NAME

SimilarityMatricesFingerprints.pl - Calculate similarity matrices using fingerprints strings data in SD, FP and CSV/TSV text file(s)

SYNOPSIS

SimilarityMatricesFingerprints.pl SDFile(s) FPFile(s) TextFile(s)...

SimilarityMatricesFingerprints.pl [--alpha *number*] [--beta *number*] [-b, --BitVectorComparisonMode *All* | *"TanimotoSimilarity,[TverskySimilarity, ...]"*] [-c, --ColMode *ColNum* | *ColLabel*] [--CompoundIDCol *col number* | *col name*] [--CompoundIDPrefix *text*] [--CompoundIDField *DataFieldName*] [--CompoundIDMode *DataField* | *MolName* | *LabelPrefix* | *MolNameOrLabelPrefix*] [-d, --detail *InfoLevel*] [-f, --fast] [--FingerprintsCol *col number* | *col name*] [--FingerprintsField *FieldLabel*] [-h, --help] [--InDelim *comma* | *semicolon*] [--InputDataMode *LoadInMemory* | *ScanFile*] [-m, --mode *AutoDetect* | *FingerprintsBitVectorString* | *FingerprintsVectorString*] [--OutDelim *comma* | *tab* | *semicolon*] [--OutMatrixFormat *RowsAndColumns* | *IDPairsAndValue*] [--OutMatrixType *FullMatrix* | *UpperTriangularMatrix* | *LowerTriangularMatrix*] [-o, --overwrite] [-p, --precision *number*] [-q, --quote *Yes* | *No*] [-r, --root *RootName*] [-v, --VectorComparisonMode *All* | *"TanimotoSimilairy, [ManhattanDistance, ...]"*] [--VectorComparisonFormulism *All* | *"AlgebraicForm, [BinaryForm, SetTheoreticForm]"*] [-w, --WorkingDir dirname] SDFile(s) FPFile(s) TextFile(s)...

DESCRIPTION

Calculate similarity matrices using fingerprint bit-vector or vector strings data in *SD, FP and CSV/TSV* text file(s) and generate CSV/TSV text file(s) containing values for specified similarity and distance coefficients.

The scripts SimilarityMatrixSDFiles.pl and SimilarityMatrixTextFiles.pl have been removed from the current release of MayaChemTools and their functionality merged with this script.

The valid *SDFile* extensions are *.sdf* and *.sd*. All SD files in a current directory can be specified either by **.sdf* or the current directory name.

The valid *FPFile* extensions are *.fpf* and *.fp*. All FP files in a current directory can be specified either by **.fpf* or the current directory name.

The valid *TextFile* extensions are *.csv* and *.tsv* for comma/semicolon and tab delimited text files respectively. All other file names are ignored. All text files in a current directory can be specified by **.csv*, **.tsv*, or the current directory name. The --indelim option determines the format of *TextFile(s)*. Any file which doesn't correspond to the format indicated by --indelim option is ignored.

Example of *FP* file containing fingerprints bit-vector string data:

```
#
# Package = MayaChemTools 7.4
# ReleaseDate = Oct 21, 2010
#
# TimeStamp =  Mon Mar 7 15:14:01 2011
#
# FingerprintsStringType = FingerprintsBitVector
#
# Description = PathLengthBits:AtomicInvariantsAtomTypes:MinLength1:...
# Size = 1024
# BitStringFormat = HexadecimalString
# BitsOrder = Ascending
#
Cmpd1 9c8460989ec8a49913991a6603130b0a19e8051c89184414953800cc21510...
Cmpd2 0000002494008400401000420110010019804 10c00000001010088001120...
... ...
... ..
```

Example of *FP* file containing fingerprints vector string data:

```
#
# Package = MayaChemTools 7.4
# ReleaseDate = Oct 21, 2010
#
# TimeStamp =  Mon Mar 7 15:14:01 2011
#
# FingerprintsStringType = FingerprintsVector
#
# Description = PathLengthBits:AtomicInvariantsAtomTypes:MinLength1:...
# VectorStringFormat = IDsAndValuesString
```

```
# VectorValuesType = NumericalValues
#
Cmpd1 338;C F N O C:C C:N C=O CC CF CN CO C:C:C C:C:N C:CC C:CF C:CN C:
N:C C:NC CC:N CC=O CCC CCN CCO CNC NC=O O=CO C:C:C:C C:C:C:N C:C:CC...;
33 1 2 5 21 2 2 12 1 3 3 20 2 10 2 2 1 2 2 2 8 2 5 1 1 1 19 2 8 2 2 2 2
6 2 2 2 2 2 2 2 2 3 2 2 1 4 1 5 1 1 18 6 2 2 1 2 10 2 1 2 1 2 2 2 2 ...
Cmpd2 103;C N O C=N C=O CC CN CO CC=O CCC CCN CCO CNC N=CN NC=O NCN O=C
O C CC=O CCCC CCCN CCCO CCNC CNC=N CNC=O CNCN CCCC=O CCCCC CCCCN CC...;
15 4 4 1 2 13 5 2 2 15 5 3 2 2 1 1 1 2 17 7 6 5 1 1 1 2 15 8 5 7 2 2 2 2
1 2 1 1 3 15 7 6 8 3 4 4 3 2 2 1 2 3 14 2 4 7 4 4 4 4 1 1 1 2 1 1 1 ...
... ...
... ...
```

Example of *SD* file containing fingerprints bit-vector string data:

```
... ...
... ...
$$$$
... ...
... ...
... ...
41 44  0  0  0  0  0  0  0  0999 V2000
 -3.3652    1.4499    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
... ...
2  3  1  0  0  0  0
... ...
M  END
>  <CmpdID>
Cmpd1

>  <PathLengthFingerprints>
FingerprintsBitVector;PathLengthBits:AtomicInvariantsAtomTypes:MinLengt
h1:MaxLength8;1024;HexadecimalString;Ascending;9c8460989ec8a49913991a66
03130b0a19e8051c89184414953800cc2151082844a20104280013086030 8e8204d4028
00831048940e44281c00060449a5000ac80c894114e006321264401600846c050164462
08190410805000304a10205b0100e04c0038ba0fad0209c0ca8b1200012268b61c0026a
aa0660a11014a011d46

$$$$
... ...
... ...
```

Example of CSV *Text* file containing fingerprints bit-vector string data:

```
"CompoundID","PathLengthFingerprints"
"Cmpd1","FingerprintsBitVector;PathLengthBits:AtomicInvariantsAtomTypes
:MinLength1:MaxLength8;1024;HexadecimalString;Ascending;9c8460989ec8a4
9913991a6603130b0a19e8051c89184414953800cc2151082844a20104280013086030
8e8204d402800831048940e44281c00060449a5000ac80c894114e006321264401..."
... ...
... ...
```

The current release of MayaChemTools supports the following types of fingerprint bit-vector and vector strings:

```
FingerprintsVector;AtomNeighborhoods:AtomicInvariantsAtomTypes:MinRadi
us0:MaxRadius2;41;AlphaNumericalValues;ValuesString;NR0-C.X1.BO1.H3-AT
C1:NR1-C.X3.BO3.H1-ATC1:NR2-C.X1.BO1.H3-ATC1:NR2-C.X3.BO4-ATC1 NR0-C.X
1.BO1.H3-ATC1:NR1-C.X3.BO3.H1-ATC1:NR2-C.X1.BO1.H3-ATC1:NR2-C.X3.BO4-A
TC1 NR0-C.X2.BO2.H2-ATC1:NR1-C.X2.BO2.H2-ATC1:NR1-C.X3.BO3.H1-ATC1:NR2
-C.X2.BO2.H2-ATC1:NR2-N.X3.BO3-ATC1:NR2-O.X1.BO1.H1-ATC1 NR0-C.X2.B...

FingerprintsVector;AtomTypesCount:AtomicInvariantsAtomTypes:ArbitraryS
ize;10;NumericalValues;IDsAndValuesString;C.X1.BO1.H3 C.X2.BO2.H2 C.X2
.BO3.H1 C.X3.BO3.H1 C.X3.BO4 F.X1.BO1 N.X2.BO2.H1 N.X3.BO3 O.X1.BO1.H1
O.X1.BO2;2 4 14 3 10 1 1 1 3 2
```

