

SYNOPSIS

```
MACCSKeysFingerprints.pl [--AromaticityModel AromaticityModelType] [--BitsOrder Ascending | Descending] [
-b, --BitStringFormat BinaryString | HexadecimalString] [--CompoundID DataFieldName or LabelPrefixString] [
--CompoundIDLabel text] [--CompoundIDMode DataField | MolName | LabelPrefix | MolNameOrLabelPrefix] [
--DataFields "FieldLabel1,FieldLabel2,..." ] [-d, --DataFieldsMode All | Common | Specify | CompoundID] [-f,
--Filter Yes | No] [--FingerprintsLabel text] [-h, --help] [-k, --KeepLargestComponent Yes | No] [-m,
--mode MACCSKeyBits | MACCSKeyCount] [--OutDelim comma | tab | semicolon] [--output SD | FP | text | all] [-o,
--overwrite] [-q, --quote Yes | No] [-r, --root RootName] [-s, --size number] [-v, --VectorStringFormat
IDsAndValuesString | IDsAndValuesPairsString | ValuesAndIDsString | ValuesAndIDsPairsString] [-w, --WorkingDir
DirName]
```

DESCRIPTION

Example of **SD** file containing MAACS keys fingerprints string data:

[illegible]

Example of *FP* file containing MAACS keys fingerprints string data:

```
#
# Package = MayaChemTools 7.4
# Release Date = Oct 21, 2010
#
```

Example of CSV *Text* file containing MAACS keys fingerprints string data:

The current release of MayaChemTools generates the following types of MACCS keys fingerprints bit-vector and vector strings:

[illegible]

```
--AromaticityModel MDLAromaticityModel | TriposAromaticityModel | MMFFAromaticityModel |  
ChemAxonBasicAromaticityModel | ChemAxonGeneralAromaticityModel | DaylightAromaticityModel |  
MavenChemToolsAromaticityModel
```

www.MayaChemTools.org

are: *MDLAromaticityModel*, *TriposAromaticityModel*, *MMFFAromaticityModel*, *ChemAxonBasicAromaticityModel*, *ChemAxonGeneralAromaticityModel*, *DaylightAromaticityModel* or *MayaChemToolsAromaticityModel*. Default value: *MayaChemToolsAromaticityModel*.

The supported aromaticity model names along with model specific control parameters are defined in `AromaticityModelsData.csv`, which is distributed with the current release and is available under `lib/data` directory. `Molecule.pm` module retrieves data from this file during class instantiation and makes it available to method `DetectAromaticity` for detecting aromaticity corresponding to a specific model.

--BitsOrder *Ascending / Descending*

Bits order to use during generation of fingerprints bit-vector string for *MACCSKeyBits* value of -m, --mode option. Possible values: *Ascending*, *Descending*. Default: *Ascending*.

Ascending bit order which corresponds to first bit in each byte as the lowest bit as opposed to the highest bit.

Internally, bits are stored in *Ascending* order using Perl vec function. Regardless of machine order, big-endian or little-endian, vec function always considers first string byte as the lowest byte and first bit within each byte as the lowest bit.

-b, --BitStringFormat *BinaryString* | *HexadecimalString*

Format of fingerprints bit-vector string data in output SD, FP or CSV/TSV text file(s) specified by --output used during *MACCSKeyBits* value of -m, --mode option. Possible values: *BinaryString*, *HexadecimalString*. Default value: *BinaryString*.

BinaryString corresponds to an ASCII string containing 1s and 0s. *HexadecimalString* contains bit values in ASCII hexadecimal format.

Examples:

```
FingerprintsBitVector;MACCSKeyBits;166;BinaryString;Ascending;00000000
00000000000000000000000000000000000000000000000000000000000000
01001000010010000000000000000000000000000000000000000000000000
01001010101111000110110001001101100000110111101001101111111111011111
11111111111110111000
```

```
FingerprintsBitVector;MACCSKeyBits;166;HexadecimalString;Ascending;000
0000000021210210e845f8d8c60b79dfbfbbfffd1
```

[illegible][illegible]

```
--CompoundID DataFieldName or LabelPrefixString
```

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

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

Examples for *DataField* value of --CompoundIDMode:

MolID
ExtReq

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>.

```
--CompoundI DLabel text
```

Specify compound ID column label for FP or 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 FP or CSV/TSV text file(s) along with generated fingerprints for *FP | text | all* 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 | all* 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 | all* 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: *MACCSKeyFingerprints*.

