NAME

AtomNeighborhoodsFingerprints

SYNOPSIS

use Fingerprints::AtomNeighborhoodsFingerprints;

use Fingerprints::AtomNeighborhoodsFingerprints qw(:all);

DESCRIPTION

AtomNeighborhoodsFingerprints [Ref 53-56, Ref 73] class provides the following methods:

new, GenerateFingerprints, GetDescription, SetAtomIdentifierType, SetAtomicInvariantsToUse, SetFunctionalClassesToUse, SetMaxNeighborhoodRadius, SetMinNeighborhoodRadius, StringifyAtomNeighborhoodsFingerprints

AtomNeighborhoodsFingerprints is derived from Fingerprints class which in turn is derived from ObjectProperty base class that provides methods not explicitly defined in AtomNeighborhoodsFingerprints, 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 AtomNeighborhoodsFingerprints corresponding to following AtomIdentifierTypes:

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

Based on the values specified for AtomIdentifierType along with other specified sucb as AtomicInvariantsToUse and FunctionalClassesToUse, initial atom types are assigned to all non-hydrogen atoms in a molecule. Using atom neighborhoods around each non-hydrogen central atom corresponding to radii between specified values MinNeighborhoodRadius and MaxNeighborhoodRadius, unique atom types at each radii level are counted and an atom neighborhood identifier is generated.

The format of an atom neighborhood identifier around a central non-hydrogen atom at a specific radius is:

```
NR<n>-<AtomType>-ATC<n>


NR: Neighborhood radius
AtomType: Assigned atom type
ATC: Atom type count
```

The atom neighborhood identifier for non-hydrogen central atom corresponding to all specified radii is generated by concatenating neighborhood identifiers at each radii by colon as a delimiter:

```
NR<n>-<AtomType>-ATC<n>:NR<n>-<AtomType>-ATC<n>:...
```

The atom neighborhood identifiers for all non-hydrogen central atoms at all specified radii are concatenated using space as a delimiter and constitute atom neighborhood fingerprint of the molecule.

