

# Technical Report: SPIDAL Summer REU 2018

## Dihedral Analysis in MDAnalysis

Henry Mull\*, Oliver Beckstein†

### Abstract

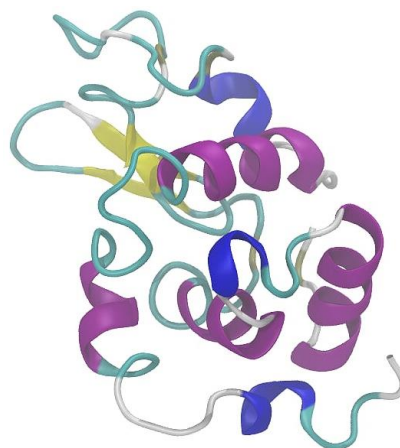
MDAnalysis is an open source Python-based library that is used to quickly and interactively analyze molecular dynamic simulations. Using functions already with, a new analysis module for analyzing dihedral angles, particularly  $\phi$ ,  $\psi$ ,  $\chi_1$  and  $\chi_2$ , can be made which will calculate all of the dihedral angles of interest, for each time step of the trajectory. This can significantly increase the speed and ease with which dihedral angles are characterized and studied.

### Introduction

Molecular dynamics (MD) simulations are important tools that compliment experiment and theory to make a complete picture of natural phenomenon. They are especially useful for simulating biological processes to understand the motion or conformational changes that occur on a molecular level. Popular molecules of interest are proteins (Figure 1<sup>1</sup>) and lipids, as MD simulations often involve some molecule of interest interacting with a fluid environment, such as a solvent or a lipid membrane<sup>2</sup>.

One useful tool for MD simulations is MDAnalysis<sup>3,4</sup>, a Python-based library for interactively analyzing MD simulations. With MDAnalysis, it is easy to access information about the system of interest. It can simply give the coordinates of atoms or the distances between them, or it can perform more complex calculations, such as determining the radius of gyration at each point in the trajectory. MDAnalysis also allows the user to specify smaller parts of a molecule and focus only on that, either simplifying the analysis itself or allowing for comparison to other parts of the molecule.

With the ability to determine coordinates of each atom, MDAnalysis opens up the possibility of an extensive analysis of protein structure. One feature that MDAnalysis was lacking was a dihedral angle analysis module. Aside from general dihedral angles formed from any four atoms, more specific areas of study are those encompassed by Ramachandran plots and Janin plots. Ramachandran plots compare  $\phi$  and  $\psi$  angles of a protein backbone (Figure 2) and are useful for analyzing protein structure as these angles define secondary and tertiary protein structure<sup>5</sup>. Janin plots compare the  $\chi_1$  and  $\chi_2$  angles of each residues side chain<sup>6</sup> (assuming the side chain has

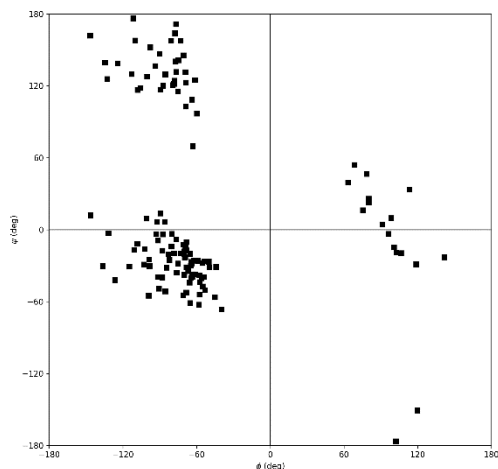


**Figure 1.** MD simulation of lysozyme from an egg (PDB ID: 1AKI). This snapshot is an example of the visualization that can be done with MD simulation software like GROMACS and VMD. Water molecules are not shown.

---

\* Department of Chemistry and Biochemistry, California Polytechnic State University, San Luis Obispo CA

† Department of Physics and Center for Biological Physics, Arizona State University, Tempe AZ



**Figure 2.** This is an example of a Ramachandran Plot made with MDAnalysis for a single frame. The protein analyzed was adenylate kinase. The axes span from  $-180^{\circ}$  to  $180^{\circ}$  for both  $\phi$  and  $\psi$ .

enough atoms for both angles). Using MDAnalysis, determining the dihedral angle for any subset of atoms, or the dihedral angles of an entire protein, and showing how it changes over time can become a relatively quick process.

