MERIZO: A RAPID AND ACCURATE DOMAIN SEGMENTATION METHOD USING INVARIANT POINT ATTENTION

Andy M. Lau
Department of Computer Science
University College London
London, WC1E 6BT
andy.m.lau@ucl.ac.uk

Shaun M. Kandathil
Department of Computer Science
University College London
London, WC1E 6BT
s.kandathil@ucl.ac.uk

David T. Jones
Department of Computer Science
University College London
London, WC1E 6BT
d.t.jones@ucl.ac.uk

February 19, 2023

ABSTRACT

Protein domains are distinct, modular and locally compact units of protein structures which may fold and function independently to the rest of the protein. Identifying the regions corresponding to a domain is non-trivial, and the classification of domain folds have been extensively documented in databases such as CATH and ECOD. With the advent of 200 million protein models generated by AlphaFold2, the ability to accurately decompose proteins into their constituent domains will allow a deep dive into their compositions and enable many new lines of research. Although there are many existing methods for identifying domains in proteins, they are either inaccurate, too efficient to run, or do not handle discontinuous domains. Here, we describe our deep learning-based approach for domain segmentation called Merizo, which differs significantly from other conventional methods by conducting segmentation in a bottom-up manner by learning to directly cluster residues into domains. Our network is trained fully end-to-end on CATH domains and outperforms current state-of-the-art methods both in predicting correct boundary positions as well as matching the overall domain topology. Merizo will be made available at https://github.com/psipred/Merizo.

1 Introduction

Domains are locally compact regions within proteins that can fold independently of the rest of the protein and can sometimes support a biological function on its own. The fold of a domain is not unique to individual proteins, but instead can be found and adopted by a variety of different sequences. Domains are well-annotated in databases such as CATH [1, 2], ECOD [3], Pfam [4] and SCOP [5], which leverage sequence, structure, function and their evolutionary relationships to provide a comprehensive hierarchical classification of fold space, each with different levels of granularity.

A long-standing challenge in structural biology is the problem of domain segmentation, or more precisely, how to divide protein structures into their constituent domains. Wetlaufer envisioned the splitting of proteins into domains as early as in 1973 [6], but even the denominations of what constitutes a domain is contested by different classification databases. The structure of Ephrin type-A receptor 2 (PDB 5NFK) for example, is classified in CATH as a two-domain protein (superfamilies 3.30.200.20 and 1.10.150.10), but in ECOD as a single domain (ECOD 206.1) [7]. The difference in assignment is due to ECOD preserving an active site formed between the N and C-terminal lobes, while CATH bases its assignment on the internal structures of the two (sub)domains.

Early segmentation methods such as PUU [8], DOMAK [9] and DETECTIVE [10] published in the 1990s relied on the proposition that domains have a high intra- to inter-domain contact ratio, and directly applied this principle to each protein structure to identify domains. Newer methods such as those used in automatic domain classification by CATH, ECOD and SCOP, instead capitalise on the extensive annotations already conducted and use existing classifications to seed and find similar domains in query structures based on various criteria [7]. CATH for example, uses CATHEDRAL

which clusters new structures to already assigned domains by drawing similarities between the secondary structure components in the protein core, using a graph theory-based algorithm [11].

Broadly, methods that identify domains can be divided into two groups based on how segmentation is conducted. PUU, DOMAK and DETECTIVE, as well as the more recent DeepDom [12], DistDom [13] and FUPred [14], all conduct segmentation in a top-down fashion, in which the most likely "cut points" along the protein sequence are determined and used to partition it into domains. A key disadvantage of this regime is that discontinuous domains - those that fold in 3D space via two or more disjointed stretches of residues, are typically left over-segmented as separate domains. The dual of the task, and a more challenging one, is to instead predict the domain membership of each residue individually. The second category of domain detection methods is therefore composed of bottom-up methods such as SWORD [15] and DomBPred [16], which decompose the input protein into fragments that are then progressively aggregated into domains. The SWORD method in particular, proposes several alternative configurations as well as an optimal one, which can be reviewed by users to identify a suitable partitioning.

In computer vision, one method of detecting or segmenting objects in an image involves predicting bounding boxes around relevant regions of an image, which are called "proposals" [17]. With these proposals, the network is expected to perform a number of tasks such as classifying the object within, or segmenting the boundary of the object from a background [17, 18]. In application, a great number of bounding boxes are usually proposed, many of which will either be redundant, contain nothing of interest, or may overlap multiple objects, further complicating the task [19]. How to adapt and apply current methodologies deployed in image segmentation to the domain segmentation problem is also confounded by issues such as how to best represent discontinuous domains in one-dimensional protein sequences using the equivalent of a bounding box. Furthermore, images typically have a low foreground-to-background ratio, while the opposite is true for protein sequences in which the "background" class, i.e. non-domain residues, may only consist of several residues at either termini, or linker regions between domains. Taken together, the reasoning above suggests that the bulk of methodologies based on bounding box proposals may be incompatible with the domain segmentation problem.

In this article, we describe our method for tackling the domain segmentation problem. Our deep neural network-based method, Merizo (from the word *merismos* meaning *to divide*), conducts bottom-up domain segmentation in a proposal-free manner by using a 2-dimensional domain map directly as a learning objective. Notably, our method makes use of the Invariant Point Attention (IPA) module introduced by AlphaFold2 [20], leveraging its ability to mix together sequence, pairwise and backbone information to directly encode a protein structure into a latent representation. Merizo is trained end-to-end, and outperforms three recently published methods in terms of both the accuracy of predicting domain boundary positions, as well as matching the overall domain map. A key advantage of Merizo over other machine learning-based methods is that domain membership is predicted directly for each residue, meaning that continuous and discontinuous domains can both be predicted, albeit making the prediction task a more difficult one.

