NAME

PyMOLVisualizeElectronDensity.py - Visualize electron density

SYNOPSIS

PyMOLVisualizeElectronDensity.py [--allowEmptyObjects <yes or no>] [--BFactorChainCartoonPutty <yes or no>] [--BFactorColorPalette <text>] [--chainIDs <First, All or ID1,ID2...>] [--EDMapSuffixes <CompositeMap,None,...>] [--ligandIDs <Largest, All or ID1,ID2...>] [--labelFontID <number>] [--meshCarveRadius <number>] [--meshComplex <yes or no>] [--meshChainComplex <yes or no>] [--meshColorCompositeMap <text>] [--meshLevelCompositeMap <number>] [--meshWidth <number>] [--mesh1ColorDiffMap <text>] [--mesh1LevelDiffMap <number>] [--mesh2ColorDiffMap <text>] [--mesh2LevelDiffMap <number>] [--PMLOut <yes or no>] [--pocketContactsLigandColor <text>] [--pocketContactsSolventColor <text>] [--pocketContactsInorganicColor <text>] [--pocketDistanceCutoff <number>] [--pocketLabelColor <text>] [--pocketSurface <yes or no>] [--surfaceComplex <yes or no>] [--surfaceChainComplex <yes or no>] [--surfaceTransparency <number>] [--overwrite] [-w <dir>] -i <infile> -o <outfile>

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

DESCRIPTION

Generate PyMOL visualization files for viewing X-ray electron density around chains, ligands, and ligand binding pockets in macromolecules including proteins and nucleic acids.

The supported input file formats are: Macromolecule - PDB (.pdb) or CIF(.cif), Electron Density - Collaborative Computational Project Number 4 (CCP4) (.ccp4)

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

Two types of CCP4 electron density map files may be used for visualizing electron density. These file types along with default file names are shown below:

```
CompositeMap (2Fobs - Fcalc) - <InfileRoot>.ccp4 (required)
DifferenceMap (Fobs - Fcalc) - <InfileRoot>_diff.ccp4 (optional)
```

The compsite map file must be present. The difference map file is optional. The mesh and surface PyMOL objects are not generated for missing difference map file.

The electron density present in composite map file is generated by adding two difference maps to a calculated map (Fcalc) as shown below:

```
Fcalc + 2(Fobs - Fcalc) = 2Fobs - Fcalc
```

The following types of meshes may be created by default for electron density present in composite and difference map files:

```
CompositeMesh - ContourLevel: 1; Color: Blue
DiffMesh1 - ContourLevel: 3; Color: Green
DiffMesh2 - ContourLevel: -3; Color: Red
```

The two meshes created for difference maps correspond to false negative and false positive in terms of electron density present in the model. The first mesh shown in green color corresponds to observed electron density missing in the model. The second mesh in in red color indicates model electron density not observed in the experiment.

A variety of PyMOL groups and objects may be created for visualization of electron density present in map files. These groups and objects correspond to maps, meshes, 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
        .2Fo-Fc
            .Map
            .Mesh
```

```
                        .Surface
                .Fo-Fc
                    .Map
                    .Mesh1
                    .Surface1
                    .Mesh2
                    .Surface2
        .Chain<ID>
            .Complex
                .Complex
                .2Fo-Fc
                        .Mesh
                        .Surface
                .Fo-Fc
                    .Mesh1
                    .Surface1
                    .Mesh2
                    .Surface2
            .Chain
                .Chain
                .BFactor
            .Solvent
            .Inorganic
            .Ligand<ID>
                .Ligand
                    .Ligand
                    .2Fo-Fc
                        .Mesh
                        .Surface
                    .Fo-Fc
                        .Mesh1
                        .Surface1
                        .Mesh2
                        .Surface2
                .Pocket
                    .Pocket
                    .2Fo-Fc
                        .Mesh
                        .Surface
                    .Fo-Fc
                        .Mesh1
                        .Surface1
                        .Mesh2
                        .Surface2
                    .Polar_Contacts
                    .Surface
                .Pocket_Solvent
                    .Pocket_Solvent
                    .2Fo-Fc
                        .Mesh
                        .Surface
                    .Fo-Fc
                        .Mesh1
                        .Surface1
                        .Mesh2
                        .Surface2
                    .Polar_Contacts
                .Pocket_Inorganic
                    .Pocket_Inorganic
                    .2Fo-Fc
                        .Mesh
                        .Surface
                    .Fo-Fc
                        .Mesh1
```

```
                        .Surface1
                        .Mesh2
                        .Surface2
                  .Polar_Contacts
          .Ligand<ID>
              .Ligand
                  ... ... ...
              .Pocket
                  ... ... ...
              .Pocket_Solvent
                  ... ... ...
              .Pocket_Inorganic
                  ... ... ...
      .Chain<ID>
          ... ... ...
          .Ligand<ID>
              ... ... ...
          .Ligand<ID>
              ... ... ...
      .Chain<ID>
          ... ... ...
```

The meshes and surfaces are not created for complete complex in input file by default. A word to the wise: The creation of these surface and mesh objects may slow down loading of PML file and generation of PSE file, based on the size of input complex and map files. The generation of PSE file may also fail.

OPTIONS

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

Allow creation of empty PyMOL objects corresponding to solvent and inorganic atom selections across chains, ligands, and ligand binding pockets in input file.

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

List of chain IDs to use for visualizing electron density. 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.

-b, --BFactorChainCartoonPutty <yes or no> [default: yes]

A cartoon putty around individual chains colored by B factors. The minimum and maximum values for B factors are automatically detected. These values indicate spread of electron density around atoms. The 'blue_white_red' color palette is deployed for coloring the cartoon putty.

--BFactorColorPalette <text> [default: blue_white_red]

Color palette for coloring cartoon putty around chains generated using B factors. An valid PyMOL color palette name is allowed. No validation is performed. The complete list of valid color palette names is a available at: pymolwiki.org/index.php/Spectrum. Examples: blue_white_red, blue_white_magenta, blue_red, green_white_red, green_red.

-e, --examples

Print examples.

--EDMapSuffixes <CompositeMap,None,...> [default: auto]

Electron density map file suffixes for generating names of map files from the root of input file. It is a pairwise comma delimited list of 'EDMapType' and file suffix.

Supported values for 'EDMapType': 'CompositeMap, DifferenceMap'. Supported value for file suffix: Any valid string.

Default value: 'CompositeMap,None,DifferenceMap,_diff'

The default names of the map files, generated form a combination of 'InfileRoot' and 'EDSMapType' are shown below:

```
CompositeMap (2Fobs - Fcalc) - <InfileRoot>.ccp4
DifferenceMap (Fobs - Fcalc) - <InfileRoot>_diff.ccp4
```

The composite map file must be present. The difference map file is optional.

-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 visualizing electron density across ligands and ligand binding pockets. 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.

--meshCarveRadius <number> [default: 1.6]

Radius in Angstroms around atoms for including electron density.

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

Create meshes for complete complex in input file corresponding to composite and difference maps. A total of three meshes, one for composite map and two for difference map, are created for the complete complex.

The composite and difference maps are always loaded for the complex.

--meshChainComplex <yes, no, or auto> [default: auto]

Create meshes for individual chain complex in input file corresponding to composite and difference maps. A total of three meshes, one for composite map map and two for difference map, are created for each chain complex. By default, the meshes are automatically created for chain complexes without any ligands.

--meshColorCompositeMap <text> [default: blue]

Line color for mesh corresponding to composite map. The specified value must be valid color. No validation is performed.

--meshLevelCompositeMap <number> [default: 1.0]

Contour level in sigma units for generating mesh corresponding to composite map.

--meshWidth <number> [default: 0.5]

Line width for mesh lines corresponding to composite and difference maps.

--mesh1ColorDiffMap <text> [default: green]

Line color for first mesh corresponding to difference map at contour level specified by '--mesh1LevelDiffMap'. The specified value must be valid color. No validation is performed.

--mesh1LevelDiffMap <number> [default: 3.0]

Contour level in sigma units for generating first mesh corresponding to to difference map.

--mesh2ColorDiffMap <text> [default: red]

Line color for second mesh corresponding to difference map at contour level specified by '--mesh2LevelDiffMap'. The specified value must be valid color. No validation is performed.

--mesh2LevelDiffMap <number> [default: -3.0]

Contour level in sigma units for generating second mesh corresponding to difference map.

-o, --outfile <outfile>

Output file name.

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

>   Save PML file during generation of PSE file.

--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.

--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.

--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.

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

>   Create surfaces for complete complex in input file corresponding to composite and difference maps. A total of three surfaces, one for composite map and two for difference map, are created for the complete complex.
>
>   The composite and difference maps are always loaded for the complex.

--surfaceChainComplex <yes, no or auto> [default: auto]

>   Create surfaces for individual chain complexes in input file corresponding to composite and difference maps. A total of three surfaces, one for composite map and two for difference map, are created for each chain complex. By default, the surfaces are automatically created for chain complexes without any ligands.

--surfaceTransparency <number> [default: 0.25]

>   Surface transparency for molecular and electron density surfaces.

--overwrite

>   Overwrite existing files.

-w, --workingdir <dir>

>   Location of working directory which defaults to the current directory.

EXAMPLES

To visualize electron density for the largest ligand in the first chain, and ligand binding pockets to highlight ligand interactions with pockect residues, solvents and inorganics, in a PDB and using corresponding map files, and generate a PML file, type:

```
% PyMOLVisualizeElectronDensity.py -i Sample3.pdb -o Sample3.pml
```

To visualize electron density for all ligands in all chains, and ligand binding pockets to highlight ligand interactions with pockect residues, solvents and inorganics, in a PDB and using corresponding map files, and generate a PML file, type:

```
% PyMOLVisualizeElectronDensity.py -i Sample3.pdb -o Sample3.pml
   -c All -l All
```

To visualize electron density for all chains and ligands, along with displaying meshes and surfaces for complete complex and individual chains, in a PDB file and using corresponding map files, and generate a PML file, type:

```
% PyMOLVisualizeElectronDensity.py -i Sample3.pdb -o Sample3.pml
  --chainIDs All --ligandIDs All --meshComplex yes --surfaceComplex yes
  --meshChainComplex yes --surfaceChainComplex yes
```

To visualize electron density for ligand ADP in chain E along with ligand binding pocket, in a PDB and using corresponding map files, and generate a PSE file, type:

```
% PyMOLVisualizeElectronDensity.py -i Sample3.pdb -o Sample3.pse
  --chainIDs E --ligandIDs ADP
```

To visualize electron density for all igands in all chains along with their binding pockets in a PDB file and using explicit file name suffixes for map files, and generate a PML file, type:

```
% PyMOLVisualizeElectronDensity.py -i Sample3.pdb -o Sample3.pml
  --chainIDs All --ligandIDs All --EDMapSuffixes "CompositeMap,None,
  DifferenceMap,_diff"
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

DownloadPDBFiles.pl, PyMOLVisualizeCryoEMDensity.py, PyMOLVisualizeMacromolecules.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.