

## NAME

TopologicalAtomTripletsFingerprints

## SYNOPSIS

```
use TopologicalAtomTripletsFingerprints;

use TopologicalAtomTripletsFingerprints qw(:all);
```

## DESCRIPTION

TopologicalAtomTripletsFingerprints [ Ref 57, Ref 59, Ref 72 ] class provides the following methods:

new, GenerateFingerprints, GetAtomTripletIDs, GetDescription, SetAtomIdentifierType, SetAtomicInvariantsToUse, SetFunctionalClassesToUse, SetMaxDistance, SetMinDistance, StringifyTopologicalAtomTripletsFingerprints

TopologicalAtomTripletsFingerprints is derived from Fingerprints class which in turn is derived from ObjectProperty base class that provides methods not explicitly defined in TopologicalAtomTripletsFingerprints, 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>();
```

The current release of MayaChemTools supports generation of TopologicalAtomTripletsFingerprints corresponding to following Atomtoml identifierTypes:

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

Based on the values specified for AtomIdentifierType along with other specified parameters such as AtomicInvariantsToUse and FunctionalClassesToUse, initial atom types are assigned to all non-hydrogen atoms in a molecule. Using the distance matrix for the molecule and initial atom types assigned to non-hydrogen atoms, all unique atom triplets within MinDistance and MaxDistance are identified and counted. An atom triplet identifier is generated for each unique atom triplet; the format of atom triplet identifier is:

```
<ATx>-Dyz-<ATy>-Dxz-<ATz>-Dxy
```

```
ATx, ATy, ATz: Atom types assigned to atom x, atom y, and atom z
Dxy: Distance between atom x and atom y
Dxz: Distance between atom x and atom z
Dyz: Distance between atom y and atom z
```

```
where <AT1>-D23 <= <AT2>-D13 <= <AT3>-D12
```

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

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

```
FingerprintsVector;TopologicalAtomTriplets:AtomicInvariantsAtomTypes:M
inDistance1:MaxDistance10;3096;NumericalValues;IDsAndValuesString;C.X1
.B01.H3-D1-C.X1.B01.H3-D1-C.X3.B03.H1-D2 C.X1.B01.H3-D1-C.X2.B02.H2-D1
0-C.X3.B04-D9 C.X1.B01.H3-D1-C.X2.B02.H2-D3-N.X3.B03-D4 C.X1.B01.H3-D1
-C.X2.B02.H2-D4-C.X2.B02.H2-D5 C.X1.B01.H3-D1-C.X2.B02.H2-D6-C.X3...;
1 2 2 2 2 2 2 8 8 4 8 4 4 2 2 2 2 4 2 2 2 4 2 2 2 2 1 2 2 4 4 4 2 2
2 4 4 4 8 4 4 2 4 4 4 2 4 4 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 8...
```

```
FingerprintsVector;TopologicalAtomTriplets:AtomicInvariantsAtomTypes:M
inDistance1:MaxDistance10;3096;NumericalValues;IDsAndValuesPairsString
;C.X1.B01.H3-D1-C.X1.B01.H3-D1-C.X3.B03.H1-D2 1 C.X1.B01.H3-D1-C.X2.B0
2.H2-D10-C.X3.B04-D9 2 C.X1.B01.H3-D1-C.X2.B02.H2-D3-N.X3.B03-D4 2 C.X
1.B01.H3-D1-C.X2.B02.H2-D4-C.X2.B02.H2-D5 2 C.X1.B01.H3-D1-C.X2.B02.H2
-D6-C.X3.B03.H1-D5 2 C.X1.B01.H3-D1-C.X2.B02.H2-D6-C.X3.B03.H1-D7 2...
```



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

```
Molecule = ''
Type = 'TopologicalAtomTriplets'
MinDistance = 1
MaxDistance = 10
UseTriangleInequality = 1
AtomIdentifierType = ''
AtomicInvariantsToUse = ['AS', 'X', 'BO', 'H', 'FC']
FunctionalClassesToUse = ['HBD', 'HBA', 'PI', 'NI', 'Ar', 'Hal']
```

Examples:

```
$TopologicalAtomTripletsFingerprints = new TopologicalAtomTripletsFingerprints(
    'Molecule' => $Molecule,
    'AtomIdentifierType' =>
        'AtomicInvariantsAtomTypes');