2 Methods

2.1 Domain segmentation via affinity learning

The goal of domain segmentation is to assign to each residue r_i, a label k_i, which allows residues belonging to the same domain to be grouped together via label k. A property of label k is that it is in a quotient space, that is, the exact value of index k is not important and all labels are equivalent, so long as residues belonging to the same domain share the same label [19, 21]. Forcing a domain to use a particular label increases the difficulty of the segmentation task, as similar domains may inadvertently have conflicting labels. A solution to the quotient space problem is to use an index-invariant learning objective such as via affinity learning ([19]). In affinity learning, the output probability distributions of residues belonging to the same domain are encouraged to be similar to one another when measured by a metric such as cosine similarity, and vice versa when not belonging to the same domain [21]. In the context of proteins, for each pair of residues r_i and r_j in an input protein, the output R_{ij} of the model is trained to satisfy

$$R_{ij} = \begin{cases} 1, & \text{if } r_i, r_j \text{ belong to the same domain} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

For a protein of N residues, the $N \times N$ matrix R contains assignments for every residue pair. A convenience of matrix R is that it can be used directly as a training objective; we also refer to it as the *domain map*. The similarity (i.e. affinity) between the embeddings of two residues r_i and r_j can be calculated via a measure such as cosine similarity, and when calculated for all residue pairs, produces an $N \times N$ affinity map A (where high values close to 1 signify that two residue embeddings are very similar, and low values, dissimilar). During training, the goal is to encourage affinity map A to reproduce domain map R. The loss of each residue pair can be calculated as

$$\mathcal{L}_{ij} = R_{ij}(1 - \mathcal{S}_{ij}) + (1 - R_{ij})\mathcal{S}_{ij} \tag{2}$$

$$\mathcal{S}_{ij} = \mathcal{S}_{cos}(P_i, P_j) \tag{3}$$

Where $\mathcal{S}_{ij}$ is the affinity between residues r_i and r_j; P_i and P_j are their embeddings of some hidden dimension d, and $\mathcal{S}_{cos}$ is the cosine similarity between P_i and P_j. Either of the two terms in equation 2 evaluate to zero depending on whether R_{ij} is 0 or 1 in the ground truth domain map. The final loss for a single protein can be calculated as:

$$\mathcal{L}_{affinity} = \frac{1}{N^2} \sum_i \sum_j \mathcal{L}_{ij} \tag{4}$$

A benefit of affinity loss (equation 4) is that it can be calculated from an embedding of any number of hidden dimensions, as the application of cosine similarity in equation 3 reduces the affinity between any two vectors to a single value.

2.2 Datasets

The PDB chains and domain annotations used for training our network were sourced from version 4.2 of the CATH database [2]. In order to later assess our method's ability to generalise to folds not seen before during training, we devised a training-test split which did not overlap at the CATH homologous superfamily (H) level. Splitting the dataset at the superfamily level is imperative, as homology can occur even at low sequence identities. To generate non-overlapping training and testing datasets, we constructed an adjacency matrix containing all CATH superfamilies across classes 1 to 6, with edges drawn between superfamilies when there exists at least one PDB chain containing domains from the two superfamilies (Supplementary Figure 1). The resulting graph contains 2490 components and is highly disproportionate, with the first (and largest) component containing 40.2% of all superfamilies (2295) and 76.4% of all domains (52,537; Supplementary Figure 2). The other 2389 components each consist of 19 or fewer superfamilies and represent 23.6% of all domains. Of the set of 2389, 1908 are singletons (structures containing domains from a single superfamily), and 581 are composed of multiple superfamilies.

By iterating over graph components, each can be assigned to either the training or test set, without PDB chains overlapping at the homologous superfamily level. The largest component is naturally assigned to the training set since it contains the majority of domains. Within the 581 multi-superfamily components, 20% were assigned to the test set, selected at random. This ensures that the testing set contains sufficient numbers of multi-domain chains which span multiple CATH superfamilies. To expose the network to a more diverse range of folds, most singleton components were assigned to the training set, with a small set of 12 sequestered to the testing set for later evaluating performance on single domains.

Overall, the training set contains 40,103 PDB chains composed of 66,119 domains across 5422 superfamilies. The testing set likewise contains 1418 chains and 2608 domains across 293 superfamilies. Additional summary statistics and figures are presented in Supplementary Table 1 and Supplementary Figures 1-3. During training, we opted to not use a validation set in order to maximise the number of training and testing examples.

CATH maintains a list of ambiguous domains which have not yet been assigned to any superfamily - referred to as being in the "holding pen". Such domains are unfinalised in their classification and boundary annotations, and as such are masked out during training as to not pollute the network with learning these regions as single domains or as non-domain residues.

2.3 Network architecture

Our method, called Merizo, is a small encoder-decoder network (approximately 20 million parameters) that operates directly on PDB files to predict their domain segmentation maps (Figure 1). At the core of our network is the Invariant Point Attention (IPA) encoder, which makes use of the IPA module found within the structure module of AlphaFold2 ([20]) The role of the IPA module is to facilitate information mixing between the single and pairwise information channels, while iteratively organising the backbone frames towards the ground truth structure. In our usage, we re-purpose the IPA module in an input-reversed fashion to instead read a folded structure into a latent representation. The IPA encoder in Merizo is composed of 4 non-weight-shared blocks, each with 16 attention heads and takes four inputs - three primary inputs and one additional input for positional encoding (Table 1 and Figure 1).

The first input is a 1-D "single representation", which is the primary sequence of the input protein chain, one-hot encoded into 20-residue classes and then embedded into 128 dimensions per residue using a linear layer. Prior to entering the

