A Discretized Approach for Modeling the Angular Distributions in Protein Structures

Clara Lestruhaut[1], Octave Borgard[1], and Supervisor : Juan Cortes[2]

[1] Student at INSA Toulouse, Applied Mathematics department
[2] LAAS

05.13.2025

Abstract

The study of proteins is fundamental to understanding biological systems, as these macromolecules play a central role in nearly all cellular processes. A key aspect of protein structure lies in the angular distributions of amino acids, which influence folding, stability, and function. While existing models for these distributions rely on advanced statistical or machine learning techniques, they are often computationally demanding and difficult to implement. Despite the success of these advanced techniques, no approach currently utilizes simpler statistical methods capable of achieving similar goals. In this work, we propose an alternative method based on data discretization and random sampling. This approach aims to model amino acid angular distributions through a simple pipeline, enabling the generation of new samples while keeping computations efficient. We applied various sampling strategies (ranging from uniform to quadratic interpolations) to generate new angular configurations. We then evaluates the quality of the results using statistical tests: the torus test [1] confirmed its effectiveness; for most amino acid combinations, the generated distributions were statistically indistinguishable from real data, especially when using linear or positive quadratic interpolation. Although some limitations persist (such as lower performance in cases involving central proline residues) this approach offers a lightweight and adaptable alternative for modeling protein structures. It could serve as a foundation for future applications requiring fast and interpretable sampling methods.

Keywords: Protein modeling, Amino acid angles, Statistical sampling, Discretization, Interpolation.

Contents

1 Introduction

Beyond their biological significance, proteins are of great interest in various applied fields. In pharmacology, they serve as therapeutic targets; in biotechnology, their catalytic properties are widely exploited; and in bio-nanotechnology, they are used as structural components of nanodevices. In all of these domains, a comprehensive understanding of the intricate relationships between protein sequence, structure, and function is essential.

Proteins are composed of linear chains of amino acids, of which there are only twenty, each represented by a single-letter code. The primary structure of a protein refers to the specific sequence of these amino acids. However, protein function is largely determined by its three-dimensional conformation, which results from successive levels of structural organization. The secondary structure describes local structural motifs, such as α-helices and β-sheets, which fold further into the tertiary structure, forming the overall 3D shape of the protein. The polypeptide backbone consists of three covalent bonds per amino acid residue. Since the peptide bond is planar, only two degrees of rotational freedom remain, defined by the dihedral angles ϕ and ψ, as shown in Figure 1. These angles are critical for modeling protein folding and function.

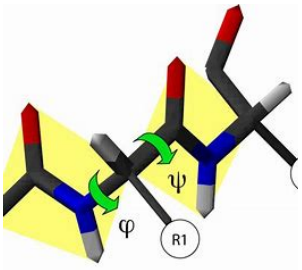

Figure 1: Representation of the dihedral angles ϕ and ψ.

Selecting an appropriate method

During the first semester, we have conducted a comparative analysis of three methodologies aimed at improving the understanding and modeling of amino acid angular distributions.

The first article, [2] “Neighbor-Dependent Ramachandran Probability Distributions of Amino Acids Developed from a Hierarchical Dirichlet Process Model.” proposed an advanced statistical model based on the hierarchical Dirichlet process while considering the effects of neighboring amino acids. Despite its very good precision for loop conformation predictions, implementing the HDP is complex and requires expensive algorithms.

The second article, [3] “Getting ‘$\phi\psi$al’ with proteins: Minimum Message Length Inference of Joint Distributions of Backbone and Side Chain Dihedral Angles,” proposed a second approach to the protein conformation modeling problem, based on the Minimum Message Length. This method is quite efficient and seems easier to implement.

The last article, [4] “Deep Learning Methods for Protein Torsion Angle Prediction,” has developed four models based on deep learning, using neural networks and the Boltzmann machine. These methods requires an advanced computing infrastructure to train the models.