-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 *MACCSKeyBits | MACCSKeyCount*

Specify type of MACCS keys [Ref 45-47] fingerprints to generate for molecules in *SDFFile(s)*. Possible values: *MACCSKeyBits, MACCSKeyCount*. Default value: *MACCSKeyBits*.

For *MACCSKeyBits* value of -m, --mode option, a fingerprint bit-vector string containing zeros and ones is generated and for *MACCSKeyCount* value, a fingerprint vector string corresponding to number of MACCS keys is generated.

MACCSKeyBits | MACCSKeyCount values for -m, --mode option along with two possible *166 | 322* values of -s, --size supports generation of four different types of MACCS keys fingerprint: *MACCS166KeyBits, MACCS166KeyCount, MACCS322KeyBits, MACCS322KeyCount*.

Definition of MACCS keys uses the following atom and bond symbols to define atom and bond environments:

Atom symbols for 166 keys [Ref 47]:

```
A : Any valid periodic table element symbol  
Q : Hetro atoms; any non-C or non-H atom
```

X : Halogens; F, Cl, Br, I
Z : Others; other than H, C, N, O, Si, P, S, F, Cl, Br, I

Atom symbols for 322 keys [Ref 46]:

A : Any valid periodic table element symbol
Q : Hetro atoms; any non-C or non-H atom
X : Others; other than H, C, N, O, Si, P, S, F, Cl, Br, I
Z is neither defined nor used

Bond types:

- : Single
= : Double
T : Triple
: Triple
~ : Single or double query bond
% : An aromatic query bond

None : Any bond type; no explicit bond specified

\$: Ring bond; \$ before a bond type specifies ring bond
! : Chain or non-ring bond; ! before a bond type specifies chain bond

@ : A ring linkage and the number following it specifies the
atoms position in the line, thus @1 means linked back to the first
atom in the list.

Aromatic: Kekule or Arom5

Kekule: Bonds in 6-membered rings with alternate single/double bonds
or perimeter bonds

Arom5: Bonds in 5-membered rings with two double bonds and a hetro
atom at the apex of the ring.

MACCS 166 keys [Ref 45-47] are defined as follows:

Key Description

1 ISOTOPE
2 103 < ATOMIC NO. < 256
3 GROUP IVA,VA,VIA PERIODS 4-6 (Ge...)
4 ACTINIDE
5 GROUP IIIB,IVB (Sc...)
6 LANTHANIDE
7 GROUP VB,VIB,VIIB (V...)
8 QAAA@1
9 GROUP VIII (Fe...)
10 GROUP IIA (ALKALINE EARTH)
11 4M RING
12 GROUP IB,IIIB (Cu...)
13 ON(C)C
14 S-S
15 OC(O)O
16 QAA@1
17 CTC
18 GROUP IIIA (B...)
19 7M RING
20 SI
21 C=C(Q)Q
22 3M RING
23 NC(O)O
24 N-O
25 NC(N)N
26 C\$=C(\$A)\$A
27 I
28 QCH2Q
29 P
30 CQ(C)(C)A

31 QX
32 CSN
33 NS
34 CH2=A
35 GROUP IA (ALKALI METAL)
36 S HETEROCYCLE
37 NC(O)N
38 NC(C)N
39 OS(O)O
40 S-O
41 CTN
42 F
43 QHAQH
44 OTHER
45 C=CN
46 BR
47 SAN
48 OQ(O)O
49 CHARGE
50 C=C(C)C
51 CSO
52 NN
53 QHAAAQH
54 QHAAQH
55 OSO
56 ON(O)C
57 O HETEROCYCLE
58 QSQ
59 Snot%A%A
60 S=O
61 AS(A)A
62 A\$A!A\$A
63 N=O
64 A\$A!S
65 C%N
66 CC(C)(C)A
67 QS
68 QHQH (&...)
69 QQH
70 QNQ
71 NO
72 OAAO
73 S=A
74 CH3ACH3
75 A!N\$A
76 C=C(A)A
77 NAN
78 C=N
79 NAAN
80 NAAAN
81 SA(A)A
82 ACH2QH
83 QAAAA@1
84 NH2
85 CN(C)C
86 CH2QCH2
87 X!A\$A
88 S
89 OAAAO
90 QHAAACH2A
91 QHAAACH2A
92 OC(N)C
93 QCH3
94 QN
95 NAAO
96 5M RING
97 NAAAO
98 QAAAAA@1
99 C=C