FingerprintsVector;AtomTypesCount:SLogPAtomTypes:ArbitrarySize;16;Nume
ricalValues;IDsAndValuesString;C1 C10 C11 C14 C18 C20 C21 C22 C5 CS F
N11 N4 O10 O2 O9;5 1 1 1 14 4 2 1 2 2 1 1 1 1 3 1

FingerprintsVector;AtomTypesCount:SLogPAtomTypes:FixedSize;67;OrderedN
umericalValues;IDsAndValuesString;C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C
12 C13 C14 C15 C16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26 C27 CS N1 N
2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14 NS O1 O2 O3 O4 O5 O6 O7 O8
O9 O10 O11 O12 OS F Cl Br I Hal P S1 S2 S3 Me1 Me2;5 0 0 0 2 0 0 0 0 1
1 0 0 1 0 0 0 14 0 4 2 1 0 0 0 0 0 2 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0...

FingerprintsVector;EStateIndicies:ArbitrarySize;11;NumericalValues;IDs
AndValuesString;SaaCH SaasC SaasN SdO SdssC SsCH3 SsF SsOH SssCH2 SssN
H SsssCH;24.778 4.387 1.993 25.023 -1.435 3.975 14.006 29.759 -0.073 3
.024 -2.270

FingerprintsVector;EStateIndicies:FixedSize;87;OrderedNumericalValues;
ValuesString;0 0 0 0 0 0 0 3.975 0 -0.073 0 0 24.778 -2.270 0 0 -1.435
4.387 0 0 0 0 0 0 3.024 0 0 0 0 0 0 0 1.993 0 29.759 25.023 0 0 0 0 1
4.006 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0

FingerprintsVector;ExtendedConnectivity:AtomicInvariantsAtomTypes:Radi
us2;60;AlphaNumericalValues;ValuesString;73555770 333564680 352413391
666191900 1001270906 1371674323 1481469939 1977749791 2006158649 21414
08799 49532520 64643108 79385615 96062769 273726379 564565671 85514103
5 906706094 988546669 1018231313 1032696425 1197507444 1331250018 1338
532734 1455473691 1607485225 1609687129 1631614296 1670251330 17303...

FingerprintsVector;ExtendedConnectivityCount:AtomicInvariantsAtomTypes
:Radius2;60;NumericalValues;IDsAndValuesString;73555770 333564680 3524
13391 666191900 1001270906 1371674323 1481469939 1977749791 2006158649
2141408799 49532520 64643108 79385615 96062769 273726379 564565671...;
3 2 1 1 14 1 2 10 4 3 1 1 1 1 2 1 2 1 1 1 2 3 1 1 2 1 3 3 8 2 2 2 6 2
1 2 1 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1

FingerprintsBitVector;ExtendedConnectivityBits:AtomicInvariantsAtomTyp
es:Radius2;1024;BinaryString;Ascending;0000000000000000000000000000100
0000000001010000000110000011000000000000100000000000000000000000100001
1000000110000000000000000000000000001001100000000000000000000000000010000
0000000000000000000000000001000000000000000000001000000000000000000000000
0000000000010000100001000000000000101000000000000000100000000000000...

FingerprintsVector;ExtendedConnectivity:FunctionalClassAtomTypes:Radiu
s2;57;AlphaNumericalValues;ValuesString;24769214 508787397 850393286 8
62102353 981185303 1231636850 1649386610 1941540674 263599683 32920567
1 571109041 639579325 683993318 723853089 810600886 885767127 90326012
7 958841485 981022393 1126908698 1152248391 1317567065 1421489994 1455
632544 1557272891 1826413669 1983319256 2015750777 2029559552 20404...

FingerprintsVector;ExtendedConnectivity:EStateAtomTypes:Radius2;62;Alp
haNumericalValues;ValuesString;25189973 528584866 662581668 671034184
926543080 1347067490 1738510057 1759600920 2034425745 2097234755 21450
44754 96779665 180364292 341712110 345278822 386540408 387387308 50430
1706 617094135 771528807 957666640 997798220 1158349170 1291258082 134
1138533 1395329837 1420277211 1479584608 1486476397 1487556246 1566...

FingerprintsBitVector;MACCSKeyBits;166;BinaryString;Ascending;00000000
0000000000000000000000000000100100001001000000001001000000011100
0100101010111100011011000100110110000011011110100110111111111111011111
1111111111110111000

