NAME

RDKitEnumerateCompoundLibrary.py - Enumerate a virtual compound library

SYNOPSIS

RDKitEnumerateCompoundLibrary.py [--colmode <collabel or colnum>] [--colRxnName <text or number>] [--colRxnSMARTS <text or number>] [--compute2DCoords <yes or no>] [--infileParams <Name,Value,...>] [--mode <RxnByName or RxnBySMARTS>] [--outfileParams <Name,Value,...>] [--overwrite] [--prodMolNames <UseReactants or Sequential>] [--rxnName <text>] [--rxnNamesFile <FileName or auto>] [--smartsRxn <text>] [--sanitize <yes or no>] [-w <dir>] -i <ReactantFile1,...> -o <outfile>

RDKitEnumerateCompoundLibrary.py [--colmode <collabel or colnum>] [--colRxnName <text or number>] [--colRxnSMARTS <text or number>] [--rxnNamesFile <FileName or auto>] -l | --list

RDKitEnumerateCompoundLibrary.py -h | --help | -e | --examples

DESCRIPTION

Perform a combinatorial enumeration of a virtual library of molecules for a reaction specified using a reaction name or SMARTS pattern and reactant input files.

The SMARTS patterns for supported reactions names [Ref 134] are retrieved from file, ReactionNamesAndSMARTS.csv, available in MayaChemTools data directory. The current list of supported reaction names is shown below:

'1,2,4_triazole_acetohydrazide', '1,2,4_triazole_carboxylic_acid_ester', 3_nitrile_pyridine, Benzimidazole_derivatives_aldehyde, Benzimidazole_derivatives_carboxylic_acid_ester, Benzofuran, Benzothiazole, Benzothiophene, Benzoxazole_aromatic_aldehyde, Benzoxazole_carboxylic_acid, Buchwald_Hartwig, Decarboxylative_coupling, Fischer_indole, Friedlaender_chinoline, Grignard_alcohol, Grignard_carbonyl, Heck_non_terminal_vinyl, Heck_terminal_vinyl, Heteroaromatic_nuc_sub, Huisgen_Cu_catalyzed_1,4_subst, Huisgen_disubst_alkyne, Huisgen_Ru_catalyzed_1,5_subst, Imidazole, Indole, Mitsunobu_imide, Mitsunobu_phenole, Mitsunobu_sulfonamide, Mitsunobu_tetrazole_1, Mitsunobu_tetrazole_2, Mitsunobu_tetrazole_3, Mitsunobu_tetrazole_4, N_arylation_heterocycles, Negishi, Niementowski_quinazoline, Nucl_sub_aromatic_ortho_nitro, Nucl_sub_aromatic_para_nitro, Oxadiazole, Paal_Knorr_pyrrole, Phthalazinone, Pictet_Spengler, Piperidine_indole, Pyrazole, Reductive_amination, Schotten_Baumann_amide, Sonogashira, Spiro_chromanone, Stille, Sulfon_amide, Suzuki, Tetrazole_connect_regioisomer_1, Tetrazole_connect_regioisomer_2, Tetrazole_terminal, Thiazole, Thiourea, Triaryl_imidazole, Urea, Williamson_ether, Wittig

The supported input file formats are: SD (.sdf, .sd), SMILES (.smi, .csv, .tsv, .txt)

The supported output file formats are: SD (.sdf, .sd), SMILES (.smi)

OPTIONS

-c, --colmode <collabel or colnum> [default: collabel]

Use column number or name for the specification of columns in a CSV file containing reaction names along with reaction SMARTS. You may specify a reaction names file using '--rxnNamesFile' option.

--colRxnName <text or number> [default: auto]

Column name or number corresponding to reaction names. The default value is automatically set based on the value of '-c, --colmode': 'RxnName' for 'collabel'; Reaction name column number for 'colnum'.

--colRxnSMARTS <text or number> [default: auto]

Column name or number corresponding to reaction SMARTS strings. The default value is automatically set based on the value of '-c, --colmode': 'RxnSMARTS' for 'collabel'; Reacton SMARTS column number for 'colnum'.

--compute2DCoords <yes or no> [default: yes]

Compute 2D coordinates of product molecules before writing them out.

-i, --infiles <ReactantFile1, ReactantFile2...>

Comma delimited list of reactant file names for enumerating a compound library using reaction SMARTS. The number of reactant files must match number of reaction components in reaction SMARTS. All reactant input files must have the same format.

--infileParams <Name,Value,...> [default: auto]

A comma delimited list of parameter name and value pairs for reading molecules from files. The supported parameter names for different file formats, along with their default values, are shown below:

```
SD, MOL: removeHydrogens,yes,sanitize,yes,strictParsing,yes
SMILES: smilesColumn,1,smilesNameColumn,2,smilesDelimiter,space,
    smilesTitleLine,auto,sanitize,yes
```

Possible values for smilesDelimiter: space, comma or tab. These parameters apply to all reactant input files, which must have the same file format.

