Bioinformatics Program

Technical University of Munich

Ludwig-Maximilians-Universität München

Master's Thesis in Bioinformatics

The best of both worlds: Marrying evolutionary information with Protein Language models

Kyra Erckert

Bioinformatics Program

Technical University of Munich

Ludwig-Maximilians-Universität München

Master's Thesis in Bioinformatics

The best of both worlds: Marrying evolutionary information with Protein Language models

Das Beste aus beiden Welten: Verheiraten von evolutionärer Information mit Protein-Sprachmodellen

Author: Kyra Erckert
Supervisor: Prof. Dr. Burkhard Rost
Advisor: Michael Heinzinger
I12 - Department for Bioinformatics and Computational Biology
Fakultät für Informatik
Boltzmannstraße 3, 85748 Garching
Submitted: 15.04.2021

Master's thesis statement of originality

I confirm that this master's thesis is my own work and I have documented all sources and material used.

....

Place and date Signature

Abstract

Motivation: This project aims to explore the effect of training set variations for embedding based three state secondary structure prediction. The embeddings we used are vector representations derived from the T5 language model, that has been trained on protein sequence data. For this purpose, different preprocessing steps were used, to generate single sequence, multiple sequence alignment and weighted multiple sequence alignment embeddings and the resulting prediction performances were compared. Additionally, we explored what effect the use of different secondary structure source databases (PDB and PDBredo) have and if a change in performance can be observed between 3Å and 3.5Å as resolution thresholds for the training set. Lastly, we tested if model performance can be improved by training on sequence clusters instead of single sequences to benefit from sequence similar proteins with slightly different secondary structure.

Results: In context of this work, we generated a new dataset for secondary structure prediction. This dataset was generated with stricter redundancy reduction criteria than the commonly used sequence or profile similarity. For our redundancy reduction, structural similarity as defined by the CATH-hierarchy was used for creating dataset splits. In addition to secondary structure information, our dataset also contains additional data like disorder, phi and psi angles or CATH annotations.

We could observe a performance difference for the different parameters that have been investigated in this work. However, generalizable trends could only be identified for some of the aspects we analyzed. Training on PDB sequences seems to generally be neutral or beneficial in comparison to PDBredo. We also observed that single sequence embeddings reach higher performances than multiple sequence alignment embeddings. No clear trends could be observed for the effect of cluster training.

Zusammenfassung

Motivation: Ziel dieses Projektes war es den Effekt von Variationen des Trainingssets für Embedding basierte Drei-Kategorien-Sekundärstrukturvorhersage zu analysieren. Die Embeddings die wir benutzen sind Vektordarstellungen, die mit dem T5 Sprachmodell, das auf Proteinsequenzdaten trainiert wurde, erstellt wurden. Zu diesem Zweck wurden verschiedene Vorverarbeitungsschritte benutzt um Einzelsequenz, multiple Sequenzalignment und gewichtete multiple Sequenzalignment Embeddings zu generieren und die resultierenden Vorhersageleistungen verglichen. Zusätzlich wurde der Effekt von verschiedenen Sekundärstrukturdatenbanken (PDB und PDBredo) getestet und verglichen, ob eine Änderung des Auflösungslimits für das Trainingsset von 3Å auf 3.5Å eine Veränderung der Vorhersageleistung bewirkt. Zuletzt testeten wir ob die Vorhersageleistung des Models verbessert werden kann, indem wir mit Sequenzclustern statt Einzelsequenzen trainieren, um von Proteinen mit ähnlicher Sequenz aber leicht unterschiedlicher Sekundärstruktur zu profitieren.
Ergebnisse: Im Rahmen dieser Arbeit wurde ein neues Datenset für Sekundärstrukturvorhersage generiert. Dieses Datenset unterliegt einer strikteren Redundanzreduzierung als die üblicherweise genutzte Sequenz- oder Profilreduzierung. Für unsere Redundanzreduzierung wurde strukturelle Ähnlichkeit auf Basis der CATH-Hierarchie zur Datensatzaufteilung benutzt. Neben Sekundärstrukturinformationen sind auch zusätzliche Daten wie Disorder, Phi- und Psi-Winkel oder CATH Annotationen enthalten.
Wir konnten beobachten, dass die verschiedenen Parameter, die in dieser Arbeit analysiert wurden, zu Unterschieden in der Vorhersageleistung führen. Allerdings konnten generalisierbare Trends nur für mache der analysierten Aspekte identifiziert werden. Training mit PDB Sequenzen scheint im Algemeinen neutrale oder positive Auswirkungen im Vergleich zu PDBredo zu habe. Außerdem war die erhöhte Vorhersageleistung für Einzelsequenzembeddings gegenüber multiplen Sequenzalignmentembeddings beobachtbar. Generalisierbare Trends für Cluster Training konnten nicht identifiziert werden.

Contents

Contents **5**

I Thesis 8

1 Introduction **9**
1.1 Similarity between Protein Sequences . . . 9
1.2 Comparison of Secondary Structure Prediction Methods . . . 10
1.3 Secondary Structure Prediction Methods . . . 11
1.4 Protein Language Models . . . 11
1.5 Databases for Secondary Structure Information . . . 11
1.5.1 PDB . . . 11
1.5.2 PDBredo . . . 12
1.5.3 CATH . . . 12
1.6 Aim of this work . . . 12

2 Methods **14**
2.1 Data Collection . . . 14
2.1.1 PDB . . . 14
2.1.2 PDBredo . . . 15
2.1.3 CATH . . . 16
2.1.4 PISCES Matrix . . . 16
2.1.5 Pairwise and Multiple Sequence Alignments . . . 16
2.2 Data Preprocessing . . . 16
2.2.1 Filtering . . . 17
2.2.2 PID Computation from MMseqs2 Results . . . 18
2.2.3 Hval Computation from MMseqs2 Results . . . 18
2.3 Data Analysis . . . 18
2.3.1 Sequence Differences between PDB and PDBredo . . . 18
2.3.2 Experimental Method . . . 18
2.3.3 CATH Class Sizes . . . 19
2.3.4 Amino Acid Distribution . . . 19
2.3.5 Secondary Structure Distribution . . . 21
2.3.6 Residue and Sequence Number Distribution in MSAs . . . 21
2.4 Dataset Creation . . . 24
2.4.1 Test Set . . . 24
2.4.2 Validation Set . . . 25

2.4.3 Training Sets . . . 26
2.4.4 Dataset Distributions . . . 26
2.5 Model Development . . . 29
2.5.1 Embeddings . . . 29
2.5.2 Model Architecture . . . 30
2.5.3 Model Training . . . 30
2.6 Secondary Structure Prediction Evaluation . . . 31
2.6.1 Q_3 . . . 31
2.6.2 Matthews Correlation Coefficient . . . 32
2.6.3 Fractional Overlap of Segments (SOV) . . . 32
2.6.4 Confusion Matrix . . . 33

3 Results **34**
3.1 Effect of Sampling different Cluster Members during Training . . . 34
3.2 PDBredo vs. PDB . . . 36
3.3 Effect of Method Resolution for Training Set Generation . . . 37
3.4 Single Sequence Embeddings vs. averaged MSA Embeddings . . . 37
3.5 Effect of weighting by Sequence Similarity in averaged MSA Embeddings . 38

4 Discussion **40**
4.1 Conclusions . . . 40
4.1.1 Cluster Training . . . 40
4.1.2 Database Selection . . . 41
4.1.3 Method Resolution . . . 41
4.1.4 Embedding Variations . . . 41
4.2 Future Research . . . 42

II Appendix **43**

5 Glossary **44**

6 Additional Tables **45**

7 Additional Figures **51**
7.1 Venn Diagram of IDs . . . 51
7.2 Method Distribution PDB . . . 52
7.3 Frequency Histograms . . . 52
7.4 MSA Query Sequence Length Distribution . . . 53
7.5 Eight State Structure Distributions . . . 54
7.6 Eight State Segment Length Distributions . . . 55
7.7 Validation Set Performance Bar Plots . . . 60
7.7.1 Single Sequence Embeddings . . . 60
7.7.2 Averaged MSA Embeddings . . . 64
7.7.3 Weighted MSA Embeddings . . . 68
7.7.4 Inverse Weighted MSA Embeddings . . . 72
7.8 Confusion Matrices . . . 76
7.8.1 Single Sequence Embeddings . . . 76

7.8.2 Averaged MSA Embeddings 81
7.8.3 Weighted MSA Embeddings 86
7.8.4 Inverse Weighted MSA Embeddings 91

8 Acknowledgment **96**

Bibliography **97**

Part I

Thesis

1. Introduction

To fully understand the function of a given protein, knowledge about its properties like structure, interaction partners or localization in the cell can be crucial, but experimental data is only available for a small subset of all known protein sequences. For many proteins, annotations can be inferred if a protein with a similar sequence is known. However, this approach is limited by the fact that there are still many sequences, for which we do not know any close homologs. For those sequences, protein feature prediction methods can be especially useful, but have remained a challenging problem to this day.
Early prediction tools were often based on using protein sequences as an input and over time, more complicated methods were developed that tried to use additional evolutionary information, often in the form of multiple sequence alignments (MSAs) or position specific scoring matrices (PSSMs). Most recently, researchers attention has shifted towards modeling a protein sequence as a sentence, with the different amino acids representing words. Self-supervised language models have been trained to automatically embed information from proteins, using sequence alone as an input and it has been shown that those embeddings capture certain signals that can be used for protein classification [5].
We theorized that combining embedding based approaches with evolutionary information from MSAs may further improve language model based predictions. For our study, the secondary structure prediction task was used to asses performance improvements but the results may be valid for other prediction tasks as well.

1.1 Similarity between Protein Sequences

Protein sequences are evolutionarily related and it is therefore important to define a measure of similarity between two of them. The best known approach for obtaining such a measure is through generating a local or global sequence alignment of the candidate sequence, with a positive score increase for matching amino acids [19]. It has been observed that some amino acid changes have a stronger effect on protein structure and function than others. To account for those differences, substitution matrices are often used to score aligned residues. The most commonly used substitution matrices are BLOSUM and PAM.
More sophisticated approaches try to build profiles by using hidden Markov models [2].
Another approach of determining relatedness between protein sequences is to compute the distance to the HSSP curve. The curve represents an empirically determined threshold for automatic family assignment, based on the number of aligned residues and the percentage sequence identity (PID). PID is defined as the number of identical residues in an alignment multiplied by 100, divided by the length of the alignment (L), or more formally:

$$PID = \frac{\#identical\ residues\ in\ alignment * 100}{L} \tag{1.1}$$

The distance to the HSSP curve is computed by the following formula:

$$Hval = PID - \begin{cases} 100 & \text{for L} \leq 11 \\ 480 * L^{-0.32*[1+exp(-L/1000)]} & \text{for } 11 < \text{L} \leq 450 \\ 19.5 & \text{for L} > 450 \end{cases} \tag{1.2}$$

An Hval of above 0 means that the two sequences are expected to have similar structures, with closer relatedness being implied for higher values. Values below zero give an estimate of the dissimilarity of two sequences [12].
Protein similarity measures can be used to transfer annotations from known proteins, for which experimental data is available, to others, if the pairwise similarity exceeds a certain threshold. For proteins with sequence similarity above 35%, fold can be transferred most of the time. Between 20-35% sequence identity, the number of proteins with different structures increases significantly [12].

1.2 Comparison of Secondary Structure Prediction Methods

New secondary structure prediction methods are often evaluated by measuring the performance on a reference dataset. This reference dataset may differ from the dataset used to evaluate existing methods and it is therefore difficult to compare the new method to previously existing ones. Because of this problem, performance in the Critical Assessment of Structure Prediction (CASP) competition has become the gold standard for model evaluation. Sequences with recently solved, but at the time of the CASP competition, unpublished protein structure are used as prediction targets, ensuring that none of the methods could have encountered them during training. The difficulty of the prediction is determined by analyzing the degree of homology of the new structure to already existing ones. Sequences with high homology to existing ones are generally considered easier prediction targets, while sequences with no detectable related structure are expected to be more difficult.
Up to this date, 14 CASP competitions have been held since 1994, taking place every two years. The results are usually published in a special issue of the PROTEINS journal along with paper contributions of some of the most successful participants [8].
While CASP provides a fair comparison between different prediction methods, it is still important to have other non redundant datasets for method development and evaluation and with the introduction of deep learning approaches to computational biology the number of proteins needed in those datasets increases. The most common approach for generating a redundancy reduced dataset is by using a maximum sequence or profile similarity between structures for sequence selection [15]. Moffat at al. recently used CATH topology as a measure for homology, while allowing a comparably high similarity of 70% between structures, for the generation of their training and validation set [10].

1.3 Secondary Structure Prediction Methods

Many methods have been developed for secondary structure prediction and performance has improved from around 60% in 1978 [4], achieved by simple methods that use stochastic analysis on amino acid k-mers, to over 80% for modern methods [7], which leverage todays computational power for training neural networks on structure predictions.
NetSurfP-2.0, one of the more recent structure prediction tools, achieves a Q_3 of 82,4% on the CASP12 dataset, using a one hot encoding of the amino acid sequence and hidden Markov model profiles. The model consists of multiple CNN and LSTM layers and predicts 6 different types of outputs, including three-state secondary structure predictions and protein disorder [7].
Comparisons between different structure prediction tools can be difficult due to different datasets being used for training, hyperparameter tuning and performance evaluation.

1.4 Protein Language Models

Recently, multiple groups have used protein language models for unsupervised sequence pre-training that captures some biological features like amino acid properties or protein structure information [3, 5, 17]. For this approach, protein sequences are viewed as a language with the different amino acids representing individual words. Language models are trained on large sets of sequences and can later be used to generate sequence embeddings as a representation of the amino acid sequence. Those embeddings are used as input for machine learning architectures, which are trained on protein prediction tasks like secondary structure prediction. Models trained on this type of input tend to require less training time due to lowering the necessary number of epochs for convergence during training [5].
First advancements have been made towards combining embedding based information with evolutionary information [11, 14] but at this point in time most approaches are limited to embedding base input only.

1.5 Databases for Secondary Structure Information

Multiple databases exist for protein secondary structure annotation and additional meta data. In this project, data from the Protein Data Bank (PDB), CATH and PDBredo were used for dataset generation.

1.5.1 PDB

The Protein Data Bank (PDB) is one of the oldest community-driven data collections for biological data. It is centered around protein structure data from X-ray, nuclear magnetic resonance (NMR) and electron microscopy (EM) [1]. Each protein entry contains a 3D view of the biological assembly as well as meta data related to the entry, like the functional classification, deposit date or primary literature. Additionally, users are given the option to download information in different file formats like fasta, mmCIF or PDB format for further processing.

The PDB can be searched either manually through the web page or computationally through the Application Programming Interface (API) endpoints.

1.5.2 PDBredo

PDBredo consists of two main parts: First, a server that implements a fully automated decision making system for model refinement, rebuilding and validation and second, a databank with optimized versions of existing PDB entries. The Server is recommended for use in the late stages of model building for validation or model polishing, before submitting the new protein model to PDB.
The database has been build by applying the servers decision making system to protein structures contained in PDB at the time. For each refined entry, PDBredo includes a description of the model changes that have been made by the server [6].
Structure data from PDBredo may be beneficial for downstream tasks because the refined structures, that have been computed with modern hard- and software, may be more accurate and include less noise. Using those newer structures also ensures a standardized preprocessing, which removes potential biases that may come from distribution shifts in structure quality due to age.

1.5.3 CATH

CATH provides classifications of protein domains based on structural data in a hierarchical system. At the top level, proteins are sorted into classes (C) based on their structural content. The following architecture level (A) sorts the proteins by there structure arrangement in 3D-space. In the topology level (T), proteins are grouped based on connection and arrangement of secondary structure elements and finally, at the homologous superfamily level (H), they are sorted based on good evidence of domains being related [16].

1.6 Aim of this work

This thesis is a continuation of a preliminary work that was done as part of the Masterpraktikum 2020 at the Rostlab (TUM). In the preliminary work, the information from precomputed multiple sequence alignments (MSAs) from ConSurfDB were combined with SeqVec embeddings to train multiple machine learning architectures. The results obtained from combining embeddings with evolutionary information from MSAs were compared to training on single sequence SeqVec embeddings, PSSMs and hot encoded amino acid sequence. The results obtained suggested that methods relying on embeddings can be improved by using additional evolutionary information from MSAs. At the end of the preliminary work, it was not clear if more diverse MSAs could further improve performance. The precomputed alignments that were used only contained highly similar sequences (E-value threshold: 0,0001) and were also restricted to a maximum of 300 sequences. Furthermore, it was not clear what effect the dataset used and its size may have had on the obtained results.
The question of how embeddings and MSAs are combined best, to obtain as much additional information as possible, also remained unanswered. In the preliminary work, two different approaches were tested: The simpler approach embedded each sequence in the MSA and performed individual structure predictions for each of the embedded sequences.

Those structure predictions were than combined into a single structure prediction for the protein in question by a majority vote for each position in the MSA. The second approach also started by embedding each sequence in a MSA. Embeddings were than combined into a single one by averaging over the individual embeddings in the MSA column wise. The resulting averaged embedding was used to do one single structure prediction for the protein in question. The second approach resulted in a performance improvement of 3-5% (depending on the architecture used) in comparison to single sequence embeddings. It was left open for further investigation if the performance could be further improved by weighting parts of the MSAs differently (e.g. by sequence similarity).
The aim of this work is to explore both of these aspects, as well as investigate the effect of additional modifications to the training set like structure resolution quality and training on sequence clusters instead of single sequences. First of all, a new dataset was generated, containing MSAs with less similar sequences than the ones used in the preliminary work. This dataset was redundancy reduced based on CATH Topology, as well as Hval instead of the commonly used profile similarity, to get a more conservative performance estimate. This dataset was then used to reproduce the results from the preliminary work, as well as extending the previous findings.

2. Methods

For dataset creation, amino acid sequences, secondary structure and meta data was collected from PDB, PDBredo and CATH. Additional pairwise data that could be used for filtering was obtained from the Dunbrack lab and pairwise alignments were computed with MMseqs2. The data was used to generate training, test and validation sets for machine learning, with CATH class, PID and Hval as the main criteria for redundancy reduction. For the training set, sequence clusters were generated instead of picking single sequences. For each sequence in the final dataset, a single sequence, unweighted averaged MSA and two types of weighted by sequence similarity MSA embeddings, were generated.
For training, the following parameters were considered:

- Resolution of training set (3Å/3.5Å)
- Embedding type (single sequence/ unweighted/ weighted/ inverse weighted)
- Database used to obtain structure data (PDB/PDBredo)
- Cluster usage during training (yes/no)

For all combinations of parameters, a convolutional neural network was trained with otherwise identical hyperparameters and the results were compared based on Q_3, MCC and SOV.

2.1 Data Collection

Sequence data and meta information have been collected from PDB, PDBredo and CATH. For redundancy reduction, a similarity matrix was obtained from the PISCES lab and sequence alignments were computed with MMseqs2.

2.1.1 PDB

PDBs advanced search was used to obtain general information about all available protein chains below a certain resolution threshold. The original resolution threshold, that was used on the 26th of October 2020, was set to 3Å and the following columns were selected and exported as comma separated files (CSVs):

- Entry ID
- Angle alpha (°)
- Angle beta (°)

- Angle gamma (°)
- EM resolution (Å)
- EM Diffraction Resolution
- Experimental Method
- PDB ID
- Resolution (Å)
- Average B Factor
- R Free
- R Work
- R All
- R Observed
- Sequence
- Chain ID
- Entity ID
- Entry Id (Polymer Entity Identifiers)

After deciding to include protein chains with higher resolutions at a later point in time in the project, the data collection step was repeated at 3.5Å and the following two columns were added to allow filtering by release and deposit date:

- Deposition Date
- Release Date

This resulted in a total of 156 897 proteins and 573 479 chains.
Secondary structures for PDB entries have been obtained from PDB at `https://www.rcsb.org/pdb/static.do?p=download/http/index.html#ss` on 26th of October 2020. This page was part of the legacy API of PDB and was shut down on December 9th 2020. To reuse the existing pipeline from this project for dataset creation, secondary structures will have to be obtained from the new API and some of the existing code may need to be modified to accept a different input format.

2.1.2 PDBredo

A previous database dump of PDBredo was available on the Rostlab server. A crawler was used to collect all available DSSP and pdb files from the directory and any possible subdirectories. The total number of proteins, obtained in this way was 136 267 and a total of 424 587 chains.

2.1.3 CATH

The daily snapshot of CATH from the 21st of October 2020 was obtained from the CATH homepage. CATH annotations were obtained from the "cath-b-newest-all.gz" archive with a simple file parser. 140 383 proteins and 357 296 chains were obtained this way.

2.1.4 PISCES Matrix

The similarity matrix used by the PISCES server was obtained through personal correspondence with Qifang Xu from the Dunbrack lab. The matrix consists of two files: one file containing the mapping of PDB ids to internal identifiers and a second file that contains percent values for the profile identity, probability of two sequences to be homologous and an alignment score (personal correspondence, 11th of November 2020). The corresponding alignments were computed with all-against-all Hidden Markov model (HMM) to HMM alignments by the hhpred tool and if hhpred could not find an alignment, proteins are assumed to have 0% identity (personal correspondence, 9th of November 2020).

2.1.5 Pairwise and Multiple Sequence Alignments

MMseqs2 was used to compute pairwise sequence similarities for all sequences with a resolution of 3,5Å or less that were available in PDB, PDBredo and CATH, as well as for generating MSAs for all sequences that were later selected for the data set.
The following command was used for pairwise alignments:

```
easy-search fastaFile.fasta fastaFile.fasta db_intersect_res tmp
--num-iterations 3 --db-load-mode 2 -a -s 7.5 --min-seq-id 0.2
--format-output query,target,pident,fident,nident,alnlen,mismatch,
gapopen,qstart,qend,tstart,tend,evalue,bits,qcov,tcov,cigar
```

The pairwise MMseqs2 output contained ids of the sequence pair, percentage, fraction and number of identical matches, alignment length, number of mismatches, number of gap open events, start and end position of query and target sequence, e-value, bit score, fraction of query and target sequence that was covered by the alignment and the cigar string. The cigar string contains the alignment as a short string, with each position containing M (match), D (deletion) and I (insertion).
Stockholm format was chosen as the output format for the MSAs. Additionally, position specific scoring matrices (PSSMs) for each MSA were generated.

2.2 Data Preprocessing

All obtained proteins and chains from Section 2.1.1, 2.1.2 and 2.1.3 were filtered to the ones contained in all three databases, which had a sequence length of at least 40 residues and a resolution of 3.5Å or less. For each pair in the remaining set, PID and Hval were computed based on the MMseqs2 results and later used for dataset generation.

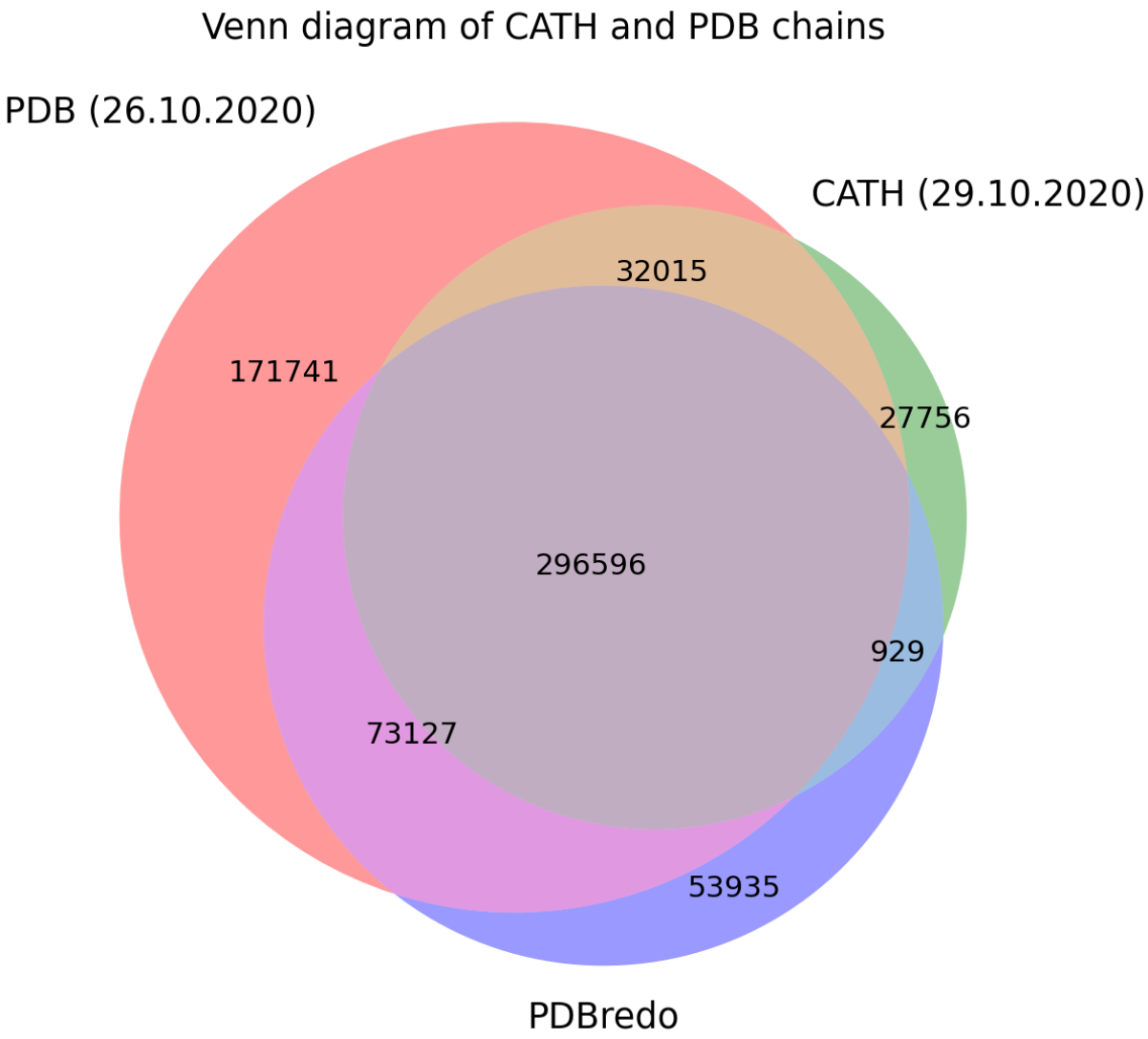

Figure 2.1: Venn diagram of the number of protein chains in PDB, PDBredo and CATH and the size of the individual overlaps with timestamps for data collection

2.2.1 Filtering

Protein chains obtained from CASP, PDB and PDBredo were filtered to the ones available in all 3 databases. The intersection of all three databases contained 296 596 chains (from 117 623 proteins).
Figure 2.1 shows that the largest section, that was filtered out, were chains exclusive to PDB (171 741). Another large section that was removed were chains that are either exclusive to PDBredo (53 935) or chains that are in PDB and PDBredo but can't be found in CATH (73 127). Chains that are exclusive to PDBredo are assumed to be chains from PDB entries that have become deprecated since the structures were recomputed for PDBredo. 27 756 more chains were excluded due to being exclusive to CATH and another 32 944 due to being only in the intersection of CATH and PDB or CATH and PDBredo. Additional sequences were filtered to a minimum number of 40 residues and a resolution of 3.5Å or less.

2.2.2 PID Computation from MMseqs2 Results

The PID value was computed by dividing the number of identical residues by the length of the alignment without gaps and then multiplied by 100 (see Formula 1.1). The length of the alignment without gaps was computed from the MMseqs2 pairwise output by counting all M-states from the cigar string.

2.2.3 Hval Computation from MMseqs2 Results

PID values and the sum of M-states as value for L were used to compute the pairwise Hvals, by applying Formula 1.1.

Example:
cigar: 150M8D8M2I6M13D14M1D71M
n_ident: 93
L: sum of match states in cigar string = 249
PID: $\frac{n\_ident*100}{L} = \frac{93*100}{249} \sim 37.3$
Hval ~ 16.6

2.3 Data Analysis

Before dataset generation, the remaining chains and proteins were checked for any unintended or unexpected biases and trends in the available data. We also evaluated if there was any contradicting data between the different databases or any features that may be relevant for training a network on the underlying data

2.3.1 Sequence Differences between PDB and PDBredo

Some differences have been observed between the PDB and PDBredo, resulting in slightly different amino acid sequences. DSSP files from PDBredo often do not contain the first and last few residues of a protein. Furthermore, position and sequence segments may be missing because of detected backbone discontinuity (recorded as ! and * in the DSSP files). Besides the already mentioned changes, sequence residues may deviate slightly from each other.
To account for those changes, we decided to train our networks either on PDB sequences and the annotations obtained from there, or on the PDBredo counterpart, creating embeddings for sequences obtained from both databases.

2.3.2 Experimental Method

For all proteins in the intersection of CATH, PDBredo and PDB at a resolution of $\leq$3Å, counts for resolution methods were collected (see Figure 2.2). The majority of structures at this resolution level have been determined by X-ray diffraction (150 613), followed by electron microscopy (6 012). All other methods have been used less then 180 times in this subset, with electron crystallography (167) and neutron diffraction (175) being the most common ones. Additional data can be found in the Appendix in Figure 7.2.

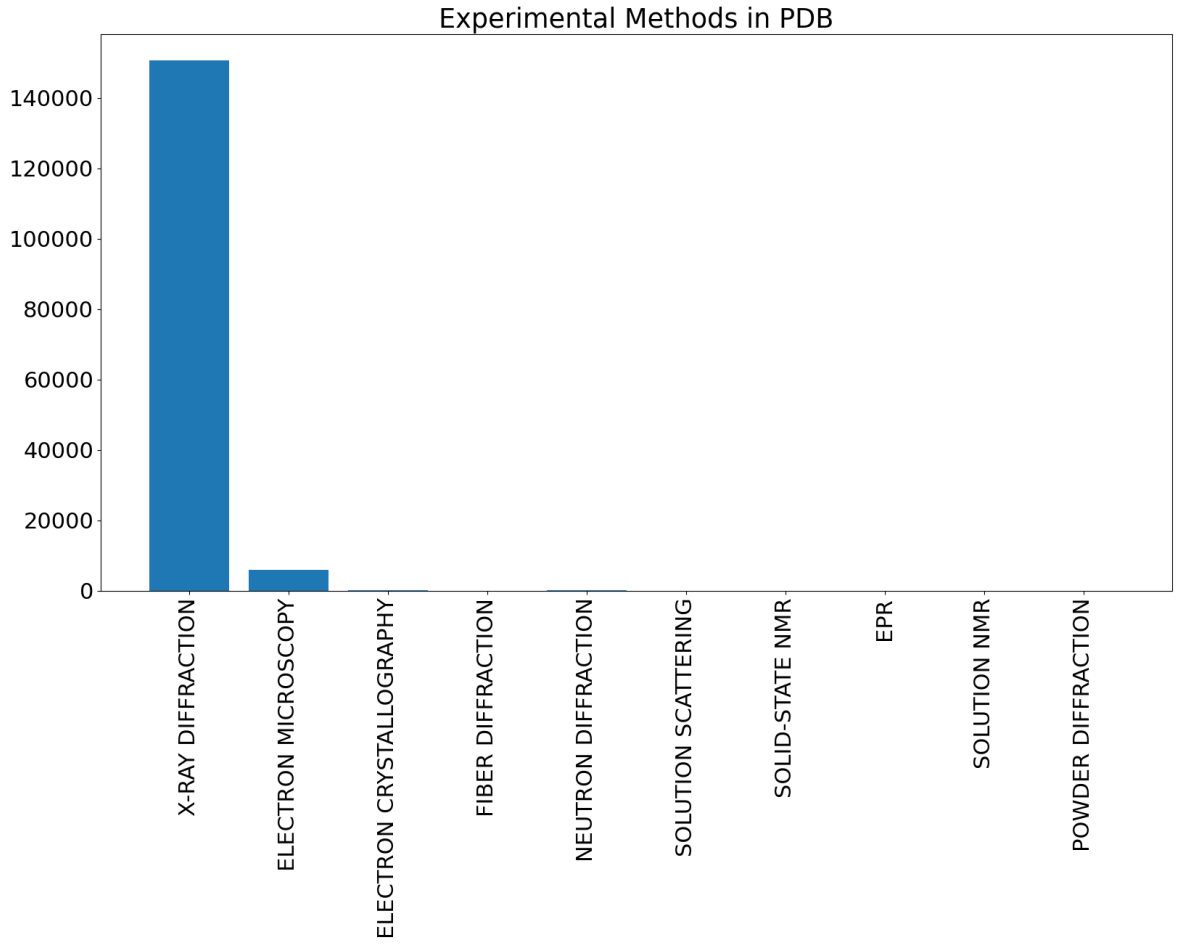

Figure 2.2: Experimental structure resolution methods for the intersection of PDB, PDBredo and CATH at a resolution of 3Å

2.3.3 CATH Class Sizes

Figure 2.3 shows, that the majority of proteins with CATH annotations belongs to the alpha beta class (3), making up over half of all structures in CATH. The next largest classes are mainly alpha (1) and mainly beta (2), with the latter one being slightly larger. All remaining proteins fall either in the small class of few secondary structures (4) or only had preliminary domain assignments (6) at the time of data collection.

2.3.4 Amino Acid Distribution

Figure 2.4 shows, that most amino acids have very similar frequencies, independent of the CATH class. However, some amino acids have an increased/decreased frequency in specific CATH classes.
The polar Serine (S) and Threonine (T) as well as the hydrophobic Valine (V) are more common in mainly beta proteins than any other CATH class but frequencies of hydrophobic Alanine (A) and Methionine (M), negatively charged Glutamic Acid (E) and positively charged Arginine(R) are decreased.
Mainly alpha proteins show higher frequencies for Glutamic Acid (E) and hydrophobic Leucine (L), Isoleucine (I) and Methionine (M) and a lower frequencies for Glycine (G).

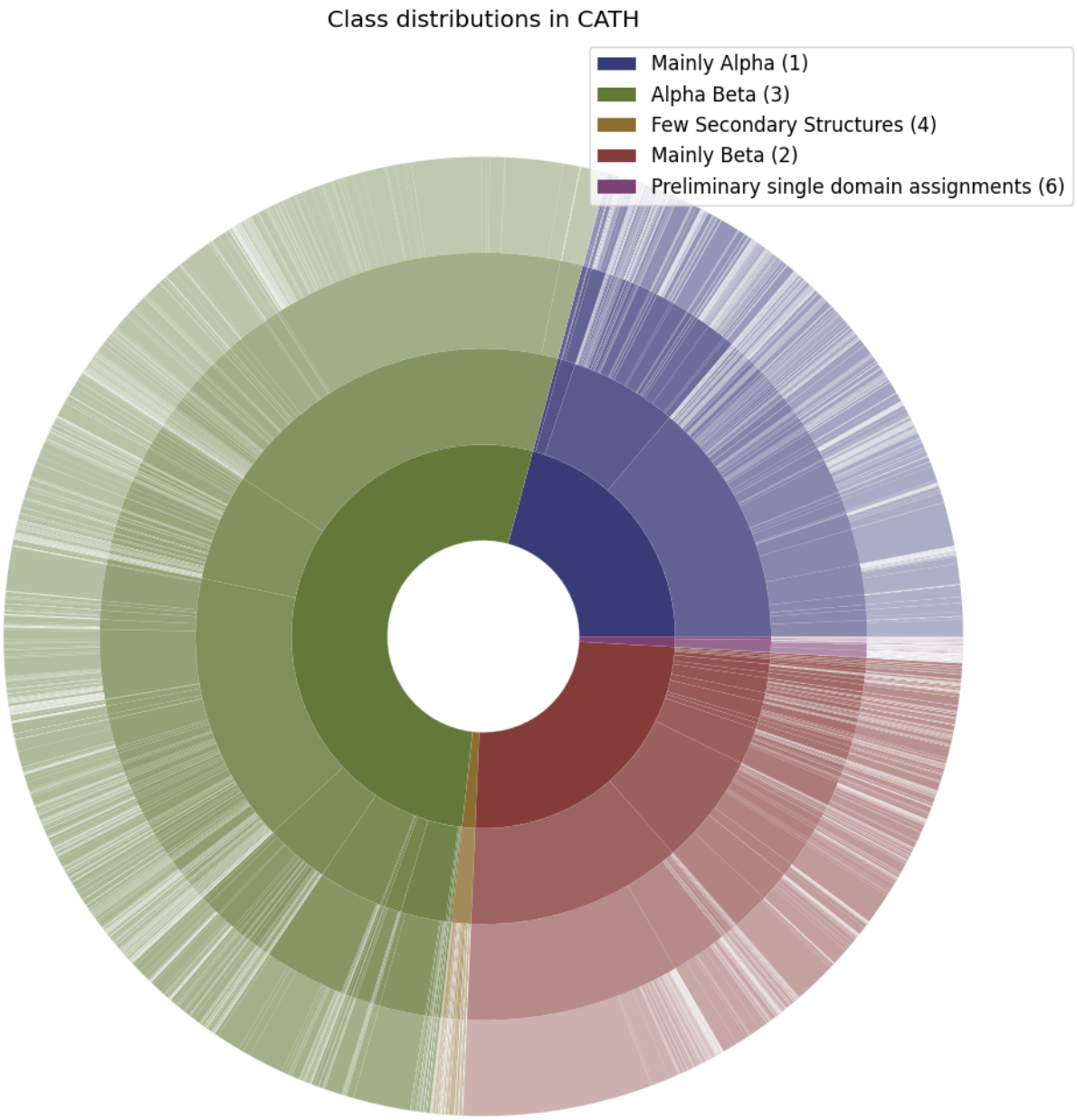

Figure 2.3: Distribution of CATH classes for all proteins obtained from CATH. The inner most circle represents annotations on the class (C) level, second layer for architecture (A), third layer for topology (T) and the outermost layer represents homologous super family (H) annotations.

It can also be observed that certain amino acids like Methionine (M), Cysteine (C) and Tryptophan (W) appear much less frequently than to be expected by random chance and Leucine (L), Glycine (G), Glutamic Acid (E) , Alanine (A) and Serine (S) are more common. The more common amino acids may end up being easier prediction targets, especially if they are very common for a specific secondary structure element, while the less common residues will probably be harder to learn due to the limited number of samples.

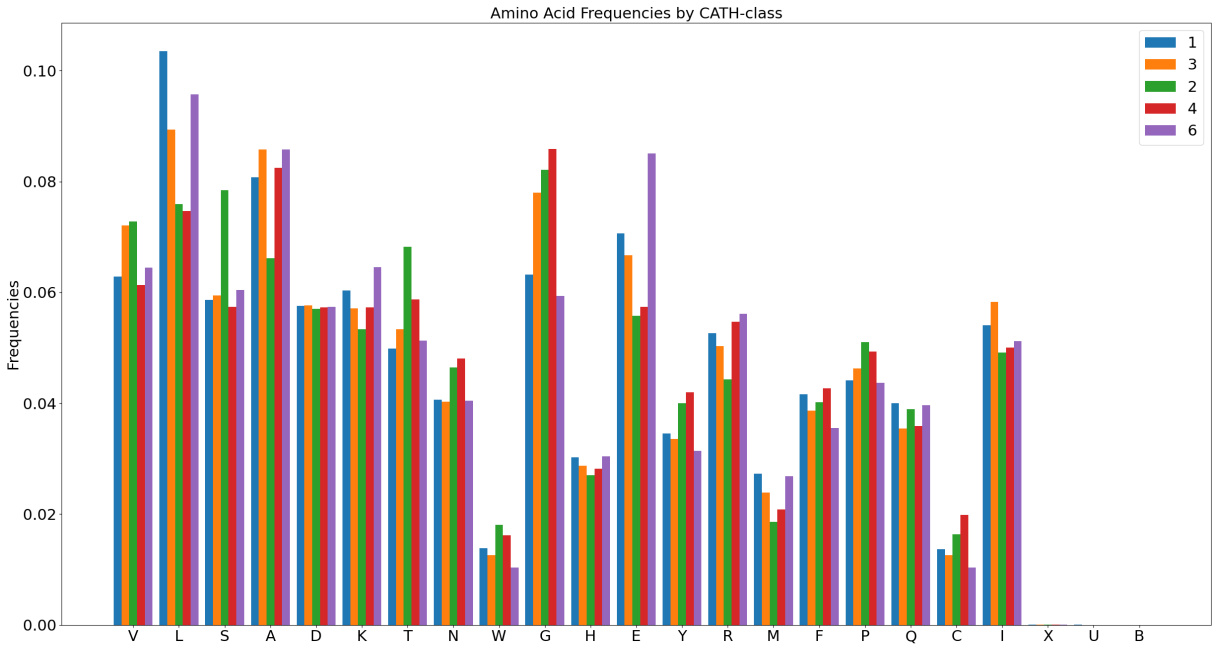

Figure 2.4: Amino acid distributions by CATH class were computed by counting occurrence of each one letter code in all proteins that belong to the same CATH class. The following CATH classes were considered: mainly alpha (1), mainly beta (2), alpha beta (3), few secondary structures(4), preliminary single domain assignments (6)

2.3.5 Secondary Structure Distribution

Secondary structure distributions were determined by randomly selecting 1000 sequences for different Armstrong thresholds (2Å, 3Å and 3.5Å) from the intersection of PDB, PDBredo and CATH. For those sequences, the number of helix, strand and other residues was counted. This step was repeated 1000 times for each resolution threshold, resulting in a distribution for each 3-state secondary structure type at each resolution threshold.

Figure 2.5 shows the distribution for helix residues. Distributions for this secondary structure deviate by approximately 0.5% from the rest in the 2Å subset, but 3Å and 3,5Å are almost identical.
Helix frequency in the intersection set lies between 33% and 36%.
Frequency for strands is around 23% and all other structures make up around 43%. Additional figures for strand and other can be found in the Appendix in Figure 7.4 and 7.3.

2.3.6 Residue and Sequence Number Distribution in MSAs

The MSA alignments, generated by MMseqs2, can contain up to ∼4 000 aligned sequences with the mean number of sequences being ∼934.4. The most common sizes are around 450 and around 600 aligned sequences. For each of those alignment sizes, there are ∼800 proteins in the dataset with an MSA of those sizes. For each bucket above 1 000 sequences in an MSA, the total number of proteins with such a large alignment is below 200, decreasing the larger the alignment gets (see Figure 2.6).

The mean number of residues per MSA is ∼178 668.9. Over 2 000 MSAs have between ∼40 000 and 80 000 residues, with all residue counts above that value appearing less than 1 000 times and the counts quickly decreasing the higher the residue counts get (see

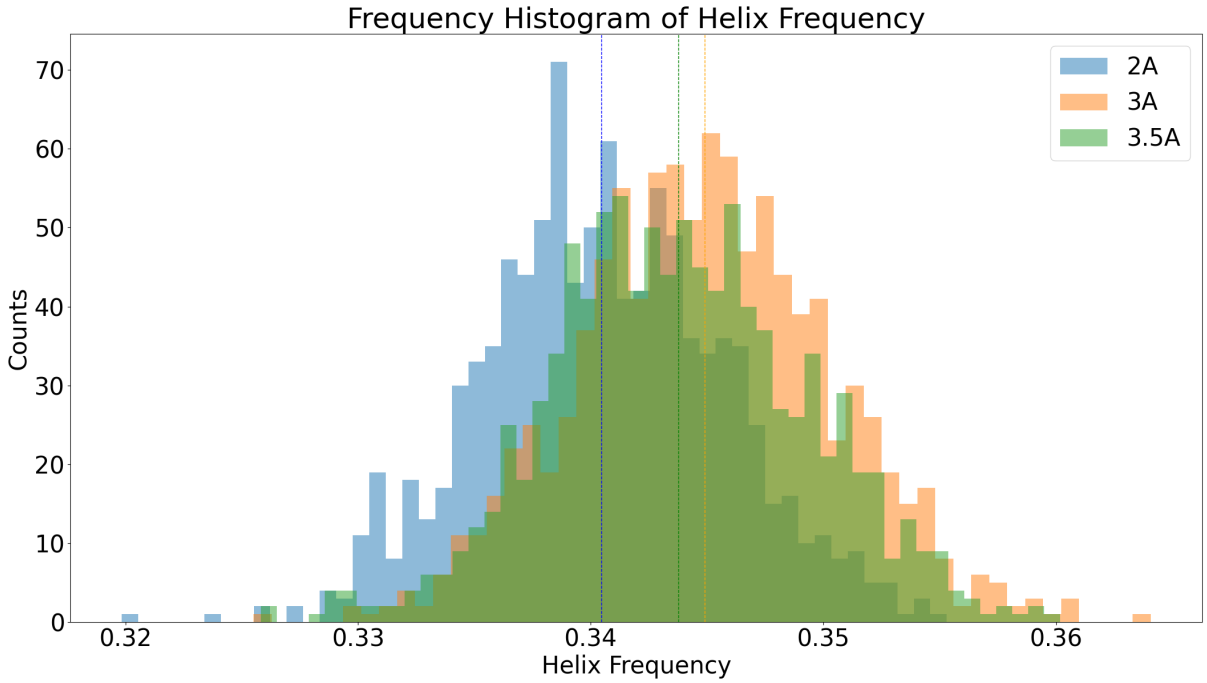

Figure 2.5: Distribution of helix residues after bootstrapping from the chains that were in PDB, PDBredo and CATH. Bootstrapping was done by selecting 1000 random sequences 1000 times and counting the number of helix residues for each iteration. Vertical lines indicate the mean for each resolution.

