NPRG070 – Research Project Report

Project title: Predicting Protein Folding in Reduced Alphabet Protein Sequences

Solver: Bc. Hugo Hrbáň

Supervisor: doc. RNDr. David Hoksza, Ph.D.

1 Introduction

Today, all proteins consist of 20 standard amino acids. However, during early stages of life's formation on Earth, over 4.5 billion years ago, it is hypothesized that only a subset of 10 of these amino acids were available [9]. We refer to these as early, prebiotic, or ancestral amino acids, and similarly we refer to proteins containing only early amino acids as early, prebiotic, or ancestral proteins.

This project was done in collaboration with Klára Hlouchová's research group (`https://khlab.org`) at the Faculty of Science (PřF UK), which focuses on protein evolution and the effect of the amino acid alphabet on protein structure, trying to answer the question whether contemporary proteins can be built using only the early amino acids.

This project was divided into two main parts. In the first one, we analyzed a dataset of protein sequences from an experimental assay, provided to us by Klára Hlouchová. We describe the properties of this dataset in the following section. In the second part of this project, we developed a method for translating a given protein sequence into the prebiotic alphabet by iteratively substituting the non-prebiotic residues, while keeping the protein structure as similar to the original structure as possible. We analyzed how the method performs on a small dataset of proteins with diverse folds, and finally designed a few candidate translations of a particular protein of interest, which will be experimentally created and analyzed.

1.1 Biological Background

In this subsection, we briefly introduce the main biological background and terms used in this work. This section is adapted from [5].

1.1.1 Proteins

Proteins are macromolecules crucial for all life on Earth. They are built from chains of amino acids connected together by peptide bonds that fold into complex three-dimensional shapes, and this structure gives them remarkable versatility. Proteins can act as enzymes that catalyze chemical reactions, structural components that provide support and shape, or signaling molecules that coordinate processes across cells.

1.2 Amino acids

Amino acids are the fundamental building blocks of proteins. There are over 500 amino acids in nature, however only 20 of them are the so-called standard amino acids which are encoded in the standard genetic code. Amino acids are characterized by having a central carbon atom (C_α) to which an amino group (–NH2), a carboxyl group

(–COOH), a hydrogen atom, and a variable side chain (R group) are attached. The properties of an amino acid are determined by its unique side chain, which can be, for example, polar or nonpolar, acidic or basic, hydrophobic or hydrophilic, etc. These properties affect the amino acid's role in proteins, and the protein's function as a whole. Each amino acid is assigned a unique three-letter and one-letter code [1].

Peptide bonds form when the carboxyl group of one amino acid reacts with the amino group of another, releasing a molecule of water as a byproduct. This process creates chains of amino acids linked by peptide bonds. A protein consists of one or more polypeptide chains. After the peptide bond formation, amino acids are often referred to as residues that are connected to the protein backbone [1].

1.2.1 Levels of protein structure

Structure of proteins can be described at the following four levels ([4]):

Primary structure (usually referred to as protein sequence) is the linear sequence of amino acids that form a protein. The length of a protein sequence can vary widely, ranging from just a few tens to several thousands of amino acid residues. The direction of the sequence matters, so the sequence of amino acid residues `MGEAK` is not the same as `KAEGM`. By convention, the sequence is written from the N-terminus to the C-teminus, where N-terminus is the end of the polypeptide chain with the free amino group (–NH2) and C-terminus is the free carboxyl group (–COOH). This choice is due to the way proteins are synthesized in cells during the process of translation, although this choice is somewhat arbitrary and not biologically significant.

Secondary structure refers to local conformations of the protein backbone. There exist two types of secondary structures:

– *Alpha helix*: backbone is twisted into a right-hand coil shape formed by hydrogen bonds between amide and carbonyl groups four residues apart in the sequence.

– *Beta sheet*: they consist of beta strands laid out next to each other and connected by hydrogen bonds between residues of adjacent strands. They can be parallel or anti-parallel, depending on the orientation of the beta strands.

While the secondary structure describes three-dimensional features of a protein, it can be described using a one dimensional representation, because we can assign each residue as being part of an alpha helix, beta sheet or a loop.

Tertiary structure is a three dimensional representation of the whole protein formed by the folding of its polypeptide chain. This structure is very useful for understanding the protein's biological function.

Quaternary structure of a protein describes the arrangement of multiple chains into a larger functional complex. Some proteins, such as hemoglobin, which carries oxygen in the blood, occur in nature as an assembly of several polypeptide chains. However, in this work we do not analyze any protein complexes with a quaternary structure.

Visualizations of all levels of protein structure can be found in Fig. 1 and are adapted from [3] and [2]. Note, that when we talk about protein structure, we usually refer to the tertiary structure (3D representation) and when we mention protein sequence we are talking about the primary structure (1D string).

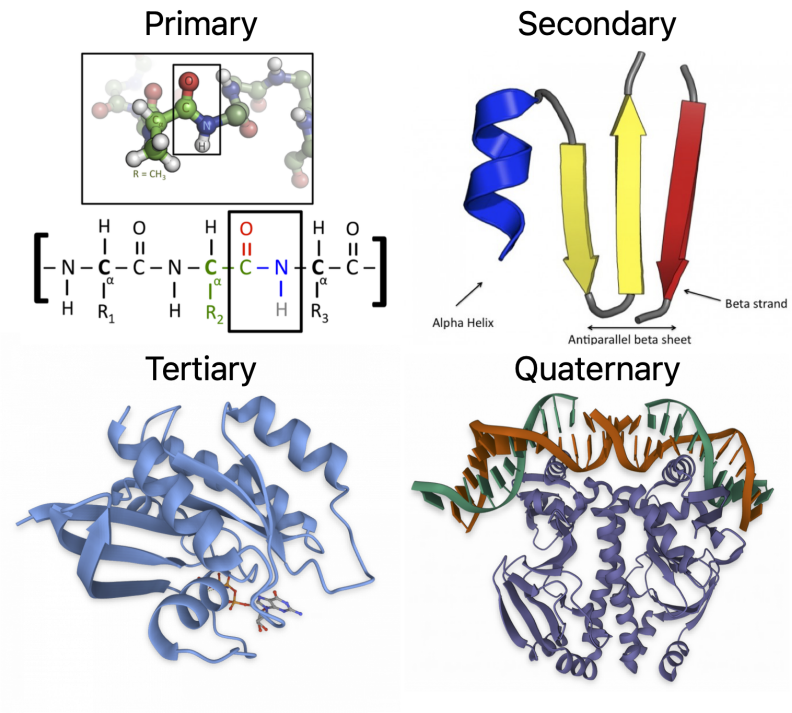

Figure 1: Example visualizations of primary, secondary, tertiary and quaternary protein structure

1.2.2 Relationship between sequence, structure and function

Genetic information about the protein sequence is stored in the DNA which is a long chain of nucleobases, where consecutive triplets of nucleobases, so-called *codons*, each encode a specific amino acid. During the process of protein synthesis, this information is used to produce the polypeptide chain of a protein, which then assumes a three-dimensional shape. This process is known as protein folding. Knowing the structure of a protein means knowing the coordinates of every atom in it.

The function of a protein in the organism is closely linked to its structure, which determines how it interacts with other molecules ([1]). Binding sites, which are parts of a protein where a ligand can bind, are usually highly specific, allowing only particular ligands to bind to it. The ligand can be for example an ion, small molecule, another protein, RNA or DNA.

