Structural Insight Visualization: The Ramachandran Plot

Abstract

This project uses protein structure data from two datasets, calculates the ϕ and ψ angles of its amino acid residues, generates Ramachandran plots, and compares the differences between them. Possible reasons for these differences are explored, and data filtering strategies are proposed so that the Ramachandran plot of the dataset_2 looks closer to that of the dataset_1. Other matters that should be considered are also discussed and the plots are well optimized.

1. What is the Ramachandran plot?

The Ramachandran plot originally developed in 1963 by G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan[1],is a way to visualize energetically allowed regions for backbone dihedral angles ψ against ϕ of amino acid residues in protein structure.

Figure 1 illustrates the definition of ϕ and ψ. The ω angle at the peptide bond is normally 180°, since the partial-double-bond character keeps the peptide bond planar[2]. Figure 2 shows the allowed conformational regions of the main chain ϕ, ψ dihedral angles calculated by Ramachandran et al. from 1963 to 1968 based on the rigid spherical model: full radius in solid outline, reduced radius in dashed, and relaxed tau (N-Cα-C) angle in dotted lines[3]. Because dihedral angle values are circular and 0° is the same as 360°, the edges of the Ramachandran plot "wrap" right-to-left and bottom-to-top. The Ramachandran plot in Figure 3 contains 100,000 data points acquired from the high-resolution crystal structure, residues in the α-helical conformation are labelled α and residues in the β-strand conformation are labelled β. The data clusters in the upper right quadrant primarily represent turns.

By checking Ramachandran plots, we can identify helices, folds, and other shapes in proteins, helping us predict protein properties such as stability and folding rate, which are important for our understanding of protein structure.

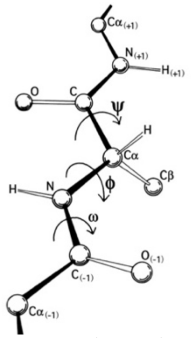

Figure 1: The definition of the φ and ψ

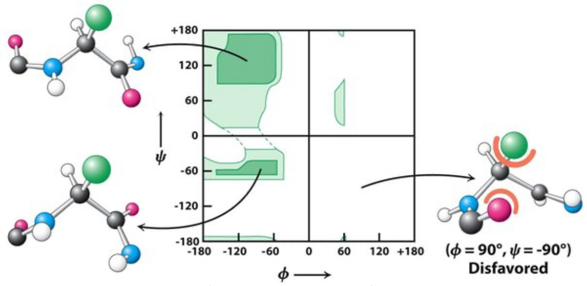

Figure 2: The allowed conformational regions of the main chain φ, ψ

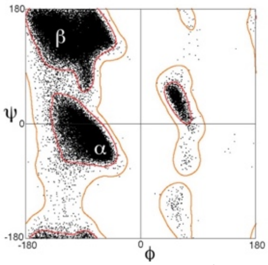

Figure 3: Ramachandran plot for the general case

2. Data used

Dataset_1 and dataset_2 each contain about 1,600 protein names, detailed structure files can be obtained from Protein Data Bank (https://files.rcsb.org). The dataset contains multiple types of proteins, such as Plant Protein (3X3S), Isomerase (4AL0), Antibilti (1A7Y), etc. These come from different organisms such as bacteria, fungi, plants, or animals and cover different conformations and functions, provide rich information that can be used for a variety of analyses and studies, including the Ramachandran plot. Drawing and comparing the Ramachandran plots of these two datasets can help us better understand the conformational space and structural features of them.

3. Generate Ramachandran plots

A total of 709,607 pairs of ϕ and ψ were calculated in Dataset_1, and 21,710,336 pairs of ϕ and ψ were calculated in Dataset_2, which is approximately 30 times the amount of dataset_1. The Ramachandran plots created using the atomic structure data in dataset_1 and dataset_2 are as follows.

