NAME

AtomTypesFingerprints

SYNOPSIS

use Fingerprints::AtomTypesFingerprints;

use Fingerprints::AtomTypesFingerprints qw(:all);

DESCRIPTION

AtomTypesFingerprints class provides the following methods:

new, GenerateFingerprints, GetDescription, SetAtomIdentifierType, SetAtomTypesSetToUse, SetAtomicInvariantsToUse, SetFunctionalClassesToUse, SetType, StringifyAtomTypesFingerprints

AtomTypesFingerprints 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 AtomTypesFingerpritns corresponding to following AtomIdentifierTypes:

```
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 or all atoms in a molecule.

Using the assigned atom types and specified Type, one of the following types of fingerprints are generated:

```
AtomTypesCount - A vector containing count of atom types
AtomTypesBits - A bit vector indicating presence/absence of atom types
```

For *AtomTypesCount* fingerprints, two types of atom types set size is allowed:

```
ArbitrarySize - Corresponds to only atom types detected in molecule
FixedSize - Corresponds to fixed number of atom types previously defined
```

For *AtomTypesBits* fingerprints, only *FixedSize* atom type set is allowed.

ArbitrarySize corresponds to atom types detected in a molecule where as *FixedSize* implies a fix number of all possible atom types previously defined for a specific *AtomIdentifierType*.

Fix number of all possible atom types for supported *AtomIdentifierTypes* in current release of MayaChemTools are:

AtomIdentifier	Total	TotalWithoutHydrogens
DREIDINGAtomTypes	37	34
EStateAtomTypes	109	87
MMFF94AtomTypes	212	171
SLogPAtomTypes	72	67
SYBYLAtomTypes	45	44
TPSAAtomTypes	47	47
UFFAtomTypes	126	124

Combination of Type and AtomTypesSetToUse along with AtomtomIdentifierType allows generation of following different atom types fingerprints:

Type	AtomIdentifierType	AtomTypesSetToUse
AtomTypesCount	AtomicInvariantsAtomTypes	ArbitrarySize
AtomTypesCount	DREIDINGAtomTypes	ArbitrarySize
AtomTypesCount	DREIDINGAtomTypes	FixedSize
AtomTypesBits	DREIDINGAtomTypes	FixedSize
AtomTypesCount	EStateAtomTypes	ArbitrarySize
AtomTypesCount	EStateAtomTypes	FixedSize
AtomTypesBits	EStateAtomTypes	FixedSize

```
AtomTypesCount        FunctionalClassAtomTypes     ArbitrarySize

AtomTypesCount        MMFF94AtomTypes              ArbitrarySize
AtomTypesCount        MMFF94AtomTypes              FixedSize
AtomTypesBits         MMFF94AtomTypes              FixedSize

AtomTypesCount        SLogPAtomTypes               ArbitrarySize
AtomTypesCount        SLogPAtomTypes               FixedSize
AtomTypesBits         SLogPAtomTypes               FixedSize

AtomTypesCount        SYBYLAtomTypes               ArbitrarySize
AtomTypesCount        SYBYLAtomTypes               FixedSize
AtomTypesBits         SYBYLAtomTypes               FixedSize

AtomTypesCount        TPSAAtomTypes                FixedSize
AtomTypesBits         TPSAAtomTypes                FixedSize

AtomTypesCount        UFFAtomTypes                 ArbitrarySize
AtomTypesCount        UFFAtomTypes                 FixedSize
AtomTypesBits         UFFAtomTypes                 FixedSize
```

The current release of MayaChemTools generates the following types of atom types fingerprints bit-vector and vector strings:

```
FingerprintsVector;AtomTypesCount:AtomicInvariantsAtomTypes:ArbitraryS
ize;10;NumericalValues;IDsAndValuesString;C.X1.BO1.H3 C.X2.BO2.H2 C.X2
.BO3.H1 C.X3.BO3.H1 C.X3.BO4 F.X1.BO1 N.X2.BO2.H1 N.X3.BO3 O.X1.BO1.H1
O.X1.BO2;2 4 14 3 10 1 1 1 3 2