-e, --examples

Print examples.

-h, --help

Print this help message.

-l, --list

List available reaction names along with corresponding SMARTS patterns without performing any enumeration. In addition, reaction SMARTS patterns are validated.

-m, --mode <RxnByName or RxnBySMARTS> [default: RxnByName]

Indicate whether a reaction is specified by a reaction name or a SMARTS pattern. Possible values: RxnByName or RxnBySMARTS.

-o, --outfile <outfile>

Output file name.

--outfileParams <Name,Value,...> [default: auto]

A comma delimited list of parameter name and value pairs for writing molecules to files. The supported parameter names for different file formats, along with their default values, are shown below:

```
SD: kekulize,yes,forceV3000,no
SMILES: smilesKekulize,no,smilesDelimiter,space, smilesIsomeric,yes,
    smilesTitleLine,yes
```

-p, --prodMolNames <UseReactants or Sequential> [default: UseReactants]

Generate names of product molecules using reactant names or assign names in a sequential order. Possible values: UseReactants or Sequential. Format of molecule names: UseReactants - <ReactName1>_<ReactName2>..._Prod<Num>; Sequential - Prod<Num>

--overwrite

Overwrite existing files.

-r, --rxnName <text>

Name of a reaction to use for enumerating a compound library. This option is only used during 'RxnByName' value of '-m, --mode' option.

--rxnNamesFile <FileName or auto> [default: auto]

Specify a file name containing data for names of reactions and SMARTS patterns or use default file, ReactionNamesAndSMARTS.csv, available in MayaChemTools data directory.

Default reactions SMARTS file format: RxnName,RxnSMARTS.

The local file format is assumed to be same as the default file format. You may explicitly specify column names or numbers for reaction name and reaction SMARTS using '--colRxnName' and '--colRxnSMARTS' options.

-s, --smartsRxn <text>

SMARTS pattern of a reaction to use for enumerating a compound library. This option is only used during 'RxnBySMARTS' value of '-m, --mode' option.

--sanitize <yes or no> [default: yes]

Sanitize product molecules before writing them out.

-w, --workingdir <dir>

Location of working directory which defaults to the current directory.

EXAMPLES

To list all available reaction names along with their SMARTS pattern, type:

```
    % RDKitEnumerateCompoundLibrary.py -l
```

To perform a combinatorial enumeration of a virtual compound library corresponding to named amide reaction, Schotten_Baumann_amide, and write out a SMILES file type:

```
    % RDKitEnumerateCompoundLibrary.py -r Schotten_Baumann_amide
      -i 'SampleAcids.smi,SampleAmines.smi' -o SampleOutCmpdLibrary.smi
```

To run the previous command using a local reaction names file with explicit specification of column names containing reaction names and SMARTS, and write out a SMILES file type:

```
    % RDKitEnumerateCompoundLibrary.py -r Schotten_Baumann_amide
      --rxnNamesFile ReactionNamesAndSMARTS.csv
      --colmode collabel --colRxnName RxnName --colRxnSMARTS RxnSMARTS
      -i 'SampleAcids.smi,SampleAmines.smi' -o SampleOutCmpdLibrary.smi
```

To perform a combinatorial enumeration of a virtual compound library corresponding to an amide reaction specified using a SMARTS pattern and write out a SD file containing sanitized molecules, computed 2D coordinates, and generation of molecule names from reactant names, type:

```
    % RDKitEnumerateCompoundLibrary.py -m RxnBySMARTS
      -s '[O:2]=[C:1][OH].[N:3]>>[O:2]=[C:1][N:3]'
      -i 'SampleAcids.smi,SampleAmines.smi' -o SampleOutCmpdLibrary.sdf
```

To perform a combinatorial enumeration of a virtual compound library corresponding to an amide reaction specified using a SMARTS pattern and write out a SD file containing unsanitized molecules, without generating 2D coordinates, and a sequential generation of molecule names, type:

```
    % RDKitEnumerateCompoundLibrary.py -m RxnBySMARTS -c no --sanitize no
      -p Sequential -s '[O:2]=[C:1][OH].[N:3]>>[O:2]=[C:1][N:3]'
      -i 'SampleAcids.smi,SampleAmines.smi' -o SampleOutCmpdLibrary.sdf
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

RDKitConvertFileFormat.py, RDKitFilterPAINS.py, RDKitSearchFunctionalGroups.py, RDKitSearchSMARTS.py

COPYRIGHT

The functionality available in this script is implemented using RDKit, an open source toolkit for cheminformatics developed by Greg Landrum.

This file is part of MayaChemTools.