

## NAME

PathLengthFingerprints.pl - Generate atom path length based fingerprints for SD files

## SYNOPSIS

PathLengthFingerprints.pl SDFFile(s)...

```
PathLengthFingerprints.pl [--CompoundID DataFieldName or LabelPrefixString] [--CompoundIDLabel text] [
--CompoundIDMode DataField | MolName | LabelPrefix | MolNameOrLabelPrefix] [--DataFields FieldLabel1,FieldLabel2,...
] [-d, --DataFieldsMode All | Common | Specify | CompoundID] [--DetectAromaticity Yes | No] [
--FingerprintsLabel text] [-f, --FingerprintsOutput BinaryString | HexadecimalString | RawBinaryString] [--fold Yes |
No] [--FoldedSize number] [-h, --help] [-i, --IgnoreHydrogens Yes | No] [-k, --KeepLargestComponent Yes |
No] [-m, --mode AtomPathsWithoutRings | AtomPathsWithRings | AllAtomPathsWithoutRings | AllAtomPathsWithRings] [
--MinPathLength number] [--MaxPathLength number] [--OutDelim comma | tab | semicolon] [--output SD | text |
both] [-q, --quote Yes | No] [-r, --root RootName] [-s, --size number] [-u, --UseBondSymbols Yes | No] [-w,
--WorkingDir dirname] SDFFile(s)...
```

## DESCRIPTION

Generate atom path length based fingerprints for *SDFFile(s)* and create appropriate SD or CSV/TSV text containing bit-strings corresponding to molecular fingerprints.

Based on values specified for -m, --mode, --MinPathLength and --MaxPathLength, all appropriate atom paths are generated for each atom in the molecule and collected in a list. For each atom path in the atom paths list, an atom path string is created using atom symbol and bond order and collected in an atom path string list. Duplicate atom path strings are removed from the list. For each unique atom path string, a 32 bit unsigned integer hash key is generated using TextUtil::HashCode function. Using the hash key as a seed for a random number generator, a random integer value between 0 and --Size is used to set corresponding bits in the fingerprint bit-string.

Multiple SDFFile names are separated by spaces. The valid file extensions are *.sdf* and *.sd*. All other file names are ignored. All the SD files in a current directory can be specified either by *\*.sdf* or the current directory name.

## OPTIONS

--CompoundID *DataFieldName* or *LabelPrefixString*

This value is --CompoundIDMode specific and indicates how compound ID is generated.

For *DataField* value of --CompoundIDMode option, it corresponds to datafield label name whose value is used as compound ID; otherwise, it's a prefix string used for generating compound IDs like *LabelPrefixString<Number>*. Default value, *Cmpd*, generates compound IDs which look like *Cmpd<Number>*.

Examples for *DataField* value of --CompoundIDMode:

```
MolID
ExtReg
```

Examples for *LabelPrefix* or *MolNameOrLabelPrefix* value of --CompoundIDMode:

```
Compound
```

The value specified above generates compound IDs which correspond to *Compound<Number>* instead of default value of *Cmpd<Number>*.

--CompoundIDLabel *text*

Specify compound ID column label for CSV/TSV text file(s) used during *CompoundID* value of --DataFieldsMode option. Default: *CompoundID*.

--CompoundIDMode *DataField* | *MolName* | *LabelPrefix* | *MolNameOrLabelPrefix*

Specify how to generate compound IDs and write to CSV/TSV text file(s) along with generated fingerprints for *text* | *both* values of --output option: use a *SDFFile(s)* datafield value; use molname line from *SDFFile(s)*; generate a sequential ID with specific prefix; use combination of both MolName and LabelPrefix with usage of LabelPrefix values for empty molname lines.

Possible values: *DataField* | *MolName* | *LabelPrefix* | *MolNameOrLabelPrefix*. Default: *LabelPrefix*.

For *MolNameAndLabelPrefix* value of --CompoundIDMode, molname line in *SDFFile(s)* takes precedence over sequential compound IDs generated using *LabelPrefix* and only empty molname values are replaced with sequential compound IDs.

This is only used for *CompoundID* value of --DataFieldsMode option.

--DataFields *FieldLabel1,FieldLabel2,...*

Comma delimited list of *SDFFile(s)* data fields to extract and write to CSV/TSV text file(s) along with generated fingerprints for *text* | *both* values of --output option.

This is only used for *Specify* value of --DataFieldsMode option.

Examples:

```
Extreg  
MolID,CompoundName
```

-d, --DataFieldsMode *All* | *Common* | *Specify* | *CompoundID*

Specify how data fields in *SDFFile(s)* are transferred to output CSV/TSV text file(s) along with generated fingerprints for *text* | *both* values of --output option: transfer all SD data field; transfer SD data files common to all compounds; extract specified data fields; generate a compound ID using molname line, a compound prefix, or a combination of both. Possible values: *All* | *Common* | *specify* | *CompoundID*. Default value: *CompoundID*.