FingerprintsBitVector;MACCSKeyBits;322;BinaryString;Ascending;11101011
1110011111100101111111000111101100110000000000000011100010000000000000
0000000000000000000000000000000000000000000000010100000000000000000000

```
0000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000110000000000000000000000000000000000
0000000000000000000000000000000000000000

FingerprintsVector;MACCSKeyCount;166;OrderedNumericalValues;ValuesStri
ng;0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 3 0 0 0 0 4 0 0 2 0 0 0 0 0 0 0 0 0 2 0 0 2 0 0 0 0
0 0 0 0 1 1 8 0 0 0 1 0 0 1 0 1 0 1 0 3 1 3 1 0 0 0 1 2 0 11 1 0 0 0
5 0 0 1 2 0 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 4 0 0 1 1 0 4 6 1 1 1 2 1 1
3 5 2 2 0 5 3 5 1 1 2 5 1 2 1 2 4 8 3 5 5 2 2 0 3 5 4 1

FingerprintsVector;MACCSKeyCount;322;OrderedNumericalValues;ValuesStri
ng;14 8 2 0 2 0 4 4 2 1 4 0 0 2 5 10 5 2 1 0 0 2 0 5 13 3 28 5 5 3 0 0
0 4 2 1 1 0 1 1 0 0 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 22 5 3 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11 0 2 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...

FingerprintsBitVector;PathLengthBits:AtomicInvariantsAtomTypes:MinLeng
th1:MaxLength8;1024;BinaryString;Ascending;0010000100110101010110001 10
0100010101011000101001011100110001000010001001101000001001001001001000
0010110100000111001001000000100101010010010000000001100000010100101110 0
0010000001000101010100000100111100110111011011011000000010110111001101
0101100011000000010001000011000010100011101100001000001000100000000...

FingerprintsVector;PathLengthCount:AtomicInvariantsAtomTypes:MinLength
1:MaxLength8;432;NumericalValues;IDsAndValuesPairsString;C.X1.BO1.H3 2
C.X2.BO2.H2 4 C.X2.BO3.H1 14 C.X3.BO3.H1 3 C.X3.BO4 10 F.X1.BO1 1 N.X
2.BO2.H1 1 N.X3.BO3 1 O.X1.BO1.H1 3 O.X1.BO2 2 C.X1.BO1.H3C.X3.BO3.H1
2 C.X2.BO2.H2C.X2.BO2.H2 1 C.X2.BO2.H2C.X3.BO3.H1 4 C.X2.BO2.H2C.X3.BO
4 1 C.X2.BO2.H2N.X3.BO3 1 C.X2.BO3.H1:C.X2.BO3.H1 10 C.X2.BO3.H1:C....

FingerprintsVector;PathLengthCount:MMFF94AtomTypes:MinLength1:MaxLengt
h8;463;NumericalValues;IDsAndValuesPairsString;C5A 2 C5B 2 C=ON 1 CB 1
8 COO 1 CR 9 F 1 N5 1 NC=O 1 O=CN 1 O=CO 1 OC=O 1 OR 2 C5A:C5B 2 C5A:N
5 2 C5ACB 1 C5ACR 1 C5B:C5B 1 C5BC=ON 1 C5BCB 1 C=ON=O=CN 1 C=ONNC=O 1
CB:CB 18 CBF 1 CBNC=O 1 COO=O=CO 1 COOCR 1 COOOC=O 1 CRCR 7 CRN5 1 CR
OR 2 C5A:C5B:C5B 2 C5A:C5BC=ON 1 C5A:C5BCB 1 C5A:N5:C5A 1 C5A:N5CR ...

FingerprintsVector;TopologicalAtomPairs:AtomicInvariantsAtomTypes:MinD
istance1:MaxDistance10;223;NumericalValues;IDsAndValuesString;C.X1.BO1
.H3-D1-C.X3.BO3.H1 C.X2.BO2.H2-D1-C.X2.BO2.H2 C.X2.BO2.H2-D1-C.X3.BO3.
H1 C.X2.BO2.H2-D1-C.X3.BO4 C.X2.BO2.H2-D1-N.X3.BO3 C.X2.BO3.H1-D1-...;
2 1 4 1 1 10 8 1 2 6 1 2 2 1 2 1 2 2 1 2 1 5 1 10 12 2 2 1 2 1 9 1 3 1
1 1 2 2 1 3 6 1 6 14 2 2 2 3 1 3 1 8 2 2 1 3 2 6 1 2 2 5 1 3 1 23 1...

FingerprintsVector;TopologicalAtomPairs:FunctionalClassAtomTypes:MinDi
stance1:MaxDistance10;144;NumericalValues;IDsAndValuesString;Ar-D1-Ar
Ar-D1-Ar.HBA Ar-D1-HBD Ar-D1-Hal Ar-D1-None Ar.HBA-D1-None HBA-D1-NI H
BA-D1-None HBA.HBD-D1-NI HBA.HBD-D1-None HBD-D1-None NI-D1-None No...;
23 2 1 1 2 1 1 1 1 2 1 1 7 28 3 1 3 2 8 2 1 1 1 5 1 5 24 3 3 4 2 13 4
1 1 4 1 5 22 4 4 3 1 19 1 1 1 1 1 2 2 3 1 1 8 25 4 5 2 3 1 26 1 4 1 ...

FingerprintsVector;TopologicalAtomTorsions:AtomicInvariantsAtomTypes;3
3;NumericalValues;IDsAndValuesString;C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-
C.X3.BO4 C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-N.X3.BO3 C.X2.BO2.H2-C.X2.BO
2.H2-C.X3.BO3.H1-C.X2.BO2.H2 C.X2.BO2.H2-C.X2.BO2.H2-C.X3.BO3.H1-O...;
2 2 1 1 2 2 1 1 3 4 4 8 4 2 2 6 2 2 1 2 1 1 2 1 1 2 6 2 4 2 1 3 1

FingerprintsVector;TopologicalAtomTorsions:EStateAtomTypes;36;Numerica
lValues;IDsAndValuesString;aaCH-aaCH-aaCH-aaCH aaCH-aaCH-aaCH-aasC aaC
H-aaCH-aasC-aaCH aaCH-aaCH-aasC-aasC aaCH-aaCH-aasC-sF aaCH-aaCH-aasC-
ssNH aaCH-aasC-aasC-aasC aaCH-aasC-aasC-aasN aaCH-aasC-ssNH-dssC a...;
4 4 8 4 2 2 6 2 2 2 4 3 2 1 3 3 2 2 2 1 2 1 1 1 2 1 1 1 1 1 1 1 2 1 1 2
```

```
FingerprintsVector;TopologicalAtomTriplets:AtomicInvariantsAtomTypes:M
inDistance1:MaxDistance10;3096;NumericalValues;IDsAndValuesString;C.X1
.BO1.H3-D1-C.X1.BO1.H3-D1-C.X3.BO3.H1-D2 C.X1.BO1.H3-D1-C.X2.BO2.H2-D1
0-C.X3.BO4-D9 C.X1.BO1.H3-D1-C.X2.BO2.H2-D3-N.X3.BO3-D4 C.X1.BO1.H3-D1
-C.X2.BO2.H2-D4-C.X2.BO2.H2-D5 C.X1.BO1.H3-D1-C.X2.BO2.H2-D6-C.X3....;
1 2 2 2 2 2 2 2 8 8 4 8 4 4 2 2 2 2 4 2 2 2 4 2 2 2 2 1 2 2 4 4 4 2 2
2 4 4 4 8 4 4 2 4 4 4 2 4 4 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 8...

FingerprintsVector;TopologicalAtomTriplets:SYBYLAtomTypes:MinDistance1
:MaxDistance10;2332;NumericalValues;IDsAndValuesString;C.2-D1-C.2-D9-C
.3-D10 C.2-D1-C.2-D9-C.ar-D10 C.2-D1-C.3-D1-C.3-D2 C.2-D1-C.3-D10-C.3-
D9 C.2-D1-C.3-D2-C.3-D3 C.2-D1-C.3-D2-C.ar-D3 C.2-D1-C.3-D3-C.3-D4 C.2
-D1-C.3-D3-N.ar-D4 C.2-D1-C.3-D3-O.3-D2 C.2-D1-C.3-D4-C.3-D5 C.2-D1-C.
3-D5-C.3-D6 C.2-D1-C.3-D5-O.3-D4 C.2-D1-C.3-D6-C.3-D7 C.2-D1-C.3-D7...

FingerprintsVector;TopologicalPharmacophoreAtomPairs:ArbitrarySize:Min
Distance1:MaxDistance10;54;NumericalValues;IDsAndValuesString;H-D1-H H
-D1-NI HBA-D1-NI HBD-D1-NI H-D2-H H-D2-HBA H-D2-HBD HBA-D2-HBA HBA-D2-
HBD H-D3-H H-D3-HBA H-D3-HBD H-D3-NI HBA-D3-NI HBD-D3-NI H-D4-H H-D4-H
BA H-D4-HBD HBA-D4-HBA HBA-D4-HBD HBD-D4-HBD H-D5-H H-D5-HBA H-D5-...;
18 1 2 1 22 12 8 1 2 18 6 3 1 1 1 22 13 6 5 7 2 28 9 5 1 1 1 36 16 10
3 4 1 37 10 8 1 35 10 9 3 3 1 28 7 7 4 18 16 12 5 1 2 1

FingerprintsVector;TopologicalPharmacophoreAtomPairs:FixedSize:MinDist
ance1:MaxDistance10;150;OrderedNumericalValues;ValuesString;18 0 0 1 0
0 0 2 0 0 1 0 0 0 0 22 12 8 0 0 1 2 0 0 0 0 0 0 0 0 18 6 3 1 0 0 0 1
0 0 1 0 0 0 0 22 13 6 0 0 5 7 0 0 2 0 0 0 0 0 28 9 5 1 0 0 0 1 0 0 1 0
0 0 0 36 16 10 0 0 3 4 0 0 1 0 0 0 0 0 37 10 8 0 0 0 0 1 0 0 0 0 0 0
0 35 10 9 0 0 3 3 0 0 1 0 0 0 0 0 28 7 7 4 0 0 0 0 0 0 0 0 0 0 0 18...

FingerprintsVector;TopologicalPharmacophoreAtomTriplets:ArbitrarySize:
MinDistance1:MaxDistance10;696;NumericalValues;IDsAndValuesString;Ar1-
Ar1-Ar1 Ar1-Ar1-H1 Ar1-Ar1-HBA1 Ar1-Ar1-HBD1 Ar1-H1-H1 Ar1-H1-HBA1 Ar1
-H1-HBD1 Ar1-HBA1-HBD1 H1-H1-H1 H1-H1-HBA1 H1-H1-HBD1 H1-HBA1-HBA1 H1-
HBA1-HBD1 H1-HBA1-NI1 H1-HBD1-NI1 HBA1-HBA1-NI1 HBA1-HBD1-NI1 Ar1-...;
46 106 8 3 83 11 4 1 21 5 3 1 2 2 1 1 1 100 101 18 11 145 132 26 14 23
28 3 3 5 4 61 45 10 4 16 20 7 5 1 3 4 5 3 1 1 1 1 5 4 2 1 2 2 2 1 1 1
119 123 24 15 185 202 41 25 22 17 3 5 85 95 18 11 23 17 3 1 1 6 4 ...

FingerprintsVector;TopologicalPharmacophoreAtomTriplets:FixedSize:MinD
istance1:MaxDistance10;2692;OrderedNumericalValues;ValuesString;46 106
8 3 0 0 83 11 4 0 0 0 1 0 0 0 0 0 0 0 0 21 5 3 0 0 1 2 2 0 0 1 0 0 0
0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100 101 18 11 0 0 145 132 26
14 0 0 23 28 3 3 0 0 5 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 61 45 10 4 0
0 16 20 7 5 1 0 3 4 5 3 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 5 ...
```

OPTIONS

--alpha *number*

Value of alpha parameter for calculating *Tversky* similarity coefficient specified for -b, --BitVectorComparisonMode option. It corresponds to weights assigned for bits set to "1" in a pair of fingerprint bit-vectors during the calculation of similarity coefficient. Possible values: *0 to 1*. Default value: <0.5>.

