NAME

       PyMOLVisualizeInterfaces.py - Visualize macromolecular interfaces

SYNOPSIS

       PyMOLVisualizeInterfaces.py [--allowEmptyObjects <yes or no>] [--chainIDs <ChainID1 or ChainID1,ChainID2>] [--interfaceLabelColor <text>] [--interfaceContactsCutoff <number>] [--interfaceHydrophobicContacts <yes or no>] [--interfaceHydrophobicContactsColor <text>] [--interfacePolarContacts <yes or no>] [--interfacePolarContactsColor <text>] [--interfaceResidueTypes <yes or no>] [--interfaceSurface <yes or no>] [--interfaceSurfaceElectrostatics <yes or no>] [--labelFontID <number>] [--ligandIDs <Largest, All, None or ID1,ID2...>] [--method <text>] [--methodCutoff <number>] [--PMLOut <yes or no>] [--residueTypes <Type,Color,ResNames,...>] [--surfaceChain <yes or no>] [--surfaceChainElectrostatics <yes or no>] [--surfaceChainComplex <yes or no>] [--surfaceComplex <yes or no>] [--surfaceColors <ColorName1,ColorName2>] [--surfaceColorPalette <RedToWhite or WhiteToGreen>] [--surfaceAtomTypesColors <ColorType,ColorSpec,...>] [--surfaceTransparency <number>] [--overwrite] [-w <dir>] -i <infile1,...> -o <outfile>

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

DESCRIPTION

       Generate PyMOL visualization files for viewing interfaces between macromolecules including proteins and nucleic acids. The interfaces may be generated between pairs of chains in a single file or across two different files.

       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 macromolecular interfaces. These groups and objects correspond to complexes, surfaces, chains, ligands, and interfaces. A complete hierarchy of all possible PyMOL groups and objects is shown below:

```
<PDBFileRoot>
    .Complex
        .Complex
        .Surface
    .Chain<ID>
        .Complex
            .Complex
            .Surface
        .Chain
            .Chain
            .NonInterface
                .Chain
                .Surface
                    .Surface
                    .Hydrophobicity
                    .Hydrophobicity_Charge
                    .Vacuum_Electrostatics
                        .Contact_Potentials
                        .Map
                        .Legend
                        .Volume
        .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>
            ... ... ...
    <Interfaces>
        .Chain<IDs1>_Chain<IDs2>
            .Polar_Contacts
            .Hydrophobic_Contacts
            .Chain<ID> or Chain<ID>_<PDBFileRoot>
                .Chain
                .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
            .Chain<ID> or <PDBFileRoot>_Chain<ID>
                .Chain
                .Residues
                    ... ... ...
                .Surface
                    ... ... ...
        .Chain<IDs>_Chain<IDs>
            .Polar_Contacts
            .Hydrophobic_Contacts
            .Chain<ID> or Chain<ID>_<PDBFileRoot>
                .Chain
                .Residues
                    ... ... ...
                .Surface
                    ... ... ...
```

```
        .Chain<ID> or Chain<ID>_<PDBFileRoot>
            .Chain
            .Residues
                ... ... ...
            .Surface
                ... ... ...
```

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

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

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

-c, --chainIDs <ChainID1,ChainD2,...> [default: Auto]

Pairwise comma delimited list of chain IDs for the identification of macromolecular interfaces. All chain IDs must be present in the same file for a single input file. Otherwise, the first and second chain ID(s) in a pair belong to the first and second input file.

The default values for interface chain IDs depend on the number of input files as shown below:

One input file: First two chains Two input files: First chain in each input file

Each chain may contain multiple chain IDs delimited by a plus sign. For example, A+B,C+D chain pair specifies interface between chain complexes A+B and C+D in first input file or across two input files.

-e, --examples

Print examples.

-h, --help

Print this help message.

-i, --infiles <infile or infile1,infile2>

Name of an input file or a comma delmited list of names for two input files.

--interfaceLabelColor <text> [default: magenta]

Color for drawing residue or atom level labels for residues in an interface. The specified value must be valid color. No validation is performed.

--interfaceContactsCutoff <number> [default: 4.0]

Distance in Angstroms for identifying polar and hyrdophobic contacts between atoms in interface reisudes.

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

Hydrophobic contacts between residues in an interface. The hydrophobic contacts are shown between pairs of carbon atoms not connected to hydrogen bond donor or acceptors atoms as identified by PyMOL.

--interfaceHydrophobicContactsColor <text> [default: purpleblue]

Color for drawing hydrophobic contacts between residues in an interface. The specified value must be valid color. No validation is performed.

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

Polar contacts between residues in an interface.

