1 Methods

1.1 Overview

AF_ClaSeq is a novel computational framework for analyzing and classifying protein sequences based on their predicted structural properties using AlphaFold2. The method employs an iterative sampling approach combined with structural analysis and voting mechanisms to identify sequence-structure relationships. The framework consists of three main components: (1) Iterative Sequence Shuffling, (2) M-fold Sampling, and (3) Sequence Voting and Classification.

1.2 Iterative Sequence Shuffling

The iterative sequence shuffling process aims to explore the sequence space efficiently while maintaining structural diversity. For a given multiple sequence alignment (MSA):

Algorithm 1 Iterative Sequence Shuffling

[1] MSA file $\mathcal{M}$, number of iterations N, group size k, coverage threshold θ Filtered and shuffled sequence groups $S \leftarrow \text{FilterSequences}(\mathcal{M}, \theta)$ **for** $i \leftarrow 1$ *to* N **do**

$S_i \leftarrow \text{RandomShuffle}(S)$ $G_i \leftarrow \text{GroupSequences}(S_i, k)$ **for** *each group* $g \in G_i$ **do**

$\text{PredictStructure}(g)$ $\text{CalculateMetrics}(g)$ $S \leftarrow \text{FilterByMetrics}(G_i)$

The sequence filtering is performed using a coverage threshold θ:

$$\text{Coverage}(s) = 1 - \frac{\text{GapCount}(s)}{\text{Length}(s)} \tag{1}$$

1.3 M-fold Sampling

The M-fold sampling strategy implements a Bayesian-inspired approach to sample the sequence space systematically:

Algorithm 2 M-fold Sampling

[1] Filtered sequences S, fold number M, group size k Sampled sequence groups with structural predictions $G_0 \leftarrow \text{InitialRandomSplit}(S, k)$ **for** $i \leftarrow 1$ *to* M **do**

$G_i \leftarrow \text{CreateSamplingSplits}(G_{i-1})$ **for** *each group* $g \in G_i$ **do**

PredictStructure(g) AnalyzeStructure(g)

1.4 Structural Analysis

The framework employs multiple structural metrics for analysis:

1.4.1 TM-score Calculation

TM-score is calculated using the standard formula:

$$\text{TM-score} = \frac{1}{L} \sum_{i=1}^{L_{\text{ali}}} \frac{1}{1 + (\frac{d_i}{d_0(L)})^2} \tag{2}$$

where L is the length of the target protein, L_{ali} is the number of aligned residues, d_i is the distance between the i-th pair of aligned residues, and $d_0(L)$ is a length-dependent scale.

1.4.2 Domain Angle Calculation

The angle between domains is computed using their centers of mass (COM):

$$\theta = \arccos\left(\frac{\vec{v_1} \cdot \vec{v_2}}{|\vec{v_1}||\vec{v_2}|}\right) \tag{3}$$

where $\vec{v_1} = \text{COM}_{\text{domain1}} - \text{COM}_{\text{hinge}}$ and $\vec{v_2} = \text{COM}_{\text{domain2}} - \text{COM}_{\text{hinge}}$.

1.5 Sequence Voting and Classification

The voting mechanism implements a consensus-based approach:

Algorithm 3 Sequence Voting

[1] Structural predictions P, metric thresholds T, bin count B Sequence classifications **for** *each sequence* s **do**

$V_s \leftarrow \emptyset$ Initialize vote collection **for** *each prediction* $p \in P$ *containing* s **do**

$m \leftarrow \text{CalculateMetrics}(p)$ $b \leftarrow \text{AssignBin}(m, T, B)$ $V_s \leftarrow V_s \cup \{b\}$ $\text{classification}_s \leftarrow \text{MajorityVote}(V_s)$

The bin assignment for continuous metrics uses quantile-based thresholds:

$$\text{bin}_i = \left\lfloor \frac{(x - x_{\min})(B-1)}{x_{\max} - x_{\min}} \right\rfloor + 1 \tag{4}$$

where B is the number of bins, and x is the metric value.

1.6 Implementation Details

The framework is implemented in Python with the following key components:

- **Sequence Processing**: Efficient handling of MSA files in A3M format with specialized filtering and grouping operations
- **Structural Analysis**: Integration with BioPython for structure manipulation and TMalign for structural comparison
- **Parallel Processing**: Implementation of concurrent processing using Python's ProcessPoolExecutor for efficient computation
- **Visualization**: Matplotlib-based plotting utilities for structural analysis and voting distribution visualization

The implementation supports both single-metric and multi-metric analysis modes, with configurable parameters for each component of the pipeline.