

NAME

TopologicalPharmacophoreAtomPairsFingerprints

SYNOPSIS

```
use TopologicalPharmacophoreAtomPairsFingerprints;

use TopologicalPharmacophoreAtomPairsFingerprints qw(:all);
```

DESCRIPTION

TopologicalPharmacophoreAtomPairsFingerprints [Ref 60-62, Ref 65, Ref 68] class provides the following methods:

new, GenerateFingerprints, GetDescription, GetAtomPairIDs, SetAtomTypesToUse, SetAtomTypesWeight, SetFuzzFactor, SetFuzzificationMethodology, SetFuzzificationMode, SetMaxDistance, SetMinDistance, SetNormalizationMethodology, SetValuePrecision, StringifyTopologicalPharmacophoreAtomPairsFingerprints

TopologicalPharmacophoreAtomPairsFingerprints is derived from Fingerprints class which in turn is derived from ObjectProperty base class that provides methods not explicitly defined in TopologicalPharmacophoreAtomPairsFingerprints, Fingerprints or ObjectProperty classes using Perl's AUTOLOAD functionality. These methods are generated on-the-fly for a specified object property:

```
Set<PropertyName>(<PropertyValue>);
$PropertyValue = Get<PropertyName>();
Delete<PropertyName>();
```

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:

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

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

So for default values of Dmin = 1, Dmax = 10 and P = 5,

$$PPD = (5 * (5 - 1)) / 2 + 5 = 15$$

$$PPT = ((10 - 1) + 1) * 15 = 150$$

The pharmacophore atom pairs basis set includes 150 values.

The atom pair IDs correspond to:

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

For example: H-D1-H, H-D2-HBA, PI-D5-PI and so on

Using distance matrix and pharmacophore atom types, occurrence of unique pharmacophore atom pairs is counted. The contribution of each atom type to atom pair interaction is optionally weighted by specified `AtomTypesWeight` before assigning its count to appropriate distance bin. Based on `NormalizationMethodology` option, pharmacophore atom pairs count is optionally normalized. Additionally, pharmacophore atom pairs count is optionally fuzzified before or after the normalization controlled by values of `FuzzifyAtomPairsCount`, `FuzzificationMode`, `FuzzificationMethodology` and `FuzzFactor`.

The final pharmacophore atom pairs count along with atom pair identifiers involving all non-hydrogen atoms, with optional normalization and fuzzification, constitute pharmacophore topological atom pairs fingerprints of the molecule.

For *ArbitrarySize* value of `AtomPairsSetSizeToUse`, the fingerprint vector correspond to only those topological pharmacophore atom pairs which are present and have non-zero count. However, for *FixedSize* value of `AtomPairsSetSizeToUse`, the fingerprint vector contains all possible valid topological pharmacophore atom pairs with both zero and non-zero count values.

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

```
FingerprintsVector;TopologicalPharmacophoreAtomPairs:ArbitrarySize:MinDistance:MaxDistance10;54;NumericalValues;IDsAndValuesString;H-D1-H H-D1-NI HBA-D1-NI HBD-D1-NI H-D2-H H-D2-HBA H-D2-HBD HBA-D2-HBA HBA-D2-HBD H-D3-H H-D3-HBA H-D3-HBD H-D3-NI HBA-D3-NI HBD-D3-NI H-D4-H H-D4-HBA H-D4-HBD HBA-D4-HBA HBA-D4-HBD HBD-D4-HBD H-D5-H H-D5-HBA H-D5-...;
18 1 2 1 22 12 8 1 2 18 6 3 1 1 1 22 13 6 5 7 2 28 9 5 1 1 1 36 16 10 3 4 1 37 10 8 1 35 10 9 3 3 1 28 7 7 4 18 16 12 5 1 2 1
```

```
FingerprintsVector;TopologicalPharmacophoreAtomPairs:FixedSize:MinDistance:MaxDistance10;150;OrderedNumericalValues;ValuesString;18 0 0 1 0 0 0 2 0 0 1 0 0 0 0 22 12 8 0 0 1 2 0 0 0 0 0 0 0 0 18 6 3 1 0 0 0 1 0 0 1 0 0 0 22 13 6 0 0 5 7 0 0 2 0 0 0 0 0 28 9 5 1 0 0 0 1 0 0 1 0 0 0 0 36 16 10 0 0 3 4 0 0 1 0 0 0 0 0 37 10 8 0 0 0 0 1 0 0 0 0 0 0 0 35 10 9 0 0 3 3 0 0 1 0 0 0 0 0 28 7 7 4 0 0 0 0 0 0 0 0 0 0 0 0 0 18...
```

```
FingerprintsVector;TopologicalPharmacophoreAtomPairs:FixedSize:MinDistance:MaxDistance10;150;OrderedNumericalValues;IDsAndValuesString;H-D1-H H-D1-HBA H-D1-HBD H-D1-NI H-D1-PI HBA-D1-HBA HBA-D1-HBD HBA-D1-NI HBA-D1-PI HBD-D1-HBD HBD-D1-NI HBD-D1-PI NI-D1-NI NI-D1-PI PI-D1-PI H-D2-H H-D2-HBA H-D2-HBD H-D2-NI H-D2-PI HBA-D2-HBA HBA-D2-HBD HBA-D2...;
18 0 0 1 0 0 0 2 0 0 1 0 0 0 0 22 12 8 0 0 1 2 0 0 0 0 0 0 0 0 18 6 3 1 0 0 0 1 0 0 0 1 0 0 0 0 22 13 6 0 0 5 7 0 0 2 0 0 0 0 0 28 9 5 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 36 16 10 0 0 3 4 0 0 1 0 0 0 0
```

