NAME

TopologicalAtomTorsionsFingerprints.pl - Generate topological atom torsions fingerprints for SD files

SYNOPSIS

TopologicalAtomTorsionsFingerprints.pl SDFile(s)...

TopologicalAtomTorsionsFingerprints.pl [--AromaticityModel *AromaticityModelType*] [-a, --AtomIdentifierType *AtomicInvariantsAtomTypes*] [--AtomicInvariantsToUse *"AtomicInvariant,AtomicInvariant..."*] [--FunctionalClassesToUse *"FunctionalClass1,FunctionalClass2..."*] [--CompoundID *DataFieldName or LabelPrefixString*] [--CompoundIDLabel *text*] [--CompoundIDMode] [--DataFields *"FieldLabel1,FieldLabel2,..."*] [-d, --DataFieldsMode *All* | *Common* | *Specify* | *CompoundID*] [-f, --Filter *Yes* | *No*] [--FingerprintsLabel *text*] [-h, --help] [-k, --KeepLargestComponent *Yes* | *No*] [--OutDelim *comma* | *tab* | *semicolon*] [--output *SD* | *FP* | *text* | *all*] [-o, --overwrite] [-q, --quote *Yes* | *No*] [-r, --root *RootName*] [-v, --VectorStringFormat] [-w, --WorkingDir dirname] SDFile(s)...

DESCRIPTION

Generate topological atom torsions fingerprints [Ref 58, Ref 72] for *SDFile(s)* and create appropriate SD, FP or CSV/TSV text file(s) containing fingerprints vector strings corresponding to molecular fingerprints.

Multiple SDFile names are separated by spaces. The valid file extensions are *.sdf* and *.sd*. All other file names are ignored. All the SD files in a current directory can be specified either by **.sdf* or the current directory name.

The current release of MayaChemTools supports generation of topological atom torsions fingerprints corresponding to following -a, --AtomIdentifierTypes:

```
AtomicInvariantsAtomTypes, DREIDINGAtomTypes, EStateAtomTypes,
FunctionalClassAtomTypes, MMFF94AtomTypes, SLogPAtomTypes,
SYBYLAtomTypes, TPSAAtomTypes, UFFAtomTypes
```

Based on the values specified for -a, --AtomIdentifierType and --AtomicInvariantsToUse, initial atom types are assigned to all non-hydrogen atoms in a molecule. All unique atom torsions are identified and an atom torsion identifier is generated; the format of atom torsion identifier is:

```
<AtomType1>-<AtomType2>-<AtomType3>-<AtomType4>

AtomType1, AtomType2, AtomType3, AtomTyp4: Assigned atom types

where AtomType1 <= AtomType2 <= AtomType3 <= AtomType4
```

The atom torsion identifiers for all unique atom torsions corresponding to non-hydrogen atoms constitute topological atom torsions fingerprints of the molecule.

Example of *SD* file containing topological atom torsions fingerprints string data:

```
... ...
... ...
$$$$
... ...
... ...
... ...
41 44  0  0  0  0  0  0  0  0999 V2000
 -3.3652    1.4499    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
... ...
2  3  1  0  0  0  0
... ...
M  END
>  <CmpdID>
Cmpd1

>  <TopologicalAtomTorsionsFingerprints>
FingerprintsVector;TopologicalAtomTorsions:AtomicInvariantsAtomTypes;33
;NumericalValues;IDsAndValuesString;C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-C.
X3.BO4 C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-N.X3.BO3 C.X2.BO2.H2-C.X2.BO2.H
2-C.X3.BO3.H1-C.X2.BO2.H2 C.X2.BO2.H2-C.X2.BO2.H2-C.X3.BO3.H1-O.X1...;
2 2 1 1 2 2 1 1 3 4 4 8 4 2 2 6 2 2 1 2 1 1 2 1 1 2 6 2 4 2 1 3 1

$$$$
... ...
... ...
```

