NAME

PyMOLVisualizeFpockets.py - Visualize fpockets for macromolecules.

SYNOPSIS

PyMOLVisualizeFpockets.py [--align <yes or no>] [--alignMethod <align, cealign, super>] [--alignMode <FirstChain or Complex>] [--alignRefFile <filename>] [--chainIDs <First, All or ID1,ID2...>] [--fpocketMode <All, TopN, or Specify>] [--fpocketIDs <Value or Value1,Value2...>] [--fpocketPropertiesAppend <yes or no>] [--labelFontID <number>] [--outfilesDir <outfilesDir>] [--pocketColorByPocketNum <yes or no>] [--pocketLabel <yes or no>] [--pocketLabelType <OneLetter or ThreeLetter>] [--pocketSurface <yes or no>] [--sphereScale <number>] [--sphereTransparency <number>] [--surfaceChain <yes or no>] [--surfaceAtomTypesColors <ColorType,ColorSpec,...>] [--surfaceColor <ColorName>] [--surfaceColorPalette <RedToWhite or WhiteToGreen>] [--surfaceTransparency <number>] [--overwrite] [-w <dir>] -i <infile1,infile2,infile3...> -o <outfile>

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

DESCRIPTION

Generate a PyMOL visualization file for visualizing pockets in macromolecules detected by an open source package named Fpocket [Ref 166].

The results of Fpocket calculations must be available in the current directory for all input files. A complete set of expected results is shown below:

```
Dir: <PDBFileRoot>_out
    <PDBFileRoot>_out.pdb
       ... .. ...
    Dir: pockets
       <PDBFileRoot><PocketID>_atm.pdb
       ... .. ...
```

The supported input file format is: PDB (.pdb)

The supported output file formats is: PyMOL script file (.pml)

The following directory and files are created for the visualization of pockets detected by Fpocket:

```
Dir: <OutFileRoot>_out_PyMOL or <OutfilesDir>
    <OutfileRoot>.pml
    <PDBFileRoot>.pdb
    <PDBFileRoot>_out.pdb
```

You may visualize pockets in PyMOL by loading <OutfileRoot>.pml from <OutfileRoot>_out_PyMOL or <OutfilesDir> directory.

A variety of PyMOL groups and objects may be created for visualization of fpockets in macromolecules. These groups and objects correspond to complexes, chains, fpockets, and surfaces. A complete hierarchy of all possible PyMOL groups and objects is shown below:

```
<PDBFileRoot>
    .Complex
        .Initial_PDB
        .Fpocket_PDB
    .Chain<ID>
        .Complex
            .Complex
        .Chain
            .Chain
            .Surface
                .Surface
                .Hydrophobicity
                .Hydrophobicity_Charge
        .FPocket<ID>
            .FPocket
            .Residues
            .Surface
                .Surface
                .Hydrophobicity
                .Hydrophobicity_Charge
```

```
            .Chain<ID>
                ... ... ...
              .FPocket<ID>
                ... ... ...
              .FPocket<ID>
                ... ... ...
            .Chain<ID>
                ... ... ...
        <PDBFileRoot>
            .Complex
                ... ... ...
            .Chain<ID>
                ... ... ...
              .FPocket<ID>
                ... ... ...
              .FPocket<ID>
                ... ... ...
            .Chain<ID>
                ... ... ...
```

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.

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

List of chain IDs to use for visualizing fpockets in 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.

-f, --fpocketMode <All, TopN, or Specify> [default: All]

Fpockets specification mode for visualizing fpockets across chains in macromolecules. Possible values: All, TopN, or specify. By default, all available fpockets are visualized across specified chains.

The fpocket IDs must be specified using '--fpocketIDs' option for 'TopN' and 'Specifiy' value of '--fpocketMode '.

--fpocketIDs <Value or Value1,Value2...>

List of Fpocket IDs for visualizing fpockets across chains. This value is dependent on the value of '--fpocketMode'. The possible values are either a number or a comma delimited list of number for 'TopN' and 'Specify' value '--fpocketMode' option. For example:

This option is ignored during 'All' value of '--fpocketMode' option.