After thorough assessment, we identified the Minimum Message Length (MML) method as the optimal compromise between predictive accuracy and computational feasibility. Consequently, we planned to implement this method in the second semester. However, the proprietary nature of the MML algorithm posed a significant challenge, as the organization holding the rights to the code declined to share it. Given the complexity of re-implementing the entire algorithm from scratch, we were forced to explore an alternative approach. This situation illustrates a common issue encountered by engineers: although a solution may exist and effectively address the problem, access to it can be restricted by intellectual property rights. In such cases, engineers must devise an alternative strategy, even if it involves certain limitations in performance.

The alternative approach

In response to these constraints, we opted for a simpler yet effective method: build a discrete model of the distribution and implement a simple sampling algorithm based on this model (detailed methodology to follow). Despite its relative simplicity, this approach allows for robust analysis of amino acid angular distributions while maintaining computational efficiency.

First, we describe the methodology used to model the data, focusing on sampling, discretization, and interpolation. In the second part, we present our results and the tests performed to compare our distribution with the actual one.

2 Materials and Methods

2.1 Dataset Overview

We collected data describing the dihedral angles ϕ, ψ, and ω for each of the 8,000 possible tripeptide combinations of amino acids. Our analysis focused on the ϕ and ψ angles of the *central* amino acid, as these are the most relevant for assessing local backbone conformation. All angles were defined on the interval $[-\pi, \pi]$.

To ensure reliable statistical estimation, we restricted our study to combinations with more than 60 observed values. Below this quantity, sample sizes are too small to allow for meaningful estimation of the distribution, and this could lead to the creation of samples too similar to the original.

The angle ω determines the peptide bond conformation: $\omega \approx 0$ corresponds to the rare *cis* conformation (U-form), while $\omega \approx \pi$ corresponds to the common *trans* conformation (Z-form). Due to the scarcity of cis configurations, we excluded those data points from the analysis.

After applying these filtering criteria, 7,363 tripeptide combinations were retained for further study.

2.2 Discretization Strategy

We stored the data as a vector of floating-point vectors, allowing easy access to the values of interest. Our goal was to express ψ as a function of ϕ, and to represent the result on a grid.

To begin with, we had a matrix of (ϕ, ψ) points of size $n \times 2$, where n is the number of observations in the considered sample. We then added π to all angle values to shift the domain from $[-\pi, \pi]$ to $[0, 2\pi]$, which facilitates division by the chosen resolution. Each value was then divided by the resolution and rounded to the nearest integer. In this way, all values in the interval $[-\pi\ ,\ -\pi + \frac{\text{resolution}}{2}]$ fall into the same grid cell. More generally, all values in the intervals $[-\pi + k \cdot \frac{\text{resolution}}{2}\ ,\ -\pi + (k+1) \cdot \frac{\text{resolution}}{2}]$ for any $k \in N$ are grouped into the same grid cell. The procedure is detailed in Algorithm 1.

This value grouping process was applied to both ϕ and ψ angles. We then counted the number of values in each cell and stored the resulting grid in plain text format (.txt) for further processing. As a result, we obtained 7363 text files (one for each selected tripeptide combination), each representing the angular distribution across various intervals at the desired resolution.

Algorithm 1 Discretization of angular data on a 2D grid

```
procedure DISCRETIZATION(resolution)
        Retrieve data in the format of a vector of vectors called *data*
    $N \leftarrow$ round($2\pi$/resolution) + 1
    Initialize an $M$ matrix of size $N \times N$ filled with zeros
    for $i = 1$ to size(data) - 1 do
        $\psi \leftarrow data[i][0] + \pi$
        $\phi \leftarrow data[i][1] + \pi$
        $x_{\text{idx}} \leftarrow$ round($\psi$/resolution)
        $y_{\text{idx}} \leftarrow$ round($\phi$/resolution)
        $M[y_{\text{idx}}][x_{\text{idx}}] \leftarrow M[y_{\text{idx}}][x_{\text{idx}}] + 1$
    end for
    Save the matrix $M$ in a text file
end procedure
```

Once normalized, the resulting matrix can be interpreted as an approximation of the angular distribution function. It can be visualized as an image, with regions of high probability density in red and areas of lower concentration in blue, as illustrated in Figure 2.