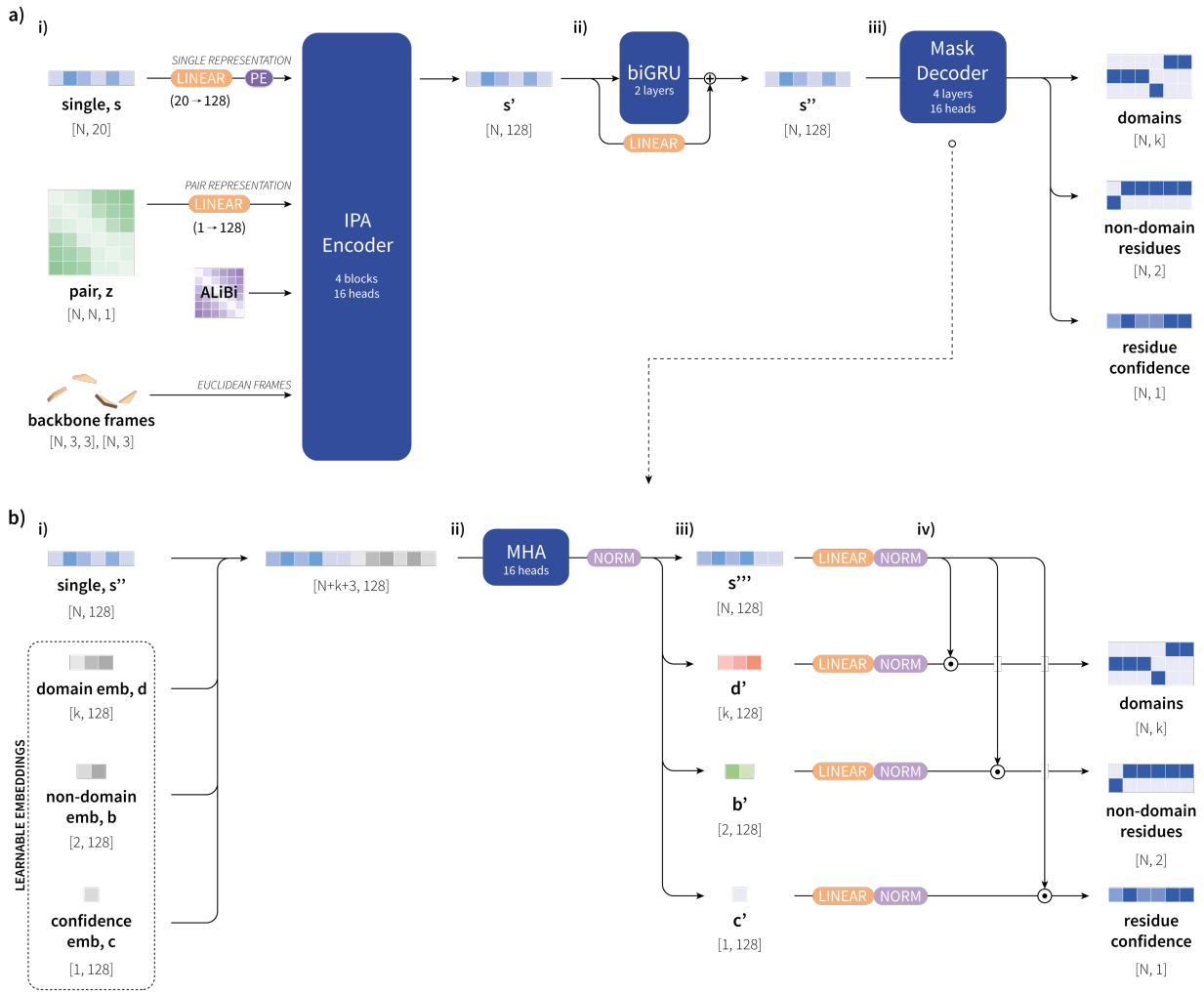

Figure 1: Overview of the Merizo network. a) Summary of the overall network architecture. **i**) Network inputs consist of single and pairwise representations and backbone frames, as well as the ALiBi matrix for positional encoding. The IPA encoder returns an updated single representation which passes through a bi-directional GRU (**ii**). **iii**) Predictions are made via a mask transformer decoder which produces per-residue domain predictions as well as masks for non-domain residues and confidence score predictions. **b**) Architecture of the masked transformer decoder. **i**) The decoder learns embeddings for domain, non-domain and confidence predictions. Learnable embeddings are concatenated with the single representation and passed through four MHA blocks (**ii**). **iii**) The combined embedding is re-divided into the original dimensions and each undergo a linear projection followed by normalisation. **iv**) Predictions are made via the dot product of the single representation with each of the learned domain, non-domain and confidence embeddings to produce the final outputs.

Table 1: Network Inputs

Input channel	Feature channel	Dimension	Description
Single representation, s	Protein sequence	$[N, 20]$	One-hot encoded 20-class amino acids
Pair representation, z	Cα distance map	$[N, N]$	Pairwise Euclidean distances between Cα atoms
Backbone Frames (r, t)	N-Cα-C "frames"	$[N, 3, 3]$ $[N, 3]$	Orthonormal basis vectors for each residue, and translation vectors to move each frame from the origin to Cα.
Residue indices	PDB residue index	$[N]$	Residue index from PDB (for positional encoding)

IPA module, sinusoidal positional encoding ([22]) is additively applied to the single representation. Throughout the network, we use a hidden dimension of 128 to represent the single representation. The second input is the 2-D "pairwise representation", where we use the Cα-Cα distance map of the input protein, embedded into 128 dimensions. The third input is the 3-D Gram-Schmidt backbone frames calculated from the N-Cα-C "frame" of each residue. Finally, residue indices (i.e. the original residue numbering from the PDB) are used for relative positional encoding via ALiBi ([23]). The ALiBi positional encoding scheme applies a penalty matrix to the pre-softmax query-key scores within each attention head of each IPA block, with penalties scaled according to the sequence separation between the two residues. The further apart two residues are in the sequence, the greater the penalty applied.

The output of the IPA module is an updated single representation which has been conditioned with information from the 2-D and 3-D feature channels. The single representation next enters a two-layer bi-directional gated recurrent unit (GRU) which processes the residue embeddings sequentially (in forward and backward directions), and acts as a bottleneck in the network. Inclusion of the GRU led to cleaner segmentation maps with less noisy predictions (e.g. where single residues were assigned incorrectly to single domains), as visible in Supplementary Figure 5.