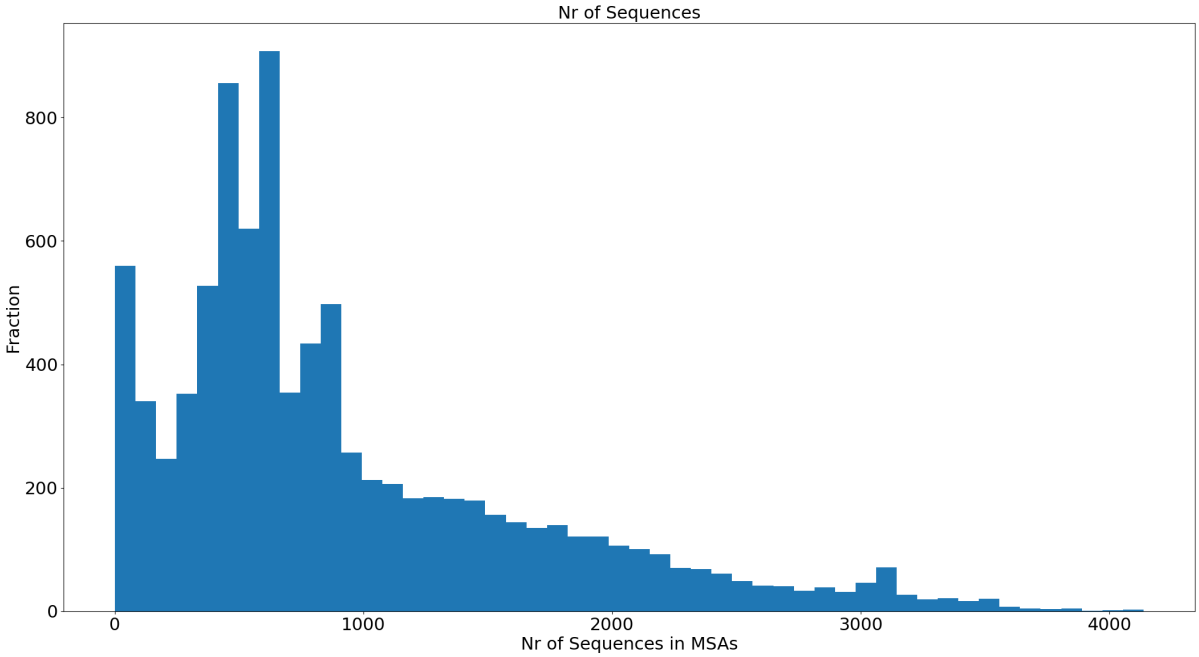

Figure 2.6: Distribution of the number of aligned sequences in the MMseqs2 MSAs. The majority of MSAs have up to 2 000 aligned sequences.

Figure 2.7).
Query sequences are between 40 and ∼800 residues long, with a few exceptions reaching a query sequence length of up to ∼1 600 amino acids (Supplementary Figure 7.5). The mean length of the query sequences is ∼231.7 and there are no query sequences with less

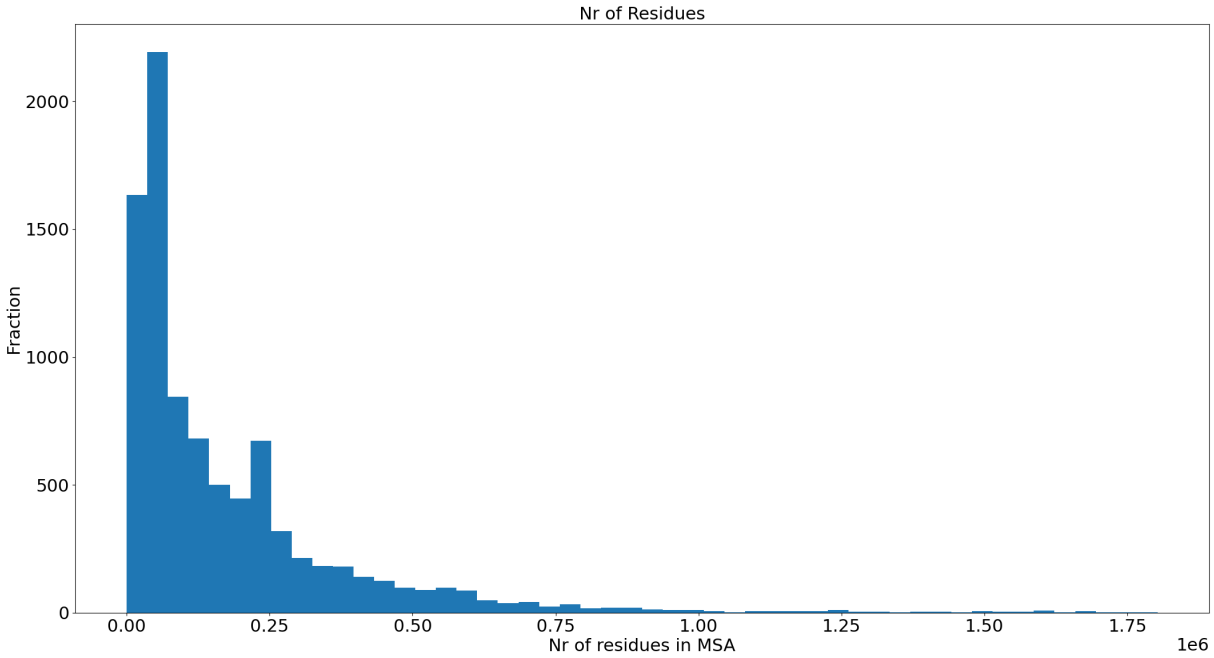

Figure 2.7: Distribution of number of residues for MMseqs2 MSAs. Most MSAs have below 1e6 residues but there are very few MSAs with up to almost 2e6 residues.

than 40 residues.

2.4 Dataset Creation

Chains for the training, test and validation set were sampled from the ∼297 000 chains that are in the intersection of PDB, PDBredo and CATH.
As a first idea for dataset creation, those chains were filtered to any with a resolution of 2Å or less (∼103 000) and then reduced to the ones with a pairwise profile similarity of less than 20%, based on the similarity matrix that was obtained from the Dunbrack lab. This resulted in a total of ∼6 000 protein chains, which was considered too small for creating large enough training, test and validation sets for machine learning. For detailed information refer to Table 6.1 in the Appendix.
As an alternative, the probability score, that is contained in the similarity matrix, was considered as a cutoff. The probability score is an estimate of how likely it is that two sequences are at least partially homologous and is based on the secondary structure score and the alignment score of the pairwise alignment, that was used for the generation of the similarity matrix. If only the probability score is considered at a cutoff of 50%, the number of selected chains after filtering drops to 2 829 (see supplementary Table 6.2).
As another alternative selection mechanism that was explored, a clustering approach was tested. All chains with a similarity above 20% and a probability above 50% to the cluster representative were considered to belong to the same cluster. Additionally, the resolution cutoff was set to 2.5Å. As the cluster representative, the longest sequence in the cluster was selected. Roughly ∼9 000 clusters were generated with this approach (see supplementary Table 6.4). From those clusters, 100 were randomly selected as a test set and another 100 for the validation set. To ensure a very strict redundancy reduction, only clusters with representatives that did not have a CATH topology (T), that is either present in test or validation, were considered for the training set. This restriction resulted in a maximum of 3 451 clusters, which could contribute to the training set. If the homology level (H) was used instead, 6 219 clusters remained for the training set. To ensure that no unintentional bias was introduced by the sampling and clustering process, the pairwise similarity and probability scores between all representatives, for any cluster, that was selected either for training, test or validation, were plotted, if they were available (see Figure 2.8). We expected a low profile similarity to be correlated with a low percentage of the two sequences to be homologous. Contrary to our expectation, we observed, that a pair with a low profile similarity very often still has a high chance for the two sequences to be homologous, according to the probability score obtained from the PISCES matrix. Based on this finding, we had to reconsider using those two scores in combination for clustering sequences.
Based on our results, we decided to come up with a different approach for dataset chain selection that was not based on the similarity matrix from the Dunbrack lab. For the new approach, we decided on using the computed distance to the HSSP curve (Hval) as well as the CATH topology as the main criteria for redundancy reduction.

2.4.1 Test Set

From the chains that remained after filtering according to section 2.2.1, 100 were selected for the test set. For those chains, only the ones deposited after the NetSurfP-2.0 release in April 2018 were considered if, and only if, they had a resolution of 2Å or less. Furthermore any pair (a,b) with a,b ∈ Test had to have an Hval≤0.

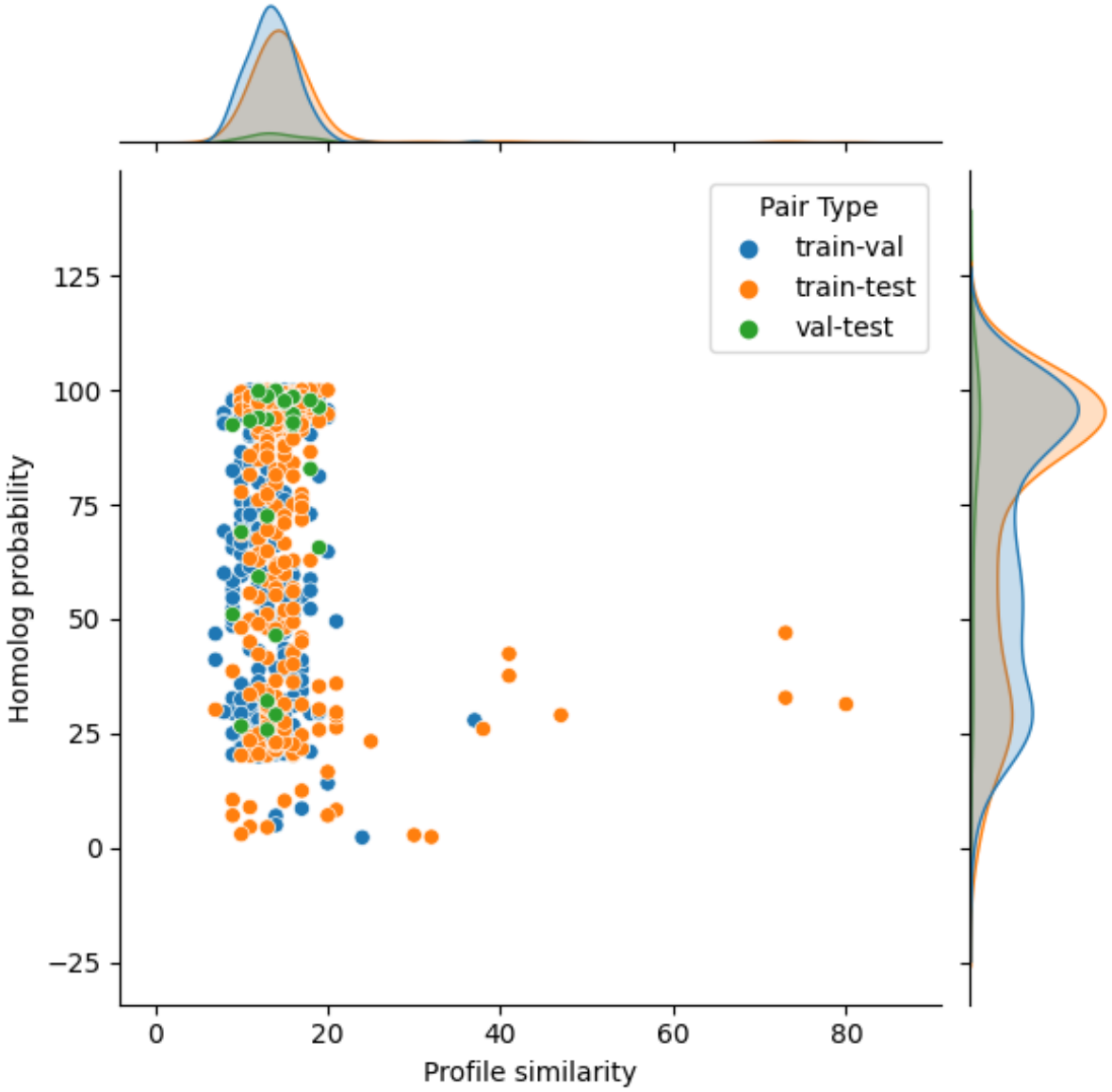

Figure 2.8: Scatterplot showing the similarity and probability value for cluster representative pairs with density curves.
Representatives are picked with the following approach: Chains are filtered down to the ones that appear in PDBredo, PDB and CATH, are at least 20 residues long and have a resolution of $\leq$ 2.5A. The remaining chains are sorted by length and then either added to an already existing cluster if they have an identity of >20 and a probability of >50 to the representative of an existing cluster or become the representative of a new cluster. The resulting representatives are filtered again for the ones with a resolution of $\leq$2A, that have been resolved by X-ray diffraction. From the remaining chains, 100 each are sampled without replacement for the test and validation set. The list of all representatives (resolution $\leq$2.5A) is then filtered by CATH class, removing all chains that have the same CATH class as any of the ones selected for test or validation (removing all previously picked chains from test or validation in the process). All remaining chains are used as the training set. Only chain pairs with a corresponding PISCES matrix entry are shown.

2.4.2 Validation Set

For the validation set 100 additional chains were selected after the test set generation. Those chains were chosen from the subset of chains that were deposited before

the NetSurfP-2.0 release and had a resolution of 2Å or less. Additional, any pair (a,b) with a∈Test or a∈Validation and b∈Validation had to have an Hval≤0.

2.4.3 Training Sets

For training, two sets with different resolution cutoffs were created. The high resolution set was restricted to chains deposited before 30th of April 2018 with a maximum resolution of 3Å. Any chain that was added to the training also needed to fulfill the following criteria:

- The CATH annotation on the topology level (T) had to be different from any in the test and validation set
- Hval≤0 for any pair (a,b) with a∈Test or a∈Validation and b∈Training
- PID≤70 for any pair (x,y) with x,y∈Training if x≠y

Those restrictions resulted in a set of 8 287 chains.
The second training set allowed chains with a resolution of up to 3.5Å. All other restrictions applied for selection were identical to the ones for the high resolution set. The selected set consists of 8 490 chains.

2.4.4 Dataset Distributions

To ensure that no biases were introduced during dataset creation, structure and segment length distributions for the training, test and validation sets were computed and compared with each other.

Validation and test set have a 4-5% higher occurrence of helix residues compared to

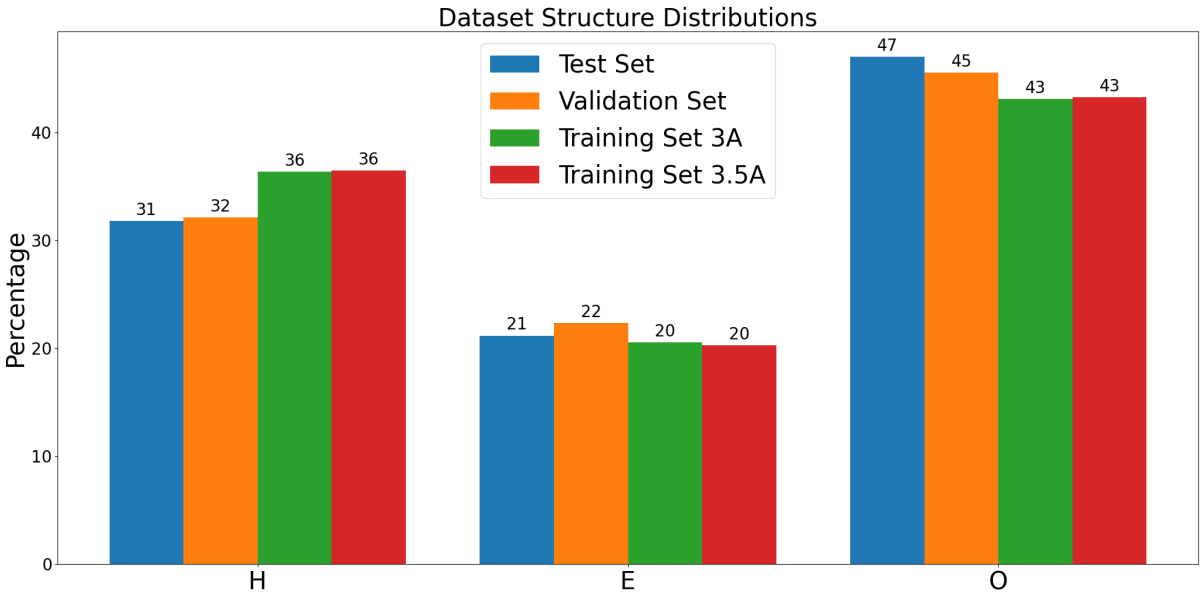

Figure 2.9: 3-state secondary structure distribution for training, test and validation sets

the two training sets, but also a 2-4% lower fraction of other residues. The percentage of strand residues is most similar between the two training sets, but test and validation also don't deviate as much for this residue type as for O and H (see Figure 2.9).

The training sets are most similar to the other distribution of the underlying distribution, obtained for the intersection of PDB, PDBredo and CATH (refer to Figure 7.4). Helix and Strand distribution of the underlying intersection show more similarity to the distributions of the validation and test set (refer to Figures 2.5 and 7.3).

Figure 2.10 shows the segment length distribution for continuous helix segments. Most

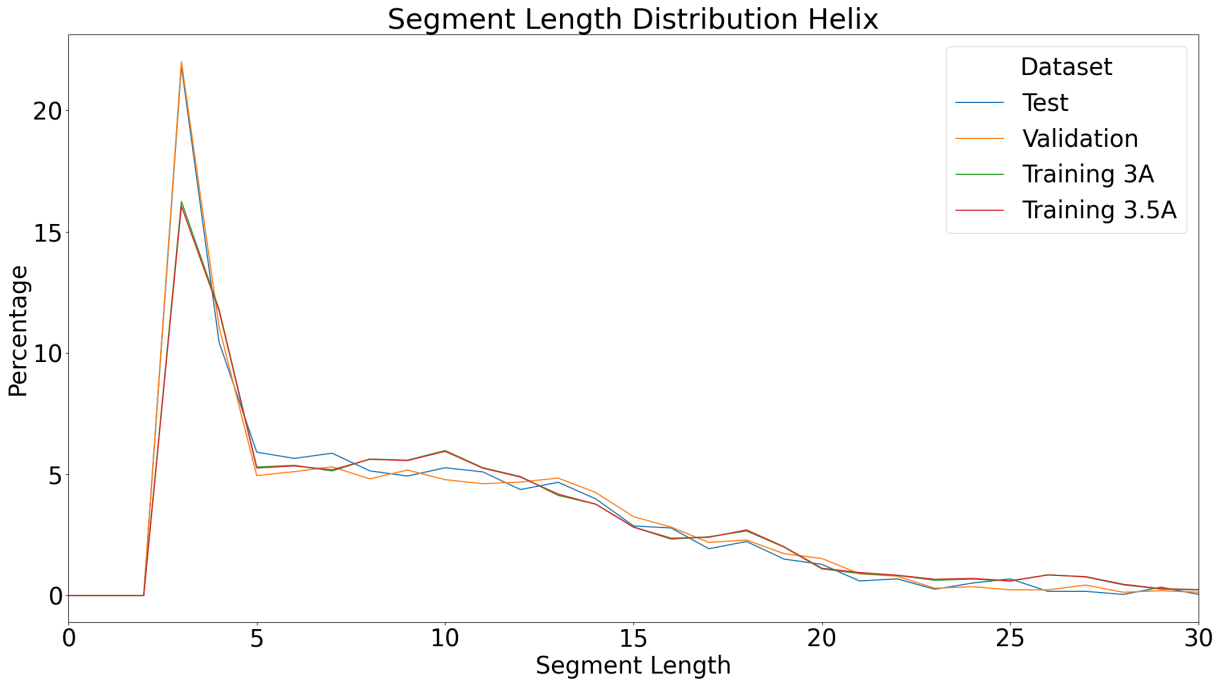

Figure 2.10: Segment length distribution for helix residues in training, test and validation datasets

helix segments contain three continuous helix residues. For the test and validation set, over 20% of all helix segments are of this length and for the training sets this length represents a bit more than 15% of all helix segments. For all other helix segment lengths, distributions between training, test and validation show more similarity. Overall, the training sets have almost identical curves, while the deviation between test and validation, test and training and validation and training are higher.

For strand residues, a higher percentage of single residue segments can be observed in the test set but the difference is less than the observed difference for short helix segments (see Figure 2.11). Another notable difference is that the test set is the only one with a higher percentage, while both training sets as well as the validation set have a very similar percentage of those short segments. Again, both training sets have almost identical curves and test and validation deviate a bit more. At a segment length of 15 continuous residues, all curves converge close to 0%, indicating that longer continuous strand sections are very rare.

Figure 2.12 shows patterns for segments of the other type that are similar to the observations for helix and strand. Both training sets follow almost identical curves, with a lower percentage of segments of length two in the training dataset than observed for validation and test. Again, this difference is a lot smaller than the difference observed for short helix segments. All datasets have a very low percentage of almost 0% for segments longer than 24 consecutive residues of type other.

Additional data for distributions on 8-state secondary structure can be found in the Ap-

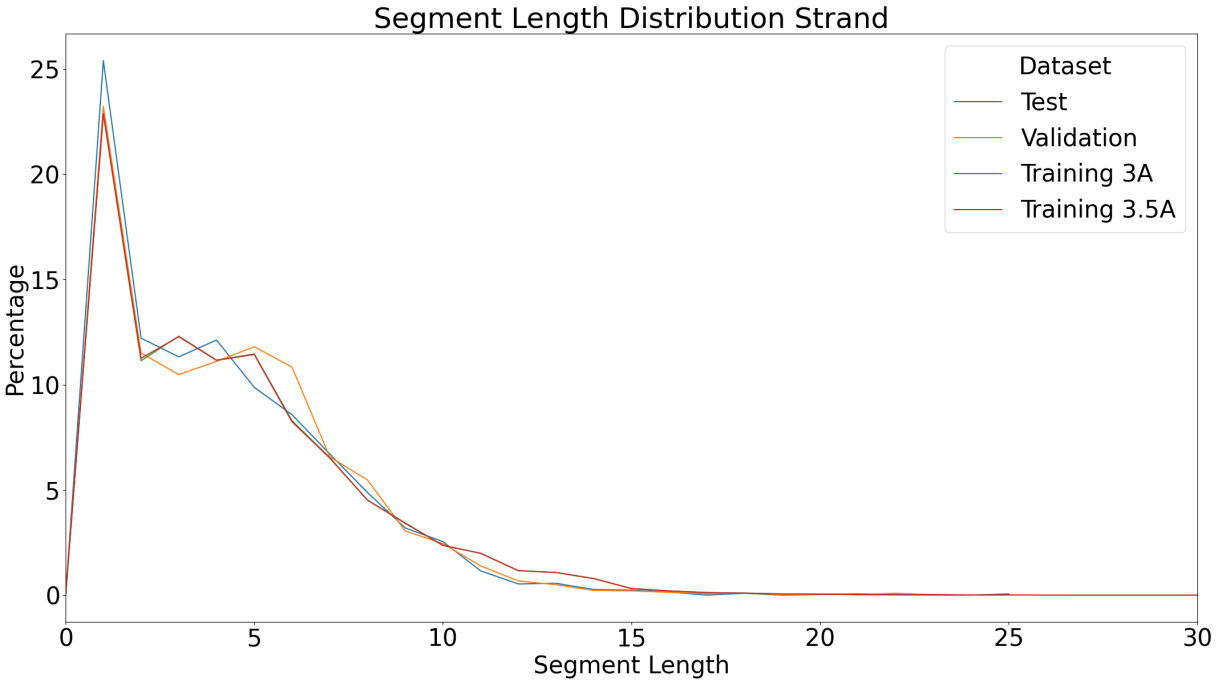

Figure 2.11: Segment length distribution for strand residues in training, test and validation datasets

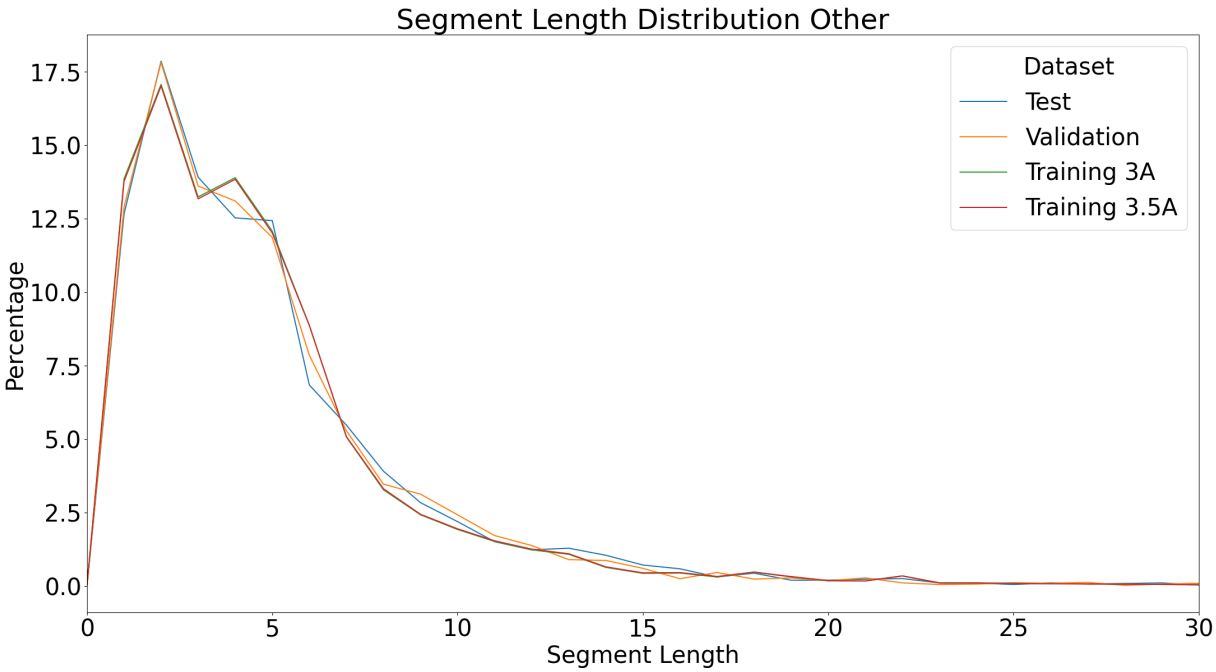

Figure 2.12: Segment length distribution for other residues in training, test and validation datasets

pendix in Figure 7.6, 7.7, 7.8, 7.9, 7.10, 7.11, 7.12, 7.13, 7.14 and 7.15

2.5 Model Development

As the model input, three different types of embedding were generated to investigate the effect of different ways of using the unsupervised language model information. An architecture from previous work was used and trained with identical learning rate, number of epochs and momentum. Type of embedding input and certain aspect of the training sets were varied to evaluate the effect of those changes on model performance.

2.5.1 Embeddings

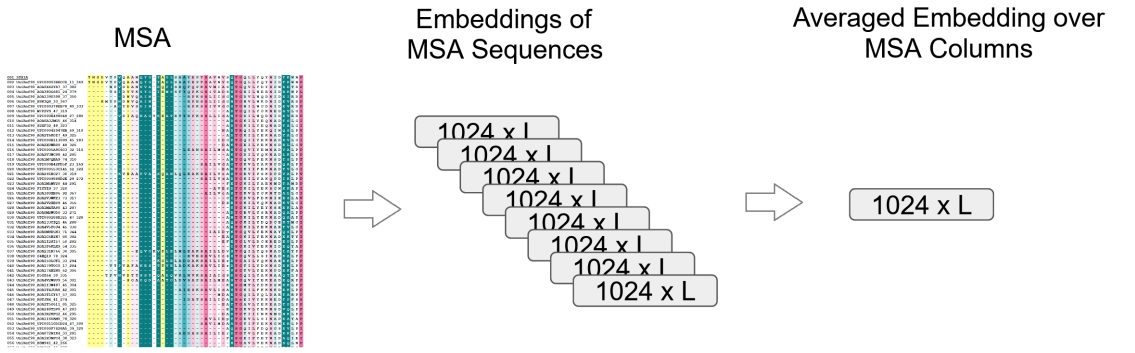

Figure 2.13: Schematic of the column wise averaged MSA creation

Embeddings were chosen as inputs for the structure prediction models. We decided to use different embedding types: Single sequence embeddings and MSA based embeddings. For the single sequence embeddings, the pretrained T5EncoderModel [3] from the python transformers package was used. For the averaged MSA embeddings all sequences in the corresponding alignment were embedded with the T5EncoderModel [3]. For each position in the original sequence, a new vector of length 1024 was computed, by summing over the embeddings for all sequences with a match at that position, and dividing each entry in the 1024 dimensional result vector by the number of matches. Only upper case letters were considered matches. A schematic of this process can be found in Figure 2.13.
We hypothesized, that different approaches of computing MSA embeddings may have an effect on prediction performance. To explore this idea, two ways of weighted MSA embeddings were computed. The first approach weights sequences in the MSA by their similarity to the query sequence. A single sequences weight is computed by the following formula:

$$sequence\ weight = \frac{\#aligned\ residues}{alignment\ length} * 100 \tag{2.1}$$

By having high weights for increasingly similar sequences and lower weights for more divergent sequences, emphasis is put on highly similar and well conserved sections of the protein. As an alternative, averaged embeddings, that empathize dissimilarity between aligned sequences, were tested by using the following formula for sequence weights:

$$sequence\ weight = 1 - \frac{\#aligned\ residues}{alignment\ length} * 100 \tag{2.2}$$

Averaged MSA embeddings, computed by using the second formula, are expected to capture the divergence in the protein space better.

2.5.2 Model Architecture

The best performing architecture from the preliminary work was slightly modified and used for the secondary structure predictions of this work. The network expects the embeddings of seven consecutive residues, centered around the current prediction target, as input. At the beginning and end of a sequence, the input was zero padded. The embeddings can be either single sequence or MSA embeddings, but they should correspond to the type of embedding used during training.
As an input layer, a 2D convolutional layer with one input and ten output channels, a kernel size of three and a stride of one was used with a leaky relu as its activation function. This is followed by a 2D max pooling layer. Afterwards, a second convolutional layer is used with ten input channels, ten output channels and a kernel size of two with another leaky relu as the activation function and again followed by a 2D max pooling layer. The output from the max pooling layer is reshaped by using the view function and then further processed by another two linear layers with leaky relu activation functions. The first linear layer has an input dimension of 2550 and 1275 outputs, which are then reduced to the three prediction outputs by the second linear layer.
The model was trained for ten epochs with cross entropy loss and stochastic gradient descent as optimizer. The learning rate was set to 0,001 and the momentum to 0,9.

2.5.3 Model Training

For model training, only the following 4 parameters were modified:

- Training set resolution
- Embedding input type
- Structure database
- Use of clusters during training

All other hyperparameters were kept the same, to allow for model comparison and effect estimation of the selected features.
To compare the effect of training set resolution, two training sets were generated as described in Section 2.4.3. This comparison was done to determine whether allowing sequences with higher resolutions improves performance, due to the availability of more data or decreases it, due to increased noisiness in the data and potentially less reliable structure data.
The networks were all trained on embedding based input. Results of the preliminary work suggested performance improvements if MSA based embeddings were used instead of single sequence embeddings. To confirm this finding, networks were trained on either single sequence or MSA based embeddings. We hypothesized, that different ways of generating MSA embeddings may have different effects on performance as well and generated weighted MSA embeddings to test this idea. Details on the embedding generation process can be found in Section 2.5.1.
Data quality is usually an important factor in machine learning tasks. For this project, we wanted to test if training on PDBredo secondary structure data improves performances compared to the data that can be obtained from PDB. To ensure a fair comparison, the same chains were used for the PDBredo and the PDB dataset. This avoids any differences

that may occur due to more available data or more difficult prediction targets in either one of the cases.
Lastly, we decided to investigate if training on sequence clusters instead of single sequences could be used to improve performance. It may be possible to benefit from small differences between highly similar sequences by using a different sequence of a cluster during each training epoch. To test this idea, half of our networks were trained using only the selected cluster representative, resulting in an identical set of protein sequences each epoch. The other half of our networks was trained on randomly selected members of our clusters, picking one of each cluster per epoch. To ensure consistency between the different networks that were trained on cluster members, a fixed seed was picked. This approach ensured that we would always get the same cluster members for a specific epoch.

2.6 Secondary Structure Prediction Evaluation

Multiple performance measures have been used for assessing the performance of structure prediction methods. The most common ones include Q_3, Matthews correlation coefficient and the fraction overlap of segments. Table 2.1 shows an example of how strongly those measures can deviate, even if one of them indicates identical performances.

For each trained network, a prediction for the validation set was written to a fasta

Table 2.1: Comparison of different structure predictions with different performance measures

		Q_3	MCC_{coil}	SOV
Amino acid sequence	AVRGWDRSAE			
Experimental secondary structure	HHHCCCEEEE			
Prediction 1:	CHHHCEEEEC	60%	∼0.047	72.5%
Prediction 2:	HHHCCCHHHH	60%	1	60%
Prediction 3:	EHECECECEE	60%	∼0.524	66.4%

file, containing the sequence and chain identifier and the predicted secondary structure. Those files were later used to compute Q_3, SOV and MCC. A summary of all results can be found in the Appendix in Tables 6.5, 6.6, 6.7 and 6.8
Additionally to the predicted secondary structure, averaged loss values were collected during training and plotted afterwards to visualize if different parameter combinations could have an effect on the required training time for reaching a local maximum.

2.6.1 Q_3

A performance measure that is very often used for secondary structure prediction accuracy is Q_3 for 3-state prediction and Q_8 for 8-state prediction. Q_x measures the percentage of residues for which x-state secondary structure is predicted correctly, with values ranging between 0 and 1. The following formula is used to compute it:

$$Q_3 = \frac{\#residues\ predicted\ correctly}{\#total\ residues} \tag{2.3}$$

The Q_x value expected for a random prediction depends on the secondary structure distribution of the underlying dataset. Comparison of different methods, based on Q_x can be difficult because it is possible for two predictions to result in an identical Q_x value, despite one of them being more meaningful than the other (see Table 2.1).
For Q_3 performance, the total number of residues and the correctly predicted residues were counted on a per protein basis and Q_3 was calculated according to Formula 2.3. The resulting distribution was used for mean and standard deviation (SD) computation and the standard error (SE) was computed using the following formula:

$$SE = \frac{SD}{\sqrt{n-1}} \tag{2.4}$$

with n being the number of predicted proteins.

2.6.2 Matthews Correlation Coefficient

Matthews correlation coefficient (MCC) is another frequently used performance measure and is defined as:

$$MCC_x = \frac{TP * TN - FN * FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2.5}$$

where TP is the number of correctly predicted residues with label x (true positives), FP the number of residues that have been predicted as x incorrectly (false positives), TN the number of residues correctly predicted as non x (true negative) and FN the number of residues incorrectly predicted as non x (false negative). MCC values range between -1 and 1, with a value of 0 indicating a completely random prediction, 1 being a perfect prediction and -1 meaning that all residues of type x have been predicted as non x and vice versa.
MCC was computed according to Formula 2.5 for each of the 3-state secondary structure prediction targets. Counts for TP, FP, TN and FN were collected on a per protein base and mean and standard deviation were computed from the resulting distribution. The standard error was computed according to Formula 2.4

2.6.3 Fractional Overlap of Segments (SOV)

As an alternative measure for secondary structure prediction accuracy, the fractional overlap of segments (SOV) has been proposed by Rost et al. in 1994 [13] and the formula has undergone refinement multiple times since then [9, 18]. SOV is computed by the following formula:

$$SOV = \frac{100}{N} \sum_{S_0} [\frac{minov(s_{obs}, s_{pred}) + \delta(s_{obs}, s_{pred})}{maxov(s_{obs}, s_{pred})} len(s_{obs})] \tag{2.6}$$

where s_{obs} is an experimentally observed segment, s_{pred} a predicted segment and s_0 the set of all overlapping pairs of s_{obs} and s_{pred}. $minov()$ is the length of the actual overlap between s_{obs} and s_{pred} and $maxov()$ the number of residues for which either s_{obs} or s_{pred} is in the relevant state. $len()$ is the number of residues in a specific state.
In protein sequences, some variation in segment boundaries can be observed and this is

represented by δ as a factor that allows for accounting for those variations.
For segment overlap score computation (SOV) the Pearl script from the Z.Wang Lab was used [9]. The script returns a SOV score according to the formula from 1999 as well as for the refined SOV score.
SOV was computed for each models final structure prediction of the validation set.

2.6.4 Confusion Matrix

For each model prediction of the validation set, a confusion matrix was generated by collecting counts for each matrix cell on a per protein basis. Confusion matrix cells were annotated with the mean performance over all predicted proteins and the standard error, which was computed on a per protein basis according to Formula 2.4.

3. Results

In the preliminary work, a performance of 73.2% (SE:0.6) was achieved on single sequence SeqVec embeddings and 78.3% (SE:0.6) for MSA embeddings. In this work, we were able to achieve a Q_3 performance of 80.2% (SE: 0.4) for single sequence T5 embeddings and 75.7% (SE: 0.5) for averaged MSA embeddings on the validation set.
The result presentation in the following sections will mostly focus on Q_3 performance.
Figures showing all performance measures for each trained network can be found in the Appendix from Figure 7.16 to 7.47, as well as the corresponding confusion matrices from Figure 7.48 to 7.79.

3.1 Effect of Sampling different Cluster Members during Training

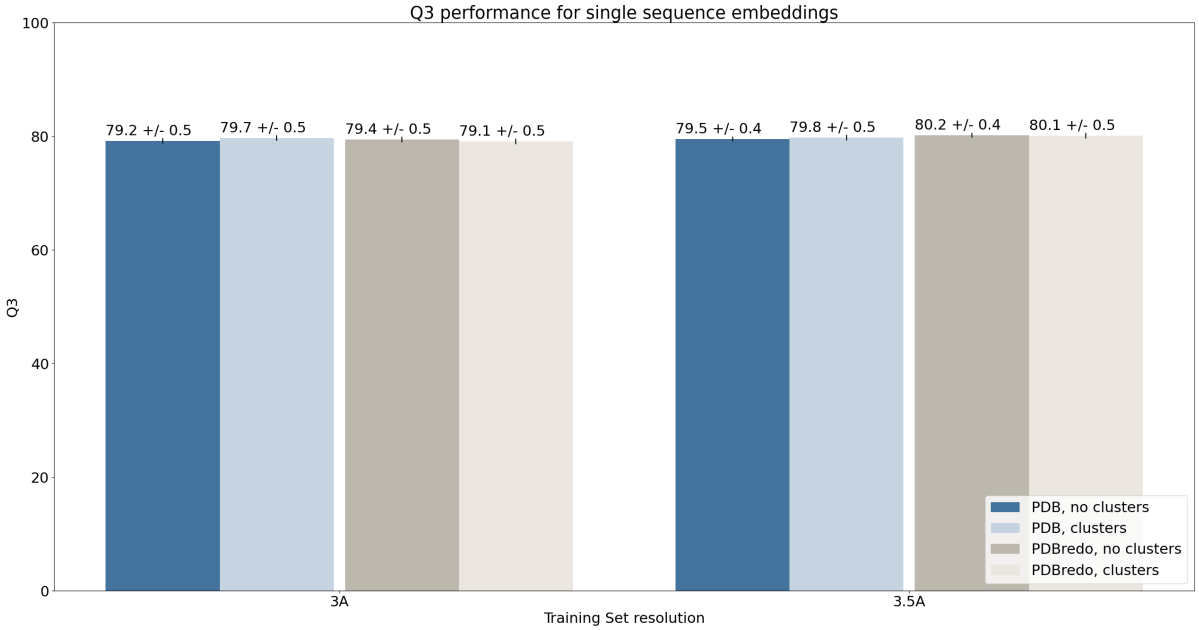

Figure 3.1: For single sequence embeddings Q_3 is between 79.1% and 80.2%. Neither training set resolution nor database or cluster training had a significant effect on the performance.

Figure 3.1 shows that for single sequence embeddings, no significant effect could be observed by training on different sequences from a cluster in each epoch compared to picking a cluster representative for each cluster and using the same sequence each epoch for either resolution training set from PDB. In cases of the 3Å set, a Q_3 of 79.2% (SE: 0.5)

was achieved for training without clusters and 79.7% (SE: 0.5) for training with clusters. For the 3.5Å training set, Q_3 values of 79.8% (SE:0.5) were achieved for cluster training and 79.5% (SE:0.4) without clusters. Required training time for convergence also didn't seem to in- or decrease for cluster training.
Q_3 prediction performance comparisons for cluster vs. representative training on the PDBredo training sets did not show a significant difference either.
For SOV scores, the trend was mostly confirmed. Only training on the 3Å PDB training set showed a slight performance increase of 1.9% from 67.9% (SE: 0.7) to 69.8% (SE: 0.7) for training on sequence clusters. For PDBredo and the 3.5Å PDB training set, changes were within the standard error margin.

Unweighted MSA embeddings did show performance differences for cluster training.

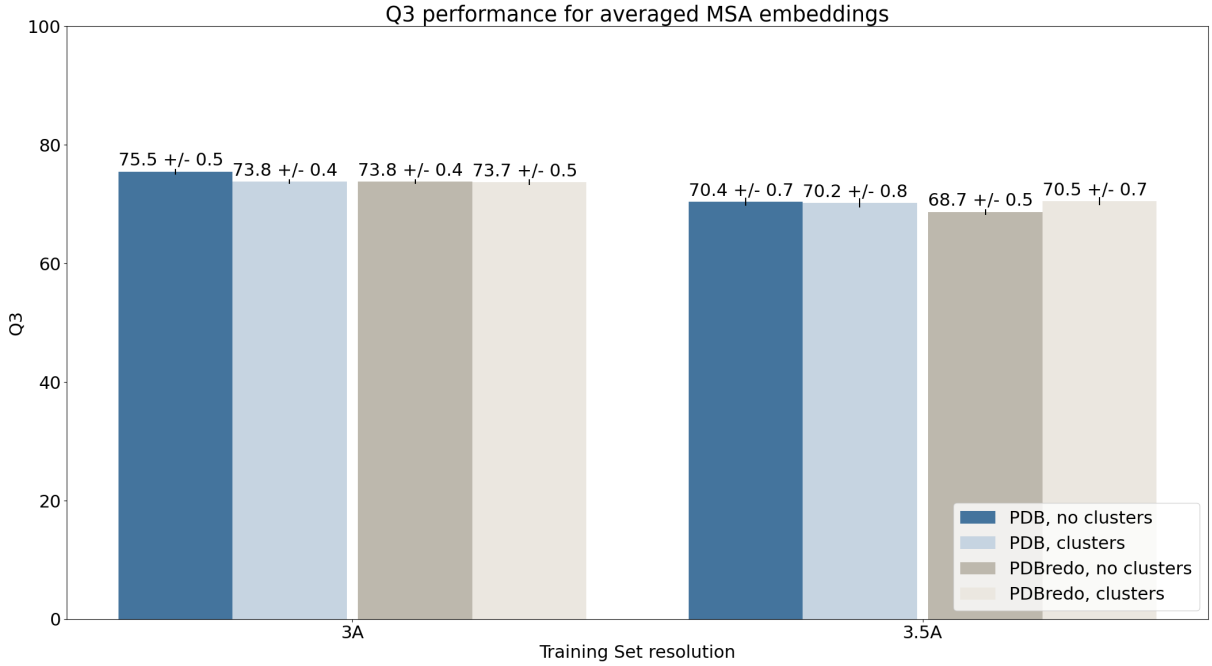

Figure 3.2: For averaged MSA embeddings Q_3 performance is between 68.7% and 75.5%. Using clusters during training caused a performance drop on the PDB 3Å set and a slight performance increase for the 3.5Å PDBredo training set.