100 ACH2N
101 8M RING
102 QO
103 CL
104 QHACH2A
105 A\$A(\$A)\$A
106 QA(Q)Q
107 XA(A)A
108 CH3AAACH2A
109 ACH2O
110 NCO
111 NACH2A
112 AA(A)(A)A
113 Onot%A%A
114 CH3CH2A
115 CH3ACH2A
116 CH3AACH2A
117 NAO
118 ACH2CH2A > 1
119 N=A
120 HETEROCYCLIC ATOM > 1 (&...)
121 N HETEROCYCLE
122 AN(A)A
123 OCO
124 QQ
125 AROMATIC RING > 1
126 A!O!A
127 A\$A!O > 1 (&...)
128 ACH2AAACH2A
129 ACH2AACH2A
130 QQ > 1 (&...)
131 QH > 1
132 OACH2A
133 A\$A!N
134 X (HALOGEN)
135 Nnot%A%A
136 O=A > 1
137 HETEROCYCLE
138 QCH2A > 1 (&...)
139 OH
140 O > 3 (&...)
141 CH3 > 2 (&...)
142 N > 1
143 A\$A!O
144 Anot%A%Anot%A
145 6M RING > 1
146 O > 2
147 ACH2CH2A
148 AQ(A)A
149 CH3 > 1
150 A!A\$A!A
151 NH
152 OC(C)C
153 QCH2A
154 C=O
155 A!CH2!A
156 NA(A)A
157 C-O
158 C-N
159 O > 1
160 CH3
161 N
162 AROMATIC
163 6M RING
164 O
165 RING
166 FRAGMENTS

MACCS 322 keys set as defined in tables 1, 2 and 3 [Ref 46] include:

- . 26 atom properties of type P, as listed in Table 1
- . 32 one-atom environments, as listed in Table 3
- . 264 atom-bond-atom combinations listed in Table 4

Total number of keys in three tables is : 322

Atom symbol, X, used for 322 keys [Ref 46] doesn't refer to Halogens as it does for 166 keys. In order to keep the definition of 322 keys consistent with the published definitions, the symbol X is used to imply "others" atoms, but it's internally mapped to symbol X as defined for 166 keys during the generation of key values.

Atom properties-based keys (26):

Key	Description
1	A(AAA) or AA(A)A - atom with at least three neighbors
2	Q - heteroatom
3	Anot%not-A - atom involved in one or more multiple bonds, not aromatic
4	A(AAAA) or AA(A)(A)A - atom with at least four neighbors
5	A(QQ) or QA(Q) - atom with at least two heteroatom neighbors
6	A(QQQ) or QA(Q)Q - atom with at least three heteroatom neighbors
7	QH - heteroatom with at least one hydrogen attached
8	CH2(AA) or ACH2A - carbon with at least two single bonds and at least two hydrogens attached
9	CH3(A) or ACH3 - carbon with at least one single bond and at least three hydrogens attached
10	Halogen
11	A(-A-A-A) or A-A(-A)-A - atom has at least three single bonds
12	AAAAAA@1 > 2 - atom is in at least two different six-membered rings
13	A(\$A\$A\$A) or A\$A(\$A)\$A - atom has more than two ring bonds
14	A\$A!A\$A - atom is at a ring/chain boundary. When a comparison is done with another atom the path passes through the chain bond.
15	Anot%A%Anot%A - atom is at an aromatic/nonaromatic boundary. When a comparison is done with another atom the path passes through the aromatic bond.
16	A!A!A - atom with more than one chain bond
17	A!A\$A!A - atom is at a ring/chain boundary. When a comparison is done with another atom the path passes through the ring bond.
18	A%Anot%A%A - atom is at an aromatic/nonaromatic boundary. When a comparison is done with another atom the path passes through the nonaromatic bond.
19	HETEROCYCLE - atom is a heteroatom in a ring.
20	rare properties: atom with five or more neighbors, atom in four or more rings, or atom types other than H, C, N, O, S, F, Cl, Br, or I
21	rare properties: atom has a charge, is an isotope, has two or more multiple bonds, or has a triple bond.
22	N - nitrogen
23	S - sulfur
24	O - oxygen
25	A(AA)A(A)A(AA) - atom has two neighbors, each with three or more neighbors (including the central atom).
26	CHACH2 - atom has two hydrocarbon (CH2) neighbors