METHODS

new

```
$TPAPFP = new TopologicalPharmacophoreAtomPairsFingerprints(
    %NamesAndValues);
```

Using specified *TopologicalPharmacophoreAtomPairsFingerprints* property names and values hash, new method creates a new object and returns a reference to newly created *TopologicalPharmacophoreAtomPairsFingerprints* object. By default, the following properties are initialized:

```
Molecule = ''
Type = 'TopologicalPharmacophoreAtomPairs'
MinDistance = 1
MaxDistance = 10
NormalizationMethodology = 'None'
AtomTypesToUse = ['HBD', 'HBA', 'PI', 'NI', 'H']

FuzzifyAtomPairsCount = 0
FuzzificationMode = 'AfterNormalization'
FuzzificationMethodology = 'FuzzyBinning'
FuzzFactor = 0.15

ValuesPrecision = 2
```

Examples:

```
$TPAPFP = new TopologicalPharmacophoreAtomPairsFingerprints(
    'Molecule' => $Molecule);

$TPAPFP = new TopologicalPharmacophoreAtomPairsFingerprints(
    'Molecule' => $Molecule,
    'AtomPairsSetSizeToUse' => 'ArbitrarySize',
    'MinDistance' => 1,
    'MaxDistance' => 10,
```

```

'NormalizationMethodology' => 'None',
'AtomTypesToUse' => ['HBD', 'HBA', 'PI', 'NI', 'H'],
'FuzzifyAtomPairsCount' => 0);

$TPAPFP = new TopologicalPharmacophoreAtomPairsFingerprints(
'Molecule' => $Molecule,
'AtomPairsSetSizeToUse' => 'FizedSize',
'MinDistance' => 1,
'MaxDistance' => 10,
'NormalizationMethodology' => 'None',
'AtomTypesToUse' => ['HBD', 'HBA', 'PI', 'NI', 'H'],
'FuzzifyAtomPairsCount' => 1,
'FuzzificationMethodology' => 'FuzzyBinning',
'FuzzFactor' => 0.15,
'ValuesPrecision' => 2);

$TPAPFP->GenerateFingerprints();
print "$TPAPFP\n";

```

GetDescription

```
$Description = $TopologicalPharmacophoreAtomPairsFP->GetDescription();
```

Returns a string containing description of topological pharmacophore atom pairs fingerprints.

GenerateFingerprints

```
$TopologicalPharmacophoreAtomPairsFP->GenerateFingerprints();
```

Generates topological pharmacophore atom pairs fingerprints and returns *TopologicalPharmacophoreAtomPairsFP*.

GetAtomPairIDs

```

$AtomPairIDsRef = $TopologicalPharmacophoreAtomPairsFP->GetAtomPairIDs();
@AtomPairIDs = $TopologicalPharmacophoreAtomPairsFP->GetAtomPairIDs();

```

Returns atom pair IDs corresponding to atom pairs count values in topological pharmacophore atom pairs fingerprints vector as an array or reference to an array.

SetAtomPairsSetSizeToUse

```
$TopologicalPharmacophoreAtomPairsFP->SetAtomPairsSetSizeToUse($Values);
```

Sets pharmacophore atom pairs set size to use for topological pharmacophore fingerprints generation and returns *TopologicalPharmacophoreAtomPairsFingerprints*.

Possible values for pharmacophore atom pairs set size are: *ArbitrarySize*, *FixedSize*. Default value: *ArbitrarySize*.