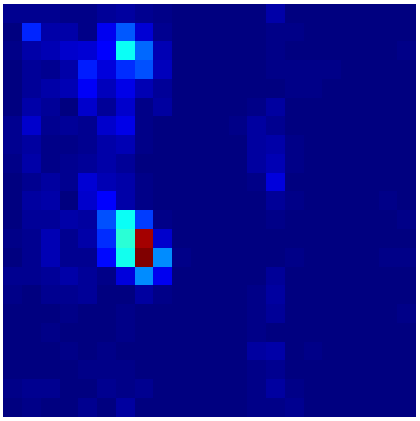

Figure 2: Distribution of angles ϕ and ψ for Alanine-Alanine-Alanine composition, resolution = 0.3.

2.3 Random Sampling Methods

We now had the normalized matrix representing the distribution of the angle pairs (ψ, ϕ). Then, we attempted to generate new angle samples from this data. For this, we explored several approaches: spiked sampling, stepwise sampling, and quadratic sampling. Subsequently, we perform a distribution comparison test to select the best sampling method.

2.3.1 Spiked Sampling

We did a weighted random sampling on the grid structure containing the probabilities of each angle pair. To do this, a random number between 0 and 1 was drawn. Then, by iterating through the grid, the cumulative sum was calculated case by case until this sum exceeded the drawn value. The corresponding angle pair was then considered to be in the interval corresponding to the found case. We then proposed to return the central value of the intervals as the values for the angles. This approach is the simplest, but it has a major drawback: it can only return a finite number of values, one per interval. We expected this method to give very poor results, and it will primarily be used to compare the performance of the following methods. This method is described in the pseudocode below : Algorithm 2.

Algorithm 2 Spiked Sampling

```
$r \leftarrow$ uniform draw in $[0, 1]$
$count \leftarrow 0$
for $i \leftarrow 0$ to $N$ do
    for $j \leftarrow 0$ to $N$ do
        $count \leftarrow count + data[i][j]$
        if $count \geq r$ then
            $\phi \leftarrow j \cdot resolution - \pi$
            $\psi \leftarrow i \cdot resolution - \pi$
            return $(\phi, \psi)$
        end if
    end for
end for
```

2.3.2 Stepwise Sampling

For this method, we first selected the intervals for the angles ψ and ϕ as described earlier (using the cumulative distribution of the probabilities). Then, we performed a random draw for two uniform variables, one for the selected interval for ψ, and the other for the selected interval for ϕ. These two values were then chosen as the generated angle pair.

2.3.3 Linear Sampling

Again, we began by selecting the intervals for the angles ψ and ϕ as previously (again using the cumulative distribution of the probabilities). We then proceeded separately for the two angles. For each angle, we retrieved the probability values of the neighboring intervals.
For instance, we considered the sampling of the angle ψ. We defined $I_1 = [a, b]$ the interval selected for ψ, and I_2 the corresponding interval selected for ϕ. To refine the sampling within I_1, we considered the neighboring intervals:

- $p_{-1} = P(\psi \in [a - \text{resolution}, a] \mid \phi \in I_2)$, the probability just before I_1
- $p_{+1} = P(\psi \in [b, b + \text{resolution}] \mid \phi \in I_2)$, the probability just after I_1
- and $p_0 = P(\psi \in I_1 \mid \phi \in I_2)$, the probability of the selected interval itself.

The matrix borders are handled with a periodic effect, meaning that the first point has as its left neighbor the last point, and vice versa for the last point.
We then performed linear interpolation between p_0 and p_{-1}, and between p_0 and p_1. Next, we applied the rejection sampling method using our linear interpolations as the cumulative distribution function. If the abscissa selected by the rejection sampling method is less than p_0, we used the interpolation between p_0 and p_{-1}; otherwise, we used the interpolation between p_0 and p_1.