Atomic environments properties-based keys (32):

Key	Description
27	C(CC)
28	C(CCC)
29	C(CN)
30	C(CCN)
31	C(NN)
32	C(NNC)
33	C(NNN)
34	C(CO)
35	C(CCO)
36	C(NO)
37	C(NCO)
38	C(NNO)
39	C(OO)
40	C(COO)
41	C(NOO)

42	C(OOO)
43	Q(CC)
44	Q(CCC)
45	Q(CN)
46	Q(CCN)
47	Q(NN)
48	Q(CNN)
49	Q(NNN)
50	Q(CO)
51	Q(CCO)
52	Q(NO)
53	Q(CNO)
54	Q(NNO)
55	Q(OO)
56	Q(COO)
57	Q(NOO)
58	Q(OOO)

Note: The first symbol is the central atom, with atoms bonded to the central atom listed in parentheses. Q is any non-C, non-H atom. If only two atoms are in parentheses, there is no implication concerning the other atoms bonded to the central atom.

Atom-Bond-Atom properties-based keys: (264)

Key	Description
59	C-C
60	C-N
61	C-O
62	C-S
63	C-Cl
64	C-P
65	C-F
66	C-Br
67	C-Si
68	C-I
69	C-X
70	N-N
71	N-O
72	N-S
73	N-Cl
74	N-P
75	N-F
76	N-Br
77	N-Si
78	N-I
79	N-X
80	O-O
81	O-S
82	O-Cl
83	O-P
84	O-F
85	O-Br
86	O-Si
87	O-I
88	O-X
89	S-S
90	S-Cl
91	S-P
92	S-F
93	S-Br
94	S-Si
95	S-I
96	S-X
97	Cl-Cl
98	Cl-P
99	Cl-F
100	Cl-Br
101	Cl-Si
102	Cl-I
103	Cl-X

104	P-P
105	P-F
106	P-Br
107	P-Si
108	P-I
109	P-X
110	F-F
111	F-Br
112	F-Si
113	F-I
114	F-X
115	Br-Br
116	Br-Si
117	Br-I
118	Br-X
119	Si-Si
120	Si-I
121	Si-X
122	I-I
123	I-X
124	X-X
125	C=C
126	C=N
127	C=O
128	C=S
129	C=Cl
130	C=P
131	C=F
132	C=Br
133	C=Si
134	C=I
135	C=X
136	N=N
137	N=O
138	N=S
139	N=Cl
140	N=P
141	N=F
142	N=Br
143	N=Si
144	N=I
145	N=X
146	O=O
147	O=S
148	O=Cl
149	O=P
150	O=F
151	O=Br
152	O=Si
153	O=I
154	O=X
155	S=S
156	S=Cl
157	S=P
158	S=F
159	S=Br
160	S=Si
161	S=I
162	S=X
163	Cl=Cl
164	Cl=P
165	Cl=F
166	Cl=Br
167	Cl=Si
168	Cl=I
169	Cl=X
170	P=P
171	P=F
172	P=Br

