NAME

Psi4PerformTorsionScan.py - Perform torsion scan

SYNOPSIS

Psi4PerformTorsionScan.py [--basisSet <text>] [--confParams <Name,Value,...>] [--energyDataFieldLabel <text>] [--energyRelativeDataFieldLabel <text>] [--energyUnits <text>] [--infile3D <yes or no>] [--infileParams <Name,Value,...>] [--maxIters <number>] [--methodName <text>] [--modeMols <First or All>] [--modeTorsions <First or All>] [--mp <yes or no>] [--mpLevel <Molecules or TorsionAngles>] [--mpParams <Name,Value,...>] [--outfileMolName <yes or no>] [--outfileParams <Name,Value,...>] [--outPlotParams <Name,Value,...>] [--outPlotRelativeEnergy <yes or no>] [--outPlotTitleTorsionSpec <yes or no>] [--overwrite] [--precision <number>] [--psi4OptionsParams <Name,Value,...>] [--psi4RunParams <Name,Value,...>] [--quiet <yes or no>] [--reference <text>] [--torsionsFilterbyAtomIndices <Index1, Index2, ...>] [--torsionMaxMatches <number>] [--torsionMinimize <yes or no>] [--torsionRangeMode <Range or Angles>] [--torsionRange <Start,Stop,Step or Angle1,Angle2,...>] [--useChirality <yes or no>] [-w <dir>] -t <torsions> -i <infile> -o <outfile>

Psi4PerformTorsionScan.py -h | --help | -e | --examples

DESCRIPTION

Perform torsion scan for molecules around torsion angles specified using SMILES/SMARTS patterns. A molecule is optionally minimized before performing a torsion scan using a forcefield. A set of initial 3D structures are generated for a molecule by scanning the torsion angle across the specified range and updating the 3D coordinates of the molecule. A conformation ensemble is optionally generated for each 3D structure representing a specific torsion angle using a combination of distance geometry and forcefield followed by constrained geometry optimization using a quantum chemistry method. The conformation with the lowest energy is selected to represent the torsion angle. An option is available to skip the generation of the conformation ensemble and simply calculate the energy for the initial 3D structure for a specific torsion torsion angle using a quantum chemistry method.

The torsions are specified using SMILES or SMARTS patterns. A substructure match is performed to select torsion atoms in a molecule. The SMILES pattern match must correspond to four torsion atoms. The SMARTS patterns containing atom map numbers may match more than four atoms. The atom map numbers, however, must match exactly four torsion atoms. For example: [s:1][c:2]([aX2,cH1])!@[CX3:3](O)=[O:4] for thiophene esters and carboxylates as specified in Torsion Library (TorLib) [Ref 146].

A Psi4 XYZ format geometry string is automatically generated for each molecule in input file. It contains atom symbols and 3D coordinates for each atom in a molecule. In addition, the formal charge and spin multiplicity are present in the the geometry string. These values are either retrieved from molecule properties named 'FormalCharge' and 'SpinMultiplicty' or dynamically calculated for a molecule.

A set of four output files is generated for each torsion match in each molecule. The names of the output files are generated using the root of the specified output file. They may either contain sequential molecule numbers or molecule names as shown below:

```
<OutfileRoot>_Mol<Num>.sdf
<OutfileRoot>_Mol<Num>_Torsion<Num>_Match<Num>.sdf
<OutfileRoot>_Mol<Num>_Torsion<Num>_Match<Num>_Energies.csv
<OutfileRoot>_Mol<Num>_Torsion<Num>_Match<Num>_Plot.<ImgExt>


or


<OutfileRoot>_<MolName>.sdf
<OutfileRoot>_<MolName>_Torsion<Num>_Match<Num>.sdf
<OutfileRoot>_<MolName>_Torsion<Num>_Match<Num>_Energies.csv
<OutfileRoot>_<MolName>_Torsion<Num>_Match<Num>_Plot.<ImgExt>
```

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

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

OPTIONS

-b, --basisSet <text> [default: auto]

Basis set to use for energy calculation or constrained energy minimization. Default: 6-31+G** for sulfur

containing molecules; Otherwise, 6-31G** [Ref 150]. The specified value must be a valid Psi4 basis set. No validation is performed.