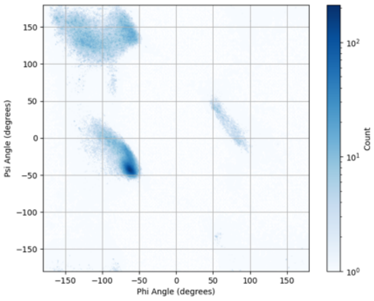

Figure 4: Ramachandran plot of dataset_1

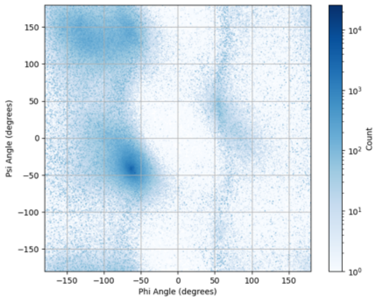

Figure 5: Ramachandran plot of dataset_2

The code to calculate and draw the Ramachandran plots is as follows:

```
for chain in model:
    for residue in chain:
        next_res = chain.next_residue(residue)
        prev_res = chain.previous_residue(residue)
        if next_res:
            phi, psi = gemmi.calculate_phi_psi(prev_res, residue, next_res)
            if not isnan(phi) and not isnan(psi):
                phi_angles.append(degrees(phi))
                psi_angles.append(degrees(psi))
plt.imshow(hist_masked.T, extent=[xedges.min(), xedges.max(), yedges.min(), yedges.max()],
           origin='lower',cmap='Blues', interpolation='gaussian', norm='log')
```

4. Why are there differences between the two plots?

There are clear differences between the two Ramachandran plots. The Ramachandran plot of dataset_1 is more consistent with the theoretical distribution, and its structural distribution is more concentrated; the Ramachandran plot of dataset_2 shows a wider range of ϕ and ψ combinations, and the distribution is more dispersed. This may be because the protein structures in dataset_2 are more diverse and flexible, or it may be due to different experimental measurement standards. To explore the reasons for this difference, a comparative analysis of the following variables was conducted for the two datasets: Resolution Value, B-Factor Value, Peptide Bond.

4.1 Resolution Value

The resolution value is an important indicator in X-ray diffraction experiments, reflecting the analytical ability of the experiment. Higher resolution X-ray data generally results in higher quality three-dimensional structures. The figure below shows the B-Factor distribution of the protein structures corresponding to the two datasets.

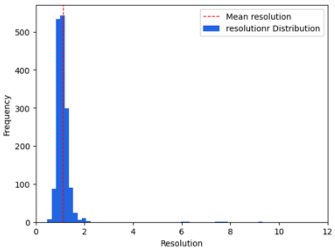

Figure 6: Resolution value of dataset_1

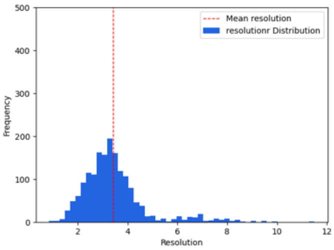

Figure 7: Resolution value of dataset_2

Most of the proteins in Dataset_1 are measured with an average resolution value of 1.11Å, a variance of 0.19, and 99.44% of the resolution values falling within the [0, 2] interval. The proteins in dataset_2 are mostly measured with a mean value of 3.43 Å and a variance of 1.59, 75.12% of the resolution values fall within the [2, 4] interval. The higher resolution value is a main reason why the Ramachandran plot distribution of dataset_2 is relatively scattered. Part of the code to calculate and draw the B-Factor density distribution map is as follows:

```
for path in tqdm(gemmi.CoorFileWalk(pdb_directory), total=len(pdb_list)):
    structure = gemmi.read_structure(path)
    resolution = extract_resolution(structure)
    if resolution:
        high_resolution_values.append(resolution)
plt.hist(high_resolution_values, bins=50, color=custom_color, alpha=0.7)
```

4.2 B-Factor

B-Factor is an indicator that describes the vibration of atoms in a crystal structure. Each atom can have a corresponding B-Factor value. The larger the B-Factor value, the more violent the atomic vibration, the greater the uncertainty of the position in the crystal structure, and the higher the conformational flexibility of the protein. The figure below shows the B-Factor distribution of the two data sets.