Finally, the GRU output is decoded by a masked transformer decoder, modified from the Segmenter model ([24]; Figure 1b). The decoder jointly predicts three outputs: (a) domain predictions of shape $[N, k]$ which are the probability distributions of assigning each residue to one of k classes; (b) predictions of non-domain residues of shape $[N, 2]$, which correspond to residues in the PDB chain that do not belong to a CATH domain (typically terminal and inter-domain residues); and (c) residue confidence predictions of shape $[N, 1]$, which are values between 0-1 that predict whether a residue's domain assignment is correct (where scores close to 1 represent high confidence of a correct assignment, and 0 low confidence). To make these predictions, learnable embeddings of dimensions $[k, 128]$, $[2, 128]$ and $[1, 128]$ (for domain, non-domain and confidence predictions respectively), are concatenated together with the single representation tensor and passed through 4 standard multi-head attention (MHA) blocks with 16 heads each (Figure 1b-i). The input and output of the MHA blocks are a $[N + k + 3, 128]$ tensor which is re-divided to recover the updated single, domain, non-domain and confidence embeddings (Figure 1b-ii). Domain, non-domain and confidence predictions are made by taking the dot product between the single representation of dimension $[N, 128]$ with each of the three conditioned embeddings (Figure 1b-iv) to generate $[N, k]$ domain predictions, $[N, 2]$ non-domain residue predictions and $[N, 1]$ confidence predictions, respectively.

To make the final domain assignments, the argmax of both the $[N, k]$ domain predictions as well as the $[N, 2]$ non-domain residue tensors are taken to generate two index arrays of length N. The non-domain residue array is then applied as a mask to the domain index array to mask out positions predicted as not belonging to any domain.

2.4 Training procedures

Our method is trained fully end-to-end in PyTorch, with all features calculated directly from PDB files. Training was conducted in two phases: initial and fine tuning. Initial training was carried out for 200 steps using the Rectified Adam (RAdam) optimiser with a learning rate of 1e-3, and each target chain was randomly cropped to a window of 256 residues. During fine tuning, we gradually decreased the learning rate to 1e-5 and increased the crop size to 768 residues. We additionally augmented the dataset by masking out half the domains in a target (selected at random) during training, with a probability of 0.5. A minibatch size of 1 was used throughout, and gradients were accumulated and back-propagated every 4 minibatches. All training was conducted on either one or six NVIDIA GTX 1080Ti GPUs.

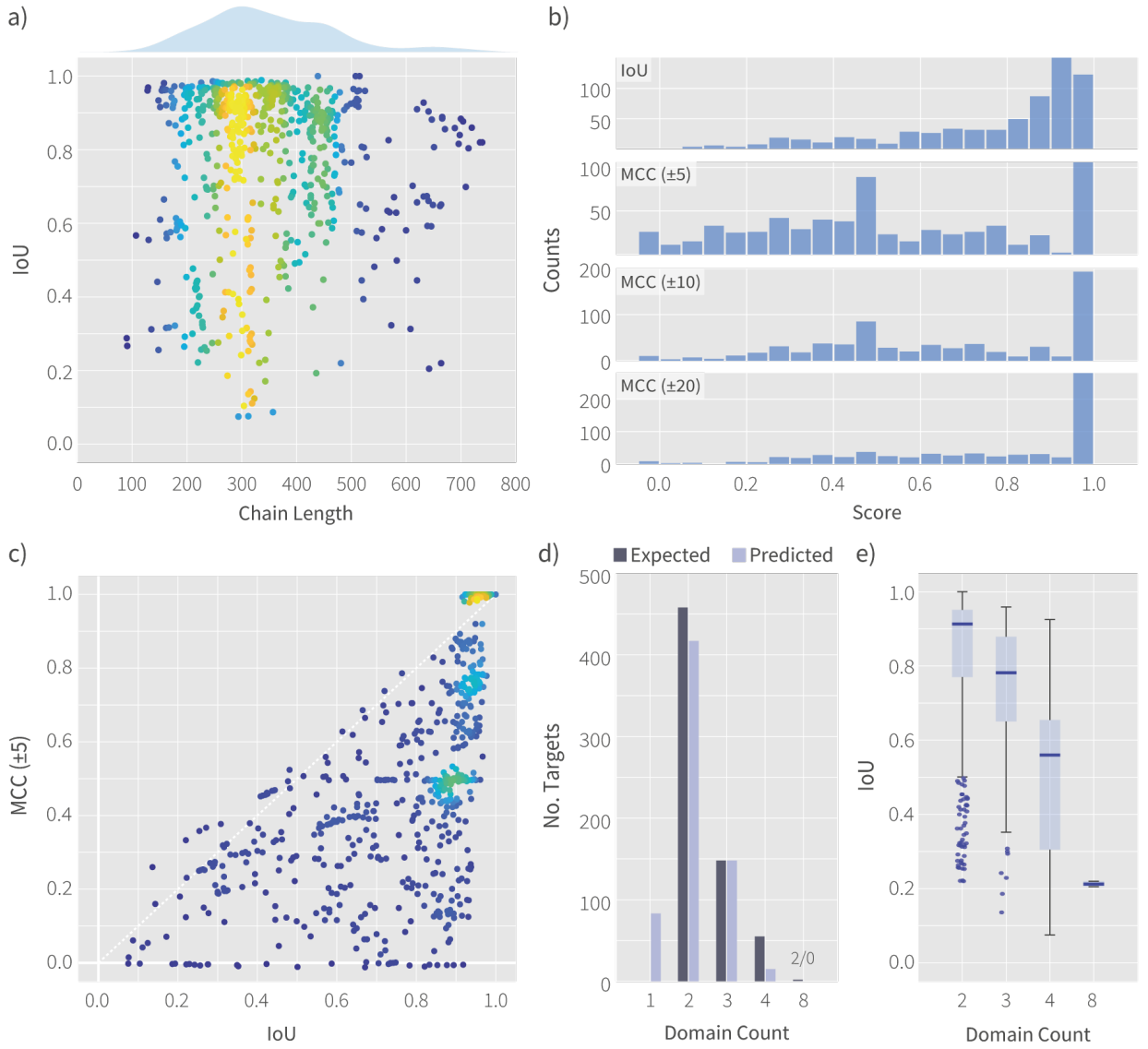

Figure 2: Performance on 663 multi-domain proteins. a) IoU scores achieved across target chain lengths. Colour gradient on data points represents the normalised density across both axes, with yellow and navy as high and low respectively. **b**) IoU and MCC at ±5, 10 and 20 distributions across multi-domain targets. Each bin represents a score width of 0.05. **c**) Comparison of IoU and MCC (±5) scores for each target. **d**) Expected (dark) and predicted (light) domain counts for multi-domain targets. **e**) IoU score achieved for each domain count bin. Blue bars mark the median of the distribution. Outliers are defined as data points with IoU outside 1.5xIQR.