The following list shows a representative sample of basis sets available in Psi4:

```
STO-3G, 6-31G, 6-31+G, 6-31++G, 6-31G*, 6-31+G*,  6-31++G*,
6-31G**, 6-31+G**, 6-31++G**, 6-311G, 6-311+G, 6-311++G,
6-311G*, 6-311+G*, 6-311++G*, 6-311G**, 6-311+G**, 6-311++G**,
cc-pVDZ, cc-pCVDZ, aug-cc-pVDZ, cc-pVDZ-DK, cc-pCVDZ-DK, def2-SVP,
def2-SVPD, def2-TZVP, def2-TZVPD, def2-TZVPP, def2-TZVPPD
```

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

A comma delimited list of parameter name and value pairs for generating initial 3D coordinates for molecules in input file at specific torsion angles. A conformation ensemble is optionally generated for each 3D structure representing a specific torsion angle using a combination of distance geometry and forcefield followed by constrained geometry optimization using a quantum chemistry method. The conformation with the lowest energy is selected to represent the torsion angle.

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

```
confMethod,ETKDGv2,
forceField,MMFF, forceFieldMMFFVariant,MMFF94,
enforceChirality,yes,embedRMSDCutoff,0.5,maxConfs,250,
maxConfsTorsions,50,useTethers,yes

confMethod,ETKDGv2   [ Possible values: SDG, KDG, ETDG,
    ETKDG , or ETKDGv2]
forceField, MMFF   [ Possible values: UFF or MMFF ]
forceFieldMMFFVariant,MMFF94   [ Possible values: MMFF94 or MMFF94s ]
enforceChirality,yes   [ Possible values: yes or no ]
useTethers,yes   [ Possible values: yes or no ]
```

confMethod: Conformation generation methodology for generating initial 3D coordinates. Possible values: Standard Distance Geometry (SDG), Experimental Torsion-angle preference with Distance Geometry (ETDG), basic Knowledge-terms with Distance Geometry (KDG) and Experimental Torsion-angle preference along with basic Knowledge-terms and Distance Geometry (ETKDG or ETKDGv2) [Ref 129, 167] .

forceField: Forcefield method to use for energy minimization. Possible values: Universal Force Field (UFF) [Ref 81] or Merck Molecular Mechanics Force Field [Ref 83-87] .

enforceChirality: Enforce chirality for defined chiral centers during forcefield minimization.

maxConfs: Maximum number of conformations to generate for each molecule during the generation of an initial 3D conformation ensemble using a conformation generation methodology. The conformations are minimized using the specified forcefield. The lowest energy structure is selected for performing the torsion scan.

maxConfsTorsion: Maximum number of 3D conformations to generate for conformation ensemble representing a specific torsion. The conformations are constrained at specific torsions angles and minimized using the specified forcefield and a quantum chemistry method. The lowest energy conformation is selected to calculate final torsion energy and written to the output file.

embedRMSDCutoff: RMSD cutoff for retaining initial set of conformers embedded using distance geometry and forcefield minimization. All embedded conformers are kept for 'None' value. Otherwise, only those conformers which are different from each other by the specified RMSD cutoff, 0.5 by default, are kept. The first embedded conformer is always retained.

useTethers: Use tethers to optimize the final embedded conformation by applying a series of extra forces to align matching atoms to the positions of the core atoms. Otherwise, use simple distance constraints during the optimization.

--energyDataFieldLabel <text> [default: auto]

Energy data field label for writing energy values. Default: Psi4_Energy (<Units>).

--energyRelativeDataFieldLabel <text> [default: auto]

Relative energy data field label for writing energy values. Default: Psi4_Relative_Energy (<Units>).

--energyUnits <text> [default: kcal/mol]

Energy units. Possible values: Hartrees, kcal/mol, kJ/mol, or eV.

-e, --examples

Print examples.

-h, --help

Print this help message.

-i, --infile <infile>

Input file name.

--infile3D <yes or no> [default: no]

Skip generation and minimization of initial 3D structures for molecules in input file containing 3D coordinates.

