NAME

PyMOLVisualizeElectronDensity.py - Visualize electron density

SYNOPSIS

PyMOLVisualizeElectronDensity.py [--align <yes or no>] [--alignMethod <align, cealign, super>] [--alignMode <FirstChain or Complex>] [--alignRefFile <filename>] [--allowEmptyObjects <yes or no>] [--BFactorChainCartoonPutty <yes or no>] [--BFactorColorPalette <text>] [--chainIDs <First, All or ID1,ID2...>] [--EDMapFiles <file1,file2,...>] [--EDMapSuffixes <CompositeMap,None,...>] [--ignoreHydrogens <yes or no>] [--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>] [--pocketContactsLigandHydrophobicColor <text>] [--pocketContactsSolventColor <text>] [--pocketContactsInorganicColor <text>] [--pocketContactsCutoff <number>] [--pocketDistanceCutoff <number>] [--pocketLabelColor <text>] [--pocketSurface <yes or no>] [--selectionsChain <ObjectName,SelectionSpec,...>] [--selectionsChainStyle <DisplayStyle>] [--surfaceComplex <yes or no>] [--surfaceChainComplex <yes or no>] [--surfaceTransparency <number>] [--volumeCarveRadius <number>] [--volumeComplex <yes or no>] [--volumeChainComplex <yes, no, or auto>] [--volumeColorRampCompositeMap <text>] [--volumeColorRampDiffMap <text>] [--overwrite] [-w <dir>] -i <infile1,infile2...> -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, volume, and surface PyMOL objects are not generated for missing difference map file.

The electron density present in a 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 and volumes may be created by default for electron density present in composite and difference map files:

```
CompositeVolume - VolumeColorRamp: 2fofc
CompositeMesh - ContourLevel: 1; Color: Blue
DiffVolume - VolumeColorRamp: fofc
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, volumes, meshes, surfaces,chains, ligands, inorganics, ligand binding pockets, 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
            .Volume
            .Mesh
            .Surface
        .Fo-Fc
            .Map
            .Volume
            .Mesh1
            .Surface1
            .Mesh2
            .Surface2
    .Chain<ID>
        .Complex
            .Complex
            .2Fo-Fc
                .Volume
                .Mesh
                .Surface
            .Fo-Fc
                .Volume
                .Mesh1
                .Surface1
                .Mesh2
                .Surface2
        .Chain
            .Chain
            .BFactor
            .Selections
                .<Name1>
                    .Selection
                    .2Fo-Fc
                        .Volume
                        .Mesh
                        .Surface
                    .Fo-Fc
                        .Volume
                        .Mesh1
                        .Surface1
                        .Mesh2
                        .Surface2
                .<Name2>
                    ... ... ..
        .Solvent
        .Inorganic
        .Ligand<ID>
            .Ligand
                .Ligand
                .2Fo-Fc
                    .Volume
                    .Mesh
                    .Surface
                .Fo-Fc
                    .Volume
                    .Mesh1
                    .Surface1
                    .Mesh2
                    .Surface2
            .Pocket
```

```
                        .Pocket
                        .2Fo-Fc
                            .Volume
                            .Mesh
                            .Surface
                        .Fo-Fc
                            .Volume
                            .Mesh1
                            .Surface1
                            .Mesh2
                            .Surface2
                        .Polar_Contacts
                        .Hydrophobic_Contacts
                        .Surface
                    .Pocket_Solvent
                        .Pocket_Solvent
                        .2Fo-Fc
                            .Volume
                            .Mesh
                            .Surface
                        .Fo-Fc
                            .Volume
                            .Mesh1
                            .Surface1
                            .Mesh2
                            .Surface2
                        .Polar_Contacts
                    .Pocket_Inorganic
                        .Pocket_Inorganic
                        .2Fo-Fc
                            .Volume
                            .Mesh
                            .Surface
                        .Fo-Fc
                            .Volume
                            .Mesh1
                            .Surface1
                            .Mesh2
                            .Surface2
                        .Polar_Contacts
                .Ligand<ID>
                    .Ligand
                        ... ... ...
                    .Pocket
                        ... ... ...
                    .Pocket_Solvent
                        ... ... ...
                    .Pocket_Inorganic
                        ... ... ...
            .Chain<ID>
                ... ... ...
                .Ligand<ID>
                    ... ... ...
                .Ligand<ID>
                    ... ... ...
            .Chain<ID>
                ... ... ...
        <PDBFileRoot>
            .Complex
                ... ... ...
            .Chain<ID>
                ... ... ...
                .Ligand<ID>
                    ... ... ...
```

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

The meshes, volumes, and surfaces are not created for complete complex in each input file by default. A word to the wise: The creation of these surface, volume, 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, --align <yes or no> [default: no]

Align input files to a reference file before visualization along with available electron density map files.

--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, ligands, and ligand binding pockets in input file(s).

-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 each 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. Any 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.

--EDMapFiles <file1,file1,file3...> [default: auto]

Pairwise comma delimited list of composite and difference electron density map files corresponding to input files. By default, the names of electron density files are automatically generated using a combination of input file names and file suffixes '--EDMapSuffixes'.