--beta *number*

Value of beta parameter for calculating *WeightedTanimoto* and *WeightedTversky* similarity coefficients specified for -b, --BitVectorComparisonMode option. It is used to weight the contributions of bits set to "0" during the calculation of similarity coefficients. Possible values: *0 to 1*. Default value of <1> makes *WeightedTanimoto* and *WeightedTversky* equivalent to *Tanimoto* and *Tversky*.

-b, --BitVectorComparisonMode *All* | *"TanimotoSimilarity,[TverskySimilarity,...]"*

Specify what similarity coefficients to use for calculating similarity matrices for fingerprints bit-vector strings data values in *TextFile(s)*: calculate similarity matrices for all supported similarity coefficients or specify a comma delimited list of similarity coefficients. Possible values: *All* | *"TanimotoSimilarity,[TverskySimilarity,...]*. Default: *TanimotoSimilarity*

All uses complete list of supported similarity coefficients: *BaroniUrbaniSimilarity, BuserSimilarity,*

CosineSimilarity, DiceSimilarity, DennisSimilarity, ForbesSimilarity, FossumSimilarity, HamannSimilarity, JacardSimilarity, Kulczynski1Similarity, Kulczynski2Similarity, MatchingSimilarity, McConnaugheySimilarity, OchiaiSimilarity, PearsonSimilarity, RogersTanimotoSimilarity, RussellRaoSimilarity, SimpsonSimilarity, SkoalSneath1Similarity, SkoalSneath2Similarity, SkoalSneath3Similarity, TanimotoSimilarity, TverskySimilarity, YuleSimilarity, WeightedTanimotoSimilarity, WeightedTverskySimilarity. These similarity coefficients are described below.

For two fingerprint bit-vectors A and B of same size, let:

```
Na = Number of bits set to "1" in A
Nb = Number of bits set to "1" in B
Nc = Number of bits set to "1" in both A and B
Nd = Number of bits set to "0" in both A and B

Nt = Number of bits set to "1" or "0" in A or B (Size of A or B)
Nt = Na + Nb - Nc + Nd

Na - Nc = Number of bits set to "1" in A but not in B
Nb - Nc = Number of bits set to "1" in B but not in A
```

Then, various similarity coefficients [Ref. 40 - 42] for a pair of bit-vectors A and B are defined as follows:

BaroniUrbaniSimilarity: (SQRT(Nc * Nd) + Nc) / (SQRT (Nc * Nd) + Nc + (Na - Nc) + (Nb - Nc)) (same as Buser)

BuserSimilarity: (SQRT (Nc * Nd) + Nc) / (SQRT (Nc * Nd) + Nc + (Na - Nc) + (Nb - Nc)) (same as BaroniUrbani)

CosineSimilarity: Nc / SQRT (Na * Nb) (same as Ochiai)

DiceSimilarity: (2 * Nc) / (Na + Nb)

DennisSimilarity: (Nc * Nd - ((Na - Nc) * (Nb - Nc))) / SQRT (Nt * Na * Nb)

ForbesSimilarity: (Nt * Nc) / (Na * Nb)

FossumSimilarity: (Nt * ((Nc - 1/2) ** 2) / (Na * Nb)

HamannSimilarity: ((Nc + Nd) - (Na - Nc) - (Nb - Nc)) / Nt

JaccardSimilarity: Nc / ((Na - Nc) + (Nb - Nc) + Nc) = Nc / (Na + Nb - Nc) (same as Tanimoto)

Kulczynski1Similarity: Nc / ((Na - Nc) + (Nb - Nc)) = Nc / (Na + Nb - 2Nc)

Kulczynski2Similarity: ((Nc / 2) * (2 * Nc + (Na - Nc) + (Nb - Nc))) / ((Nc + (Na - Nc)) * (Nc + (Nb - Nc))) = 0.5 * (Nc / Na + Nc / Nb)

MatchingSimilarity: (Nc + Nd) / Nt

McConnaugheySimilarity: (Nc ** 2 - (Na - Nc) * (Nb - Nc)) / (Na * Nb)

OchiaiSimilarity: Nc / SQRT (Na * Nb) (same as Cosine)

PearsonSimilarity: ((Nc * Nd) - ((Na - Nc) * (Nb - Nc)) / SQRT (Na * Nb * (Na - Nc + Nd) * (Nb - Nc + Nd))

RogersTanimotoSimilarity: (Nc + Nd) / ((Na - Nc) + (Nb - Nc) + Nt) = (Nc + Nd) / (Na + Nb - 2Nc + Nt)

RussellRaoSimilarity: Nc / Nt

SimpsonSimilarity: Nc / MIN (Na, Nb)

SkoalSneath1Similarity: Nc / (Nc + 2 * (Na - Nc) + 2 * (Nb - Nc)) = Nc / (2 * Na + 2 * Nb - 3 * Nc)

SkoalSneath2Similarity: (2 * Nc + 2 * Nd) / (Nc + Nd + Nt)

SkoalSneath3Similarity: (Nc + Nd) / ((Na - Nc) + (Nb - Nc)) = (Nc + Nd) / (Na + Nb - 2 * Nc)

TanimotoSimilarity: Nc / ((Na - Nc) + (Nb - Nc) + Nc) = Nc / (Na + Nb - Nc) (same as Jaccard)

TverskySimilarity: Nc / (alpha * (Na - Nc) + (1 - alpha) * (Nb - Nc) + Nc) = Nc / (alpha * (Na - Nb) + Nb)

YuleSimilarity: ((Nc * Nd) - ((Na - Nc) * (Nb - Nc))) / ((Nc * Nd) + ((Na - Nc) * (Nb - Nc)))

Values of Tanimoto/Jaccard and Tversky coefficients are dependent on only those bit which are set to "1" in both A and B. In order to take into account all bit positions, modified versions of Tanimoto [Ref. 42] and Tversky [Ref. 43] have been developed.

Let:

```
Na' = Number of bits set to "0" in A
Nb' = Number of bits set to "0" in B
```

```
Nc' = Number of bits set to "0" in both A and B
```

Tanimoto': Nc' / ((Na' - Nc') + (Nb' - Nc') + Nc') = Nc' / (Na' + Nb' - Nc')

Tversky': Nc' / (alpha * (Na' - Nc') + (1 - alpha) * (Nb' - Nc') + Nc') = Nc' / (alpha * (Na' - Nb') + Nb')

Then:

WeightedTanimotoSimilarity = beta * Tanimoto + (1 - beta) * Tanimoto'

WeightedTverskySimilarity = beta * Tversky + (1 - beta) * Tversky'

-c, --ColMode *ColNum* | *ColLabel*

Specify how columns are identified in *TextFile(s)*: using column number or column label. Possible values: *ColNum or ColLabel*. Default value: *ColNum*.

--CompoundIDCol *col number* | *col name*

This value is -c, --ColMode mode specific. It specifies input *TextFile(s)* column to use for generating compound ID for similarity matrices in output *TextFile(s)*. Possible values: *col number or col label*. Default value: *first column containing the word compoundID in its column label or sequentially generated IDs*.