Figure 3 provides a graphical representation of the method. We randomly selected the interval from which the angle is to be sampled. This determines the specific region of the histogram to be used. A linear interpolation was then applied to define the sampling distribution within this interval.

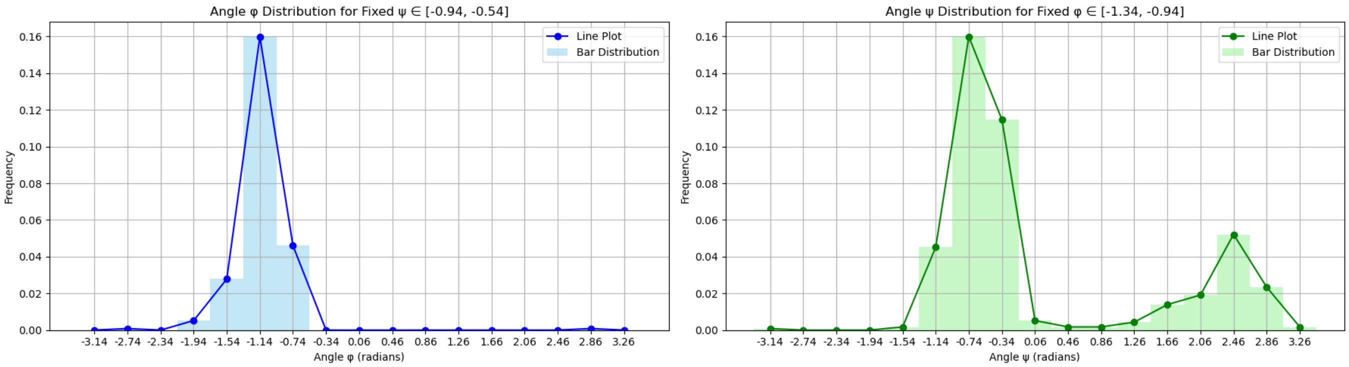

Figure 3: Example of Linear Sampling: Alanine-Alanine-Glutamate, resolution = 0.4

2.3.4 Quadratic Sampling

In the same way as before, we drew a random number between 0 and 1, and as soon as the cumulative sum of probabilities exceeded this number, we considered the point to fall into that cell. However, this time, we refined the process with quadratic interpolation. We took three points (the center and two neighbors) and returned three coefficients a, b, c of a second-degree polynomial $ax^2 + bx + c$. A local parabolic fit to the density function of ϕ or ψ was then constructed, as shown in Figure 4. We proceeded with a rejection sampling method: we sampled x within an interval (restricted around the cell) and accepted the sample if a second random number is less than $ax^2 + bx + c$.

As seen in Figure 4, multiple interpolations appear over a given interval. This occurs because the interpolation depends on the previously selected interval. Thus, only the interpolation associated with the three relevant points (the selected interval and its two neighbors) was used for the chosen case.

