


```

FingerprintsVector;PathLengthCount:AtomicInvariantsAtomTypes;27;
NumericalValues;IDsAndValuesPairsString;C 8 0 1 C:C 8 C:O 2 C:C:C 9
C:C:O 3 C:O:C 1 C:C:C:C 10 C:C:C:O 4 C:C:O:C 3 C:C:C:C:C 10 ...

FingerprintsVector;PathLengthCount:DREIDINGAtomTypes;141;
NumericalValues;IDsAndValuesPairsString;C_2 3 C_3 12 N_2 1 N_3 3 O_2 2
O_3 2 C_2=N_2 1 C_2=O_2 2 C_2C_3 2 C_2N_3 3 C_2O_3 1 C_3C_3 11 C_3N_3 2
C_3O_3 1 C_2C_3C_3 2 C_2N_3C_3 2 C_3C_2=O_2 2 C_3C_2N_3 1 C_3C_2O_3 1...

FingerprintsVector;PathLengthCount:EStateAtomTypes;226;NumericalValues;
IDsAndValuesPairsString;dNH 1 dO 2 dssC 3 sCH3 3 sNH2 1 sOH 2 ssCH2 3
ssNH 2 sssCH 6 dNH=dssC 1 dO=dssC 2 dssCsCH3 1 dssCsNH2 1 dssCsOH 1
dssCsshNH 2 dssCsshCH 1 sCH3sshCH2 2 sOHsshCH 1 ssCH2sshCH 4 ssNHsshCH 2...

FingerprintsVector;PathLengthCount:FunctionalClassAtomTypes;137;
NumericalValues;IDsAndValuesPairsString;HBA 2 HBA.HBD 2 HBD 3 HBD.PI 1
NI 1 None 14 HBA.HBDNI 1 HBA.HBDNone 1 HBA=NI 1 HBA=None 1 HBD.PINone 1
HBD=None 1 HBDNone 4 NINone 1 NoneNone 12 HBA.HBDNI=HBA 1 HBA.HBDNI...

FingerprintsVector;PathLengthCount:MMFF94AtomTypes;171;NumericalValues;
IDsAndValuesPairsString;C=ON 1 CGD 1 COO 1 CR 12 N=C 1 NC=N 2 NC=O 1
O=CN 1 O=CO 1 OC=O 1 OR 1 C=ON=O=CN 1 C=ONCR 1 C=ONNC=O 1 CGD=N=C 1
CGDNC=N 2 COO=O=CO 1 COOCR 1 COOOC=O 1 CRCR 11 CRNC=N 1 CRNC=O 1 CRO...

FingerprintsVector;PathLengthCount:SLogPAtomTypes;235;NumericalValues;
IDsAndValuesPairsString;C1 6 C2 3 C5 3 CS 3 N1 1 N2 2 N5 1 O2 2 O9 2
C1C1 2 C1C2 3 C1C5 1 C1CS 1 C2C5 1 C2CS 5 C5=N5 1 C5=O9 2 C5N1 1 C5N2 2
C5O2 1 CSN2 2 CSO2 1 C1C1C2 2 C1C2C1 1 C1C2C5 1 C1C2CS 3 C1C5=O9 1...

FingerprintsVector;PathLengthCount:SYBYLAtomTypes;168;NumericalValues;
IDsAndValuesPairsString;C.2 2 C.3 12 C.cat 1 N.am 1 N.pl3 3 O.2 1 O.3 1
O.co2 2 C.2=O.2 1 C.2=O.co2 1 C.2C.3 2 C.2N.am 1 C.2O.co2 1 C.3C.3 11
C.3N.am 1 C.3N.pl3 1 C.3O.3 1 C.cat=N.pl3 1 C.catN.pl3 2 C.2C.3C.3 2 ...

FingerprintsVector;PathLengthCount:TPSAAAtomTypes;105;NumericalValues;
IDsAndValuesPairsString;N10 1 N7 2 N9 1 None 15 O3 2 O4 2 N10None 1
N7None 4 N9=None 1 None=O3 2 NoneNone 13 NoneO4 2 N10None=N9 1
N10NoneN7 1 N7None=N9 1 N7None=O3 1 N7NoneNone 5 NoneN7None 2...

FingerprintsVector;PathLengthCount:UFFAAtomTypes;141;NumericalValues;
IDsAndValuesPairsString;C_2 3 C_3 12 N_2 1 N_3 3 O_2 2 O_3 2 C_2=N_2 1
C_2=O_2 2 C_2C_3 2 C_2N_3 3 C_2O_3 1 C_3C_3 11 C_3N_3 2 C_3O_3 1 C_2C_3
C_3 2 C_2N_3C_3 2 C_3C_2=O_2 2 C_3C_2N_3 1 C_3C_2O_3 1 C_3C_3C_3 13...