For training on the 3Å dataset from PDB, cluster training resulted in a Q_3 performance of 73.8% (SE: 0.4) and training on representatives in 75.5% (SE: 0.5). Q_3 for the 3.5Å set did not change significantly. For the PDBredo dataset, no significant performance change was observed on the 3Å set, but the performance on the 3.5Å set increased from 68.7% (SE: 0.5) to 70.5% (SE: 0.7) (see Figure 3.2).
The refined SOV score confirms the performance drop for the 3Å PDB training set for cluster training, but the performance difference for the 3.5Å training set is within the standard error margin.
For weighted MSA embeddings, no significant performance increases could be observed for cluster training but a large performance drop from 69.3% (SE:0.6) to 62.8% (SE: 0.8) for the PDBredo 3.5Å training set. A smaller decrease in performance was also observed for the PDB 3Å training set (see Figure 3.3). If inverse weighted MSA embeddings were used instead, the result for the 3.5Å set was flipped: Performance increased from 67.7%

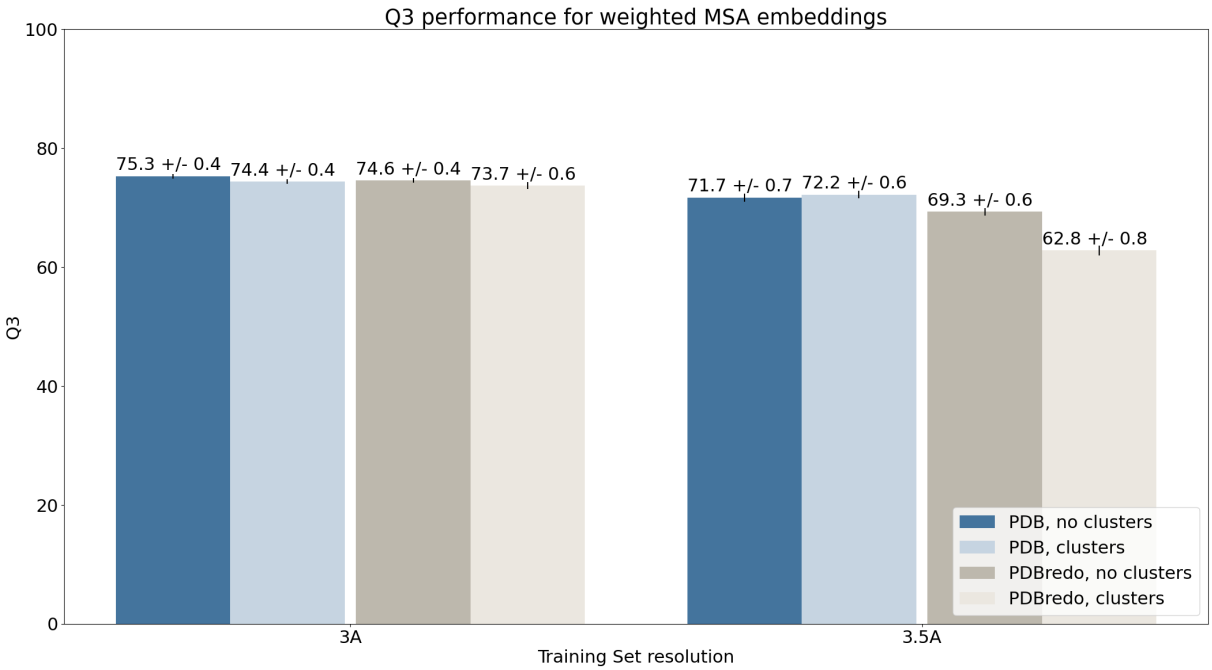

Figure 3.3: For weighted MSA embeddings, Q_3 performance is between 62.8% and 75.3%. Using clusters during training caused a performance drop on the PDB 3Å as well as for the 3.5Å PDBredo training set.

(SE:0.6) to 71.9% (SE:0.6). Differences for the PDB dataset were mostly within the standard error margin but a small performance decrease from 73.4% (SE:0.5) to 72.1% (SE:0.6) could be observed (refer to supplementary Table 6.8).

3.2 PDBredo vs. PDB

Figure 3.1 shows that Q_3 values for models trained on single sequence embeddings for PDB and PDBredo were within standard error range of each other (79.2% ± 0.5 for PDB for the 3Å training set without clusters and 79.4% ± 0.5 for PDBredo), but a difference could be observed for SOV results. Based on the SOV computation from 99, a performance of 72.9% and 67.9% (SE: 0.7) for the refined SOV score could be achieved on the PDB training set at 3Å, that trained on representatives. For PDBredo, a refined SOV score of 65.7% (SE: 1.0) and 71.2% based on the old SOV formula could be achieved (refer to Figure 3.4 and supplementary Table 6.5). The performance increase of 2.2 % (no clusters) and 4.3% suggests a significant improvement if PDB data is used for training instead of PDBredo.
Comparison of performance measures for 3.5Å and training with clusters show the same pattern. In this case, the performance can be improved by 2.6% by using PDB data for network training. The difference for training without clusters at this resolution is to small to be considered significant.

Q_3 results for MSA embeddings generally support that training on PDB data may have beneficial effects. The most noticeable difference can be observed for 3.5Å and clusters on weighted MSA embeddings. The performance drops from 72.2% (SE: 0.6) to 62.8% (SE:0.8), as shown in Figure 3.3.

Figure 3.4: For single sequence embeddings, SOV (refined) is between 65.5% and 70%. Training set resolution does not have a significant effect on the performance. A difference between cluster and representative training can only be observed for the PDB training set. Training on PDB also results in a slight performance improvement in most cases compared to PDBredo.

3.3 Effect of Method Resolution for Training Set Generation

Including sequences of up to 3.5Å did not seem to affect prediction performance on single sequence embeddings. The models trained on PDB with clusters achieved a Q_3 of 79.7% (SE:0.5) for the 3Å set and 79.8% (SE:0.5) for the 3.5Å set. The PDB models that were trained without clusters achieved a Q_3 performance of 79.2% (SE: 0.5) and 79.5% (SE: 0.4).
Using PDBredo training sets instead of PDB did not result in Q_3 improvements or decreases for different resolution cutoffs either (see Figure 3.1).

The overall picture changes if we look at MSA based embeddings. For most cases of MSA based embeddings performance was significantly lower for the 3.5Å sets (refer to Figure 3.2, 3.3 and supplementary Table 6.8).

3.4 Single Sequence Embeddings vs. averaged MSA Embeddings

Figure 3.5 shows that a performance drop of 3.5% to 10% could be observed if averaged MSA embeddings were used instead of single sequence embeddings, with single sequence embeddings reaching a Q_3 performance of close to 80% and averaged MSA embeddings a maximum of 75.7% on the PDB training set.

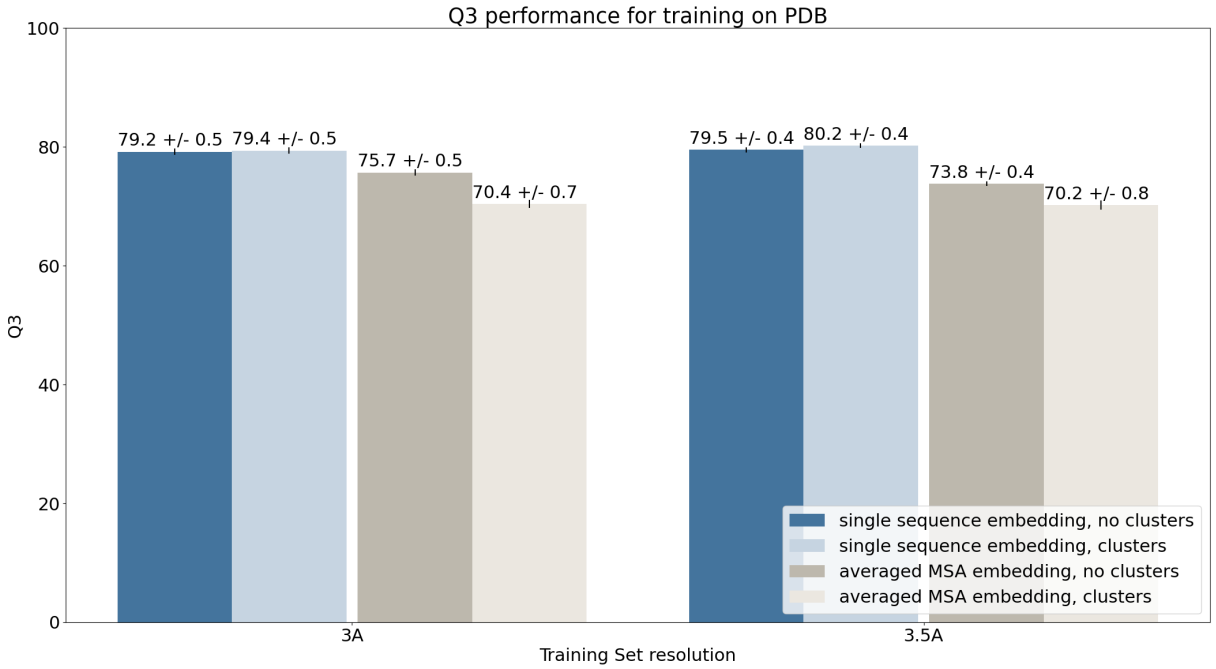

Figure 3.5: Q_3 scores for single sequence embeddings on the PDB dataset reach $\sim$ 80%. Performances for averaged MSA embeddings are significantly lower, reaching less than 76%. Using cluster training seems to additionally decrease the prediction performance for embedding type.

As already discussed in Section 3.1, performance decreases further to around 70% if different sequences from the same cluster are used in each training epoch.

3.5 Effect of weighting by Sequence Similarity in averaged MSA Embeddings

Figure 3.6 shows that averaged MSA embeddings without any weighting performed best on the 3Å training set without clusters, reaching a Q_3 of 75.7% (SE: 0.5) and inversely weighted averaged MSA embeddings worst at 73.8% (SE:0.4). A contradicting trend can be observed for the 3.5Å training set without clusters and unweighted averaged embeddings achieving the lowest performance at 70.4% (SE: 0.7) and inverse weighted embeddings getting a Q_3 of 73.7% (SE: 0.6).
For the sets trained with clusters, no significant difference can be observed for the 3Å set. The results on the 3.5Å set suggests that weighted embeddings may be able to achieve a higher performance. The difference between those weighted and inverse weighted MSA embeddings is not significant enough to draw conclusions about which type of embedding is better suited.

For PDBredo (Figure 3.7), the most significant result is the performance drop to 62.8% for the model trained on weighted MSA embeddings, sequence clusters and the training set with a resolution of up to 3.5Å.

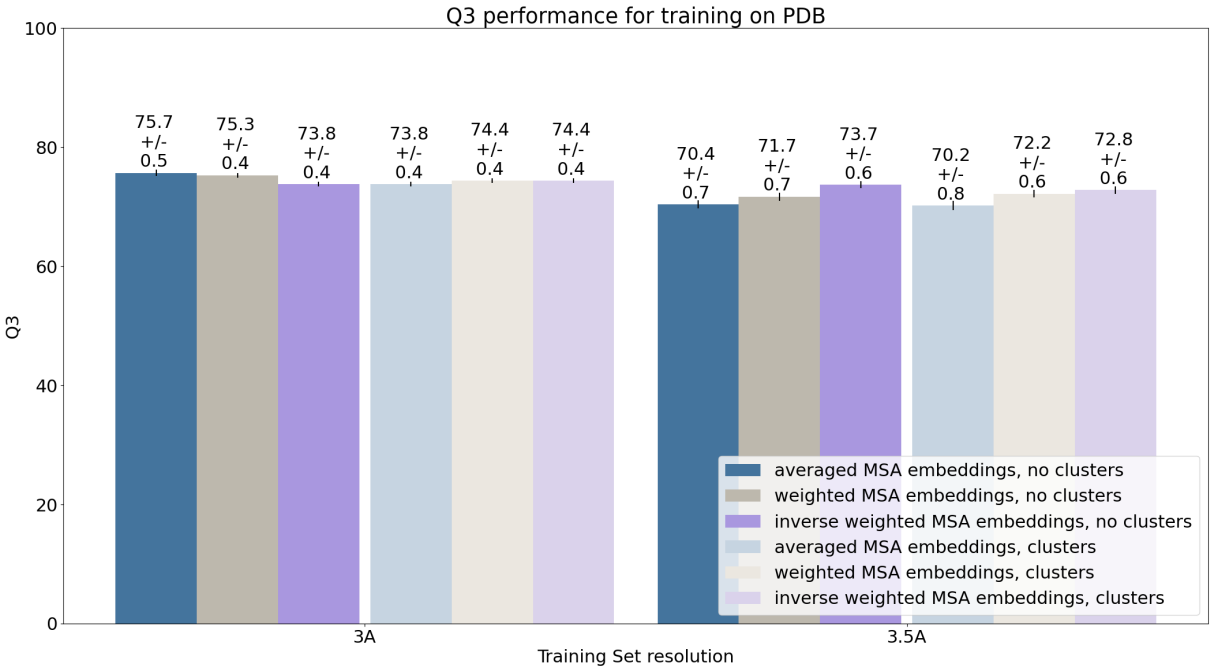

Figure 3.6: Q_3 performances for different types of averaged MSA embeddings show contradictory results for the different resolution training sets. The bars show the obtained results for training on PDB with and without clusters.

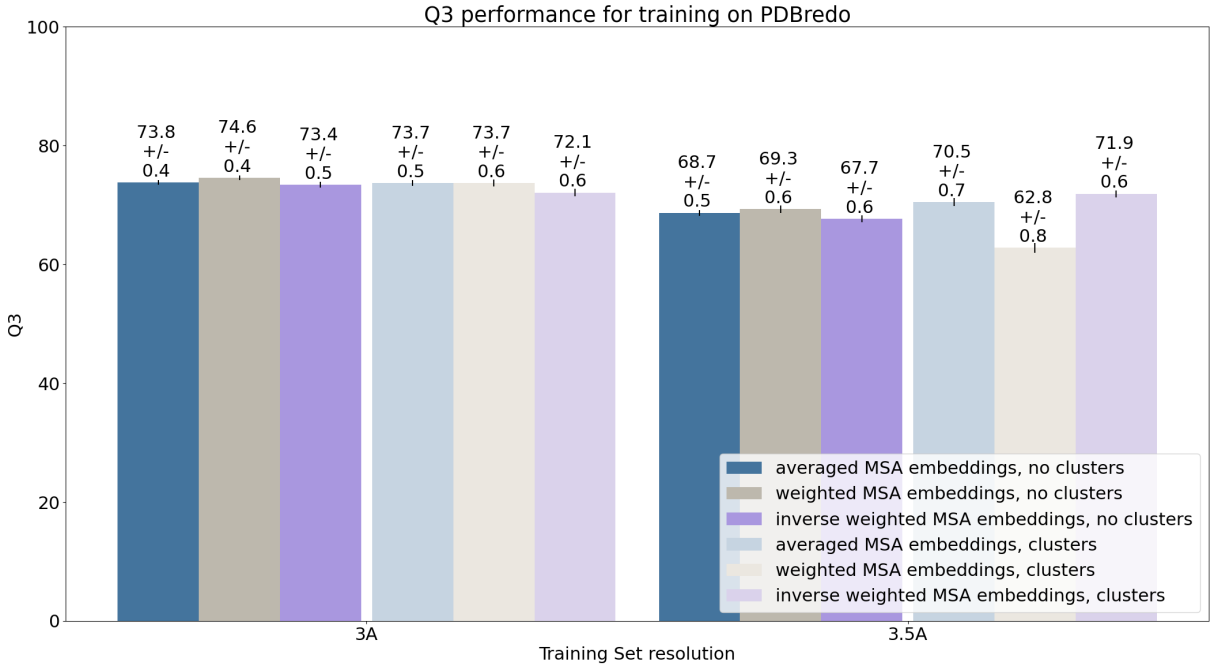

Figure 3.7: Q_3 values for training with and without clusters on the PDB redo sequences. The lowest performance can be observed for the model trained on weighted MSA embeddings with sequence clusters at a resolution of 3.5Å, achieving 62.8% (SE: 0.8), and the highest for models using the same embedding type but training only on lower resolution sequences and without clusters.

4. Discussion

In the following section, possible reasons for the observed results will be discussed. Based on those, future research questions can be identified, which could be explored in the near future.

4.1 Conclusions

In this work, we tried to investigate the effect of training set resolution, embedding type, use of clusters during training and the database from which sequence and structure were obtained. Additionally we tried to confirm the results obtained in our preliminary research.
The data obtained in this project implies a performance difference between single sequence embeddings and all tested variations of averaged MSA embeddings. Contrary to the results from the preliminary work, using MSA embeddings resulted in a performance decrease. Multiple possible reasons are discussed in Section 4.1.4, that will have to be investigated before reaching final conclusions.
For the different training parameters, we observed that Q_3 performance does not show a significant performance difference for single sequence embeddings if the database was changed, clusters were used for training or different resolution cutoffs were chosen. MSA embeddings show some differences for the different parameters. If the refined SOV score is used for performance evaluation, instead of Q_3, small differences can be observed for those parameters on single sequence embeddings as well.

4.1.1 Cluster Training

Q_3 performance suggests that there is no significant difference between cluster training and training on representatives for single sequence embeddings, but some differences could be observed for the different types of MSA embeddings. However, even on MSA embeddings, no clear trends could be observed. In some cases, clusters resulted in a performance increase but decreases were observed as well.
It is possible that the training time of only ten epochs is not enough to benefit from the different sequences in the clusters. Training the network for longer would allow it to learn from more similar sequences of the same cluster. Furthermore, in this work, sequences from the same cluster were selected randomly. Including a selection criteria that either selects the most similar sequences to the cluster representative or tries to diversify the selection as much as possible, could lead to a stronger observable effect.

4.1.2 Database Selection

The most likely reason for the improvements achieved by using PDB are sequence differences between PDB and PDBredo. The DSSP files from PDBredo are often missing residues at the beginning and the end of the sequence. These same residues are usually annotated as disordered in PDB. The amino acids in those positions may carry additional information that restricts the possible structures in other parts of the sequence. This additional information may be either captured in the embeddings or affect model performance during predictions by adding additional information during the convolution step for residues next to disordered regions. For future work, we recommend including additional sequence information if available, even if no structural information can be obtained. The residues with missing structural annotations can be masked during training and for performance evaluation.

4.1.3 Method Resolution

Method resolution did not seem to have a significant effect on models trained on single sequence embeddings, but generally decreased performance for higher resolutions if MSA based embeddings were used. Those results could imply that well resolved structures may be better suited for MSA embedding approaches.

4.1.4 Embedding Variations

There are multiple possible causes for the unexpected performance decrease of averaged MSA embeddings as compared to single sequence embedding. The first possible problem could be a simple bug in the MSA embedding pipeline that causes the averaged MSA embeddings to be calculated incorrectly.
A second possible explanation may be a checkpoint change of the pre-trained T5 transformer model that was used for sequence embedding creation. MSA embeddings were created over a time span of multiple days and processed in multiple batches. The exact day and time of the checkpoint change is currently unknown, but we have been made aware of a recent change of it. If some of the MSA embeddings were created with the previous checkpoint and others with the newer one, the resulting embeddings may have some important differences that make it impossible to reach the best possible performance with a combination of embeddings from both checkpoints. To exclude this potential error source, embeddings will have to be recomputed and models retrained.
Thirdly, the performance decrease could be caused by the embeddings themselves. In the preliminary work SeqVeq embeddings were used instead of T5. It is possible that SeqVec embeddings are more suitable for this kind of embedding modification due to different feature representations. T5 embeddings may already capture some of what can be gained by averaging over MSAs and the information contained in the embeddings could become more difficult to learn with our approach.
We have to also consider the differences in the MSAs used as a possible reason for the differing results of the preliminary work and that presented here. In the preliminary work, MSAs contained a maximum of 300 sequences, which were also highly similar to each other (E-value threshold: 0,0001). In this work, the sequences in the MSAs were filtered to only keep the most diverse sequences, using the --diff flag. Additionally, the mean

number of sequences in an alignment is more than three times as high as the maximum in the preliminary work (see Section 2.3.6). Besides those aspects, variations in the number of aligned sequences per MSA position or the length of gap segments in aligned sequences could affect how suitable a specific MSA is for averaged MSA embedding creation.
Lastly, the effect of differences between the training and validation sets employed here, as compared to those used in the preliminary work, should be considered. In this work, we enforced that no CATH topology that is contained in the validation set is part of the training set. No such restriction was applied to the training and validation set in the preliminary work. The former results in the models having to learn more general aspects and not being able to gather information about any features that may be specific to the topologies in the validation set. However, if those topology-specific features are necessary to use the information encoded in averaged MSA embeddings, validation performance could be affected negatively. Other differences between the two datasets, which are currently unknown to us, may also cause the observed results.

No generalizable statement could be made about the different MSA embeddings. Different databases, resolutions and if clusters were used for training resulted in different trends, if a significant difference was observed at all. The performance drop to 62.8%, observed for weighted MSA embedding that have been trained with clusters on PDBredo, was due to the model rarely predicting any residues to be of type strand (see supplementary Figure 7.78).
The potential problems listed above for MSA embeddings are also likely to have an effect on weighted MSA embeddings. Therefore, differences between those types will have to be investigated more in future work to be able to gain a better understanding of their impact on model training.

4.2 Future Research

Future work should first focus on excluding any problem sources that may be related to the embedding creation process. This includes checking the correct functionality of the embedding pipeline itself as well as creating new MSA embeddings while ensuring the transformer checkpoint for the T5 embedder and encoder remains identical for the entire embedding process. After those steps, other factors that may influence performance, like dataset differences, suitability of different embedding types and MSA aspects, can be investigated further.
If the performance increase we observed in the preliminary work can be confirmed for T5 embeddings, future research could focus on the applicability for other protein feature prediction tasks like solvent accessibility or localization. Furthermore, currently existing methods, based on T5 embeddings, might be able to achieve better prediction performances if they are retrained on averaged T5 embeddings.
The models trained in this work were able to achieve a performance of close to 80% on the constructed validation set without optimization of most hyperparameters (e.g. learning rate, number of epochs,...). It may be possible to improve model performances further through a grid search and early stopping. It may be interesting to test those optimized models on standardized test sets like CASP14. The results obtained that way would allow for comparison to state of the art methods. Evaluating our model on CASP14 and our own test set would allow us to infer the difficulty of our set in comparison to CASP14.

Part II

Appendix

5. Glossary

API: Application Programming Interface
CASP: Critical Assessment of Structure Prediction
EM: Electron Microscopy
FN: False Negative
FP: False Positive
HMM: Hidden Markov model
HSSP curve: Curve that represents an empirically determined threshold for automatic family assignment
Hval: Distance to HSSP curve
MCC: Matthews Correlation Coefficient
MSA: Multiple Sequence Alignment
NMR: Nuclear Magnetic Resonance
PDB: Protein Data Bank
PID: Percentage Sequence Identity
PSSM: Position Specific Scoring Matrix
SD: Standard Deviation
SE: Standard Error
SOV: Segment Overlap
TN: True Negative
TP: True Positive

6. Additional Tables

Table 6.1: Number of chains in the intersection of PDB (data obtained on 26.10.20), PDBredo and CATH after filtering by resolution and profile similarity based on the PISCES similarity matrix

	No similarity constraint	**similarity ≤30%**	**similarity ≤25%**	**similarity ≤20%**
Resolution ≤3Å	225 161	16 324	13 222	8 988
Resolution ≤2.5Å	180 115	14 536	11 861	8 110
Resolution ≤2Å	102 981	10 577	8 801	6 253
Resolution ≤1.2Å	4 385	984	924	811
Resolution <1.2Å	3 609	816	762	676

Table 6.2: Number of chains in the intersection of PDB (data obtained on 26.10.20), PDBredo and CATH after filtering by resolution and probability based on the PISCES matrix

	No probability constraint	**probability ≤50%**	**probability ≤30%**	**probability ≤25%**	**probability ≤20%**
Resolution ≤ 3Å	225 161	3 795	3 492	3 413	3 290
Resolution ≤ 2.5Å	180 115	3 505	3 246	3 177	3 073
Resolution ≤ 2Å	102 981	2 829	2 627	2 569	2 483
Resolution ≤ 1.2Å	4 385	547	526	522	514
Resolution < 1.2Å	3 609	474	456	454	445

Table 6.3: Number of clusters after clustering by similarity and probability constraint. Data used for clustering: Intersection of PDB (data obtained on 11.01.21) and CATH after filtering by resolution

	No constraint	**similarity >30%, probability >50%**	**similarity >25%, probability >50**	**similarity >20%, probability >50**
Resolution ≤3Å	290 107	19 018	15 177	10 445
Resolution ≤2.5Å	218 850	16 546	13 454	9 291
Resolution ≤2Å	120 522	11 599	9 526	7 149
Resolution ≤1.2Å	4 743	1 053	982	861
Resolution <1.2Å	38 888	867	809	724

Table 6.4: Number of clusters after clustering by similarity and probability constraint. Data used for clustering: Intersection of PDB (data obtained on 11.01.21), PDBredo and CATH after filtering by resolution

	No constraint	**similarity >30%, probability >50%**	**similarity >25%, probability >50**	**similarity >20%, probability >50**
Resolution ≤3Å	267 250	18 440	14 857	9 967
Resolution ≤2.5Å	202 777	15 902	12 610	9 158
Resolution ≤2Å	112 408	11 406	9 518	6 884
Resolution ≤1.2Å	4 517	1 011	950	843
Resolution <1.2Å	3 708	834	781	698

Table 6.5: Performance measures for all tested models and evaluation methods that use single sequence embeddings as input

Training Dataset	**Structure Source**	**Clusters?**	**Input type**	**SOV 99**	**SOV refined**	**Q_3**	**MCC_H**	**MCC_E**	**MCC_O**
3Å	PDB	no	single sequence embedding	0.729	0.679	0.792	0.702	0.619	0.607
3.5Å	PDB	no	single sequence embedding	0.737	0.686	0.795	0.7	0.636	0.608
3Å	PDB	yes	single sequence embedding	0.748	0.698	0.797	0.702	0.642	0.611
3.5Å	PDB	yes	single sequence embedding	0.749	0.7	0.798	0.699	0.643	0.615
3Å	PDBredo	no	single sequence embedding	0.712	0.657	0.794	0.702	0.636	0.609
3.5Å	PDBredo	no	single sequence embedding	0.725	0.671	0.802	0.718	0.648	0.626
3Å	PDBredo	yes	single sequence embedding	0.71	0.655	0.791	0.695	0.627	0.607
3.5Å	PDBredo	yes	single sequence embedding	0.728	0.674	0.801	0.707	0.646	0.618

Table 6.6: Performance measures for all tested models and evaluation methods that use averaged MSA embeddings as input

Training Dataset	**Structure Source**	**Clusters?**	**Input type**	**SOV 99**	**SOV refined**	**Q_3**	**MCC_H**	**MCC_E**	**MCC_O**
3Å	PDB	no	averaged MSA embedding	0.7	0.644	0.757	0.648	0.603	0.538
3.5Å	PDB	no	averaged MSA embedding	0.632	0.572	0.704	0.561	0.536	0.471
3Å	PDB	yes	averaged MSA embedding	0.634	0.58	0.738	0.609	0.569	0.531
3.5Å	PDB	yes	averaged MSA embedding	0.632	0.574	0.702	0.561	0.565	0.469
3Å	PDBredo	no	averaged MSA embedding	0.636	0.575	0.738	0.603	0.548	0.529
3.5Å	PDBredo	no	averaged MSA embedding	0.54	0.482	0.687	0.579	0.481	0.48
3Å	PDBredo	yes	averaged MSA embedding	0.649	0.586	0.737	0.594	0.562	0.516
3.5Å	PDBredo	yes	averaged MSA embedding	0.548	0.493	0.705	0.651	0.415	0.489

Table 6.7: Performance measures for all tested models and evaluation methods that use weighted MSA embeddings as input

Training Dataset	**Structure Source**	**Clusters?**	**Input type**	**SOV 99**	**SOV refined**	**Q_3**	**MCC_H**	**MCC_E**	**MCC_O**
3Å	PDB	no	weighted MSA embedding	0.685	0.627	0.753	0.639	0.594	0.531
3.5Å	PDB	no	weighted MSA embedding	0.639	0.577	0.717	0.582	0.55	0.483
3Å	PDB	yes	weighted MSA embedding	0.646	0.59	0.744	0.623	0.563	0.536
3.5Å	PDB	yes	weighted MSA embedding	0.647	0.587	0.722	0.587	0.562	0.488
3Å	PDBredo	no	weighted MSA embedding	0.643	0.583	0.746	0.628	0.566	0.548
3.5Å	PDBredo	no	weighted MSA embedding	0.551	0.492	0.693	0.58	0.518	0.493
3Å	PDBredo	yes	weighted MSA embedding	0.645	0.584	0.737	0.594	0.575	0.517
3.5Å	PDBredo	yes	weighted MSA embedding	0.423	0.373	0.628	0.538	0.062	0.397

Table 6.8: Performance measures for all tested models and evaluation methods that use inverse weighted MSA embeddings as input

Training Dataset	**Structure Source**	**Clusters?**	**Input type**	**SOV 99**	**SOV refined**	**Q_3**	**MCC_H**	**MCC_E**	**MCC_O**
3Å	PDB	no	inverse weighted MSA embedding	0.678	0.619	0.738	0.62	0.571	0.504
3.5Å	PDB	no	inverse weighted MSA embedding	0.66	0.6	0.737	0.607	0.559	0.512
3Å	PDB	yes	inverse weighted MSA embedding	0.649	0.593	0.744	0.622	0.564	0.534
3.5Å	PDB	yes	inverse weighted MSA embedding	0.658	0.596	0.728	0.584	0.57	0.496
3Å	PDBredo	no	inverse weighted MSA embedding	0.632	0.571	0.734	0.609	0.539	0.527
3.5Å	PDBredo	no	inverse weighted MSA embedding	0.53	0.471	0.677	0.556	0.507	0.47
3Å	PDBredo	yes	inverse weighted MSA embedding	0.626	0.562	0.721	0.568	0.527	0.503
3.5Å	PDBredo	yes	inverse weighted MSA embedding	0.589	0.529	0.719	0.626	0.479	0.508

7. Additional Figures

7.1 Venn Diagram of IDs

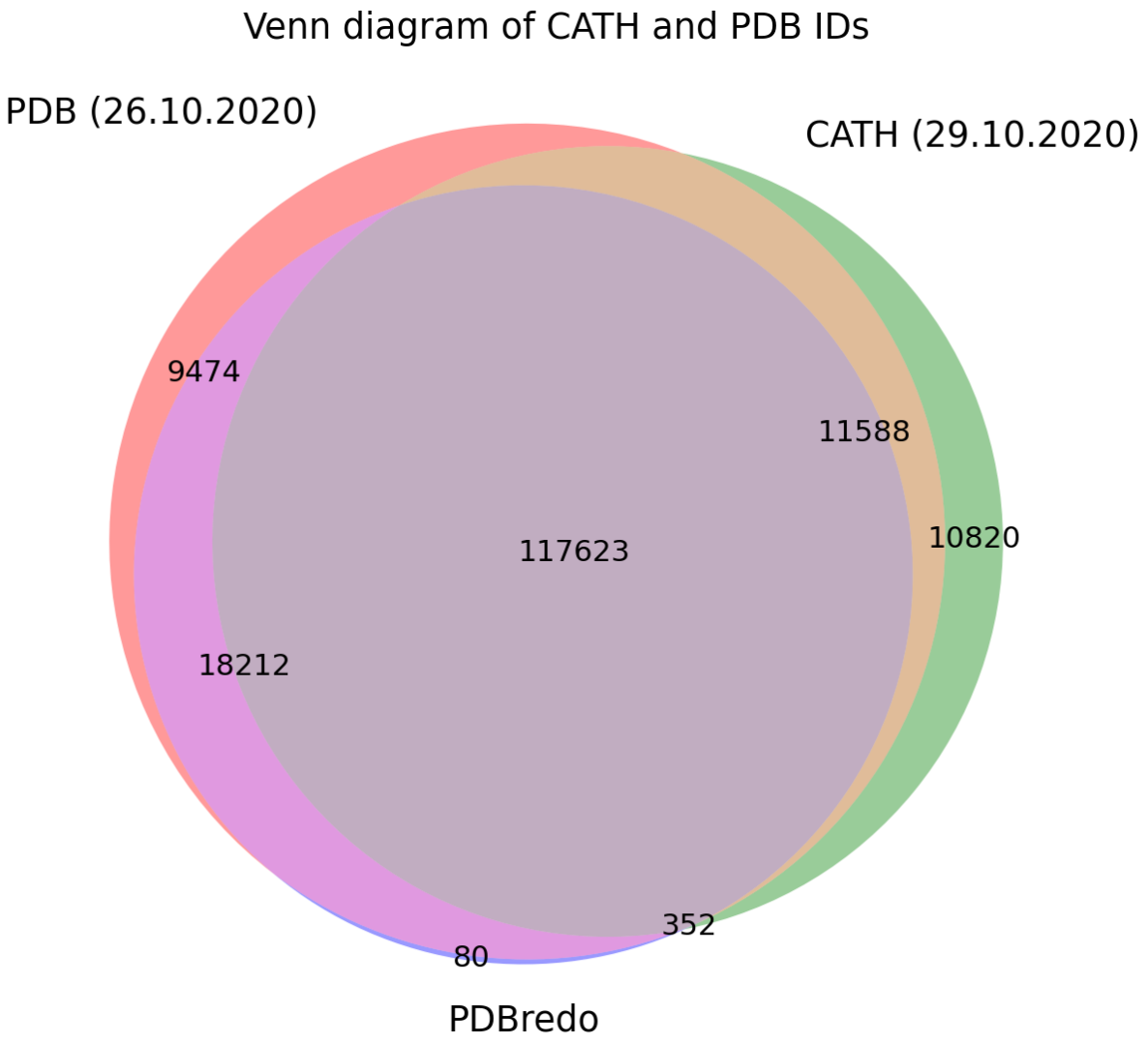

Figure 7.1: Venn diagram of the number of proteins in PDB, PDBredo and CATH and the size of the individual overlaps

7.2 Method Distribution PDB

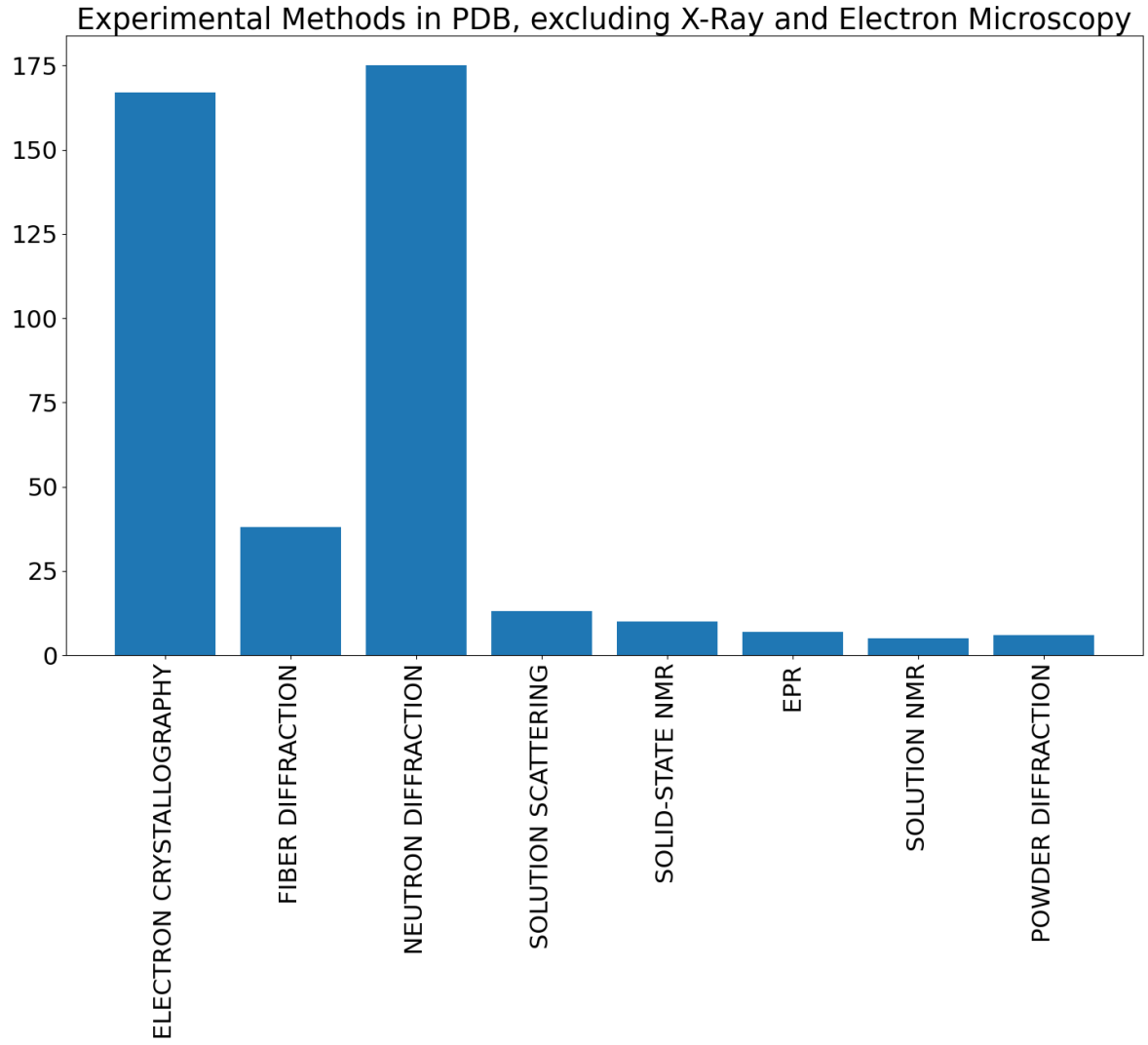

Figure 7.2: Experimental structure resolution methods for the intersection of PDB, PDBredo and CATH at a resulution of 3Å after excluding X-ray diffraction and electron microscopy. Individual counts: Electron crystallography: 167, Fiber diffraction: 38, Neutron diffraction: 175, solution scattering: 13, solid-state nmr: 10, EPR: 7, Solution NMR: 5, powder diffraction: 6

7.3 Frequency Histograms

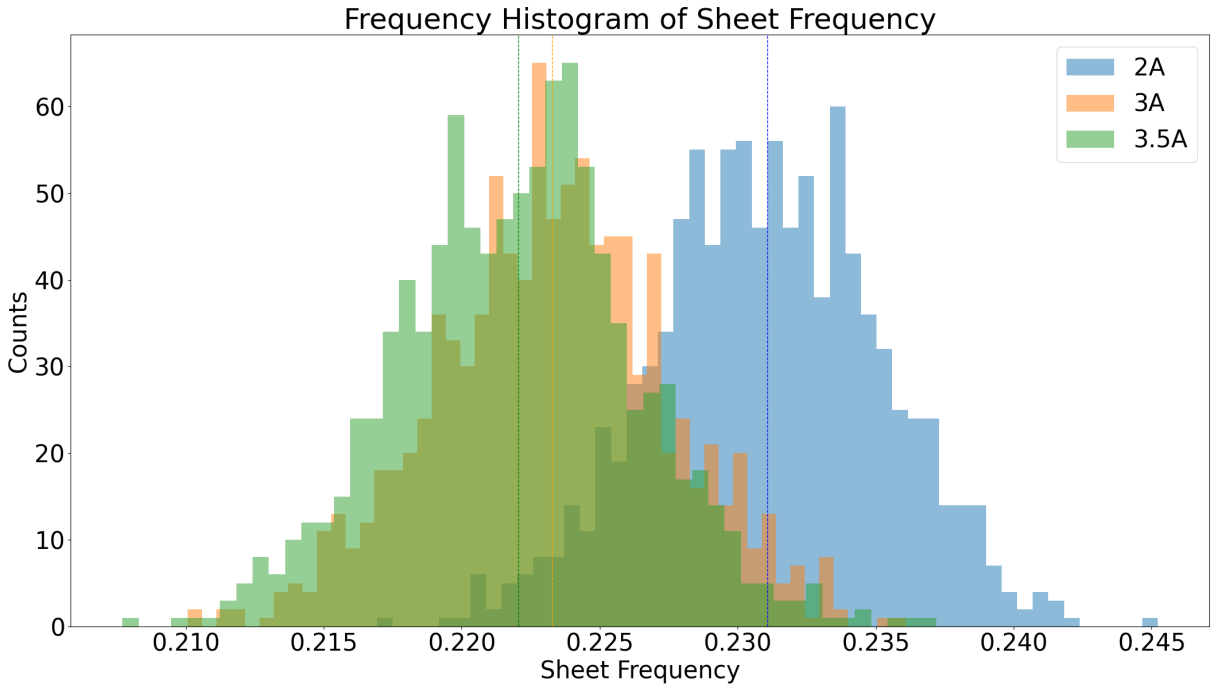

Figure 7.3: Distribution of strand residues after bootstrapping from the chains that were in PDB, PDBredo and CATH. Bootstraping was done by selecting 1000 random sequences 1000 times and counting the number of helix residues for each iteration. Vertical lines indicate the mean for each resolution.

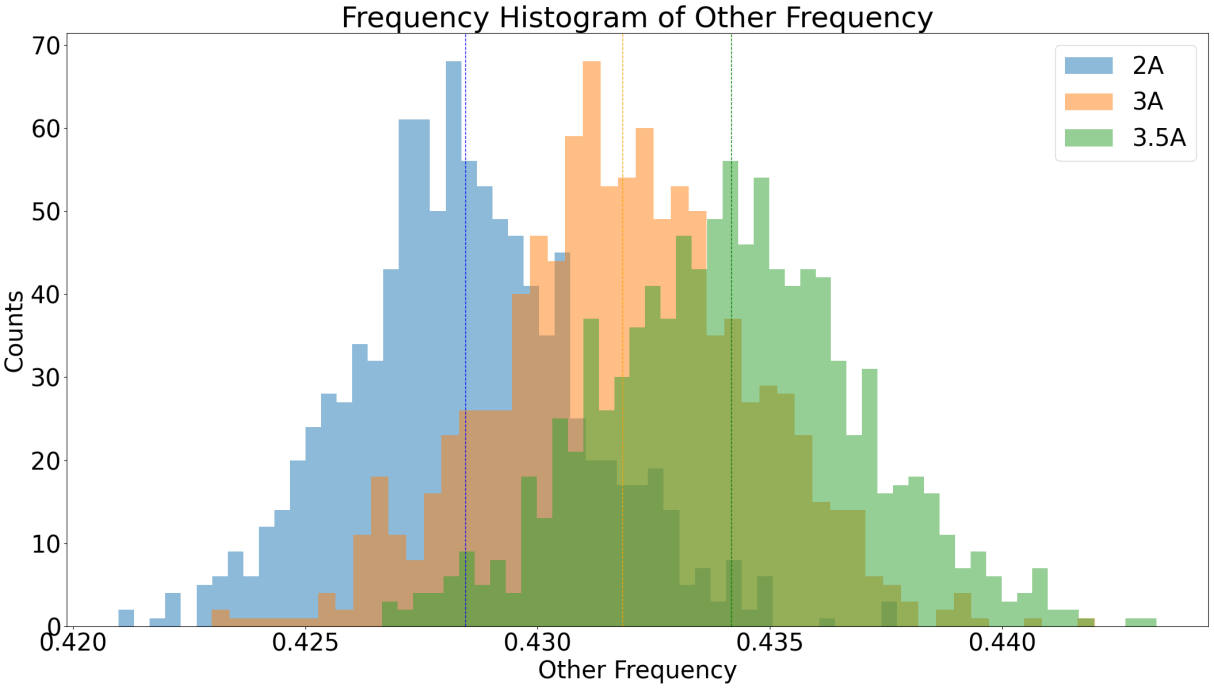

Figure 7.4: Distribution of other residues after bootstrapping from the chains that were in PDB, PDBredo and CATH. Bootstraping was done by selecting 1000 random sequences 1000 times and counting the number of helix residues for each iteration. Vertical lines indicate the mean for each resolution.