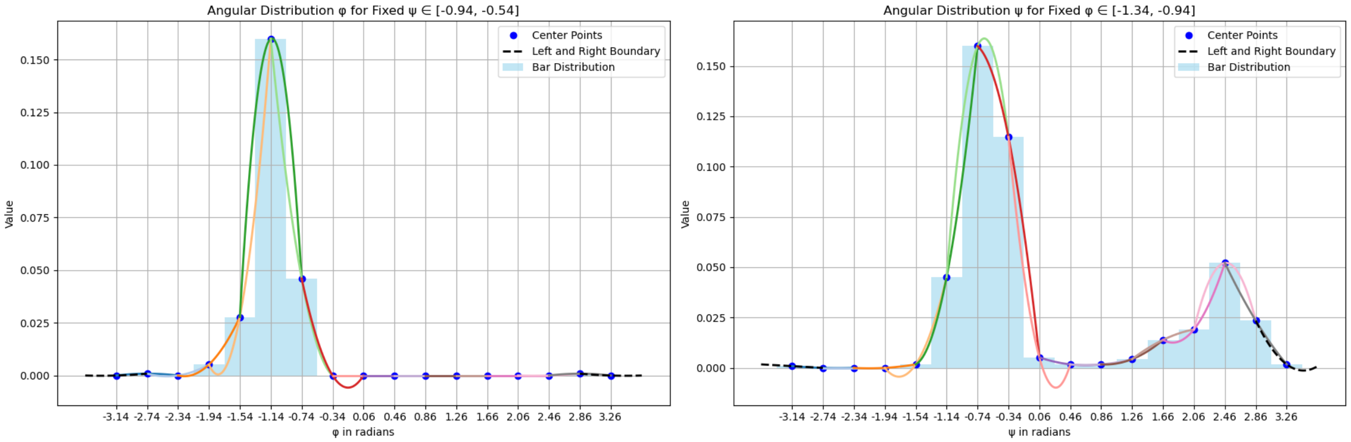

Figure 4: Example of Parabolic Interpolations Sampling: Alanine-Alanine-Glutamate, resolution = 0.4

2.3.5 Positive Quadratic Sampling

The quadratic approximation method presents a major issue: certain parts of the approximation become negative, which is theoretically impossible for a probability distribution function. In the context of rejection sampling, a negative section is equivalent to a zero sampling probability. This is problematic as it results from approximation artifacts and does not reflect the true underlying distribution.

To address this issue, we took advantage of the fact that all values are non-negative, since they represent probabilities, by applying a square root transformation. Specifically, we performed the quadratic interpolation on the square root of the probabilities, and then squared the resulting polynomial. This guarantees that the final polynomial is non-negative over the considered intervals (see Figure 5).

It is worth noting that the computational cost remains comparable to that of the standard quadratic method. However, squaring the interpolated polynomial yields a fourth-degree polynomial, in which the coefficients of the odd-degree terms are zero.

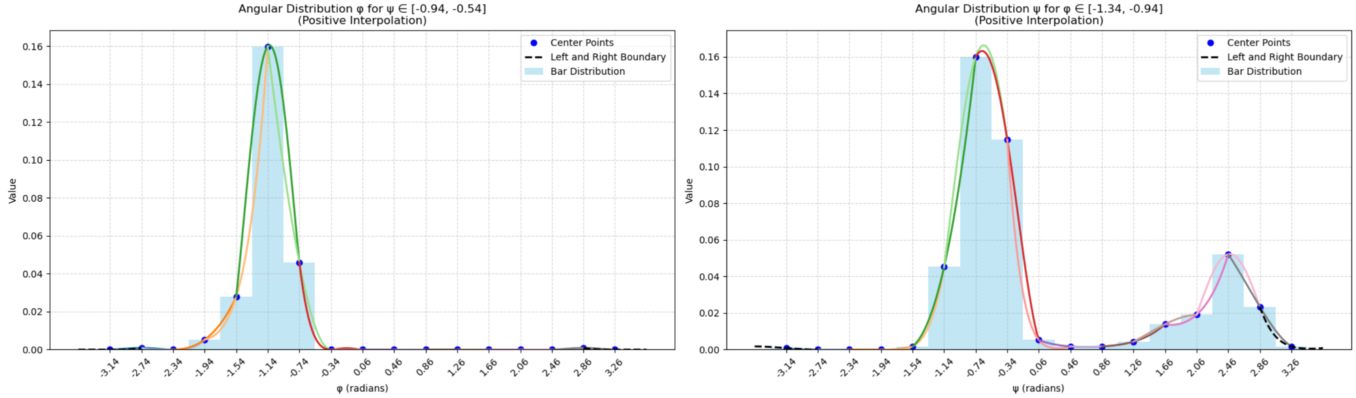

Figure 5: Example of Positive Parabolic Interpolations Sampling: Alanine-Alanine-Glutamate, resolution = 0.4

2.4 Torus Test

To compare the performance of the different methods, we employed a statistical test. Our objective is to assess whether the distributions of the generated samples match those of the original samples. We thus tested the null hypothesis H_0: both samples follow the same distribution, against the alternative hypothesis H_1: the distributions differ.