173	P=Si
174	P=I
175	P=X
176	F=F
177	F=Br
178	F=Si
179	F=I
180	F=X
181	Br=Br
182	Br=Si
183	Br=I
184	Br=X
185	Si=Si
186	Si=I
187	Si=X
188	I=I
189	I=X
190	X=X
191	C#C
192	C#N
193	C#O
194	C#S
195	C#Cl
196	C#P
197	C#F
198	C#Br
199	C#Si
200	C#I
201	C#X
202	N#N
203	N#O
204	N#S
205	N#Cl
206	N#P
207	N#F
208	N#Br
209	N#Si
210	N#I
211	N#X
212	O#O
213	O#S
214	O#Cl
215	O#P
216	O#F
217	O#Br
218	O#Si
219	O#I
220	O#X
221	S#S
222	S#Cl
223	S#P
224	S#F
225	S#Br
226	S#Si
227	S#I
228	S#X
229	Cl#Cl
230	Cl#P
231	Cl#F
232	Cl#Br
233	Cl#Si
234	Cl#I
235	Cl#X
236	P#P
237	P#F
238	P#Br
239	P#Si
240	P#I
241	P#X

242	F#F
243	F#Br
244	F#Si
245	F#I
246	F#X
247	Br#Br
248	Br#Si
249	Br#I
250	Br#X
251	Si#Si
252	Si#I
253	Si#X
254	I#I
255	I#X
256	X#X
257	C\$C
258	C\$N
259	C\$O
260	C\$S
261	C\$Cl
262	C\$P
263	C\$F
264	C\$Br
265	C\$Si
266	C\$I
267	C\$X
268	N\$N
269	N\$O
270	N\$S
271	N\$Cl
272	N\$P
273	N\$F
274	N\$Br
275	N\$Si
276	N\$I
277	N\$X
278	O\$O
279	O\$S
280	O\$Cl
281	O\$P
282	O\$F
283	O\$Br
284	O\$Si
285	O\$I
286	O\$X
287	S\$S
288	S\$Cl
289	S\$P
290	S\$F
291	S\$Br
292	S\$Si
293	S\$I
294	S\$X
295	Cl\$Cl
296	Cl\$P
297	Cl\$F
298	Cl\$Br
299	Cl\$Si
300	Cl\$I
301	Cl\$X
302	P\$P
303	P\$F
304	P\$Br
305	P\$Si
306	P\$I
307	P\$X
308	F\$F
309	F\$Br
310	F\$Si

311	F\$I
312	F\$X
313	Br\$Br
314	Br\$Si
315	Br\$I
316	Br\$X
317	Si\$Si
318	Si\$I
319	Si\$X
320	I\$I
321	I\$X
322	X\$X

```
--OutDelim comma | tab | semicolon
```

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

```
--output SD / FP / text / all
```

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

-o, --overwrite

Overwrite existing files.

-q, --quote Yes / No

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

`-r, --root RootName`

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

-s, --size *number*

Size of MACCS keys [Ref 45-47] set to use during fingerprints generation. Possible values: 166 or 322.
Default value: 166.

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

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

Examples:

```
FingerprintsVector;MACCSKeyCount;166;OrderedNumericalValues;ValuesStr  
ing;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 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;ValuesString;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 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 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 ...
```

-w, --WorkingDir *DirName*

Location of working directory. Default: current directory.

EXAMPLES

To generate MACCS keys fingerprints of size 166 in binary bit-vector string format and create a SampleMACCS166FPBin.csv file containing sequential compound IDs along with fingerprints bit-vector strings data, type:

```
% MACCSKeysFingerprints.pl -r SampleMACCS166FPBin -o Sample.sdf
```

To generate MACCS keys fingerprints of size 166 in binary bit-vector string format and create SampleMACCS166FPBin.sdf, SampleMACCS166FPBin.csv and SampleMACCS166FPBin.csv files containing sequential compound IDs in CSV file along with fingerprints bit-vector strings data, type:

```
% MACCSKeysFingerprints.pl --output all -r SampleMACCS166FPBin
-o Sample.sdf
```

To generate MACCS keys fingerprints of size 322 in binary bit-vector string format and create a SampleMACCS322FPBin.csv file containing sequential compound IDs along with fingerprints bit-vector strings data, type:

```
% MACCSKeysFingerprints.pl -size 322 -r SampleMACCS322FPBin -o Sample.sdf
```