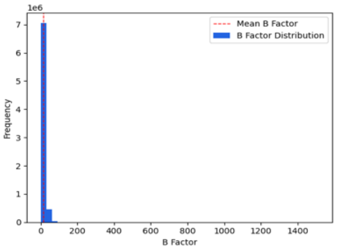

Figure 8: B-Factor value of dataset_1

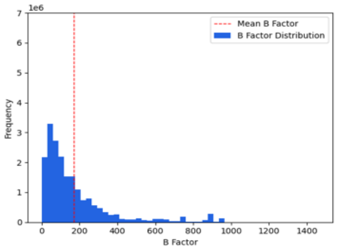

Figure 9: B-Factor value of dataset_2

The mean B-Factor value of dataset_1 is 15.57Å^2, with 99.75% of the values falling within the range [0,100]. The average B-Factor value of dataset_2 is as high as 169.66Å^2, and only 47.60% of the B-Factor values fall within the range [0,100]. The high B-Factor values of the proteins in dataset_2 indicate high conformational flexibility, resulting in a broader distribution in the Ramachandran plot. This is also the second reason why its distribution is relatively scattered. Part of the code to calculate and draw the Resolution Value density distribution map is as follows:

```
for file_path in tqdm(file_paths, desc="Processing files"):
    doc = gemmi.cif.read(file_path)
    structure = gemmi.read_structure(file_path)
    for model in structure:
        for chain in model:
            for residue in chain:
                for atom in residue:
                    b_factors.append(atom.b_iso)
plt.hist(b_factors, bins=50, color=custom_color)
plt.axvline(avg_b_factor, color='red', linestyle='dashed', linewidth=1)
```

4.3 Peptide Bond

In the main chain structure of a protein, multiple amino acid residues are connected to each other through peptide bonds to form a linear polypeptide chain. Peptide bonds consist of covalent bonds between an amino group (NH) and a carbonyl group (C=O). The peptide bond angle determines the spatial arrangement and covalent bond angle between two adjacent amino acid residues on the protein backbone. Different peptide bond angles will lead to changes in the conformational space of the main chain. For example, some peptide bond angles may favor the formation of α-helical structures, while others favor the formation of β-sheet structures. Therefore, different values of the peptide bond angle will limit the conformational space of the protein backbone, thereby affecting the distribution of data points in the Ramachandran diagram. The figure below shows the peptide bond turn angle distribution for the two data sets:

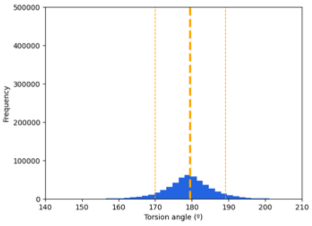

Figure 10: Trans angle of dataset_1

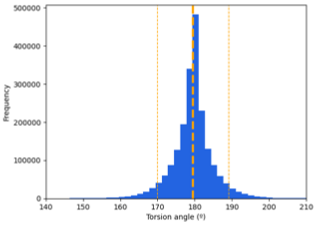

Figure 11: Trans angle of dataset_2

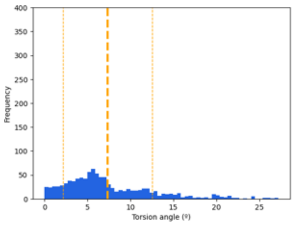

Figure 12: Cis angle of dataset_1

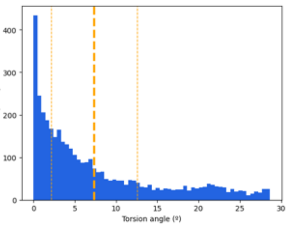

Figure 13: Cis angle of dataset_2

As mentioned above, Dataset_1 contains 709,607 sets of ϕ and ψ data, and Dataset_2 contains 21,710,336 sets of ϕ and ψ data, which is approximately 30 times the amount of data in Dataset_1. In Dataset_2, the number of 180° peptide bonds is about 6 times that of Dataset_1. Therefore, from a probabilistic perspective, the Ramachandran plot of Dataset_2 is more likely to exhibit a more dispersed distribution. However, objectively speaking, this influencing factor is not as great as the two factors introduced previously. Part of the code for calculating the Peptide bond torsion angle density distribution chart is as follows:

```
for chain in model :
    for residue in chain :
      next_res = chain.next_residue ( residue )
      if next_res:
        omega = gemmi.calculate_omega(residue, next_res)
        if not isnan(degrees(omega)):
          if omega < 0.0 :
            omega = omega + 2*pi
          if omega < 0.5 :
            cis_omega_angles.append ( degrees ( omega ) )
          else :
            trans_omega_angles.append ( degrees ( omega ) )
```