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

```
FingerprintsVector;AtomNeighborhoods:AtomicInvariantsAtomTypes:MinRadi
us0:MaxRadius2;41;AlphaNumericalValues;ValuesString;NR0-C.X1.BO1.H3-AT
C1:NR1-C.X3.BO3.H1-ATC1:NR2-C.X1.BO1.H3-ATC1:NR2-C.X3.BO4-ATC1 NR0-C.X
1.BO1.H3-ATC1:NR1-C.X3.BO3.H1-ATC1:NR2-C.X1.BO1.H3-ATC1:NR2-C.X3.BO4-A
TC1 NR0-C.X2.BO2.H2-ATC1:NR1-C.X2.BO2.H2-ATC1:NR1-C.X3.BO3.H1-ATC1:NR2
-C.X2.BO2.H2-ATC1:NR2-N.X3.BO3-ATC1:NR2-O.X1.BO1.H1-ATC1 NR0-C.X2.B...


FingerprintsVector;AtomNeighborhoods:DREIDINGAtomTypes:MinRadius0:MaxR
adius2;41;AlphaNumericalValues;ValuesString;NR0-C_2-ATC1:NR1-C_3-ATC1:
NR1-O_2-ATC1:NR1-O_3-ATC1:NR2-C_3-ATC1 NR0-C_2-ATC1:NR1-C_R-ATC1:NR1-N
_3-ATC1:NR1-O_2-ATC1:NR2-C_R-ATC3 NR0-C_3-ATC1:NR1-C_2-ATC1:NR1-C_3-AT
C1:NR2-C_3-ATC1:NR2-O_2-ATC1:NR2-O_3-ATC2 NR0-C_3-ATC1:NR1-C_3-ATC1:NR
1-N_R-ATC1:NR2-C_3-ATC1:NR2-C_R-ATC2 NR0-C_3-ATC1:NR1-C_3-ATC1:NR2-...


FingerprintsVector;AtomNeighborhoods:EStateAtomTypes:MinRadius0:MaxRad
ius2;41;AlphaNumericalValues;ValuesString;NR0-aaCH-ATC1:NR1-aaCH-ATC1:
NR1-aasC-ATC1:NR2-aaCH-ATC1:NR2-aasC-ATC1:NR2-sF-ATC1 NR0-aaCH-ATC1:NR
1-aaCH-ATC1:NR1-aasC-ATC1:NR2-aaCH-ATC1:NR2-aasC-ATC1:NR2-sF-ATC1 NR0-
aaCH-ATC1:NR1-aaCH-ATC1:NR1-aasC-ATC1:NR2-aaCH-ATC1:NR2-aasC-ATC2 NR0-
aaCH-ATC1:NR1-aaCH-ATC1:NR1-aasC-ATC1:NR2-aaCH-ATC1:NR2-aasC-ATC2 N...


FingerprintsVector;AtomNeighborhoods:FunctionalClassAtomTypes:MinRadiu
```

```
s0:MaxRadius2;41;AlphaNumericalValues;ValuesString;NR0-Ar-ATC1:NR1-Ar-
ATC1:NR1-Ar.HBA-ATC1:NR1-None-ATC1:NR2-Ar-ATC2:NR2-None-ATC4 NR0-Ar-AT
C1:NR1-Ar-ATC2:NR1-Ar.HBA-ATC1:NR2-Ar-ATC5:NR2-None-ATC1 NR0-Ar-ATC1:N
R1-Ar-ATC2:NR1-HBD-ATC1:NR2-Ar-ATC2:NR2-None-ATC1 NR0-Ar-ATC1:NR1-Ar-A
TC2:NR1-Hal-ATC1:NR2-Ar-ATC2 NR0-Ar-ATC1:NR1-Ar-ATC2:NR1-None-ATC1:...

FingerprintsVector;AtomNeighborhoods:MMFF94AtomTypes:MinRadius0:MaxRad
ius2;41;AlphaNumericalValues;ValuesString;NR0-C5A-ATC1:NR1-C5B-ATC1:NR
1-CB-ATC1:NR1-N5-ATC1:NR2-C5A-ATC1:NR2-C5B-ATC1:NR2-CB-ATC3:NR2-CR-ATC
1 NR0-C5A-ATC1:NR1-C5B-ATC1:NR1-CR-ATC1:NR1-N5-ATC1:NR2-C5A-ATC1:NR2-C
5B-ATC1:NR2-C=ON-ATC1:NR2-CR-ATC3 NR0-C5B-ATC1:NR1-C5A-ATC1:NR1-C5B-AT
C1:NR1-C=ON-ATC1:NR2-C5A-ATC1:NR2-CB-ATC1:NR2-CR-ATC1:NR2-N5-ATC1:N...

FingerprintsVector;AtomNeighborhoods:SLogPAtomTypes:MinRadius0:MaxRadi
us2;41;AlphaNumericalValues;ValuesString;NR0-C1-ATC1:NR1-C10-ATC1:NR1-
CS-ATC1:NR2-C1-ATC1:NR2-N11-ATC1:NR2-O2-ATC1 NR0-C1-ATC1:NR1-C11-ATC1:
NR2-C1-ATC1:NR2-C21-ATC1 NR0-C1-ATC1:NR1-C11-ATC1:NR2-C1-ATC1:NR2-C21-
ATC1 NR0-C1-ATC1:NR1-C5-ATC1:NR1-CS-ATC1:NR2-C1-ATC1:NR2-O2-ATC2:NR2-O
9-ATC1 NR0-C1-ATC1:NR1-CS-ATC2:NR2-C1-ATC2:NR2-O2-ATC2 NR0-C10-ATC1...

FingerprintsVector;AtomNeighborhoods:SYBYLAtomTypes:MinRadius0:MaxRadi
us2;41;AlphaNumericalValues;ValuesString;NR0-C.2-ATC1:NR1-C.3-ATC1:NR1
-O.co2-ATC2:NR2-C.3-ATC1 NR0-C.2-ATC1:NR1-C.ar-ATC1:NR1-N.am-ATC1:NR1-
O.2-ATC1:NR2-C.ar-ATC3 NR0-C.3-ATC1:NR1-C.2-ATC1:NR1-C.3-ATC1:NR2-C.3-
ATC1:NR2-O.3-ATC1:NR2-O.co2-ATC2 NR0-C.3-ATC1:NR1-C.3-ATC1:NR1-N.ar-AT
C1:NR2-C.3-ATC1:NR2-C.ar-ATC2 NR0-C.3-ATC1:NR1-C.3-ATC1:NR2-C.3-ATC...

FingerprintsVector;AtomNeighborhoods:TPSAAtomTypes:MinRadius0:MaxRadiu
s2;41;AlphaNumericalValues;ValuesString;NR0-N21-ATC1:NR1-None-ATC3:NR2
-None-ATC5 NR0-N7-ATC1:NR1-None-ATC2:NR2-None-ATC3:NR2-O3-ATC1 NR0-Non
e-ATC1:NR1-N21-ATC1:NR1-None-ATC1:NR2-None-ATC3 NR0-None-ATC1:NR1-N21-
ATC1:NR1-None-ATC2:NR2-None-ATC6 NR0-None-ATC1:NR1-N21-ATC1:NR1-None-A
TC2:NR2-None-ATC6 NR0-None-ATC1:NR1-N7-ATC1:NR1-None-ATC1:NR1-O3-AT...

FingerprintsVector;AtomNeighborhoods:UFFAtomTypes:MinRadius0:MaxRadius
2;41;AlphaNumericalValues;ValuesString;NR0-C_2-ATC1:NR1-C_3-ATC1:NR1-O
_2-ATC1:NR1-O_3-ATC1:NR2-C_3-ATC1 NR0-C_2-ATC1:NR1-C_R-ATC1:NR1-N_3-AT
C1:NR1-O_2-ATC1:NR2-C_R-ATC3 NR0-C_3-ATC1:NR1-C_2-ATC1:NR1-C_3-ATC1:NR
2-C_3-ATC1:NR2-O_2-ATC1:NR2-O_3-ATC2 NR0-C_3-ATC1:NR1-C_3-ATC1:NR1-N_R
-ATC1:NR2-C_3-ATC1:NR2-C_R-ATC2 NR0-C_3-ATC1:NR1-C_3-ATC1:NR2-C_3-A...
```

