

## NAME

TopologicalPharmacophoreAtomPairsFingerprints.pl - Generate topological pharmacophore atom pairs fingerprints for SD files

## SYNOPSIS

TopologicalPharmacophoreAtomPairsFingerprints.pl SDFfile(s)...

```
TopologicalPharmacophoreAtomPairsFingerprints.pl [-a, --AtomTypesToUse "AtomType1, AtomType2..." [
--AtomTypesWeight "AtomType1, Weight1, AtomType2, Weight2..." [--CompoundID DataFieldName or LabelPrefixString
] [--CompoundIDLabel text] [--CompoundIDMode] [--DataFields "FieldLabel1, FieldLabel2,..." ] [-d,
--DataFieldsMode All | Common | Specify | CompoundID] [-f, --Filter Yes | No] [--FingerprintsLabelMode
FingerprintsLabelOnly | FingerprintsLabelWithIDs] [--FingerprintsLabel text] [--FuzzifyAtomPairsCount Yes | No] [
--FuzzificationMode FuzzyBinning | FuzzyBinSmoothing] [--FuzzificationMethodology FuzzyBinning |
FuzzyBinSmoothing] [--FuzzFactor number] [-h, --help] [-k, --KeepLargestComponent Yes | No] [--MinDistance
number] [--MaxDistance number] [-n, --NormalizationMethodology None | ByHeavyAtomsCount | ByAtomTypesCount
] [--OutDelim comma | tab | semicolon] [--output SD | text | both] [-o, --overwrite] [-q, --quote Yes | No] [-r,
--root RootName] [--ValuesPrecision number] [-v, --VectorStringFormat ValuesString, IDsAndValuesString |
IDsAndValuesPairsString | ValuesAndIDsString | ValuesAndIDsPairsString] [-w, --WorkingDir dirname] SDFfile(s)...
```

## DESCRIPTION

Generate topological pharmacophore atom pairs fingerprints [ Ref 60-62, Ref 65, Ref 68 ] for *SDFfile(s)* and create appropriate SD or CSV/TSV text file(s) containing fingerprints vector strings corresponding to molecular fingerprints.

Multiple SDFfile 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.

Based on the values specified for *--AtomTypesToUse*, pharmacophore atom types are assigned to all non-hydrogen atoms in a molecule and a distance matrix is generated. A pharmacophore atom pairs basis set is initialized for all unique possible pairs within *--MinDistance* and *--MaxDistance* range.

Let:

P = Valid pharmacophore atom type

Px = Pharmacophore atom type x

Py = Pharmacophore atom type y

Dmin = Minimum distance corresponding to number of bonds between two atoms

Dmax = Maximum distance corresponding to number of bonds between two atoms

D = Distance corresponding to number of bonds between two atoms

Px-Dn-Py = Pharmacophore atom pair ID for atom types Px and Py at distance Dn

P = Number of pharmacophore atom types to consider

PPDn = Number of possible unique pharmacophore atom pairs at a distance Dn

PPT = Total number of possible pharmacophore atom pairs at all distances between Dmin and Dmax

Then:

$$PPD = (P * (P - 1)) / 2 + P$$

$$PPT = ((Dmax - Dmin) + 1) * ((P * (P - 1)) / 2 + P) \\ = ((Dmax - Dmin) + 1) * PPD$$



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

--AtomTypesWeight "*AtomType1,Weight1,AtomType2,Weight2...*"

Weights of specified pharmacophore atom types to use during calculation of their contribution to atom pair count. Default value: *None*. Valid values: real numbers greater than 0. In general it's comma delimited list of valid atom type and its weight.

The weight values allow to increase the importance of specific pharmacophore atom type in the generated fingerprints. A weight value of 0 for an atom type eliminates its contribution to atom pair count where as weight value of 2 doubles its contribution.

--CompoundID *DataFieldName* or *LabelPrefixString*

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

For *DataField* value of --CompoundID mode 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 --CompoundID mode:

```
MolID
ExtReg
```