For this purpose, we used the Torus Test developed by Javier González-Delgado [1], a method specifically designed to compare distributions involving angular variables.

We chose not to delve into the theoretical details of this method, but will instead provide a general overview of its principles. The method consists of projecting the two distributions on several geodesics (straight lines adapted to the periodic geometry of the flat torus). These projections simplify the problem by reducing it to several one-dimensional comparisons. For each geodesic, the Wasserstein distance is calculated between the projected distributions. The distances calculated over all geodesics are then combined to form a single overall test statistic. To evaluate the statistical significance of the observed difference, a p-value is calculated, representing the probability of obtaining a test statistic as extreme or more extreme, assuming the null hypothesis is true, meaning that the two distributions are identical. A low p-value indicates that the observed discrepancy is unlikely under the null, suggesting a significant difference between the distributions. We applied the method using a randomly selected set of geodesics, which has only a minor influence on the resulting p-values.

3 Results and Discussion

3.1 Presentation of results

3.1.1 Random Sampling Methods

As shown in Figure 6, the distributions generated by the different methods can be observed for the specific case of the Alanine–Alanine–Glutamate composition, using a discretization with interval size 0.4 (for a fixed ϕ interval). To obtain a representative graph, 80,000 samples were drawn with each method. The stepwise sampling method clearly exhibits abrupt jumps between intervals. Similarly, the quadratic approaches also display discontinuities each time the interval changes, revealing their limitations. Only the linear interpolation method guarantees a continuous distribution, as expected. The spiked sampling method produces the least convincing distribution, as we anticipated.

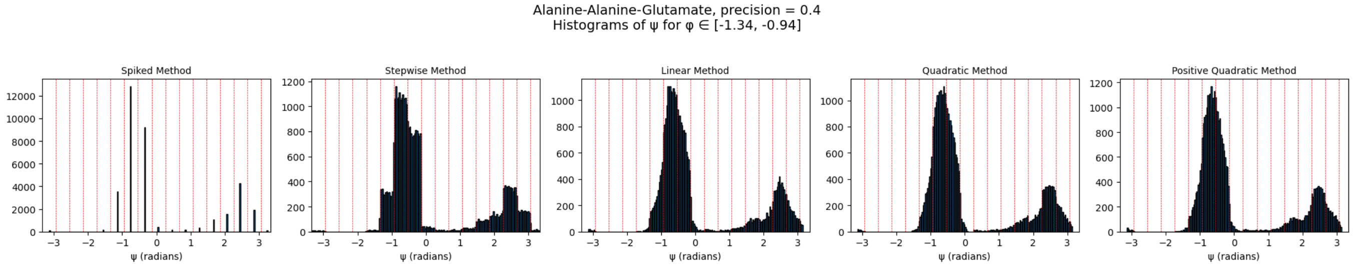

Figure 6: Example of distribution obtained

3.1.2 Rejection rate according to resolution and Sampling Method

As shown in Table 1, which presents the rejection rate of H_0 evaluated over 60 randomly selected samples, the rejection rate at the significance level $\alpha = 0.05$ increases as the interpolation precision becomes coarser. This trend is expected : lower interpolation resolution results in a greater loss of information from the original data, making it easier to statistically distinguish generated samples from real ones. In contrast, with higher resolution, the generated data becomes more similar to the original, resulting in lower rejection rates. However, excessively high resolution can result in overfitting, where the generated samples are nearly identical to the original data, limiting the model's ability to generalize. Among the tested methods, the spiked method exhibits consistently higher rejection rates, suggesting that it produces samples that diverge more noticeably from the original distribution compared to the other interpolation strategies.