--CompoundIDPrefix *text*

Specify compound ID prefix to use during sequential generation of compound IDs for input *SDFile(s)* and *TextFile(s)*. Default value: *Cmpd*. The default value generates compound IDs which look like Cmpd<Number>.

For input *SDFile(s)*, this value is only used during *LabelPrefix* | *MolNameOrLabelPrefix* values of --CompoundIDMode option; otherwise, it's ignored.

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

```
Compound
```

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

--CompoundIDField *DataFieldName*

Specify input *SDFile(s)* datafield label for generating compound IDs. This value is only used during *DataField* value of --CompoundIDMode option.

Examples for *DataField* value of --CompoundIDMode:

```
MolID
ExtReg
```

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

Specify how to generate compound IDs from input *SDFile(s)* for similarity matrix CSV/TSV text file(s): use a *SDFile(s)* datafield value; use molname line from *SDFile(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 *SDFile(s)* takes precedence over sequential compound IDs generated using *LabelPrefix* and only empty molname values are replaced with sequential compound IDs.

-d, --detail *InfoLevel*

Level of information to print about lines being ignored. Default: *1*. Possible values: *1, 2 or 3*.

-f, --fast

In this mode, fingerprints columns specified using --FingerprintsCol for *TextFile(s)* and --FingerprintsField for *SDFile(s)* are assumed to contain valid fingerprints data and no checking is performed before calculating similarity matrices. By default, fingerprints data is validated before computing pairwise similarity and distance coefficients.

--FingerprintsCol *col number* | *col name*

This value is -c, --colmode specific. It specifies fingerprints column to use during calculation similarity matrices for *TextFile(s)*. Possible values: *col number or col label*. Default value: *first column containing the word Fingerprints in its column label*.

--FingerprintsField *FieldLabel*

Fingerprints field label to use during calculation similarity matrices for *SDFile(s)*. Default value: *first data field label containing the word Fingerprints in its label*

-h, --help

Print this help message.

--InDelim *comma* | *semicolon*

Input delimiter for CSV *TextFile(s)*. Possible values: *comma or semicolon*. Default value: *comma*. For TSV files, this option is ignored and *tab* is used as a delimiter.

--InputDataMode *LoadInMemory* | *ScanFile*

Specify how fingerprints bit-vector or vector strings data from *SD, FP and CSV/TSV* fingerprint file(s) is processed: Retrieve, process and load all available fingerprints data in memory; Retrieve and process data for fingerprints one at a time. Possible values : *LoadInMemory* | *ScanFile*. Default: *LoadInMemory*.

During *LoadInMemory* value of --InputDataMode, fingerprints bit-vector or vector strings data from input file is retrieved, processed, and loaded into memory all at once as fingerprints objects for generation for similarity matrices.

During *ScanFile* value of --InputDataMode, multiple passes over the input fingerprints file are performed to retrieve and process fingerprints bit-vector or vector strings data one at a time to generate fingerprints objects used during generation of similarity matrices. A temporary copy of the input fingerprints file is made at the start and deleted after generating the matrices.

ScanFile value of --InputDataMode allows processing of arbitrary large fingerprints files without any additional memory requirement.

-m, --mode *AutoDetect* | *FingerprintsBitVectorString* | *FingerprintsVectorString*

Format of fingerprint strings data in *TextFile(s)*: automatically detect format of fingerprints string created by MayaChemTools fingerprints generation scripts or explicitly specify its format. Possible values: *AutoDetect* | *FingerprintsBitVectorString* | *FingerprintsVectorString*. Default value: *AutoDetect*.

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

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

--OutMatrixFormat *RowsAndColumns* | *IDPairsAndValue*

Specify how similarity or distance values calculated for fingerprints vector and bit-vector strings are written to the output CSV/TSV text file(s): Generate text files containing rows and columns with their labels corresponding to compound IDs and each matrix element value corresponding to similarity or distance between corresponding compounds; Generate text files containing rows containing compoundIDs for two compounds followed by similarity or distance value between these compounds.

Possible values: *RowsAndColumns, or IDPairsAndValue*. Default value: *RowsAndColumns*.

The value of --OutMatrixFormat in conjunction with --OutMatrixType determines type of data written to output files and allows generation of up to 6 different output data formats:

```
OutMatrixFormat OutMatrixType

RowsAndColumns  FullMatrix   [ DEFAULT ]
RowsAndColumns  UpperTriangularMatrix
RowsAndColumns  LowerTriangularMatrix

IDPairsAndValue FullMatrix
IDPairsAndValue UpperTriangularMatrix
IDPairsAndValue LowerTriangularMatrix
```

Example of data in output file for *RowsAndColumns* --OutMatrixFormat value for *FullMatrix* valueof --OutMatrixType:

```
"","Cmpd1","Cmpd2","Cmpd3","Cmpd4","Cmpd5","Cmpd6",... ...
"Cmpd1","1","0.04","0.25","0.13","0.11","0.2",... ...
"Cmpd2","0.04","1","0.06","0.05","0.19","0.07",... ...
"Cmpd3","0.25","0.06","1","0.12","0.22","0.25",... ...
"Cmpd4","0.13","0.05","0.12","1","0.11","0.13",... ...
"Cmpd5","0.11","0.19","0.22","0.11","1","0.17",... ...
"Cmpd6","0.2","0.07","0.25","0.13","0.17","1",... ...
... ... ..
... ... ..
... ... ..
```

Example of data in output file for *RowsAndColumns* --OutMatrixFormat value for *UpperTriangularMatrix*

value of --OutMatrixType:

```
"","Cmpd1","Cmpd2","Cmpd3","Cmpd4","Cmpd5","Cmpd6",... ...
"Cmpd1","1","0.04","0.25","0.13","0.11","0.2",... ...
"Cmpd2","1","0.06","0.05","0.19","0.07",... ...
"Cmpd3","1","0.12","0.22","0.25",... ...
"Cmpd4","1","0.11","0.13",... ...
"Cmpd5","1","0.17",... ...
"Cmpd6","1",... ...
... ... ..
... ... ..
... ... ..
```

Example of data in output file for *RowsAndColumns* --OutMatrixFormat value for *LowerTriangularMatrix* value of --OutMatrixType:

```
"","Cmpd1","Cmpd2","Cmpd3","Cmpd4","Cmpd5","Cmpd6",... ...
"Cmpd1","1"
"Cmpd2","0.04","1"
"Cmpd3","0.25","0.06","1"
"Cmpd4","0.13","0.05","0.12","1"
"Cmpd5","0.11","0.19","0.22","0.11","1"
"Cmpd6","0.2","0.07","0.25","0.13","0.17","1"
... ... ..
... ... ..
... ... ..
```

Example of data in output file for *IDPairsAndValue* --OutMatrixFormat value for <FullMatrix> value of OutMatrixType:

```
"CmpdID1","CmpdID2","Coefficient Value"
"Cmpd1","Cmpd1","1"
"Cmpd1","Cmpd2","0.04"
"Cmpd1","Cmpd3","0.25"
"Cmpd1","Cmpd4","0.13"
... ... ...
... ... ...
... ... ...
"Cmpd2","Cmpd1","0.04"
"Cmpd2","Cmpd2","1"
"Cmpd2","Cmpd3","0.06"
"Cmpd2","Cmpd4","0.05"
... ... ...
... ... ...
... ... ...
"Cmpd3","Cmpd1","0.25"
"Cmpd3","Cmpd2","0.06"
"Cmpd3","Cmpd3","1"
"Cmpd3","Cmpd4","0.12"
... ... ...
... ... ...
... ... ...
```