2 Experimental Data Analysis

The dataset was originally created by our colleagues to investigate the properties of ancestral proteins and what makes them work well in practice. In order to get an understanding of how to generate new ancestral sequences which are expressed and properly folded, they created thousands of sequences and experimentally measured their properties. Our goal was to analyze this experimental data and to develop a method for estimating whether a new sequence is going to be functional in practice or not. This would make future experiments more efficient, since we could filter out sequences, which we consider unlikely to be successful designs.

The dataset we analyzed contains artificially generated protein sequences, all of which have the same length, and residues were sampled randomly with a given probability for each position, so it is a so-called combinatorial library ([9]). In the original experimental assay, for each of the sequences, a gene was inserted into an E. coli cell to develop and all of the cells were sequenced to obtain the reference library and a number of reads for each sequence. Subsequently, after an incubation period to allow the cells to grow and become enriched, a dual-reporter system based on fluorescence-activated cell sorting (FACS) was used, which measures two fluorescence signals: GFP activity (green fluorescent protein), which indicates whether the protein in a particular cell is properly folded, and mScarlet activity, which indicates the level of protein expression in that cell. Based on certain thresholds, cells, and their sequences, were split into three subsets: misfolded but expressed, properly folded and expressed, not expressed (figure 2). For simplicity, in the following text, we will refer to these parts of the dataset as *misfolded (high GFP)*, *folded (low GFP)* and *not expressed (low mScarlet)*, respectively. Then, all three of these subsets were sequenced, and the number of reads was obtained for each protein sequence. The ratio between the number of reads in the sorted subset (obtained from the FACS sorting) and the number of reads in reference library is referred to as the *enrichment*, or *fold change*. Sequences with a high enrichment value are considered to belong to the particular subset with a high likelihood, and not just by chance, e.g. if a sequence has a high enrichment, let's say 10.0, in the 'folded' subset, it appeared 10-times more often in the sorted subset than in the reference library, we consider this sequence a true positive, and that protein is actually properly folded in the cell. We also observe many sequences with a low enrichment value (< 1), which are almost certainly false positives.

Figure 2 part A shows the output of the biological experiment: each dot represents a cell, and we see three

predetermined thresholds of GFP and mScarlet values used for the FACS sorting of the cells, which were later enriched and sequenced. Part B of figure 2 shows the sequence logo of the whole dataset, size of the letter is proportional to observing that residue in that particular position.

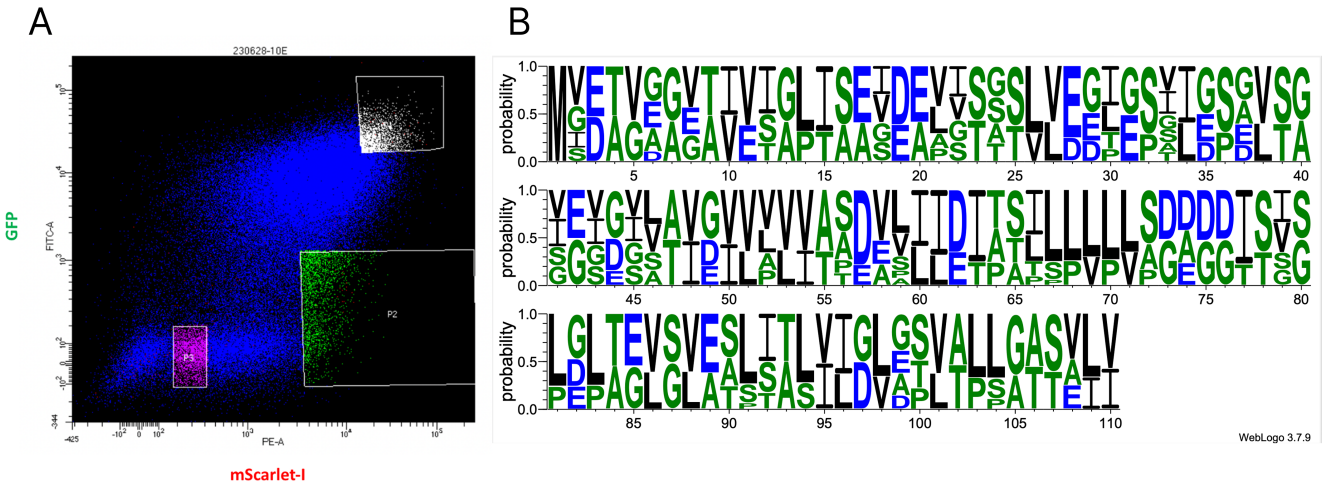

Figure 2: A: Diagram of the FACS sorting. B: Sequence logo for the whole dataset.

2.1 Dataset Exploration

The first question we wanted to answer, is whether there is any statistically significant difference between different parts of the dataset. Because all sequences have the same length, and are highly conserved, we consider them an MSA (multiple sequence alignment) without any insertions or deletions and we analyzed each position independently to get an overall idea of which, if any, positions are important for expression and for proper protein folding.

We filtered the dataset to only include sequences with enrichment value of at least 10. This threshold was selected based on a discussion with our collaborators who carried out the experiment. After applying this filtering, we obtained 1990, 886, and 695 sequences for the folded, misfolded, and not expressed parts, respectively. This filtration criteria was also applied in all following analyses in this paper.

For each position in the MSA, we counted the number of occurrences of all amino acids appearing in that position, and performed a χ^2 test of homogeneity, which tests the null hypothesis that the categorical outcome has the same distribution in the two populations. In our case, the categorical outcome was the type of amino acid, and the two populations are expressed sequences compared to not expressed ones, and folded compared to misfolded proteins.

For investigating protein expression, we merged the *folded* and *misfolded* parts, since both of them had high mScarlet values and therefore were *expressed*. We identified 39 positions in the alignment, where the p-value was lower than 0.05, as shown in figure 3, which shows p-values on a $-\log_{10}$ scale, such that lower p-values appear higher. Interestingly, we saw the lowest p-value on the very last position in the alignment, where in case of expressed proteins there is a 65.9% chance of observing I (isoleucine) and 33.8% chance of V (valine) amino acids, whereas not expressed sequences had an I with a probability of 40.3% and a V with probability 59.1%. Other amino acids

