteredByRulesMaxCount <All or number> [default: 10]

Write out SD files containing filtered molecules for specified number of top N torsion rules triggering alerts for the largest number of molecules or for all torsion rules triggering alerts across all molecules.

These output files are only generated for 'MaxSingleEnergy' value of '-a, --alertsMode' option.

--outfileSummary <yes or no> [default: auto]

Write out a CVS text file containing summary of torsions rules responsible for triggering torsion alerts. Its name is automatically generated from the specified output file. Default: <OutfileRoot>_AlertsSummary.csv.

Default value: 'yes' for 'MaxSingleEnergy' of '-a, --alertsMode' option'; otherwise, 'no'.

The summary output file is only generated for 'MaxSingleEnergy' of '-a, --alertsMode' option.

The following alerts information is written to summary text file:

```
TorsionRule, HierarchyClass, HierarchySubClass, EnergyMethod,
MaxSingleEnergyTorsionAlertTypes, MaxSingleEnergyTorsionAlertCount,
MaxSingleEnergyTorsionAlertMolCount
```

The double quotes characters are removed from SMART patterns before before writing them to a CSV file. In addition, the torsion rules are sorted by TorsionAlertMolCount.

--outfileSDFieldLabels <Type,Label,...> [default: auto]

A comma delimited list of SD data field type and label value pairs for writing torsion alerts information along with molecules to SD files.

The supported SD data field label type along with their default values are shown below:

```
For all SD files:

RotBondsCountLabel, RotBondsCount,

TotalEnergyLabel, TotalEnergy,
TotalEnergyLowerBoundCILabel, TotalEnergyLowerBoundCI,
TotalEnergyUpperBoundCILabel, TotalEnergyUpperBoundCI,

MaxSingleEnergyLabel, MaxSingleEnergy,
MaxSingleEnergyAlertsCountLabel,
    MaxSingleEnergyAlertsCount

AnglesNotObservedCountLabel,
    AnglesNotObservedCount

TorsionAlertsLabel, TorsionAlerts(RotBondIndices TorsionIndices
    TorsionAngle Energy EnergyLowerBoundCI EnergyUpperBoundCI
    HierarchyClass HierarchySubClass TorsionRule
    EnergyMethod AngleNotObserved)

For individual SD files filtered by torsion rules:

TorsionRuleLabel, TorsionRule (HierarchyClass HierarchySubClass
    EnergyMethod TorsionRule)
TorsionRuleMaxSingleEnergyAlertsCountLabel,
    TorsionRuleMaxSingleEnergyAlertsCount,
TorsionRuleAnglesNotObservedCountLabel,
    TorsionRuleAnglesNotObservedCount,
TorsionRuleAlertsLabel, TorsionRuleAlerts (RotBondIndices
    TorsionIndices TorsionAngle Energy EnergyLowerBoundCI
    EnergyUpperBoundCI EnergyMethod AngleObserved)
```

--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,yes,forceV3000,no
```

--overwrite

Overwrite existing files.

--precision <number> [default: 4]

Floating point precision for writing torsion strain energy values.

-r, --rotBondsSMARTSMode <NonStrict, SemiStrict,...> [default: SemiStrict]

SMARTS pattern to use for identifying rotatable bonds in a molecule for matching against torsion rules in the torsion library. Possible values: NonStrict, SemiStrict, Strict or Specify. The rotatable bond SMARTS matches are filtered to ensure that each atom in the rotatable bond is attached to at least two

heavy atoms.

The following SMARTS patterns are used to identify rotatable bonds for different modes:

```
NonStrict: [!$(*#*)&!D1]-&!@[!$(*#*)&!D1]

SemiStrict:
[!$(*#*)&!D1&!$(C(F)(F)F)&!$(C(Cl)(Cl)Cl)&!$(C(Br)(Br)Br)
&!$(C([CH3])([CH3])[CH3])]-!@[!$(*#*)&!D1&!$(C(F)(F)F)
&!$(C(Cl)(Cl)Cl)&!$(C(Br)(Br)Br)&!$(C([CH3])([CH3])[CH3])]