Example of *FP* file containing topological atom torsions fingerprints string data:

```
#
# Package = MayaChemTools 7.4
# Release Date = Oct 21, 2010
#
# TimeStamp = Fri Mar 11 15:17:20 2011
#
# FingerprintsStringType = FingerprintsVector
#
# Description = TopologicalAtomTorsions:AtomicInvariantsAtomTypes
# VectorStringFormat = IDsAndValuesString
# VectorValuesType = NumericalValues
#
Cmpd1 33;C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-C.X3.BO4...;2 2 1 1 2 2 ...
Cmpd2 23;C.X1.BO1.H3-C.X2.BO2.H2-C.X3.BO3.H1-C.X2.BO2.H2...;2 2 1 5 ...
... ...
... ..
```

Example of CSV *Text* file containing topological atom torsions fingerprints string data:

```
"CompoundID","TopologicalAtomTorsionsFingerprints"
"Cmpd1","FingerprintsVector;TopologicalAtomTorsions:AtomicInvariantsAto
mTypes;33;NumericalValues;IDsAndValuesString;C.X1.BO1.H3-C.X3.BO3.H1-C.
X3.BO4-C.X3.BO4 C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-N.X3.BO3 C.X2.BO2.H2-C
.X2.BO2.H2-C.X3.BO3.H1-C.X2.BO2.H2 C.X2.BO2.H2-C.X2.BO2.H2-C.X3.BO3.H1-O...;
2 2 1 1 2 2 1 1 3 4 4 8 4 2 2 6 2 2 1 2 1 1 2 1 1 2 6 2 4 2 1 3 1
... ...
... ...
```