Examples for *LabelPrefix* or *MolNameOrLabelPrefix* value of --CompoundID mode:

```
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 CSV/TSV text file(s) along with generated fingerprints for *text* | *both* values of --output option: use a *SDFFile(s)* datafield value; use molname line from *SDFFile(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 --CompoundID mode, molname line in *SDFFile(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 *SDFFile(s)* data fields to extract and write to CSV/TSV text file(s) along with generated fingerprints for *text* | *both* 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 *SDFFile(s)* are transferred to output CSV/TSV text file(s) along with generated fingerprints for *text* | *both* 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 SDFfile(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.

--FingerprintsLabelMode *FingerprintsLabelOnly | FingerprintsLabelWithIDs*

Specify how fingerprints label is generated in conjunction with --FingerprintsLabel option value: use fingerprints label generated only by --FingerprintsLabel option value or append topological atom pair count value IDs to --FingerprintsLabel option value.

Possible values: *FingerprintsLabelOnly | FingerprintsLabelWithIDs*. Default value: *FingerprintsLabelOnly*.

Topological atom pairs IDs appended to --FingerprintsLabel value during *FingerprintsLabelWithIDs* values of --FingerprintsLabelMode correspond to atom pair count values in fingerprint vector string.

--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: *TopologicalPharmacophoreAtomPairsFingerprints*.

--FuzzifyAtomPairsCount *Yes | No*

To fuzzify or not to fuzzify atom pairs count. Possible values: *Yes or No*. Default value: *No*.

--FuzzificationMode *BeforeNormalization | AfterNormalization*

When to fuzzify atom pairs count. Possible values: *BeforeNormalization | AfterNormalizationYes*. Default value: *AfterNormalization*.

--FuzzificationMethodology *FuzzyBinning | FuzzyBinSmoothing*

How to fuzzify atom pairs count. Possible values: *FuzzyBinning | FuzzyBinSmoothing*. Default value: *FuzzyBinning*.

In conjunction with values for options --FuzzifyAtomPairsCount, --FuzzificationMode and --FuzzFactor, --FuzzificationMethodology option is used to fuzzify pharmacophore atom pairs count.

Let:

Px = Pharmacophore atom type x  
 Py = Pharmacophore atom type y  
 PPxy = Pharmacophore atom pair between atom type Px and Py

PPxyDn = Pharmacophore atom pairs count between atom type Px and Py  
 at distance Dn  
 PPxyDn-1 = Pharmacophore atom pairs count between atom type Px and Py  
 at distance Dn - 1  
 PPxyDn+1 = Pharmacophore atom pairs count between atom type Px and Py  
 at distance Dn + 1

FF = FuzzFactor for FuzzyBinning and FuzzyBinSmoothing

Then:

For *FuzzyBinning*:

PPxyDn = PPxyDn (Unchanged)

PPxyDn-1 = PPxyDn-1 + PPxyDn \* FF

PPxyDn+1 = PPxyDn+1 + PPxyDn \* FF

For *FuzzyBinSmoothing*:

PPxyDn = PPxyDn - PPxyDn \* 2FF for Dmin < Dn < Dmax

PPxyDn = PPxyDn - PPxyDn \* FF for Dn = Dmin or Dmax

PPxyDn-1 = PPxyDn-1 + PPxyDn \* FF

PPxyDn+1 = PPxyDn+1 + PPxyDn \* FF

In both fuzzification schemes, a value of 0 for FF implies no fuzzification of occurrence counts. A value of 1 during *FuzzyBinning* corresponds to maximum fuzzification of occurrence counts; however, a value of 1 during *FuzzyBinSmoothing* ends up completely distributing the value over the previous and next distance bins.

So for default value of --FuzzFactor (FF) 0.15, the occurrence count of pharmacophore atom pairs at distance  $D_n$  during *FuzzyBinning* is left unchanged and the counts at distances  $D_n - 1$  and  $D_n + 1$  are incremented by  $PP_{xyD_n} * 0.15$ .

And during *FuzzyBinSmoothing* the occurrence counts at Distance  $D_n$  is scaled back using multiplicative factor of  $(1 - 2 * 0.15)$  and the occurrence counts at distances  $D_n - 1$  and  $D_n + 1$  are incremented by  $PP_{xyD_n} * 0.15$ . In other words, occurrence bin count is smoothed out by distributing it over the previous and next distance value.

--FuzzFactor *number*

Specify by how much to fuzzify atom pairs count. Default value: *0.15*. Valid values: For *FuzzyBinning* value of --FuzzificationMethodology option: *between 0 and 1.0*; For *FuzzyBinSmoothing* value of --FuzzificationMethodology option: *between 0 and 0.5*.

-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.

--MinDistance *number*

Minimum bond distance between atom pairs for generating topological pharmacophore atom pairs. Default value: *1*. Valid values: positive integers and less than --MaxDistance.

--MaxDistance *number*

Maximum bond distance between atom pairs for generating topological pharmacophore atom pairs. Default value: *10*. Valid values: positive integers and greater than --MinDistance.

-n, --NormalizationMethodology *None | ByHeavyAtomsCount | ByAtomTypesCount*

Normalization methodology to use for scaling the occurrence count of pharmacophore atom pairs within specified distance range. Possible values: *None, ByHeavyAtomsCount or ByAtomTypesCount*. Default value: *None*.

--OutDelim *comma | tab | semicolon*

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

--output *SD | text | both*

Type of output files to generate. Possible values: *SD, text, or both*. 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:  $\langle \text{Root} \rangle . \langle \text{Ext} \rangle$ . Default for new file names:  $\langle \text{SDFileName} \rangle \langle \text{TopologicalPharmacophoreAtomPairsFP} \rangle . \langle \text{Ext} \rangle$ . The file type determines  $\langle \text{Ext} \rangle$  value. The *sdf*, *csv*, and *tsv*  $\langle \text{Ext} \rangle$  values are used for *SD*, *comma/semicolon*, and *tab delimited text files*, respectively. This option is ignored for multiple input files.

--ValuesPrecision *number*