The first file with in each pairs of filenames correspond to composite electron density map. A composite file must be present for each input file. The second file corresponds to difference electron density map. The difference map file is optional. A value of 'None' must be used to represent a missing difference map file.

The number of specified files must be twice the number of input files.

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

Electron density map file suffixes for generating names of map files from the root of input files. It is a

pairwise comma delimited list of 'EDMapType' and file suffix.

This option is ignored during explicit specification of electron density map files using '--EDMapFiles'.

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

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

This option is only used for 'Auto' value of '--EDMapFilesMode' option.

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 files must be present. The difference map files are optional.

-h, --help

Print this help message.

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

Input file names.

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

Ignore hydrogens for ligand and pocket views.

-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 each 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 each input file using corresponding 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 each input file using corresponding 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 meshes corresponding to composite maps. The specified value must be valid color. No validation is performed.

--meshLevelCompositeMap <number> [default: 1.0]

Contour level in sigma units for generating meshes corresponding to composite maps.

--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 maps 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 maps.

--mesh2ColorDiffMap <text> [default: red]

Line color for second mesh corresponding to difference maps 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 maps.

-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.

--pocketContactsLigandHydrophobicColor <text> [default: purpleblue]

Color for drawing hydrophobic contacts between ligand and pocket residues. The specified value must be valid color. No validation is performed. The hydrophobic contacts are shown between pairs of carbon atoms not connected to hydrogen bond donor or acceptors atoms as identified by PyMOL.

--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.

--pocketContactsCutoff <number> [default: 4.0]

Distance in Angstroms for identifying polar and hyrdophobic contacts between atoms in pocket residues and ligands.

--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.

--selectionsChain <ObjectName,SelectionSpec,...> [default: None]

Custom selections for chains. It is a pairwise of list comma delimited values corresponding to PyMOL object names and selection specifications. The selection specification must be a valid PyMOL specification. No validation is performed.

The PyMOL objects are created for each chain corresponding to the specified selections. The display style for PyMOL objects is set using value of '--selectionsChainStyle' option.

The specified selection specification is automatically appended to appropriate chain specification before

creating PyMOL objects.

For example, the following specification for '--selectionsChain' option will generate PyMOL objects for chains containing Cysteines and Serines:

```
Cysteines,resn CYS,Serines,resn SER
```

--selectionsChainStyle <DisplayStyle> [default: sticks]

Display style for PyMOL objects created for '--selectionsChain' option. It must be a valid PyMOL display style. No validation is performed.

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

Create surfaces for complete complex in each input file using corresponding 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 each input file using corresponding 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.

--volumeCarveRadius <number> [default: 1.6]

Radius in Angstroms around atoms for including electron density during generation of volume objects.

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

Create volumes for complete complex in input file using corresponding composite and difference maps. A total of two volumes, one each for composite and difference maps, are created for the complete complex.

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

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

--volumeColorRampCompositeMap <text> [default: 2fofc]

Name of volume color ramp for composite maps. The specified value must be a valid name. No validation is performed. The following volume color ramps are currently available in PyMOL: default, 2fofc, fofc, rainbow, and rainbow2.

--volumeColorRampDiffMap <text> [default: fofc]

Name of volume color ramp for difference maps. The specified value must be a valid name. No validation is performed. The following volume color ramps are currently available in PyMOL: default, 2fofc, fofc, rainbow, and rainbow2.

--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 file by using default map files, and generate a PML file, type:

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

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

```
% PyMOLVisualizeElectronDensity.py -i Sample3.pdb -o Sample3.pml
  --selectionsChain "Cysteines,resn cys,Serines,resn ser"
```

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 file by using default 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, volumes, and surfaces for complete complex and individual chains, in a PDB file by using default map files, and generate a PML file, type:

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

To visualize electron density for ligand ADP in chain E along with ligand binding pocket, in a PDB file by using default 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"
```

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

```
% PyMOLVisualizeElectronDensity.py -i Sample3.pdb -o Sample3.pml
  --chainIDs All --ligandIDs All --EDMapFiles "Sample3.ccp4,
  Sample3_diff.ccp4"
```

To align and visualize electron density for all ligands in all chains along with their binding pockets in PDB files by using explicit file names for map files, and generate a PML file, type:

```
% PyMOLVisualizeElectronDensity.py -a yes -i "Sample3.pdb,Sample4.pdb"
  -o SampleOut.pml --chainIDs All --ligandIDs All --EDMapFiles
 "Sample3.ccp4,Sample3_diff.ccp4,Sample4.ccp4,Sample4_diff.ccp4"
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

DownloadPDBFiles.pl, PyMOLVisualizeCavities.py, PyMOLVisualizeCryoEMDensity.py, PyMOLVisualizeInterfaces.py, PyMOLVisualizeMacromolecules.py, PyMOLVisualizeSurfaceAndBuriedResidues.py

COPYRIGHT

Copyright (C) 2026 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.