The current release of MayaChemTools generates the following types of topological atom torsions fingerprints vector strings:

```
FingerprintsVector;TopologicalAtomTorsions:AtomicInvariantsAtomTypes;3
3;NumericalValues;IDsAndValuesString;C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-
C.X3.BO4 C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-N.X3.BO3 C.X2.BO2.H2-C.X2.BO
2.H2-C.X3.BO3.H1-C.X2.BO2.H2 C.X2.BO2.H2-C.X2.BO2.H2-C.X3.BO3.H1-O...;
;2 2 1 1 2 2 1 1 3 4 4 8 4 2 2 6 2 2 1 2 1 1 2 1 1 2 6 2 4 2 1 3 1
```

```
FingerprintsVector;TopologicalAtomTorsions:AtomicInvariantsAtomTypes;3
3;NumericalValues;IDsAndValuesPairsString;C.X1.BO1.H3-C.X3.BO3.H1-C.X3
.BO4-C.X3.BO4 2 C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-N.X3.BO3 2 C.X2.BO2.H
2-C.X2.BO2.H2-C.X3.BO3.H1-C.X2.BO2.H2 1 C.X2.BO2.H2-C.X2.BO2.H2-C.X3.B
O3.H1-O.X1.BO1.H1 1 C.X2.BO2.H2-C.X2.BO2.H2-N.X3.BO3-C.X3.BO4 2 C.X2.B
O2.H2-C.X3.BO3.H1-C.X2.BO2.H2-C.X3.BO3.H1 2 C.X2.BO2.H2-C.X3.BO3.H1...
```

```
FingerprintsVector;TopologicalAtomTorsions:DREIDINGAtomTypes;27;Numeri
calValues;IDsAndValuesString;C_2-C_3-C_3-C_3 C_2-C_3-C_3-O_3 C_2-C_R-C
_R-C_3 C_2-C_R-C_R-C_R C_2-C_R-C_R-N_R C_2-N_3-C_R-C_R C_3-C_3-C_2-O_2
C_3-C_3-C_2-O_3 C_3-C_3-C_3-C_3 C_3-C_3-C_3-N_R C_3-C_3-C_3-O_3 C_...;
1 1 1 2 1 2 1 1 3 1 3 2 2 2 1 1 1 3 1 2 2 32 2 2 5 3 1
```

```
FingerprintsVector;TopologicalAtomTorsions:EStateAtomTypes;36;Numerica
lValues;IDsAndValuesString;aaCH-aaCH-aaCH-aaCH aaCH-aaCH-aaCH-aasC aaC
H-aaCH-aasC-aaCH aaCH-aaCH-aasC-aasC aaCH-aaCH-aasC-sF aaCH-aaCH-aasC-
ssNH aaCH-aasC-aasC-aasC aaCH-aasC-aasC-aasN aaCH-aasC-ssNH-dssC a...;
4 4 8 4 2 2 6 2 2 2 4 3 2 1 3 3 2 2 2 1 2 1 1 1 2 1 1 1 1 1 1 1 2 1 1 2
```

```
FingerprintsVector;TopologicalAtomTorsions:FunctionalClassAtomTypes;26
;NumericalValues;IDsAndValuesString;Ar-Ar-Ar-Ar Ar-Ar-Ar-Ar.HBA Ar-Ar-
Ar-HBD Ar-Ar-Ar-Hal Ar-Ar-Ar-None Ar-Ar-Ar.HBA-Ar Ar-Ar-Ar.HBA-None Ar
-Ar-HBD-None Ar-Ar-None-HBA Ar-Ar-None-HBD Ar-Ar-None-None Ar-Ar.H...;
32 5 2 2 3 3 3 2 2 2 2 1 2 1 1 1 2 1 1 1 1 3 1 1 1 3
```

```
FingerprintsVector;TopologicalAtomTorsions:MMFF94AtomTypes;43;Numerica
lValues;IDsAndValuesString;C5A-C5B-C5B-C5A C5A-C5B-C5B-C=ON C5A-C5B-C5
B-CB C5A-C5B-C=ON-NC=O C5A-C5B-C=ON-O=CN C5A-C5B-CB-CB C5A-CB-CB-CB C5
A-N5-C5A-C5B C5A-N5-C5A-CB C5A-N5-C5A-CR C5A-N5-CR-CR C5B-C5A-CB-C...;
1 1 1 1 1 2 2 2 1 1 2 2 2 2 1 1 2 1 1 2 1 2 1 1 1 2 1 1 1 2 18 2 2 1 1
1 1 2 1 1 3 1 3
```

```
FingerprintsVector;TopologicalAtomTorsions:SLogPAtomTypes;49;Numerical
Values;IDsAndValuesPairsString;C1-C10-N11-C20 1 C1-C10-N11-C21 1 C1-C1
```

```
1-C21-C21 2 C1-C11-C21-N11 2 C1-CS-C1-C10 1 C1-CS-C1-C5 1 C1-CS-C1-CS
2 C10-C1-CS-O2 1 C10-N11-C20-C20 2 C10-N11-C21-C11 1 C10-N11-C21-C21 1
C11-C21-C21-C20 1 C11-C21-C21-C5 1 C11-C21-N11-C20 1 C14-C18-C18-C20
2 C18-C14-C18-C18 2 C18-C18-C14-F 2 C18-C18-C18-C18 4 C18-C18-C18-C...


FingerprintsVector;TopologicalAtomTorsions:SYBYLAtomTypes;26;Numerical
Values;IDsAndValuesPairsString;C.2-C.3-C.3-C.3 1 C.2-C.3-C.3-O.3 1 C.2
-C.ar-C.ar-C.3 1 C.2-C.ar-C.ar-C.ar 2 C.2-C.ar-C.ar-N.ar 1 C.2-N.am-C.
ar-C.ar 2 C.3-C.3-C.2-O.co2 2 C.3-C.3-C.3-C.3 3 C.3-C.3-C.3-N.ar 1 C.3
-C.3-C.3-O.3 3 C.3-C.3-C.ar-C.ar 2 C.3-C.3-C.ar-N.ar 2 C.3-C.3-N.ar-C.
ar 2 C.3-C.ar-C.ar-C.ar 1 C.3-C.ar-N.ar-C.3 1 C.3-C.ar-N.ar-C.ar 1 ...


FingerprintsVector;TopologicalAtomTorsions:TPSAAtomTypes;8;NumericalVa
lues;IDsAndValuesPairsString;N21-None-None-None 9 N7-None-None-None 4
None-N21-None-None 10 None-N7-None-None 3 None-N7-None-O3 1 None-None-
None-None 44 None-None-None-O3 3 None-None-None-O4 5


FingerprintsVector;TopologicalAtomTorsions:UFFAtomTypes;27;NumericalVa
lues;IDsAndValuesPairsString;C_2-C_3-C_3-C_3 1 C_2-C_3-C_3-O_3 1 C_2-C
_R-C_R-C_3 1 C_2-C_R-C_R-C_R 2 C_2-C_R-C_R-N_R 1 C_2-N_3-C_R-C_R 2 C_3
-C_3-C_2-O_2 1 C_3-C_3-C_2-O_3 1 C_3-C_3-C_3-C_3 3 C_3-C_3-C_3-N_R 1 C
_3-C_3-C_3-O_3 3 C_3-C_3-C_R-C_R 2 C_3-C_3-C_R-N_R 2 C_3-C_3-N_R-C_R 2
 C_3-C_R-C_R-C_R 1 C_3-C_R-N_R-C_3 1 C_3-C_R-N_R-C_R 1 C_3-N_R-C_R-...
```

