

NAME

PyMOLVisualizeMacromolecules.py - Visualize macromolecules

SYNOPSIS

```
PyMOLVisualizeMacromolecules.py [--align <yes or no>] [--alignMethod <align, cealign, super>] [--alignMode <FirstChain or
Complex>] [--alignRefFile <filename>] [--allowEmptyObjects <yes or no>] [--chainIDs <First, All or ID1,ID2...>] [
--ligandIDs <Largest, All or ID1,ID2...>] [--labelFontID <number>] [--PMLOut <yes or no>] [
--pocketContactsInorganicColor <text>] [--pocketContactsLigandColor <text>] [--pocketContactsSolventColor <text>] [
--pocketDistanceCutoff <number>] [--pocketLabelColor <text>] [--pocketSurface <yes or no>] [--surfaceChain <yes or
no>] [--surfaceChainComplex <yes or no>] [--surfaceComplex <yes or no>] [--surfaceColorPalette <RedToWhite or
WhiteToGreen>] [--surfaceElectrostaticsChain <yes or no>] [--surfaceTransparency <number>] [--overwrite] [-w <dir>]
-i <infile1,infile2,infile3...> -o <outfile>
```

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

DESCRIPTION

Generate PyMOL visualization files for viewing surfaces, chains, ligands, ligand binding pockets, and interactions between ligands and binding pockets in macromolecules including proteins and nucleic acids.

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

The supported output file formats are: PyMOL script file (.pml), PyMOL session file (.pse)

A variety of PyMOL groups and objects may be created for visualization of macromolecules. These groups and objects correspond to complexes, surfaces, chains, ligands, inorganics, ligand binding pockets, pocket, polar interactions, and pocket hydrophobic surfaces. A complete hierarchy of all possible PyMOL groups and objects is shown below:

```
<PDBFileRoot>
  .Complex
    .Complex
    .Surface
  .Chain<ID>
    .Complex
      .Complex
      .Surface
    .Chain
      .Chain
      .Surface
      .Vacuum_Electrostatics
        .Contact_Potentials
        .Map
        .Legend
  .Solvent
  .Inorganic
  .Ligand<ID>
    .Ligand
      .Ligand
      .BallAndStick
    .Pocket
      .Pocket
      .Polar_Contacts
      .Surface
    .Pocket_Solvent
      .Pocket_Solvent
      .Polar_Contacts
    .Pocket_Inorganic
      .Pocket_Inorganic
      .Polar_Contacts
  .Ligand<ID>
    .Ligand
      ... ..
    .Pocket
      ... ..
    .Pocket_Solvent
      ... ..
    .Pocket_Inorganic
      ... ..
  .Chain<ID>
    ... ..
  .Ligand<ID>
    ... ..
  .Ligand<ID>
    ... ..
```

```
.Chain<ID>
... ..
```

The hydrophobic surfaces are not created for complete complex and chain complex in input file(s) by default. In addition, the creation of chain surface corresponding to vacuum electrostatics contact potential is turned off by default. A word to the wise: The creation of surface objects may slow down loading of PML file and generation of PSE file, based on the size of input complexes. The generation of PSE file may also fail.

OPTIONS

-a, --align <yes or no> [default: no]

Align input files to a reference file before visualization.

--alignMethod <align, cealign, super> [default: super]

Alignment methodology to use for aligning input files to a reference file.

--alignMode <FirstChain or Complex> [default: FirstChain]

Portion of input and reference files to use for spatial alignment of input files against reference file. Possible values: FirstChain or Complex.

The FirstChain mode allows alignment of the first chain in each input file to the first chain in the reference file along with moving the rest of the complex to coordinate space of the reference file. The complete complex in each input file is aligned to the complete complex in reference file for the Complex mode.

--alignRefFile <filename> [default: FirstInputFile]

Reference input file name. The default is to use the first input file name specified using '-i, --infiles' option.

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

Allow creation of empty PyMOL objects corresponding to solvent and inorganic atom selections across chains and ligands in input file(s). By default, the empty objects are marked for deletion.

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

List of chain IDs to use for visualizing macromolecules. Possible values: First, All, or a comma delimited list of chain IDs. The default is to use the chain ID for the first chain in input file.

-e, --examples

Print examples.

-h, --help

Print this help message.

-i, --infiles <infile1,infile2,infile3...>

Input file name.

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

List of ligand IDs present in chains for visualizing macromolecules to highlight ligand interactions. 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.

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.

--labelFontID <number> [default: 7]

Font ID for drawing labels. Default: 7 (Sans Bold). Valid values: 5 to 16. The specified value must be a valid PyMOL font ID. No validation is performed. The complete lists of valid font IDs is available at: pymolwiki.org/index.php/Label_font_id. Examples: 5 - Sans; 7 - Sans Bold; 9 - Serif; 10 - Serif Bold.

-o, --outfile <outfile>

Output file name.

-p, --PMLOut <yes or no> [default: yes]

Save PML file during generation of PSE file.

--pocketContactsInorganicColor <text> [default: deepsalmon]

Color for drawing polar contacts between inorganic and pocket residues. The specified value must be valid color. No validation is performed.

--pocketContactsLigandColor <text> [default: orange]

Color for drawing polar contacts between ligand and pocket residues. The specified value must be valid color. No validation is performed.