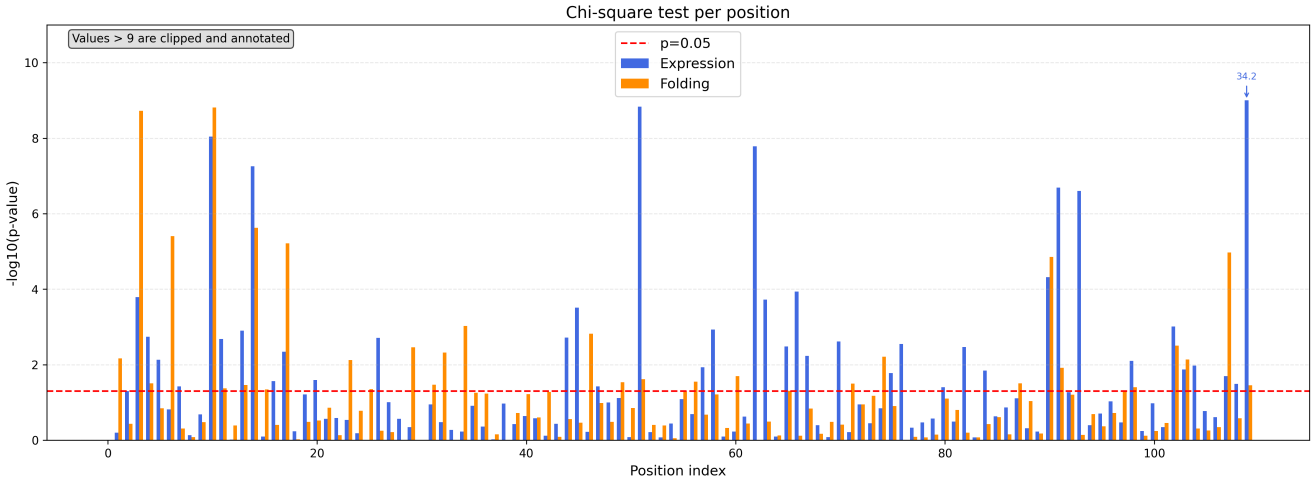

Figure 3: Chi-square test of homogeneity per position, for both expression and folding data. y-axis shows negative logarithm of the p-value.

appeared in that position with much lower probabilities. However, this phenomenon might have been just an experimental artifact, since it occured in the last position.

For protein folding, we looked for differences in the *folded* and *misfolded* parts, ignoring the *not expressed* sequences. We found 33 positions where the amino acid distributions were statistically significant (p-value < 0.05). Notably, 16 of the positions were statistically significant in both tests.

We found that although the sequence logos may appear similar, statistical analysis reveals positions where the distributions differ significantly.

2.2 Sequence Similarity Search

Firstly, we wanted to find out if there were any natural sequences similar to the ones in our artificial dataset.

We used MMseqs2 [7] to search for sequence homologs in UniRef90 [8], containing nearly 200 million sequences, as of November 2024. We found no hits for any of the sequences in our dataset. This was most likely due to the dataset being artificially constructed, containing only early amino acid residues, and being quite conserved, with all sequences having high sequence identity to one another.

2.3 Sequence Clustering

We investigated whether proteins with high enrichment tend to be more similar in the sequence space to other proteins with high enrichment and vice versa. In this analysis, we included all sequences regardless of the enrichment values. First, we clustered the sequences at a certain sequence similarity threshold, which we needed to determine. We used MMseqs2 to perform the clustering at 8 different thresholds, and we show the overall distribution of cluster sizes in figure 4 for all thresholds. Based on the number of clusters and average cluster size, we decided to use the clustering at 55%, because lower values created too few clusters and higher values caused the average number of sequences in a cluster to be too low (less than 3).

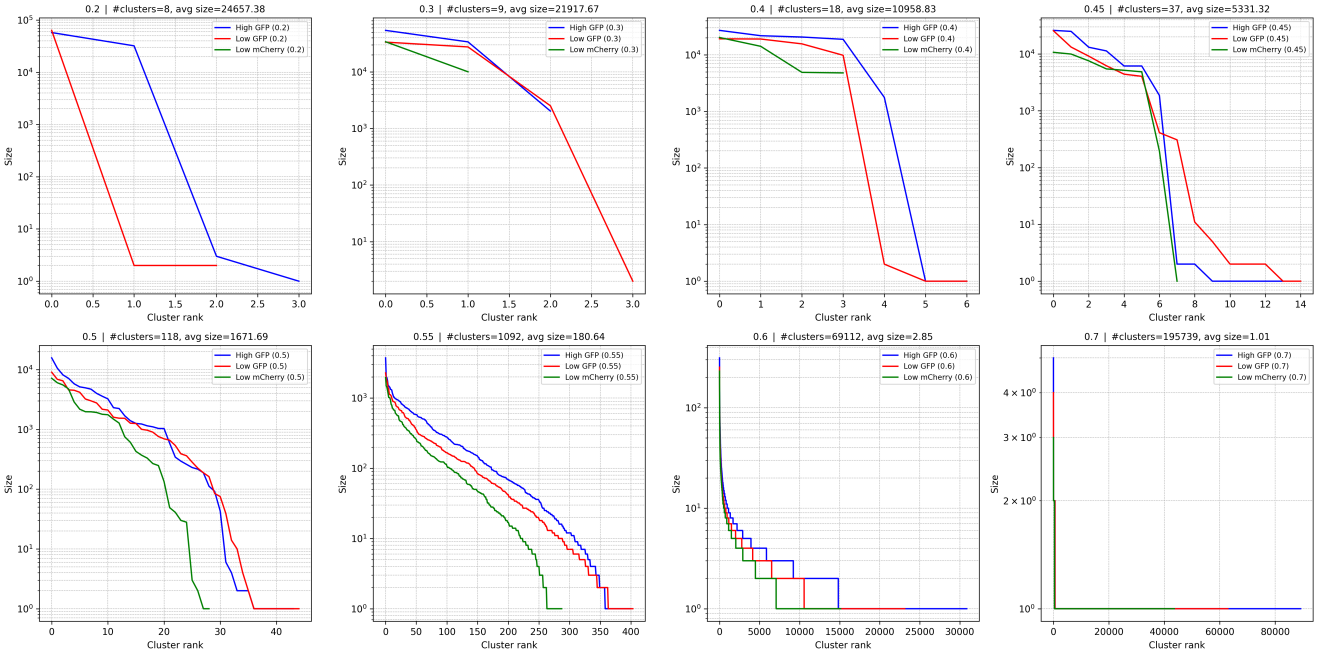

Figure 4: Distributions of cluster size after clustering at various sequence similarity thresholds

After clustering each of the three subsets of the dataset at the 55% threshold, we visualized the relationship between cluster size and average enrichment of sequences in a cluster, see figure 5. We computed the Pearson correlation coefficient for all three subsets but found no meaningful correlation ($\rho = -0.11, p = 0.02$), ($\rho = -0.06, p = 0.23$), ($\rho = 0.03, p = 0.60$) for the *folded*, *misfolded* and *not expressed* parts, respectively.

Furthermore, by manually inspecting the scatterplots in figure 5, we found no clear outliers which would correspond to larger clusters with an above- or below-average enrichment.

2.4 Nearest Neighbors Analysis

Since we found no evidence of highly enriched proteins belonging to the same clusters, we decided to only look at the nearest neighbor sequences. We investigated the relationship between enrichment of a protein and enrichment of its nearest neighbor. Since all sequences in our dataset have the same length, we used Hamming distance, and sequence identity between two sequences is defined as the percentage of positions which contain identical residues.

To get an overall idea about how conserved the dataset is, we visualized the distribution of sequence identity to the nearest neighbor in a histogram in figure 6. We notice a clear peak around 66%, and also a smaller peak at 96%. Even though by visualizing the sequence logos the sequences may appear very conserved, they are quite well spread out in the sequence space.

We also visualized the relationship of the enrichment of a sequence and the enrichment of its nearest neighbor, but found virtually no correlation. Figure 7 shows the scatterplot for each of the three subsets and Pearson r correlation coefficient with the p-value. Visually, we noticed that all three sets contain some outliers having high values in both x and y axes, the vast majority of points has low x value (below 10), low y value, or both, and not many points are close to the diagonal.