--DetectAromaticity *Yes* | *No*

Detect aromaticity before generating fingerprints. Possible values: *Yes* or *No*. Default value: *Yes*.

*no* --DetectAromaticity forces usage of atom and bond aromaticity values from *SDFFile(s)* and skips the step which detects and assigns aromaticity.

--FingerprintsLabel *text*

SD data label or text file column label to use for fingerprints bit-string in output SD or CSV/TSV text file(s) specified by --output. Default value: *PathLenghFingerprints*.

-f, --FingerprintsOutput *BinaryString* | *HexadecimalString* | *RawBinaryString*

Format of fingerprints bit-string data in output SD or CSV/TSV text file(s) specified by --output. Possible values: *BinaryString*, *HexadecimalString*, *RawBinary*. Default value: *HexadecimalString*.

*BinaryString* corresponds to an ASCII string containing 1s and 0s. *HexadecimalString* contains bit values in ASCII hexadecimal format and it's more compact than *BinaryString*. *RawBinary* contains unpacked byte data.

--fold *Yes* | *No*

Fold fingerprints to increase bit density. Possible values: *Yes* or *No*. Default value: *No*.

--FoldedSize *number*

Size of folded fingerprint. Default value: *256*. Valid values correspond to any positive integer which is less than -s, --size and meets the criteria for its value.

Examples:

```
128  
512
```

-h, --help

Print this help message

-i, --IgnoreHydrogens *Yes* | *No*

Ignore hydrogens during fingerprints generation. Possible values: *Yes* or *No*. Default value: *Yes*.

For *yes* value of -i, --IgnoreHydrogens, any explicit hydrogens are also used for generation of atoms path lengths and fingerprints; implicit hydrogens are still ignored.

-k, --KeepLargestComponent *Yes* | *No*

Generate fingerprints for only the largest component in molecule. Possible values: *Yes* or *No*. Default value: *Yes*.

For molecules containing multiple connected components, fingerprints can be generated in two different ways: use all connected components or just the largest connected component. By default, all atoms except for the largest connected component are deleted before generation of fingerprints.

**-m, --mode** *AtomPathsWithoutRings | AtomPathsWithRings | AllAtomPathsWithoutRings | AllAtomPathsWithRings*

Specify type of path length fingerprints to generate for molecules in *SDF* file(s). Possible values: *AtomPathsWithoutRings*, *AtomPathsWithRings*, *AllAtomPathsWithoutRings*, *AllAtomPathsWithRings*. Default value: *AllAtomPathsWithRings*.

For molecules with no rings, first two and last two options are equivalent and generate same set of atom paths starting from each atom with length between *--MinPathLength* and *--MaxPathLength*. However, all these four options can result in the same set of final atom paths for molecules containing fused, bridged or spiro rings.

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

*AtomPathsWithoutRings* - Atom paths containing no rings and without sharing of bonds in traversed paths.

*AtomPathsWithRings* - Atom paths containing rings and without any sharing of bonds in traversed paths.

*AllAtomPathsWithoutRings* - All possible atom paths containing no rings and without any sharing of bonds in traversed paths.

*AllAtomPathsWithRings* - All possible atom paths containing rings and with sharing of bonds in traversed paths.

Atom path traversal is terminated at the ring atom.

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 which in turn is used as a seed for a random number generation in range of 1 to fingerprint size for setting corresponding bit in bit vector.

Based on values specified for *-m*, *--mode*, *--MinPathLength* and *--MaxPathLength*, all appropriate atom paths are generated for each atom in the molecule and collected in a list. For each atom path in the atom paths list, an atom path string is created using atom symbol and bond order and collected in an atom path string list. Atom symbol corresponds to element symbol and characters used to represent bond order are: *1 - None*; *2 - '='*; *3 - '#'*; *1.5 or aromatic - ':'*; *others: bond order value*.

Atom path strings corresponding purely to atom symbols can also be generated using *--UseBondSymbols* option. By default, bond symbols are included in atom path strings. Exclusion of bond symbols in atom path strings results in fingerprints which correspond purely to atom paths without considering bonds.

Duplicate atom path strings are removed from the list. For each unique atom path string, a 32 bit unsigned integer hash key is produced. Using hash key as a seed for a random number generator, a random integer value between 0 and *--Size* is used to set corresponding bit in the fingerprint bit-string.

For molecule containing rings, combination of *-m*, *--mode* and *--UseBondSymbols* allows generation of up to 8 different types of fingerprints:

Default atom path length fingerprints generation for molecules containing rings with *AllAtomPathsWithRings* value for *-m*, *--mode*, *Yes* value for *--UseBondSymbols*, *2* value for *--MinPathLength* and *8* value for *--MaxPathLength* is the most time consuming. Combinations of other options can substantially speed up fingerprint generation for molecules containing complex ring systems.