```

OPTIONS

-a, --AtomIdentifierType *AtomicInvariantsAtomTypes* | *DREIDINGAtomTypes* | *EStateAtomTypes* | *FunctionalClassAtomTypes* | *MMFF94AtomTypes* | *SLogPAtomTypes* | *SYBYLAtomTypes* | *TPSAAAtomTypes* | *UFFAAtomTypes*

Specify atom identifier type to use during generation of atom path strings corresponding to path length fingerprints. Possible values in the current release are: *AtomicInvariantsAtomTypes*, *DREIDINGAtomTypes*, *EStateAtomTypes*, *FunctionalClassAtomTypes*, *MMFF94AtomTypes*, *SLogPAtomTypes*, *SYBYLAtomTypes*, *TPSAAAtomTypes*, *UFFAAtomTypes*. Default value: *AtomicInvariantsAtomTypes*.

--AtomicInvariantsToUse "*AtomicInvariant1,AtomicInvariant2...*"

This value is used during *AtomicInvariantsAtomTypes* value of a, --AtomIdentifierType option. It's a list of comma separated valid atomic invariant atom types.

Possible values for atomic invariants are: *AS, X, BO, LBO, SB, DB, TB, H, Ar, RA, FC, MN, SM*. Default value: *AS*.

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

Examples:

Benzene: Using value of AS for --AtomicInvariantsToUse, Yes for UseBondSymbols, and AllAtomPathsWithRings for -p, --PathMode, atom path strings generated are:

C C:C C:C:C C:C:C:C C:C:C:C:C C:C:C:C:C:C C:C:C:C:C:C:C

And using AS,X,BO for --AtomicInvariantsToUse generates following atom path strings:

C.X2.B03 C.X2.B03:C.X2.B03 C.X2.B03:C.X2.B03:C.X2.B03
 C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03
 C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03
 C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03
 C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03:C.X2.B03

Urea: Using value of AS for --AtomicInvariantsToUse, Yes for UseBondSymbols, and AllAtomPathsWithRings for -p, --PathMode, atom path strings are:

C N O C=O CN NC=O NCN

And using AS,X,BO for --AtomicInvariantsToUse generates following atom path strings:

C.X3.B04 N.X1.B01 O.X1.B02 C.X3.B04=O.X1.B02
 C.X3.B04N.X1.B01 N.X1.B01C.X3.B04=O.X1.B02
 N.X1.B01C.X3.B04N.X1.B01

--FunctionalClassesToUse "*FunctionalClass1,FunctionalClass2...*"

This value is used during *FunctionalClassAtomTypes* value of a, --AtomIdentifierType option. It's a list of comma separated valid functional classes.

Possible values for atom functional classes are: Ar, CA, H, HBA, HBD, Hal, NI, PI, RA. Default value [Ref 24]:

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.

No --DetectAromaticity value is only allowed using *AtomicInvariantsAtomTypes* value of -a, --AtomIdentifierType options; for all possible values -a, --AtomIdentifierType values, it must be *Yes*.

-f, --Filter *Yes | No*

Specify whether to check and filter compound data in *SDFfile(s)*. Possible values: *Yes or No*. Default value: *Yes*.

By default, compound data is checked before calculating fingerprints and compounds containing atom data corresponding to non-element symbols or no atom data are ignored.

--FingerprintsLabel *text*

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

--fold *Yes | No*

Fold fingerprints to increase bit density during *PathLengthBits* value of -m, --mode option. Possible values: *Yes or No*. Default value: *No*.

--FoldedSize *number*

Size of folded fingerprint during *PathLengthBits* value of *-m*, *--mode* option. 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* *PathLengthBits | PathLengthCount*

Specify type of path length fingerprints to generate for molecules in *SDFFile(s)*. Possible values: *PathLengthBits*, *PathLengthCount*. Default value: *PathLengthBits*.

For *PathLengthBits* value of *-m*, *--mode* option, a fingerprint bit-vector string containing zeros and ones is generated and for *PathLengthCount* value, a fingerprint vector string corresponding to number of atom paths is generated.

--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*.