## Methods

The dihedrals module was developed using Jupyter notebooks with Python 2.7.0. It was based on the current development version of MDAnalysis, version 0.18.1, and also utilized the python packages NumPy version 1.15.0, Matplotlib version 2.2.2, Six version 1.11.0, and pytest version 3.7.1.

The classes in the dihedrals module uses a few MDAnalysis functions and methods that imposed some requirements for input and affected the classes functionality. Most importantly is the library function `distances.calc_dihedrals`, as it was the function used to calculate dihedral angles in each class. This function requires four sets of coordinates as input, one that corresponds to each “position” of an atom in a dihedral. The order of atoms is important for this function; it considers the first three positions to form one plane, and the last three positions to form the other plane. Because of this input requirement, the classes have different ways of creating the list of positions. The Dihedral class is straightforward in that it simply takes a list of AtomGroups that are four atoms long. Separating those atoms into the appropriate groups and changing them to coordinates is done within the class itself. The Ramachandran class utilizes the `phi_selection()` and `psi_selection()` methods, which return the appropriate AtomGroup of four atoms for whichever residue they are used on. Then it can use the same process as the Dihedral class to calculate both  $\phi$  and  $\psi$  angles for whatever residues were given. These selection methods make the input for the class simpler, as it can just be a ResidueGroup or an AtomGroup corresponding to the residues of interest, both of which are basic selections using MDAnalysis. For the Janin class, because there was no `chi2_selection()` and the `chi1_selection()` was not inclusive enough, the class does the selection itself if given the same kind of input as the Ramachandran class.

Reference plots that showed allowed and marginally allowed regions for both the Ramachandran and Janin plots are available to compare to any angles made with these classes. Data for these was generated using the classes discussed in this paper, but the selection of 500 PDB files used was taken from Lovell et al.<sup>7</sup>, though not all files were used, as some could no longer be retrieved or had missing or duplicate atoms. Overall, 471 files were analyzed for the Ramachandran plot, giving 100, 816 residues. However, only about half of the files were analyzed for the Janin plot, and only 33, 365 residues were plotted due to missing or duplicate atoms. The contour lines levels were chosen so the allowed region of both plots would contain 90% of the points, and the marginally allowed region of the Ramachandran and Janin plots would be 99% and 98% respectively.

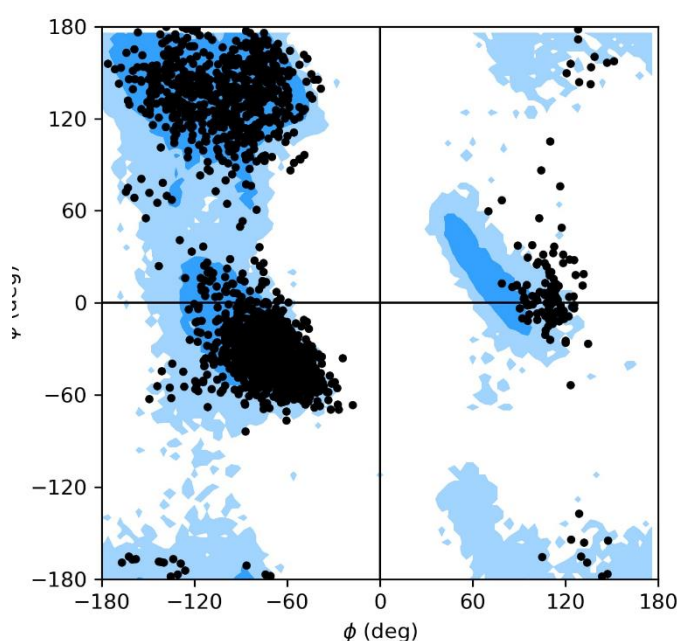
## Results

The Dihedral class calculates angles for any sets four atoms. These could even be atoms that are not bonded since the angle is calculated only from the coordinates of the atoms, though there would be little use for that. This class would be most useful for analysis of dihedral angles for very specific selections of a protein or residue, such as part of the side chain of a specific kind of residue. Using MDAnalysis, selecting the required atoms for each dihedral of interest is very simple, then all that is needed is collecting them in a list and using it as an argument for the Dihedral class. Due to the nature of the class, the input must be a list, even if there is only one angle of interest. The class will check if there is an AtomGroup in the list that is longer than four atoms, the class will raise an error. If that is not the case, the class will then return a NumPy array that contains lists of the angles calculated as separate time steps.