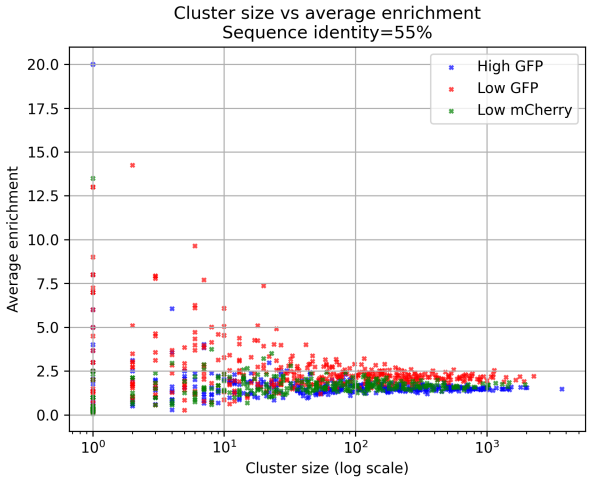

Figure 5: Relationship of cluster size and average enrichment, for all three parts of the dataset

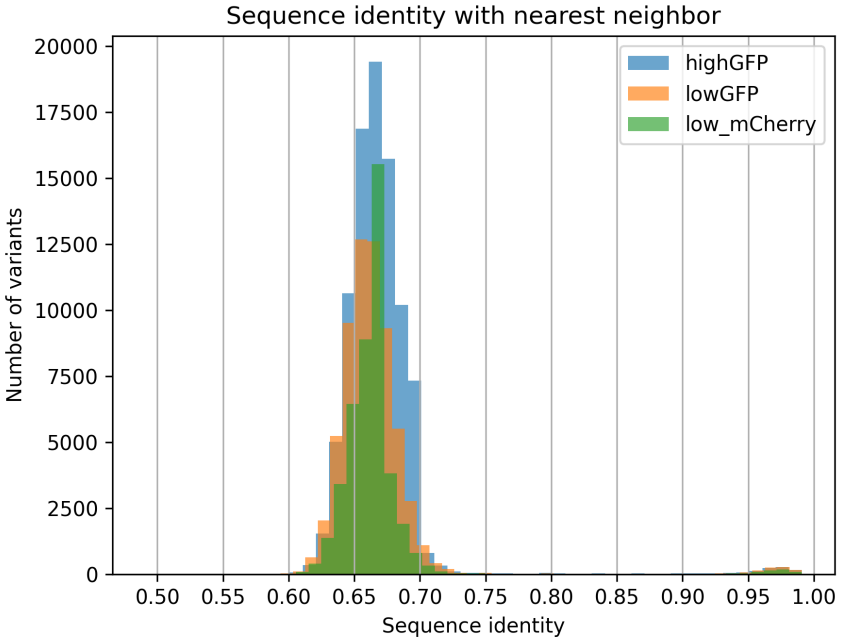

Figure 6: Histogram of sequence identity to the nearest neighbor for all three parts of the dataset

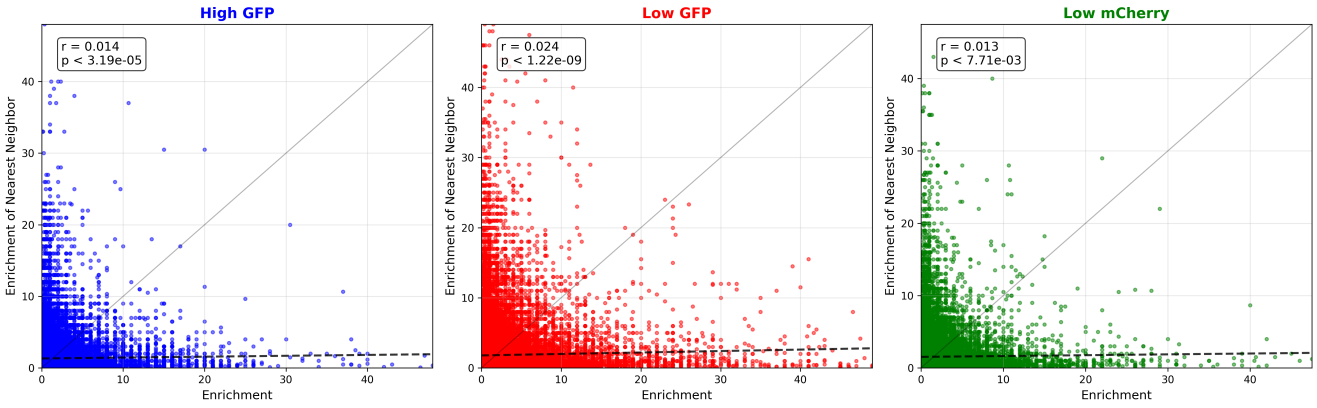

Figure 7: Enrichment of a sequence compared to the enrichment of the nearest neighbor

2.5 Structure Similarity Search

Since MMseqs2 found no hits when searching against sequence databases, likely due to our queries having a reduced amino acid alphabet, we turned to Foldseek to find homologous structures instead in AFDB50 [10]. Foldseek first converts a protein structure into a linear sequence of tokens from the 3Di alphabet, which represent the protein's conformation, and then uses the MMseqs2 algorithm to search for similar structures. The advantage of this is that by converting to a simple sequence representation we can search the database very quickly, but most importantly, our queries now use the same alphabet as the database and not just a subset of it as was the case with sequence similarity search using MMseqs2.

Along with the original dataset we received structures predicted with ESMFold [6] for each sequence in the *folded* and *misfolded* parts of the dataset. Even though we could have predicted structures of the *not expressed* proteins, it would be irrelevant since they were not expressed experimentally anyway, so their possible structure is irrelevant for this analysis. Unfortunately, all of the structures were predicted with very low pLDDT (predicted local distance difference test – ESMFold's confidence level). The value of pLDDT of a protein was calculated as the average across all its residues, and it ranges from 0 to 100. Across the entire dataset, the average pLDDT was 31.7 for both the *folded* proteins, and the *misfolded* ones. For reference, pLDDT < 70 is classified as low confidence. Notably, even after taking only structure predictions of proteins with enrichment higher than 10, the average pLDDT was 31.8 for the *folded* sequences and 31.7 for the *misfolded* sequences. It is expected to have low pLDDT for proteins that we experimentally know are misfolded, but low pLDDT for proteins which we experimentally know are folded implies poor performance of the model on these proteins.

