

NAME

RingsCountDescriptors

SYNOPSIS

```
use MolecularDescriptors::RingsCountDescriptors;

use MolecularDescriptors::RingsCountDescriptors qw(:all);
```

DESCRIPTION

RingsCountDescriptors class provides the following methods:

new, GenerateDescriptors, GetDescriptorNames, StringifyRingsCountDescriptors

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

RingsCountDescriptors class doesn't perform any ring or aromaticity detection before counting their number in a molecule. Instead, it assumes ring and aromaticity detection have been performed by caller using DetectRings [Ref 31] and DetectAromaticity methods available in Molecule.

DetectAromaticity method available in Molecule class assigns aromaticity to rings using Huckel rule as explained below:

- o Ring aromaticity is determined using Huckel's rule: a ring containing $4n + 2$ pi electrons is considered aromatic.
- o Heterocyclic rings containing N, O and S atoms fall into two classes: Basic aromatic and Non-basic aromatic. In Basic aromatic heterocyclic rings, heteroatom itself is involved in a double bond. (e.g. Pyridine) However, in non-basic heterocyclic rings, heteroatom might have an attached hydrogen atom and the remaining lone pair contribute to electron delocalization and contributes to $4n + 2$ electrons. (e.g. Pyrrole, Furan)
- o For molecules containing fused rings, each fused ring set is considered as one aromatic system for counting pi electrons to satisfy Huckel's rule; In case of a failure, rings in fused set are treated individually for aromaticity detection. Additionally, non-fused rings are handled on their own during aromaticity detection.

METHODS

new

```
$NewRingsCountDescriptors = new MolecularDescriptors::
    RingsCountDescriptors(
        %NamesAndValues);
```

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

```
Molecule = ''
Type = 'RingsCount'

@DescriptorNames = ('Rings', 'AromaticRings')
@DescriptorValues = ('None', 'None')
```

Examples:

```
$RingsCountDescriptors = new MolecularDescriptors::RingsCountDescriptors(
    'Molecule' => $Molecule);

$RingsCountDescriptors = new MolecularDescriptors::RingsCountDescriptors();

$RingsCountDescriptors->SetMolecule($Molecule);
```

```
$RingsCountDescriptors->GenerateDescriptors();  
print "RingsCountDescriptors: $RingsCountDescriptors\n";
```

GenerateDescriptors

```
$RingsCountDescriptors->GenerateDescriptors();
```

Calculate number of rings and aromatic rings in a molecule and returns *RingsCountDescriptors*.

GetDescriptorNames

```
@DescriptorNames = $RingsCountDescriptors->GetDescriptorNames();  
@DescriptorNames = MolecularDescriptors::RingsCountDescriptors::  
    GetDescriptorNames();
```

Returns all available descriptor names as an array.

StringifyRingsCountDescriptors

```
$String = $RingsCountDescriptors->  
    StringifyRingsCountDescriptors();
```

Returns a string containing information about *RingsCountDescriptors* object.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

MolecularDescriptors.pm, MolecularDescriptorsGenerator.pm

COPYRIGHT

Copyright (C) 2025 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.