```
% TopologicalPharmacophoreAtomPairsFingerprints.pl --MinDistance 1
--MaxDistance 10 --AtomTypesToUse "HBD,HBA,PI, NI" --AtomTypesWeight
"HBD,2,HBA,2,PI,1,NI,1" --NormalizationMethodology ByHeavyAtomsCount
--FuzzifyAtomPairsCount No -r SampleTPAPFP -o Sample.sdf
```

To generate topological pharmacophore atom pairs fingerprints corresponding to distances from 1 through 10 using "HBD,HBA,PI,NI,H" atom types with no weighting of atom types and normalization but with fuzzification of atom pairs count using FuzzyBinning methodology with FuzzFactor value 0.15 and create a SampleTPAPFP.csv file containing sequential compound IDs along with fingerprints vector strings data in ValuesString format, type:

```
% TopologicalPharmacophoreAtomPairsFingerprints.pl --MinDistance 1
--MaxDistance 10 --AtomTypesToUse "HBD,HBA,PI, NI,H" --AtomTypesWeight
"HBD,1,HBA,1,PI,1,NI,1,H,1" --NormalizationMethodology None
--FuzzifyAtomPairsCount Yes --FuzzificationMethodology FuzzyBinning
--FuzzFactor 0.5 -r SampleTPAPFP -o Sample.sdf
```

To generate topological pharmacophore atom pairs fingerprints corresponding to distances from 1 through 10 using default atom types with no weighting, normalization, and fuzzification of atom pairs count and create a SampleTPAPFP.csv file containing compound ID from molecule name line along with fingerprints vector strings data, type:

```
% TopologicalPharmacophoreAtomPairsFingerprints.pl --DataFieldsMode
CompoundID -CompoundIDMode MolName -r SampleTPAPFP -o Sample.sdf
```

To generate topological pharmacophore atom pairs fingerprints corresponding to distances from 1 through 10 using default atom types with no weighting, normalization, and fuzzification of atom pairs count and create a SampleTPAPFP.csv file containing compound IDs using specified data field along with fingerprints vector strings data, type:

```
% TopologicalPharmacophoreAtomPairsFingerprints.pl --DataFieldsMode
CompoundID -CompoundIDMode DataField --CompoundID Mol_ID
-r SampleTPAPFP -o Sample.sdf
```

To generate topological pharmacophore atom pairs fingerprints corresponding to distances from 1 through 10 using default atom types with no weighting, normalization, and fuzzification of atom pairs count and create a SampleTPAPFP.csv file containing compound ID using combination of molecule name line and an explicit compound prefix along with fingerprints vector strings data, type:

```
% TopologicalPharmacophoreAtomPairsFingerprints.pl --DataFieldsMode
CompoundID -CompoundIDMode MolnameOrLabelPrefix
--CompoundID Cmpd --CompoundIDLabel MolID -r SampleTPAPFP -o Sample.sdf
```

To generate topological pharmacophore atom pairs fingerprints corresponding to distances from 1 through 10 using default atom types with no weighting, normalization, and fuzzification of atom pairs count and create a SampleTPAPFP.csv file containing specific data fields columns along with fingerprints vector strings data, type:

```
% TopologicalPharmacophoreAtomPairsFingerprints.pl --DataFieldsMode
Specify --DataFields Mol_ID -r SampleTPAPFP -o Sample.sdf
```

To generate topological pharmacophore atom pairs fingerprints corresponding to distances from 1 through 10 using default atom types with no weighting, normalization, and fuzzification of atom pairs count and create a SampleTPAPFP.csv file containing common data fields columns along with fingerprints vector strings data, type:

```
% TopologicalPharmacophoreAtomPairsFingerprints.pl --DataFieldsMode
Common -r SampleTPAPFP -o Sample.sdf
```

To generate topological pharmacophore atom pairs fingerprints corresponding to distances from 1 through 10 using default atom types with no weighting, normalization, and fuzzification of atom pairs count and create both SampleTPAPFP.csv and SampleTPAPFP.sdf files containing all data fields columns in CSV file along with fingerprints data, type:

```
% TopologicalPharmacophoreAtomPairsFingerprints.pl --DataFieldsMode
```

```
All --output both -r SampleTPAPFP -o Sample.sdf
```

#### AUTHOR

Manish Sud <msud@san.rr.com>

#### SEE ALSO

InfoFingerprintsSDFFiles.pl, InfoFingerprintsTextFiles.pl, SimilarityMatrixSDFFiles.pl, SimilarityMatrixTextFiles.pl, AtomNeighborhoodsFingerprints.pl, ExtendedConnectivityFingerprints.pl, MACCSKeysFingerprints.pl, PathLengthFingerprints.pl, TopologicalAtomPairsFingerprints.pl, TopologicalAtomTorsionsFingerprints.pl, TopologicalPharmacophoreAtomTripletsFingerprints.pl

#### COPYRIGHT

Copyright (C) 2004-2010 Manish Sud. All rights reserved.

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.