--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

SMILES: smilesColumn,1,smilesNameColumn,2,smilesDelimiter,space,
    smilesTitleLine,auto,sanitize,yes
```

Possible values for smilesDelimiter: space, comma or tab.

--maxIters <number> [default: 50]

Maximum number of iterations to perform for each molecule or conformer during constrained energy minimization by a quantum chemistry method.

-m, --methodName <text> [default: auto]

Method to use for energy calculation or constrained energy minimization. Default: B3LYP [Ref 150]. The specified value must be a valid Psi4 method name. No validation is performed.

The following list shows a representative sample of methods available in Psi4:

```
B1LYP, B2PLYP, B2PLYP-D3BJ, B2PLYP-D3MBJ, B3LYP, B3LYP-D3BJ,
B3LYP-D3MBJ, CAM-B3LYP, CAM-B3LYP-D3BJ, HF, HF-D3BJ,  HF3c, M05,
M06, M06-2x, M06-HF, M06-L, MN12-L, MN15, MN15-D3BJ,PBE, PBE0,
PBEH3c, PW6B95, PW6B95-D3BJ, WB97, WB97X, WB97X-D, WB97X-D3BJ
```

--modeMols <First or All> [default: First]

Perform torsion scan for the first molecule or all molecules in input file.

--modeTorsions <First or All> [default: First]

Perform torsion scan for the first or all specified torsion pattern in molecules up to a maximum number of matches for each torsion specification as indicated by '--torsionMaxMatches' option.

--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'.

--mpLevel <Molecules or TorsionAngles> [default: Molecules]

Perform multiprocessing at molecules or torsion angles level. Possible values: Molecules or TorsionAngles. The 'Molecules' value starts a process pool at the molecules level. All torsion angles of a molecule are processed in a single process. The 'TorsionAngles' value, however, starts a process pool at the torsion

angles level. Each torsion angle in a torsion match for a molecule is processed in an individual process in the process pool.

--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. The output file root is used for generating the names of the output files corresponding to structures, energies, and plots during the torsion scan.

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

Append molecule name to output file root during the generation of the names for output files. The default is to use <MolNum>. The non alphabetical characters in molecule names are replaced by underscores.

--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
```

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

A comma delimited list of parameter name and value pairs for generating plots using Seaborn module. The supported parameter names along with their default values are shown below:

```
type,linepoint,outExt,svg,width,10,height,5.6,
title,auto,xlabel,auto,ylabel,auto,titleWeight,bold,labelWeight,bold
style,darkgrid,palette,deep,font,sans-serif,fontScale,1,
context,notebook
```

Possible values:

```
                type: linepoint, scatter, or line. Both points and lines are drawn
                    for linepoint plot type.
                outExt: Any valid format supported by Python module Matplotlib.
                    For example: PDF (.pdf), PNG (.png), PS (.ps), SVG (.svg)
                titleWeight, labelWeight: Font weight for title and axes labels.
                    Any valid value.
                style: darkgrid, whitegrid, dark, white, ticks
                palette: deep, muted, pastel, dark, bright, colorblind
                font: Any valid font name
                context: paper, notebook, talk, poster, or any valid name
```

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

> Plot relative energies in the torsion plot. The minimum energy value is subtracted from energy values to calculate relative energies.

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

> Append torsion specification to the title of the torsion plot.

--overwrite

> Overwrite existing files.

--precision <number> [default: 6]

> Floating point precision for writing energy values.

--psi4OptionsParams <Name,Value,...> [default: none]

> A comma delimited list of Psi4 option name and value pairs for setting global and module options. The names are 'option_name' for global options and 'module_name__option_name' for options local to a module. The specified option names must be valid Psi4 names. No validation is performed.

> The specified option name and value pairs are processed and passed to psi4.set_options() as a dictionary. The supported value types are float, integer, boolean, or string. The float value string is converted into a float. The valid values for a boolean string are yes, no, true, false, on, or off.

--psi4RunParams <Name,Value,...> [default: auto]

> A comma delimited list of parameter name and value pairs for configuring Psi4 jobs.