Strict:
[!$(*#*)&!D1&!$(C(F)(F)F)&!$(C(Cl)(Cl)Cl)&!$(C(Br)(Br)Br)
&!$(C([CH3])([CH3])[CH3])&!$([CD3](=[N,O,S])-!@[#7,O,S!D1])
&!$([#7,O,S!D1]-!@[CD3]=[N,O,S])&!$([CD3](=[N+])-!@[#7!D1])
&!$([#7!D1]-!@[CD3]=[N+])]-!@[!$(*#*)&!D1&!$(C(F)(F)F)
&!$(C(Cl)(Cl)Cl)&!$(C(Br)(Br)Br)&!$(C([CH3])([CH3])[CH3])]
```

The 'NonStrict' and 'Strict' SMARTS patterns are available in RDKit. The 'NonStrict' SMARTS pattern corresponds to original Daylight SMARTS specification for rotatable bonds. The 'SemiStrict' SMARTS pattern is derived from 'Strict' SMARTS pattern for its usage in this script.

You may use any arbitrary SMARTS pattern to identify rotatable bonds by choosing 'Specify' value for '-r, --rotBondsSMARTSMode' option and providing its value via '--rotBondsSMARTSPattern' option.

--rotBondsSMARTSPattern <SMARTS>

SMARTS pattern for identifying rotatable bonds. This option is only valid for 'Specify' value of '-r, --rotBondsSMARTSMode' option.

-t, --torsionEnergyLibraryFile <FileName or auto> [default: auto]

Specify a XML file name containing data for torsion starin energy library hierarchy or use default file, TorsionEnergyLibrary.xml, available in MAYACHEMTOOLS/lib/Python/TorsionAlerts directory.

The format of data in local XML file must match format of the data in Torsion Library [Ref 153] file available in MAYACHEMTOOLS directory.

-w, --workingdir <dir>

Location of working directory which defaults to the current directory.

EXAMPLES

To filter molecules containing rotatable bonds with total strain energy value of >= 6.0 (TEUs) based on torsion rules in the torsion energy library and write write out SD files containing remaining and filtered molecules, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py -i Sample3D.sdf
  -o Sample3DOut.sdf
```

To filter molecules containing any rotatable bonds with strain energy value of >= 1.8 (TEUs) based on torsion rules in the torsion energy library and write out SD files containing remaining and filtered molecules, and individual SD files for torsion rules triggering alerts along with appropriate torsion information for red alerts, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py -a MaxSingleEnergy
  -i Sample3D.sdf -o Sample3DOut.sdf
```

To filter molecules containing rotatable bonds with total strain energy value of >= 6.0 (TEUs) or any single strain energy value of >= 1.8 (TEUs) and write out SD files containing remaining and filtered molecules, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py -a TotalOrMaxSingleEnergy
  -i Sample3D.sdf -o Sample3DOut.sdf
```

To filter molecules containing rotatable bonds with specific cutoff values for total or single torsion strain energy and write out SD files containing remaining and filtered molecules, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py -a TotalOrMaxSingleEnergy
  -i Sample3D.sdf -o Sample3DOut.sdf --alertsTotalEnergyCutoff 6.0
  --alertsMaxSingleEnergyCutoff 1.8
```

To run the first example for filtering molecules and writing out torsion information for all alert types to SD files,

type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py -i Sample3D.sdf
  -o Sample3DOut.sdf --outfileAlertsMode All
```

To run the first example for filtering molecules in multiprocessing mode on all available CPUs without loading all data into memory and write out SD files, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py --mp yes -i Sample3D.sdf
 -o Sample3DOut.sdf
```

To run the first example for filtering molecules in multiprocessing mode on all available CPUs by loading all data into memory and write out a SD files, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py  --mp yes --mpParams
  "inputDataMode, InMemory" -i Sample3D.sdf  -o Sample3DOut.sdf
```

To run the first example for filtering molecules in multiprocessing mode on specific number of CPUs and chunksize without loading all data into memory and write out SD files, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py --mp yes --mpParams
  "inputDataMode,lazy,numProcesses,4,chunkSize,8"  -i Sample3D.sdf
  -o Sample3DOut.sdf
```

To list information about default torsion library file without performing any filtering, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py -l
```

To list information about a local torsion library XML file without performing any, filtering, type:

```
% RDKitFilterTorsionStrainEnergyAlerts.py --torsionEnergyLibraryFile
  TorsionStrainEnergyLibrary.xml -l
```

AUTHOR

Manish Sud (msud@san.rr.com)

COLLABORATOR

Pat Walters

SEE ALSO

RDKitFilterChEMBLAlerts.py, RDKitFilterPAINS.py, RDKitFilterTorsionLibraryAlerts.py, RDKitConvertFileFormat.py, RDKitSearchSMARTS.py

COPYRIGHT

Copyright (C) 2026 Manish Sud. All rights reserved.

This script uses the torsion strain energy library developed by Gu, S.; Smith, M. S.; Yang, Y.; Irwin, J. J.; Shoichet, B. K. [Ref 153].

The torsion strain enegy library is based on the Torsion Library jointly developed by the University of Hamburg, Center for Bioinformatics, Hamburg, Germany and F. Hoffmann-La-Roche Ltd., Basel, Switzerland.

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.