

**NAME**

PyMOLInfoMacromolecules.py - List information about macromolecules

**SYNOPSIS**

```
PyMOLInfoMacromolecules.py [--all] [--boundingBox] [--centroid] [--chains] [--countResidues] [
--header] [--inorganics] [--interfaceResidues] [--interfaceResiduesChains <ChainID1,ChainID2,...>] [
--interfaceResiduesMethod <text>] [--interfaceResiduesCutoff <number>] [--ligands] [
--pocketLigands] [--pocketDistanceCutoff <number>] [--pocketSolvents] [--pocketInorganics] [
--phiPsi] [--phiPsiMode <All or Categories>] [--phiPsiPrecision <number>] [--surfaceResidues] [
--surfaceResiduesCutoff <number>] [--surfaceResiduesIDs <yes or no>] [--solvents] [-w <dir>] -i
<infile1,infile2,infile3...>
```

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

**DESCRIPTION**

List information regarding ID, classification, experimental technique, chains, solvents, inorganics, ligands, and ligand binding pockets in macromolecules present including proteins and nucleic acids.

The supported input file format are: PDB (.pdb), mmCIF (.cif)

**OPTIONS**

-a, --all

All available information.

-b, --boundingBox

Min and max coordinates for bounding box along with its size.

-c, --chains

Number of chains and their IDs. This is also default behavior. --centroid Centroid of atomic coordinates. It corresponds to the mean of all 3D coordinates in input file. --countResidues Number of residues across chains. The chain residues are identified using polymer selection operator available in PyMOL. In addition, the non-standard amino acid residues are listed.

-e, --examples

Print examples.

-h, --help

Print this help message. --header Header information including experimental technique information along with any available resolution. This is also default behavior.

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

A comma delimited list of input files. The wildcards are also allowed in file names.

--inorganics

Inorganic residues across chains. The inorganic residues are identified using inorganic selection operator available in PyMOL.

--interfaceResidues

Interface residues between specified pairs of chains.

--interfaceResiduesChains <ChainID1,ChainID2,...> [default: Auto]

Pairwise comma delimited list of chain IDs for the identification of interface residues. Each chain ID 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.

The interface residues are identified between first two chains in input files by default.

--interfaceResiduesMethod <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 ByCAAlphaAtomsDistance.

--interfaceResiduesCutoff <number> [default: auto]

Cutoff value used by different methodologies during identification of interface residues between a pair

of chains. The default values are shown below:

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

-l, --ligands

Ligands across chains. This is also default behavior. The ligands residues are identified using organic selection operator available in PyMOL.

-p, --pocketLigands

Chain residues in ligand pockets.

--pocketDistanceCutoff <number> [default: 5.0]

Distance in Angstroms for identifying pocket residues around ligands.

--pocketSolvents

Solvent residues in ligand pockets. The solvent residues are identified using solvent selection operator available in PyMOL.

--pocketInorganics

Inorganic residues in ligand pockets. The inorganic residues are identified using Inorganic selection operator available in PyMOL.

--phiPsi

Phi and psi torsion angles across chains in macromolecules containing amino acids.

--phiPsiMode <All or Categories> [default: Categories]

List all phi and psi torsion angles for residues as a single group or split them into the following categories corresponding to four types of Ramachandran plots:

```
General: All residues except glycine, proline, or pre-proline
Glycine: Only glycine residues
Proline: Only proline residues
Pre-Proline: Only residues before proline not including glycine
or proline
```

--phiPsiPrecision <number> [default: 2]

Precision for listing phi and psi torsion angles.

-s, --solvents

Solvent residues across chains. The solvent residues are identified using solvent selection operator available in PyMOL.

--surfaceResidues

Surface and buried residues in chains.

--surfaceResiduesCutoff <number> [default: 2.5]

Solvent Accessible Surface Area (SASA) cutoff value in Angstroms\*\*2 for surface and buried residues in chains. The residues with SASA less than the cutoff value correspond to buried residues.

--surfaceResidueIDs <yes or no> [default: No]

List residue IDs for surface and buried residues during listing of the distribution of these residues for '--surfaceResidues' option.

-w, --workingdir <dir>

Location of working directory which defaults to the current directory.

## EXAMPLES

To list header, chains, and ligand information for macromolecules in input file, type:

```
% PyMOLInfoMacromolecules.py -i Sample3.pdb
```

To list all available information for macromolecules in input files, type:

```
% PyMOLInfoMacromolecules.py -a -i "Sample3.pdb,Sample4.pdb"
```

To list pockets residues information along with other default information for marcomolecules in input file, type:

```
% PyMOLInfoMacromolecules.py -p --pocketDistanceCutoff 4.5
--pocketSolvents --pocketInorganics -i Sample3.pdb
```

To list chain residues information along with other default information for marcomolecules in input file, type:

```
% PyMOLInfoMacromolecules.py -c --countResidues --solvents
--inorganics -i "Sample3.pdb,Sample4.pdb"
```

To list interface residues between first two chains by SASA change for marcomolecules in input file, type:

```
% PyMOLInfoMacromolecules.py --interfaceResidues
-i Sample3.pdb
```

To list interface residues between chains E and I by heavy atoms distance for marcomolecules in input file, type:

```
% PyMOLInfoMacromolecules.py --interfaceResidues
--interfaceResiduesChains E,I --interfaceResiduesMethod
ByHeavyAtomsDistance --interfaceResiduesCutoff 5 -i Sample3.pdb
```

To list interface residues between two sets of chains by SASA change for marcomolecules in input file, type:

```
% PyMOLInfoMacromolecules.py --interfaceResidues
--interfaceResiduesChains "A+B,C+D" -i Sample8.pdb
```

## AUTHOR

Manish Sud(msud@san.rr.com)

## SEE ALSO

DownloadPDBFiles.pl, PyMOLSplitChainsAndLigands.py, PyMOLVisualizeMacromolecules.py

## COPYRIGHT

Copyright (C) 2025 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.