FingerprintsVector;AtomTypesCount:DREIDINGAtomTypes:ArbitrarySize;8;Nu
mericalValues;IDsAndValuesString;C_2 C_3 C_R F_ N_3 N_R O_2 O_3;2 9 22
1 1 1 2 3

FingerprintsVector;AtomTypesCount:DREIDINGAtomTypes:FixedSize;34;Order
edNumericalValues;IDsAndValuesString;B_3 B_2 C_3 C_R C_2 C_1 N_3 N_R N
_2 N_1 O_3 O_R O_2 O_1 F_ Al3 Si3 P_3 S_3 Cl Ga3 Ge3 As3 Se3 Br In3 Sn
3 Sb3 Te3 I_ Na Ca Fe Zn;0 0 9 22 2 0 1 1 0 0 3 0 2 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

FingerprintsBitVector;AtomTypesBits:DREIDINGAtomTypes:FixedSize;34;Bin
aryString;Ascending;00111011001010100000000000000000000000000

FingerprintsVector;AtomTypesCount:EStateAtomTypes:ArbitrarySize;11;Num
ericalValues;IDsAndValuesString;aaCH aasC aasN dO dssC sCH3 sF sOH ssC
H2 ssNH sssCH;14 8 1 2 2 2 1 3 4 1 3

FingerprintsVector;AtomTypesCount:EStateAtomTypes:FixedSize;87;Ordered
NumericalValues;IDsAndValuesString;sLi ssBe ssssBem sBH2 ssBH sssB sss
sBm sCH3 dCH2 ssCH2 tCH dsCH aaCH sssCH ddC tsC dssC aasC aaaC ssssC s
NH3p sNH2 ssNH2p dNH ssNH aaNH tN sssNHp dsN aaN sssN ddsN aasN ss...;
0 0 0 0 0 0 0 2 0 4 0 0 14 3 0 0 2 8 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 3 2 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...

FingerprintsBitVector;AtomTypesBits:EStateAtomTypes:FixedSize;87;Binar
yString;Ascending;000000010100110011000000100000001011000010000000000
000000000000000000000000000000000000000

FingerprintsVector;AtomTypesCount:FunctionalClassAtomTypes:ArbitrarySi
ze;8;NumericalValues;IDsAndValuesString;Ar Ar.HBA HBA HBA.HBD HBD Hal
NI None;22 1 2 3 1 1 1 10

FingerprintsVector;AtomTypesCount:MMFF94AtomTypes:ArbitrarySize;13;Num
ericalValues;IDsAndValuesString;C5A C5B C=ON CB COO CR F N5 NC=O O=CN
O=CO OC=O OR;2 2 1 18 1 9 1 1 1 1 1 1 2

FingerprintsVector;AtomTypesCount:MMFF94AtomTypes:FixedSize;171;Ordere
dNumericalValues;IDsAndValuesString;CR C=C CSP2 C=O C=N CGD C=OR C=ON
CONN COO COON COOO C=OS C=S C=SN CSO2 CS=O CSS C=P CSP =C= OR OC=O OC=
C OC=N OC=S ONO2 ON=O OSO3 OSO2 OSO OS=O -OS OPO3 OPO2 OPO -OP -O-...;
9 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 18 0 0 0 0 0 0 0 0 ...
```

```
FingerprintsBitVector;AtomTypesBits:MMFF94AtomTypes:FixedSize;171;Bina
ryString;Ascending;1000000101000000000001100000000000000000101000000100
0100000000000000000000000000000000000000000000100000000000000000000000000
0000000011000000000000000001000000000000000000000000000

FingerprintsVector;AtomTypesCount:SLogPAtomTypes:ArbitrarySize;16;Nume
ricalValues;IDsAndValuesString;C1 C10 C11 C14 C18 C20 C21 C22 C5 CS F
N11 N4 O10 O2 O9;5 1 1 1 14 4 2 1 2 2 1 1 1 1 3 1