While the processes are relatively the same for the Ramachandran and Janin classes, they are slightly easier to use because much of the work needed to make AtomGroups is done by the class itself. The Ramachandran class can take either a ResidueGroup or a corresponding AtomGroup as input. This is possible because the class will determine what residues the input is a part of, and it will calculate the  $\phi$  and  $\psi$  angles for those residues. The class contains a few checks to be more user friendly. If there are any atoms in the input which are outside of the protein, the class will raise an error and explain why. Because it does not make sense to calculate the  $\phi$  and  $\psi$  angles for the N-terminus and C-terminus, they will be removed and a warning will be

raised so the user knows not to expect them in the final array. Finally, because some topologies contain alternative locations for some atoms, and `phi_selection()` and `psi_selection()` will return None for those residues, residues that return None for either angle will be removed from the list and the user will be warned. Following that, a NumPy array containing lists of  $\phi$  and  $\psi$  angles for each time step will be returned. The angles themselves can be accessed and plotted, but the Ramachandran class contains a `plot()` method which creates a scatter plot using Matplotlib. By using the keyword argument `'ref=True'`, a reference plot of allowed and marginally allowed regions can be put on the same axes (Figure 3).

The Janin class works very similarly to the Ramachandran class, but it is slightly more receptive to topologies with alternative locations. AtomGroups selections were made without using other functions such as `phi_selection()` because they did not exist or were not complete enough.



**Figure 3.** An example of the plotting function of Ramachandran with the reference graph on the axes. The protein analyzed is the same as about, adenylate kinase, but for all frames of the trajectory.

Because of this, it is possible to eliminate the presence of atoms with certain alternative locations before using the class to avoid any conflicts from having too many atoms. This is possible here and not in the Ramachandran class because `phi_selection()` and `psi_selection()` both look at the whole residue they are used on, and will collect all atoms that could be used for that selection, even duplicate atoms, which will make the selection too long and thus return None. The Janin class still contains additional checks. Like the Ramachandran class, Janin will fail if there are atoms outside of the protein. It will also check if there are residues which do not have  $\chi_1$  or  $\chi_2$  angles and remove those residues from the list after warning the user. Because it selects atoms directly from the universe, and not from lists of premade atom groups, it also checks that the final selections are all the same length, an error that can occur when residues are missing atoms in their side chains. The class returns the angles in the same format as the Ramachandran class and has the same plot functionality, but with different axes to account of the conventional angle range of Janin plots.

## Conclusion

The dihedrals module for MDAnalysis facilitates the analysis of any dihedral angle, with a particular focus on the  $\phi$ ,  $\psi$ ,  $\chi_1$ , and  $\chi_2$  dihedral angles. Especially for Janin analysis, which is often not considered in scientific software packages, dihedral analysis can be performed much faster than before. While there are some drawbacks, many of them can be attributed to the topology files themselves rather than the classes, and often those errors can be accounted for.

## Acknowledgements

Funding was provided by the National Science Foundation for a REU supplement to award ACI-1443054.

## References

- (1) Humphrey, W.; Dalke, A.; Schulten, K. *J. Molec. Graph.* **1996**, *14*, 33–38.
- (2) Mura, C.; McAnany, C. E. *Mol. Simul.* **2014**, *40* (10–11), 732–764.
- (3) N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. B. *J. Comput. Chem* **2011**, *32*, 2319–2327.
- (4) Gowers, R. J.; Linke, M.; Barnoud, J.; Reddy, T. J. E.; Melo, M. N.; Seyler, S. L.; Domański, J.; Dotson, D. L.; Buchoux, S.; Kenney, I. M.; Beckstein, O. **2016**, No. Scipy, 98–105.
- (5) Ramachandran, G. N.; Ramakrishnan, C.; Sasisekharan, V. *J. Mol. Biol.* **1963**, *7* (1), 95–99.
- (6) Janin, J.; Wodak, S.; Levitt, M.; Maigret, B. *J. Mol. Biol.* **1978**, *125* (3), 357–386.
- (7) Lovell, S. C.; Davis, I. W.; Adrendall, W. B.; de Bakker, P. I. W.; Word, J. M.; Prisant, M. G.; Richardson, J. S.; Richardson, D. C. *Proteins-Structure Funct. Genet.* **2003**, *50* (August 2002), 437–450.