

NAME

PathLengthFingerprints

SYNOPSIS

```
use PathLengthFingerprints;

use PathLengthFingerprints qw(:all);
```

DESCRIPTION

PathLengthFingerprints class provides the following methods:

new, GenerateFingerprints, SetMaxLength, SetMinLength, StringifyPathLengthFingerprints

PathLengthFingerprints is derived from Fingerprints class.

For molecules containing rings, atom paths starting from each atom can be traversed in four different ways:

- o Atom paths without any rings and sharing of bonds in traversed paths.
- o Atom paths containing rings and without any sharing of bonds in traversed paths
- o All possible atom paths without any rings and sharing of bonds in traversed paths
- o All possible atom paths containing rings and with sharing of bonds in traversed paths.

Atom path traversal is terminated at the last ring atom. For molecules containing no rings, first two and last two types described above are equivalent.

AllowSharedBonds and AllowRings variables allow generation of different types of paths to be used for fingerprints generation.

In addition to atom symbols, bond symbols are also used to generate a string for atom paths. These atom paths strings are hashed to a 32 bit integer key using TextUtil::HashCode, which in turn is used as a seed for a random number generation in range of 1 to fingerprint size for setting the corresponding bit in bit vector.

UseBondSymbol variable allow generation of atom path strings and consequently fingerprints.

The combination of AllowSharedBonds, AllowRings, and UseBondSymbols allows generation of 8 different types of path length fingerprints:

| AllowSharedBonds | AllowRings | UseBondSymbols | |
|------------------|------------|----------------|--|
| 0 | 0 | 1 | - AtomPathsNoCyclesWithBondSymbols |
| 0 | 1 | 1 | - AtomPathsWithCyclesWithBondSymbols |
| 1 | 0 | 1 | - AllAtomPathsNoCyclesWithBondSymbols |
| 1 | 1 | 1 | - AllAtomPathsWithCyclesWithBondSymbols [DEFAULT] |
| 0 | 0 | 0 | - AtomPathsNoCyclesNoBondSymbols |
| 0 | 1 | 0 | - AtomPathsWithCyclesNoBondSymbols |
| 1 | 0 | 0 | - AllAtomPathsNoCyclesNoBondSymbols |
| 1 | 1 | 0 | - AllAtomPathsWithCyclesNoWithBondSymbols |

METHODS

new

```
$NewPathLengthFingerprints = new PathLengthFingerprints(
                                %NamesAndValues);
```

Using specified *PathLengthFingerprints* 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 = 'PathLength';
Size = 1024;
```

```
MinSize = 32;
MaxSize = 2**32;
FingerprintsBitVector = '';
MinLength = 1;
MaxLength = 8;
AllowSharedBonds = 1;
AllowRings = 1;
UseBondSymbols = 1;
```

Examples:

```
$PathLengthFingerprints = new PathLengthFingerprints(
    'Molecule' => $Molecule);

$PathLengthFingerprints = new PathLengthFingerprints(
    'Molecule' => $Molecule,
    'Size' => 2048,
    'MinLength' => 1,
    'MaxLength' => 8,
    'AllowRings' => 1,
    'AllowSharedBonds' => 1,
    'UseBondSymbols' => 1);
$PathLengthFingerprints->GenerateFingerprints();
print "$PathLengthFingerprints\n";
```

GenerateFingerprints

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

Generates path length fingerprints and returns *PathLengthFingerprints*

SetMaxLength

```
$PathLengthFingerprints->SetMaxLength($Length);
```

Sets maximum value of atom path length to be used during atom path length fingerprints generation and returns *PathLengthFingerprints*

SetMinLength

```
$PathLengthFingerprints->SetMinLength($Length);
```

Sets minimum value of atom path length to be used during atom path length fingerprints generation and returns *PathLengthFingerprints*

StringifyPathLengthFingerprints

```
$String = $PathLengthFingerprints->StringifyPathLengthFingerprints();
```

Returns a string containing information about *PathLengthFingerprints* object

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

BitVector.pm, Fingerprints.pm, FingerprintsBitVector.pm

COPYRIGHT

Copyright (C) 2004-2008 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.