-n, *--NumOfBitsToSetPerPath* *number*

Number of bits to set per path during generation of fingerprints bit-vector string for *PathLengthBits* value of *-m*, *--mode* option. Default value: **1**. Valid values: positive integers.

--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.

-p, *--PathMode* *AtomPathsWithoutRings | AtomPathsWithRings | AllAtomPathsWithoutRings | AllAtomPathsWithRings*

Specify type of atom paths to use for generating pathlength fingerprints for molecules in *SDFFile(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.

Based on values specified for for `-p`, `--PathMode`, `--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 filtered atom paths list, an atom path string is created using value of `-a`, `--AtomIdentifierType` and specified values to use for a particular atom identifier type. Value of `-u`, `--UseBondSymbols` controls whether bond order symbols are used during generation of atom path string. 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*. 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.

`UseUniquePaths` controls the removal of structurally duplicate atom path strings are removed from the list.

For *PathLengthBits* value of `-m`, `--mode` option, each atom path is hashed to a 32 bit unsigned integer key 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-vector string. Value of `--NumOfBitsToSetPerPaths` option controls the number of time a random number is generated to set corresponding bits.

For *PathLengthCount* value of `-m`, `--mode` option, the number of times an atom path appears is tracked and a fingerprints count-string corresponding to count of atom paths is generated.

For molecule containing rings, combination of `-p`, `--PathMode` and `--UseBondSymbols` allows generation of up to 8 different types of atom path length strings:

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

Default atom path length fingerprints generation for molecules containing rings with *AllAtomPathsWithRings* value for `-p`, `--PathMode`, *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.

Additionally, value for option `-a`, `--AtomIdentifierType` in conjunction with corresponding specified values for atom types changes the nature of atom path length strings and the fingerprints.

`-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: `<SDFFileName><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 integer which satisfies the following criteria: power of 2, ≥ 32 and $\leq 2^{**} 32$.

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.

--UseUniquePaths *Yes | No*

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

No value option for *--UseUniquePaths* allows usage of all atom paths generated by *-p, --PathMode* option value for generation of atom path strings leading to duplicate path count during *PathLengthCount* value of *-m, --mode* option. It doesn't change fingerprint string generated during *PathLengthBits* value of *-m, --mode*.

For example, during *AllAtomPathsWithRings* value of *-p, --PathMode* option, benzene has 12 linear paths of length 2 and 12 cyclic paths length of 7, but only 6 linear paths of length 2 and 1 cyclic path of length 7 are structurally unique.

-v, --VectorStringFormat *IDsAndValuesString | IDsAndValuesPairsString | ValuesAndIDsString | ValuesAndIDsPairsString*

Format of fingerprints vector string data in output SD or CSV/TSV text file(s) specified by *--output* used during *PathLengthCount* value of *-m, --mode* option. Possible values: *IDsAndValuesString | IDsAndValuesPairsString | ValuesAndIDsString | ValuesAndIDsPairsString*. Default value: *IDsAndValuesString*.

Examples:

```
FingerprintsVector;PathLengthCount:AtomicInvariantsAtomTypes;27;
NumericalValues;IDsAndValuesPairsString;C 8 0 1 C:C 8 C:O 2 C:C:C 9
C:C:O 3 C:O:C 1 C:C:C:C 10 C:C:C:O 4 C:C:O:C 3 C:C:C:C:C 10 ...
```

-w, --WorkingDir *DirName*

Location of working directory. Default: current directory.

EXAMPLES

To generate path length fingerprints corresponding to all unique paths from length 1 through 8 in hexadecimal bit-vector string format of size 1024 and create a *SamplePLFPHex.csv* file containing sequential compound IDs along with fingerprints bit-vector strings data, type:

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

To generate path length fingerprints corresponding to all unique paths from length 1 through 8 in hexadecimal bit-vector string format of size 1024 and create both *SamplePLFPHex.csv* and *SamplePLFPHex.sdf* files containing sequential compound IDs in CSV file along with fingerprints bit-vector strings data, type:

```
% PathLengthFingerprints.pl --output both -o -r SamplePLFPHex Sample.sdf
```

To generate path length fingerprints corresponding to all unique paths from length 1 through 8 in binary bit-vector string format of size 1024 and create a *SamplePLFPBin.csv* file containing sequential compound IDs along with fingerprints bit-vector strings data, type:

```
% PathLengthFingerprints.pl --BitStringFormat BinaryString --size 2048
-o -r SamplePLFPBin Sample.sdf
```

To generate path length fingerprints corresponding to count of all unique paths from length 1 through 8 in