OPTIONS

--AromaticityModel *MDLAromaticityModel* | *TriposAromaticityModel* | *MMFFAromaticityModel* | *ChemAxonBasicAromaticityModel* | *ChemAxonGeneralAromaticityModel* | *DaylightAromaticityModel* | *MayaChemToolsAromaticityModel*

Specify aromaticity model to use during detection of aromaticity. Possible values in the current release are: *MDLAromaticityModel, TriposAromaticityModel, MMFFAromaticityModel, ChemAxonBasicAromaticityModel, ChemAxonGeneralAromaticityModel, DaylightAromaticityModel or MayaChemToolsAromaticityModel*. Default value: *MayaChemToolsAromaticityModel*.

The supported aromaticity model names along with model specific control parameters are defined in AromaticityModelsData.csv, which is distributed with the current release and is available under lib/data directory. Molecule.pm module retrieves data from this file during class instantiation and makes it available to method DetectAromaticity for detecting aromaticity corresponding to a specific model.

-a, --AtomIdentifierType *AtomicInvariantsAtomTypes* | *DREIDINGAtomTypes* | *EStateAtomTypes* | *FunctionalClassAtomTypes* | *MMFF94AtomTypes* | *SLogPAtomTypes* | *SYBYLAtomTypes* | *TPSAAtomTypes* | *UFFAtomTypes*

Specify atom identifier type to use for assignment of initial atom identifier to non-hydrogen atoms during calculation of topological atom torsions fingerprints. Possible values in the current release are: *AtomicInvariantsAtomTypes, DREIDINGAtomTypes, EStateAtomTypes, FunctionalClassAtomTypes, MMFF94AtomTypes, SLogPAtomTypes, SYBYLAtomTypes, TPSAAtomTypes, UFFAtomTypes*. Default value: *AtomicInvariantsAtomTypes*.

--AtomicInvariantsToUse *"AtomicInvariant,AtomicInvariant..."*

This value is used during *AtomicInvariantsAtomTypes* value of a, --AtomIdentifierType option. It's a list of comma separated valid atomic invariant atom types.

Possible values for atomic invariants are: *AS, X, BO, LBO, SB, DB, TB, H, Ar, RA, FC, MN, SM*. Default value: *AS,X,BO,H,FC*.

The atomic invariants abbreviations correspond to:

```
AS = Atom symbol corresponding to element symbol

X<n>   = Number of non-hydrogen atom neighbors or heavy atoms
BO<n> = Sum of bond orders to non-hydrogen atom neighbors or heavy atoms
LBO<n> = Largest bond order of non-hydrogen atom neighbors or heavy atoms
SB<n> = Number of single bonds to non-hydrogen atom neighbors or heavy atoms
DB<n> = Number of double bonds to non-hydrogen atom neighbors or heavy atoms
TB<n> = Number of triple bonds to non-hydrogen atom neighbors or heavy atoms
H<n>   = Number of implicit and explicit hydrogens for atom
Ar     = Aromatic annotation indicating whether atom is aromatic
RA     = Ring atom annotation indicating whether atom is a ring
FC<+n/-n> = Formal charge assigned to atom
MN<n> = Mass number indicating isotope other than most abundant isotope
SM<n> = Spin multiplicity of atom. Possible values: 1 (singlet), 2 (doublet) or
        3 (triplet)
```

Atom type generated by AtomTypes::AtomicInvariantsAtomTypes class corresponds to:

```
AS.X<n>.BO<n>.LBO<n>.<SB><n>.<DB><n>.<TB><n>.H<n>.Ar.RA.FC<+n/-n>.MN<n>.SM<n>
```

Except for AS which is a required atomic invariant in atom types, all other atomic invariants are optional. Atom type

specification doesn't include atomic invariants with zero or undefined values.