- pocketContactsSolventColor <text> [default: marine]
Color for drawing polar contacts between solvent and pocket residues.. The specified value must be valid color. No validation is performed.
- pocketDistanceCutoff <number> [default: 5.0]
Distance in Angstroms for identifying pocket residues around ligands.
- pocketLabelColor <text> [default: magenta]
Color for drawing residue or atom level labels for a pocket. The specified value must be valid color. No validation is performed.
- pocketSurface <yes or no> [default: yes]
Hydrophobic surface around pocket. The pocket surface is colored by hydrophobicity. It is only valid for proteins. The color of amino acids is set using the Eisenberg hydrophobicity scale. The color varies from red to white, red being the most hydrophobic amino acid.
- surfaceChain <yes or no> [default: yes]
Hydrophobic surface around individual chain. The surface is colored by hydrophobicity. It is only valid for proteins.
- surfaceChainComplex <yes or no> [default: no]
Hydrophobic surface around chain complete complex. The surface is colored by hydrophobicity. It is only valid for proteins.
- surfaceComplex <yes or no> [default: no]
Hydrophobic surface around complete complex. The surface is colored by hydrophobicity. It is only valid for proteins.
- surfaceColorPalette <RedToWhite or WhiteToGreen> [default: RedToWhite]
Color palette for hydrophobic surfaces around chains and pockets in proteins. Possible values: RedToWhite or WhiteToGreen from most hydrophobic amino acid to least hydrophobic. The colors values for amino acids are taken from color_h script available as part of the Script Library at PyMOL Wiki.
- surfaceElectrostaticsChain <yes or no> [default: no]
Vacuum electrostatics contact potential surface around individual chain. A word of to the wise from PyMOL documentation: The computed protein contact potentials are only qualitatively useful, due to short cutoffs, truncation, and lack of solvent "screening".
- surfaceTransparency <number> [default: 0.25]
Surface transparency for molecular surfaces.
- overwrite
Overwrite existing files.
- w, --workingdir <dir>
Location of working directory which defaults to the current directory.

EXAMPLES

To visualize the first chain, the largest ligand in the first chain, and ligand binding pockets to highlight ligand interaction with pocket residues, solvents and inorganics, in a PDB file, and generate a PML file, type:

```
% PyMOLVisualizeMacromolecules.py -i Sample4.pdb -o Sample4.pml
```

To visualize all chains, all ligands in all chains, and all ligand binding pockets to highlight ligand interaction with pocket residues, solvents and inorganics, in a PDB file, and generate a PML file, type:

```
% PyMOLVisualizeMacromolecules.py -c All -l All -i Sample4.pdb -o Sample4.pml
```

To visualize all chains, ligands, and ligand binding pockets along with displaying all hydrophobic surfaces and chain electrostatic surface, in a PDB file, and generate a PML file, type:

```
% PyMOLVisualizeMacromolecules.py -c All -l All
--surfaceElectrostaticsChain yes --surfaceChainComplex yes
--surfaceComplex yes -i Sample4.pdb -o Sample4.pml
```

To visualize chain E, ligand ADP in chain E, and ligand binding pockets to highlight ligand interaction with pocket residues, solvents and inorganics, in a PDB file, and generate a PML file, type:

```
% PyMOLVisualizeMacromolecules.py -c E -l ADP -i Sample3.pdb
-o Sample3.pml
```

To visualize chain E, ligand ADP in chain E, and ligand binding pockets to highlight ligand interaction with pocket residues, solvents and inorganics, in a PDB file, and generate a PSE file, type:

```
% PyMOLVisualizeMacromolecules.py -c E -l ADP -i Sample3.pdb
-o Sample3.pse
```

To visualize the first chain, the largest ligand in the first chain, and ligand binding pockets to highlight ligand interaction with pocket residues, solvents and inorganics, in PDB files, along with aligning first chain in each input file to the first chain in first input file, and generate a PML file, type:

```
% PyMOLVisualizeMacromolecules.py --align yes -i
"Sample5.pdb,Sample6.pdb,Sample7.pdb" -o SampleOut.pml
```

To visualize all chains, all ligands in all chains, and all ligand binding pockets to highlight ligand interaction with pocket residues, solvents and inorganics, in PDB files, along with aligning first chain in each input file to the first chain in first input file, and generate a PML file, type:

```
% PyMOLVisualizeMacromolecules.py --align yes -c All -l All -i
"Sample5.pdb,Sample6.pdb,Sample7.pdb" -o SampleOut.pml
```

To visualize all chains, all ligands in all chains, and all ligand binding pockets to highlight ligand interaction with pocket residues, solvents and inorganics, in PDB files, along with aligning first chain in each input file to the first chain in a specified PDB file using a specified alignment method, and generate a PML file, type:

```
% PyMOLVisualizeMacromolecules.py --align yes --alignMode FirstChain
--alignRefFile Sample5.pdb --alignMethod super -c All -l All -i
"Sample5.pdb,Sample6.pdb,Sample7.pdb" -o SampleOut.pml
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

DownloadPDBFiles.pl, PyMOLVisualizeCryoEMDensity.py, PyMOLVisualizeElectronDensity.py

COPYRIGHT

Copyright (C) 2018 Manish Sud. All rights reserved.

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.