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

Append fpocket properties to names of PyMOL fpocket groups and objects. The following properties are appended to the names of PyMOL groups using their abbreviations and values: PocketScore - S; DrugScore - D; PocketVolume - V; HydrophobicityScore - H; PolarityScore - P. For example:

```
Fpocket1_S0p50_D0p00_V638p81_Hneg6p67_P11p00.Fpocket
```

-h, --help

Print this help message.

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

Input PDB file names. The current directory must contain the results from Fpocket calculations for all the input PDB files.

--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>

PML output file name for visualizing fpockets. The PML outfile is created in a new output directory named <OutfileRoot>_out_PyMOL. In addition, the output directory contains all appropriate PDB files generated by Fpocket for visualization of fpockets in PyMOL.

--outfilesDir <outfilesDir> [default: auto]

Output files directory name. Default: <OutfileRoot>_out_PyMOL.

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

Color fpocket residues and residue labels by pocket number. Otherwise, the pocket residues are colored by element names using default PyMOL color scheme. No color is set for residue labels.

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

Display residue labels on fpocket residues. The residue number is always appended to residue label. You may specify a one or thee letter residue labels using '--pocketLabelType' option.

--pocketLabelType <OneLetter or ThreeLetter> [default: OneLetter]

Display one or three letter residue labels on fpocket residues

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

Surfaces around fpocket 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 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.

In addition, generic surfaces colored by '--surfaceColor' are always created for pockets.

--sphereScale <number> [default: 0.4]

Scaling factor for spheres used to display fpocket alpha spheres.

--sphereTransparency <number> [default: 0.25]

Transparency for spheres used to display fpocket alpha spheres.

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

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 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.

In addition, generic surfaces colored by '--surfaceColor' are always created for chains.

--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 groups of atoms.

The default values for color types 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 all fpockets available in a directory <PDBRoot>_out for the first chain, along pocket residues and surfaces, in a PDB file, and generate a PML file in a new directory <PDBRoot>_out_pymol, type:

```
% PyMOLVisualizeFpockets.py -i Sample5.pdb -o Sample5.pml
```

To rerun the first example without displaying pocket residue labels, coloring pockets by element type, and write out a PML file, type:

```
% PyMOLVisualizeFpockets.py --pocketColorByPocketNum no
  --pocketLabel no -i Sample5.pdb -o Sample5.pml
```

To rerun the first example to visualize only top 5 fpockets and write out a PML file, type:

```
% PyMOLVisualizeFpockets.py -f TopN --fpocketIDs 5 -i Sample5.pdb
  -o Sample5.pml
```

To rerun the first example to visualize a specific set of fpockets and write out a PML file, type:

```
% PyMOLVisualizeFpockets.py -f Specify --fpocketIDs "1,2,3" -i Sample5.pdb
  -o Sample5.pml
```

To rerun the first example to visualize all fpockets across all chains and write out a PML file, type:

```
% PyMOLVisualizeFpockets.py -c All -f All -i Sample5.pdb -o Sample5.pml
```

To rerun the first example without displaying hydrophobic and charge surfaces around chain and pockets and and write out a PML file, type:

```
% PyMOLVisualizeFpockets.py --surfaceChain no  --pocketSurface no
  -i Sample5.pdb -o Sample5.pml
```

To visualize top 5 fpockets available in a directories <PDBRoot>_out for the first chain, along pocket residues and surfaces, in PDB files, aligning first fchain in each input file to the first chain in first input file, and generate a PML file in a new directory <PDBRoot>_out_pymol, type:

```
% PyMOLVisualizeFpockets.py -f TopN --fpocketIDs 5 --align yes
  -i "Sample5.pdb,Sample6.pdb" -o Sample5Aligned.pml
```

To visualize top 5 fpockets available in a directories <PDBRoot>_out for the first chain, along pocket residues and surfaces, in PDB files, aligning first chain in each input file to the first chain in first chain in a specified PDB file using a specified alignment method,, and generate a PML file in a new directory <PDBRoot>_out_pymol, type:

```
% PyMOLVisualizeFpockets.py -f TopN --fpocketIDs 5 --align yes
  --alignMode FirstChain --alignRefFile Sample6.pdb --alignMethod super
  -i "Sample5.pdb,Sample6.pdb" -o Sample5Aligned.pml
```

AUTHOR

Manish Sud

COLLABORATORS

Joann Prescott-Roy and Pat Walters

SEE ALSO

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

COPYRIGHT