Table 1: Rejection rates (in %) of the null hypothesis for each interpolation method at various levels of resolution ($\alpha = 0.05$).

resolution	**Spiked**	**Stepwise**	**Linear**	**Quadratic**	**Positive Quadratic**
0.1	10.94	0.00	0.00	0.00	0.00
0.2	42.19	1.56	0.00	1.56	0.00
0.3	62.50	6.25	3.12	3.12	1.56
0.4	84.38	14.06	7.81	9.38	10.81
0.5	92.19	29.69	17.19	18.75	12.50

3.2 Selection of Discretization resolution and Sampling Strategy

Based on the observed rejection rates of the null hypothesis H_0, our goal is to select the coarsest possible resolution that still yields low rejection percentages. In this analysis, we exclude the results from the spiked method, as they are not precise enough. An optimal choice appears to be a resolution of 0.2 or 0.3, as both lead to very low rejection rates indicating that the generated distributions remain statistically consistent with the originals. Additionally, a discretization step of 0.2 radians (11.46°) or 0.3 radians (17.19°) strikes a good balance between preserving essential distributional features and promoting generalization.

Regarding the methods, uniform sampling is the least computationally demanding and yields reasonably good results. However, both linear and quadratic interpolation remain computationally lightweight, and their relatively low rejection rates make them preferable. Among the three methods (linear, quadratic and positive quadratic), the rejection percentages are comparable, suggesting similar performance in preserving the statistical properties of the original distributions. Nonetheless, the positive quadratic interpolation is likely superior to standard quadratic interpolation due to the absence of negative values.

Linear interpolation provides a continuous solution within each fixed interval. However, in the 2D plane, continuity is only ensured along the ϕ and ψ axes; for example, in the case of a diagonal movement, continuity is not guaranteed in general. Quadratic approximation, on the other hand, exhibits continuity issues in all directions. It is only continuous within each fixed interval along the ϕ and ψ axes. However, it provides a more accurate estimate of the distribution within those fixed intervals. There is no significant difference in computation time between the two methods. Computing a degree-2 polynomial requires solving a system of three equations using Gaussian elimination, while linear interpolation involves computing two degree-1 polynomials, which amounts to solving two systems of two equations. Therefore, neither approach is clearly superior in terms of efficiency. The choice between them mainly depends on the desired level of continuity in the approximations.

3.3 Limitations of the Method in the Case of Central Proline

The null hypothesis H_0 is rejected in a small percentage of cases, indicating that the vast majority of generated distributions are statistically indistinguishable from the original ones. However, a few exceptions remain. For instance, when the central amino acid is a proline, the method occasionally results in higher rejection rates. We were unable to determine the precise cause of this behavior. Visually, the generated distributions still resemble the original ones, as shown in Figure 7. This raises the question of whether the rejections stem from limitations in the statistical test or from subtle inconsistencies in the generated samples that are not perceptible by eye. A few other distributions also show elevated rejection rates without any clear explanation. Nevertheless, the method performs very well across the vast majority of amino acid combinations.

