

NAME

ExtendedConnectivityFingerprints.pl - Generate extended connectivity fingerprints for SD files

SYNOPSIS

ExtendedConnectivityFingerprints.pl SDFfile(s)...

```
ExtendedConnectivityFingerprints.pl [-a, --AtomIdentifierType AtomicInvariantsAtomTypes] [
--AtomicInvariantsToUse "AtomicInvariant,AtomicInvariant..."] [--FunctionalClassesToUse
"FunctionalClass1,FunctionalClass2..."] [--CompoundID DataFieldName or LabelPrefixString] [--CompoundIDLabel text]
[--CompoundIDMode] [--DataFields "FieldLabel1,FieldLabel2,..."] [-d, --DataFieldsMode All | Common | Specify |
CompoundID] [-f, --Filter Yes | No] [--FingerprintsLabel text] [-h, --help] [-k, --KeepLargestComponent Yes |
No] [-m, --mode ExtendedConnectivity | ExtendedConnectivityCount] [-n, --NeighborhoodRadius number] [
--OutDelim comma | tab | semicolon] [--output SD | text | both] [-o, --overwrite] [-q, --quote Yes | No] [-r, --root
RootName] [-v, --VectorStringFormat IDsAndValuesString | IDsAndValuesPairsString | ValuesAndIDsString |
ValuesAndIDsPairsString] [-w, --WorkingDir dirname] SDFfile(s)...
```

DESCRIPTION

Generate extended connectivity fingerprints [Ref 48, Ref 52] for *SDFfile(s)* and create appropriate SD or CSV/TSV text file(s) containing fingerprints vector strings corresponding to molecular fingerprints.

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.

The current release of MayaChemTools supports generation of extended connectivity fingerprints corresponding to following *-a, --AtomIdentifierTypes*:

```
AtomicInvariantsAtomTypes, DREIDINGAtomTypes, EStateAtomTypes,
FunctionalClassAtomTypes, MMFF94AtomTypes, SLogPAtomTypes,
SYBYLAtomTypes, TPSAAtomTypes, UFFAtomTypes
```

Based on values specified for *-a, --AtomIdentifierType, --AtomicInvariantsToUse* and *--FunctionalClassesToUse*, initial atom types are assigned to all non-hydrogen atoms in a molecule and these atom types strings are converted into initial atom identifier integers using `TextUtil::HashCode` function. The duplicate atom identifiers are removed.

For *-n, --NeighborhoodRadius* value of 0, the initial set of unique atom identifiers comprises the molecule fingerprints. Otherwise, atom neighborhoods are generated for each non-hydrogen atom up to specified *-n, --NeighborhoodRadius* value. For each non-hydrogen central atom at a specific radius, its neighbors at next radius level along with their bond orders and previously calculated atom identifiers are collected which in turn are used to generate a new integer atom identifier; the bond orders and atom identifier pairs list is first sorted by bond order followed by atom identifiers to make these values graph invariant.

After integer atom identifiers have been generated for all non-hydrogen atoms at all specified neighborhood radii, the duplicate integer atom identifiers corresponding to same hash code value generated using `TextUtil::HashCode` are tracked by keeping the atom identifiers at lower radius. Additionally, all structurally duplicate integer atom identifiers at each specified radius are also tracked by identifying equivalent atoms and bonds corresponding to substructures used for generating atom identifier and keeping integer atom identifier with lowest value.

For *ExtendedConnectivity* value of fingerprints *-m, --mode*, the duplicate identifiers are removed from the list and the unique atom identifiers constitute the extended connectivity fingerprints of a molecule.

For *ExtendedConnectivityCount* value of fingerprints *-m, --mode*, the occurrence of each unique atom identifiers appears is counted and the unique atom identifiers along with their count constitute the extended connectivity fingerprints of a molecule.

