NAME

PyMOLGenerateRamachandranPlots.py - Generate Ramachandran plots

SYNOPSIS

PyMOLGenerateRamachandranPlots.py [--chainIDs <First, All or ID1,ID2...>] [--figDPI <number>] [--figSize <width, height>] [--fontFamily <text>] [--fontAxesSize <number or text>] [--fontAxesWeight <number or text>] [--fontTicksSize <number or text>] [--fontTicksWeight <number or text>] [--fontTitleSize <number or text>] [--fontTitleWeight <number or text>] [--greek <yes or no>] [--grid <yes or no>] [--gridLineColor <text>] [--gridLineStyle <text>] [--gridLineWidth <number>] [--levelsAndColors <PlotType:Level,color,Level,...;...>] [--levelsAndColorsScheme <text>] [--outMode <SingleFile or MultipleFiles>] [--overwrite] [--precision <number>] [--scatterMarkerColor <text>] [--scatterMarkerSize <number>] [--scatterMarkerStyle <text>] [--ticksMajorInterval <number>] [--ticksMinor <yes or no>] [--ticksMinorInterval <number>] [-w <dir>] -i <infile> -o <outfile>

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

DESCRIPTION

Generate Ramachandran plots for amino acid residues present in macromolecules.

The Ramachandran plots are generated by plotting phi and psi backbone angles corresponding to the following four categories of amino acids:

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

In addition to the scatter plots for phi and psi angles, the filled contours are generated for the density of phi and psi angles [Ref 144] for the Ramachandran plots. The contours are generated for "favored" and "allowed" regions. The phi and psi density is retrieved from the following density files available in MAYACHEMTOOLS/lib/data/ directory:

```
General: PhiPsiDensityGeneral.csv
Glycine: PhiPsiDensityGlycine.csv
Proline: PhiPsiDensityProline.csv
PreProline: PhiPsiDensityPreProline.csv
```

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

The output image file can be saved in any format supported by Python module Matplotlib. The image format is automatically detected from the output file extension.

Some of the most common output image file formats are: EPS (.eps), PDF (.pdf), PNG (.png), PS (.ps), SVG (.svg).

OPTIONS

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

List of chain IDs to use for calculating phi and psi angles for residues in chains. Possible values: First, All, or a comma delimited list of chain IDs. The default is to use all chain IDs in input file.

-e, --examples

Print examples.

--figDPI <number> [default: 300]

Figure resolution in dots per inches. The DPI value must be supported by Matplotlib during generation of an image of a specific format. No validation is performed.

--figSize <width, height> [default: auto]

Figure dimensions in inches. The default values are dependent on the the value of '--outMode' option as shown below:

```
SingleFile: 6.4, 6.4
```

```
MultipleFiles: 6.4, 4.8
```

--fontFamily <text> [default: auto]

Font family to use for title, axes labels, and tick marks. It must be a valid Matplotlib value. The default value corresponds to the value plt.rcParams["font.family"] in your environment. For example: serif, sans-serif, cursive, etc.

--fontAxesSize <number or text> [default: 10]

Font size for labels on axes. It must be valid Matplotlib font size. For example: size in points, xx-small, x-small, small, medium, etc.

--fontAxesWeight <number or text> [default: regular]

Font weight for labels on axes. It must be valid Matplotlib value. For example: a numeric value in range 0-1000, ultralight, light, normal, regular, book, medium, etc.

--fontTicksSize <number or text> [default: 8]

Font size for tick labels. It must be a valid Matplotlib font size. For example: size in points, xx-small, x-small, small, medium, etc.

--fontTicksWeight <number or text> [default: regular]

Font weight for tick labels. It must be valid Matplotlib value. For example: a numeric value in range 0-1000, ultralight, light, normal, regular, book, medium, etc.

--fontTitleSize <number or text> [default: 10]

Font size for title. It must be a valid Matplotlib font size. For example: size in points, xx-small, x-small, small, medium, etc.

--fontTitleWeight <number or text> [default: bold]

Font weight for title. It must be a valid Matplotlib value. For example: a numeric value in range 0-1000, ultralight, light, normal, regular, book, medium, etc.

-g, --greek <yes or no> [default: yes]

Show phi and psi labels as greek characters.

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

Display grid lines at major tick marks.

--gridLineColor <text> [default: #b0b0b0]

Grid line color. It must be a valid Matplotlib value. The default color is light gray.

--gridLineStyle <text> [default: dotted]

Grid line style. It must be a valid Matplotlib value. For example: '-' or 'solid', --' or 'dashed', '-.' or 'dashdot', ':' or 'dotted' etc.

--gridLineWidth <number> [default: 0.8]

Grid line width. It must be a valid Matplotlib value.

-h, --help

Print this help message.

-i, --infile <infile>

Input file name.

-l, --levelsAndColors <PlotType:Level,color,Level,...;...> [default: auto]

Semicolon delimited list of contour levels and colors for four types of Ramachandran plots.

Three default contour level and color scheme may be specified by '--levelsAndColorsScheme' option. By default, the 'MuttedColorShades1' scheme is used. The default contour levels correspond to 'favored' and 'allowed' regions [Ref 144] for phi and psi angles.

The colors are used to fill spaces between contour levels. The values for contour levels must be ascending

order. The number of colors must be one less than the number contour levels.

The format of contour level and color specification is as follows:

```
PlotType:Level,Color,Level,...;PlotType:Level,Color,Level,...
```

The valid values for plot type are:

```
General, Glycine, Proline, or PreProline
```

The contour level must be a number. The color value must be a valid color name or a hexadecimal color string supported by Matplotlib. No validation is performed.