Example of data in output file for *IDPairsAndValue* --OutMatrixFormat value for <UpperTriangularMatrix> value of --OutMatrixType:

```
"CmpdID1","CmpdID2","Coefficient Value"
"Cmpd1","Cmpd1","1"
"Cmpd1","Cmpd2","0.04"
"Cmpd1","Cmpd3","0.25"
"Cmpd1","Cmpd4","0.13"
... ... ...
... ... ...
... ... ...
"Cmpd2","Cmpd2","1"
"Cmpd2","Cmpd3","0.06"
"Cmpd2","Cmpd4","0.05"
... ... ...
... ... ...
... ... ...
"Cmpd3","Cmpd3","1"
```

```
"Cmpd3","Cmpd4","0.12"
... ... ...
... ... ...
... ... ...
```

Example of data in output file for *IDPairsAndValue* --OutMatrixFormat value for <LowerTriangularMatrix> value of --OutMatrixType:

```
"CmpdID1","CmpdID2","Coefficient Value"
"Cmpd1","Cmpd1","1"
"Cmpd2","Cmpd1","0.04"
"Cmpd2","Cmpd2","1"
"Cmpd3","Cmpd1","0.25"
"Cmpd3","Cmpd2","0.06"
"Cmpd3","Cmpd3","1"
"Cmpd4","Cmpd1","0.13"
"Cmpd4","Cmpd2","0.05"
"Cmpd4","Cmpd3","0.12"
"Cmpd4","Cmpd4","1"
... ... ...
... ... ...
... ... ...
```

--OutMatrixType *FullMatrix* | *UpperTriangularMatrix* | *LowerTriangularMatrix*

Type of similarity or distance matrix to calculate for fingerprints vector and bit-vector strings: Calculate full matrix; Calculate lower triangular matrix including diagonal; Calculate upper triangular matrix including diagonal.

Possible values: *FullMatrix, UpperTriangularMatrix, or LowerTriangularMatrix*. Default value: *FullMatrix*.

The value of --OutMatrixType in conjunction with --OutMatrixFormat determines type of data written to output files.

-o, --overwrite

Overwrite existing files

-p, --precision *number*

Precision of calculated values in the output file. Default: up to *2* decimal places. Valid values: positive integers.

-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><BitVectorComparisonMode>.<Ext> or <Root><VectorComparisonMode><VectorComparisonFormulism>.<Ext>. The csv, and tsv <Ext> values are used for comma/semicolon, and tab delimited text files respectively. This option is ignored for multiple input files.

-v, --VectorComparisonMode *All* | *"TanimotoSimilarity,[ManhattanDistance,...]"*

Specify what similarity or distance coefficients to use for calculating similarity matrices for fingerprint vector strings data values in *TextFile(s)*: calculate similarity matrices for all supported similarity and distance coefficients or specify a comma delimited list of similarity and distance coefficients. Possible values: *All* | *"TanimotoSimilairy,[ManhattanDistance,..]"*. Default: *TanimotoSimilarity*.

The value of -v, --VectorComparisonMode, in conjunction with --VectorComparisonFormulism, decides which type of similarity and distance coefficient formulism gets used.

All uses complete list of supported similarity and distance coefficients: *CosineSimilarity, CzekanowskiSimilarity, DiceSimilarity, OchiaiSimilarity, JaccardSimilarity, SorensonSimilarity, TanimotoSimilarity, CityBlockDistance, EuclideanDistance, HammingDistance, ManhattanDistance, SoergelDistance*. These similarity and distance coefficients are described below.

FingerprintsVector.pm module, used to calculate similarity and distance coefficients, provides support to perform comparison between vectors containing three different types of values:

Type I: OrderedNumericalValues

```
. Size of two vectors are same
. Vectors contain real values in a specific order. For example: MACCS keys
  count, Topological pharmnacophore atom pairs and so on.
```

Type II: UnorderedNumericalValues

```
. Size of two vectors might not be same
. Vectors contain unordered real value identified by value IDs. For example:
  Toplogical atom pairs, Topological atom torsions and so on
```

Type III: AlphaNumericalValues

```
. Size of two vectors might not be same
. Vectors contain unordered alphanumerical values. For example: Extended
  connectivity fingerprints, atom neighborhood fingerprints.
```

Before performing similarity or distance calculations between vectors containing UnorderedNumericalValues or AlphaNumericalValues, the vectors are transformed into vectors containing unique OrderedNumericalValues using value IDs for UnorderedNumericalValues and values itself for AlphaNumericalValues.

Three forms of similarity and distance calculation between two vectors, specified using `--VectorComparisonFormulism` option, are supported: *AlgebraicForm, BinaryForm or SetTheoreticForm*.

For *BinaryForm*, the ordered list of processed final vector values containing the value or count of each unique value type is simply converted into a binary vector containing 1s and 0s corresponding to presence or absence of values before calculating similarity or distance between two vectors.

For two fingerprint vectors A and B of same size containing OrderedNumericalValues, let:

```
N = Number values in A or B

Xa = Values of vector A
Xb = Values of vector B

Xai = Value of ith element in A
Xbi = Value of ith element in B

SUM = Sum of i over N values
```

For SetTheoreticForm of calculation between two vectors, let:

```
SetIntersectionXaXb = SUM ( MIN ( Xai, Xbi ) )
SetDifferenceXaXb = SUM ( Xai ) + SUM ( Xbi ) - SUM ( MIN ( Xai, Xbi ) )
```

For BinaryForm of calculation between two vectors, let:

```
Na = Number of bits set to "1" in A = SUM ( Xai )
Nb = Number of bits set to "1" in B = SUM ( Xbi )
Nc = Number of bits set to "1" in both A and B = SUM ( Xai * Xbi )
Nd = Number of bits set to "0" in both A and B
   = SUM ( 1 - Xai - Xbi + Xai * Xbi)

N = Number of bits set to "1" or "0" in A or B = Size of A or B = Na + Nb - Nc + Nd
```

Additionally, for BinaryForm various values also correspond to:

```
Na = | Xa |
Nb = | Xb |
Nc = | SetIntersectionXaXb |
Nd = N - | SetDifferenceXaXb |

| SetDifferenceXaXb | = N - Nd = Na + Nb - Nc + Nd - Nd = Na + Nb - Nc
                      =  | Xa | + | Xb | - | SetIntersectionXaXb |
```

Various similarity and distance coefficients [Ref 40, Ref 62, Ref 64] for a pair of vectors A and B in *AlgebraicForm, BinaryForm and SetTheoreticForm* are defined as follows:

`CityBlockDistance`: (same as HammingDistance and ManhattanDistance)

AlgebraicForm: SUM (ABS (Xai - Xbi))

BinaryForm: (Na - Nc) + (Nb - Nc) = Na + Nb - 2 * Nc

SetTheoreticForm: | SetDifferenceXaXb | - | SetIntersectionXaXb | = SUM (Xai) + SUM (Xbi) - 2 * (SUM (MIN (Xai, Xbi)))

`CosineSimilarity`: (same as OchiaiSimilarityCoefficient)

AlgebraicForm: SUM (Xai * Xbi) / SQRT (SUM (Xai ** 2) * SUM (Xbi ** 2))

BinaryForm: Nc / SQRT (Na * Nb)

SetTheoreticForm: | SetIntersectionXaXb | / SQRT (|Xa| * |Xb|) = SUM (MIN (Xai, Xbi)) / SQRT (SUM (Xai) * SUM (Xbi))

CzekanowskiSimilarity: (same as DiceSimilarity and SorensonSimilarity)

AlgebraicForm: (2 * (SUM (Xai * Xbi))) / (SUM (Xai ** 2) + SUM (Xbi **2))

BinaryForm: 2 * Nc / (Na + Nb)

SetTheoreticForm: 2 * | SetIntersectionXaXb | / (|Xa| + |Xb|) = 2 * (SUM (MIN (Xai, Xbi))) / (SUM (Xai) + SUM (Xbi))

DiceSimilarity: (same as CzekanowskiSimilarity and SorensonSimilarity)

AlgebraicForm: (2 * (SUM (Xai * Xbi))) / (SUM (Xai ** 2) + SUM (Xbi **2))