FingerprintsVector;AtomTypesCount:SLogPAtomTypes:FixedSize;67;OrderedN
umericalValues;IDsAndValuesString;C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C
12 C13 C14 C15 C16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26 C27 CS N1 N
2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14 NS O1 O2 O3 O4 O5 O6 O7 O8
O9 O10 O11 O12 OS F Cl Br I Hal P S1 S2 S3 Me1 Me2;5 0 0 0 2 0 0 0 0 1
1 0 0 1 0 0 0 14 0 4 2 1 0 0 0 0 0 2 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0...

FingerprintsBitVector;AtomTypesBits:SLogPAtomTypes:FixedSize;67;Binary
String;Ascending;1000100001100100010111000001000100000100000100000011
0001000000000000000

FingerprintsVector;AtomTypesCount:SYBYLAtomTypes:ArbitrarySize;9;Numer
icalValues;IDsAndValuesString;C.2 C.3 C.ar F N.am N.ar O.2 O.3 O.co2;2
9 22 1 1 1 1 2 2

FingerprintsVector;AtomTypesCount:SYBYLAtomTypes:FixedSize;44;OrderedN
umericalValues;IDsAndValuesString;C.3 C.2 C.1 C.ar C.cat N.3 N.2 N.1 N
.ar N.am N.pl3 N.4 O.3 O.2 O.co2 S.3 S.2 S.o S.o2 P.3 F Cl Br I ANY HA
L HET Li Na Mg Al Si K Ca Cr.th Cr.oh Mn Fe Co.oh Cu Zn Se Mo Sn;9 2 0
22 0 0 0 0 1 1 0 0 2 1 2 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0

FingerprintsBitVector;AtomTypesBits:SYBYLAtomTypes:FixedSize;44;Binary
String;Ascending;11010000110011100000100000000000000000000000000000

FingerprintsVector;AtomTypesCount:TPSAAtomTypes:FixedSize;47;OrderedNu
mericalValues;IDsAndValuesString;N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N1
2 N13 N14 N15 N16 N17 N18 N19 N20 N21 N22 N23 N24 N25 N26 N O1 O2 O3 O
4 O5 O6 O S1 S2 S3 S4 S5 S6 S7 S P1 P2 P3 P4 P;0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 2 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

FingerprintsBitVector;AtomTypesBits:TPSAAtomTypes:FixedSize;47;BinaryS
tring;Ascending;00000010000000000000100000000110000000000000000000

FingerprintsVector;AtomTypesCount:UFFAtomTypes:ArbitrarySize;8;Numeric
alValues;IDsAndValuesString;C_2 C_3 C_R F_ N_3 N_R O_2 O_3;2 9 22 1 1
1 2 3

FingerprintsVector;AtomTypesCount:UFFAtomTypes;124;OrderedNumerical
Values;IDsAndValuesString;He4+4 Li Be3+2 B_3 B_2 C_3 C_R C_2 C_1 N_3 N_
R N_2 N_1 O_3 O_3_z O_R O_2 O_1 F_ Ne4+4 Na Mg3+2 Al3 Si3 P_3+3 P_3+5 P
_3+q S_3+2 S_3+4 S_3+6 S_R S_2 Cl Ar4+4 K_ Ca6+2 Sc3+3 Ti3+4 Ti6+4 V_3+
;0 0 0 0 0 12 0 3 0 3 0 1 0 2 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...

FingerprintsVector;AtomTypesCount:UFFAtomTypes:FixedSize;124;OrderedNu
mericalValues;IDsAndValuesString;He4+4 Li Be3+2 B_3 B_2 C_3 C_R C_2 C_
1 N_3 N_R N_2 N_1 O_3 O_3_z O_R O_2 O_1 F_ Ne4+4 Na Mg3+2 Al3 Si3 P_3+
3 P_3+5 P_3+q S_3+2 S_3+4 S_3+6 S_R S_2 Cl Ar4+4 K_ Ca6+2 Sc3+3 Ti...;
0 0 0 0 0 9 22 2 0 1 1 0 0 3 0 0 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...