For example:

```
General: 0.0, #FFFFFF, 0.0005, #EBF1DE, 0.02, #C3D69B, 1.0
```

--levelsAndColorsScheme <text> [default: MuttedColorShades1]

Default contour levels and colors scheme. Possible values: MuttedColorShades1, MuttedColorShades2, or BrightColorShades.

This option is only used during 'auto' value of '--levelsAndColors' option. The default contour levels correspond to 'favored' and 'allowed' regions [Ref 144] for phi and psi angles.

The default contour and color values for different default schemes are shown below:

MuttedColorShades1:

```
General: 0.0, #FFFFFF, 0.0005, #EBF1DE, 0.02, #C3D69B, 1.0
Glycine: 0.0, #FFFFFF, 0.002, #7FD9FF, 0.02, #FAC090, 1.0
Proline: 0.0, #FFFFFF, 0.002, #E6E0EC, 0.02, #B3A2C7, 1.0
PreProline: 0.0, #FFFFFF, 0.002, #DCE6F2, 0.02, #95B3D7, 1.0
```

MuttedColorShades2:

```
General: 0.0, #FFFFFF, 0.0005, #EBF1DE, 0.02, #D7E4BD, 1.0
Glycine: 0.0, #FFFFFF, 0.002, #FDEADA, 0.02, #FCD5B5, 1.0
Proline: 0.0, #FFFFFF, 0.002, #E6E0EC, 0.02, #CCC1DA, 1.0
PreProline: 0.0, #FFFFFF, 0.002, #DCE6F2, 0.02, #B9CDE5, 1.0
```

BrightColorShades: [Ref 145]

```
General: 0.0, #FFFFFF, 0.0005, #B3E8FF, 0.02, #7FD9FF, 1.0
Glycine: 0.0, #FFFFFF, 0.002, #FFE8C5, 0.02, #FFCC7F, 1.0
Proline: 0.0, #FFFFFF, 0.002, #D0FFC5, 0.02, #7FFF8C, 1.0
PreProline: 0.0, #FFFFFF, 0.002, #B3E8FF, 0.02, #7FD9FF, 1.0
```

-o, --outfile <outfile>

Output image file name.

A set of output files is optionally generated for 'MultipleFiles' value of '--outMode' option. The names of these output files are automatically generated from the the name of the specified output file as shown below:

```
General: <OutfileRoot>_General.<OutfileExt>
Glycine: <OutfileRoot>_Glycine.<OutfileExt>
Proline: <OutfileRoot>_Proline.<OutfileExt>
PreProline: <OutfileRoot>_PreProline.<OutfileExt>
```

--outMode <Single or Multiple> [default: SingleFile]

A single output file containing all four Ramachandran plots or multiple output files corresponding to different types of Ramachandran plots.

The phi and psi angles are categorized into the following groups corresponding to four types of Ramachandran plots:

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

proline

--overwrite

Overwrite existing files.

-p, --precision <number> [default: 2]

Floating point precision for plotting the calculated phi and psi angles.

--scatterMarkerColor <text> [default: #1f77b4]

Scatter marker color for plotting to phi and psi angles. It must be a valid Matplotlib value. The default color is dark blue.

--scatterMarkerSize <number> [default: 1.0]

Scatter marker size for piloting phi and psi angles. It must be a valid Matplotlib value.

--scatterMarkerStyle <text> [default: .]

Scatter marker style for piloting phi and psi angles. It must be a valid Matplotlib value. For example: '.' (point), ',' (pixel), 'o' (circle), etc.

--ticksMajorInterval <number> [default: auto]

Display major marks on axes at intervals specified in degrees for phi and psi angles. The default value is dependent on the the value of '--outMode' option: SingleFile: 180; MultipleFiles: 90

The grid lines are drawn at the locations of major tick marks.

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

Display minor tick marks. The major tick mark are always displayed.

--ticksMinorInterval <number> [default: auto]

Display minor marks on axes at intervals specified in degrees for phi and psi angles. The default value is dependent on the the value of '--outMode' option: SingleFile: 45; MultipleFiles: 10

-w, --workingdir <dir>

Location of working directory which defaults to the current directory.

EXAMPLES

To generate Ramachandran plot for all residues across all chains in input file and write out a single SVG file containing all four types of plots, type:

```
% PyMOLGenerateRamachandranPlots.py -i Sample3.pdb -o Sample3Out.svg
```

To generate Ramachandran plot for all residues across all chains in input file and write out four SVG files corresponding to four types of plots, type:

```
% PyMOLGenerateRamachandranPlots.py --outMode MultipleFiles
  -i Sample3.pdb -o Sample3Out.svg
```

To generate Ramachandran plot for all residues in a specific chain in input file and write out a single PDF file containing all four types of plots, type:

```
% PyMOLGenerateRamachandranPlots.py -c E -i Sample3.pdb
  -o Sample3Out.pdf
```

To generate Ramachandran plot for all residues across all chains in input file using specific options and write out four PNG files containing all four types of plots, type:

```
% PyMOLGenerateRamachandranPlots.py --outMode MultipleFiles
  --figSize "6,4" --figDPI 600 --fontTitleSize 10 --fontTitleWeight
  normal --greek no --grid no --levelsAndColors
  "General: 0.0, #FFFFFF, 0.0005, #B3E8FF, 0.02, #7FD9FF, 1.0"
  -i Sample3.pdb -o Sample3Out.png
```

AUTHOR

Manish Sud(msud@san.rr.com)

SEE ALSO

DownloadPDBFiles.pl, PyMOLCalculatePhiPsiAngles.py, PyMOLCalculateRMSD.py, PyMOLCalculateProperties.py

COPYRIGHT

Copyright (C) 2026 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.