Still, we decided to use Foldseek to find if any sequences in the AlphaFold database clustered at 50% identity (AFDB50) had similar structures, and if the properties of returned hits correlate in any way with the enrichment of the query sequences. In figure 8, we show the relationship between enrichment of a query sequence, number of returned hits for that query, and lowest E-value out of the returned hits. We saw that the queries with high number of hits (above 100) all had relatively low enrichment and there were no outliers. Similarly, we found no positive correlation between the lowest returned E-value and the query sequence enrichment. Notably, most E-values were

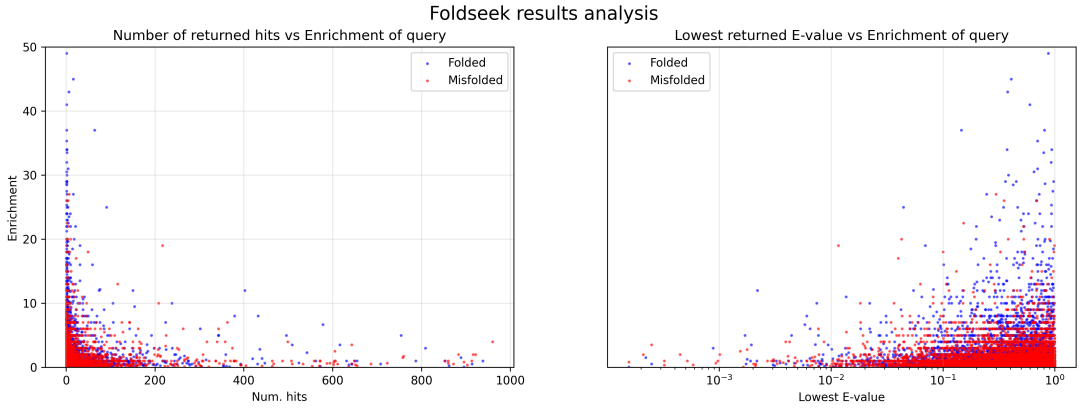

Figure 8: Distribution of Foldseek hits and their properties

fairly high and we only got a handful of examples with values lower than 10^{-2}, which could have also occurred just by chance.

As expected, we found no pattern in the outputs of Foldseek due to low confidence of predicted structures we used as queries.

2.6 Predicting sequence expression and foldability

In this section, we trained ML models for predicting whether a given input sequence is expressed or not expressed and whether it is folded or misfolded. It is important to note that, as our colleagues explained, the class of proteins with low mScarlet values (*not expressed* sequences) serves as sort of negative validation, since we know these sequences are definitely not expressed, so we expect the task of predicting if a sequence is expressed or not to be easier than predicting whether a sequence is properly folded.

2.6.1 Simple Statistical Sequence Classifier

We observed that there is a meaningful difference between the distinct parts of our data, we trained simple statistical classifiers for predicting expression or folding of a given input sequence. We used a naive Bayes classifier model to predict the probability that a given sequence is expressed (or properly folded in case of the other model). The model assumes that different positions in the sequence are independent. For a sequence of residues $s = r_1, ...r_N$, it is defined as

$$P(C_0|s) = P(C_0|r_1, ..., r_N) = \log(\frac{\pi_0}{1-\pi_0}) + \sum_{i=1}^{N} L(i, r_i)$$

, where L is the log likelihood ratio of residue frequencies on a given position $L(i, r) = \log(\frac{P_i(r|C_0)}{P_i(r|C_1)})$, π_0 is the prior probability of a sequence having label 0, and $P_i(r|C)$ denotes the probability of observing residue r on position i if the label is C, e.g. *expressed*, *not expressed*, etc. We used 80% of sequences as training data to train this naive Bayes classifier and fit a logistic regression on its outputs in order to get a model to predict the final label. Figure 9 part A shows the ROC curves for expression prediction and folding prediction model. For model predicting protein expression, the area under ROC curve (AUROC) is 0.71, Matthew's correlation coefficient (MCC) is 0.13, and for

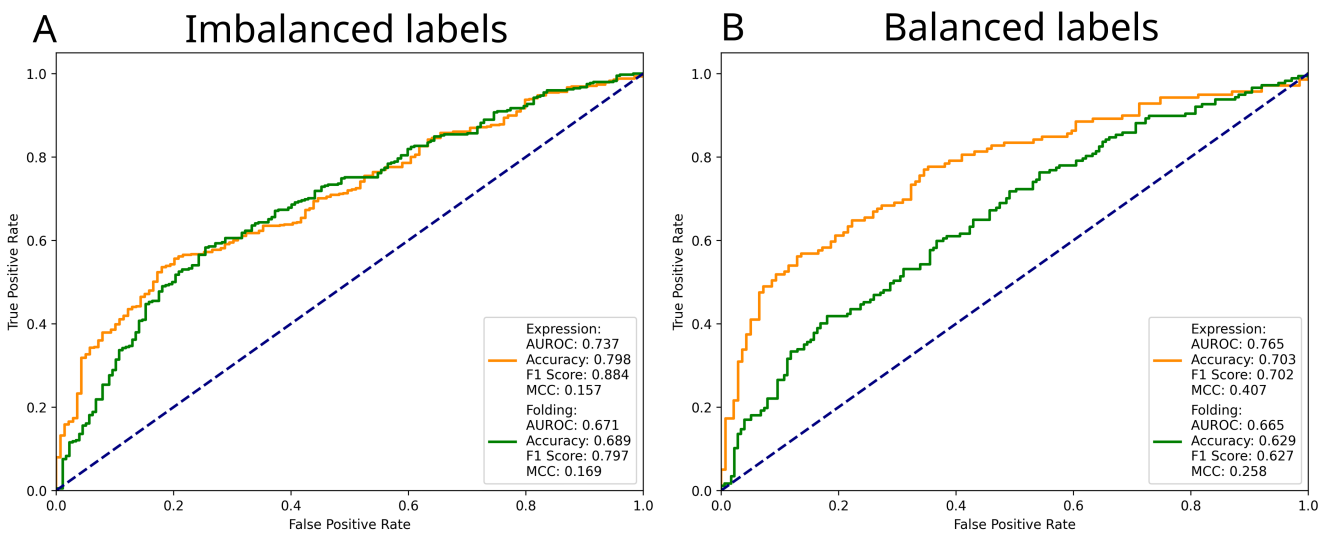

Figure 9: ROC curves for expression and folding prediction. A – imbalanced dataset (filtering by enrichment only). B – balanced dataset

the model to predict whether a protein is properly folded, AUROC is 0.65 and MCC=0.17. Interestingly, when we used a dummy model to only predict the label of the most frequent class, we achieved accuracy and F1 score slightly better than these two models. Based on AUROC and MCC being better than random, we saw that there is some signal in the data. It is important to note that due to class imbalance, AUROC can be artificially high, so MCC is a more reliable metric, which is in our case it is only slightly better than random.

Interestingly, when we took only top expressed sequences, so that there was an equal number of expressed and not expressed sequences in the training data, we got much better results in terms of MCC. For predicting protein expression, we got MCC=0.41, and for folding prediction MCC=0.26, as shown in part B of figure 9. This is of course because we only take the very top expressed sequences (effectively filtering enrichment ≥ 19.0).

2.6.2 Deep Learning Models

The model in the previous section assumed positions to be independent. In reality, amino acids don't act alone: the three-dimensional conformation of a protein brings sites into contact, and function often depends on specific combinations of residues. Thus, a change at one position can change what works at another, creating correlations in the MSA. To reflect this, we move from a position-independent model to training deep learning models that can capture dependencies between positions.

We trained several different architectures from scratch: LSTM with token embeddings, LSTM with token and positional embeddings (dim=256), VGG-style 1D CNN, with a linear head for binary classification, and a ResNet-style 1D CNN with a linear head for binary classification.