FingerprintsBitVector;AtomTypesBits:UFFAtomTypes:FixedSize;124;BinaryS
tring;Ascending;0000011101100100101000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000
```

METHODS

new

```
$NewAtomTypesFingerprints = new AtomTypesFingerprints(%NamesAndValues);
```

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

```
Molecule = '';
Type = ''
AtomIdentifierType = ''
AtomTypesSetToUse = ''
IgnoreHydrogens = 1
AtomicInvariantsToUse = ['AS', 'X', 'BO', 'H', 'FC', 'MN']
FunctionalClassesToUse = ['HBD', 'HBA', 'PI', 'NI', 'Ar', 'Hal']
```

Examples:

```
$AtomTypesFingerprints = new AtomTypesFingerprints(
                          'Molecule' => $Molecule,
                          'Type' => 'AtomTypesCount',
                          'AtomIdentifierType' =>
                                        'AtomicInvariantsAtomTypes');

$AtomTypesFingerprints = new AtomTypesFingerprints(
                          'Molecule' => $Molecule,
                          'Type' => 'AtomTypesCount',
                          'AtomIdentifierType' =>
                                        'AtomicInvariantsAtomTypes',
                          'AtomicInvariantsToUse' =>
                                        ['AS', 'X', 'BO', 'H', 'FC'] );

$AtomTypesFingerprints = new AtomTypesFingerprints(
                          'Molecule' => $Molecule,
                          'Type' => 'AtomTypesCount',
                          'AtomIdentifierType' =>
                                        'DREIDINGAtomTypes');

$AtomTypesFingerprints = new AtomTypesFingerprints(
                          'Molecule' => $Molecule,
                          'Type' => 'AtomTypesCount',
                          'AtomIdentifierType' =>
                                        'EStateAtomTypes',
                          'AtomTypesSetToUse' =>
                                        'ArbitrarySize');

$AtomTypesFingerprints = new AtomTypesFingerprints(
                          'Molecule' => $Molecule,
                          'Type' => 'AtomTypesCount',
                          'AtomIdentifierType' =>
                                        'SLogPAtomTypes',
                          'AtomTypesSetToUse' =>
                                        'FixedSize');

$AtomTypesFingerprints = new AtomTypesFingerprints(
                          'Molecule' => $Molecule,
                          'Type' => 'AtomTypesBits',
                          'AtomIdentifierType' =>
                                        'MMFF94AtomTypes',
                          'AtomTypesSetToUse' =>
                                        'FixedSize');

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

GenerateFingerprints

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

Generates atom types fingerprints and returns *AtomTypesFingerprints*.

GetDescription

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

Returns a string containing description of atom types fingerprints.

SetAtomIdentifierType

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

Sets atom *IdentifierType* to use during atom types fingerprints generation and returns *AtomTypesFingerprints*.

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

SetAtomTypesSetToUse

```
$AtomTypesFingerprints->SetAtomTypesSetToUse($Value);
```

Sets *Value* of *AtomTypesSetToUse* and returns *AtomTypesFingerprints*. Possible values: *ArbitrarySize or FixedSize*. Default for *AtomTypesCount* value of AtomTypesSetToUse: *ArbitrarySize*.

SetAtomicInvariantsToUse

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

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

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

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

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

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
```

SetType

```
$AtomTypesFingerprints->SetType($Type);
```

Sets type of AtomTypes fingerpritns and returns *AtomTypesFingerprints*. Possible values: *AtomTypesFingerprintsBits or AtomTypesFingerprintsCount*.

StringifyAtomTypesFingerprints

```
$String = $AtomTypesFingerprints->StringifyAtomTypesFingerprints();
```

Returns a string containing information about *AtomTypesFingerprints* object.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

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

COPYRIGHT