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

```
        MemoryInGB, 1
        NumThreads, 1
        OutputFile, auto   [ Possible  values: stdout, quiet, or FileName ]
        ScratchDir, auto   [ Possivle values: DirName]
        RemoveOutputFile, yes   [ Possible values: yes, no, true, or false]
```

> These parameters control the runtime behavior of Psi4.

> The default file name for 'OutputFile' is <InFileRoot>_Psi4.out. The PID is appended to output file name during multiprocessing as shown: <InFileRoot>_Psi4_<PIDNum>.out. The 'stdout' value for 'OutputType' sends Psi4 output to stdout. The 'quiet' or 'devnull' value suppresses all Psi4 output. The 'OutputFile' is set to 'quiet' for 'auto' value during 'Conformers' of '--mpLevel' option.

> The default 'Yes' value of 'RemoveOutputFile' option forces the removal of any existing Psi4 before creating new files to append output from multiple Psi4 runs.

> The option 'ScratchDir' is a directory path to the location of scratch files. The default value corresponds to Psi4 default. It may be used to override the deafult path.

-q, --quiet <yes or no> [default: no]

> Use quiet mode. The warning and information messages will not be printed.

--reference <text> [default: auto]

> Reference wave function to use for energy calculation or constrained energy minimization. Default: RHF or UHF. The default values are Restricted Hartree-Fock (RHF) for closed-shell molecules with all electrons paired and Unrestricted Hartree-Fock (UHF) for open-shell molecules with unpaired electrons.

> The specified value must be a valid Psi4 reference wave function. No validation is performed. For example:

ROHF, CUHF, RKS, etc.

The spin multiplicity determines the default value of reference wave function for input molecules. It is calculated from number of free radical electrons using Hund's rule of maximum multiplicity defined as 2S + 1 where S is the total electron spin. The total spin is 1/2 the number of free radical electrons in a molecule. The value of 'SpinMultiplicity' molecule property takes precedence over the calculated value of spin multiplicity.

-t, --torsions <SMILES/SMARTS,…,…>

SMILES/SMARTS patterns corresponding to torsion specifications. It's a comma delimited list of valid SMILES/SMART patterns.

A substructure match is performed to select torsion atoms in a molecule. The SMILES pattern match must correspond to four torsion atoms. The SMARTS patterns containing atom map numbers may match more than four atoms. The atom map numbers, however, must match exactly four torsion atoms. For example: [s:1][c:2]([aX2,cH1])!@[CX3:3](O)=[O:4] for thiophene esters and carboxylates as specified in Torsion Library (TorLib) [Ref 146].

--torsionsFilterbyAtomIndices <Index1, Index2, …> [default: none]

Comma delimited list of atom indices for filtering torsion matches corresponding to torsion specifications "-t, --torsions". The atom indices must be valid. No explicit validation is performed. The list must contain at least 4 atom indices.

The torsion atom indices, matched by "-t, --torsions" specifications, must be present in the list. Otherwise, the torsion matches are ignored.

--torsionMaxMatches <number> [default: 5]

Maximum number of torsions to match for each torsion specification in a molecule.

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

Perform constrained energy minimization on a conformation ensemble for a specific torsion angle and select the lowest energy conformation representing the torsion angle. A conformation ensemble is generated for each 3D structure representing a specific torsion angle using a combination of distance geometry and forcefield followed by constrained geometry optimization using a quantum chemistry method.

--torsionRangeMode <Range or Angles> [default: Range]

Perform torsion scan using torsion angles corresponding to a torsion range or an explicit list of torsion angles. Possible values: Range or Angles. You may use '--torsionRange' option to specify values for torsion angle or torsion angles.

--torsionRange <Start,Stop,Step or Angle1,Angle2…> [default: auto]

Start, stop, and step size angles or a comma delimited list of angles in degrees for a torsion scan.

This value is '--torsionRangeMode' specific. It must be a triplet corresponding to 'start,Stop,Step' for 'Range' value of '--torsionRange' option. Otherwise, it is comma delimited list of one or more torsion angles for 'Angles' value of '--torsionRange' option.

The default values, based on '--torsionRangeMode' option, are shown below:

```
TorsionRangeMode        Default value
Range                   0,360,5
Angles                  None
```

You must explicitly provide a list of torsion angle(s) for 'Angles' of '--torsionRangeMode' option.

