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>] [--pocketContactsLigandHydrophobicColor <text>] [--pocketContactsSolventColor <text>] [--pocketContactsCutoff <number>] [--pocketDistanceCutoff <number>] [--pocketLabelColor <text>] [--pocketResidueTypes <yes or no>] [--pocketSurface <yes or no>] [--pocketSurfaceElectrostatics <yes or no>] [--residueTypes <Type,Color,ResNames,...>] [--residueTypesChain <yes or no>] [--selectionsChain <ObjectName,SelectionSpec,...>] [--selectionsChainSurface <yes or no>] [--selectionsChainStyle <DisplayStyle>] [--surfaceChain <yes or no>] [--surfaceChainElectrostatics <yes or no>] [--surfaceChainComplex <yes or no>] [--surfaceComplex <yes or no>] [--surfaceColor <ColorName>] [--surfaceColorPalette <RedToWhite or WhiteToGreen>] [--surfaceAtomTypesColors <ColorType,ColorSpec,...>] [--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
            .Selections
                .<Name1>
                    .Selection
                    .Surface
                        .Surface
                        .Hydrophobicity
                        .Hydrophobicity_Charge
                .<Name2>
                    ... ... ..
            .Residues
                .Aromatic
                    .Residues
                    .Surface
                .Hydrophobic
                    .Residues
                    .Surface
                .Polar
                    .Residues
                    .Surface
                .Positively_Charged
```

```
                .Residues
                .Surface
            .Negatively_Charged
                .Residues
                .Surface
            .Other
                .Residues
                .Surface
        .Surface
            .Surface
            .Hydrophobicity
            .Hydrophobicity_Charge
            .Vacuum_Electrostatics
                .Contact_Potentials
                .Map
                .Legend
                .Volume
.Solvent
.Inorganic
.Ligand<ID>
    .Ligand
        .Ligand
        .BallAndStick
    .Pocket
        .Pocket
        .Polar_Contacts
        .Hydrophobic_Contacts
        .Residues
            .Aromatic
                .Residues
                .Surface
            .Hydrophobic
                .Residues
                .Surface
            .Polar
                .Residues
                .Surface
            .Positively_Charged
                .Residues
                .Surface
            .Negatively_Charged
                .Residues
                .Surface
            .Other
                .Residues
                .Surface
        .Surfaces
            .Surface
                .Surface
                .Hydrophobicity
                .Hydrophobicity_Charge
                .Vacuum_Electrostatics
                    .Contact_Potentials
                    .Map
                    .Legend
            .Cavity
                .Surface
                .Hydrophobicity
                .Hydrophobicity_Charge
                .Vacuum_Electrostatics
                    .Contact_Potentials
                    .Map
                    .Legend
    .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>
            ... ... ...
    <PDBFileRoot>
        .Complex
            ... ... ...
        .Chain<ID>
            ... ... ...
            .Ligand<ID>
                ... ... ...
            .Ligand<ID>
                ... ... ...
        .Chain<ID>
            ... ... ...
```

The hydrophobic and electrostatic surfaces are not created for complete complex and chain complex in input file(s) 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 each input file.

-e, --examples

Print examples.

-h, --help

Print this help message.

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

Input file names.

-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 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.

-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.

--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.

--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.

--pocketResidueTypes <yes or no> [default: auto]

Pocket residue types. The residue groups are generated using residue types, colors, and names specified by '--residueTypes' option. It is only valid for amino acids. By default, the residue type groups are

automatically created for pockets containing amino acids and skipped for chains only containing nucleic acids.

--pocketSurface <yes or no> [default: auto]

Surfaces around pocket residues colored by hydrophobicity alone and both hydrophobicity and charge. The hydrophobicity surface is colored at residue level using Eisenberg hydrophobicity scale for residues and color gradient specified by '--surfaceColorPalette' option. The hydrophobicity and charge surface is colored [Ref 140] at atom level using colors specified for groups of atoms by '--surfaceAtomTypesColors' option. This scheme allows simultaneous mapping of hyrophobicity and charge values on the surfaces.

The cavity surfaces around ligands are also generated. These surfaces are colored by hydrophobicity along and both hydrophobicity and charge.

This option is only valid for amino acids. By default, both surfaces are automatically created for pockets containing amino acids and skipped for pockets containing only nucleic acids.

In addition, generic pockect and cavity surfaces colored by '--surfaceColor' are always created for pocket residues containing amino acids and nucleic acids.

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

Vacuum electrostatics contact potential surface around pocket residues. A word 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".

The cavity surface around ligands is also generated. This surface is colored by vacuum electrostatics contact potential.

This option is only valid for amino acids. By default, the electrostatics surface is automatically created for chains containing amino acids and skipped for chains containing only nucleic acids.

-r, --residueTypes <Type,Color,ResNames,...> [default: auto]

Residue types, colors, and names to generate for residue groups during '--pocketResidueTypes' and '--residueTypesChain' option. It is only valid for amino acids.

It is a triplet of comma delimited list of amino acid residues type, residues color, and a space delimited list three letter residue names.

The default values for residue type, color, and name triplets are shown below:

```
Aromatic,brightorange,HIS PHE TRP TYR,
Hydrophobic,orange,ALA GLY VAL LEU ILE PRO MET,
Polar,palegreen,ASN GLN SER THR CYS,
Positively_Charged,marine,ARG LYS,
Negatively_Charged,red,ASP GLU
```

The color name must be a valid PyMOL name. No validation is performed. An amino acid name may appear across multiple residue types. All other residues are grouped under 'Other'.

--residueTypesChain <yes or no> [default: auto]

Chain residue types. The residue groups are generated using residue types, colors, and names specified by '--residueTypes' option. It is only valid for amino acids. By default, the residue type groups are automatically created for chains containing amino acids and skipped for chains only containing nucleic acids.

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

Custom selections for chains. It is a pairwise list of 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
```

--selectionsChainSurface <yes or no> [default: auto]

Surfaces around individual chain selections colored by hydrophobicity alone and both hydrophobicity and charge. This option is similar to '--surafceChain' options for creating surfaces for chain. Additional details are available in the documentation section for '--surfaceChain' options.

--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.

--surfaceChain <yes or no> [default: auto]

Surfaces around individual chain colored by hydrophobicity alone and both hydrophobicity and charge. The hydrophobicity surface is colored at residue level using Eisenberg hydrophobicity scale for residues and color gradient specified by '--surfaceColorPalette' option. The hydrophobicity and charge surface is colored [Ref 140] at atom level using colors specified for groups of atoms by '--surfaceAtomTypesColors' option. This scheme allows simultaneous mapping of hyrophobicity and charge values on the surfaces.

This option is only valid for amino acids. By default, both surfaces are automatically created for chains containing amino acids and skipped for chains containing only nucleic acids.

In addition, generic surfaces colored by '--surfaceColor' are always created for chain residues containing amino acids and nucleic acids.

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

Vacuum electrostatics contact potential surface and volume around individual chain. A word 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".

This option is only valid for amino acids. By default, the electrostatics surface and volume are automatically created for chains containing amino acids and skipped for chains containing only nucleic acids.

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

Hydrophobic surface around chain complex. The surface is colored by hydrophobicity. It is only valid for amino acids.

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

Hydrophobic surface around complete complex. The surface is colored by hydrophobicity. It is only valid for amino acids.

--surfaceAtomTypesColors <ColorType,ColorSpec,...> [default: auto]

Atom colors for generating surfaces colored by hyrophobicity and charge around chains and pockets in proteins. It's a pairwise comma delimited list of atom color type and color specification for goups of atoms.

The default values for color types [Ref 140] along wth color specifications are shown below:

```
HydrophobicAtomsColor, yellow,
NegativelyChargedAtomsColor, red,
PositivelyChargedAtomsColor, blue,
OtherAtomsColor, gray90
```

The color names must be valid PyMOL names.

The color values may also be specified as space delimited RGB triplets:

```
HydrophobicAtomsColor, 0.95 0.78 0.0,
NegativelyChargedAtomsColor, 1.0 0.4 0.4,
PositivelyChargedAtomsColor, 0.2 0.5 0.8,
OtherAtomsColor, 0.95 0.95 0.95
```

--surfaceColor <ColorName> [default: lightblue]

Color name for surfaces around chains and pockets. This color is not used for surfaces colored by hydrophobicity and charge. The color name must be a valid PyMOL name.

--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.

--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 resiudes, solvents and inorganics, in a PDB file, and generate a PML file, type:

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

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

```
% PyMOLVisualizeMacromolecules.py -i Sample4.pdb -o Sample4.pml
  --selectionsChain "Cysteines,resn cys,Serines,resn ser"
```

To visualize all chains, all ligands in all chains, and all ligand binding pockets to highlight ligand interaction with pocket resiudes, 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 hydrophibic surfaces and chain electrostatic surface, in a PDB file, and generate a PML file, type:

```
% PyMOLVisualizeMacromolecules.py -c All -l All
  --surfaceChainElectrostatics 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 resiudes, 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 resiudes, 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 resiudes, 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 resiudes, 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 resiudes, 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, PyMOLVisualizeCavities.py, PyMOLVisualizeCryoEMDensity.py, PyMOLVisualizeElectronDensity.py, PyMOLVisualizeInterfaces.py, PyMOLVisualizeSurfaceAndBuriedResidues.py

COPYRIGHT