3 Results

3.1 Performance of Merizo on multi-domain proteins

When evaluating the performance of a segmentation result, it is important to assess the accuracy of the overall predicted domain topology, as well as the exact position of the predicted boundaries between domains. A high score should reflect both global and positional accuracy, as well as correctly coupled indices for discontinuous segments. To this end, we assess our method's predictions using two scores: the Matthew's correlation coefficient (MCC) and the Intersect-over-Union (IoU). Both scores are calculated per-domain and reported as averages, weighted by domain lengths. The MCC score measures the precision of boundary predictions in matching the CATH ground truth boundaries, and is assessed with tolerances of ±5, 10 or 20 residues. The IoU on the other hand measures the accuracy between the

ground truth and predicted domain maps and complements the MCC by providing a global measure of similarity that is less sensitive to exact boundary positions.

The set of PDB chains used for testing the performance of Merizo consists of 663 multi-domain chains, none of which contain a domain that has been seen in any training example. Testing chain lengths range from 90-739 residues, with an average of 347. On this test set, Merizo achieves a median IoU of 0.870, and MCC (± 5) of 0.496, which increases to 0.630 at ± 10 and 0.849 at ± 20 residues of the true boundary (Figure 2a). High IoU scores (>0.8) achieved for the longest targets indicate that Merizo is capable of predicting accurate domain maps for edge cases. The distribution of IoU scores is unimodal, peaking at 0.85-0.90, and demonstrates that domain maps are on average predicted with high accuracy (Figure 2b). On the other hand, the MCC (± 5) distribution is bimodal, peaking at 0.5 and 1.0. The peak at 0.45-0.50 can be attributed to two-domain proteins where the inter-domain boundary is incorrectly positioned (exceeding ± 5 residues), leading to an MCC of approximately 0.5. As the tolerance increases to ± 10 and 20 residues, the distribution flattens towards a unimodal distribution peaking at the top bin (Figure 2b). Plotting IoU against MCC (± 5) demonstrates the utility of using both scores (Figure 2c). Predictions with low MCC but high IoU reflect cases where the predicted domain maps are overall very similar to the ground truth, but the predicted boundaries are too far from the CATH ground truth labels, which may themselves be imprecise. Examples of high accuracy, low precision cases are shown in Supplementary Figure 6.

Within the 663 test set targets, the majority are two-domain proteins (approximately 70%), with exponentially decreasing numbers as the number of domains increases to eight, where there are only two such targets (Figure 2d). The predicted domain count distribution across the 663 targets closely matches the expected distribution, with the exception of about 12% of the targets being predicted as single-domain chains. Examples of well-predicted targets with 3 to 4 domains are shown in Figure 3. Notably, Merizo was unable to correctly predict boundaries for the two eight-domain chains 1avcA and 1m9iA, which were instead predicted as three-domain proteins. In general, Merizo achieves lower IoU scores as the number of domains increases (Figure 2e), which is perhaps unsurprising given that higher-domain count structures are less represented in the training set.

3.2 Non-domain residue predictions and network confidence

Unlike most other segmentation methods, in addition to domain predictions, the decoder of Merizo jointly predicts non-domain residues as well as per-residue confidence scores. Non-domain residues are those which are not attributed to any CATH domain and are typically terminal residues or linkers between domains. Merizo uses predictions of non-domain residues to mask out positions in the final domain predictions (Section 2.3). Supplementary Figure 7 shows examples of output domain maps before and after masking non-domain residue predictions, leading to improved IoU scores.

Per-residue confidence estimates output from Merizo can also be useful in evaluating the network confidence in local assignments. Examples of confidence plots generated by the network are shown in Supplementary Figure 7iii, and showcase examples where areas of low confidence correspond well with non-domain residues which have been incorrectly assigned to a domain in the raw network output.

3.3 Benchmark against existing methods

To evaluate the performance of Merizo against existing domain segmentation methods, we compared the quality of its predictions against those produced by several methods. The first two, DeepDom ([12]) and Eguchi-CNN ([25]), are ML-based segmentation methods that operate on sequence and distance map inputs, respectively. The third method is SWORD ([15]), a non-ML method that operates on protein structures directly and makes domain predictions via hierarchical clustering of protein units ([26]). As a baseline, we also compare against three random-assignment methods (prefixed with 'Random'). In random assignments, we first guess the domain count g following the Domain Guess by Size method ([27]). The Random equal and unequal methods divide the target into g domains, which are either equally or randomly sized, whereas in Random assigned, each residue is randomly assigned to one of g indices.

The results of the comparison are shown in Figure 4. In subpanel (a), the MCC and IoU scores obtained by each method are shown as box plots. Merizo obtains the highest scores out of the methods tested (IoU median of 0.87). Assigning domains based on guessing the domain count by size, followed by equal partitioning of the structure in Random equal, obtains reasonable scores compared to the other two Random methods, with a median IoU on par with ML-based DeepDom and Eguchi-CNN (approximately 0.4). The results from DeepDom (1D inputs), Eguchi-CNN (2D inputs), SWORD (3D inputs) and Merizo (1D, 2D and 3D inputs) highlight a trend that greater performance is achieved by methods that make use of inputs with richer sequence and structural information.