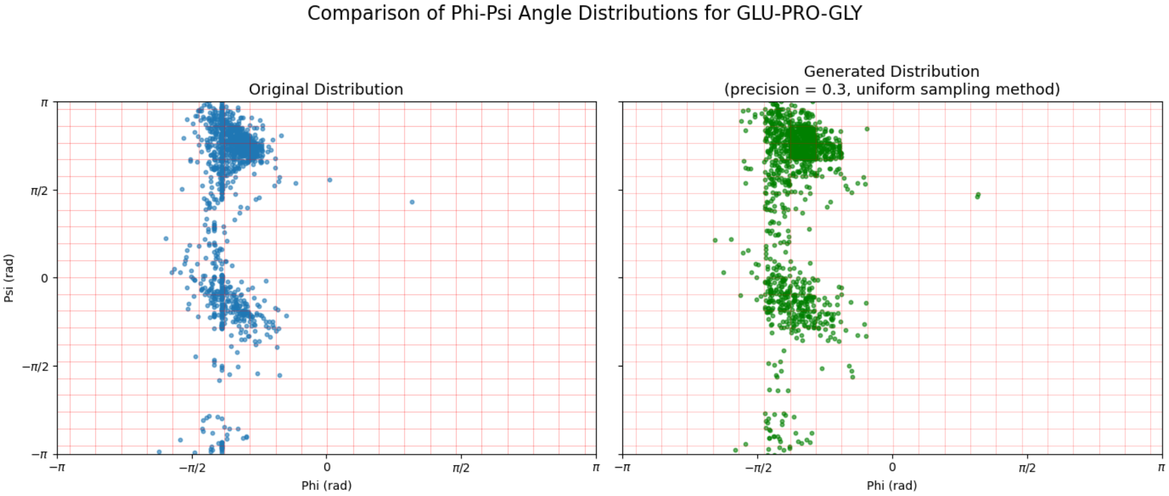

Figure 7: Comparison of real and generated distributions for the Glutamate–Proline–Glycine configuration.

4 Conclusion

The study of protein structures is central to life sciences. A detailed understanding of the dihedral angles ϕ and ψ represents a key challenge for modeling protein folding and designing new functional structures. In this context, we proposed a simple approach based on the discretization of measured samples, allowing us to efficiently represent angular distributions.

This method has several advantages: it is easy to implement, requires few computational resources, and remains flexible enough to adapt to a wide variety of amino acid combinations. Thanks to different sampling strategies (ranging from uniform sampling to quadratic interpolations) we were able to generate distributions very close to those observed in real data. The results of statistical tests, notably the torus test [1], confirm the relevance of our approach, with often high p-values, especially for linear and positive quadratic interpolations.

However, some limitations remain. For some specific cases, such as tripeptides with a central proline, the method shows less stable performance, although the precise cause could not be identified. In addition, interpolation methods exhibit discontinuities along the direction of displacement in the angular plane, which could affect the local sampling fidelity.

It is important to note that our method assumes independence between the angles ϕ and ψ within each fixed interval. Consequently, once the intervals are selected, the angles are sampled independently. While this assumption appears reasonable at low discretization resolutions, it remains a limitation of the method, as it is theoretically incorrect.

Despite these limitations, our approach provides a robust and lightweight solution for the analysis of angular distributions, and can serve as a basis for broader applications, such as the rapid generation of hypothetical protein structures or the initiation of models into more complex algorithms. A possible avenue for improvement could include dynamic adaptation of the resolution across regions of the angular plane. This method illustrates how a simple, well-calibrated solution can compete in effectiveness with more sophisticated approaches, while remaining accessible and reproducible.

Software Availability

All code developed for this project, including data processing and sampling methods, is available on GitHub at: `https://github.com/OctaveBorgard/Bio_stat_proteine`.

References

[1] J. Cortés J. González-Delgado, A. González-Sanz and P. Neuvial. Two-samplegoodness-of-fit tests on the flat torus based on wasserstein distance and their relevance to structural biology. *Electronic Journal of Statistics*, 2023. https://doi.org/10.1214/23-EJS2135.

[2] Maxim Shapovalov Rajib Mitra Michael I.Jordan Roland L. Dunbrack Jr Daniel Ting, Guoli Wang. Neighbor-dependent ramachandran probability distributions of amino acids developed from a hierarchical dirichlet process model. *PLoS computational biology*, 2010. https://doi.org/10.1371/journal.pcbi.1000763.

[3] Peter J. Stuckey Maria Garcia de la Banda Arthur M. Lesk Arun S. Konagurthu Piyumi R. Amarasinghe, Lloyd Allison. Getting 'al' with proteins: minimum message length inference of joint distributions of backbone and sidechain dihedral angles. *Bioinformatics*, 39, 2023. https://doi.org/10.1093/bioinformatics/btad251.

[4] Haiou Li, Jie Hou, Badri Adhikari, Qiang Lyu, and Jianlin Cheng. Deep learning methods for protein torsion angle prediction. *BMC bioinformatics*, 18:1–13, 2017. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-017-1834-2.