7.4 MSA Query Sequence Length Distribution

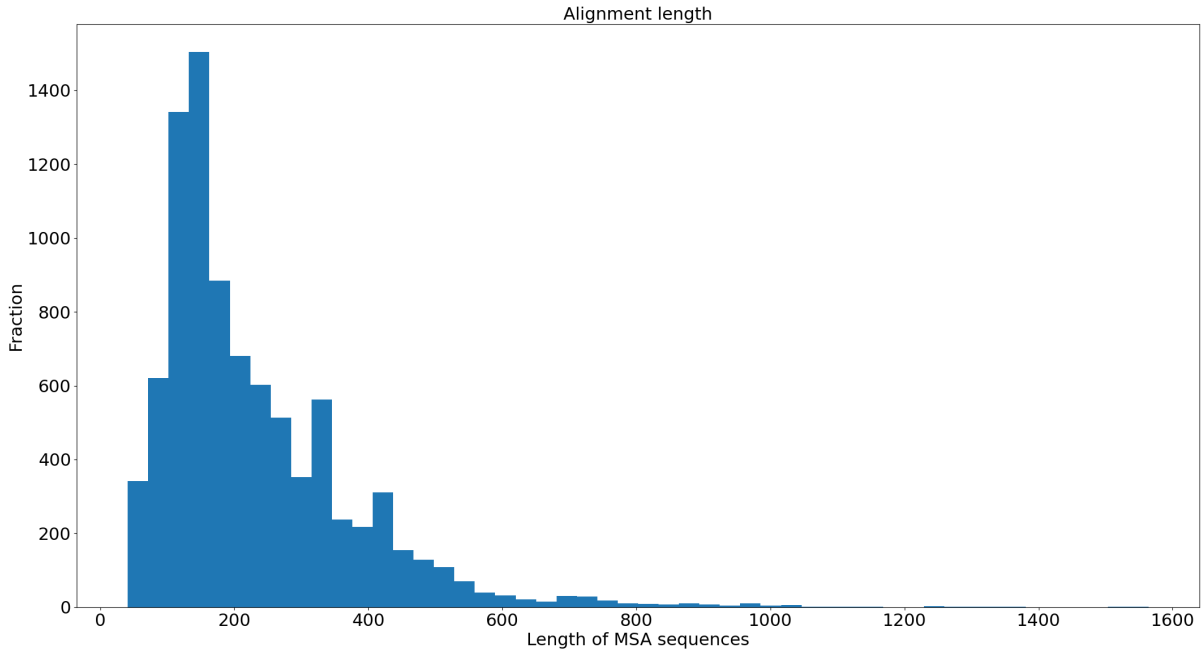

Figure 7.5: Lenght distribution of query sequences for MMseqs2 MSAs. Most query sequences are between 100 and 400 residues long. There are no query sequences with less than 40 residues and only very few have more than 1000 residues

7.5 Eight State Structure Distributions

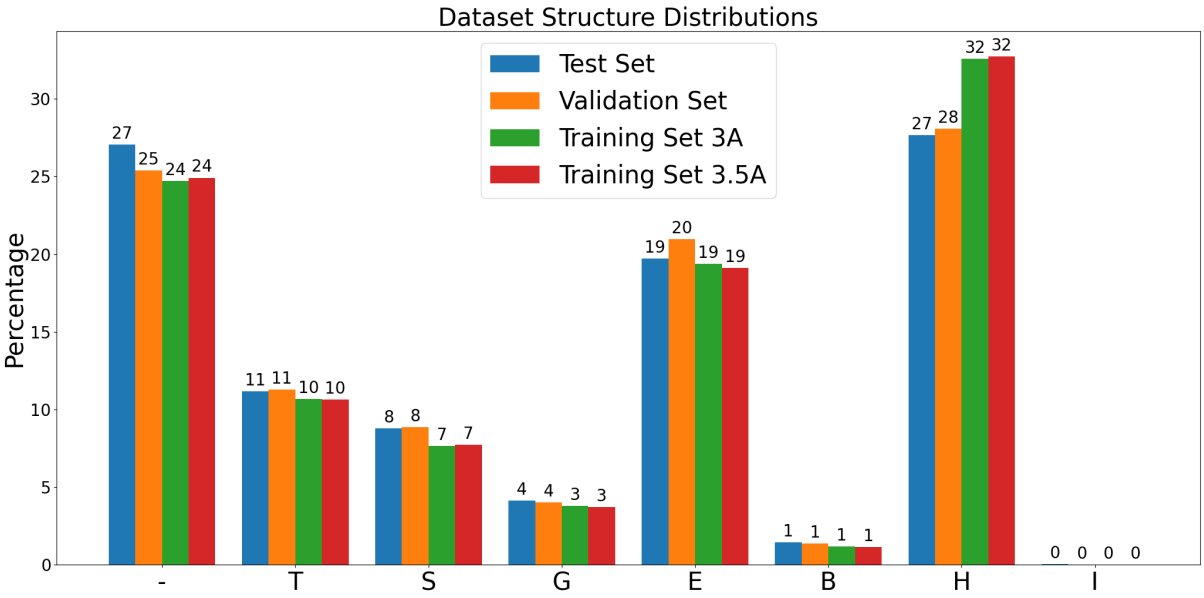

Figure 7.6: 8-state secondary structure distribution for training, test and validation sets

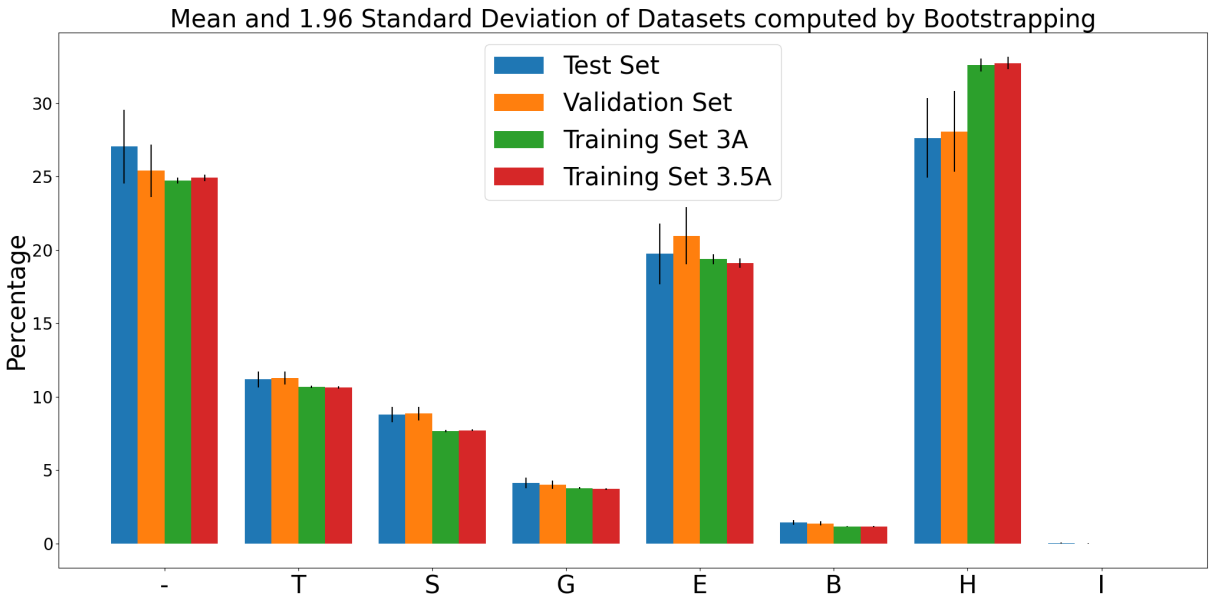

Figure 7.7: mean of 8-state secondary structure distribution for training, test and validation sets with 1,96 standard error as wiskers

7.6 Eight State Segment Length Distributions

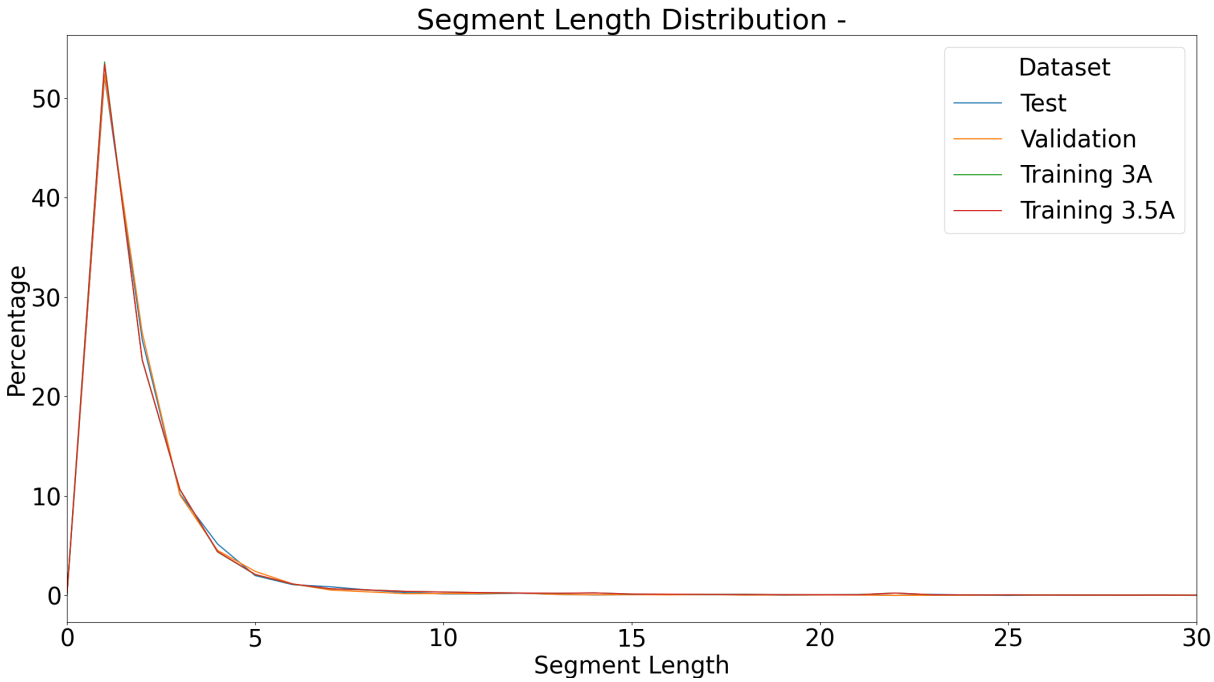

Figure 7.8: Segment length distribution for - residues in training, test and validation datasets

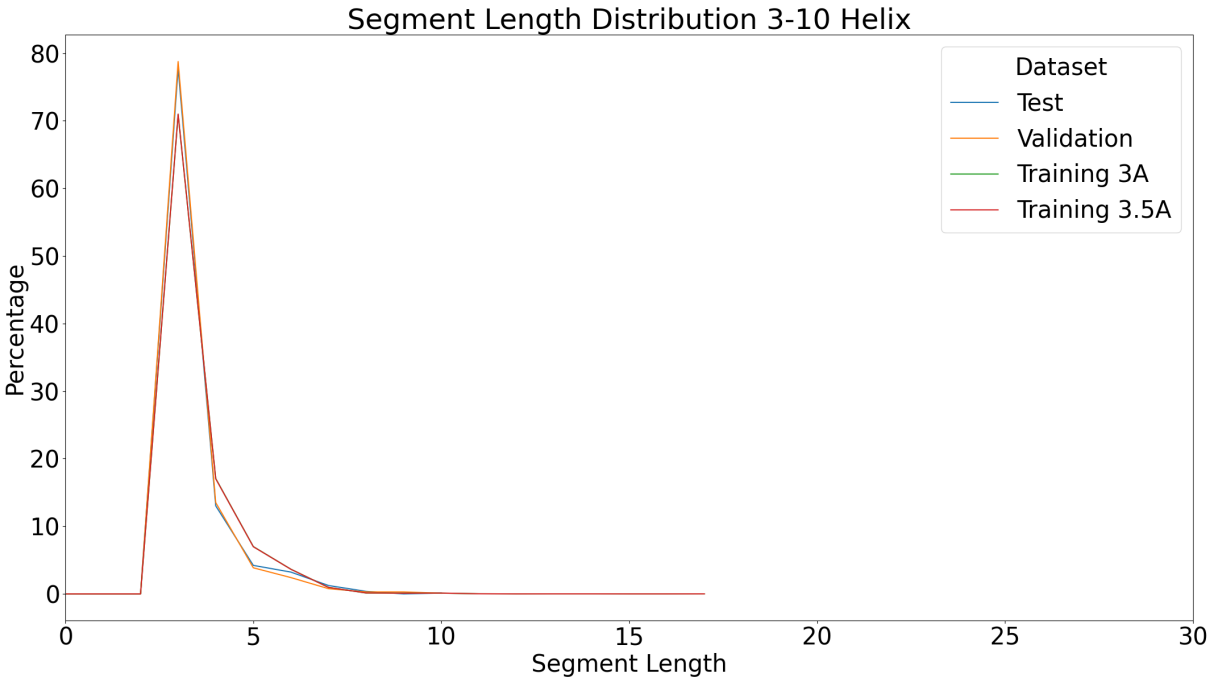

Figure 7.9: Segment length distribution for 3-10 helix residues in training, test and validation datasets

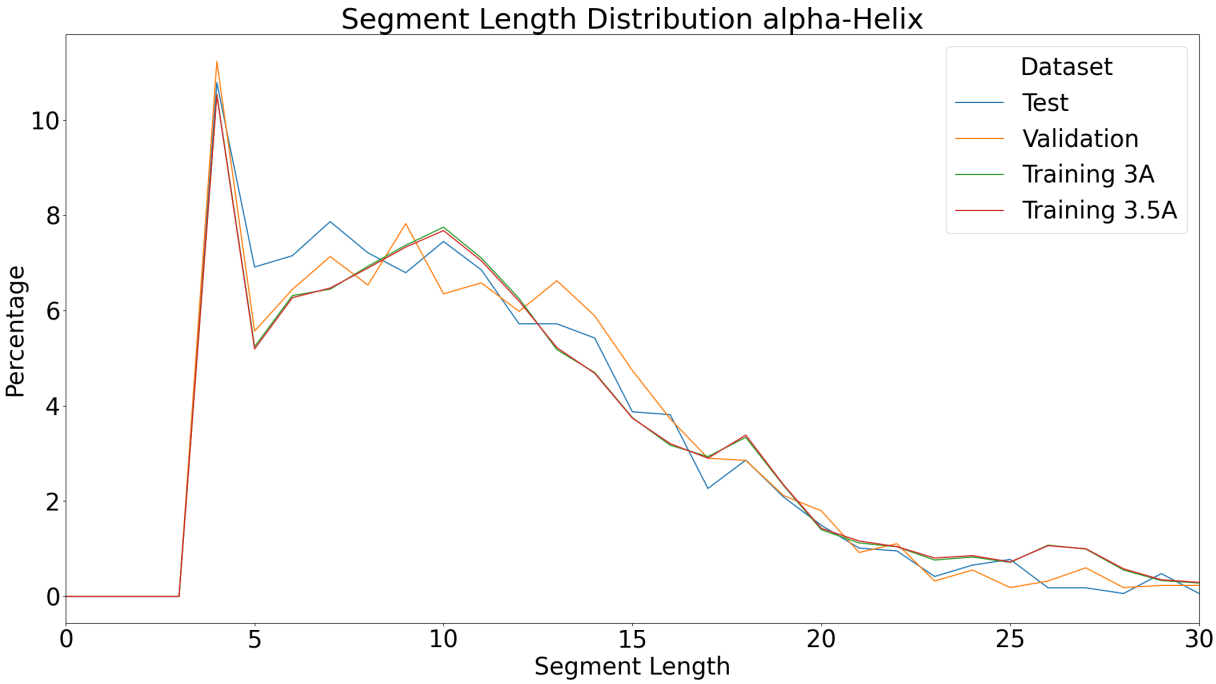

Figure 7.10: Segment length distribution for alpha helix residues in training, test and validation datasets

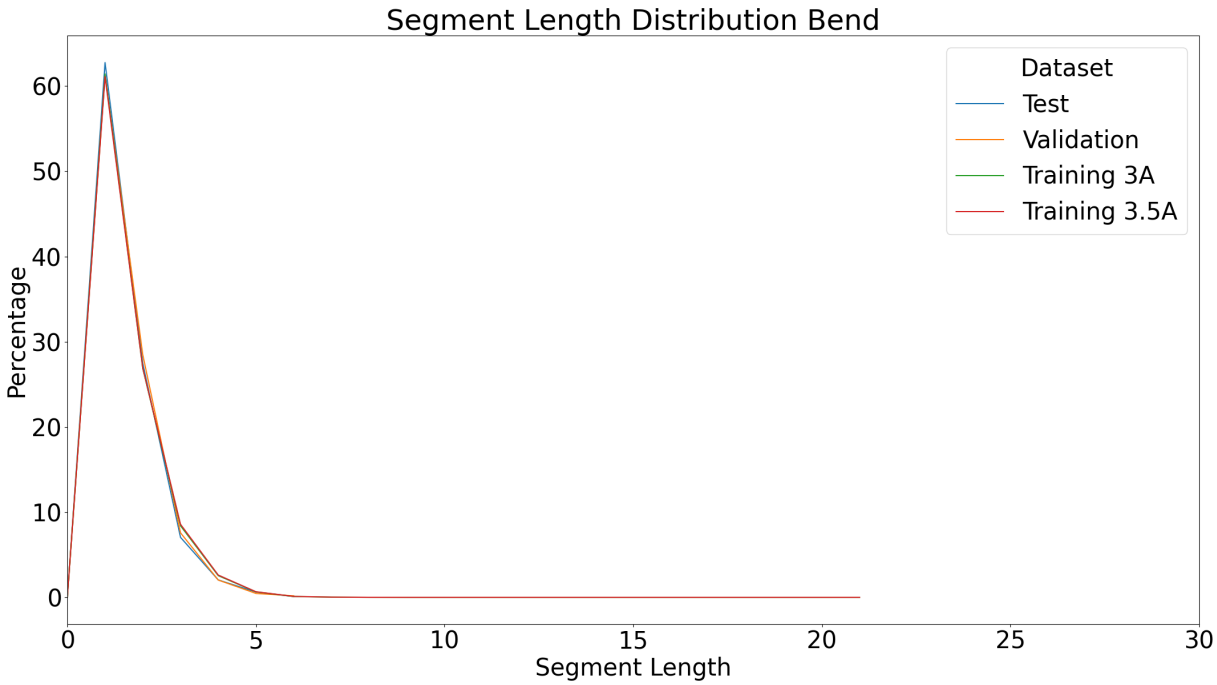

Figure 7.11: Segment length distribution for bend residues in training, test and validation datasets

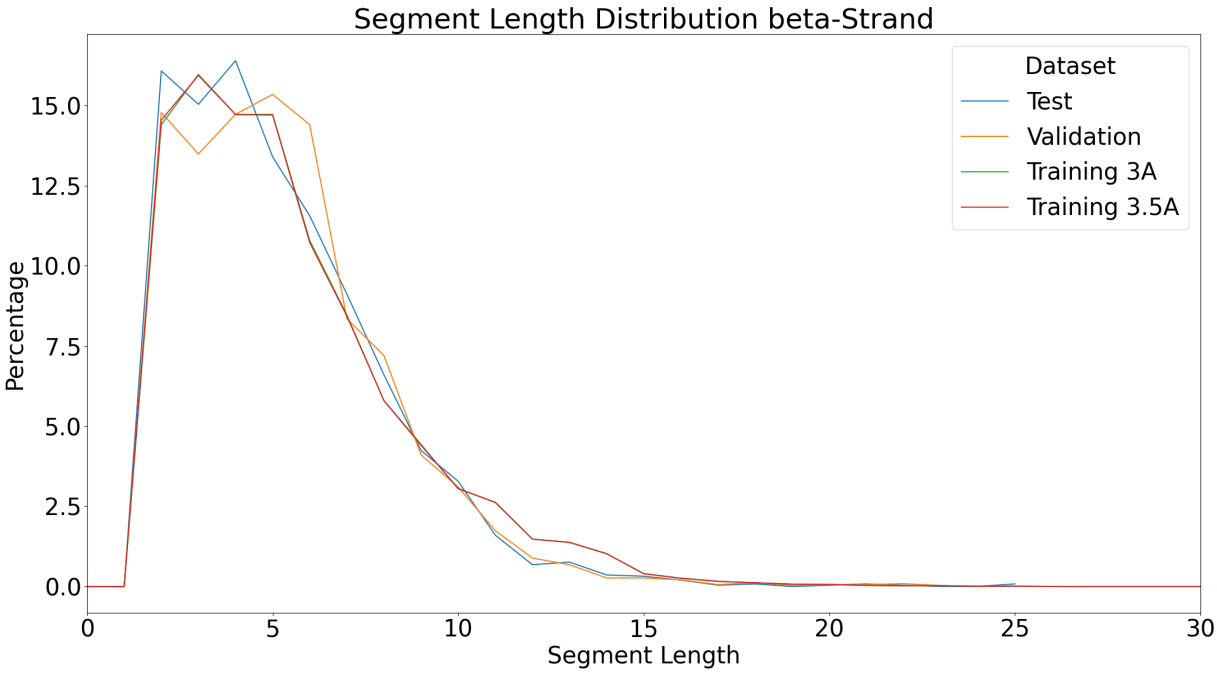

Figure 7.12: Segment length distribution for beta strand residues in training, test and validation datasets

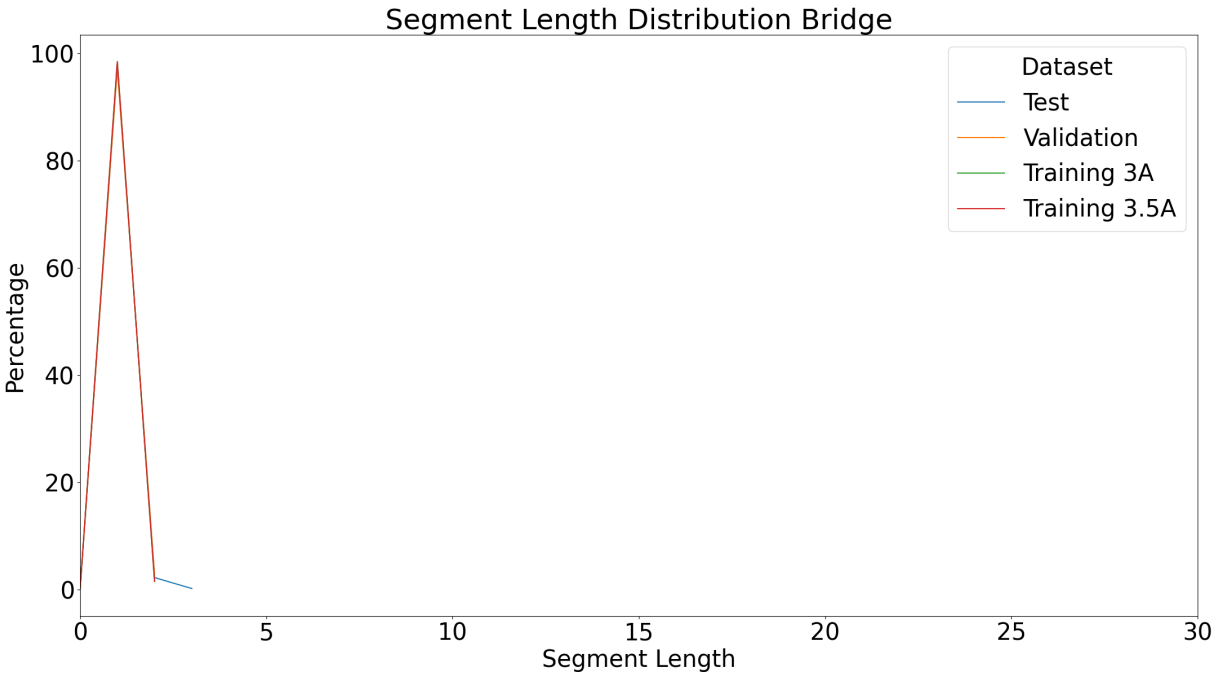

Figure 7.13: Segment length distribution for bridge residues in training, test and validation datasets

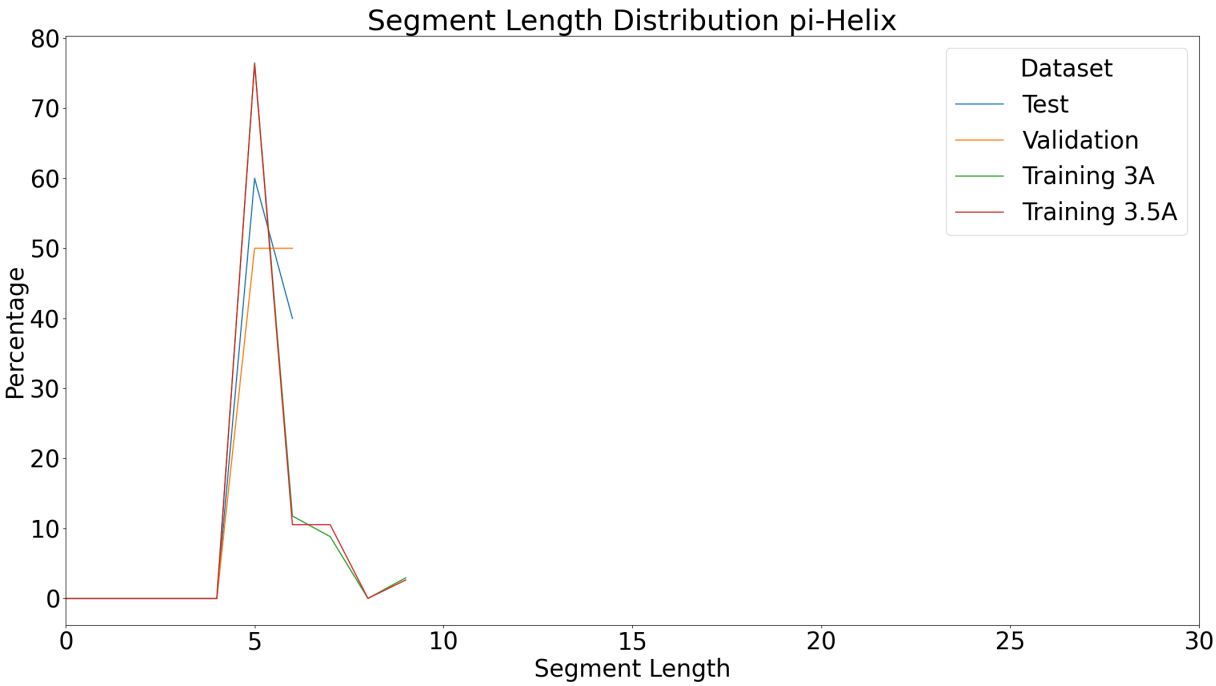

Figure 7.14: Segment length distribution for pi helix residues in training, test and validation datasets

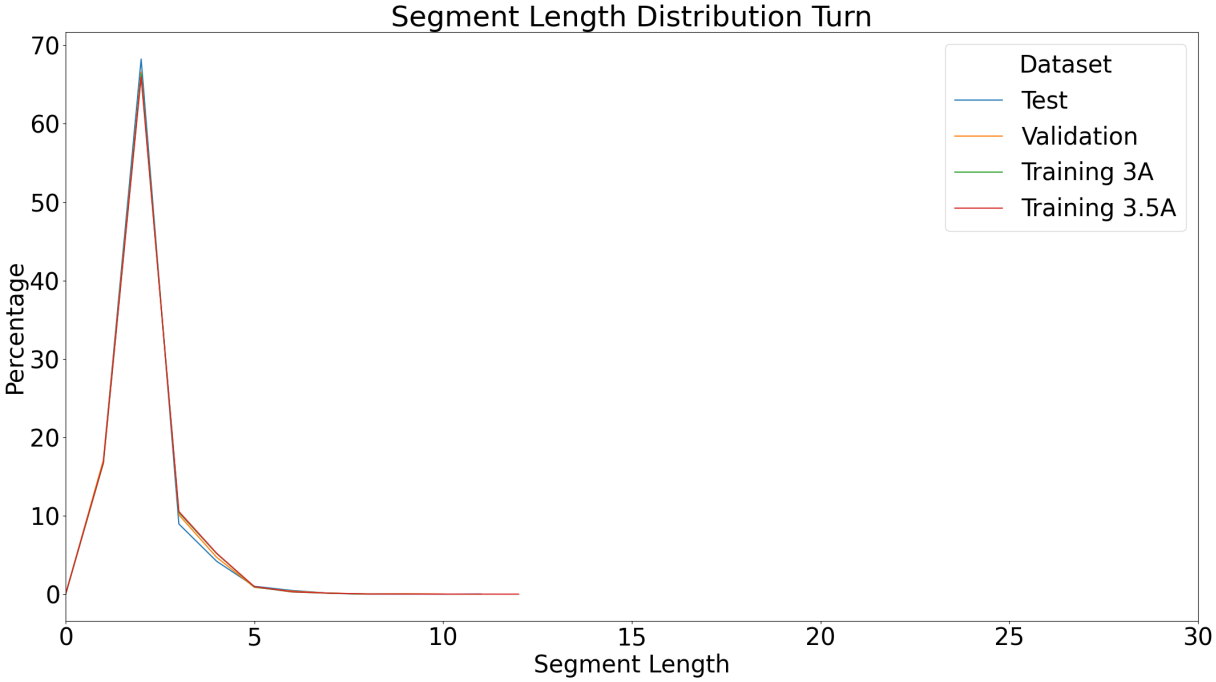

Figure 7.15: Segment length distribution for turn residues in training, test and validation datasets

7.7 Validation Set Performance Bar Plots

7.7.1 Single Sequence Embeddings

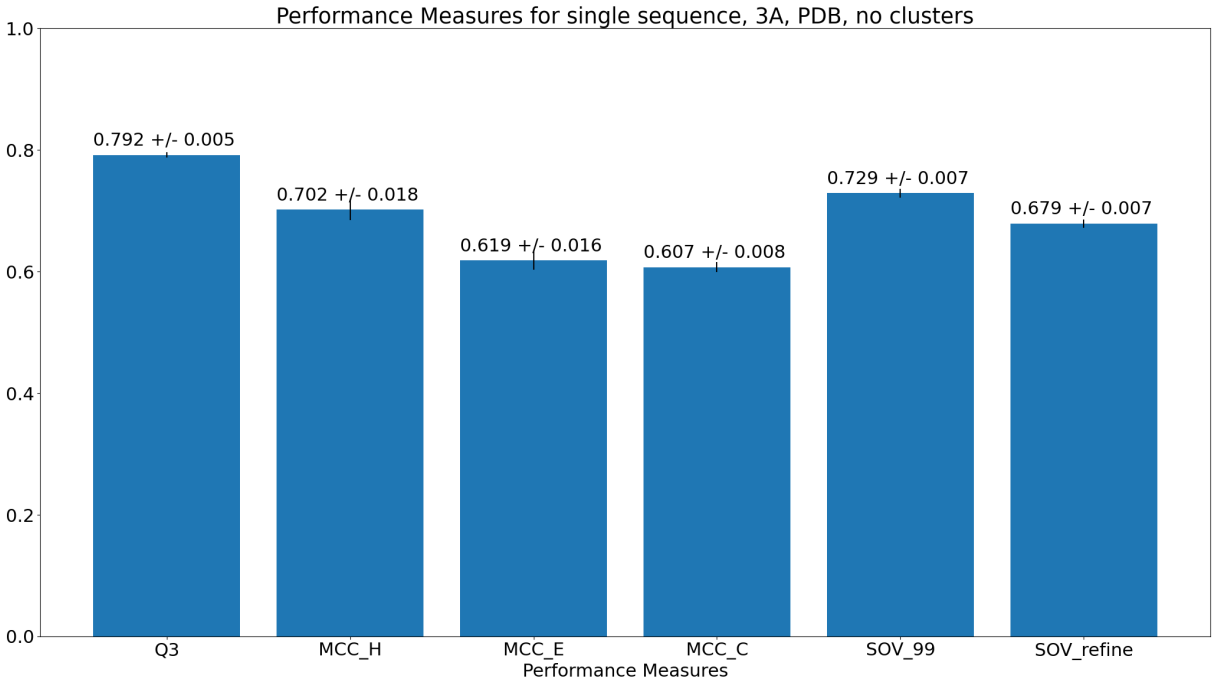

Figure 7.16: Q_3, MCC and SOV performance on the validation set for training on the 3Å training set using PDB structures as labels and single sequence embeddings as input

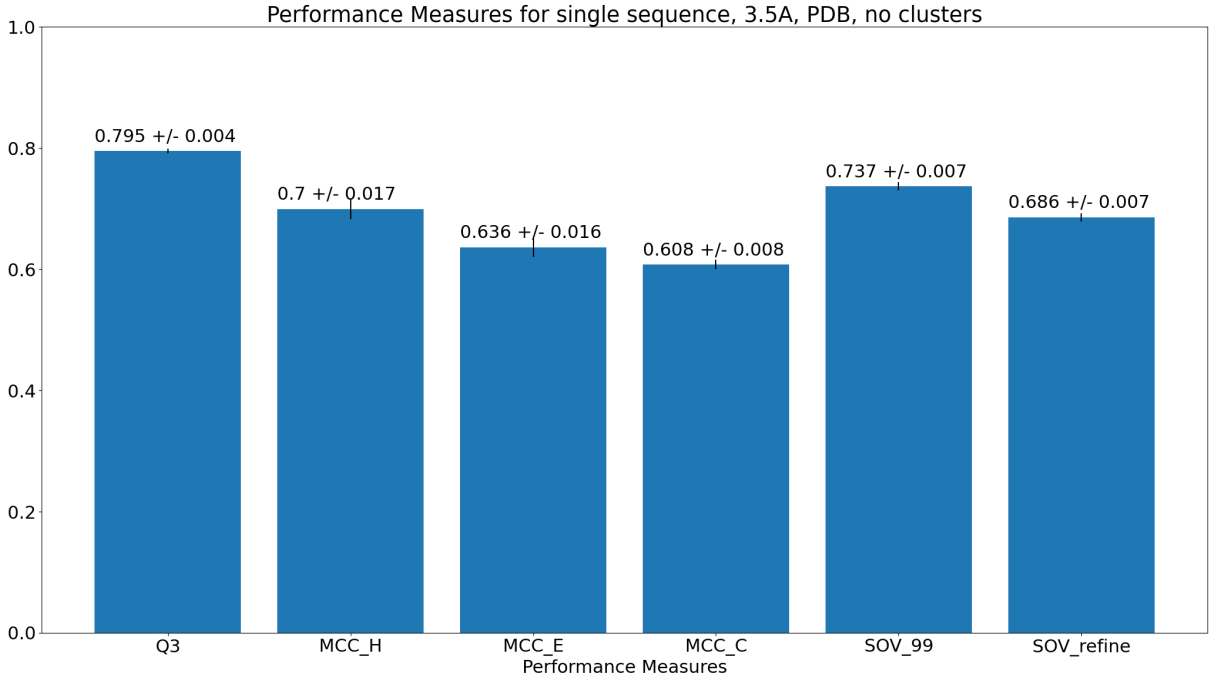

Figure 7.17: Q_3, MCC and SOV performance on the validation set for training on the 3.5Å training set using PDB structures as labels and single sequence embeddings as input

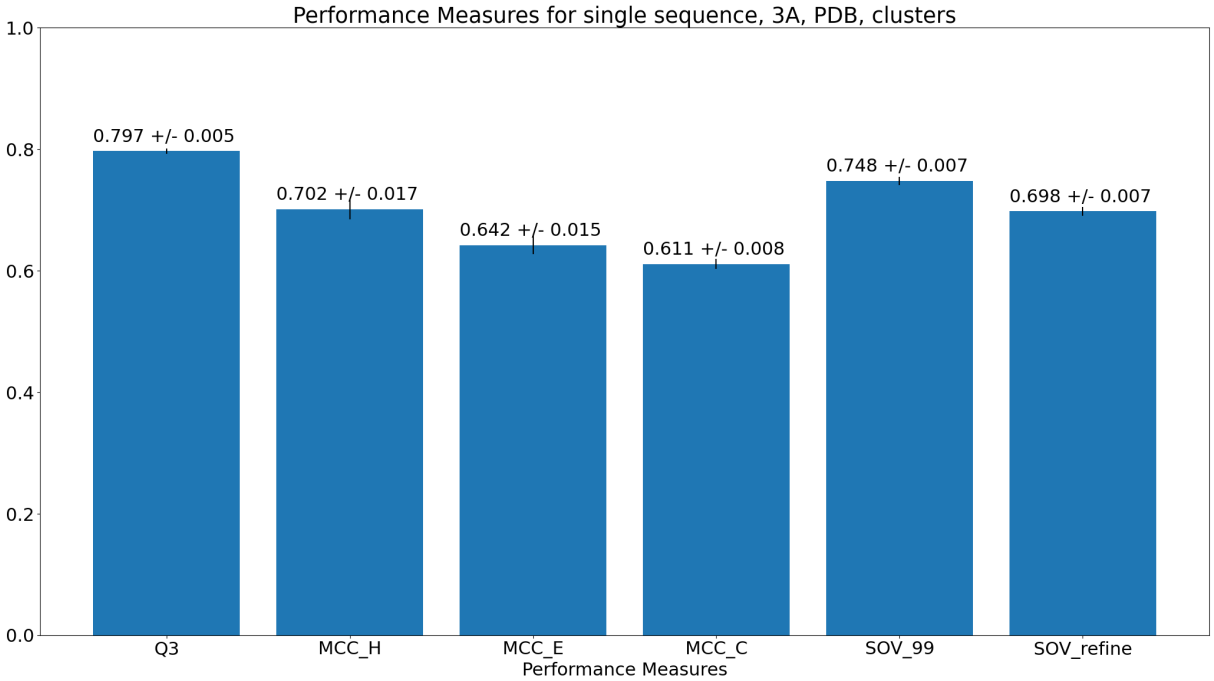

Figure 7.18: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3Å training set using PDB structures as labels and single sequence embeddings as input

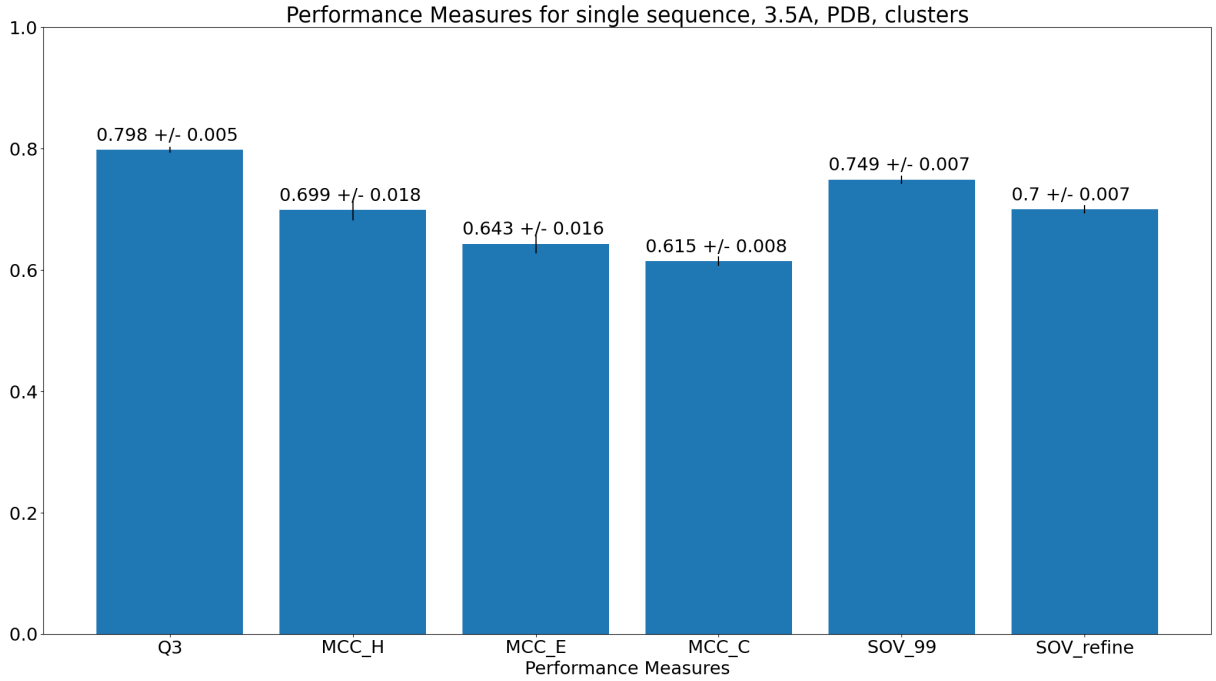

Figure 7.19: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3.5Å training set using PDB structures as labels and single sequence embeddings as input

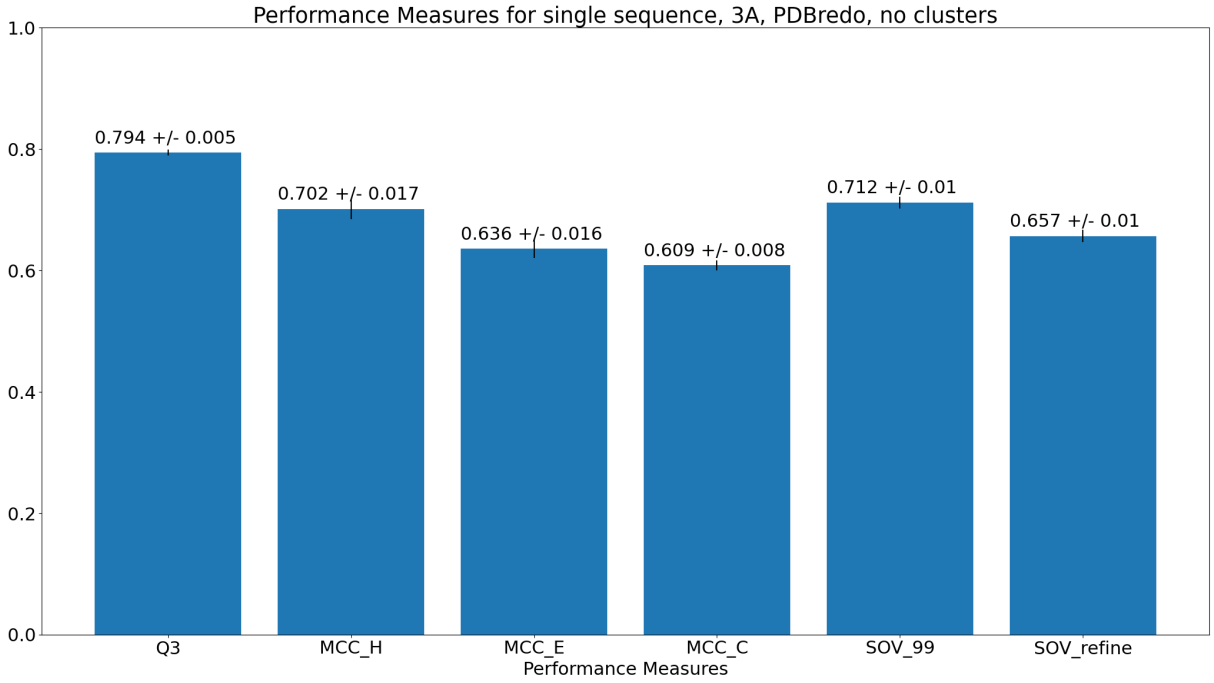

Figure 7.20: Q_3, MCC and SOV performance on the validation set for training on the 3Å training set using PDBredo structures as labels and single sequence embeddings as input