5. Data filtering strategy for dataset_2

By plotting the data in dataset_2 with a resolution in the range [0, 3.5], it can make its Ramachandran plot closer to that of dataset_1. Set a similar resolution range to make the Ramachandra plots of the two data sets more consistent and comparable. The r graph of dataset_2 drawn after filtering is as follows:

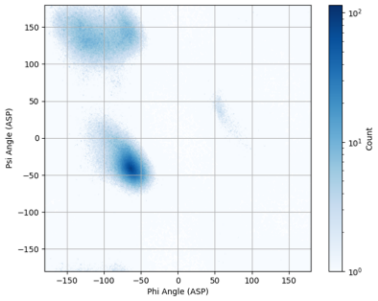

Figure 14: Ramachandran plot of filtered dataset_2

As can be seen from the above figure, after data filtering, the Ramachandran plot of dataset_2 is very similar to the Ramachandran plot of dataset_1. The code to implement data filtering is as follows:

```
resolution = structure.resolution
if resolution is not None and 0 <= resolution <= 3.5:
    for chain in model:
        prev_res = None
        for residue in chain:
            next_res = chain.next_residue(residue)
            if prev_res is not None and next_res is not None:
                phi, psi = gemmi.calculate_phi_psi(prev_res, residue, next_res)
                if not isnan(phi) and not isnan(psi):
                    phi_angles_redo.append(degrees(phi))
                    psi_angles_redo.append(degrees(psi))
            prev_res = residue
```

6. Other matters needing attention

Regardless of the residue type, the code for calculating ϕ and ψ values creates a data set, storing all ϕ and ψ values in a single list. However, different types of residues have different amino acid side chain structures and chemical properties, thus affecting the range of values of their Phi (ϕ) and Psi (ψ) angles. Some residues may be more prone to specific conformational states, while others may be more flexible or diverse, resulting in different types of residues potentially exhibiting different distributions and characteristics in the Ramachandran diagram. So, it is also important to analyse the Ramachandran plot of a specific residue separately.

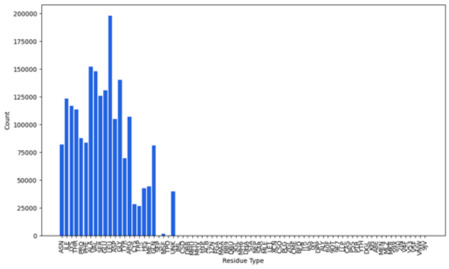

Figure 15: Number of Residues by Type of dataset_2

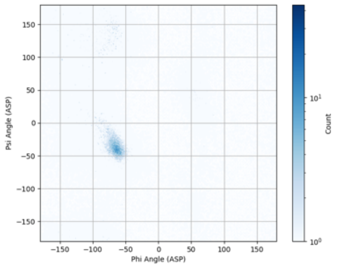

Figure 16: Ramachandran plot of ASP in dataset_2

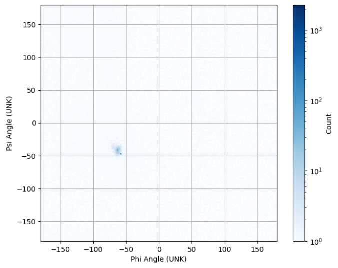

Figure 17: Ramachandran plot of UNK in dataset_2

The number of each residue is showed in Figure 13, and the Ramachandran plot of ASP and UNK was showed in Figure 14 and Figure 15 respectively, from which we can see the different distributions of Ramachandran plot for different residues and meet different research requirements. The code for using a dictionary to identify and store the ϕ and ψ angles of different residues is as follows:

```
for chain in model:
    for residue in chain:
        next_res = chain.next_residue(residue)
        prev_res = chain.previous_residue(residue)
        if next_res:
            phi, psi = gemmi.calculate_phi_psi(prev_res, residue, next_res)
            if not isnan(phi) and not isnan(psi):
                residue_type = residue.name
                if residue_type not in residue_angles:
                    residue_angles[residue_type] = {'phi': [], 'psi': []}
                residue_angles[residue_type]['phi'].append(degrees(phi))
                residue_angles[residue_type]['psi'].append(degrees(psi))
```