BinaryForm: 2 * Nc / (Na + Nb)

SetTheoreticForm: 2 * | SetIntersectionXaXb | / (|Xa| + |Xb|) = 2 * (SUM (MIN (Xai, Xbi))) / (SUM (Xai) + SUM (Xbi))

EuclideanDistance:

AlgebraicForm: SQRT (SUM (((Xai - Xbi) ** 2)))

BinaryForm: SQRT ((Na - Nc) + (Nb - Nc)) = SQRT (Na + Nb - 2 * Nc)

SetTheoreticForm: SQRT (| SetDifferenceXaXb | - | SetIntersectionXaXb |) = SQRT (SUM (Xai) + SUM (Xbi) - 2 * (SUM (MIN (Xai, Xbi))))

HammingDistance: (same as CityBlockDistance and ManhattanDistance)

AlgebraicForm: SUM (ABS (Xai - Xbi))

BinaryForm: (Na - Nc) + (Nb - Nc) = Na + Nb - 2 * Nc

SetTheoreticForm: | SetDifferenceXaXb | - | SetIntersectionXaXb | = SUM (Xai) + SUM (Xbi) - 2 * (SUM (MIN (Xai, Xbi)))

JaccardSimilarity: (same as TanimotoSimilarity)

AlgebraicForm: SUM (Xai * Xbi) / (SUM (Xai ** 2) + SUM (Xbi ** 2) - SUM (Xai * Xbi))

BinaryForm: Nc / ((Na - Nc) + (Nb - Nc) + Nc) = Nc / (Na + Nb - Nc)

SetTheoreticForm: | SetIntersectionXaXb | / | SetDifferenceXaXb | = SUM (MIN (Xai, Xbi)) / (SUM (Xai) + SUM (Xbi) - SUM (MIN (Xai, Xbi)))

ManhattanDistance: (same as CityBlockDistance and HammingDistance)

AlgebraicForm: SUM (ABS (Xai - Xbi))

BinaryForm: (Na - Nc) + (Nb - Nc) = Na + Nb - 2 * Nc

SetTheoreticForm: | SetDifferenceXaXb | - | SetIntersectionXaXb | = SUM (Xai) + SUM (Xbi) - 2 * (SUM (MIN (Xai, Xbi)))

OchiaiSimilarity: (same as CosineSimilarity)

AlgebraicForm: SUM (Xai * Xbi) / SQRT (SUM (Xai ** 2) * SUM (Xbi ** 2))

BinaryForm: Nc / SQRT (Na * Nb)

SetTheoreticForm: | SetIntersectionXaXb | / SQRT (|Xa| * |Xb|) = SUM (MIN (Xai, Xbi)) / SQRT (SUM (Xai) * SUM (Xbi))

SorensonSimilarity: (same as CzekanowskiSimilarity and DiceSimilarity)

AlgebraicForm: (2 * (SUM (Xai * Xbi))) / (SUM (Xai ** 2) + SUM (Xbi **2))

BinaryForm: 2 * Nc / (Na + Nb)

SetTheoreticForm: 2 * | SetIntersectionXaXb | / (|Xa| + |Xb|) = 2 * (SUM (MIN (Xai, Xbi))) / (SUM (Xai) + SUM (Xbi))

SoergelDistance:

AlgebraicForm: SUM (ABS (Xai - Xbi)) / SUM (MAX (Xai, Xbi))

BinaryForm: 1 - Nc / (Na + Nb - Nc) = (Na + Nb - 2 * Nc) / (Na + Nb - Nc)

SetTheoreticForm: (| SetDifferenceXaXb | - | SetIntersectionXaXb |) / | SetDifferenceXaXb | = (SUM (Xai) + SUM (Xbi) - 2 * (SUM (MIN (Xai, Xbi)))) / (SUM (Xai) + SUM (Xbi) - SUM (MIN (Xai, Xbi)))

TanimotoSimilarity: (same as JaccardSimilarity)

AlgebraicForm: SUM (Xai * Xbi) / (SUM (Xai ** 2) + SUM (Xbi ** 2) - SUM (Xai * Xbi))

BinaryForm: Nc / ((Na - Nc) + (Nb - Nc) + Nc) = Nc / (Na + Nb - Nc)

SetTheoreticForm: | SetIntersectionXaXb | / | SetDifferenceXaXb | = SUM (MIN (Xai, Xbi)) / (SUM (Xai) + SUM (Xbi) - SUM (MIN (Xai, Xbi)))

--VectorComparisonFormulism *All* | *"AlgebraicForm,[BinaryForm,SetTheoreticForm]"*

Specify fingerprints vector comparison formulism to use for calculation similarity and distance coefficients during -v, --VectorComparisonMode: use all supported comparison formulisms or specify a comma delimited. Possible values: *All* | *"AlgebraicForm,[BinaryForm,SetTheoreticForm]"*. Default value: *AlgebraicForm*.

All uses all three forms of supported vector comparison formulism for values of -v, --VectorComparisonMode option.

For fingerprint vector strings containing AlphaNumericalValues data values - ExtendedConnectivityFingerprints, AtomNeighborhoodsFingerprints and so on - all three formulism result in same value during similarity and distance calculations.

-w, --WorkingDir *DirName*

Location of working directory. Default: current directory.

EXAMPLES

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in text file present in a column name containing Fingerprint substring by loading all fingerprints data into memory and create a SampleFPHexTanimotoSimilarity.csv file containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -o SampleFPHex.csv
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in SD File present in a data field with Fingerprint substring in its label by loading all fingerprints data into memory and create a SampleFPHexTanimotoSimilarity.csv file containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl -o SampleFPHex.sdf
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in FP file by loading all fingerprints data into memory and create a SampleFPHexTanimotoSimilarity.csv file along with compound IDs retrieved from FP file, type:

```
% SimilarityMatricesFingerprints.pl -o SampleFPHex.fpf
```

To generate a lower triangular similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in text file present in a column name containing Fingerprint substring by loading all fingerprints data into memory and create a SampleFPHexTanimotoSimilarity.csv file containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -o --InputDataMode LoadInMemory
  --OutMatrixFormat RowsAndColumns --OutMatrixType LowerTriangularMatrix
  SampleFPHex.csv
```

To generate a upper triangular similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in text file present in a column name containing Fingerprint substring by loading all fingerprints data into memory and create a SampleFPHexTanimotoSimilarity.csv file in IDPairsAndValue format containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -o --InputDataMode LoadInMemory
  --OutMatrixFormat IDPairsAndValue --OutMatrixType UpperTriangularMatrix
  SampleFPHex.csv
```

To generate a full similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in text file present in a column name containing Fingerprint substring by scanning file without loading all fingerprints data into memory and create a SampleFPHexTanimotoSimilarity.csv file containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -o --InputDataMode ScanFile
  --OutMatrixFormat RowsAndColumns --OutMatrixType FullMatrix
  SampleFPHex.csv
```

To generate a lower triangular similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in text file present in a column name containing Fingerprint substring by scanning file without loading all fingerprints data into memory and create a SampleFPHexTanimotoSimilarity.csv file in IDPairsAndValue format containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -o --InputDataMode ScanFile
  --OutMatrixFormat IDPairsAndValue --OutMatrixType LowerTriangularMatrix
  SampleFPHex.csv
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient using algebraic formulism for fingerprints vector strings data corresponding to supported fingerprints in text file present in a column name containing Fingerprint substring and create a SampleFPCountTanimotoSimilarityAlgebraicForm.csv file containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -o SampleFPCount.csv
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient using algebraic formulism for fingerprints vector strings data corresponding to supported fingerprints in SD file present in a data field with Fingerprint substring in its label and create a SampleFPCountTanimotoSimilarityAlgebraicForm.csv file containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl -o SampleFPCount.sdf
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient using algebraic formulism vector strings data corresponding to supported fingerprints in FP file and create a SampleFPCountTanimotoSimilarityAlgebraicForm.csv file along with compound IDs retrieved from FP file, type:

```
% SimilarityMatricesFingerprints.pl -o SampleFPCount.fpf
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in text file present in a column name containing Fingerprint substring and create a SampleFPHexTanimotoSimilarity.csv file in IDPairsAndValue format containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl --OutMatrixFormat IDPairsAndValue -o
  SampleFPHex.csv
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in SD file present in a data field with Fingerprint substring in its label and create a SampleFPHexTanimotoSimilarity.csv file in IDPairsAndValue format containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl --OutMatrixFormat IDPairsAndValue -o
  SampleFPHex.sdf
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in FP file and create a SampleFPHexTanimotoSimilarity.csv file in IDPairsAndValue format along with compound IDs retrieved from FP file, type:

```
% SimilarityMatricesFingerprints.pl --OutMatrixFormat IDPairsAndValue -o
  SampleFPHex.fpf
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints in SD file present in a data field with Fingerprint substring in its label and create a SampleFPHexTanimotoSimilarity.csv file containing compound IDs from mol name line, type:

```
% SimilarityMatricesFingerprints.pl --CompoundIDMode MolName -o
  SampleFPHex.sdf
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create a SampleFPHexTanimotoSimilarity.csv file containing compound IDs from data field name Mol_ID, type:

```
% SimilarityMatricesFingerprints.pl --CompoundIDMode DataField
```

```
--CompoundIDField Mol_ID -o SampleFPBin.sdf
```

To generate similarity matrices corresponding to Buser, Dice and Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a column name containing Fingerprint substring and create SampleFPBin[CoefficientName]Similarity.csv files containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -b "BuserSimilarity,DiceSimilarity,
  TanimotoSimilarity" -o SampleFPBin.csv
```

To generate similarity matrices corresponding to Buser, Dice and Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create SampleFPBin[CoefficientName]Similarity.csv files containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl -b "BuserSimilarity,DiceSimilarity,
  TanimotoSimilarity" -o SampleFPBin.sdf
```

To generate similarity matrices corresponding to CityBlock distance and Tanimoto similarity coefficients using algebraic formulism for fingerprints vector strings data corresponding to supported fingerprints present in a column name containing Fingerprint substring and create SampleFPCount[CoefficientName]AlgebraicForm.csv files containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -v "CityBlockDistance,
  TanimotoSimilarity" -o SampleFPCount.csv
```

To generate similarity matrices corresponding to CityBlock distance and Tanimoto similarity coefficients using algebraic formulism for fingerprints vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create SampleFPCount[CoefficientName]AlgebraicForm.csv files containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl -v "CityBlockDistance,
  TanimotoSimilarity" -o SampleFPCount.sdf
```

To generate similarity matrices corresponding to CityBlock distance Tanimoto similarity coefficients using binary formulism for fingerprints vector strings data corresponding to supported fingerprints present in a column name containing Fingerprint substring and create SampleFPCount[CoefficientName]Binary.csv files containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -v "CityBlockDistance,
  TanimotoSimilarity" --VectorComparisonFormulism BinaryForm -o
  SampleFPCount.csv
```

To generate similarity matrices corresponding to CityBlock distance Tanimoto similarity coefficients using binary formulism for fingerprints vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create SampleFPCount[CoefficientName]Binary.csv files containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl -v "CityBlockDistance,
  TanimotoSimilarity" --VectorComparisonFormulism BinaryForm -o
  SampleFPCount.sdf
```

To generate similarity matrices corresponding to CityBlock distance Tanimoto similarity coefficients using all supported comparison formulisms for fingerprints vector strings data corresponding to supported fingerprints present in a column name containing Fingerprint substring and create SampleFPCount[CoefficientName][FormulismName].csv files containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -v "CityBlockDistance,
  TanimotoSimilarity" --VectorComparisonFormulism All -o SampleFPCount.csv
```

To generate similarity matrices corresponding to CityBlock distance Tanimoto similarity coefficients using all supported comparison formulisms for fingerprints vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create SampleFPCount[CoefficientName][FormulismName].csv files containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl -v "CityBlockDistance,TanimotoSimilarity"
  --VectorComparisonFormulism All -o SampleFPCount.sdf
```

To generate similarity matrices corresponding to all available similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a column name containing Fingerprint substring and create SampleFPHex[CoefficientName].csv files containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -m AutoDetect --BitVectorComparisonMode
  All --alpha 0.5 -beta 0.5 -o SampleFPHex.csv
```

To generate similarity matrices corresponding to all available similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create SampleFPHex[CoefficientName].csv files containing sequentially generated compound IDs with Cmpd prefix, type

```
% SimilarityMatricesFingerprints.pl -m AutoDetect --BitVectorComparisonMode
  All --alpha 0.5 -beta 0.5 -o SampleFPHex.sdf
```

To generate similarity matrices corresponding to all available similarity and distance coefficients using all comparison formulism for fingerprints vector strings data corresponding to supported fingerprints present in a column name containing Fingerprint substring and create SampleFPCount[CoefficientName][FormulismName].csv files containing compound IDs retrieved from column name containing CompoundID substring, type:

```
% SimilarityMatricesFingerprints.pl -m AutoDetect --VectorComparisonMode
  All --VectorComparisonFormulism All -o SampleFPCount.csv
```

To generate similarity matrices corresponding to all available similarity and distance coefficients using all comparison formulism for fingerprints vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create SampleFPCount[CoefficientName][FormulismName].csv files containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl -m AutoDetect --VectorComparisonMode
  All --VectorComparisonFormulism All -o SampleFPCount.sdf
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a column number 2 and create a SampleFPHexTanimotoSimilarity.csv file containing compound IDs retrieved column number 1, type:

```
% SimilarityMatricesFingerprints.pl --ColMode ColNum --CompoundIDCol 1
  --FingerprintsCol 2 -o SampleFPHex.csv
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a data field name Fingerprints and create a SampleFPHexTanimotoSimilarity.csv file containing compound IDs present in data field name Mol_ID, type:

```
% SimilarityMatricesFingerprints.pl --FingerprintsField Fingerprints
  --CompoundIDMode DataField --CompoundIDField Mol_ID -o SampleFPHex.sdf
```

To generate a similarity matrix corresponding to Tversky similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a column named Fingerprints and create a SampleFPHexTverskySimilarity.tsv file containing compound IDs retrieved column named CompoundID, type:

```
% SimilarityMatricesFingerprints.pl --BitVectorComparisonMode
  TverskySimilarity --alpha 0.5 --ColMode ColLabel --CompoundIDCol
  CompoundID --FingerprintsCol Fingerprints --OutDelim Tab --quote No
  -o SampleFPHex.csv
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create a SampleFPHexTanimotoSimilarity.csv file containing compound IDs from molname line or sequentially generated compound IDs with Mol prefix, type:

```
% SimilarityMatricesFingerprints.pl --CompoundIDMode MolnameOrLabelPrefix
  --CompoundIDPrefix Mol -o SampleFPHex.sdf
```

To generate a similarity matrix corresponding to Tanimoto similarity coefficient for fingerprints bit-vector strings data corresponding to supported fingerprints present in a data field with Fingerprint substring in its label and create a SampleFPHexTanimotoSimilarity.tsv file containing sequentially generated compound IDs with Cmpd prefix, type:

```
% SimilarityMatricesFingerprints.pl -OutDelim Tab --quote No -o SampleFPHex.sdf
```

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

InfoFingerprintsFiles.pl, SimilaritySearchingFingerprints.pl, AtomNeighborhoodsFingerprints.pl, ExtendedConnectivityFingerprints.pl, MACCSKeysFingerprints.pl, PathLengthFingerprints.pl, TopologicalAtomPairsFingerprints.pl, TopologicalAtomTorsionsFingerprints.pl, TopologicalPharmacophoreAtomPairsFingerprints.pl, TopologicalPharmacophoreAtomTripletsFingerprints.pl

COPYRIGHT

Copyright (C) 2024 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.