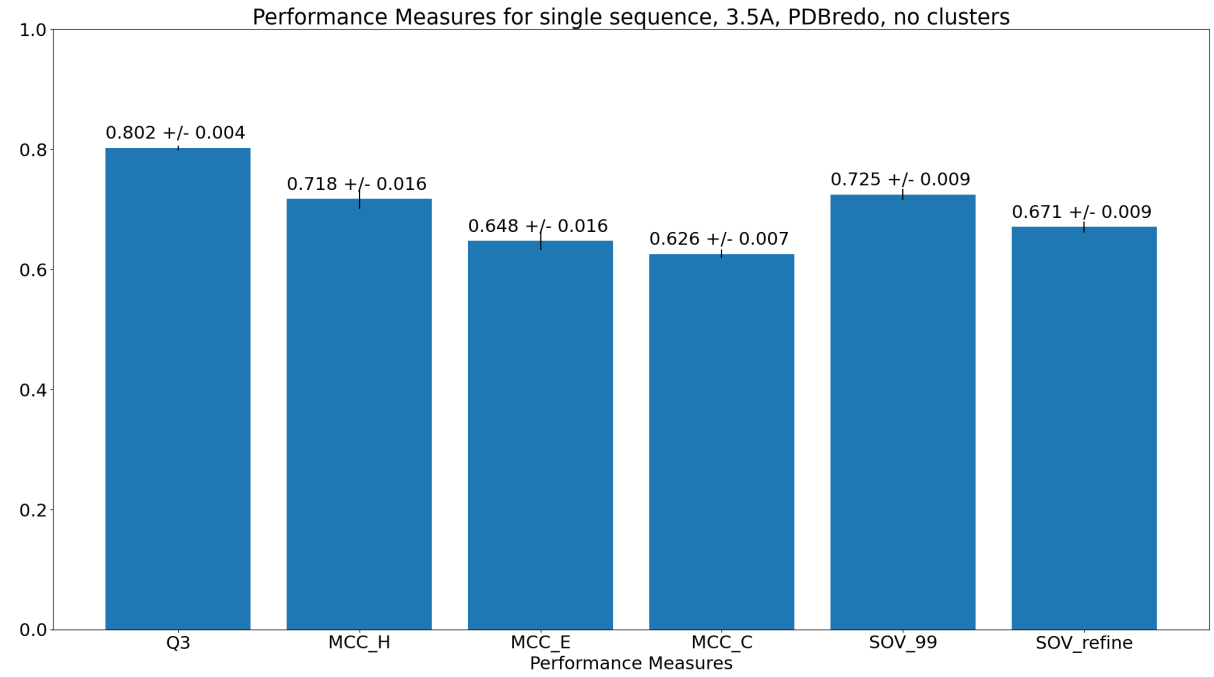

Figure 7.21: Q_3, MCC and SOV performance on the validation set for training on the 3.5Å training set using PDBredo structures as labels and single sequence embeddings as input

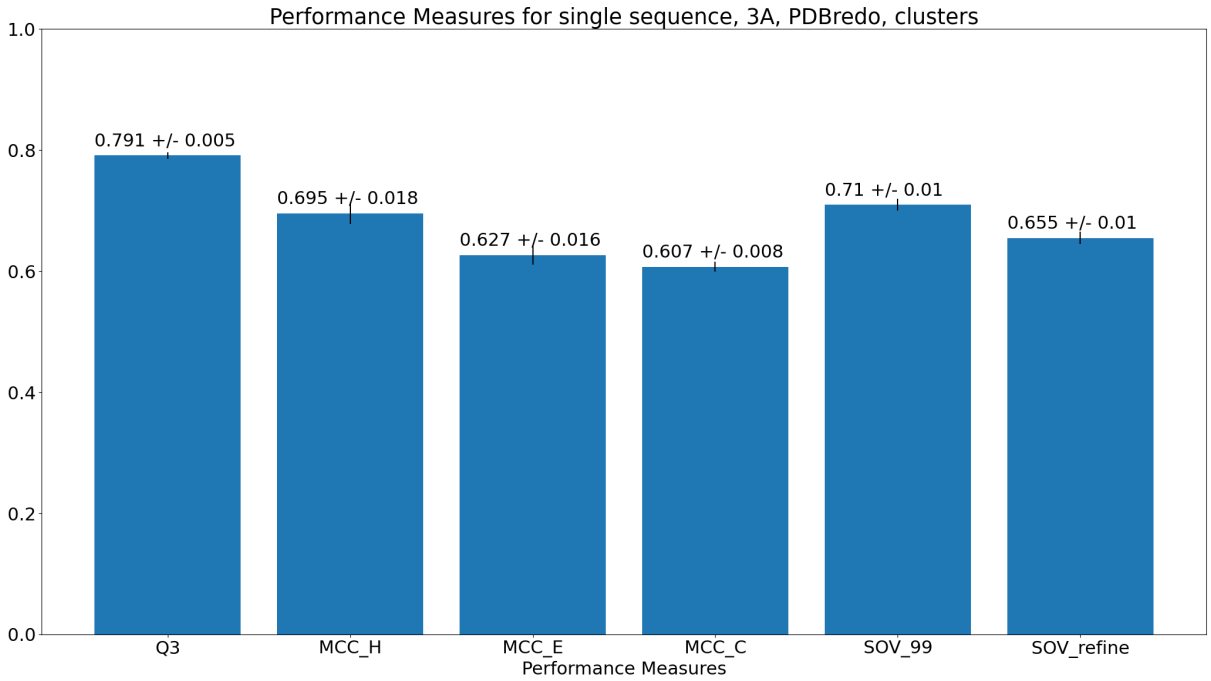

Figure 7.22: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3Å training set using PDBredo structures as labels and single sequence embeddings as input

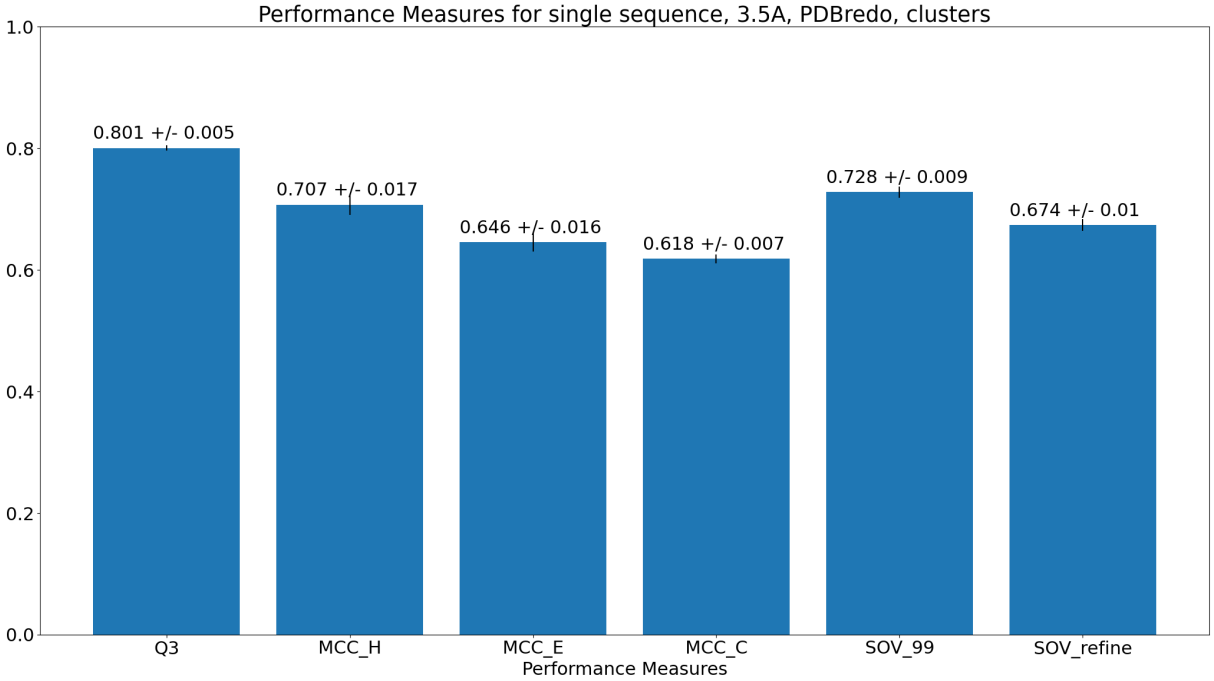

Figure 7.23: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3.5Å training set using PDBredo structures as labels and single sequence embeddings as input

7.7.2 Averaged MSA Embeddings

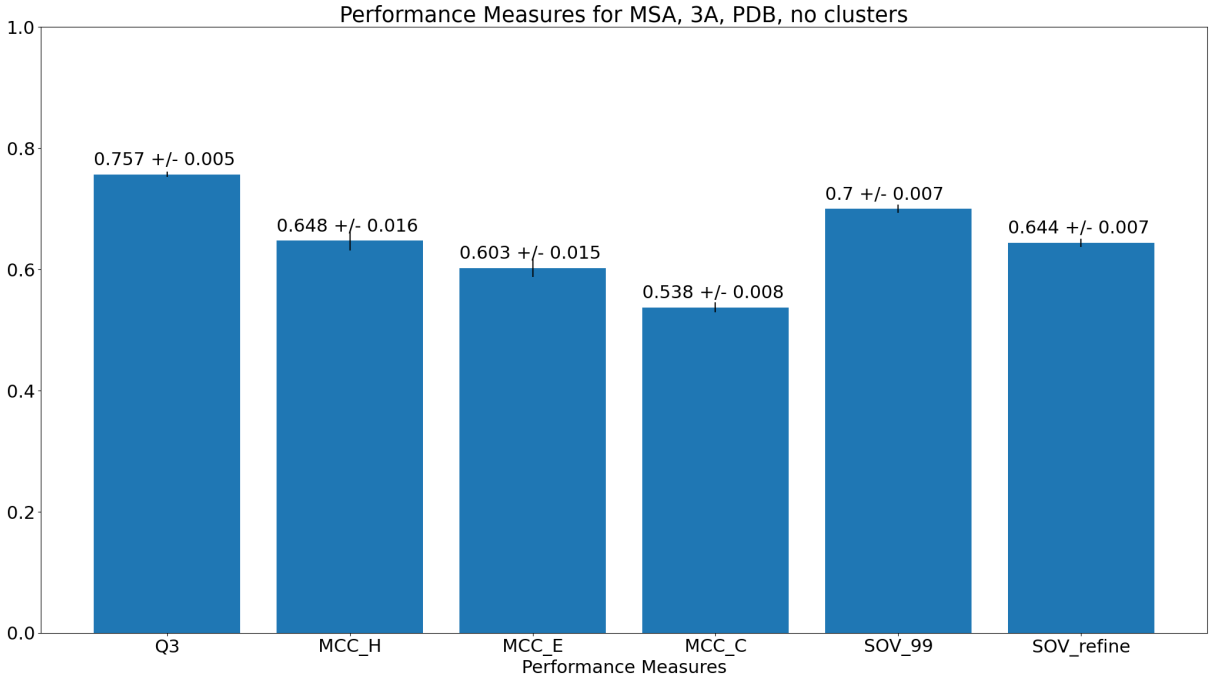

Figure 7.24: Q_3, MCC and SOV performance on the validation set for training on the 3Å training set using PDB structures as labels and averaged MSA embeddings as input

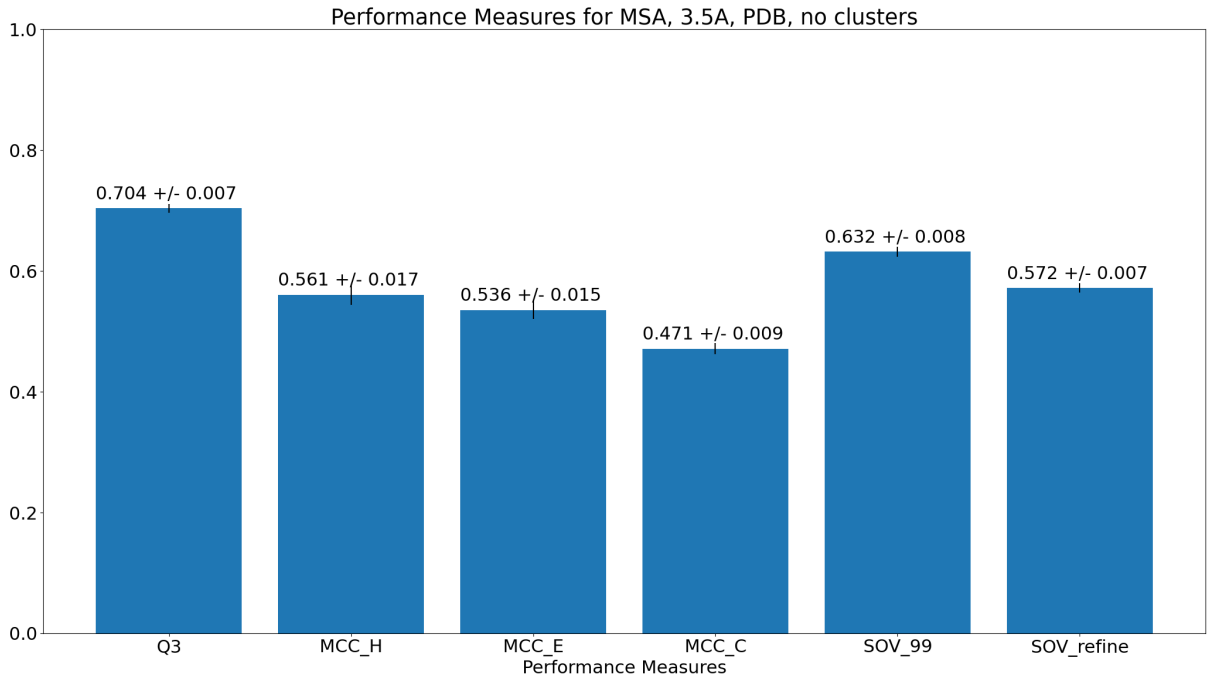

Figure 7.25: Q_3, MCC and SOV performance on the validation set for training on the 3.5Å training set using PDB structures as labels and averaged MSA embeddings as input

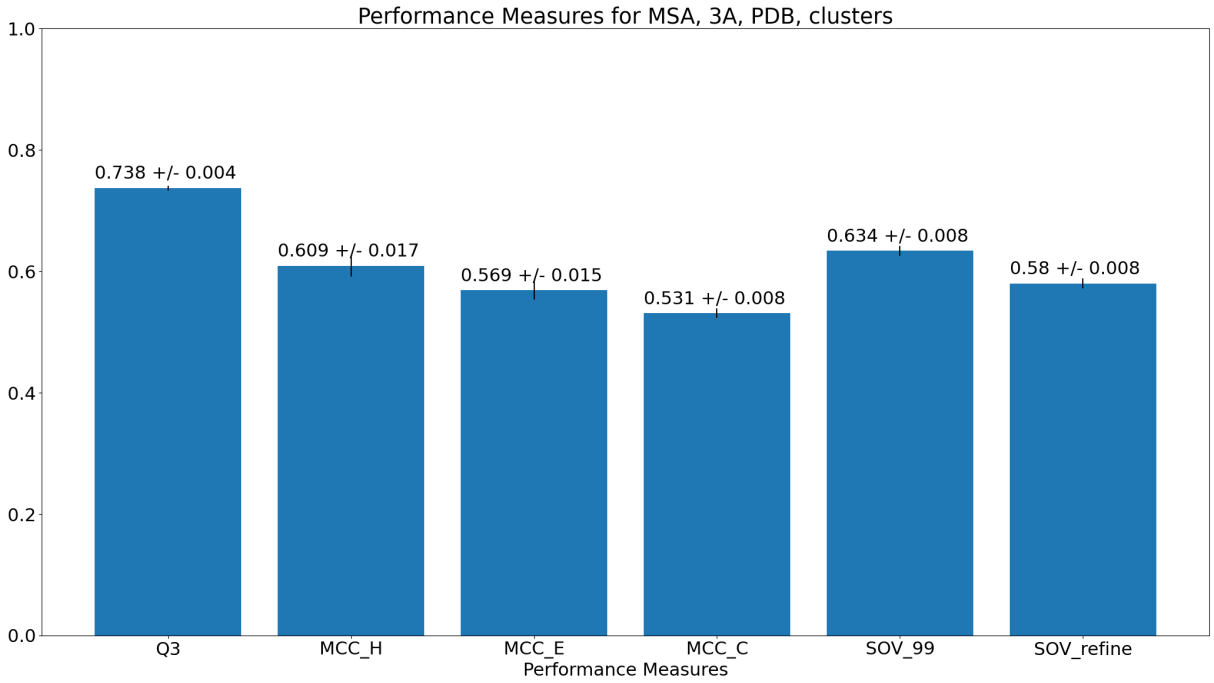

Figure 7.26: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3Å training set using PDB structures as labels and averaged MSA embeddings as input

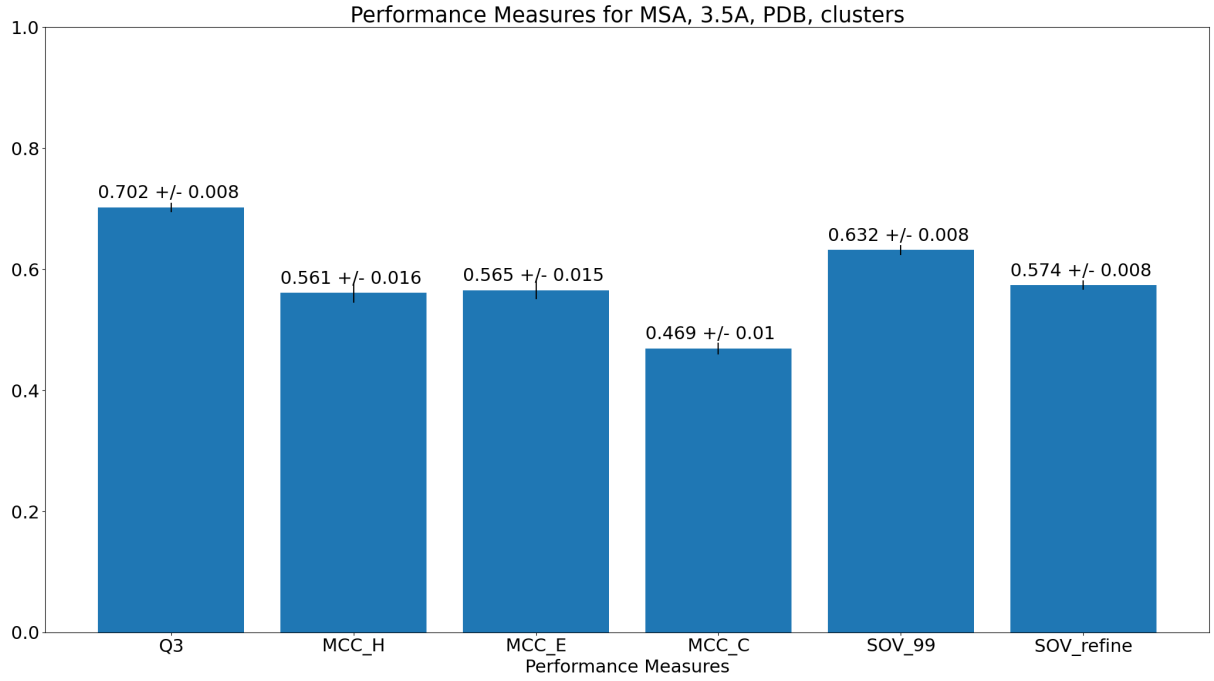

Figure 7.27: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3.5Å training set using PDB structures as labels and averaged MSA embeddings as input

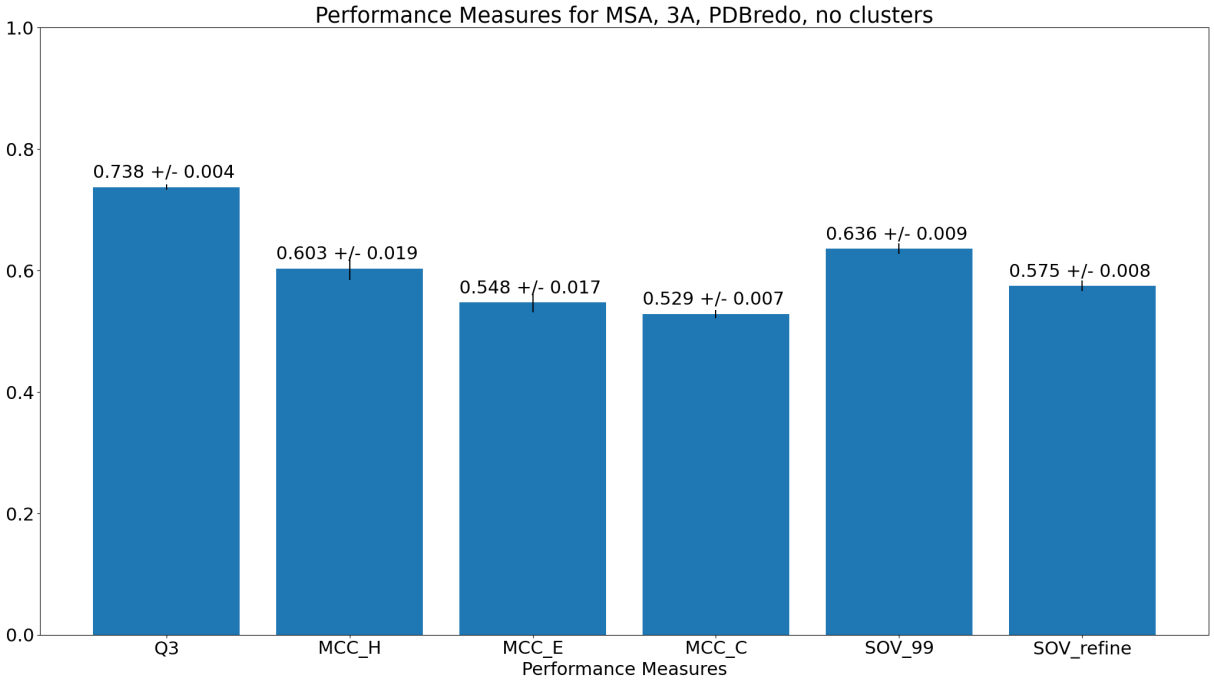

Figure 7.28: Q_3, MCC and SOV performance on the validation set for training on the 3Å training set using PDBredo structures as labels and averaged MSA embeddings as input

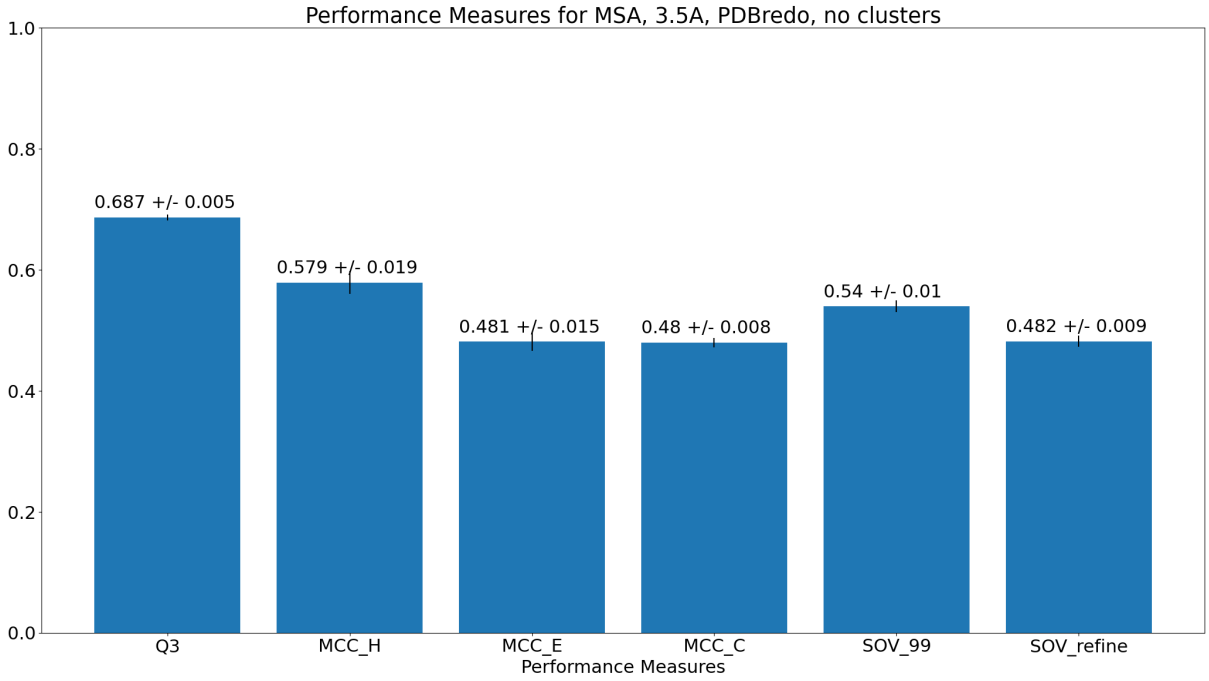

Figure 7.29: Q_3, MCC and SOV performance on the validation set for training on the 3.5Å training set using PDBredo structures as labels and averaged MSA embeddings as input

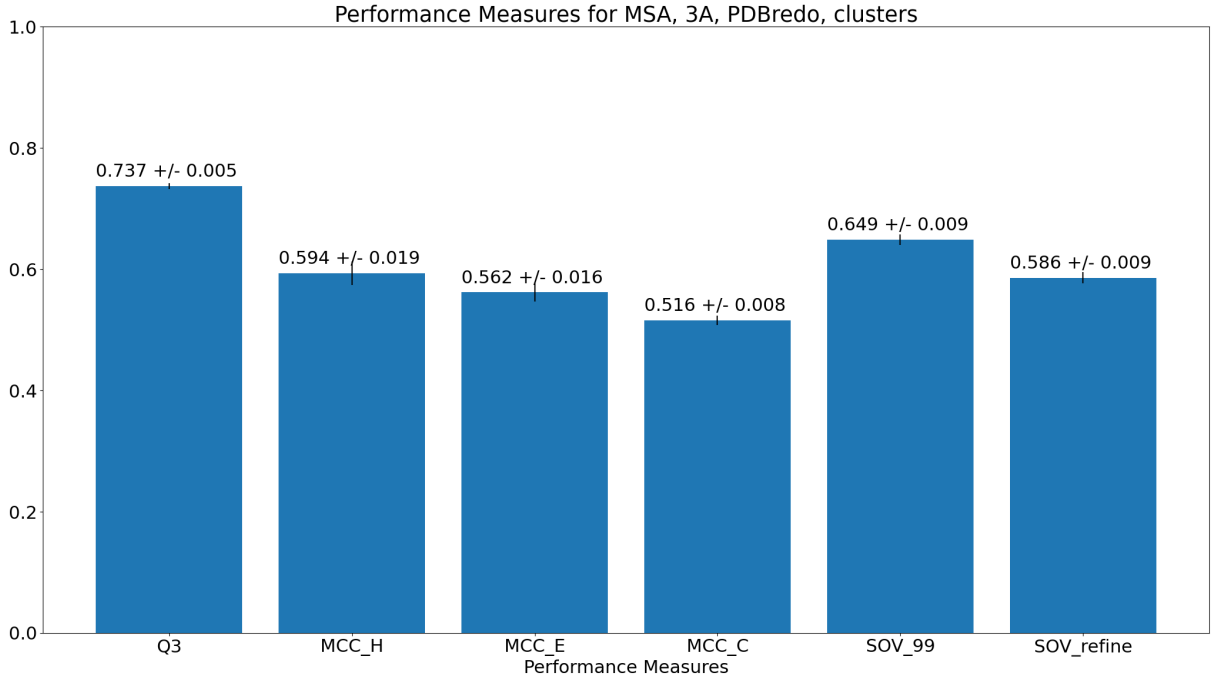

Figure 7.30: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3Å training set using PDBredo structures as labels and averaged MSA embeddings as input

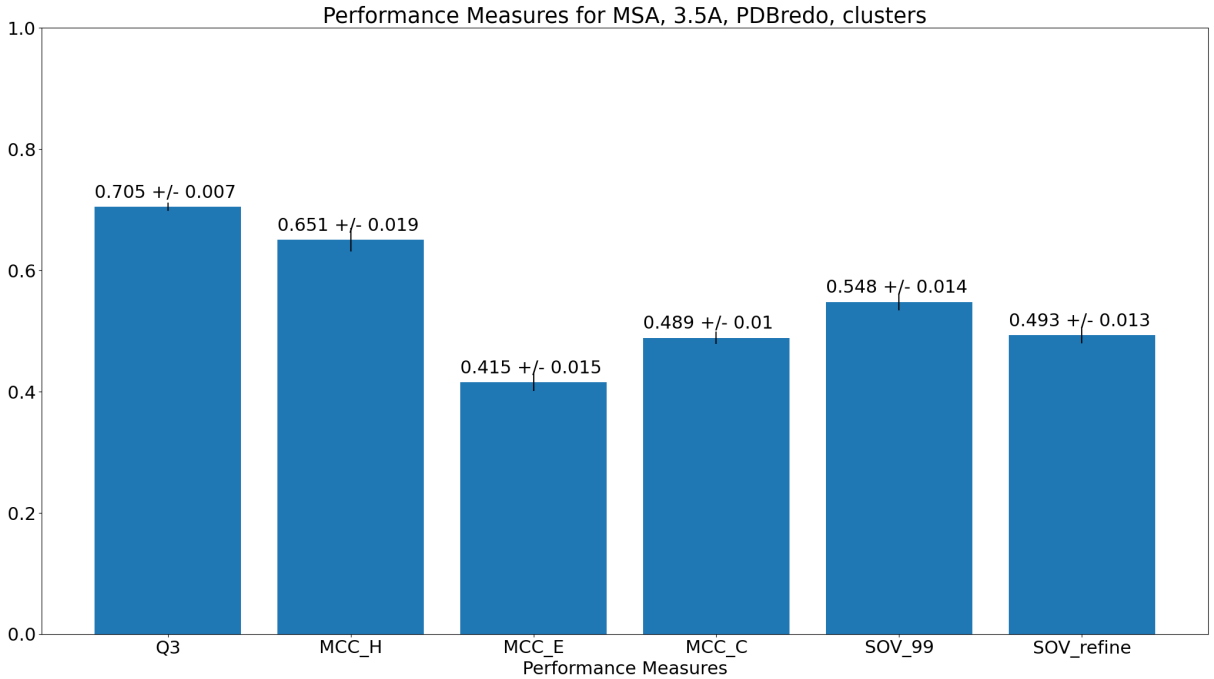

Figure 7.31: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3.5Å training set using PDBredo structures as labels and averaged MSA embeddings as input

7.7.3 Weighted MSA Embeddings

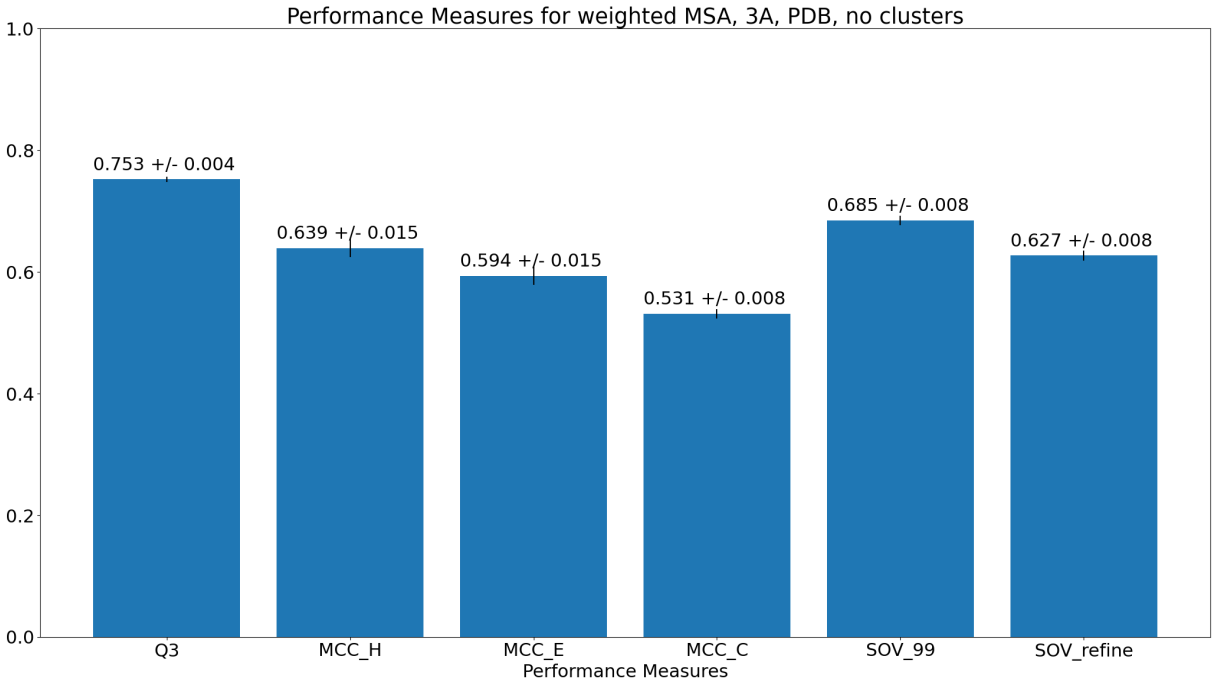

Figure 7.32: Q_3, MCC and SOV performance on the validation set for training on the 3Å training set using PDB structures as labels and weighted averaged MSA embeddings as input

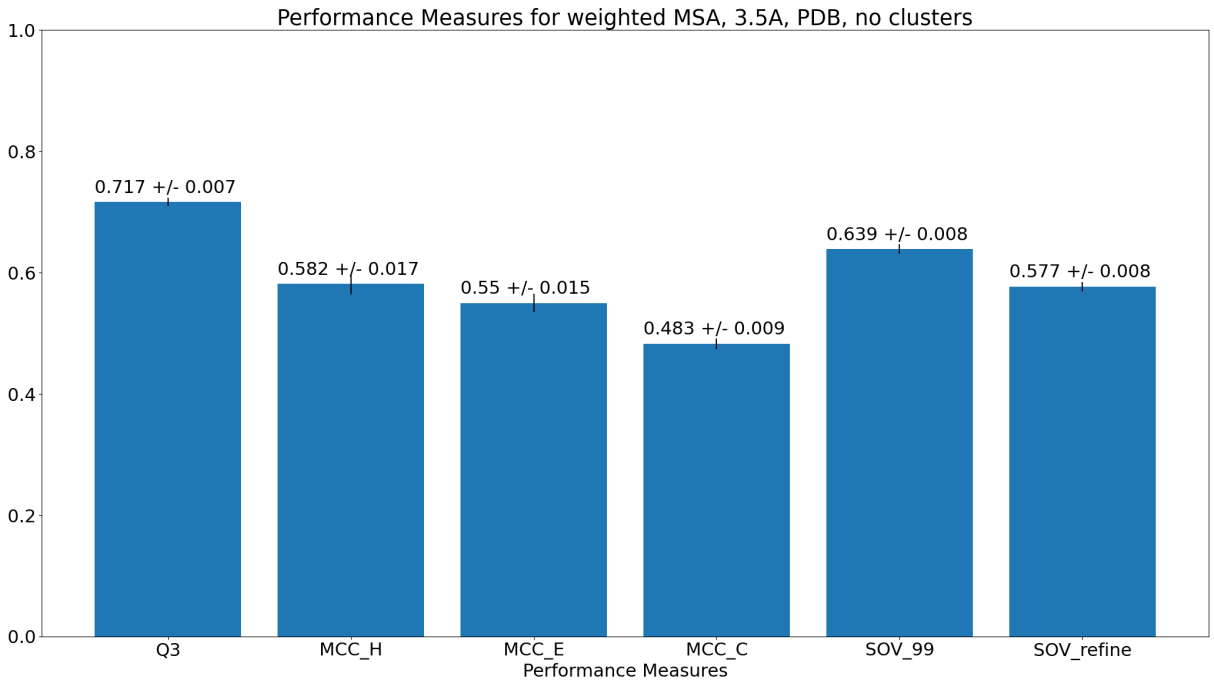

Figure 7.33: Q_3, MCC and SOV performance on the validation set for training on the 3.5Å training set using PDB structures as labels and weighted averaged MSA embeddings as input

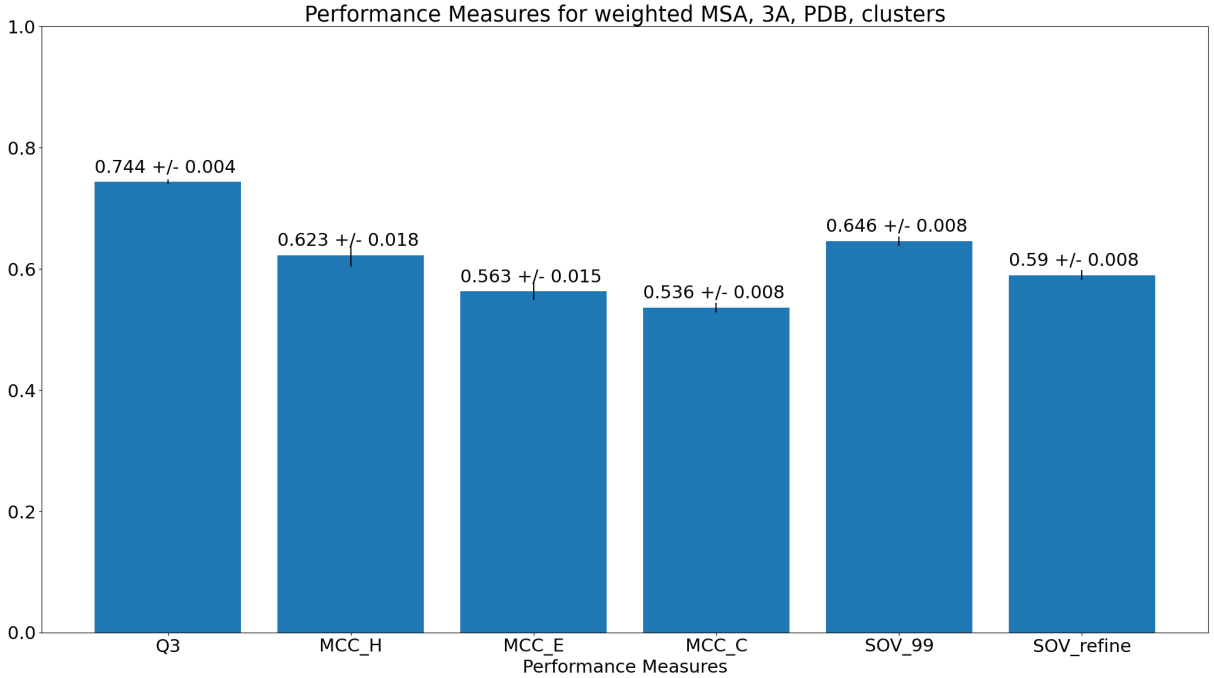

Figure 7.34: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3Å training set using PDB structures as labels and weighted averaged MSA embeddings as input

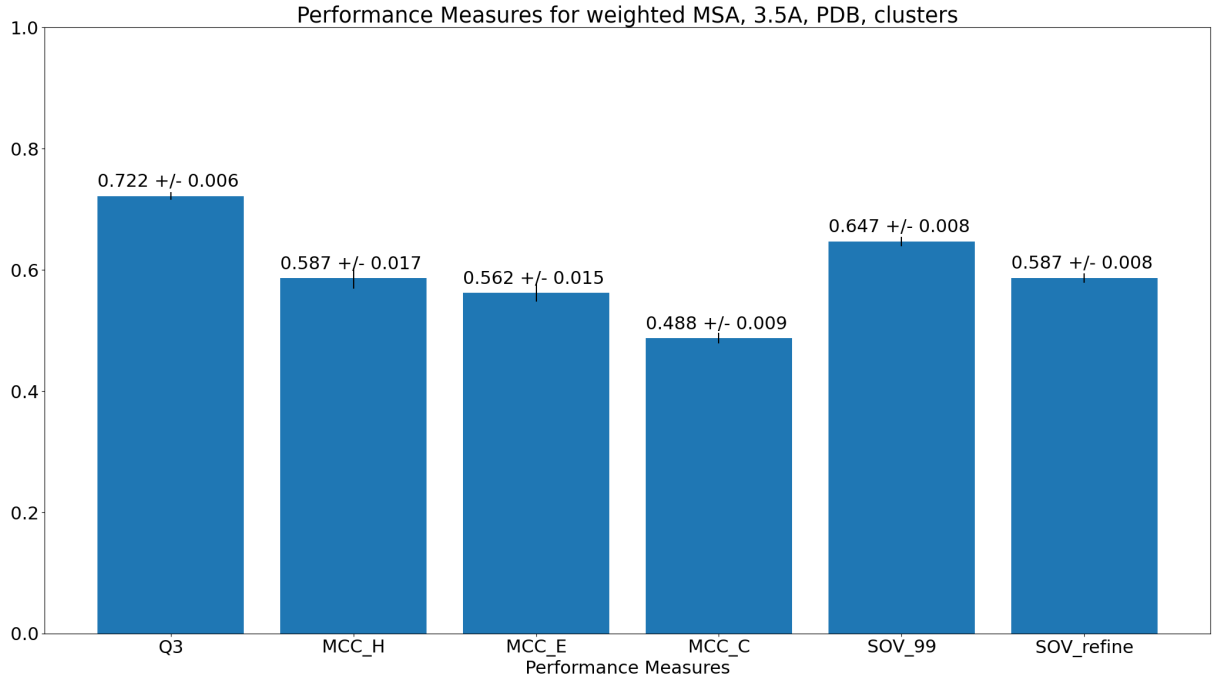

Figure 7.35: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3.5Å training set using PDB structures as labels and weighted averaged MSA embeddings as input