IDsAndValuesString format and create a SamplePLFPCount.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% PathLengthFingerprints.pl -m PathLengthCount -o -r SamplePLFPCount
Sample.sdf
```

To generate path length fingerprints corresponding to count of all unique paths from length 1 through 8 in IDsAndValuesString format using E-state atom types and create a SamplePLFPCount.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% PathLengthFingerprints.pl -m PathLengthCount --AtomIdentifierType
EStateAtomTypes -o -r SamplePLFPCount Sample.sdf
```

To generate path length fingerprints corresponding to count of all unique paths from length 1 through 8 in IDsAndValuesString format using SLogP atom types and create a SamplePLFPCount.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% PathLengthFingerprints.pl -m PathLengthCount --AtomIdentifierType
SLogPAtomTypes -o -r SamplePLFPCount Sample.sdf
```

To generate path length fingerprints corresponding to count of all unique paths from length 1 through 8 in IDsAndValuesString format and create a SamplePLFPCount.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% PathLengthFingerprints.pl -m PathLengthCount --VectorStringFormat
ValuesAndIDsPairsString -o -r SamplePLFPCount Sample.sdf
```

To generate path length fingerprints corresponding to count of all unique paths from length 1 through 8 in IDsAndValuesString format using AS,X,BO as atomic invariants and create a SamplePLFPCount.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% PathLengthFingerprints.pl -m PathLengthCount --AtomIdentifierType
AtomicInvariantsAtomTypes --AtomicInvariantsToUse "AS,X,BO" -o
-r SamplePLFPCount Sample.sdf
```

To generate path length fingerprints corresponding to count of all paths from length 1 through 8 in IDsAndValuesString format and create a SamplePLFPCount.csv file containing compound IDs from MolName line along with fingerprints vector strings data, type:

```
% PathLengthFingerprints.pl -m PathLengthCount --UseUniquePaths No
-o --CompoundIDMode MolName -r SamplePLFPCount --UseUniquePaths No
Sample.sdf
```

To generate path length fingerprints corresponding to all unique paths from length 1 through 8 in hexadecimal bit-vector string format of size 512 after folding and create both SamplePLFPHex.csv and SamplePLFPHex.sdf files containing sequential compound IDs along with fingerprints bit-vector strings data, type:

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

To generate path length fingerprints corresponding to all unique paths from length 1 through 8 containing no rings and without sharing of bonds in hexadecimal bit-vector string format of size 1024 and create a SamplePLFPHex.csv file containing sequential compound IDs along with fingerprints bit-vector strings data and all data fields, type:

```
% PathLengthFingerprints.pl -p AtomPathsWithoutRings --DataFieldsMode All
-o -r SamplePLFPHex Sample.sdf
```

To generate path length fingerprints corresponding to all unique paths from length 1 through 8 containing rings and without sharing of bonds in hexadecimal bit-vector string format of size 1024 and create a SamplePLFPHex.tsv file containing compound IDs derived from combination of molecule name line and an explicit compound prefix along with fingerprints bit-vector strings data and all data fields, type:

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

To generate path length fingerprints corresponding to count of all unique paths from length 1 through 8 in IDsAndValuesString format and create a SamplePLFPCount.csv file containing sequential compound IDs along with fingerprints vector strings data using aromaticity specified in SD file, type:

```
% PathLengthFingerprints.pl -m PathLengthCount --DetectAromaticity No
-o -r SamplePLFPCount Sample.sdf
```

To generate path length fingerprints corresponding to all unique paths from length 2 through 6 in hexadecimal bit-vector string format of size 1024 and create a SamplePLFPHex.csv file containing sequential compound IDs along with fingerprints bit-vector strings data, type:

```
% PathLengthFingerprints.pl --MinPathLength 2 --MaxPathLength 6
-o -r SamplePLFPHex Sample.sdf
```

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

InfoFingerprintsSDFiles.pl, InfoFingerprintsTextFiles.pl, SimilarityMatrixSDFiles.pl, SimilarityMatrixTextFiles.pl, AtomNeighborhoodsFingerprints.pl, ExtendedConnectivityFingerprints.pl, MACCSKeysFingerprints.pl, TopologicalAtomPairsFingerprints.pl, TopologicalAtomTorsionsFingerprints.pl, TopologicalPharmacophoreAtomPairsFingerprints.pl, TopologicalPharmacophoreAtomTripletsFingerprints.pl

COPYRIGHT

Copyright (C) 2004-2010 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.