We also fine-tuned ESM-2 ([6]) with a sequence classification head for binary classification and experimented with both using the CLS token embedding and mean across all residue embeddings as input for the classification head, since the CLS token embedding was untrained in the original model. ESM-2 is a pre-trained transformer

Model Type	Expression prediction		Folding prediction	
	AUROC	**MCC**	**AUROC**	**MCC**
LSTM	0.60	0.20	0.55	0.10
LSTM (+ pos. emb.)	0.58	0.18	0.58	0.17
CNN (VGG-like)	0.59	0.20	0.59	0.18
CNN (ResNet-like)	**0.61**	**0.22**	0.59	0.18
ESM-2 (8M, CLS token)	0.60	0.21	0.58	0.17
ESM-2 (8M, mean embedding)	0.56	0.14	0.60	0.20
ESM-2 (35M, CLS token)	0.58	0.19	**0.61**	**0.22**
ESM-2 (35M, mean embedding)	0.50	0.00	0.50	0.00

Table 1: Performance of various model architectures on tasks od predicting whether a sequence is expressed or properly folded

encoder-only model trained on UniRef50 ([8]). We fine-tuned the 8M and 35M parameter checkpoints of ESM-2. We attempted to freeze the pre-trained ESM-2 weights and only optimize the weights of the classification head, but with this approach the model did not train at all and resulted in predictions with MCC of 0.0, since it always predicted the most frequent class.

After initial manual hyperparameter tuning, we performed grid search over batch size and initial learning rate. We trained all models for 50 epochs, we used the Adam optimizer with default PyTorch parameters. In table 1 we show the best model for each model type, selected based on MCC values on unseen test data, for both tasks of expression and folding prediction. We show the AUROC and MCC values. It is important to note that in all cases, the models were very tricky to train, the test loss went up very quickly, F1 score was not better than random, and accuracy either went down rapidly or stayed at the baseline value. We see that even the best models have very low MCC and AUROC values.

Overall, we conclude this dataset was very hard to train on and we found no model that would work well for these tasks.

3 Translation into the Early Amino Acid Alphabet

In the previous section, we analyzed a large-scale dataset of thousands of prebiotic protein sequences. The ESMFold predictions we used in the analysis were of low confidence, but we hypothesize the reason for that was the way the artificial dataset was generated and contained very conserved sequences. The structure module of ESMFold takes as input continuous amino acid embeddings from ESM-2 language model, so the fact that the input sequence contains only early residues shouldn't matter. Predicting early protein structures with high confidence would suggest that the early alphabet can create the folds we see in today's proteins.

In this section, we describe a method for translating a protein sequence from the standard alphabet to the early alphabet by iteratively substituting residues until the sequence only contains early residues, while optimizing RMSD (root mean square deviation) between ESMFold prediction of the translated sequence and the original structure,

and at the same time optimizing ESMFold confidence (pLDDT). We evaluated the method against a simple baseline using a set of proteins with diverse folds. Lastly, we focused on a single protein structure from which we selected a few top candidates that will be tested experimentally.

3.1 Algorithms

We describe two algorithms: the *greedy* (baseline) algorithm, and the *clustering + beam search* algorithm. Both of them take as input a PDB structure, which we refer to as the reference structure, and make mutations in its underlying sequence.

The first approach is a simple *greedy* baseline. In each step, we randomly select a position in the sequence which contains a *late* amino acid. For each of the 10 possible *early* amino acids, we substitute the *late* amino acid in the selected position with the *early* one, and we generate the structure with ESMFold. We superimpose each of the 10 predicted structures onto the reference structure, compute the RMSD, and select the variant which has the lowest RMSD. We repeat this until all amino acids in the sequence are *early*.

The *greedy* method doesn't take into account coevolution – when two or more residues get mutated simultaneously, because they interact with each other structurally. Making coevolutionary mutations is the key to maintaining the overall protein structure. To this extent, the improved algorithm first clusters nearby late amino acids and then performs beam search within each cluster, as a heuristic to explore the combinatorial space of mutations, approximating coevolution.

We first create a contact map based on C_β coordinates (C_α for glycine) of all residues set at 7 Å. The contact map is a binary matrix A, where $A_{i,j} = 1$ if the distance between C_β atoms of residues i and j is less than 7 Å, and 0 otherwise. Since we only want to mutate positions with late residues, for each position i where i contains an early residue, we set $A_{i,*} = A_{*,i} = 0$. Now, we have an incidence matrix of late residue coordinates and we perform bottom-up hierarchical partial linkage clustering. We start with each residue in a cluster of its own. In each step, we select two clusters and count the number of edges between them. If the ratio between the number of edges and the number of possible edges (product of the two clusters' sizes) is over a predetermined threshold (we use 0.5 as the default), we merge the two clusters. We repeat this process until we can no longer merge any two clusters. At the start of each run of the clustering algorithm, we randomize the order of the singleton clusters, which influences the order of the merges and results in non-deterministic final clustering. Partial linkage clustering is a mix between single and complete linkage clustering. We use it because single linkage resulted in one cluster containing all residues and complete linkage rarely resulted in clusters with more than two members.

In the second stage of the algorithm, we translate each cluster independently using beam search. We start with a list of candidate sequences containing each of the 10 possible mutations on each possible position. Meaning that if the cluster size is 3, the initial list of candidates contains 30 sequences (we mutate one position at a time). For each candidate sequence, we predict its structure, and compute RMSD to the reference. By default, we run ESMFold with 0 recycles, since this massively influences inference time. Optionally, we factor in a term for pLDDT, since we want the sequence to be translated with high confidence. The score for a candidate is computed as $s * \text{pLDDT} - \text{RMSD}$, where s is the pLDDT scaling factor. We keep w candidates with the highest scores for the next step of the search. We refer to w as beam width. Again, for each of the top w candidates, we explore all possible mutations on all

remaining unmutated positions. So if there are now 2 remaining positions to be mutated, the new candidate list contains $w * 10 * 2$ sequences. We repeat these steps until all positions in the cluster contain early residues, and we do the beam search for all clusters in a random order, yielding a protein containing only early amino acids.

3.2 Dataset

We benchmarked our algorithms on a dataset containing 10 protein structures from the PDB, with the following IDs: 1FE4, 1IV0, 1J6Q, 1QZM, 1U5K, 1V43, 2HEO, 2JA9, 2Z1C, 6C2U. Nine of these were used in an ongoing, yet unpublished, research at K. Hlouchová's group. The last structure is a P-loop containing protein, which we will analyze in more detail in a following section, since our designs of this protein structure are going to be validated experimentally. Although the dataset is small, it is structurally diverse, containing distinct folds. Table 2 shows length of each sequence in the dataset, and properties of the initial prediction of ESMFold – predicting structure directly from the underlying sequence, which should ideally be identical to the reference structure. We notice that for most of the sequences the RMSD is quite low (below 2 Å), and TM-score (template modeling score) high (≥ 0.9), but for a few sequences the predictions are quite off, and unfortunately the worst prediction by RMSD is for the design candidate 6C2U.

PDB ID	**Length**	**Properties of initial prediction**		
		RMSD (Å)	**TM-score**	**pLDDT**
1FE4	68	0.72	0.95	92.6
1IV0	98	2.95	0.69	79.7
1J6Q	136	1.98	0.80	77.0
1QZM	94	0.38	0.99	91.4
1V43	372	1.39	0.97	89.9
2HEO	59	0.67	0.96	84.7
2JA9	175	1.11	0.96	88.0
2Z1C	75	1.71	0.85	85.5
6C2U	115	3.28	0.72	64.7

Table 2: Properties of the dataset

3.3 Results

For each structure in the dataset, we ran each of the two translation algorithms 20 times with different random seeds to get an overall idea about how each method performs. Figure 10 shows box plots for each protein and each translation method, including the minimum, median and maximum. Table 3 shows the median values for RMSD, pLDDT and TM-score. For the clustering + beam search algorithm, we used the default parameters for clustering as described above, beam width $w = 5$, pLDDT weight $s = 0$. It is important to note that the translation process is quite time- and resource-consuming. For the longest protein, 1V43, containing 372 residues, the greedy algorithm runs in 20 minutes, meanwhile the improved algorithm takes 85 minutes per translation on an NVIDIA H100 GPU.