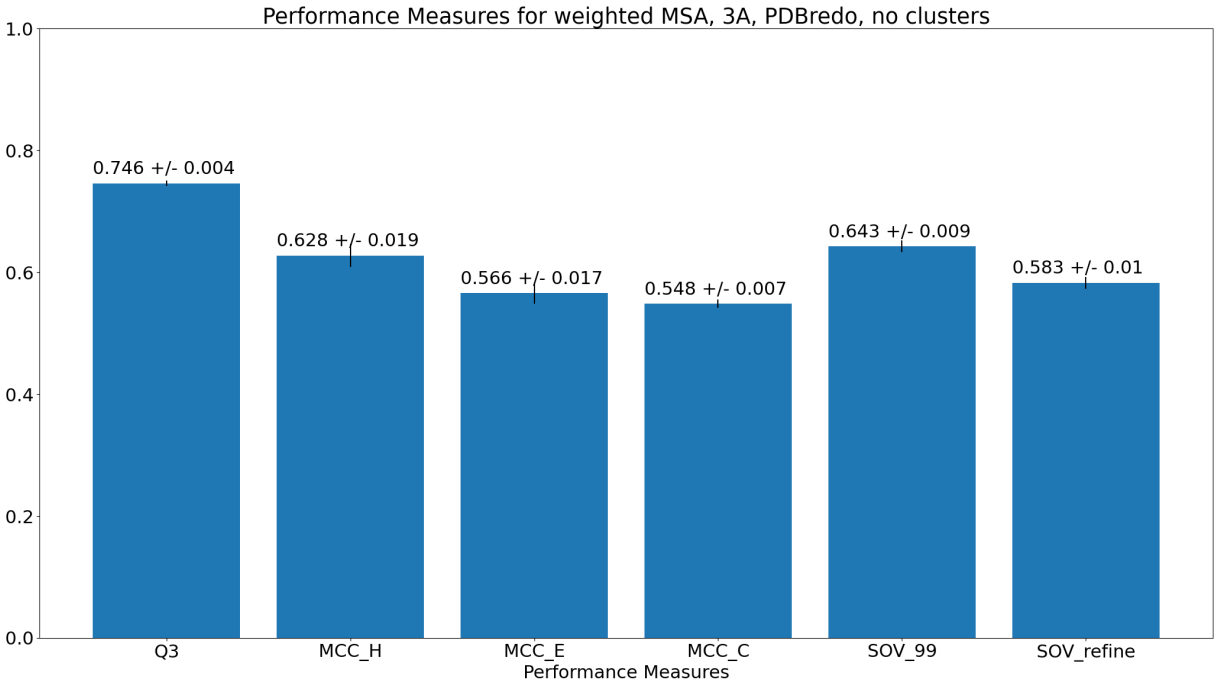

Figure 7.36: Q_3, MCC and SOV performance on the validation set for training on the 3Å training set using PDBredo structures as labels and weighted averaged MSA embeddings as input

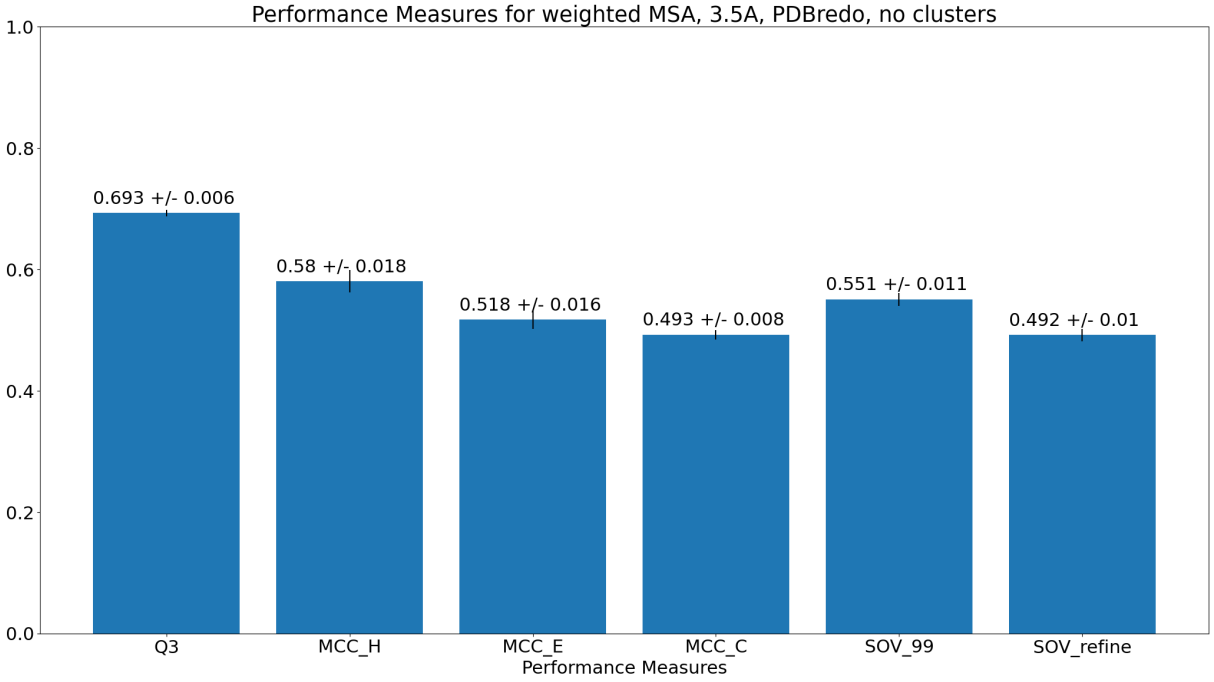

Figure 7.37: Q_3, MCC and SOV performance on the validation set for training on the 3.5Å training set using PDBredo structures as labels and weighted averaged MSA embeddings as input

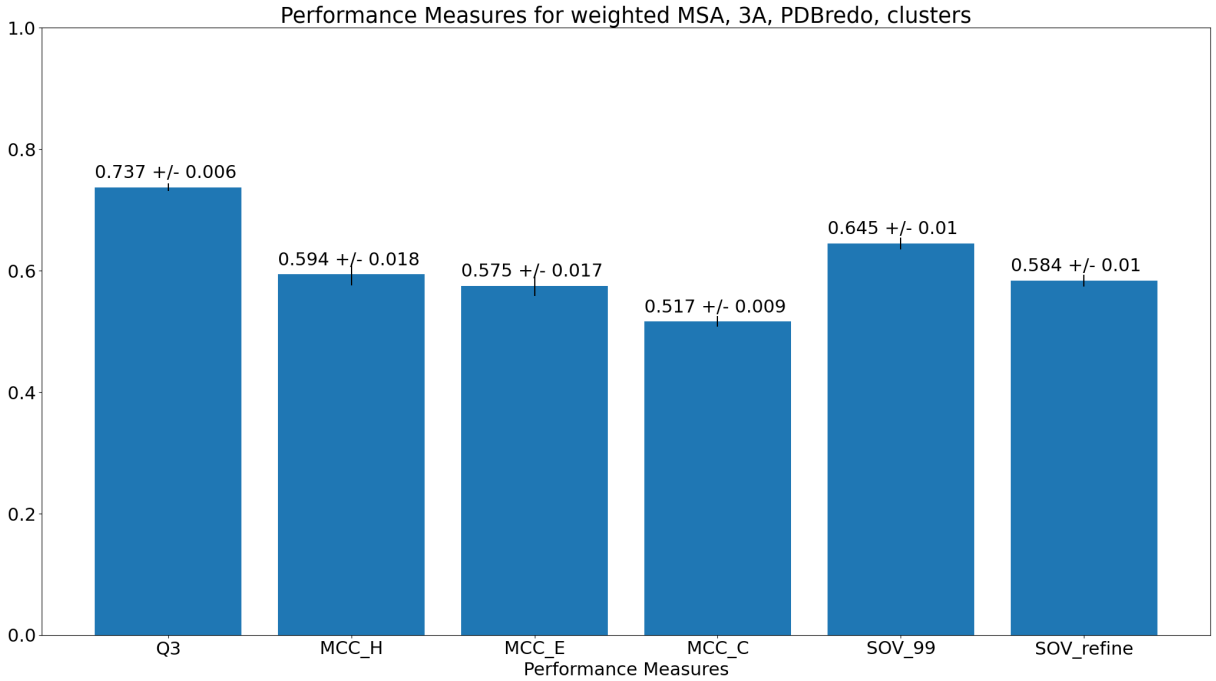

Figure 7.38: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3Å training set using PDBredo structures as labels and weighted averaged MSA embeddings as input

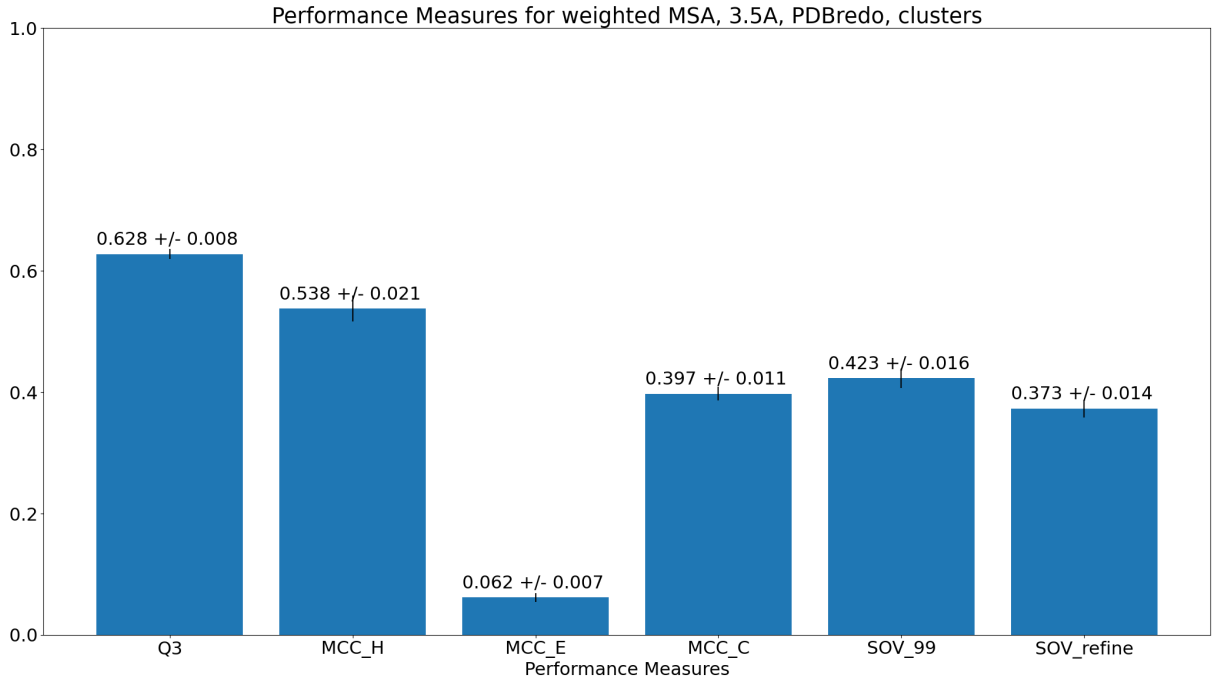

Figure 7.39: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3.5Å training set using PDBredo structures as labels and weighted averaged MSA embeddings as input

7.7.4 Inverse Weighted MSA Embeddings

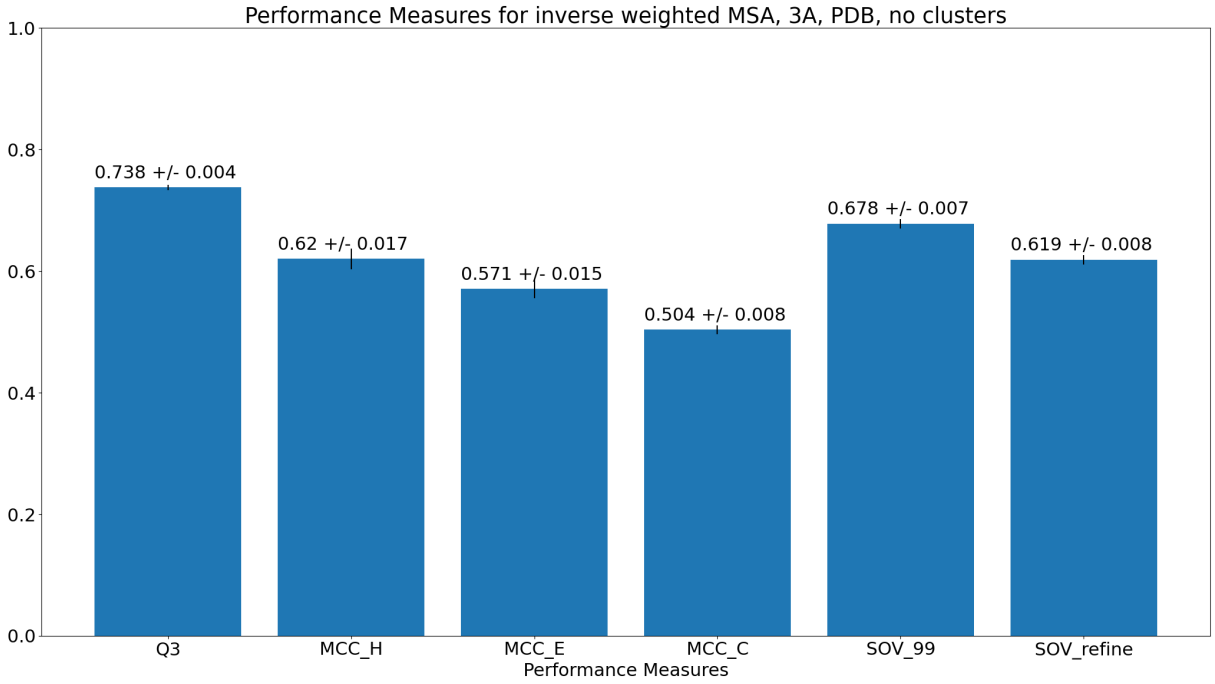

Figure 7.40: Q_3, MCC and SOV performance on the validation set for training on the 3Å training set using PDB structures as labels and inverse weighted averaged MSA embeddings as input

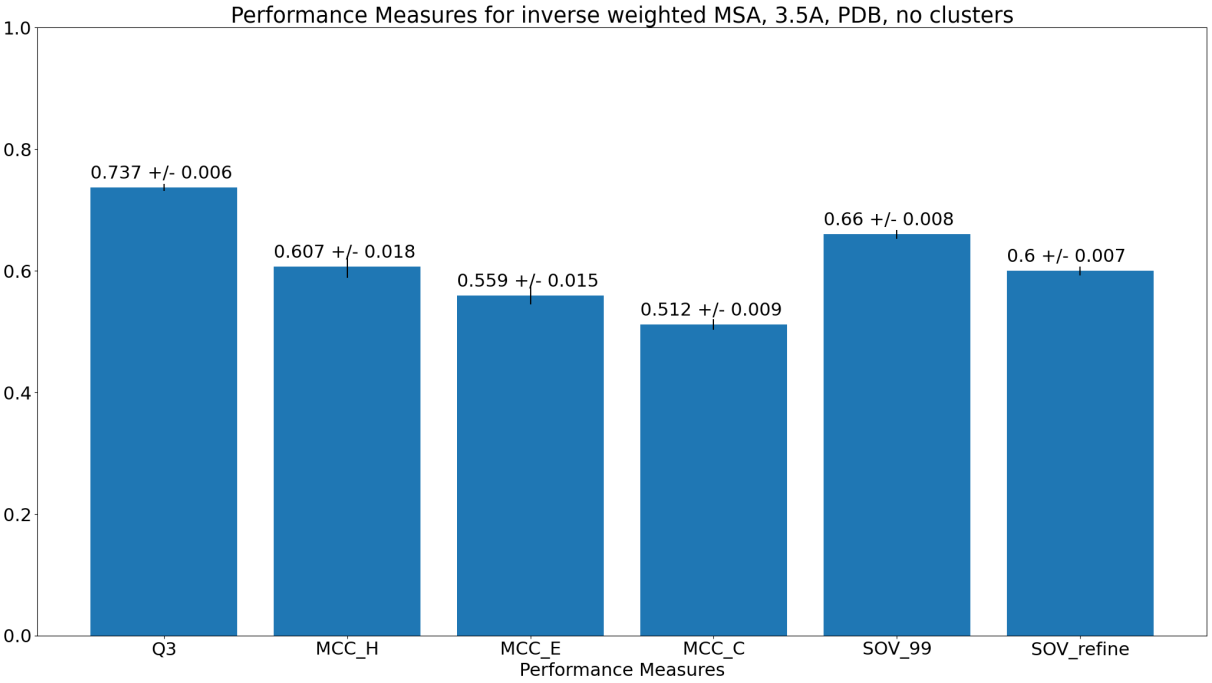

Figure 7.41: Q_3, MCC and SOV performance on the validation set for training on the 3.5Å training set using PDB structures as labels and inverse weighted averaged MSA embeddings as input

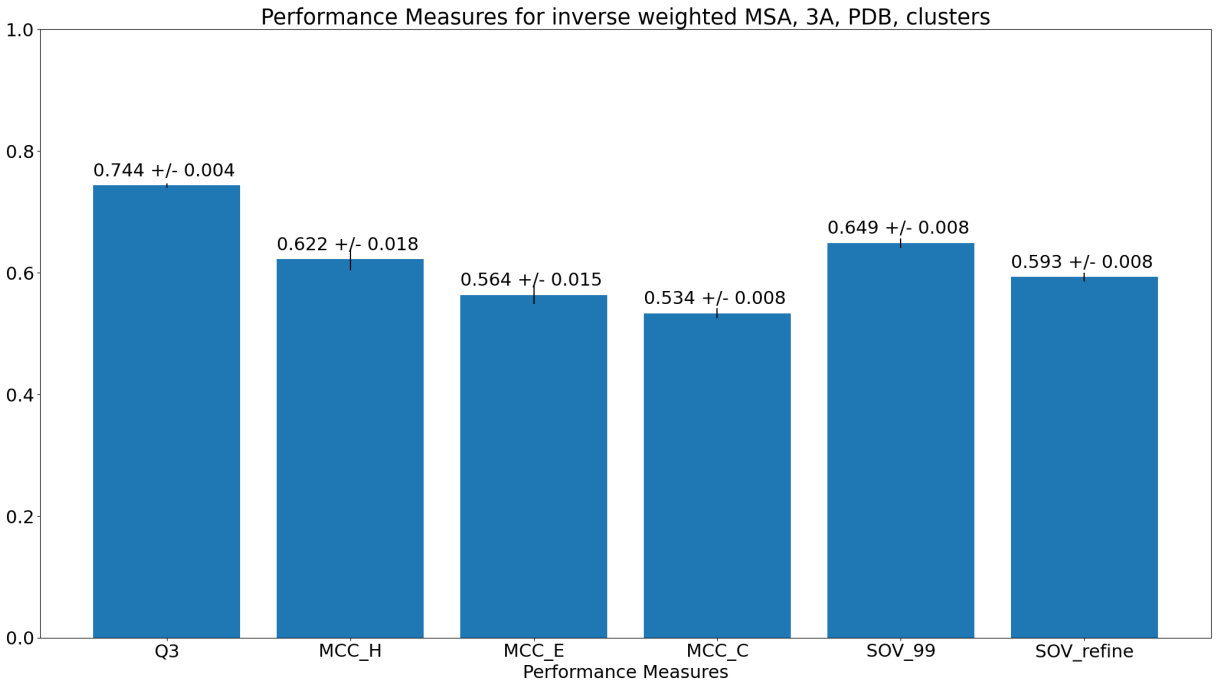

Figure 7.42: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3Å training set using PDB structures as labels and inverse weighted averaged MSA embeddings as input

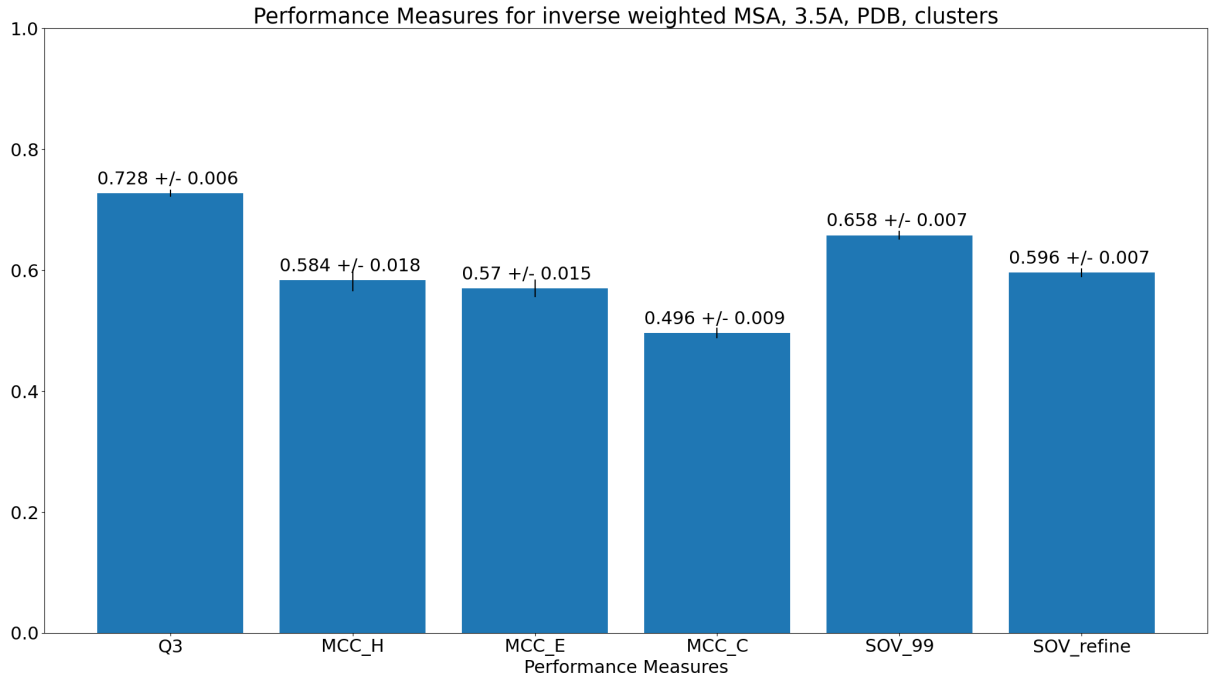

Figure 7.43: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3.5Å training set using PDB structures as labels and inverse weighted averaged MSA embeddings as input

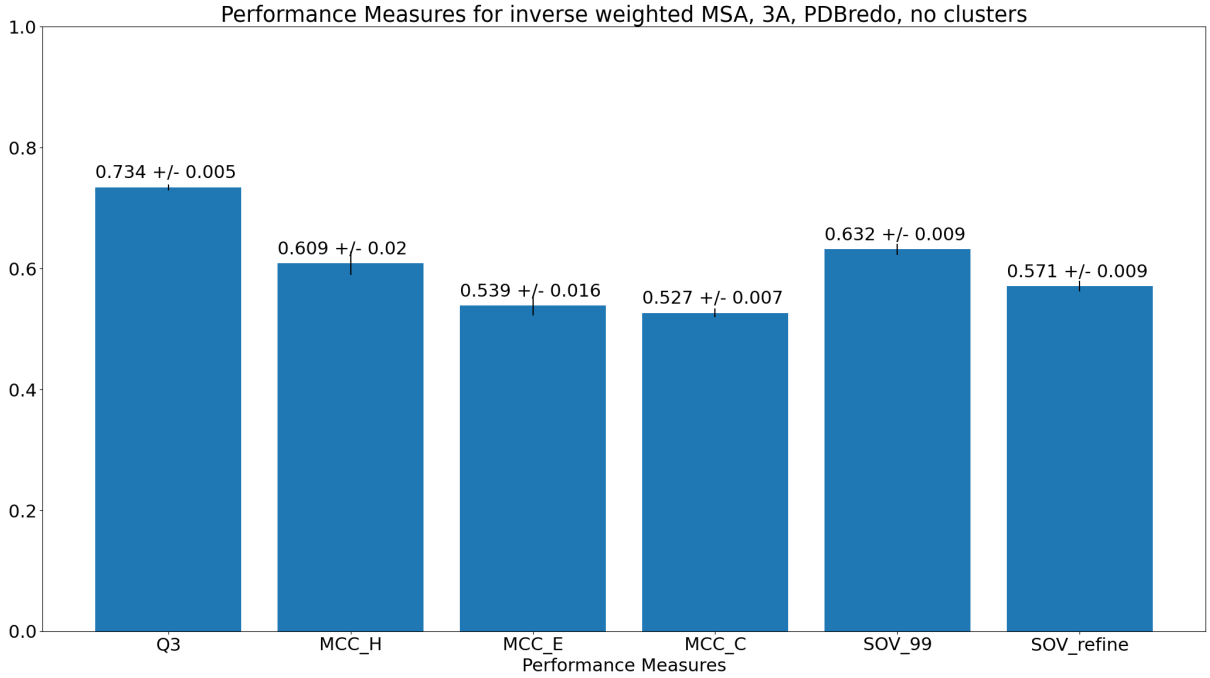

Figure 7.44: Q_3, MCC and SOV performance on the validation set for training on the 3Å training set using PDBredo structures as labels and inverse weighted averaged MSA embeddings as input

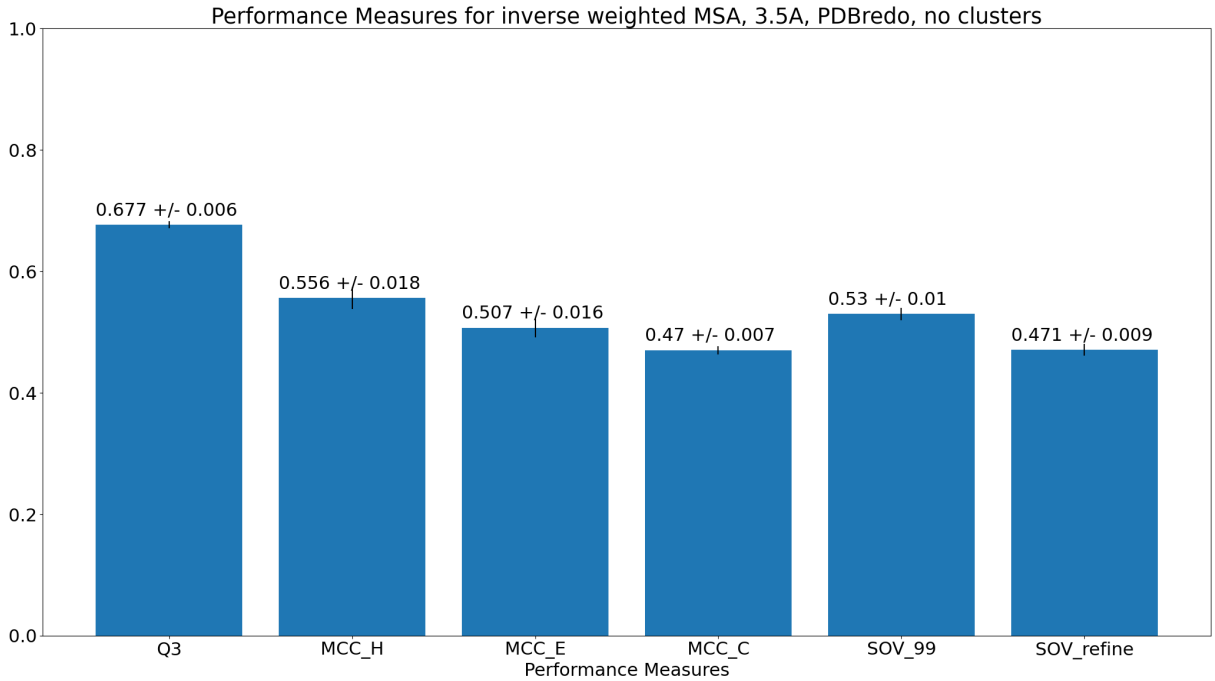

Figure 7.45: Q_3, MCC and SOV performance on the validation set for training on the 3.5Å training set using PDBredo structures as labels and inverse weighted averaged MSA embeddings as input

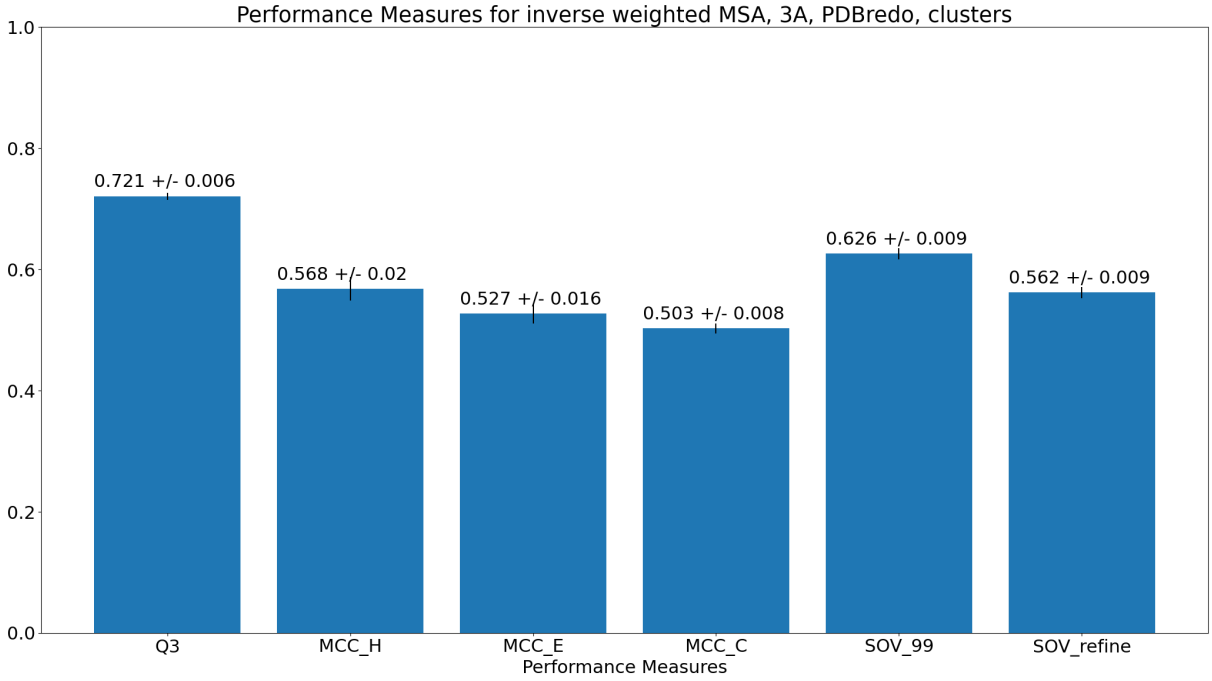

Figure 7.46: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3Å training set using PDBredo structures as labels and inverse weighted averaged MSA embeddings as input

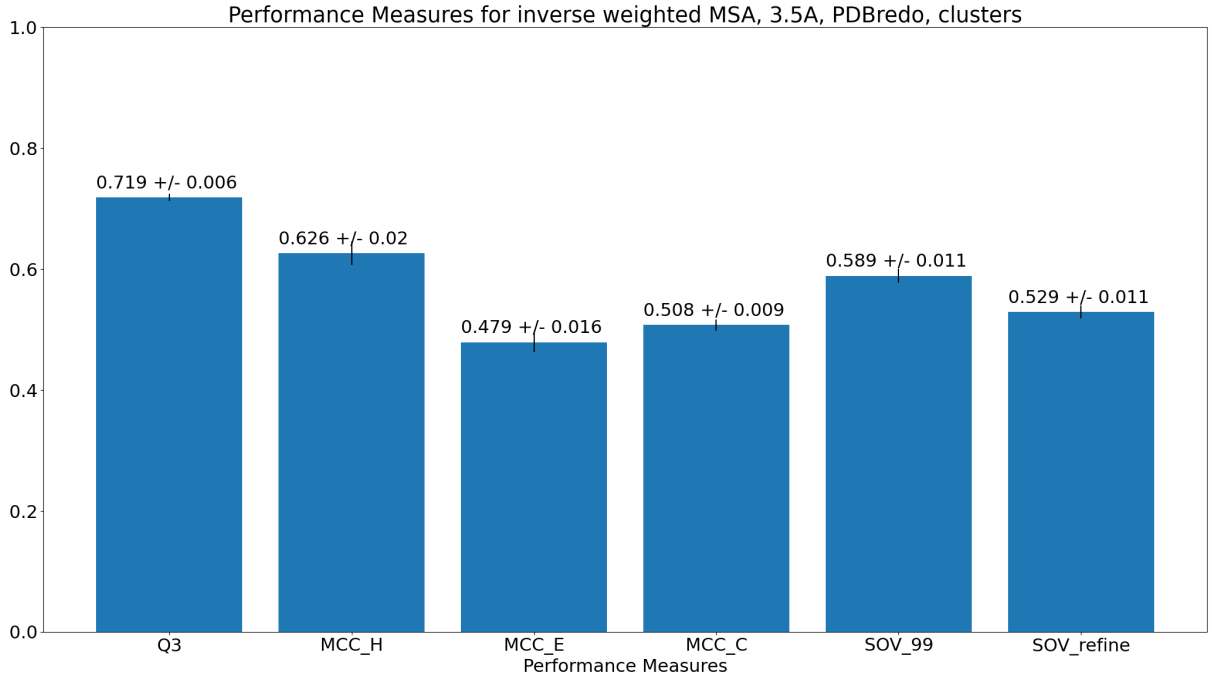

Figure 7.47: Q_3, MCC and SOV performance on the validation set for training on sequence clusters from the 3.5Å training set using PDBredo structures as labels and inverse weighted averaged MSA embeddings as input

7.8 Confusion Matrices

7.8.1 Single Sequence Embeddings

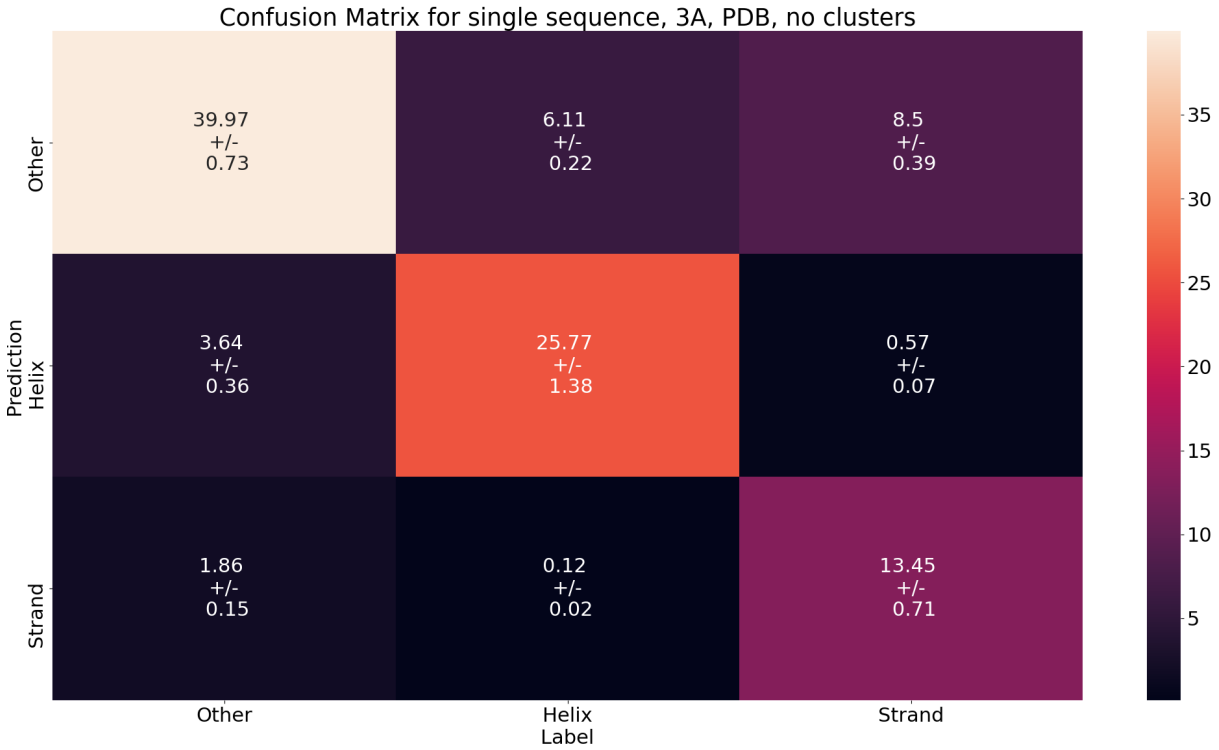

Figure 7.48: Confusion matrix for predictions of the validation set for training on the 3Å training set using PDB structure as labels and single sequence embeddings as input

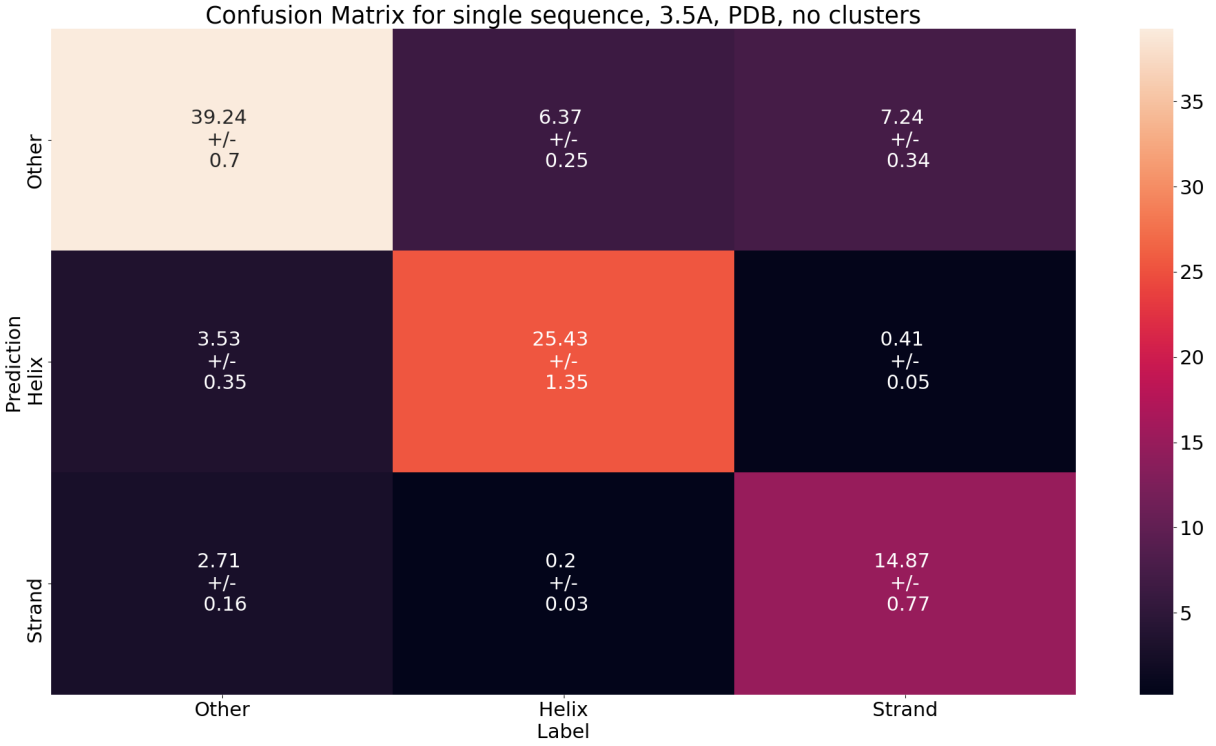

Figure 7.49: Confusion matrix for predictions of the validation set for training on the 3.5Å training set using PDB structure as labels and single sequence embeddings as input

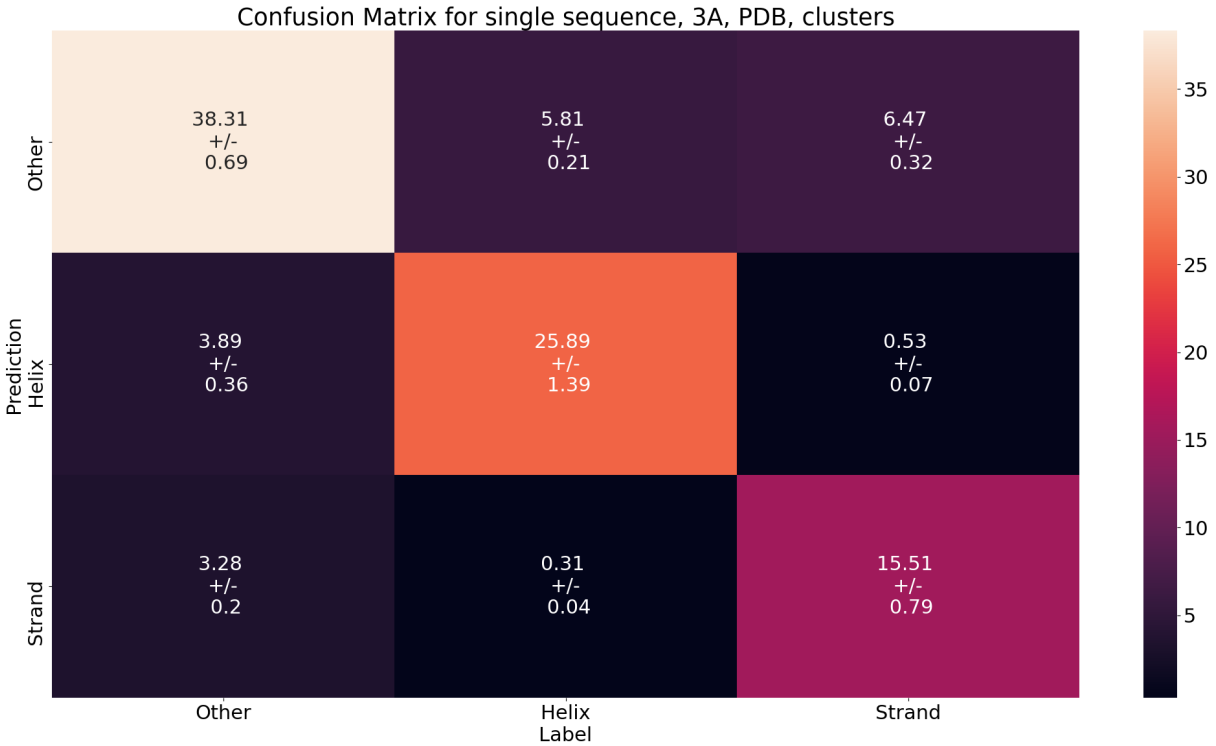

Figure 7.50: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3Å training set using PDB structure as labels and single sequence embeddings as input

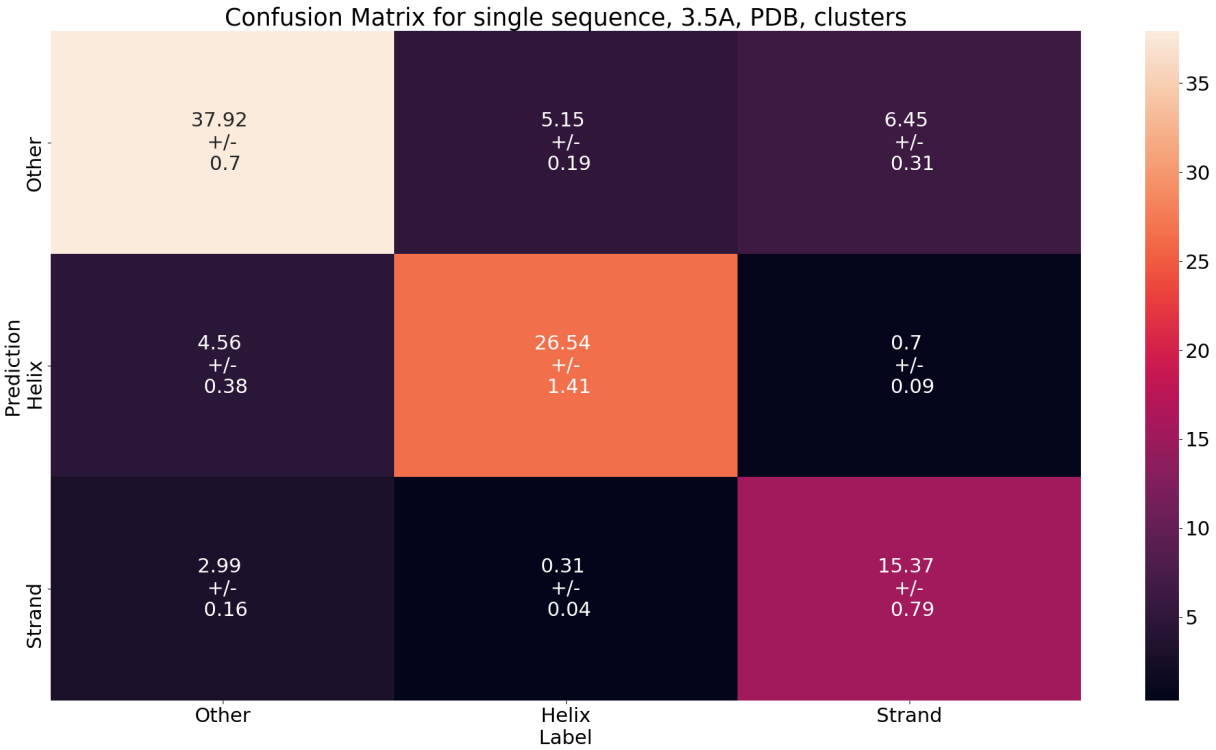

Figure 7.51: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3.5Å training set using PDB structure as labels and single sequence embeddings as input