In addition to usage of abbreviations for specifying atomic invariants, the following descriptive words are also allowed:

```
X : NumOfNonHydrogenAtomNeighbors or NumOfHeavyAtomNeighbors
BO : SumOfBondOrdersToNonHydrogenAtoms or SumOfBondOrdersToHeavyAtoms
LBO : LargestBondOrderToNonHydrogenAtoms or LargestBondOrderToHeavyAtoms
SB :  NumOfSingleBondsToNonHydrogenAtoms or NumOfSingleBondsToHeavyAtoms
DB : NumOfDoubleBondsToNonHydrogenAtoms or NumOfDoubleBondsToHeavyAtoms
TB : NumOfTripleBondsToNonHydrogenAtoms or NumOfTripleBondsToHeavyAtoms
H :  NumOfImplicitAndExplicitHydrogens
Ar : Aromatic
RA : RingAtom
FC : FormalCharge
MN : MassNumber
SM : SpinMultiplicity
```

AtomTypes::AtomicInvariantsAtomTypes module is used to assign atomic invariant atom types.

--FunctionalClassesToUse *"FunctionalClass1,FunctionalClass2..."*

This value is used during *FunctionalClassAtomTypes* value of a, --AtomIdentifierType option. It's a list of comma separated valid functional classes.

Possible values for atom functional classes are: *Ar, CA, H, HBA, HBD, Hal, NI, PI, RA*. Default value [Ref 24]: *HBD,HBA,PI,NI,Ar,Hal*.

The functional class abbreviations correspond to:

```
HBD: HydrogenBondDonor
HBA: HydrogenBondAcceptor
PI :  PositivelyIonizable
NI : NegativelyIonizable
Ar : Aromatic
Hal : Halogen
H : Hydrophobic
RA : RingAtom
CA : ChainAtom
```

```
 Functional class atom type specification for an atom corresponds to:

    Ar.CA.H.HBA.HBD.Hal.NI.PI.RA
```

AtomTypes::FunctionalClassAtomTypes module is used to assign functional class atom types. It uses following definitions [Ref 60-61, Ref 65-66]:

```
HydrogenBondDonor: NH, NH2, OH
HydrogenBondAcceptor: N[!H], O
PositivelyIonizable: +, NH2
NegativelyIonizable: -, C(=O)OH, S(=O)OH, P(=O)OH
```

--CompoundID *DataFieldName or LabelPrefixString*

This value is --CompoundIDMode specific and indicates how compound ID is generated.

For *DataField* value of --CompoundIDMode option, it corresponds to datafield label name whose value is used as compound ID; otherwise, it's a prefix string used for generating compound IDs like LabelPrefixString<Number>. Default value, *Cmpd*, generates compound IDs which look like Cmpd<Number>.

Examples for *DataField* value of --CompoundIDMode:

```
MolID
ExtReg
```

Examples for *LabelPrefix* or *MolNameOrLabelPrefix* value of --CompoundIDMode:

```
Compound
```

The value specified above generates compound IDs which correspond to Compound<Number> instead of default value of Cmpd<Number>.

--CompoundIDLabel *text*

Specify compound ID column label for CSV/TSV text file(s) used during *CompoundID* value of --DataFieldsMode option. Default value: *CompoundID*.