$TopologicalAtomTripletsFingerprints = new TopologicalAtomTripletsFingerprints(
    'Molecule' => $Molecule,
    'MinDistance' => 1,
    'MaxDistance' => 10,
    'AtomIdentifierType' =>
        'AtomicInvariantsAtomTypes',
    'AtomicInvariantsToUse' =>
        ['AS', 'X', 'BO', 'H', 'FC'] );

$TopologicalAtomTripletsFingerprints = new TopologicalAtomTripletsFingerprints(
    'Molecule' => $Molecule,
    'AtomIdentifierType' =>
        'DREIDINGAtomTypes');

$TopologicalAtomTripletsFingerprints = new TopologicalAtomTripletsFingerprints(
    'Molecule' => $Molecule,
    'AtomIdentifierType' =>
        'MMFF94AtomTypes');

$TopologicalAtomTripletsFingerprints = new TopologicalAtomTripletsFingerprints(
    'Molecule' => $Molecule,
    'AtomIdentifierType' =>
        'TPSAAtomTypes');

$TopologicalAtomTripletsFingerprints = new TopologicalAtomTripletsFingerprints(
    'Molecule' => $Molecule,
    'MinDistance' => 1,
    'MaxDistance' => 10,
    'AtomIdentifierType' =>
        'FunctionalClassAtomTypes',
    'FunctionalClassesToUse' =>
        ['HBD', 'HBA', 'PI', 'NI', 'Ar', 'Hal']);

$TopologicalAtomTripletsFingerprints->GenerateFingerprints();
print "$TopologicalAtomTripletsFingerprints\n";
```

GetDescription

```
$Return = $TopologicalAtomTripletsFingerprints->GetDescription();
```

Returns a string containing description of topological atom triplets fingerprints.

GenerateFingerprints

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

Generates topological atom triplets fingerprints and returns *TopologicalAtomTripletsFingerprints*.

GetAtomTripletIDs

```
$AtomTripletIDsRef = $TopologicalAtomTripletsFingerprints->GetAtomTripletIDs();
@AtomTripletIDs = $TopologicalAtomTripletsFingerprints->GetAtomTripletIDs();
```

Returns atom triplet IDs corresponding to atom triplets count values in topological atom triplets fingerprints vector as an array or reference to an array.

#### SetAtomIdentifierType

```
$TopologicalAtomTripletsFingerprints->SetAtomIdentifierType($IdentifierType);
```

Sets atom *IdentifierType* to use during atom triplets fingerprints generation and returns *TopologicalAtomTripletsFingerprints*.

Possible values: *AtomicInvariantsAtomTypes*, *DREIDINGAtomTypes*, *EStateAtomTypes*, *FunctionalClassAtomTypes*, *MMFF94AtomTypes*, *SLogPAtomTypes*, *SYBYLAtomTypes*, *TPSAAtomTypes*, *UFFAtomTypes*.

#### SetAtomicInvariantsToUse

```
$TopologicalAtomTripletsFingerprints->SetAtomicInvariantsToUse($ValuesRef);
$TopologicalAtomTripletsFingerprints->SetAtomicInvariantsToUse(@Values);
```

Sets atomic invariants to use during *AtomicInvariantsAtomTypes* value of *AtomIdentifierType* for topological atom triplets fingerprints generation and returns *TopologicalAtomTripletsFingerprints*.

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.

#### SetFunctionalClassesToUse

```
$TopologicalTripletsFingerprints->SetFunctionalClassesToUse($ValuesRef);
$TopologicalTripletsFingerprints->SetFunctionalClassesToUse(@Values);
```

Sets functional classes invariants to use during *FunctionalClassAtomTypes* value of *AtomIdentifierType* for topological atom triplets fingerprints generation and returns *TopologicalAtomTripletsFingerprints*.

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 or None
```

*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
```

#### SetMaxDistance

```
$TopologicalAtomTripletsFingerprints->SetMaxDistance($Distance);
```

Sets maximum distance to use during topological atom triplets fingerprints generation and returns *TopologicalAtomTripletsFingerprints*.

#### SetMinDistance

```
$TopologicalAtomTripletsFingerprints->SetMinDistance($Distance);
```

Sets minimum distance to use during topological atom triplets fingerprints generation and returns *TopologicalAtomTripletsFingerprints*.

#### StringifyTopologicalAtomTripletsFingerprints

```
$String = $TopologicalAtomTripletsFingerprints->
StringifyTopologicalAtomTripletsFingerprints();
```

Returns a string containing information about *TopologicalAtomTripletsFingerprints* 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, TopologicalAtomTorsionsFingerprints.pm, TopologicalPharmacophoreAtomPairsFingerprints.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.