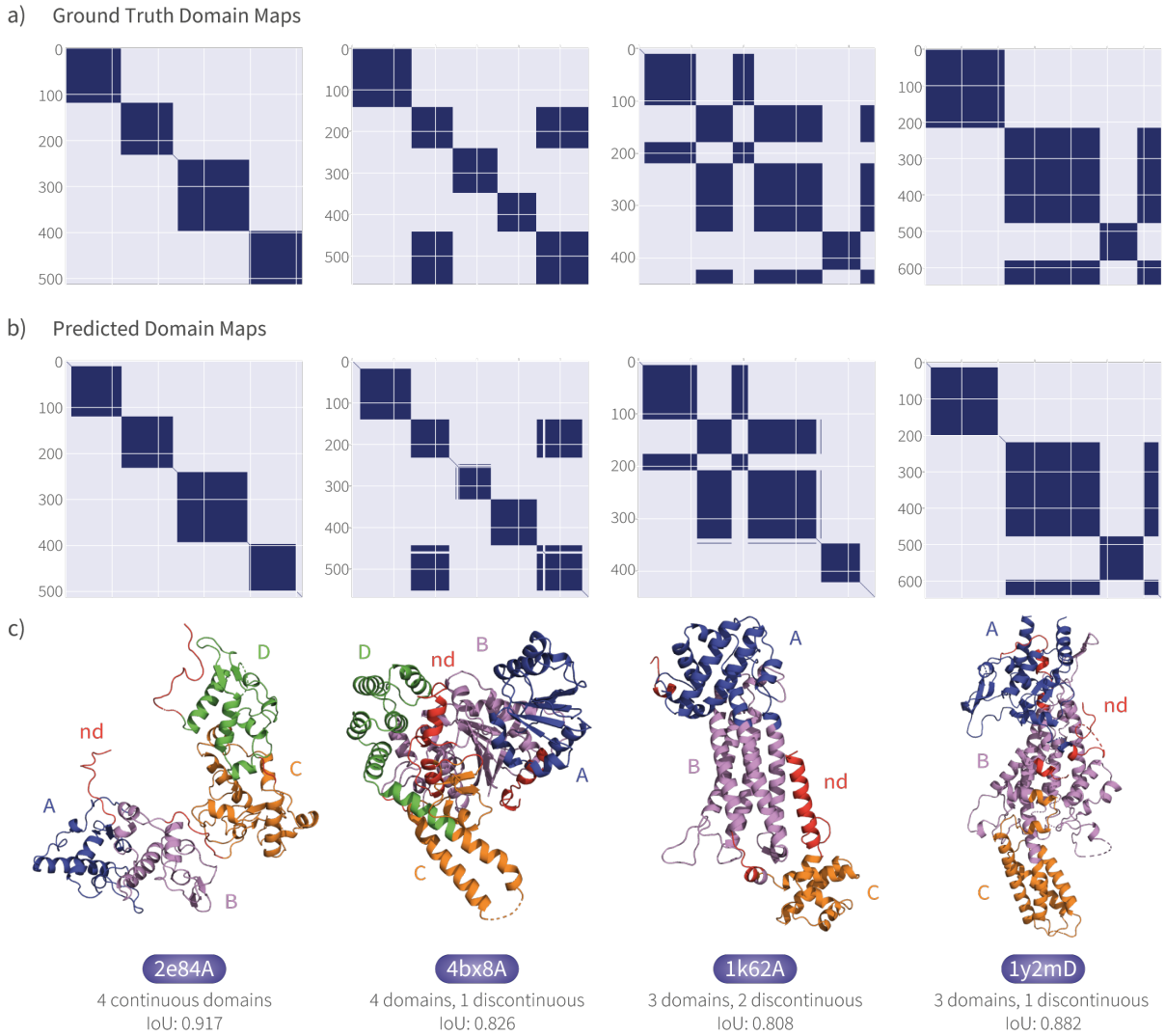

Figure 3: Examples of well-segmented targets. a) Ground truth and **b**) predicted domain maps for four multi-domain targets. Dark regions represent stretches of residues belonging to the same domain. **c**) PDB structure of example targets. Colours represent domains assigned in **b** and are in sequential order. Non-domain residues (e.g. terminal loops and inter-domain linker regions) are coloured in red. The number of domains as well as IoU scores for each target are shown under each structure.

