NAME

    PyMOLVisualizeCavities.py - Visualize cavities and pockets in macromolecules

SYNOPSIS

    PyMOLVisualizeCavities.py [--align <yes or no>] [--alignMethod <align, cealign, super>] [--alignMode
    <FirstChain or Complex>] [--alignRefFile <filename>] [--allowEmptyObjects <yes or no>] [--cavityCulled
    <Yes or No>] [--cavityCullSize <number>] [--cavityCutoff <number>] [--cavityRadius <number>] [
    --chainIDs <First, All or ID1,ID2...>] [--labelFontID <number>] [--ignoreHydrogens <yes or no>] [
    --ligandIDs <Largest, All or ID1,ID2...>] [--PMLOut <yes or no>] [--residueTypes
    <Type,Color,ResNames,...>] [--residueTypesChain <yes or no>] [--surfaceChain <yes or no>] [
    --surfaceChainElectrostatics <yes or no>] [--surfaceChainComplex <yes or no>] [--surfaceComplex <yes
    or no>] [--surfaceAtomTypesColors <ColorType,ColorSpec,...>] [--surfaceColors <ColorName1,ColorName2>]
    [--surfaceColorPalette <RedToWhite or WhiteToGreen>] [--surfaceTransparency <number>] [--overwrite] [
    -w <dir>] -i <infile1,infile2,infile3...> -o <outfile>

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

DESCRIPTION

    Generate PyMOL visualization files for viewing cavities and 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 cavities and pockets in macromolecules.
    These groups and objects correspond to complexes, surfaces, chains, ligands, inorganics, cavities, and pockets. A
    complete hierarchy of all possible PyMOL groups and objects is shown below:

        <PDBFileRoot>
            .Complex
                .Complex
                .Surface
            .Chain<ID>
                .Complex
                    .Complex
                    .Surface
                .Chain
                    .Chain
                    .Residues
                        .Aromatic
                            .Residues
                            .Surface
                        .Hydrophobic
                            .Residues
                            .Surface
                        .Polar
                            .Residues
                            .Surface
                        .Positively_Charged
                            .Residues
                            .Surface
                        .Negatively_Charged
                            .Residues
                            .Surface
                        .Other
                            .Residues
                            .Surface
                .Cavities
                    .Surface
                    .Hydrophobicity
                    .Hydrophobicity_Charge
                    .Vacuum_Electrostatics
                        .Contact_Potentials

```
                                        .Map
                                        .Legend
                            .Surface
                                .Surface
                                .Hydrophobicity
                                .Hydrophobicity_Charge
                                .Vacuum_Electrostatics
                                    .Contact_Potentials
                                    .Map
                                    .Legend
                    .Solvent
                    .Inorganic
                    .Ligand<ID>
                        .Ligand
                            .Ligand
                            .BallAndStick
                    .Ligand<ID>
                        .Ligand
                            ... ... ...
                .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.

--cavityCulled <Yes or No> [default: Yes]

Cull cavities and pockets. The cavities and pockets are culled by default. This value is used to set of PyMOL parameter surface_cavity_mode as shown below:

```
Yes: 1 (Cavities and Pockets Only)
No: 2 (Cavities and Pockets Culled)
```

--cavityCullSize <number> [default: 2.0]

Approximate dimension of the cavity in Angstroms for detecting cavities and pockets in the interior of a macromolecule. The higher value makes PyMOL less sensitive to detection of smaller cavities.

--cavityCutoff <number> [default: 3.0]

Cavity cutoff in terms of number of solvent radii for detecting cavities and pockets. This value is used to set value of PyMOL parameter surface_cavity_cutoff.

--cavityRadius <number> [default: 5.0]

Cavity detection radius in terms of number of solvent radii for detecting cavities and pockets. The detection of larger pockets is ignored for lower value for the cavity radius. This value is used to set of PyMOL parameter surface_cavity_radius.

-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.

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

Ignore hydrogens for ligand views.

--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.

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

List of ligand IDs to show in chains during visualizing of cavities in macromolecules. Possible values: Largest, All, or a comma delimited list of ligand IDs. The default is to show all ligands 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.

-o, --outfile <outfile>

Output file name.

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

Save PML file during generation of PSE file.

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

Residue types, colors, and names to generate for residue groups during '--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.

--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 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 is 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
```

--surfaceColors <ColorName1,ColorName2> [default: salmon,lightblue]

>   Color names for surfaces around cavities and chains. These colors are not used for surfaces and cavities colored by hydrophobicity and charge. The color names must be valid PyMOL names.

--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 cavities in the first chain along with the largest ligand in the first chain, solvents, and inorganics, in a PDB file, and generate a PML file, type:

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

>   To visualize cavities in all chain along with all ligands, solvents, and inorganics, in a PDB file, and generate a PML file, type:

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

>   To visualize cavities in the first chain at a specific cavity radius and cutoff using specifc colors for surfaces corresponding to cavities and non-cavities, and generate a PML file, type:

```
% PyMOLVisualizeCavities.py  --cavityRadius 3 --cavityCutoff 5
   --surfaceColors "red,blue" -i Sample4.pdb -o Sample4.pml
```

>   To visualize cavities in the first chain along with the largest ligand in the first chain, solvents, and inorganics, in PDB files, along with aligning first chain in each input file to the first chain inand generate a PML file, type:

```
% PyMOLVisualizeCavities.py --align yes
  -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, PyMOLVisualizeInterfaces.py PyMOLVisualizeMacromolecules.py, PyMOLVisualizeSurfaceAndBuriedResidues.py

COPYRIGHT

>   Copyright (C) 2022 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.