--interfacePolarContactsColor <text> [default: orange]

Color for drawing polar contacts between residues in an interface. The specified value must be valid color. No validation is performed.

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

Interface 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 interfaces containing amino acids and skipped for chains only containing nucleic acids.

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

Surfaces around interface 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.

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 surfaces colored by '--surfaceColors' are always created for interface residues containing amino acids and nucleic acids.

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

Vacuum electrostatics contact potential surface around interface 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".

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.

--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, None or ID1,ID2...> [default: All]

List of ligand IDs to show in chains during visualization of interfaces. Possible values: Largest, All, None, or a comma delimited list of ligand IDs. The default is to show all ligands present in chains involved in interfaces.

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.

-m, --method <text> [default: BySASAChange]

Methodology for the identification of interface residues between a pair of chains in an input file. The interface residues may be identified by change in solvent accessible surface area (SASA) for a residue between a chain and chains complex, distance between heavy atoms in two chains, or distance between CAlpha atoms. Possible values: BySASAChange, ByHeavyAtomsDistance, or ByCAlphaAtomsDistance.

--methodCutoff <number> [default: auto]

Cutoff value used by different methodologies during the identification of interface residues between a pair of chains. The default values are shown below:

```
BySASAChange: 1.0; Units: Angstrom**2 [ Ref 141 ]
ByHeavyAtomsDistance: 5.0; Units: Angstrom [ Ref 142 ]
ByCAlphaAtomsDistance: 8.0; Units: Angstrom [ Ref 143 ]
```

-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 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'.

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

Surfaces around non-interface residues in 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 '--surfaceColors' are always created for non-interface residues containing amino acids and nucleic acids.

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

Vacuum electrostatics contact potential surface and volume around non-interface residues in 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.

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

Color names for surfaces around interface residues and non-interface residues in chains. These colors are not used for surfaces 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 interface residues 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.

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

Atom colors for generating surfaces colored by hyrophobicity and charge around chains and interface residues 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
```

--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 interface residues between the first two chains in a PDB file, using default methodology to identify interfaces, and and generate a PML file, type:

```
% PyMOLVisualizeInterfaces.py -i Sample8.pdb -o Sample8.pml
```

To visualize interface residues between a pair of specific chains in a PDB file using a specific methodology and cutoff value to identify interfaces, and generate a PML file, type:

```
% PyMOLVisualizeInterfaces.py -m BySASAChange --methodCutoff 1.0
-c "A,B" -i Sample8.pdb -o Sample8.pml
```

To visualize interface residues between multiple pairs of specified chains in a PDB file using a specific methodology and cutoff value to identify interfaces, and generate a PML file, type:

```
% PyMOLVisualizeInterfaces.py -m ByHeavyAtomsDistance
--methodCutoff 5.0 -c "A,B,B,D" -i Sample8.pdb -o Sample8.pml
```

To visualize interface residues between a pair of specified chains, each member containing multiple chains, a PDB file using a specific methodology and cutoff value to identify interfaces, and generate a PML file, type:

```
% PyMOLVisualizeInterfaces.py -m ByCAlphaAtomsDistance
--methodCutoff 8.0 -c "A+C,B+D" -i Sample8.pdb -o Sample8.pml
```

To visualize interface residues between a pair of specific chains across two PDB files using a specific methodology and cutoff value to identify interfaces, and generate a PML file, type:

```
% PyMOLVisualizeInterfaces.py -m BySASAChange --methodCutoff 1.0
-c "A,B" -i Sample8Part1.pdb,Sample8Part2.pdb
-o Sample8.pml
```

To visualize interface residues between multiple pairs of specified chains across two PDB files using a specific methodology and cutoff value to identify interfaces, and generate a PML file, type:

```
% PyMOLVisualizeInterfaces.py -m ByHeavyAtomsDistance
--methodCutoff 5.0  -c "A,B,C,B" -i Sample8Part1.pdb,Sample8Part2.pdb
-o Sample8.pml
```

To visualize interface residues between a pair of specified chains, each member containing multiple chains, across two PDB files using a specific methodology and cutoff value to identify interfaces, and generate a PML file, type:

```
% PyMOLVisualizeInterfaces.py -m ByCAlphaAtomsDistance
--methodCutoff 8.0  -c "A+C,B+D" -i "Sample8Part1.pdb,Sample8Part2.pdb"
-o Sample8.pml
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

DownloadPDBFiles.pl, PyMOLVisualizeCryoEMDensity.py, PyMOLVisualizeElectronDensity.py,
PyMOLVisualizeMacromolecules.py, PyMOLVisualizeSurfaceAndBuriedResidues.py

COPYRIGHT

Copyright (C) 2020 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.