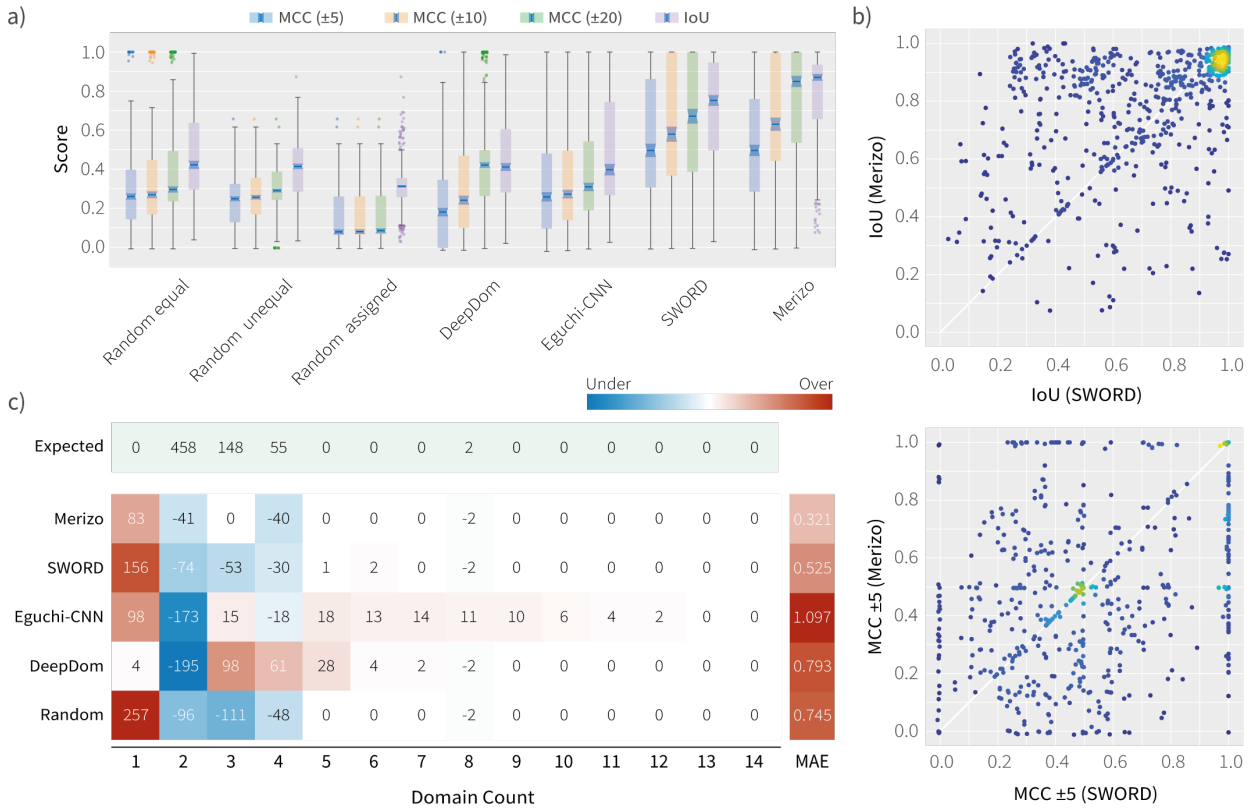

Figure 4: Comparison of Merizo with other methods on multi-domain proteins. a) Comparison of IoU and MCC scores achieved by each method. The distribution median is denoted by the blue bar. Box notches represent 95% confidence intervals on medians. Outliers are defined as data points outside 1.5xIQR. **b**) Comparison of IoU and MCC (± 5) between SWORD and Merizo. Colour gradient on data points represents the normalised density across both axes, with yellow and navy as high and low respectively. **c**) Heatmap showing (top) the expected domain count distribution and (bottom rows) deviation from expected domain counts for each method. Under and over-predicted domain counts are shown in blue and red respectively.

Comparing the per-target performance of the top two methods, Merizo and SWORD (Figure 4b), we find that although Merizo outperforms SWORD on the majority of targets on IoU, both methods obtain a similar median MCC (± 5) of approximately 0.5. However, as the boundary tolerance increases to ± 10 and 20, the gap between Merizo and SWORD widens (0.849 and 0.671 respectively at ± 20), indicating that Merizo is mostly limited by small discrepancies in the predicted boundary positions, rather than incorrectly predicting the topology of the overall domain map. It is also worth noting that SWORD outputs several different segmentation proposals, and the most CATH-like assignment may be hidden as an alternative assignment. The spread of data points in the MCC comparison between Merizo and SWORD also suggests that each method performs well on a different subset of targets.

3.4 Performance on over and under-segmentation

Another important facet of domain segmentation is correctly recognising the number of domains within a target. Figure 4c shows the difference in predicted domain counts for each method, compared to the expected ground truth distribution. Values in blue and red indicate whether a method has under- or over-estimated the number of chains in a particular domain count bin. Most methods, with the exception of DeepDom, incorrectly assess a proportion of multi-domain proteins as single-domain chains, while conversely under-predict in the two-domain category. Most notably, Eguchi-CNN over-predicts in high-domain-count bins (>4), producing a long tail of predictions that reach up to 12 domains.

Interestingly, both Merizo and SWORD were unable to correctly segment the two eight-domain chains 1avcA and 1m9iA, both of which are formed of eight annexin domains, packed together into two sets of four, separated by a linker between them (Supplementary Figure 8). Merizo predicts these chains as consisting of three domains, with one set

of four annexins predicted as one domain, and the other four as two domains. SWORD on the other hand, correctly predicts the eight annexins as individual domains, but incorrectly couples domains 1 and 4, and 5 and 8 as discontinuous domains. Additionally, Merizo partially predicts a linker between the two sets of four domains, while SWORD assigns the entire linker to one domain (Supplementary Figure 8).

Deviations between the predicted domain count and the expected count can be expressed using mean absolute error (MAE), which describes on average, how many more or fewer domains are predicted compared to the ground truth across the test set. On average, domain count predictions by Merizo are 0.321 domains from the expected ground truth value, with the next best method being SWORD with an MAE of 0.525. Eguchi-CNN performs the worst, on average over- or under-predicting a whole domain, with an MAE of 1.097. Guessing the number of domains using chain length information obtains comparable results to DeepDom, which obtains an MAE of approximately 0.7-0.8. Overall, these results indicate that although Merizo typically tends to under-predict the number of domains within a target, the segmentation maps produced are of high quality in terms of overall likeness to the ground truth (high IoU), but may deviate a little when it comes to the exact predicted boundaries.

Table 2: Comparison of runtimes on 663 test set targets

Method	Hardware	Average time per target (s)	Total runtime (s)	Relative
Merizo	GPU	0.112	74.32	1.00
	CPU	1.095	725.77	9.77
SWORD	CPU	9.602	6366.00	85.65
DeepDom	GPU	0.020	13.29	0.18
	CPU	0.055	36.69	0.49
Eguchi-CNN	GPU[a]	-	-	-
	CPU	4.475	2966.77	39.92

[a] No GPU option provided

3.5 Benchmark on AlphaFold2 Models

[To be added]

4 Discussion and conclusions

In this study, we have developed a fast and lightweight domain parser capable of handling discontinuous domain boundaries as well as non-domain residues which are often overlooked. The potential to rapidly decompose protein structures into constituent domains, and in an evolution-coherent manner, presents several advantages including the ability to tackle the staggering number of models that have recently been made available by the AlphaFold Protein Structure Database (200 million) [28] and models of metagenomic sequences in the ESM Metagenomic Atlas (617 million) [29]. Merizo on average takes 0.1s to process a single structure (Table 2), meaning that it would take just over 10 days on 100 GPUs to process 800 million models. Merizo runtimes also compare favourably with other methods such as SWORD, when accuracy is considered (Table 2 and Figure 4).

More recently, developments in single-sequence and language model-based prediction methods [29, 30, 31] have also been accompanied by faster run times, which will additionally boost the rate at which models will be made available. As these methods continue to improve in predictive accuracy, models such as those from the ESM Metagenomic Atlas will likely be improved upon and re-released, making it a new requirement that any downstream analysis, such as domain segmentation or function prediction, must regularly re-process large numbers of models.

Furthermore, classification schemes such as CATH, which we used for our ground truth labels, would benefit from having domains pre-parsed from these large model databases in order to facilitate rapid and high-throughput sorting into families. Although we have based our segmentation predictions on CATH, we recognise that other databases such as ECOD or SCOP could have been used. However, as other studies have pointed out, domain assignments for the same protein are not necessarily agreed upon between different schemes, and classification by function, secondary structure or spatial separation may give different, but equally valid assignments [15]. In the context of machine learning, it may be advantageous to base segmentation labels used for training on a single database in order to avoid inadvertently introducing conflicting ground truths.