We saw that, as expected, for most proteins the clustering + beam search algorithm has lower RMSD both for the median and the minimum. We also noticed that the distributions of its outputs were less spread out, so the algorithm is more robust, which is what we hoped to achieve by keeping coevolution in mind. During the greedy algorithm, for some proteins, the RMSD sometimes rapidly increased after a single mutation, suggesting that residue had some structural importance.

Overall, we saw that for most of the structures we are able to successfully translate the sequence to the early alphabet and keep the (predicted) structures quite similar. Nine out of the ten proteins have at least one translation with RMSD better than 2 Å compared to the reference.

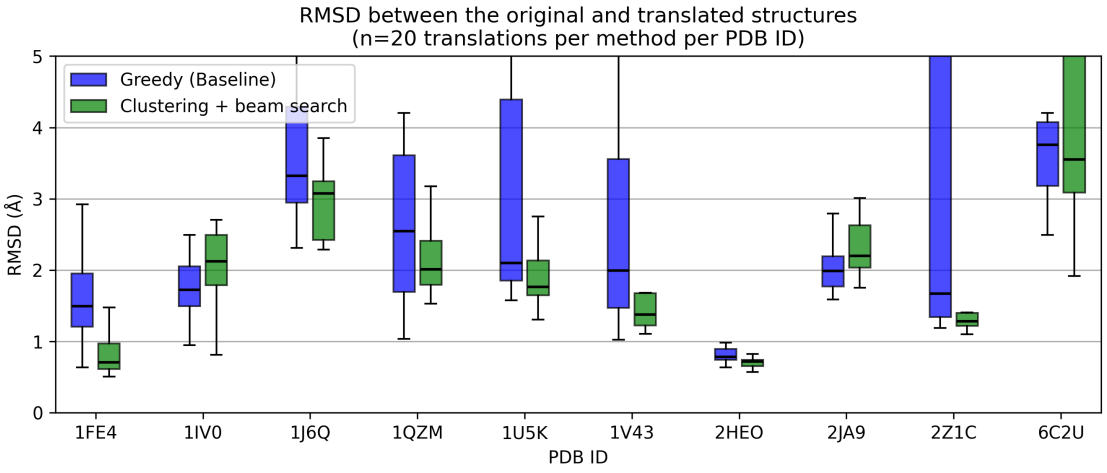

Figure 10: Distributions of RMSD, comparing two translation algorithms

PDB ID	Greedy (Baseline)			Clustering + Beam Search		
	RMSD (Å)	pLDDT	TM-score	RMSD (Å)	pLDDT	TM-score
1FE4	1.50	66.2	0.86	**0.71**	**92.2**	**0.95**
1IV0	**1.72**	75.2	0.68	2.12	**78.7**	0.68
2HEO	0.78	89.2	0.94	**0.72**	**92.9**	**0.95**
1J6Q	3.32	62.5	0.78	**3.08**	**82.9**	**0.82**
1QZM	2.59	67.2	0.85	**2.08**	**70.0**	**0.89**
1U5K	2.10	74.4	0.85	**1.76**	**78.8**	**0.88**
1V43	1.99	84.2	0.87	**1.40**	**86.8**	**0.89**
2JA9	**1.99**	67.5	**0.90**	2.20	**70.2**	0.88
2Z1C	1.67	85.8	0.86	**1.28**	**89.6**	**0.87**
6C2U	3.76	54.1	0.73	**3.54**	**62.7**	**0.75**

Table 3: Comparing results of two translation algorithms (n=20 per method per protein). Values shown are the medians.

3.4 Design Candidate – 6C2U

Our final goal was to apply the translation algorithm on a particular protein of interest, a P-loop (Walker-A motif) containing protein, with ID 6C2U from the PDB. The P-loop is an ATP binding site and our collaborators are interested to see whether even after translating the protein into the early alphabet, the protein can still bind ATP. The P-loop can be characterized by a regular expression GxxGxGK[TS], where x denotes any amino acid, brackets denote that one of the contained amino acids is in that position and remaining letters denote particular amino acids. Out of these, the only late amino acid in the P-loop is lysine (K).

We used the *clustering + beam search* approach, where for the clustering we used the default parameters again, 7 Å for the contact map threshold, 0.5 for partial linkage clustering threshold. We investigated how beam width w and pLDDT weight s influence the final outputs. Part A of figure 11 shows results of final RMSD and pLDDT values when we varied the parameter s, and fix $w = 10$. Interestingly, for $s = 0$, we see the widest distribution of RMSD, with the lowest minimum overall, but also the largest median and maximum. However, looking also at the scatterplot, the pLDDT values are all under 70. We saw the lowest median RMSD for $s = 10$, so we proceeded with the next experiment where we set $s = 10$ and varied beam width w. Part B of figure 11 shows the results. In terms of final RMSD, beam size did not seem to have much impact, but looking at the top candidates in the scatterplot, we saw the translations with the highest pLDDT were designed with w of 10 or 20. Notably, we saw that many of our predictions are better in both metrics than the initial prediction, also shown in scatterplots in figure 11, which is a success showing that the prebiotic alphabet can support modern folds, and that ESMFold can make predictions on this class of proteins.

We also tried a modification of this algorithm, where after clustering the late residues, we assign each early residue to the nearest cluster and therefore also potentially mutate the early residues in the beam search step. As can be seen in figure 12, this approach was not any better than the original one in terms of pLDDT, which is crucial for successful designs.

We also tried the approach of having just one cluster with all late residues in it. This approach is slow and deterministic, and for parameters $w = 5, s = 10$, resulted in a design with RMSD 2.2 Å, which is one of the lowest overall, but not a great confidence in the prediction, pLDDT = 69.0.

After discussing these results with our colleagues, they are looking for top 5-10 designs with the pLDDT > 80, and RMSD ideally lower than the initial prediction. This threshold for pLDDT is based on their previous experience with synthesizing this type of proteins.

4 Discussion

The next step in this work is for our colleagues to evaluate the top design candidates in detail, which includes visually inspecting the designed structures and analyzing the P-loop binding site in more detail, and later of course experimentally synthesizing them.