**--MinPathLength** *number*

Minimum atom path length to include in fingerprints. Default value: *1*. Valid values: positive integers and less than *--MaxPathLength*. Path length of *1* correspond to a path containing only one atom.

**--MaxPathLength** *number*

Maximum atom path length to include in fingerprints. Default value: *8*. Valid values: positive integers and greater than *--MinPathLength*.

**--OutDelim** *comma | tab | semicolon*

Delimiter for output CSV/TSV text file(s). Possible values: *comma*, *tab*, or *semicolon*. Default value: *comma*

**--output** *SD | text | both*

Type of output files to generate. Possible values: *SD*, *text*, or *both*. Default value: *text*

**-o, --overwrite**

Overwrite existing files

**-q, --quote** *Yes | No*

Put quote around column values in output CSV/TSV text file(s). Possible values: *Yes* or *No*. Default value: *Yes*

**-r, --root *RootName***

New file name is generated using the root: <Root>.<Ext>. Default for new file names: <SDFileName><PathLengthFP>.<Ext>. The file type determines <Ext> value. The sdf, csv, and tsv <Ext> values are used for SD, comma/semicolon, and tab delimited text files, respectively. This option is ignored for multiple input files.

**-s, --Size *number***

Size of fingerprints. Default value: **1024**. Valid values correspond to any positive interger which satisfies the following criteria: power of 2, >= 32 and <= 2 \*\* 32.

Examples:

```
256
512
2048
```

**-u, --UseBondSymbols *Yes / No***

Specify whether to use bond symbols for atom paths during generation of atom path strings. Possible values: **Yes or No**. Default value: **Yes**.

**No** value option for **-u, --UseBondSymbols** allows the generation of fingerprints corresponding purely to atoms disregarding all bonds.

**-w, --WorkingDir *DirName***

Location of working directory. Default: current directory

## EXAMPLES

To generate path length fingerprints in hexadecimal bit-string format of size 1024 and create a SampleHexPLFP.csv file containing sequential compound IDs along with fingerprint bit-string data, type

```
% PathLengthFingerprints.pl -o -r SampleHexPLFP Sample.sdf
```

To generate path length fingerprints in binary bit-string format of size 512 and create a SampleBinPLFP.csv file containing sequential compound IDs along with fingerprint bit-string data , type

```
% PathLengthFingerprints.pl -o --FingerprintsOutput BinaryString
--Size 512 -r SampleBinPLFP Sample.sdf
```

To generate folded path length fingerprints in hexadecimal bit-string format and create SampleHexPLFP.sdf and SampleHexPLFP.csv files containing sequential compound IDs along with fingerprint bit-string data, type

```
% PathLengthFingerprints.pl --output both --Fold Yes --FoldedSize 512
-o -r SampleHexPLFP Sample.sdf
```

To generate path length fingerprints corresponding to atom paths containing no rings and without sharing of bonds in hexadecimal bit-string format and create a SampleHexPLFP.csv file containing all data fields along with fingerprint bit-string data, type

```
% PathLengthFingerprints.pl -o -m AtomPathsWithoutRings
--DataFieldsMode All -r SampleHexPLFP Sample.sdf
```

To generate path length fingerprints corresponding to atom paths containing rings and without sharing of bonds in hexadecimal bit-string format and create a SampleHexPLFP.tsv file containing compound IDs derived from combination of molecule name line and an explicit compound prefix and fingerprints column name along with fingerprint bit-string data, type

```
% PathLengthFingerprints.pl -o -m AtomPathsWithRings --DataFieldsMode
CompoundID --CompoundIDMode MolnameOrLabelPrefix --CompoundIDLabel
MolID --FingerprintsLabel PathLengthFP --OutDelim Tab -r SampleHexPLFP
Sample.sdf
```

To generate path length fingerprints in hexadecimal bit-string format and create a SampleHexPLFP.csv file containing sequential compounds ID along with fingerprint bit-string data using aromaticity specified in SD file,

---

```
type % PathLengthFingerprints.pl -o --DetectAromaticity No -r SampleHexPLFP
Sample.sdf
```

To generate path length fingerprints in hexadecimal bit-string format corresponding purely to atoms without any consideration of bonds between the atoms, path lengths between 2 and 6, and create a SampleHexPLFP.csv file containing sequential compound IDs along with fingerprint bit-string data, type

```
% PathLengthFingerprints.pl -o --MinPathLength 2 --MaxPathLength 6
--UseBondSymbols No -r SampleHexPLFP Sample.sdf
```

#### AUTHOR

Manish Sud <msud@san.rr.com>

#### SEE ALSO

InfoFingerprintsSDFFiles.pl, InfoFingerprintsTextFiles.pl, SimilarityMatrixSDFFiles.pl, SimilarityMatrixTextFiles.pl

#### 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.