Fast and accurate segmentation methods could also play a role in determining the domain arrangement of newly discovered folds and structures, which is especially applicable to exercises such as the Critical Assessment of Structure

Prediction (CASP). In CASP, tools such as Merizo could be used by the organisers to determine the domain boundaries of prediction targets, particularly in the free-modelling category, in which targets have no known homologues in the Protein Data Bank. Supplementary Figure 9 shows three multi-domain targets from the CASP15 exercise which we predicted the domain boundaries for. Two of these targets, T1170 and T1121, are annotated by the CASP organisers as two-domain proteins, however are segmented into three plausible domains by Merizo. It is interesting to speculate how the prediction performance of some participating groups may have changed depending on the domain definitions used for assessment.

A limiting factor for our study was the amount of training data available, specifically the number of multi-domain protein chains that were labelled by CATH (19,067 chains). The majority of these models were comprised of 2-3 domains and it is perhaps unsurprising that the performance of Merizo suffered on higher domain count targets. One way to remedy this may be to augment the data set with either synthetic multi-domain proteins (modelled for example, by AlphaFold2), or through self-knowledge distillation using models provided by large structure databases. Doing so will allow the network to encounter combinations of domains which may not be represented in the training set, as well as to add structural variation to the domains that are familiar to the network, enhancing its ability to generalise.

References

[1] Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

[2] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla SM Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, et al. Cath: increased structural coverage of functional space. *Nucleic acids research*, 49(D1):D266–D273, 2021.

[3] Hua Cheng, R Dustin Schaeffer, Yuxing Liao, Lisa N Kinch, Jimin Pei, Shuoyong Shi, Bong-Hyun Kim, and Nick V Grishin. Ecod: an evolutionary classification of protein domains. *PLoS computational biology*, 10(12):e1003926, 2014.

[4] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik LL Sonnhammer, Silvio CE Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, et al. Pfam: The protein families database in 2021. *Nucleic acids research*, 49(D1):D412–D419, 2021.

[5] Antonina Andreeva, Eugene Kulesha, Julian Gough, and Alexey G Murzin. The scop database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic acids research*, 48(D1):D376–D382, 2020.

[6] Donald B Wetlaufer. Nucleation, rapid folding, and globular intrachain regions in proteins. *Proceedings of the National Academy of Sciences*, 70(3):697–701, 1973.

[7] R Dustin Schaeffer, Lisa N Kinch, Jimin Pei, Kirill E Medvedev, and Nick V Grishin. Completeness and consistency in structural domain classifications. *ACS omega*, 6(24):15698–15707, 2021.

[8] Liisa Holm and Chris Sander. Parser for protein folding units. *Proteins: Structure, Function, and Bioinformatics*, 19(3):256–268, 1994.

[9] Asim S Siddiqui and Geoffrey J Barton. Continuous and discontinuous domains: an algorithm for the automatic generation of reliable protein domain definitions. *Protein Science*, 4(5):872–884, 1995.

[10] Mark B Swindells. A procedure for detecting structural domains in proteins. *Protein Science*, 4(1):103–112, 1995.

[11] Oliver C Redfern, Andrew Harrison, Tim Dallman, Frances M G Pearl, and Christine A Orengo. Cathedral: a fast and effective algorithm to predict folds and domain boundaries from multidomain protein structures. *PLoS computational biology*, 3(11):e232, 2007.

[12] Yuexu Jiang, Duolin Wang, and Dong Xu. Deepdom: Predicting protein domain boundary from sequence alone using stacked bidirectional lstm. In *BIOCOMPUTING 2019: Proceedings of the Pacific Symposium*, pages 66–75. World Scientific, 2018.

[13] Sajid Mahmud, Zhiye Guo, Farhan Quadir, Jian Liu, and Jianlin Cheng. Multi-head attention-based u-nets for predicting protein domain boundaries using 1d sequence features and 2d distance maps. *bioRxiv*, 2022.

[14] Wei Zheng, Xiaogen Zhou, Qiqige Wuyun, Robin Pearce, Yang Li, and Yang Zhang. Fupred: detecting protein domains through deep-learning-based contact map prediction. *Bioinformatics*, 36(12):3749–3757, 2020.

[15] Guillaume Postic, Yassine Ghouzam, Romain Chebrek, and Jean-Christophe Gelly. An ambiguity principle for assigning protein structural domains. *Science Advances*, 3(1):e1600552, 2017.

[16] Zhong-ze Yu, Chun-xiang Peng, Jun Liu, Biao Zhang, Xiao-gen Zhou, and Gui-jun Zhang. Dombpred: protein domain boundary prediction based on domain-residue clustering using inter-residue distance. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2022.

[17] Huiyan Zhang, Hao Sun, Wengang Ao, and Georgi Dimirovski. A survey on instance segmentation: Recent advances and challenges. *International Journal of Innovative Computing, Information and Control*, 17(3):1041–53, 2021.

[18] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[19] Long Jin, Zeyu Chen, and Zhuowen Tu. Object detection free instance segmentation with labeling transformations. *CoRR*, abs/1611.08991, 2016.

[20] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[21] Yen-Chang Hsu, Zheng Xu, Zsolt Kira, and Jiawei Huang. Learning to cluster for proposal-free instance segmentation. *CoRR*, abs/1803.06459, 2018.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[23] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

[24] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[25] Raphael R Eguchi and Po-Ssu Huang. Multi-scale structural analysis of proteins by deep semantic segmentation. *Bioinformatics*, 36(6):1740–1749, 2020.

[26] Jean-Christophe Gelly, Alexandre G De Brevern, and Serge Hazout. 'protein peeling': an approach for splitting a 3d protein structure into compact fragments. *Bioinformatics*, 22(2):129–133, 2006.

[27] Sarah J Wheelan, Aron Marchler-Bauer, and Stephen H Bryant. Domain size distributions can predict domain boundaries. *Bioinformatics*, 16(7):613–618, 2000.

[28] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

[29] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, pages 2022–07, 2022.

[30] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022.

[31] Konstantin Weissenow, Michael Heinzinger, Martin Steinegger, and Burkhard Rost. Ultra-fast protein structure prediction to capture effects of sequence variation in mutation movies. *bioRxiv*, pages 2022–11, 2022.