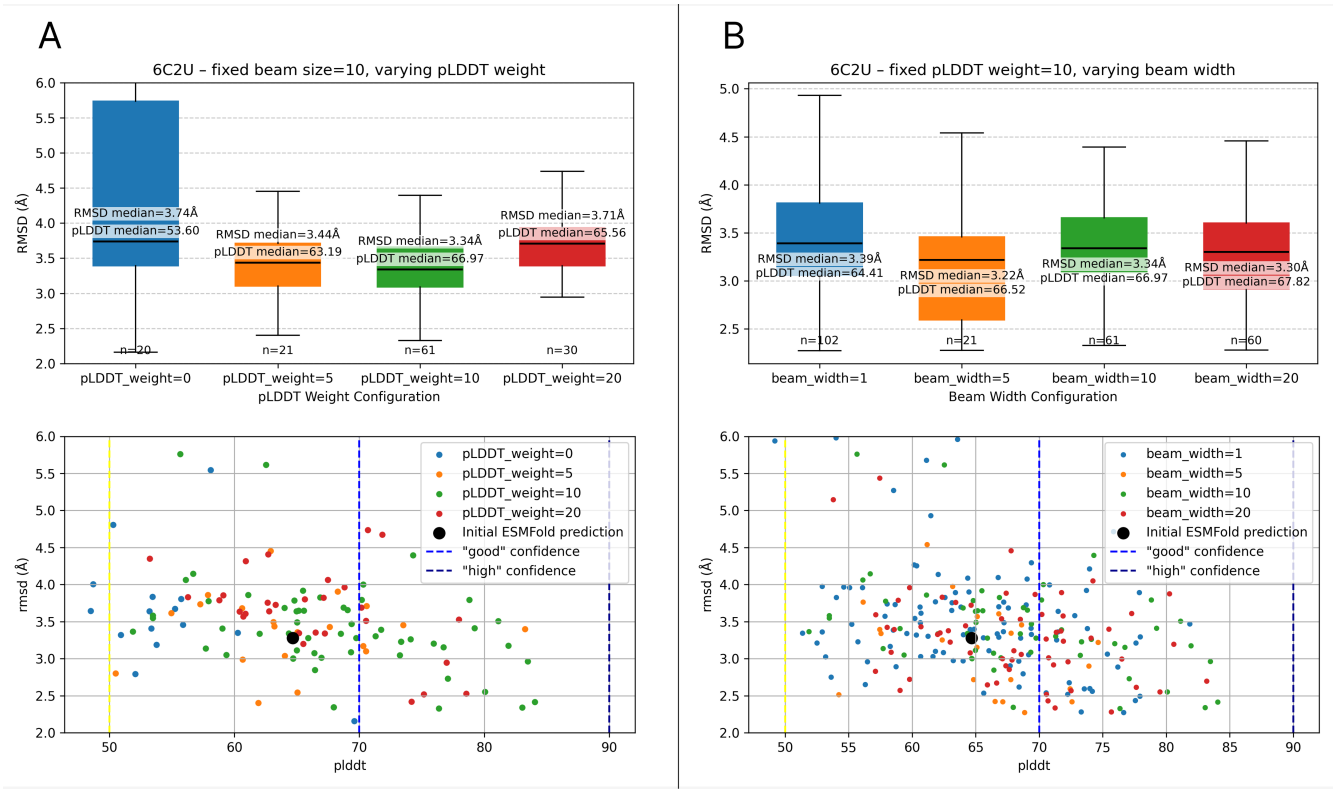

Figure 11: Distribution of translation results of 6C2U, varying beam width and pLDDT weight.

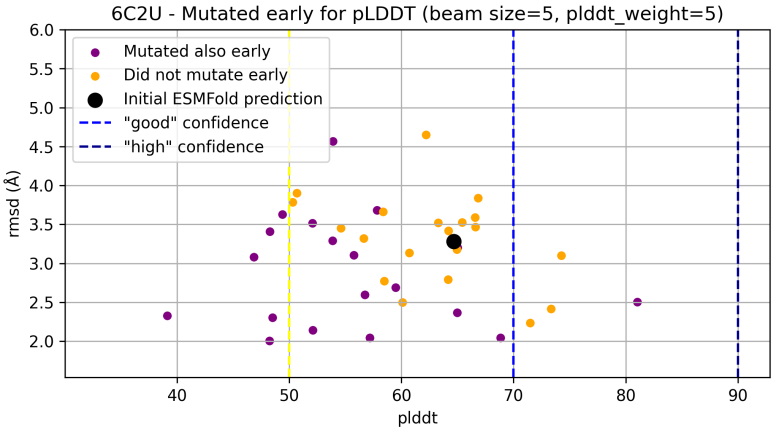

Figure 12: RMSD and pLDDT when mutating also the early residues

4.1 Presentation at ISMB/ECCB 2025

This work was published and presented as a poster at the joint conference Intelligent Systems for Molecular Biology, with European Conference on Computational Biology (ISMB/ECCB), which took place July 20-24 2025 in Liverpool, UK. The website for this presentation can be found at `https://hugo.matfyz.cz/eccb2025`, and contains the poster PDF, animations of the translation process, and a link to the GitHub repo.

4.2 Provided Code

Repository for this project is at `https://github.com/hugohrban/nprg070`. There are two main directories `experimental_data_a` which relates to section 2 and `protein-translator` related to section 3. We do not include the code for running various tools like MMseqs2, Foldseek, etc, code for plotting figures and analyzing output data. Description of files in `experimental_data_analysis`:

- `predicting_expression_and_foldability` - sections 2.1, 2.6
  - `stat_tests.py` - relating to sections 2.1 and 2.6, chi-squared test per position, training simple statistical models
  - `models.py` - definitions of deep learning models
  - `train.py` - main training loop
  - `utils.py` - utility functions

The code related to section 3 is in a separate repo (`https://github.com/hugohrban/protein-translator`), added as a git submodule. The separate repository was created due to the poster presentation at the conference. Brief description of the files in `protein-translator/src`:

- `constants.py` - definition of important constants, like amino acid names
- `download_pdbs.py` - script for downloading FASTA and CIF files from the PDB server
- `get_plddt.py` - script for getting pLDDT of a structure
- `kabsch.py` - calculating RMSD, superposition using Kabsch algorithm
- `translate.py` - main function for running the translation, run with `--help` to get the full explanation
- `utils.py` - utility functions, but can also be ran directly

The repo also contains the poster PDF and a README.md with usage explanation.

References

[1] B. Alberts, D. Bray, K. Hopkin, A.D. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential Cell Biology*. CRC Press, 2015.

[2] Alex Bateman. *Sequence Classification of Protein Families: Pfam and other Resources*, chapter 2, pages 25–36. John Wiley & Sons, Ltd, 2013.

[3] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.

[4] C.I. Brändén and J. Tooze. *Introduction to Protein Structure*. Garland Pub., 1999.

[5] Hugo Hrbáň. *Generování proteinových sekvencí s danou charakteristikou.* Bakalářská práce, Univerzita Karlova, Matematicko-fyzikální fakulta, Katedra softwarového inženýrství, Praha, 2024.

[6] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[7] Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, Nov 2017.

[8] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[9] Vyacheslav Tretyachenko, Jiří Vymětal, Tereza Neuwirthová, Jiří Vondrášek, Kosuke Fujishima, and Klára Hlouchová. Modern and prebiotic amino acids support distinct structural profiles in proteins. *Open Biology*, 12(6):220040, 2022.

[10] Mihaly Varadi, Damian Bertoni, Paulyna Magana, Urmila Paramval, Ivanna Pidruchna, Malarvizhi Radhakrishnan, Maxim Tsenkov, Sreenath Nair, Milot Mirdita, Jingi Yeo, Oleg Kovalevskiy, Kathryn Tunyasuvunakool, Agata Laydon, Augustin Žídek, Hamish Tomlinson, Dhavanthi Hariharan, Josh Abrahamson, Tim Green, John Jumper, Ewan Birney, Martin Steinegger, Demis Hassabis, and Sameer Velankar. Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic Acids Research*, 52(D1):D368–D375, 11 2023.