METHODS

new

```
$NewAtomNeighborhoodsFingerprints = new AtomNeighborhoodsFingerprints(
                                                %NamesAndValues);
```

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

```
Molecule = ''
Type = 'AtomNeighborhoods'
MinNeighborhoodRadius = 0
MaxNeighborhoodRadius = 2
AtomIdentifierType = ''
AtomicInvariantsToUse = ['AS', 'X', 'BO', 'H', 'FC', 'MN']
FunctionalClassesToUse = ['HBD', 'HBA', 'PI', 'NI', 'Ar', 'Hal']
```

Examples:

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

$AtomNeighborhoodsFingerprints = new AtomNeighborhoodsFingerprints(
                            'Molecule' => $Molecule,
                            'MinNeighborhoodRadius' => 0,
                            'MaxNeighborhoodRadius' => 2,
                            'AtomIdentifierType' =>
                                        'AtomicInvariantsAtomTypes',
                            'AtomicInvariantsToUse' =>
                                        ['AS', 'X', 'BO', 'H', 'FC'] );

$AtomNeighborhoodsFingerprints = new AtomNeighborhoodsFingerprints(
                            'Molecule' => $Molecule,
```

```
                                'AtomIdentifierType' =>
                                            'SYBYLAtomTypes');

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

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

    $AtomNeighborhoodsFingerprints = new AtomNeighborhoodsFingerprints(
                                'Molecule' => $Molecule,
                                'MinNeighborhoodRadius' => 0,
                                'MaxNeighborhoodRadius' => 2,
                                'AtomIdentifierType' =>
                                            'FunctionalClassAtomTypes',
                                'FunctionalClassesToUse' =>
                                        ['HBD', 'HBA', 'PI', 'NI', 'Ar', 'Hal'] );

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

GenerateFingerprints

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

Generates atom neighborhood fingerprints and returns *AtomNeighborhoodsFingerprints*.

GetDescription

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

Returns a string containing description of atom neighborhood fingerprints.

SetAtomIdentifierType

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

Sets atom *IdentifierType* to use during atom neighborhood fingerprints generation and returns *AtomNeighborhoodsFingerprints*.

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

SetAtomicInvariantsToUse

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

Sets atomic invariants to use during *AtomicInvariantsAtomTypes* value of *AtomIdentifierType* for atom neighborhood fingerprints generation and returns *AtomNeighborhoodsFingerprints*.

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

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

Sets functional classes invariants to use during *FunctionalClassAtomTypes* value of *AtomIdentifierType* for atom neighborhoods fingerprints generation and returns *AtomNeighborhoodsFingerprints*.

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
```

SetMaxNeighborhoodRadius

```
$AtomNeighborhoodsFingerprints->SetMaxNeighborhoodRadius($Radius);
```

Sets maximum neighborhood radius to use during atom neighborhood fingerprints generation and returns *AtomNeighborhoodsFingerprints*.

SetMinNeighborhoodRadius

```
$AtomNeighborhoodsFingerprints->SetMinNeighborhoodRadius($Radius);
```

Sets minimum neighborhood radius to use during atom neighborhood fingerprints generation and returns *AtomNeighborhoodsFingerprints*.

StringifyAtomNeighborhoodsFingerprints

```
$String = $Fingerprints->StringifyAtomNeighborhoodsFingerprints();
```

Returns a string containing information about *AtomNeighborhoodsFingerprints* object.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

Fingerprints.pm, FingerprintsStringUtil.pm, AtomTypesFingerprints.pm, EStateIndiciesFingerprints.pm, ExtendedConnectivityFingerprints.pm, MACCSKeys.pm, PathLengthFingerprints.pm, TopologicalAtomPairsFingerprints.pm, TopologicalAtomTripletsFingerprints.pm, TopologicalAtomTorsionsFingerprints.pm, TopologicalPharmacophoreAtomPairsFingerprints.pm, TopologicalPharmacophoreAtomTripletsFingerprints.pm

COPYRIGHT

Copyright (C) 2018 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.