For *ArbitrarySize* value of *AtomPairsSetSizeToUse*, the fingerprint vector correspond to only those topological pharmacophore atom pairs which are present and have non-zero count. However, for *FixedSize* value of *AtomPairsSetSizeToUse*, the fingerprint vector contains all possible valid topological pharmacophore atom pairs with both zero and non-zero count values.

SetAtomTypesToUse

```

$TopologicalPharmacophoreAtomPairsFP->SetAtomTypesToUse($ValuesRef);
$TopologicalPharmacophoreAtomPairsFP->SetAtomTypesToUse(@Values);

```

Sets pharmacophore atom types to use for topological pharmacophore fingerprints generation and returns *TopologicalPharmacophoreAtomPairsFingerprints*.

Possible values for pharmacophore atom types are: *Ar*, *CA*, *H*, *HBA*, *HBD*, *Hal*, *NI*, *PI*, *RA*. Default value [Ref 60-62] : *HBD,HBA,PI,NI,H*.

The pharmacophore atom types abbreviations correspond to:

```

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

```

AtomTypes::FunctionalClassAtomTypes module is used to assign pharmacophore 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
```

SetAtomTypesWeight

```
$TopologicalPharmacophoreAtomPairsFP->SetAtomTypesWeight(
    %AtomTypesToWeight);
```

Sets weights of specified pharmacophore atom types to use during calculation of their contribution to atom pair count and returns *TopologicalPharmacophoreAtomPairsFP*. Default values: *1 for each atom type*.

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.

SetFuzzFactor

```
$TopologicalPharmacophoreAtomPairsFP->SetFuzzFactor($Value);
```

Sets fuzz factor value to use during fuzzification of atom pairs count and returns *TopologicalPharmacophoreAtomPairsFP*. Default value: *0.15*.

Valid values: For *FuzzyBinning* value of FuzzificationMethodology: *between 0 and 1.0*; For *FuzzyBinSmoothing* value of FuzzificationMethodology: *between 0 and 0.5*.

SetFuzzificationMethodology

```
$TopologicalPharmacophoreAtomPairsFP->SetFuzzificationMethodology($Value);
```

Sets fuzzification methodology to use for fuzzification of atom pairs count and returns *TopologicalPharmacophoreAtomPairsFP*. Default value: *FuzzyBinning*. Possible values: *FuzzyBinning | FuzzyBinSmoothing*.

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 Dn during FuzzyBinning is left unchanged and the counts at distances Dn - 1 and Dn + 1 are incremented by PPxyDn * 0.15.

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

SetFuzzificationMode

```
$TopologicalPharmacophoreAtomPairsFP->SetFuzzificationMode($Value);
```

Sets fuzzification mode to use for fuzzification of atom pairs count and returns *TopologicalPharmacophoreAtomPairsFP*. Default value: *AfterNormalization*. Possible values: *BeforeNormalization* | *AfterNormalization*.

SetMaxDistance

```
$TopologicalPharmacophoreAtomPairsFP->SetMaxDistance($Value);
```

Sets maximum bond distance between atom pairs for generating topological pharmacophore atom pairs fingerprints and returns *TopologicalPharmacophoreAtomPairsFP*.

SetMinDistance

```
$TopologicalPharmacophoreAtomPairsFP->SetMinDistance($Value);
```

Sets minimum bond distance between atom pairs for generating topological pharmacophore atom pairs fingerprints and returns *TopologicalPharmacophoreAtomPairsFP*.

SetNormalizationMethodology

```
$TopologicalPharmacophoreAtomPairsFP->SetNormalizationMethodology($Value);
```

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

SetValuesPrecision

```
$TopologicalPharmacophoreAtomPairsFP->SetValuesPrecision($Value);
```

Sets precision of atom pairs count real values which might be generated after normalization or fuzzification and returns *TopologicalPharmacophoreAtomPairsFP*. Default: up to 2 decimal places.

StringifyTopologicalPharmacophoreAtomPairsFingerprints

```
$String = $TopologicalPharmacophoreAtomPairsFP->  
StringifyTopologicalPharmacophoreAtomPairsFingerprints();
```

Returns a string containing information about *TopologicalPharmacophoreAtomPairsFingerprints* object.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

Fingerprints.pm, FingerprintsStringUtil.pm, AtomNeighborhoodsFingerprints.pm, AtomTypesFingerprints.pm, EStateIndicesFingerprints.pm, ExtendedConnectivityFingerprints.pm, MACCSKeys.pm, PathLengthFingerprints.pm, TopologicalAtomPairsFingerprints.pm, TopologicalAtomTripletsFingerprints.pm, TopologicalAtomTorsionsFingerprints.pm, TopologicalPharmacophoreAtomTripletsFingerprints.pm

COPYRIGHT

Copyright (C) 2004-2012 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.