NAME

PyMOLSplitChainsAndLigands.py - Split macromolecule into chains and ligands

SYNOPSIS

PyMOLSplitChainsAndLigands.py [--chainIDs <First, All or ID1,ID2...>] [--ligandIDs <Largest, All or ID1,ID2...>] [--ligandFileFormat <PDB, SDF, MDLMOL>] [--mode <Chains or ChainsLigands>] [--keepInorganics <yes or no>] [--keepSolvents <yes or no>] [--overwrite] [-w <dir>] -i <infile>

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

DESCRIPTION

Spit a macromolecule into chains and ligands, and write them out to different files. The solvents and inorganic molecules may be optionally removed from chains. You may also skip the generation of ligand files and write out a chain along with associated ligands into the same chain file.

The supported input file format is: PDB (.pdb), CIF (.cif)

The supported output file formats are: Chains - PDB (.pdb); Ligands: PDB (.pdb), SD file (.sdf, .sd), MDL MOL (.mol)

The names of the output files are automatically generated from the name of input file as shown below:

```
Chains: <InfileRoot>_<ChainID>.pdb
Ligands: <InfileRoot>_<ChainID>.{pdb,sdf,sd,mol}
```

OPTIONS

-c, --chainIDs <First, All or ID1,ID2...> [default: All]

List of chain IDs for splitting input file. Possible values: First, All, or a comma delimited list of chain IDs. The default is to use all chain IDs in input file.

-e, --examples

Print examples.

-h, --help

Print this help message.

-i, --infile <infile>

Input file name.

-l, --ligandIDs <Largest, All or ID1,ID2...> [default: Largest]

List of ligand IDs present in chains for splitting input file. Possible values: Largest, All, or a comma delimited list of ligand IDs. The default is to use the largest ligand present in all or specified chains in input file. This option is ignored during 'Chains' value of '--mode' option.

Ligands are identified using organic selection operator available in PyMOL. It'll also identify buffer molecules as ligands. The largest ligand contains the highest number of heavy atoms.

--ligandFileFormat <PDB, SDF, MDLMOL> [default: SDF]

Ligand file format.

-m, --mode <Chains or ChainsLigands> [default: ChainsLigands]

Split input file into chains or chains and ligands. The ligands are kept together chains in the output files for 'Chains' mode. Separate files are generated for ligands during 'ChainsAndLigands' mode.

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

Keep inorganic molecules during splitting of input file and write them to output files. The inorganic molecules are identified using inorganic selection operator available in PyMOL.

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

Keep solvent molecules during splitting of input file and write them to output files. The solvent molecules are identified using solvent selection operator available in PyMOL.

--overwrite

Overwrite existing files.

-w, --workingdir <dir>

Location of working directory which defaults to the current directory.

EXAMPLES

To split a macromolecule into the first chain and the largest ligand in the first chain along with solvent and inorganic molecules, and write chain PDB and ligand SDF files, type:

```
% PyMOLSplitChainsAndLigands.py -i Sample3.pdb
```

To split a macromolecule into all chains and all ligands across all chains along with solvent and inorganic molecules, and write out corresponding chain and ligand files, type:

```
% PyMOLSplitChainsAndLigands.py -i Sample3.pdb -c All -l All
```

To split a macromolecule into all chains along with any associated ligands without any solvent and inorganic molecules, and write corresponding PDB files for chains and skipping generation of any ligand files, type:

```
% PyMOLSplitChainsAndLigands.py -c all -m Chains --keepSolvents no
  --keepInorganics no -i Sample3.pdb
```

To split a macromolecule into a specific chain and a specific ligand in the chain along with solvent and inorganic molecules, and write chain PDB and ligand MDLMOL files, type:

```
% PyMOLSplitChainsAndLigands.py -c E -l ADP --ligandFileFormat MDLMOL
  -i Sample3.pdb
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

PyMOLAlignChains.py, PyMOLVisualizeMacromolecules.py

COPYRIGHT

The functionality available in this script is implemented using PyMOL, a molecular visualization system on an open source foundation originally developed by Warren DeLano.

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.