7. Advanced Features

Figure 4 and Figure 14 showed above are the Ramachandran plots of dataset_1 and dataset_2 respectively after optimization. Optimization strategies include adding grid lines, smoothing curves, and adjusting the default color scheme. For better comparison, here is the Ramachandran plot before optimization. It can be observed that the following two graphics before optimization have problems such as smoothness, unclear color change, and weak sense of layering. In comparison, the optimized Ramachandran diagram presents a clearer and more readable image.

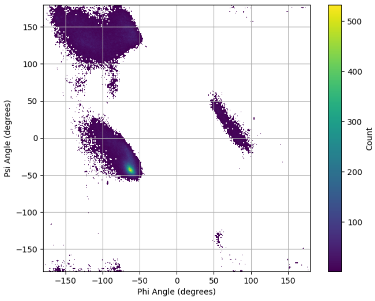

Figure 18: Ramachandran plot of dataset_1 before optimization

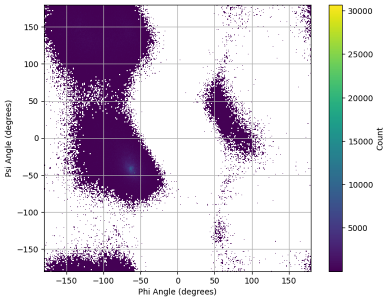

Figure 19: Ramachandran plot of dataset_2 before optimization

The key code for optimizing the Ramachandran plot is as follows. The origin parameter specifies the coordinate origin position of the image, and 'lower' means that the origin is in the lower left corner of the image. The cmap parameter specifies the color map used to display the image. 'Blues' maps low values to dark blue and high values to light blue. The interpolation parameter specifies the interpolation method of the image. Here, the image is smoothed by fitting a Gaussian distribution to make the image look more continuous and smoother. The norm parameter uses logarithmic normalization, which logarithmically transforms the values in the image to better demonstrate the distribution characteristics of the data. The plt.grid() function is used to display grid lines on the image to improve the readability of the image.

```
plt.imshow(hist.T, extent=[
    xedges.min(), xedges.max(), yedges.min(), yedges.max()],
           origin='lower',cmap='Blues', interpolation='gaussian', norm='log')
plt.grid(True)
```

8. Results and discussion

Clear differences were found while analysing the Ramachandran plots of both datasets. The plot of Dataset_1 is more consistent with the theoretical distribution, with a more concentrated distribution of structures, while the plot of Dataset_2 shows a wider range of ϕ and ψ combinations, with a more dispersed distribution. This difference is due to reasons such as more diverse and flexible protein structures in Dataset_2, higher B-Factor values, and different experimental measurement standards for resolution values. Data filtering strategy was proposed to improve its consistency so that the Ramachandran plot of the dataset_2 looks closer to the first plot. Other matters that should be considered are also discussed and the plot is well optimized.

In future studies, the impact of other factors on the Ramachandran plot can also be considered, such as solvent environment, bound ligands, etc. By comprehensively analysing these factors, a more comprehensive understanding of the characteristics and properties of different protein structures can be obtained.

References

1. Ramachandran, G. N.; Ramakrishnan, C.; Sasisekharan, V., Stereochemistry of polypeptide chain configurations. Journal of Molecular Biology 1963, 7 (1), 95-99.

2. Pauling, L.; Corey, R. B.; Branson, H. R., The structure of proteins; two hydrogen-bonded helical configurations of the polypeptide chain. Proc Natl Acad Sci U S A 1951, 37 (4), 205-11.

3. Ramachandran, G. N.; Sasisekharan, V., Conformation of Polypeptides and Proteins**The literature survey for this review was completed in September 1967, with the journals which were then available in Madras and the preprinta which the authors had received.††By the authors' request, the publishers have left certain matters of usage and spelling in the form in which they wrote them. In Advances in Protein Chemistry, Anfinsen, C. B.; Anson, M. L.; Edsall, J. T.; Richards, F. M., Eds. Academic Press: 1968; Vol. 23, pp 283-437.