The current release of MayaChemTools generates the following types of extended connectivity fingerprints vector strings:

```
FingerprintsVector;ExtendedConnectivity:AtomicInvariantsAtomTypes;14;
AlphaNumericalValues;ValuesString;333564680 1142173602 1481469939
1977749791 2006158649 291020918 443330853 692611812 816539344
1731657806 2039728782 931045615 1273931663 1317501190
```

```
FingerprintsVector;ExtendedConnectivityCount:AtomicInvariantsAtomTypes;
14;NumericalValues;IDsAndValuesString;333564680 1142173602 1481469939
1977749791 2006158649 291020918 443330853 692611812 816539344
1731657806 2039728782 931045615 1273931663 1317501190;1 1 1 1 2 3 2
1 1 1 1 6 2 3
```

```
FingerprintsVector;ExtendedConnectivity:FunctionalClassAtomTypes;11;
AlphaNumericalValues;ValuesString;862102353 981185303 12517955598
10600886 885767127 1452087973 1878436093 2029559552 1465773182
1530666307 2113761516
```

```
FingerprintsVector;ExtendedConnectivityCount:FunctionalClassAtomTypes;
11;NumericalValues;IDsAndValuesString;862102353 981185303 1251795559
810600886 885767127 1452087973 1878436093 2029559552 1465773182
1530666307 2113761516;4 1 1 3 1 3 1 1 6 2 3
```

```
FingerprintsVector;ExtendedConnectivity:DREIDINGAtomTypes;15;Alpha
NumericalValues;ValuesString;231807347 357928343 55012922 184406295
457110059 889520939 45814423 1348717690 1516420830 1534199128
1717300304 191443006 1112070914 1234390980 1447003606
```

```
FingerprintsVector;ExtendedConnectivity:EStateAtomTypes;17;Alpha
NumericalValues;ValuesString;406582964 855201352 2097234755 160913154
745705355 1076302651 1158349170 1496016517 625118768 651926759 719394488
1000989853 2050657618 51481310 755278492 1305278396 2038549440
```

```
FingerprintsVector;ExtendedConnectivity:MMFF94AtomTypes;18;AlphaNumerical
Values;ValuesString;1060998981 1248384926 1521678386 2104274756
124289185 1054288509 1430674316 1907913980 2065055516 346240228 521034016
564893642 1098707420 1668672617 70500360 925073633 1320730945 1826181053
```

```
FingerprintsVector;ExtendedConnectivity:SLogPAtomTypes;17;AlphaNumerical
Values;ValuesString;584713253 1162561799 1213937480 402133841 710983509
1407253949 1556730933 2126735539 595240120 743621032 1192326134
1345825197 1958724617 271986151 625414237 1435532638 1963373071
```

```
FingerprintsVector;ExtendedConnectivity:SYBYLAtomTypes;15;AlphaNumerical
Values;ValuesString;199957044 1678585943 20149180 287462970 899262994
2104279970 127740685 384590638 1338054494 1374156860 1594701903 31761908
160813541 266317439 1021400620
```

```
FingerprintsVector;ExtendedConnectivity:TPSAAAtomTypes;15;AlphaNumerical
Values;ValuesString;862102353 942511871 259370156 329505411 1536501334
1805025917 150007748 185291616 774304872 907763938 1382915700 166259969
953795886 1069431962 1641257490
```

```
FingerprintsVector;ExtendedConnectivity:UFFAtomTypes;15;AlphaNumerical
Values;ValuesString;231807347 357928343 55012922 184406295 457110059
889520939 45814423 1348717690 1516420830 1534199128 1717300304 191443006
1112070914 1234390980 1447003606
```

OPTIONS

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

Specify atom identifier type to use for assignment of initial atom identifier to non-hydrogen atoms during calculation of extended connectivity fingerprints [Ref 48, Ref 52]. Possible values in the current release are: *AtomicInvariantsAtomTypes*, *FunctionalClassAtomTypes*, *DREIDINGAtomTypes*, *EStateAtomTypes*, *MMFF94AtomTypes*, *SLogPAtomTypes*, *SYBYLAtomTypes*, *TPSAAAtomTypes*, *UFFAtomTypes*. Default value: *AtomicInvariantsAtomTypes*.

--AtomicInvariantsToUse "AtomicInvariant,AtomicInvariant..."

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 [Ref 24]: *AS,X,BO,H,FC,MN*.

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.

--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]: *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

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

--CompoundID *DataFieldName* or *LabelPrefixString*

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

For *DataField* value of --CompoundID Mode 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 --CompoundID Mode:

MolID
 ExtReg

Examples for *LabelPrefix* or *MolNameOrLabelPrefix* value of --CompoundID Mode:

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 --CompoundID Mode, 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*.

-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: *ExtendedConnectivityFingerprints*.

-h, --help

Print this help message.

-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 *ExtendedConnectivity | ExtendedConnectivityCount*

Specify type of extended connectivity fingerprints to generate for molecules in *SDFfile(s)*. Possible values: *ExtendedConnectivity or ExtendedConnectivityCount*. Default value: *ExtendedConnectivity*.

For *ExtendedConnectivity* value of fingerprints -m, --mode, a fingerprint vector string containing unique atom identifiers constitute the extended connectivity fingerprints of a molecule.

For *ExtendedConnectivityCount* value of fingerprints -m, --mode, a fingerprint vector containing unique atom identifiers along with their count constitute the extended connectivity fingerprints of a molecule.

-n, --NeighborhoodRadius *number*

Atomic neighborhood radius for generating extended connectivity neighborhoods. Default value: 2. Valid values: ≥ 0 . Neighborhood radius of zero correspond to just the list of non-hydrogen atoms.

Default value of 2 for atomic neighborhood radius generates extended connectivity fingerprints corresponding to path length or diameter value of 4 [Ref 52b].

--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: <SDFFileName><ExtendedConnectivityFP>.<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.

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

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

Default value during <ExtendedConnectivityCount> value of -m, --mode option: *IDsAndValuesString*.

Default value during <ExtendedConnectivity> value of -m, --mode option: *ValuesString*.