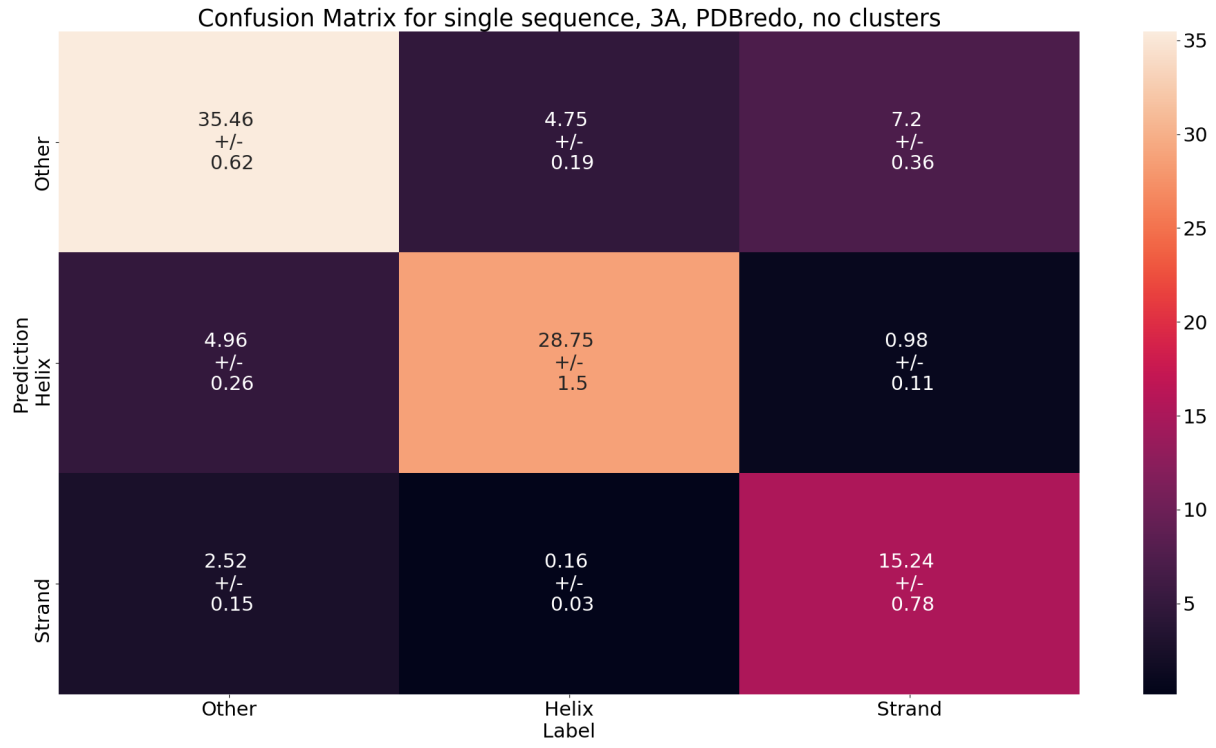

Figure 7.52: Confusion matrix for predictions of the validation set for training on the 3Å training set using PDBredo structures as labels and single sequence embeddings as input

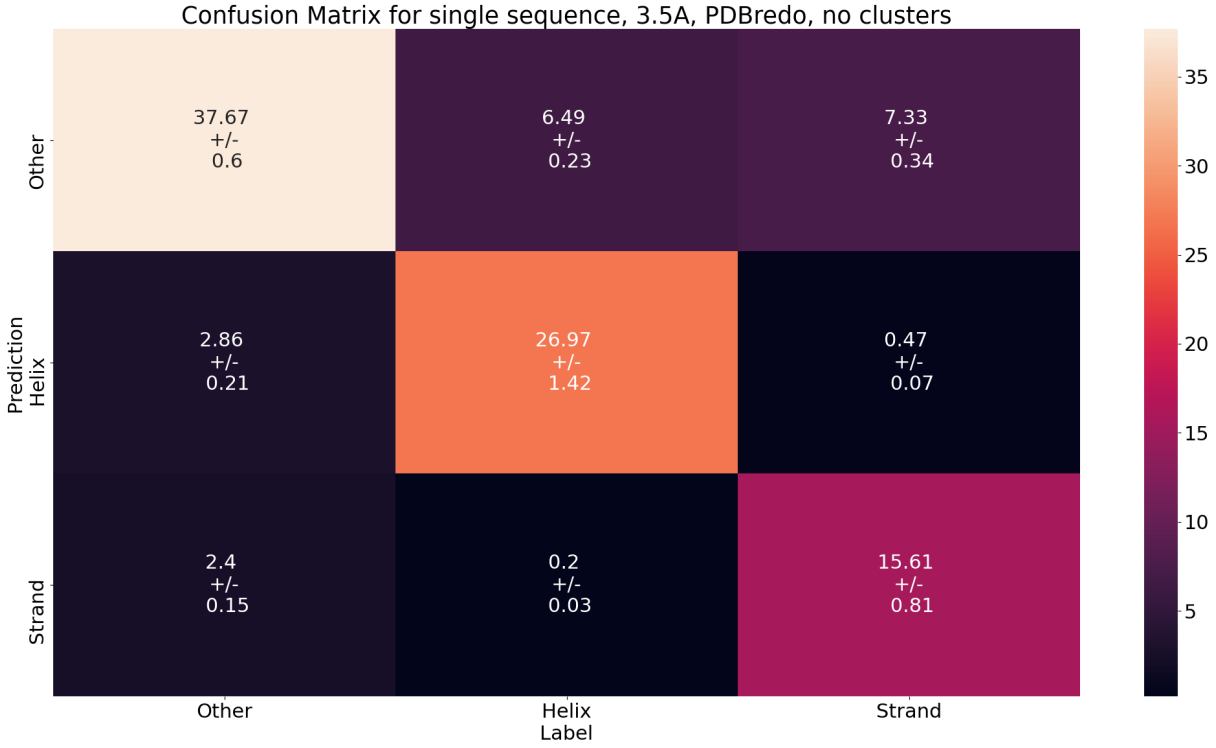

Figure 7.53: Confusion matrix for predictions of the validation set for training on the 3.5Å training set using PDBredo structures as labels and single sequence embeddings as input

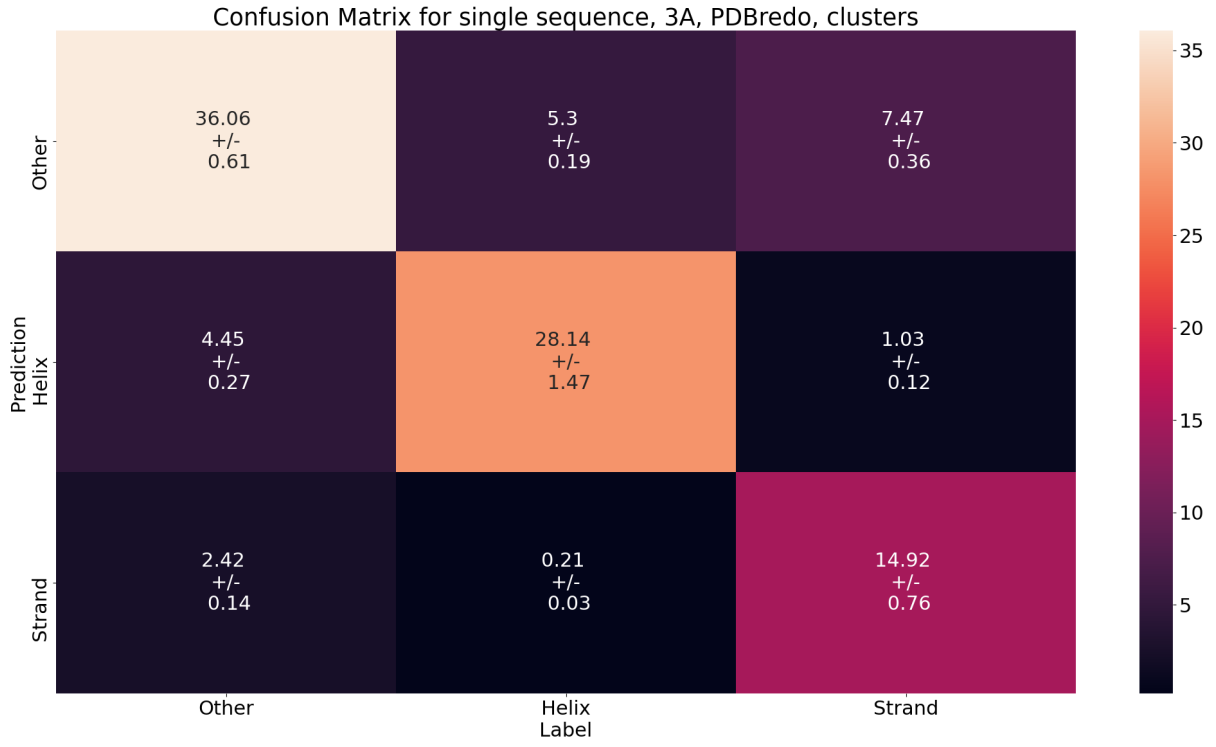

Figure 7.54: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3Å training set using PDBredo structures as labels and single sequence embeddings as input

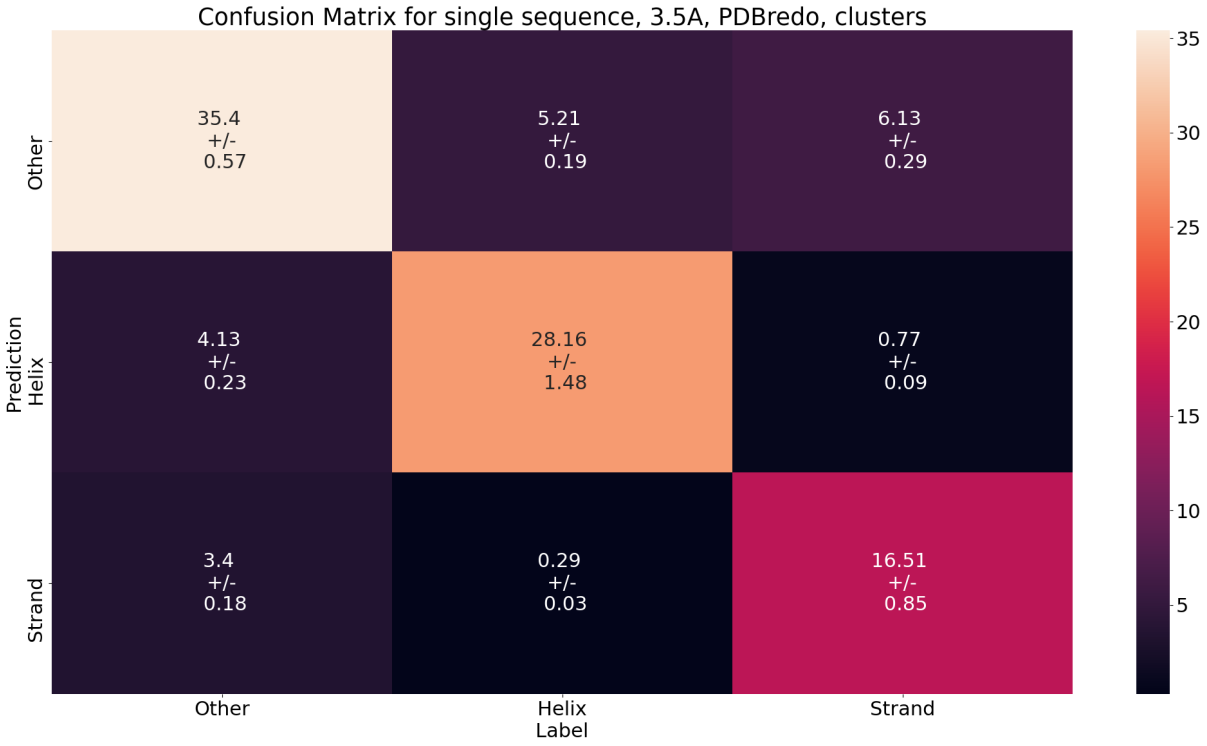

Figure 7.55: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3.5Å training set using PDBredo structures as labels and single sequence embeddings as input

7.8.2 Averaged MSA Embeddings

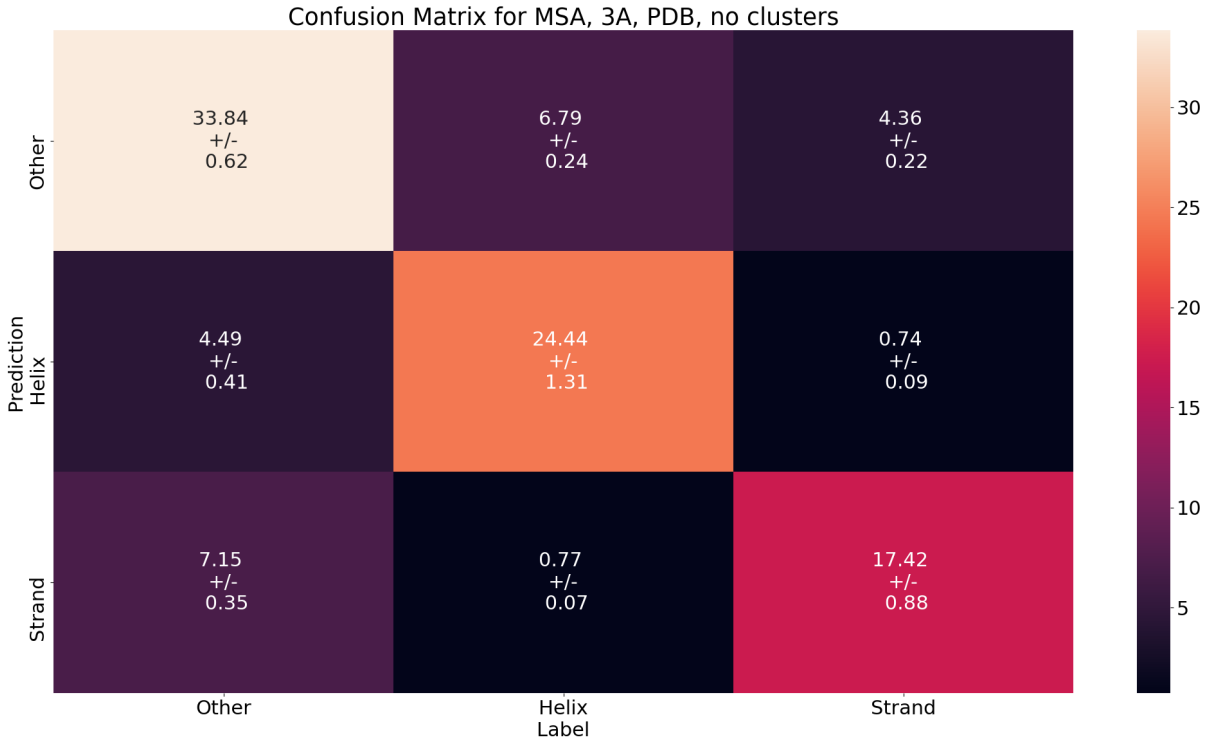

Figure 7.56: Confusion matrix for predictions of the validation set for training on the 3Å training set using PDB structures as labels and MSA embeddings as input

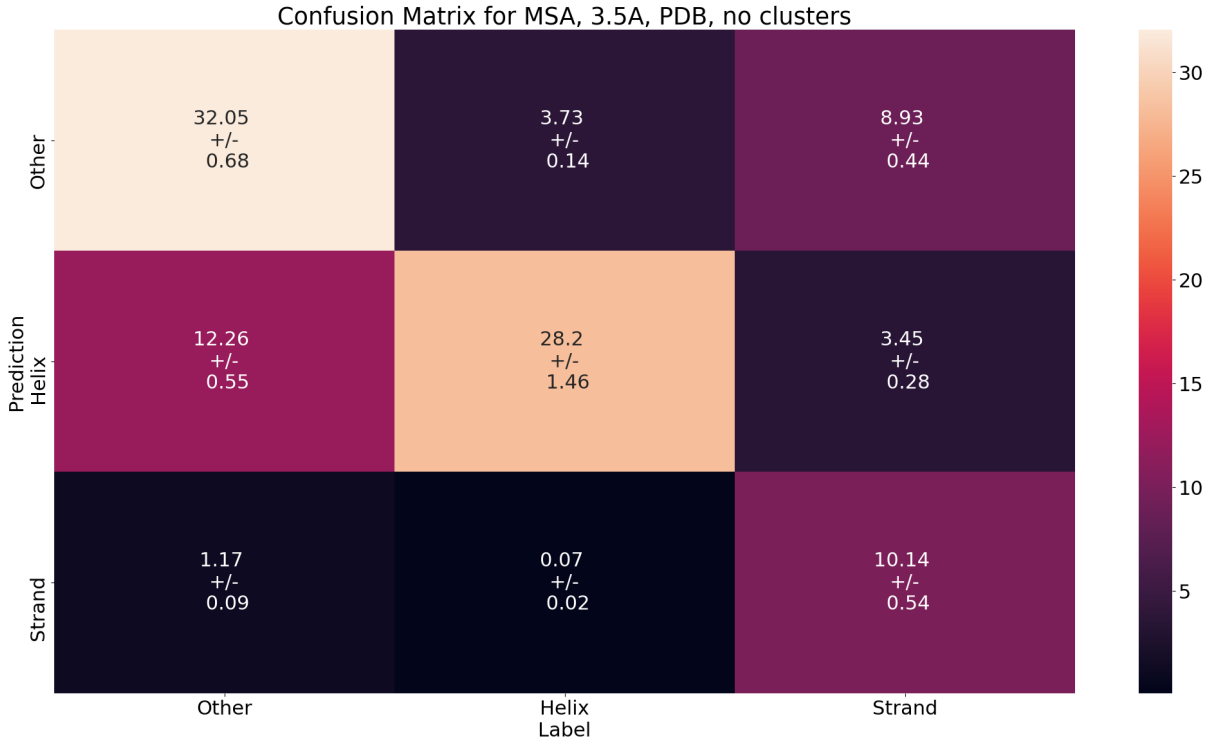

Figure 7.57: Confusion matrix for predictions of the validation set for training on the 3.5Å training set using PDB structures as labels and MSA embeddings as input

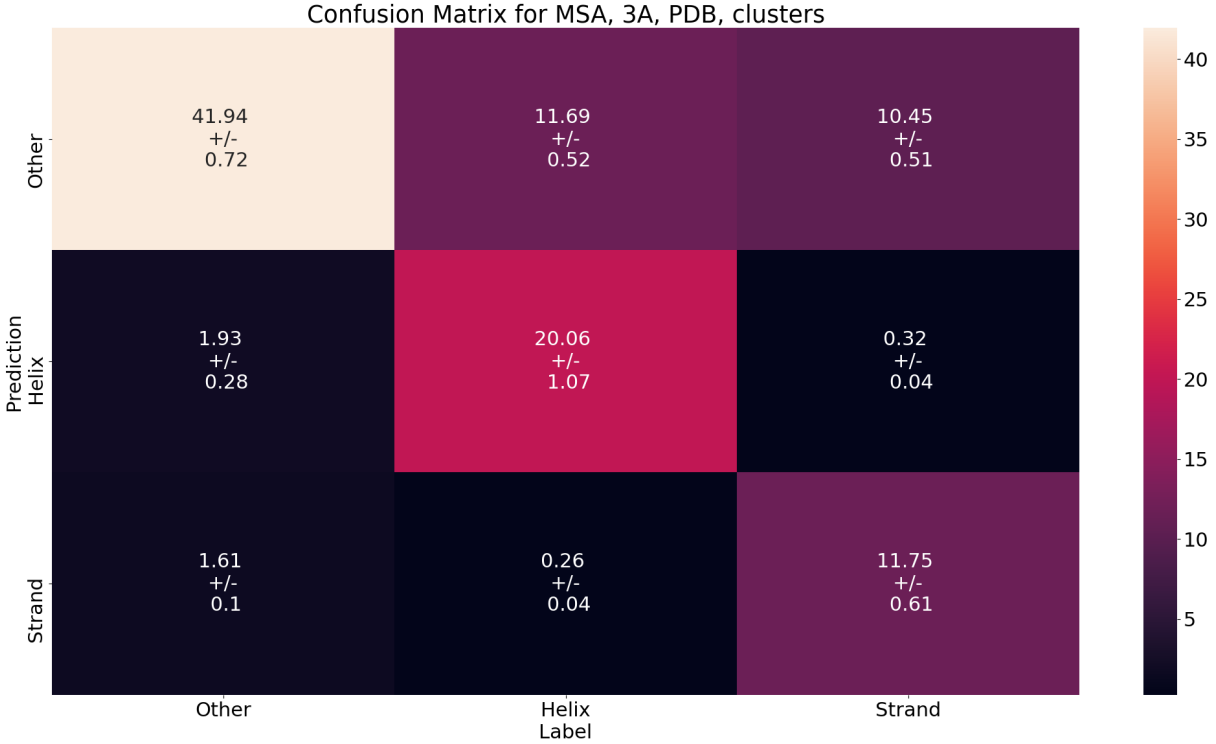

Figure 7.58: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3Å training set using PDB structures as labels and MSA embeddings as input

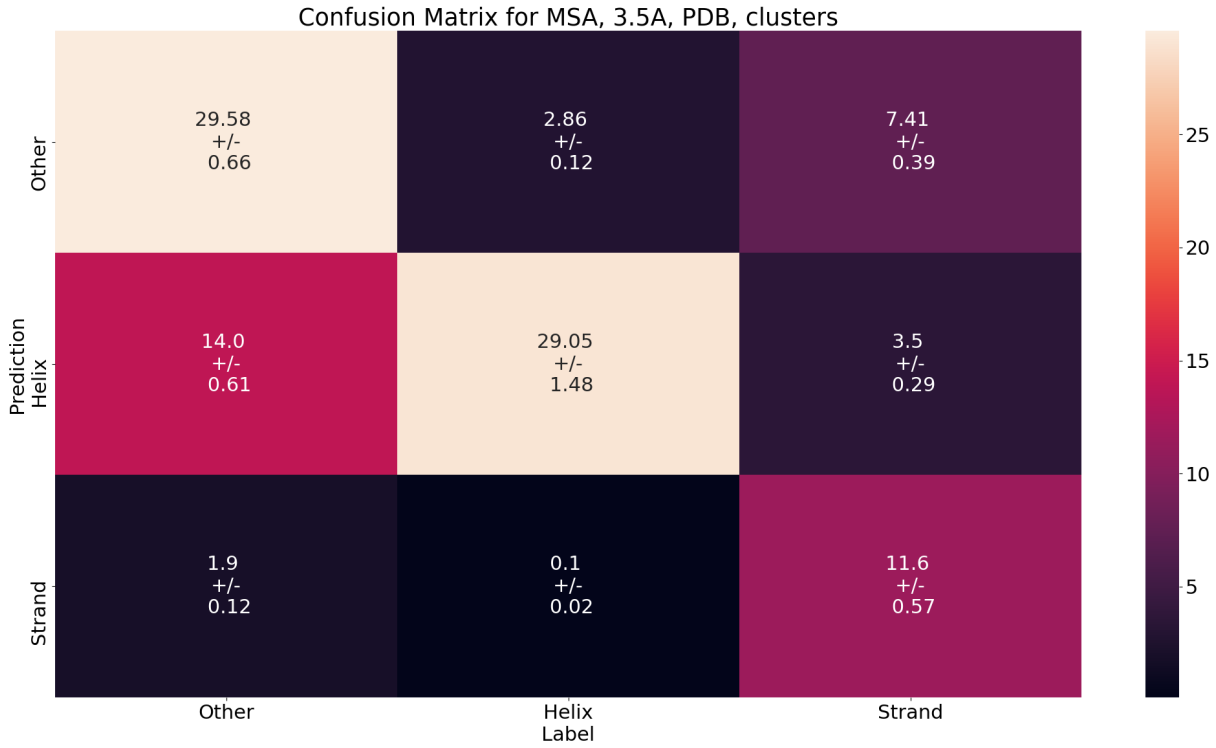

Figure 7.59: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3.5Å training set using PDB structures as labels and MSA embeddings as input

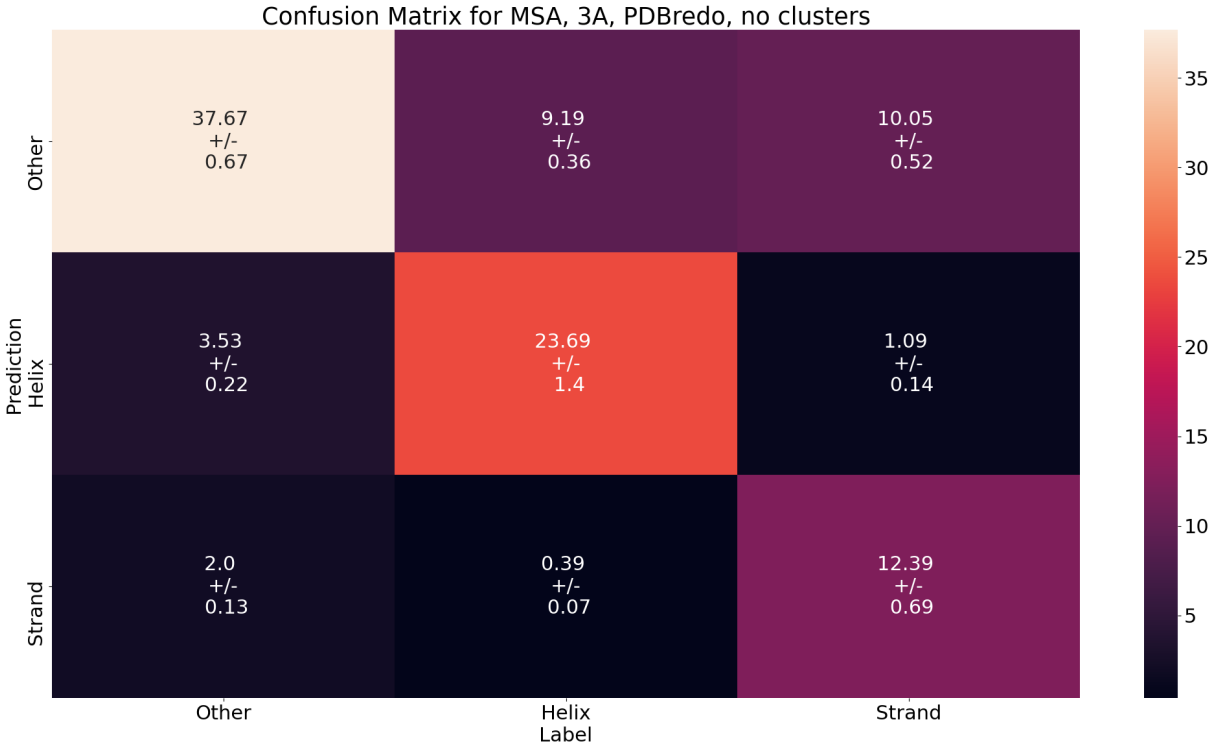

Figure 7.60: Confusion matrix for predictions of the validation set for training on the 3Å training set using PDBredo structures as labels and MSA embeddings as input

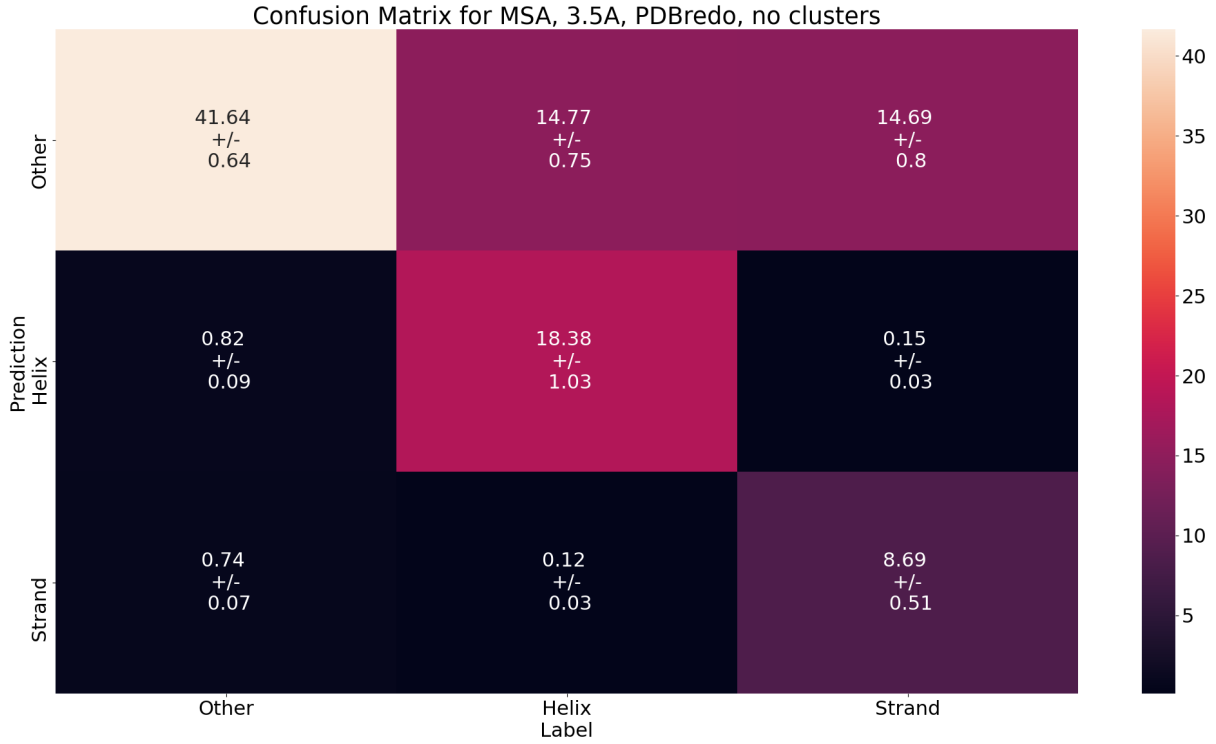

Figure 7.61: Confusion matrix for predictions of the validation set for training on the 3.5Å training set using PDBredo structures as labels and MSA embeddings as input

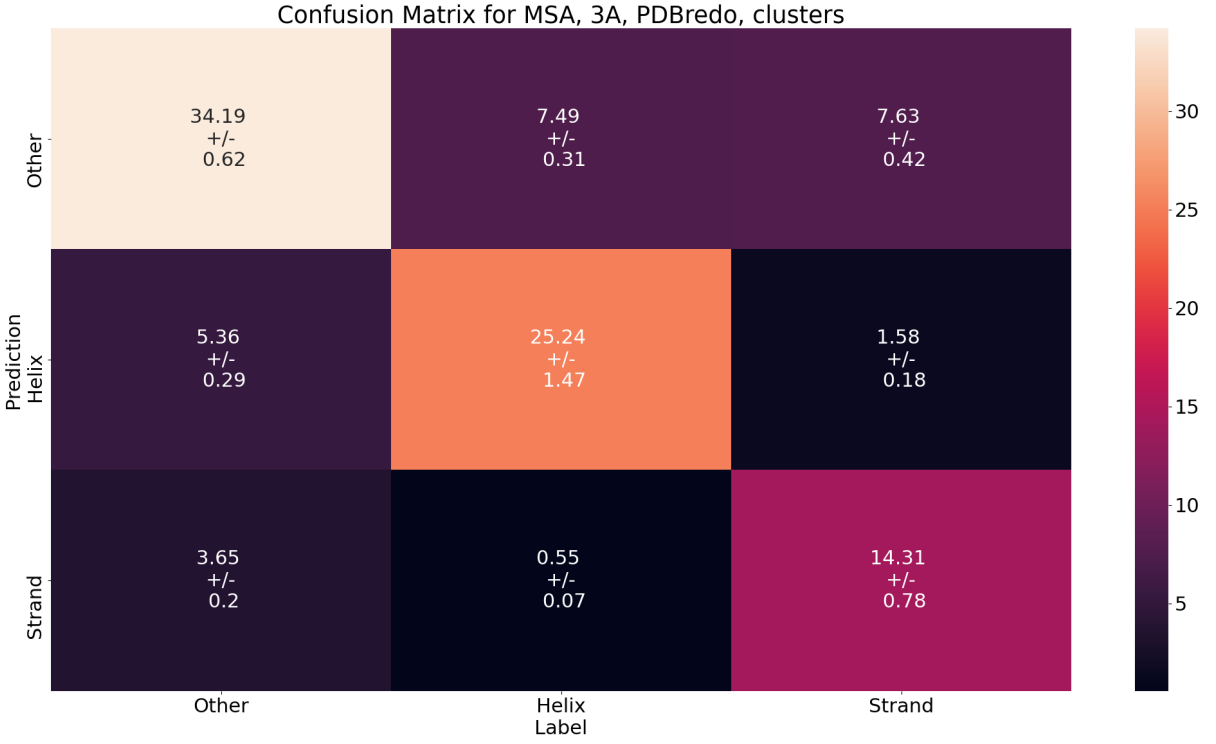

Figure 7.62: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3Å training set using PDBredo structures as labels and MSA embeddings as input

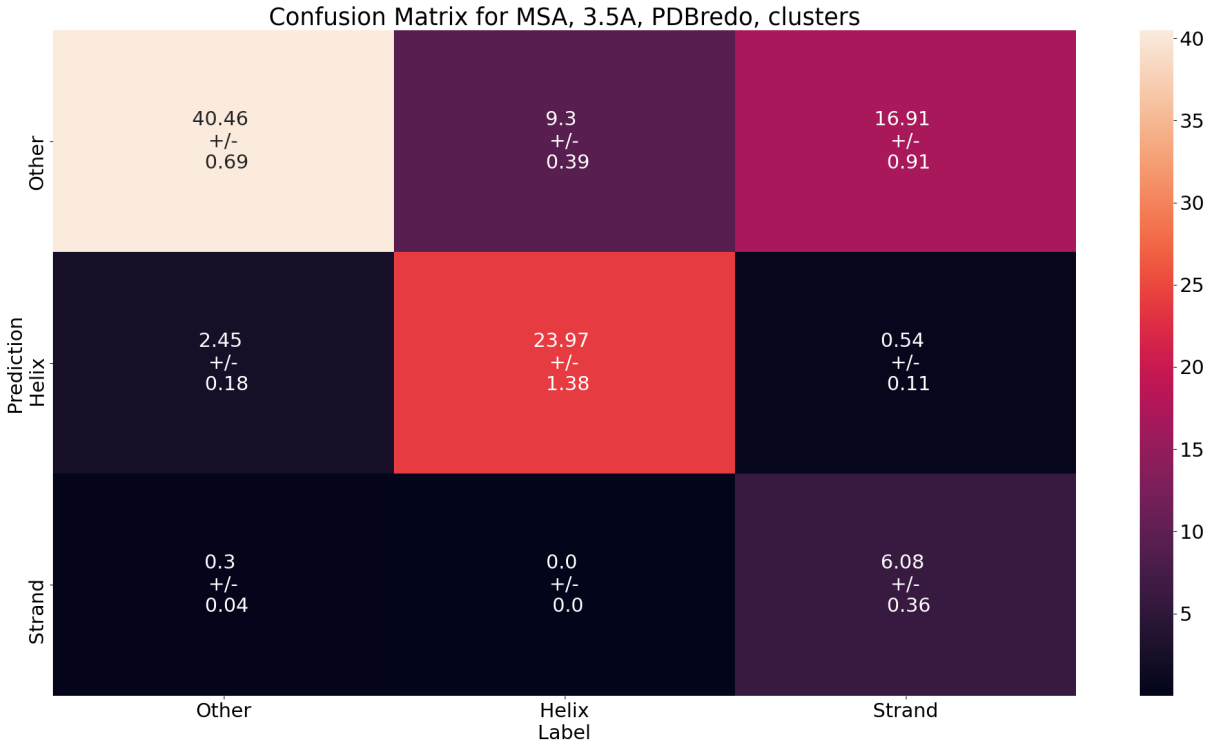

Figure 7.63: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3.5Å training set using PDBredo structures as labels and MSA embeddings as input

7.8.3 Weighted MSA Embeddings

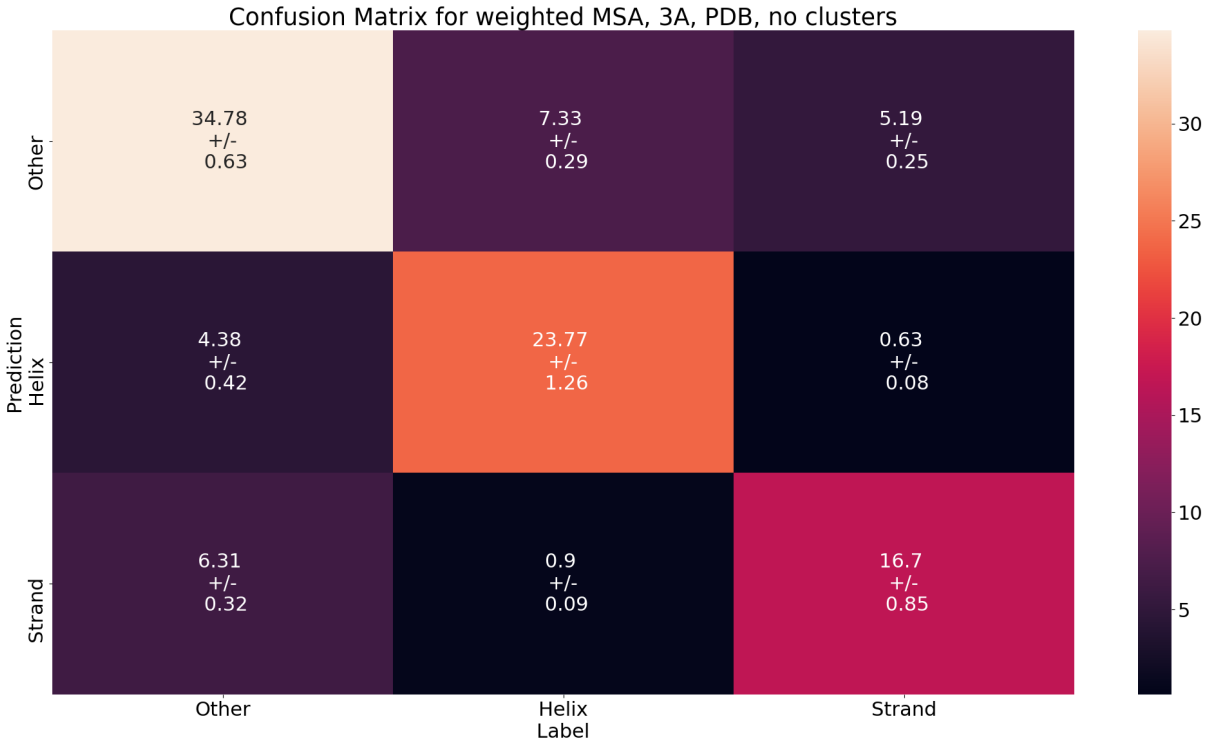

Figure 7.64: Confusion matrix for predictions of the validation set for training on the 3Å training set using PDB structures as labels and weighted MSA embeddings as input

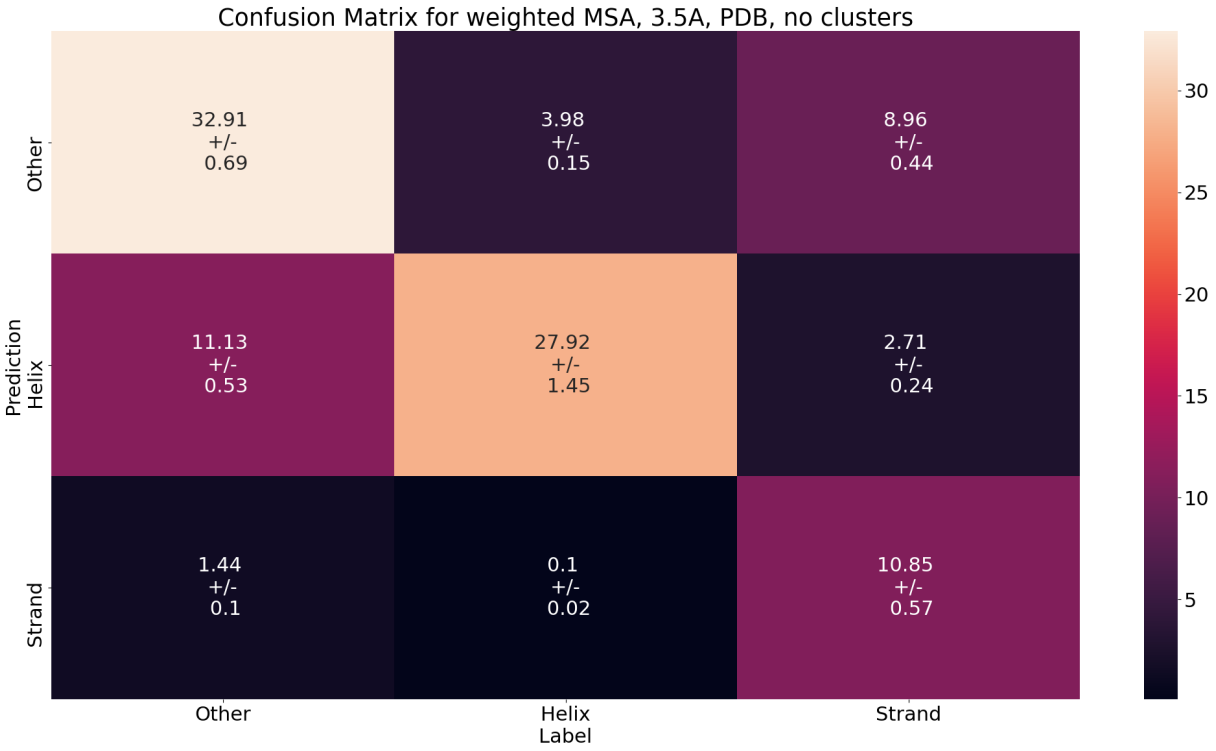

Figure 7.65: Confusion matrix for predictions of the validation set for training on the 3.5Å training set using PDB structures as labels and weighted MSA embeddings as input

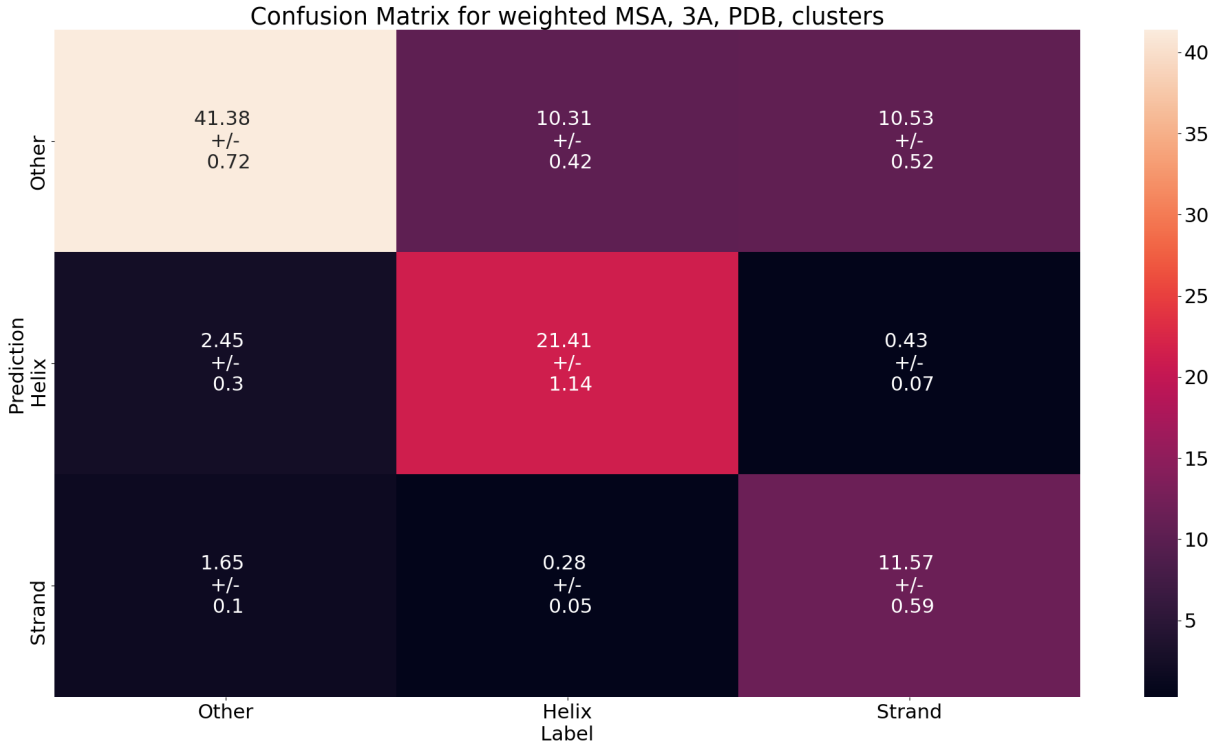

Figure 7.66: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3Å training set using PDB structures as labels and weighted MSA embeddings as input