To generate MACCS keys fingerprints of size 166 corresponding to count of keys in ValuesString format and create a SampleMACCS166FPCount.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyCount -r SampleMACCS166FPCount
-o Sample.sdf
```

To generate MACCS keys fingerprints of size 322 corresponding to count of keys in ValuesString format and create a SampleMACCS322FPCount.csv file containing sequential compound IDs along with fingerprints vector strings data, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyCount -size 322
-r SampleMACCS322FPCount -o Sample.sdf
```

To generate MACCS keys fingerprints of size 166 in hexadecimal bit-vector string format with ascending bits order and create a SampleMACCS166FPHex.csv file containing compound IDs from MolName along with fingerprints bit-vector strings data, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyBits --size 166 --BitStringFormat
HexadecimalString --BitsOrder Ascending --DataFieldsMode CompoundID
--CompoundIDMode MolName -r SampleMACCS166FPBin -o Sample.sdf
```

To generate MACCS keys fingerprints of size 166 corresponding to count of keys in IDsAndValuesString format and create a SampleMACCS166FPCount.csv file containing compound IDs from MolName line along with fingerprints vector strings data, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyCount --size 166
--VectorStringFormat IDsAndValuesString --DataFieldsMode CompoundID
--CompoundIDMode MolName -r SampleMACCS166FPCount -o Sample.sdf
```

To generate MACCS keys fingerprints of size 166 corresponding to count of keys in IDsAndValuesString format and create a SampleMACCS166FPCount.csv file containing compound IDs using specified data field along with fingerprints vector strings data, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyCount --size 166
--VectorStringFormat IDsAndValuesString --DataFieldsMode CompoundID
--CompoundIDMode DataField --CompoundID Mol_ID -r
SampleMACCS166FPCount -o Sample.sdf
```

To generate MACCS keys fingerprints of size 322 corresponding to count of keys in ValuesString format and create a SampleMACCS322FPCount.tsv file containing compound IDs derived from combination of molecule name line and an explicit compound prefix along with fingerprints vector strings data in a column labels MACCSKeyCountFP, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyCount -size 322 --DataFieldsMode
CompoundID --CompoundIDMode MolnameOrLabelPrefix --CompoundID Cmpd
--CompoundIDLabel MolID --FingerprintsLabel MACCSKeyCountFP --OutDelim
Tab -r SampleMACCS322FPCount -o Sample.sdf
```

To generate MACCS keys fingerprints of size 166 corresponding to count of keys in ValuesString format and create a SampleMACCS166FPCount.csv file containing specific data fields columns along with fingerprints vector strings data, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyCount --size 166
--VectorStringFormat ValuesString --DataFieldsMode Specify --DataFields
Mol_ID -r SampleMACCS166FPCount -o Sample.sdf
```

To generate MACCS keys fingerprints of size 322 corresponding to count of keys in ValuesString format and create a SampleMACCS322FPCount.csv file containing common data fields columns along with fingerprints vector strings data, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyCount --size 322
--VectorStringFormat ValuesString --DataFieldsMode Common -r
SampleMACCS322FPCount -o Sample.sdf
```

To generate MACCS keys fingerprints of size 166 corresponding to count of keys in ValuesString format and create SampleMACCS166FPCount.sdf, SampleMACCS166FPCount.fpf and SampleMACCS166FPCount.csv files containing all data fields columns in CSV file along with fingerprints vector strings data, type:

```
% MACCSKeysFingerprints.pl -m MACCSKeyCount --size 166 --output all
--VectorStringFormat ValuesString --DataFieldsMode All -r
SampleMACCS166FPCount -o Sample.sdf
```

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

InfoFingerprintsFiles.pl, SimilarityMatricesFingerprints.pl, AtomNeighborhoodsFingerprints.pl, ExtendedConnectivityFingerprints.pl, PathLengthFingerprints.pl, TopologicalAtomPairsFingerprints.pl, TopologicalAtomTorsionsFingerprints.pl, TopologicalPharmacophoreAtomPairsFingerprints.pl, TopologicalPharmacophoreAtomTripletsFingerprints.pl

COPYRIGHT

Copyright (C) 2025 Manish Sud. All rights reserved.

This file is part of MayaChemTools.

MayaChemTools is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.