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

Use chirrality during substructure matches for identification of torsions.

-w, --workingdir <dir>

Location of working directory which defaults to the current directory.

EXAMPLES

To perform a torsion scan on the first molecule in a SMILES file using a minimum energy structure of the molecule selected from an initial ensemble of conformations generated using distance geometry and forcefield, skip generation of conformation ensembles for specific torsion angles and constrained energy minimization of the

ensemble, calculating single point at a specific torsion angle energy using B3LYP/6-31G** and B3LYP/6-31+G** for non-sulfur and sulfur containing molecules, generate output files corresponding to structure, energy and torsion plot, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" -i Psi4SampleTorsionScan.smi
  -o SampleOut.sdf
```

To run the previous example for performing a torsion scan using a specific list of torsion angles, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" -i Psi4SampleTorsionScan.smi
  -o SampleOut.sdf --torsionRangeMode Angles
  --torsionRange "0,180,360"
```

To run the previous example on the first molecule in a SD file containing 3D coordinates and skip the generations of initial 3D structure, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC"  --infile3D yes
  -i Psi4SampleTorsionScan3D.sdf  -o SampleOut.sdf
```

To run the first example on all molecules in a SD file, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" --modeMols All
  -i Psi4SampleTorsionScan.sdf -o SampleOut.sdf
```

To run the first example on all molecules in a SD file containing 3D coordinates and skip the generation of initial 3D structures, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC"  --infile3D yes
  --modeMols All -i Psi4SampleTorsionScan3D.sdf  -o SampleOut.sdf
```

To perform a torsion scan on the first molecule in a SMILES file using a minimum energy structure of the molecule selected from an initial ensemble of conformations generated using distance geometry and forcefield, generate up to 50 conformations for specific torsion angles using ETKDGv2 methodology followed by initial MMFF forcefield minimization and final energy minimization using B3LYP/6-31G** and B3LYP/6-31+G** for non-sulfur and sulfur containing molecules, generate output files corresponding to minimum energy structure, energy and torsion plot, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" --torsionMinimize Yes
   -i Psi4SampleTorsionScan.smi -o SampleOut.sdf
```

To run the previous example on all molecules in a SD file, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" --modeMols All
   --torsionMinimize Yes -i Psi4SampleTorsionScan.sdf -o SampleOut.sdf
```

To run the previous example on all molecules in a SD file containing 3D coordinates and skip the generation of initial 3D structures, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" --modeMols All
   --infile3D yes --modeMols All  --torsionMinimize Yes
   -i Psi4SampleTorsionScan.sdf -o SampleOut.sdf
```

To run the previous example in multiprocessing mode at molecules level on all available CPUs without loading all data into memory and write out a SD file, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" -i Psi4SampleTorsionScan.smi
  -o SampleOut.sdf --modeMols All --torsionMinimize Yes --mp yes
```

To run the previous example in multiprocessing mode at torsion angles level on all available CPUs without loading all data into memory and write out a SD file, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" -i Psi4SampleTorsionScan.smi
```

```
    -o SampleOut.sdf --modeMols All --torsionMinimize Yes --mp yes
    --mpLevel TorsionAngles
```

To run the previous example in multiprocessing mode on all available CPUs by loading all data into memory and write out a SD file, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" -i Psi4SampleTorsionScan.smi
  -o SampleOut.sdf --modeMols All --torsionMinimize Yes --mp yes
  --mpParams "inputDataMode,InMemory"
```

To run the previous example in multiprocessing mode on specific number of CPUs and chunk size without loading all data into memory and write out a SD file, type:

```
% Psi4PerformTorsionScan.py  -t "CCCC" -i Psi4SampleTorsionScan.smi
  -o SampleOut.sdf --modeMols All --torsionMinimize Yes --mp yes
  --mpParams "inputDataMode,Lazy,numProcesses,4,chunkSize,8"
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

Psi4CalculateEnergy.py, Psi4GenerateConformers.py, Psi4GenerateConstrainedConformers.py, Psi4PerformConstrainedMinimization.py

COPYRIGHT

Copyright (C) 2025 Manish Sud. All rights reserved.

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.