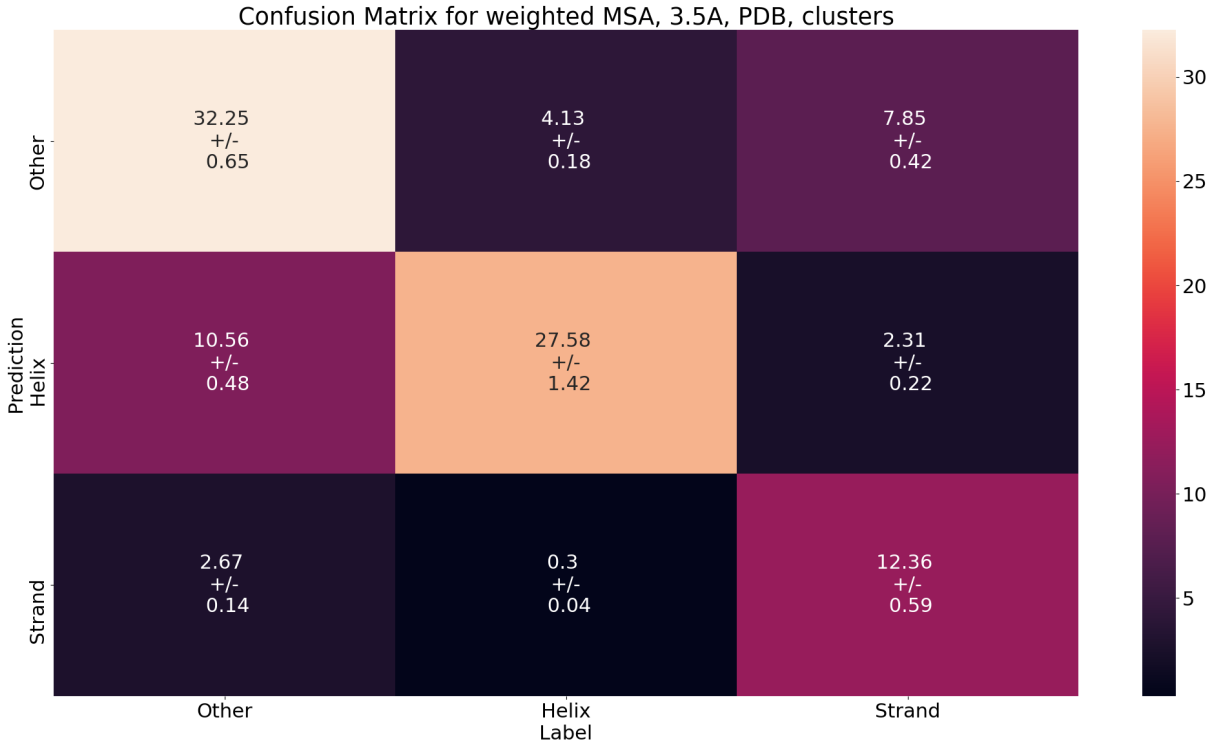

Figure 7.67: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3.5Å training set using PDB structures as labels and weighted MSA embeddings as input

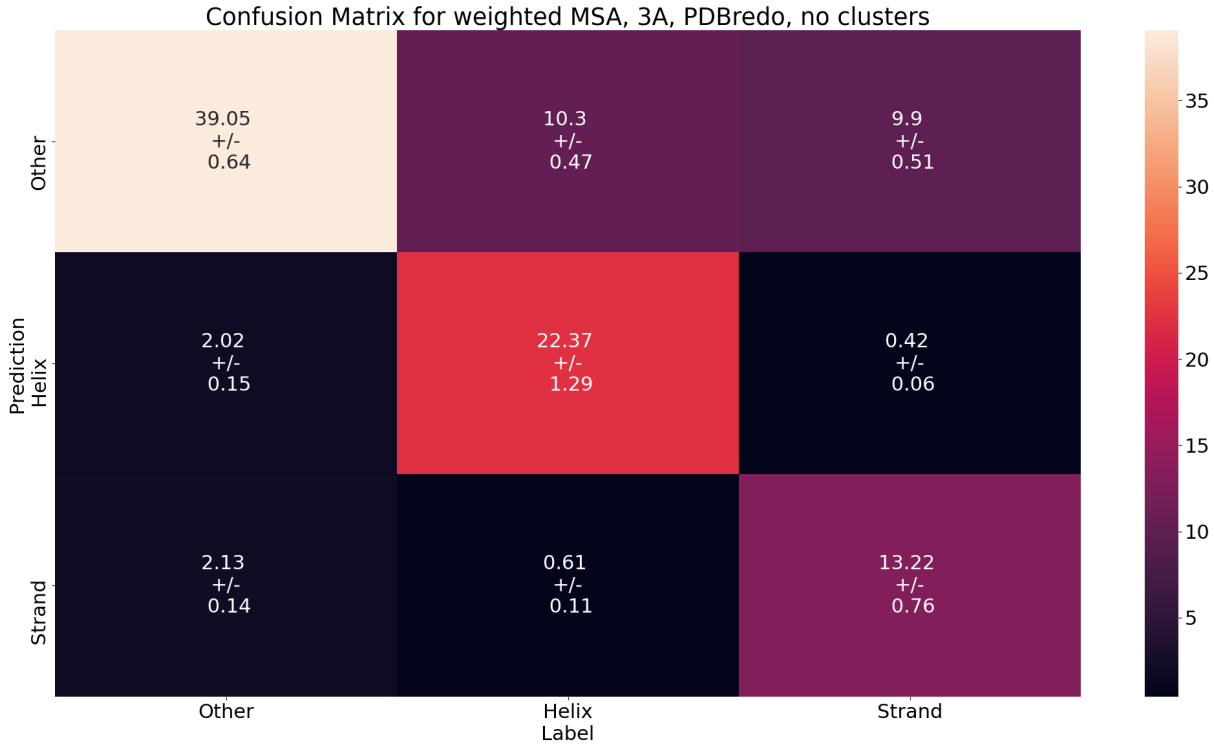

Figure 7.68: Confusion matrix for predictions of the validation set for training on the 3Å training set using PDBredo structures as labels and weighted MSA embeddings as input

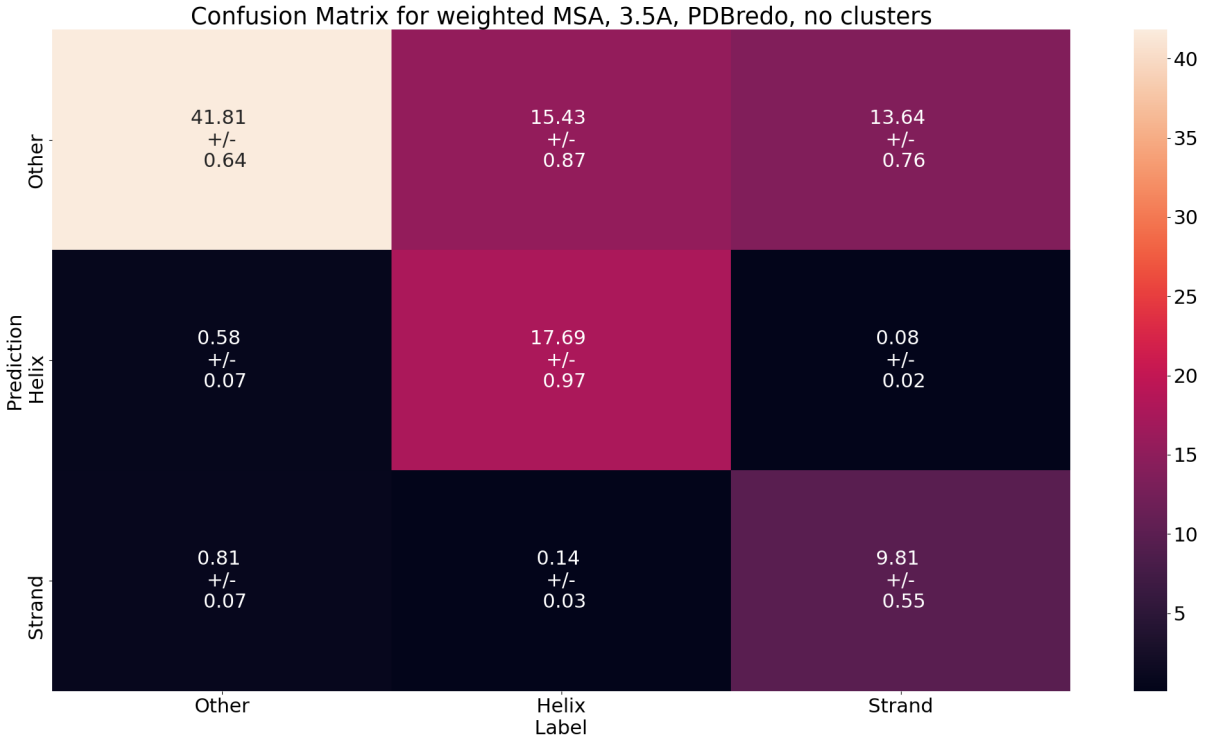

Figure 7.69: Confusion matrix for predictions of the validation set for training on the 3.5Å training set using PDBredo structures as labels and weighted MSA embeddings as input

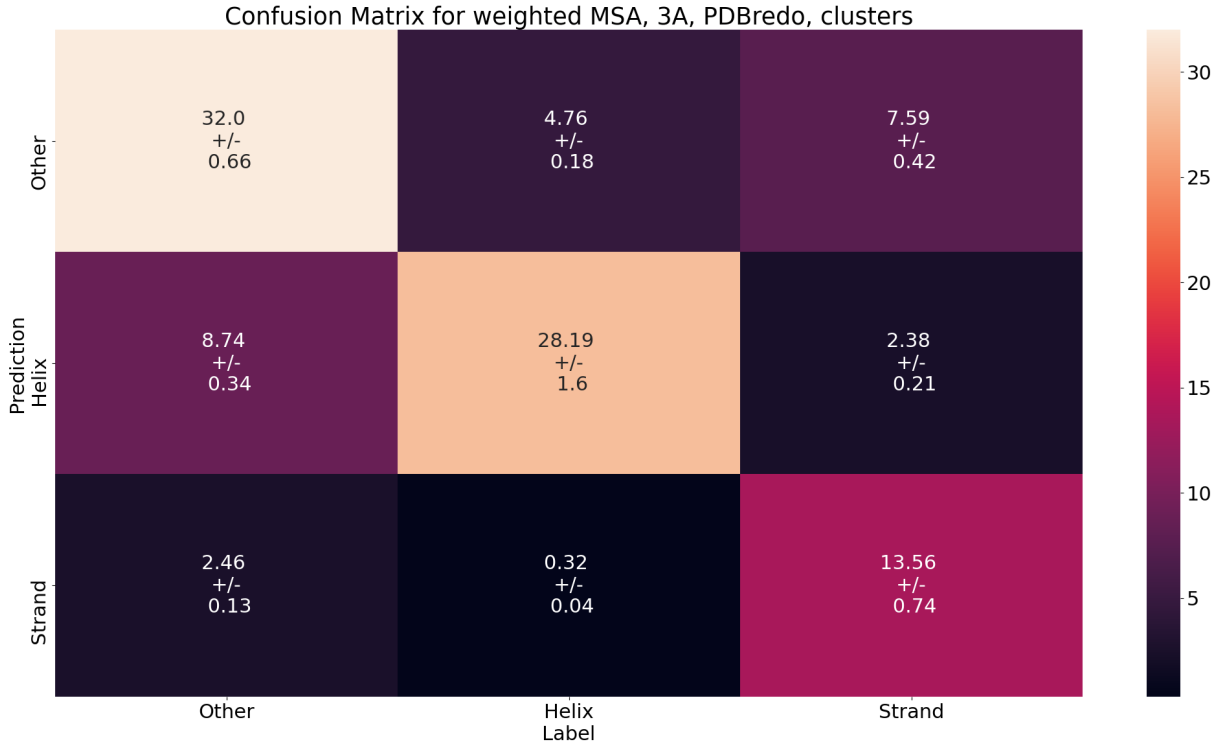

Figure 7.70: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3Å training set using PDBredo structures as labels and weighted MSA embeddings as input

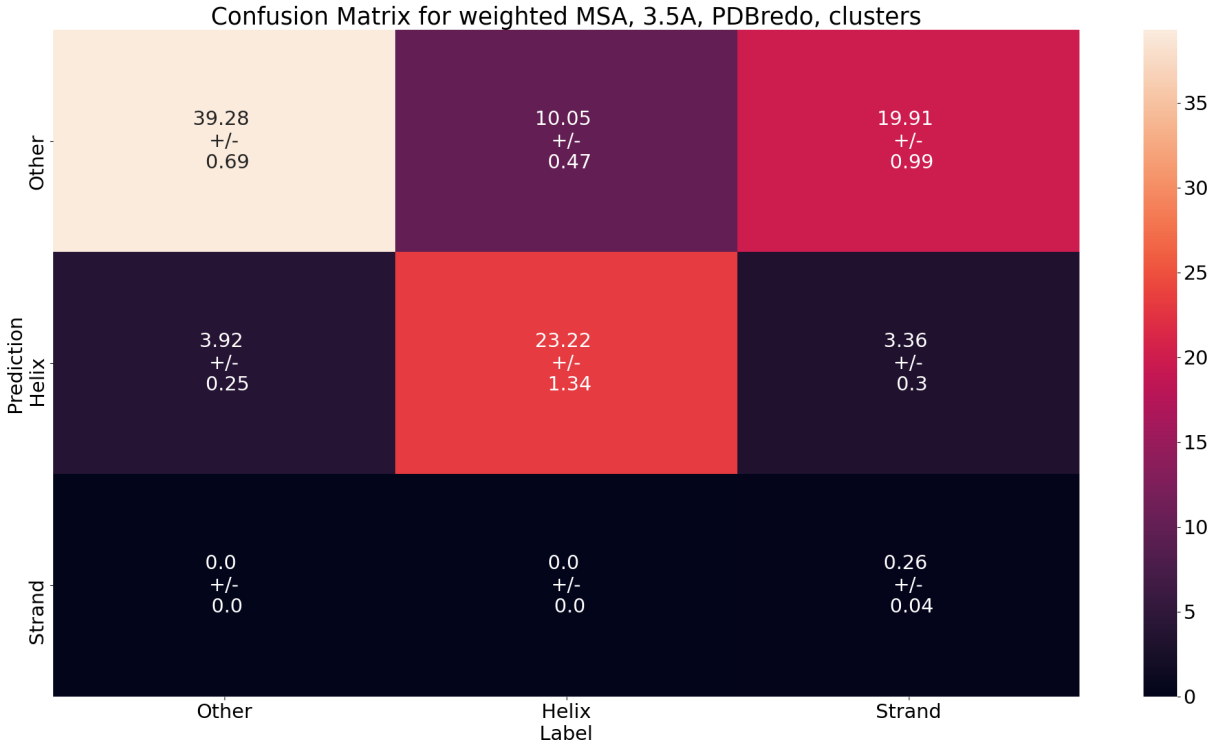

Figure 7.71: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3.5Å training set using PDBredo structures as labels and weighted MSA embeddings as input

7.8.4 Inverse Weighted MSA Embeddings

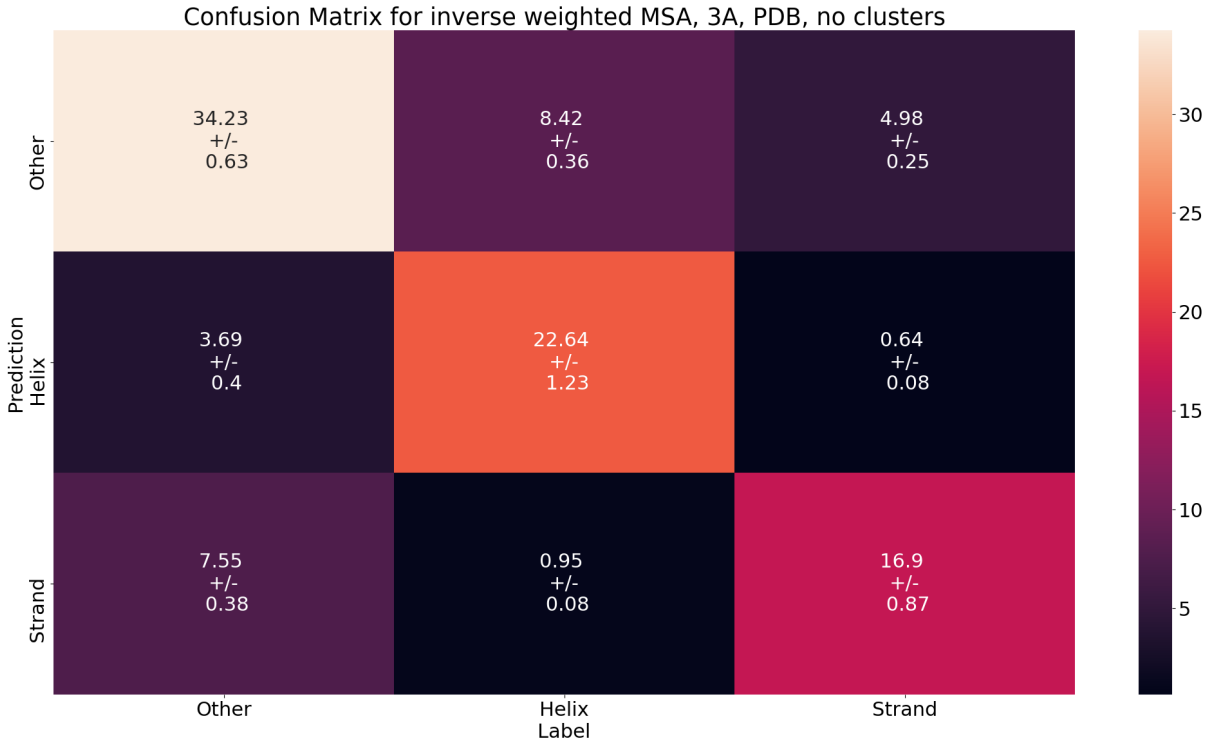

Figure 7.72: Confusion matrix for predictions of the validation set for training on the 3Å training set using PDB structures as labels and inverse weighted MSA embeddings as input

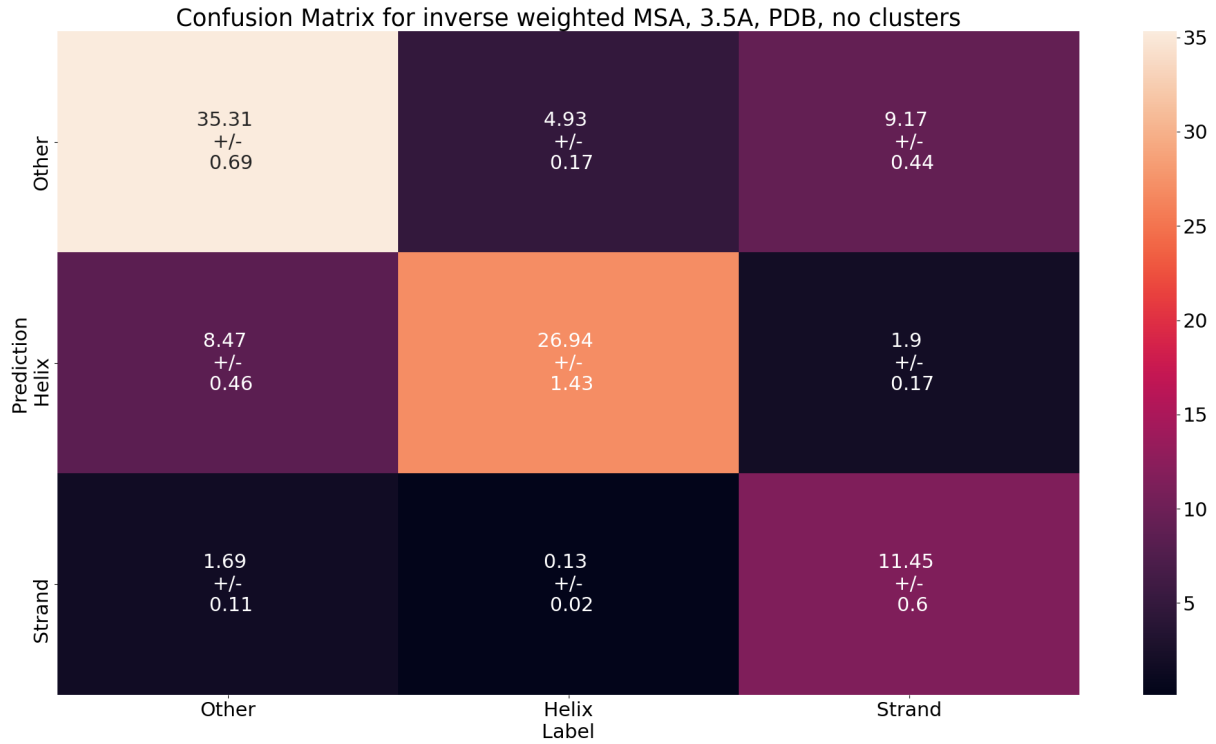

Figure 7.73: Confusion matrix for predictions of the validation set for training on the 3.5Å training set using PDB structures as labels and inverse weighted MSA embeddings as input

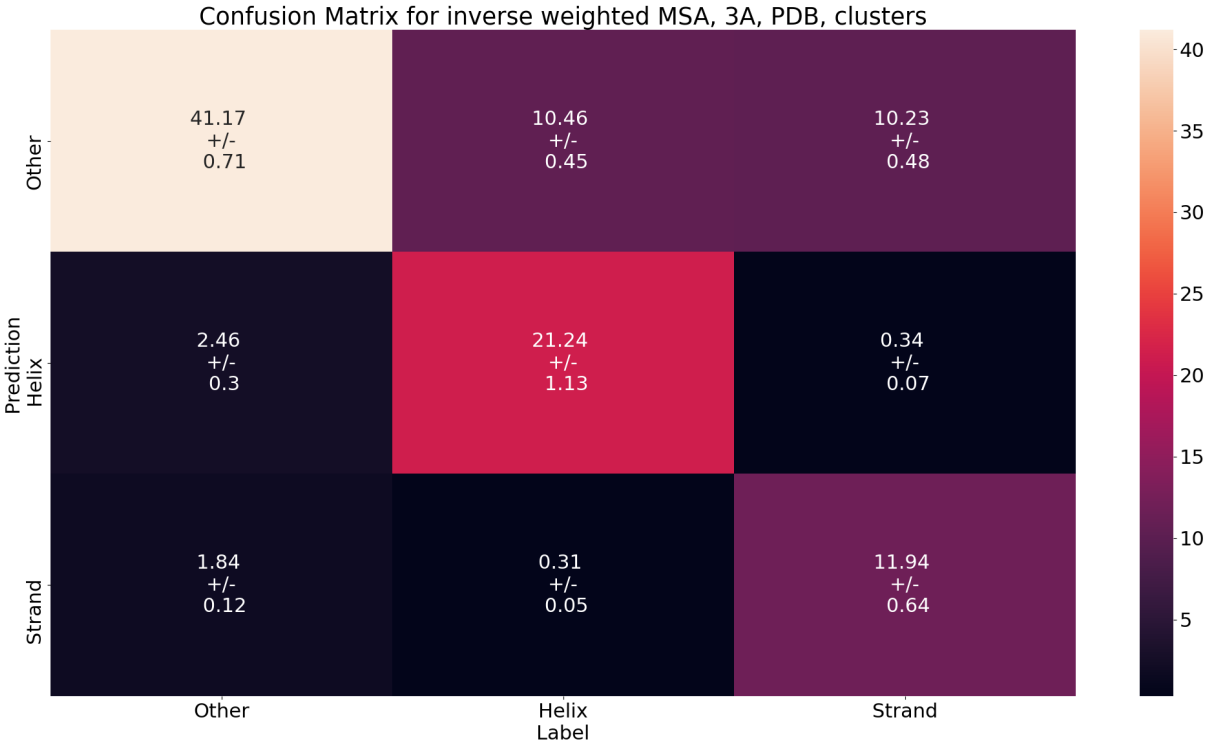

Figure 7.74: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3Å training set using PDB structures as labels and inverse weighted MSA embeddings as input

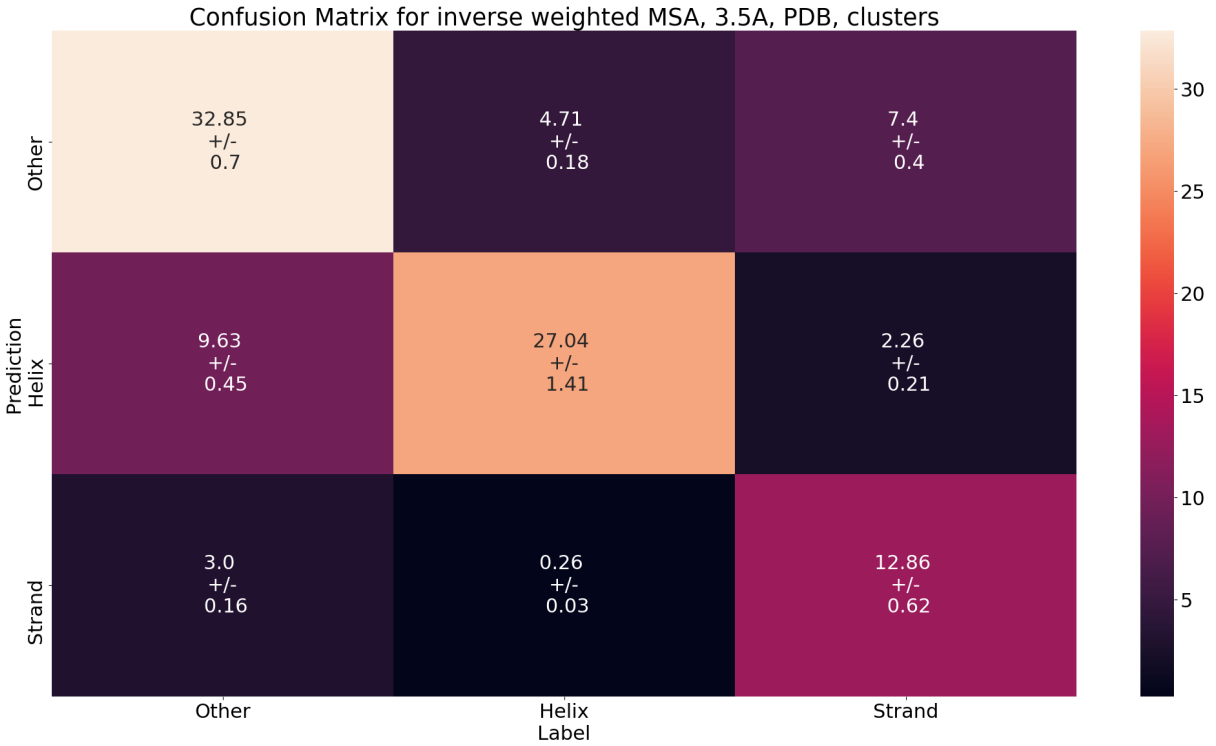

Figure 7.75: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3.5Å training set using PDB structures as labels and inverse weighted MSA embeddings as input

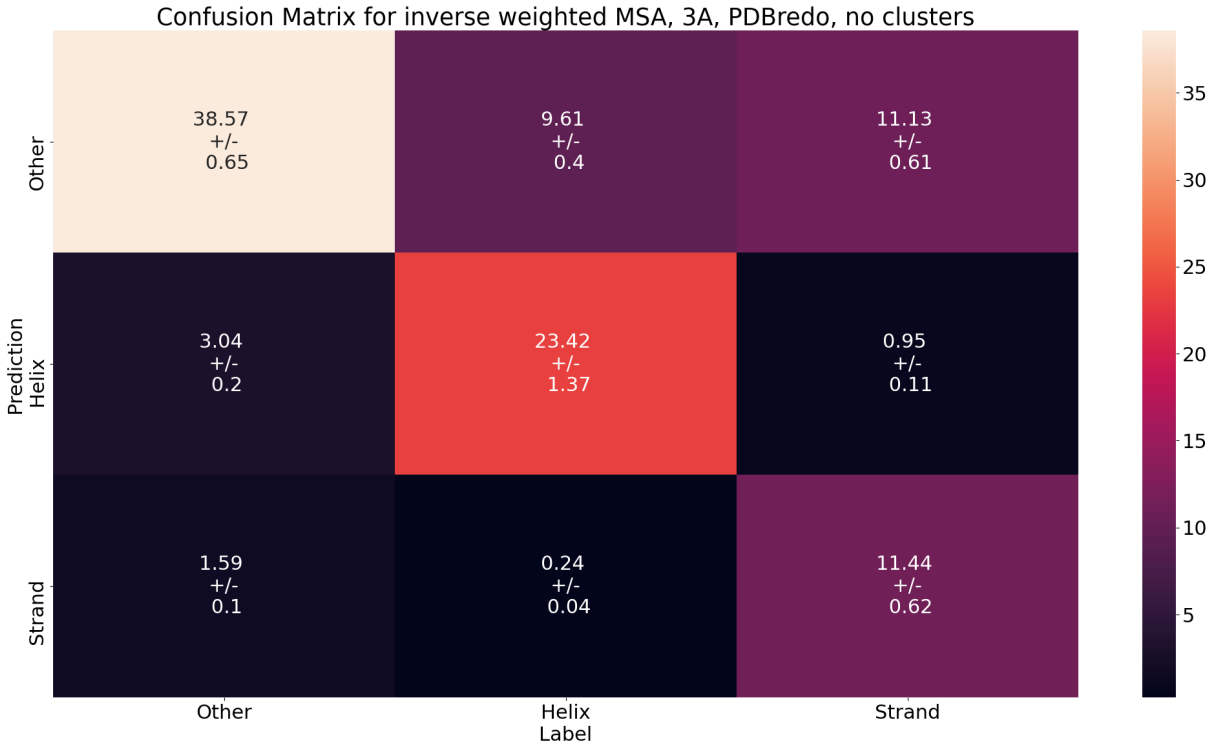

Figure 7.76: Confusion matrix for predictions of the validation set for training on the 3Å training set using PDBredo structures as labels and inverse weighted MSA embeddings as input

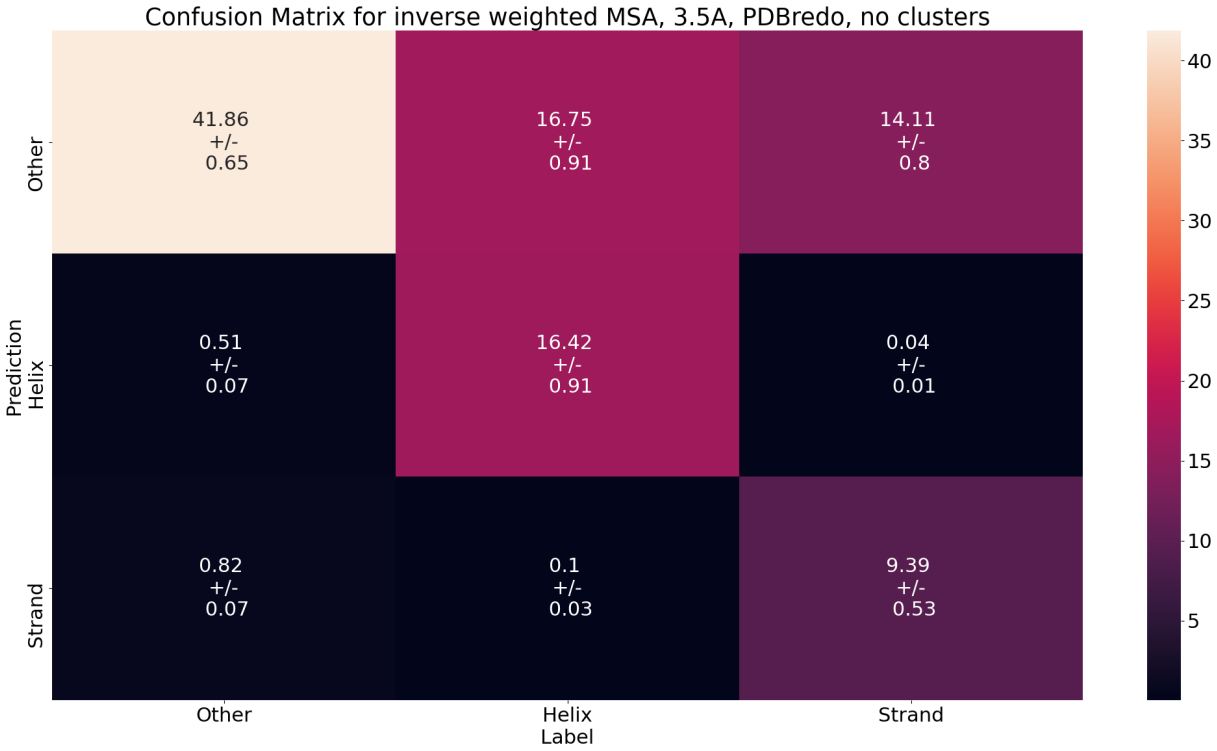

Figure 7.77: Confusion matrix for predictions of the validation set for training on the 3.5Å training set using PDBredo structures as labels and inverse weighted MSA embeddings as input

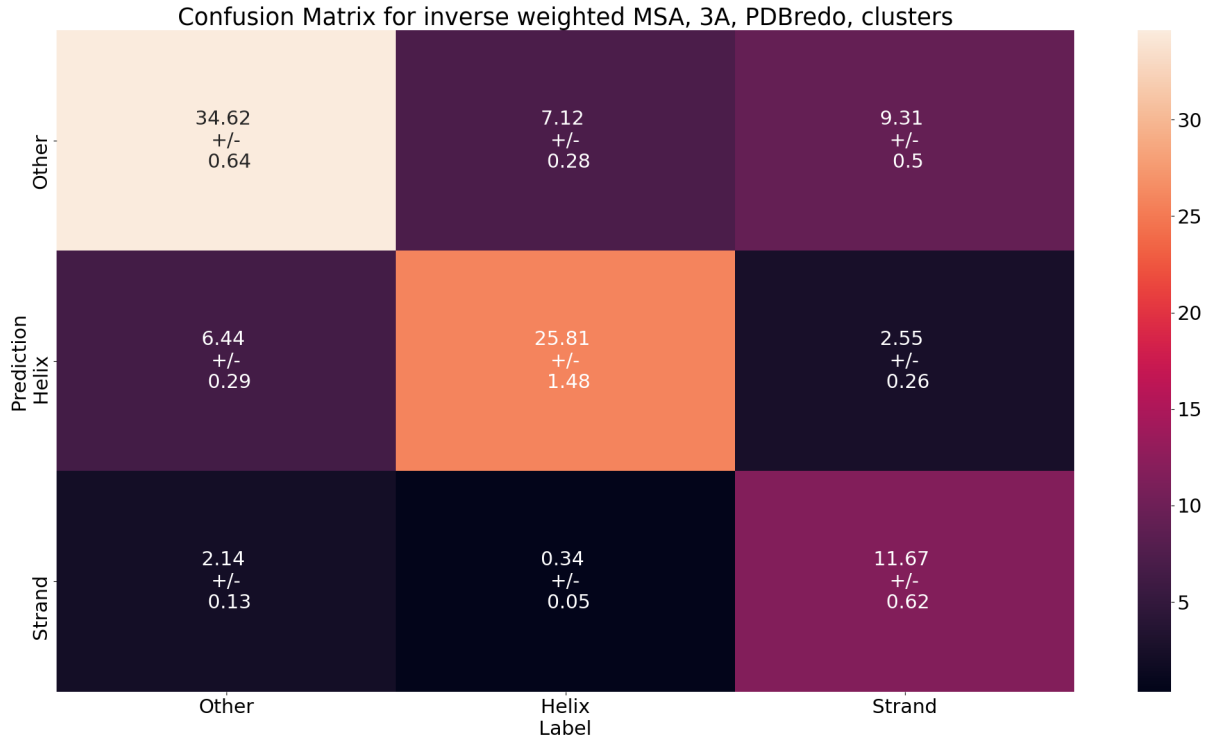

Figure 7.78: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3Å training set using PDBredo structures as labels and inverse weighted MSA embeddings as input

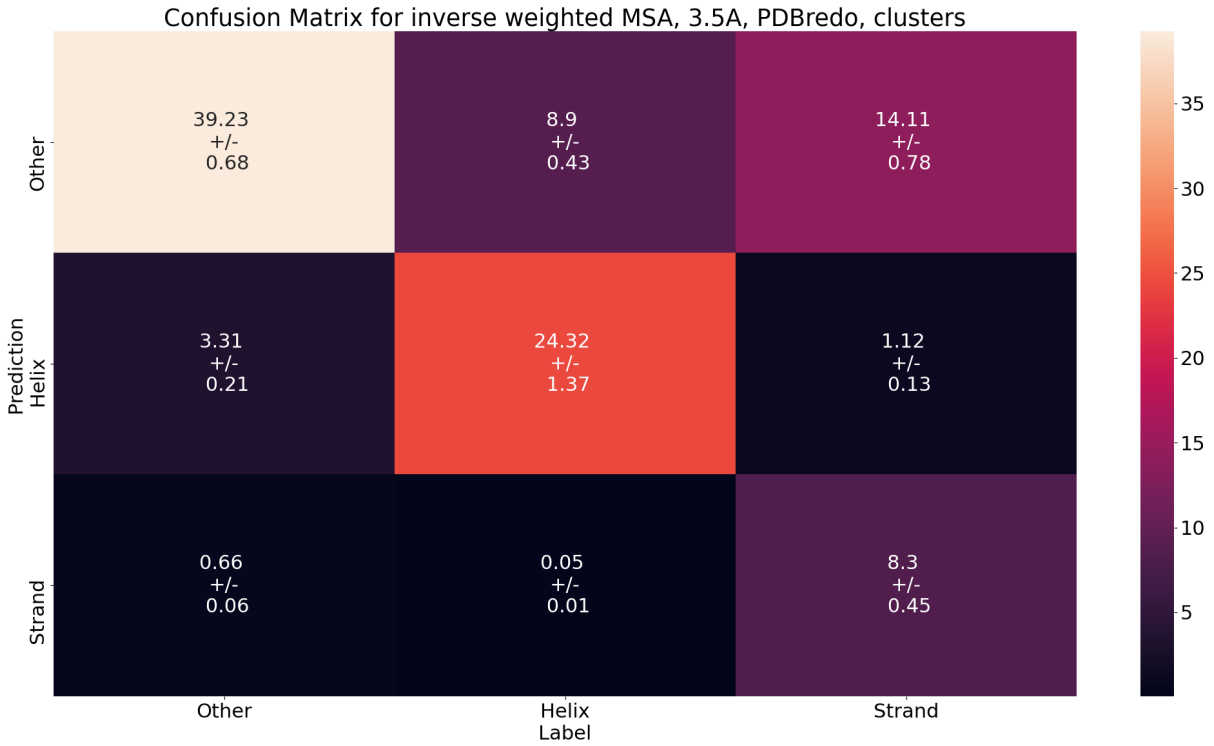

Figure 7.79: Confusion matrix for predictions of the validation set for training on sequence clusters from the 3.5Å training set using PDBredo structures as labels and inverse weighted MSA embeddings as input

8. Acknowledgment

I would first like to thank Michael Heinzinger, Christian Dallago and Burkhard Rost from the Rostlab at the Technical University of Munich for their ongoing support and for answering my questions during the project.
Furthermore, I would like to thank Qifang Xu from the Dunbrack Lab at the Fox Chase Cancer Center for sharing data and taking the time to answer all questions related to it.
Last but not least, I would also like to thank all those people, that deposit experimental data to public databases, as well as those annotating and maintaining these databases. Without them, this work would not have been possible.

Bibliography

[1] Frances C. Bernstein, Thomas F. Koetzle, Graheme J.B. Williams, Edgar F. Meyer, Michael D. Brice, John R. Rodgers, Olga Kennard, Takehiko Shimanouchi, and Mitsuo Tasumi. The protein data bank: A computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, 112(3):535–542, 1977.

[2] Manuele Bicego, Vittorio Murino, and Mário A.T. Figueiredo. Similarity-based classification of sequences using hidden markov models. *Pattern Recognition*, 37(12):2281–2291, 2004.

[3] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv*, 2020.

[4] J. Garnier, D.J. Osguthorpe, and B. Robson. Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*, 120(1):97–120, 1978.

[5] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling the language of life – deep learning protein sequences. *bioRxiv*, 2019.

[6] Robbie P. Joosten, Tim A.H. te Beek, Elmar Krieger, Maarten L. Hekkelman, Rob W.W. Hooft, Reinhard Schneider, Chris Sander, and Gert Vriend. A series of PDB related databases for everyday needs. *Nucleic Acids Research*, 39(suppl_1):D411–D419, 11 2010.

[7] Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjærgaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Sønderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, and Paolo Marcatili. Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, 2019.

[8] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (CASP)-Round XIII. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1011–1020, 2019.

[9] Tong Liu and Zheng Wang. Sov_refine: A further refined definition of segment overlap score and its significance for protein structure similarity. *Source code for biology and medicine*, 13(1), 2018.

[10] Lewis Moffat and David T. Jones. Semi-supervised learning of protein secondary structure from single sequences. *bioRxiv*, 2020.

[11] Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F. Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. *bioRxiv*, 2021.

[12] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, 12(2):85–94, 02 1999.

[13] Burkhard Rost, Chris Sander, and Reinhard Schneider. Redefining the goals of protein secondary structure prediction. *Journal of Molecular Biology*, 235(1):13–26, 1994.

[14] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, Jan 2020.

[15] Kresimir Sikic and Oliviero Carugo. Protein sequence redundancy reduction: comparison of various method. *Bioinformation*, 5(6):234–239, Nov 2010. 21364823[pmid].

[16] Ian Sillitoe, Natalie Dawson, Tony E. Lewis, Sayoni Das, Jonathan G. Lees, Paul Ashford, Adeyelu Tolulope, Harry M. Scholes, Ilya Senatorov, Andra Bujan, Fatima Ceballos Rodriguez-Conde, Benjamin Dowling, Janet Thornton, and Christine A. Orengo. CATH: expanding the horizons of structure-based functional annotations for genome sequences. *Nucleic Acids Research*, 47(D1):D280–D284, 11 2018.

[17] Jesse Vig, Ali Madani, Lav R. Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. Bertology meets biology: Interpreting attention in protein language models, 2020.

[18] Adam Zemla, Česlovas Venclovas, Krzysztof Fidelis, and Burkhard Rost. A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Bioinformatics*, 34(2):220–223, 1999.

[19] Marketa Zvelebil and Jeremy O. Baum. *understanding bioinformatics*. Garland Science, 1 edition, 2007.