--CompoundIDMode *DataField* | *MolName* | *LabelPrefix* | *MolNameOrLabelPrefix*

Specify how to generate compound IDs and write to FP or CSV/TSV text file(s) along with generated fingerprints for *FP* | *text* | *all* values of --output option: use a *SDFile(s)* datafield value; use molname line from *SDFile(s)*; generate a sequential ID with specific prefix; use combination of both MolName and LabelPrefix with usage of LabelPrefix values for empty molname

lines.Possible values: *DataField* | *MolName* | *LabelPrefix* | *MolNameOrLabelPrefix*. Default value: *LabelPrefix*.

For *MolNameAndLabelPrefix* value of --CompoundIDMode, molname line in *SDFile(s)* takes precedence over sequential compound IDs generated using *LabelPrefix* and only empty molname values are replaced with sequential compound IDs.

This is only used for *CompoundID* value of --DataFieldsMode option.

--DataFields *"FieldLabel1,FieldLabel2,..."*

Comma delimited list of *SDFiles(s)* data fields to extract and write to CSV/TSV text file(s) along with generated fingerprints for *text* | *all* values of --output option.

This is only used for *Specify* value of --DataFieldsMode option.

Examples:

```
Extreg
MolID,CompoundName
```

-d, --DataFieldsMode *All* | *Common* | *Specify* | *CompoundID*

Specify how data fields in *SDFile(s)* are transferred to output CSV/TSV text file(s) along with generated fingerprints for *text* | *all* values of --output option: transfer all SD data field; transfer SD data files common to all compounds; extract specified data fields; generate a compound ID using molname line, a compound prefix, or a combination of both. Possible values: *All* | *Common* | *specify* | *CompoundID*. Default value: *CompoundID*.

-f, --Filter *Yes* | *No*

Specify whether to check and filter compound data in SDFile(s). Possible values: *Yes or No*. Default value: *Yes*.

By default, compound data is checked before calculating fingerprints and compounds containing atom data corresponding to non-element symbols or no atom data are ignored.

--FingerprintsLabel *text*

SD data label or text file column label to use for fingerprints string in output SD or CSV/TSV text file(s) specified by --output. Default value: *TopologicalAtomTorsionsFingerprints*.

-h, --help

Print this help message.

-k, --KeepLargestComponent *Yes* | *No*

Generate fingerprints for only the largest component in molecule. Possible values: *Yes or No*. Default value: *Yes*.

For molecules containing multiple connected components, fingerprints can be generated in two different ways: use all connected components or just the largest connected component. By default, all atoms except for the largest connected component are deleted before generation of fingerprints.

--OutDelim *comma* | *tab* | *semicolon*

Delimiter for output CSV/TSV text file(s). Possible values: *comma, tab, or semicolon* Default value: *comma*.

--output *SD* | *FP* | *text* | *all*

Type of output files to generate. Possible values: *SD, FP, text, or all*. Default value: *text*.

-o, --overwrite

Overwrite existing files.

-q, --quote *Yes* | *No*

Put quote around column values in output CSV/TSV text file(s). Possible values: *Yes or No*. Default value: *Yes*

-r, --root *RootName*

New file name is generated using the root: <Root>.<Ext>. Default for new file names: <SDFileName><TopologicalAtomTorsionsFP>.<Ext>. The file type determines <Ext> value. The sdf, fpf, csv, and tsv <Ext> values are used for SD, FP, comma/semicolon, and tab delimited text files, respectively.This option is ignored for multiple input files.

-v, --VectorStringFormat *IDsAndValuesString* | *IDsAndValuesPairsString* | *ValuesAndIDsString* | *ValuesAndIDsPairsString*

Format of fingerprints vector string data in output SD, FP or CSV/TSV text file(s) specified by --output option. Possible values: *IDsAndValuesString* | *IDsAndValuesPairsString* | *ValuesAndIDsString* | *ValuesAndIDsPairsString*. Defaultvalue: *IDsAndValuesString*.