Examples:

```
FingerprintsVector;ExtendedConnectivity:AtomicInvariantsAtomTypes;14;
AlphaNumericalValues;ValuesString;333564680 1142173602 1481469939
1977749791 2006158649 291020918 443330853 692611812 816539344
1731657806 2039728782 931045615 1273931663 1317501190

FingerprintsVector;ExtendedConnectivityCount:AtomicInvariantsAtomTypes;
14;NumericalValues;IDsAndValuesString;333564680 1142173602 1481469939
1977749791 2006158649 291020918 443330853 692611812 816539344
1731657806 2039728782 931045615 1273931663 1317501190;1 1 1 1 2 3 2
1 1 1 1 6 2 3
```

-w, --WorkingDir *DirName*

Location of working directory. Default: current directory.

EXAMPLES

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using atomic invariants atom types in vector string format and create a SampleECAIFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -r SampleECAIFP -o Sample.sdf
```

To generate extended connectivity count fingerprints corresponding to neighborhood radius up to 2 using atomic invariants atom types in vector string format and create a SampleECAIFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -m ExtendedConnectivityCount
-r SampleECAIFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using atomic invariants atom types in vector string format and create both SampleECAIFP.csv and SampleECAIFP.sdf files containing sequential compound IDs in CSV file along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl --output both -r SampleECAIFP
-o Sample.sdf
```

To generate extended connectivity count fingerprints corresponding to neighborhood radius up to 2 using atomic invariants atom types in vector string format and create both SampleECAIFP.csv and SampleECAIFP.sdf files containing sequential compound IDs in CSV file along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -m ExtendedConnectivityCount
--output both -r SampleECAIFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using functional class atom types in vector string format and create a SampleECFCFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a FunctionalClassAtomTypes
-r SampleECFCFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using DREIDING atom types in vector string format and create a SampleECFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a DREIDINGAtomTypes
-r SampleECFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using E-state atom types in vector string format and create a SampleECFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a EStateAtomTypes  
-r SampleECFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using MMFF94 atom types in vector string format and create a SampleECFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a MMFF94AtomTypes  
-r SampleECFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using SLogP atom types in vector string format and create a SampleECFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a SLogPAtomTypes  
-r SampleECFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using SYBYL atom types in vector string format and create a SampleECFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a SYBYLAtomTypes  
-r SampleECFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using TPSA atom types in vector string format and create a SampleECFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a TPSAAtomTypes  
-r SampleECFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using UFF atom types in vector string format and create a SampleECFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a UFFAtomTypes  
-r SampleECFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 3 using atomic invariants atom types in vector string format and create a SampleECAIFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a AtomicInvariantsAtomTypes -n 3  
-r SampleECAIFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 3 using functional class atom types in vector string format and create a SampleECFCFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a FunctionalClassAtomTypes -n 3  
-r SampleECFCFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using only AS,X atomic invariants atom types in vector string format and create a SampleECAIFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a AtomicInvariantsAtomTypes  
--AtomicInvariantsToUse "AS,X" -r SampleECAIFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using only HBD,HBA functional class atom types in vector string format and create a SampleECFCFP.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a FunctionalClassAtomTypes
--FunctionalClassesToUse "HBD,HBA" -r SampleECFCFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using atomic invariants atom types in vector string format and create a SampleECAIFP.csv file containing compound ID from molecule name line along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a AtomicInvariantsAtomTypes
--DataFieldsMode CompoundID -CompoundIDMode MolName
-r SampleECAIFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using functional class atom types in vector string format and create a SampleECFCFP.csv file containing compound IDs using specified data field along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a FunctionalClassAtomTypes
--DataFieldsMode CompoundID -CompoundIDMode DataField --CompoundID Mol_ID
-r SampleECFCFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using atomic invariants atom types in vector string format and create a SampleECAIFP.tsv file containing compound ID using combination of molecule name line and an explicit compound prefix along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a AtomicInvariantsAtomTypes
--DataFieldsMode CompoundID -CompoundIDMode MolnameOrLabelPrefix
--CompoundID Cmpd --CompoundIDLabel MolID -r SampleECAIFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using functional class atom types in vector string format and create a SampleECFCFP.csv file containing specific data fields columns along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a FunctionalClassAtomTypes
--DataFieldsMode Specify --DataFields Mol_ID -r SampleECFCFP
-o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using atomic invariants atom types in vector string format and create a SampleECAIFP.tsv file containing common data fields columns along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a AtomicInvariantsAtomTypes
--DataFieldsMode Common -r SampleECAIFP -o Sample.sdf
```

To generate extended connectivity fingerprints corresponding to neighborhood radius up to 2 using functional class atom types in vector string format and create both SampleECFCFP.csv and SampleECFCFP.sdf files containing all data fields columns in CSV file along with fingerprints vector strings data, type:

```
% ExtendedConnectivityFingerprints.pl -a FunctionalClassAtomTypes
--DataFieldsMode All --output both -r SampleECFCFP
-o Sample.sdf
```

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

InfoFingerprintsSDFFiles.pl, InfoFingerprintsTextFiles.pl, SimilarityMatrixSDFFiles.pl, SimilarityMatrixTextFiles.pl, AtomNeighborhoodsFingerprints.pl, MACCSKeysFingerprints.pl, PathLengthFingerprints.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.