Examples:

```
FingerprintsVector;TopologicalAtomTorsions:AtomicInvariantsAtomTypes;3
3;NumericalValues;IDsAndValuesString;C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-
C.X3.BO4 C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-N.X3.BO3 C.X2.BO2.H2-C.X2.BO
2.H2-C.X3.BO3.H1-C.X2.BO2.H2 C.X2.BO2.H2-C.X2.BO2.H2-C.X3.BO3.H1-O...;
2 2 1 1 2 2 1 1 3 4 4 8 4 2 2 6 2 2 1 2 1 1 2 1 1 2 6 2 4 2 1 3 1
```

```
FingerprintsVector;TopologicalAtomTorsions:AtomicInvariantsAtomTypes;3
3;NumericalValues;IDsAndValuesPairsString;C.X1.BO1.H3-C.X3.BO3.H1-C.X3
.BO4-C.X3.BO4 2 C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-N.X3.BO3 2 C.X2.BO2.H
2-C.X2.BO2.H2-C.X3.BO3.H1-C.X2.BO2.H2 1 C.X2.BO2.H2-C.X2.BO2.H2-C.X3.B
O3.H1-O.X1.BO1.H1 1 C.X2.BO2.H2-C.X2.BO2.H2-N.X3.BO3-C.X3.BO4 2 C.X2.B
O2.H2-C.X3.BO3.H1-C.X2.BO2.H2-C.X3.BO3.H1 2 C.X2.BO2.H2-C.X3.BO3.H1...
```

-w, --WorkingDir *DirName*

Location of working directory. Default value: current directory.

EXAMPLES

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesString format and create SampleTATFP.sdf, SampleTATFP.fpf and SampleTATFP.csv files containing sequential compound IDs in CSV file along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl --output all -r SampleTATFP
  -o Sample.sdf
```

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesPairsString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl --VectorStringFormat
  IDsAndValuesPairsString -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using DREIDING atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a DREIDINGAtomTypes
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using E-state atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a EStateAtomTypes
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using functional class atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a FunctionalClassAtomTypes
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using MMFF94 atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a MMFF94AtomTypes
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using SLogP atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a SLogPAtomTypes
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using SYBYL atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a SYBYLAtomTypes
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using TPSA atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a TPSAAtomTypes
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using UFF atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a UFFAtomTypes
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using only AS,X atomic invariants atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a AtomicInvariantsAtomTypes
 --AtomicInvariantsToUse "AS,X" -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing compoundID from molecule name line along with fingerprints vector strings, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a AtomicInvariantsAtomTypes
  --DataFieldsMode CompoundID -CompoundIDMode MolName
  -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing compound IDs using specified data field along with fingerprints vector strings, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a AtomicInvariantsAtomTypes
  --DataFieldsMode CompoundID -CompoundIDMode DataField --CompoundID
  Mol_ID -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing compound ID using combination of molecule name line and an explicit compound prefix along with fingerprints vector strings data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a AtomicInvariantsAtomTypes
  --DataFieldsMode CompoundID -CompoundIDMode MolnameOrLabelPrefix
  --CompoundID Cmpd --CompoundIDLabel MolID -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing specific data fields columns along with fingerprints vector strings, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a AtomicInvariantsAtomTypes
  --DataFieldsMode Specify --DataFields Mol_ID -r SampleTATFP
  -o Sample.sdf
```

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesString format and create a SampleTATFP.csv file containing common data fields columns along with fingerprints vector strings, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a AtomicInvariantsAtomTypes
  --DataFieldsMode Common -r SampleTATFP -o Sample.sdf
```

To generate topological atom torsions fingerprints using atomic invariants atom types in IDsAndValuesString format and create SampleTATFP.sdf, SampleTATFP.fpf and SampleTATFP.csv files containing all data fields columns in CSV file along with fingerprints data, type:

```
% TopologicalAtomTorsionsFingerprints.pl -a AtomicInvariantsAtomTypes
  --DataFieldsMode All  --output all -r SampleTATFP
  -o Sample.sdf
```

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

InfoFingerprintsFiles.pl, SimilarityMatricesFingerprints.pl, AtomNeighborhoodsFingerprints.pl, ExtendedConnectivityFingerprints.pl, MACCSKeysFingerprints.pl, PathLengthFingerprints.pl, TopologicalAtomPairsFingerprints.pl, TopologicalPharmacophoreAtomPairsFingerprints.pl, TopologicalPharmacophoreAtomTripletsFingerprints.pl

COPYRIGHT