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

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

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

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

May 11, 2023

ABSTRACT

The AlphaFold Protein Structure Database (AFDB), containing predictions for over 200 million proteins, has been met with enthusiasm over its potential in enriching structural biological research and beyond. Currently, access to the information within the AFDB is precluded by an urgent need of tools that allow the efficient traversal, discovery and documentation of its contents. Identifying the regions that correspond to domains in the AFDB is a non-trivial endeavour and doing so will aid our understanding of protein structure and function, while facilitating drug discovery and comparative genomics. Here, we describe a deep learning method for accurate domain segmentation called Merizo, which learns to cluster residues into domains in a bottom-up manner. Merizo is trained on CATH domains and fine-tuned on a subset of AFDB models via self-distillation, enabling it to be applicable to both experimental and AFDB models. As proof of concept, we apply Merizo to models of the human proteome, and identify 40,818 putative domains that can be matched to CATH representative domains. Merizo is available at `https://github.com/psipred/Merizo`.

Keywords Protein domain segmentation · deep learning · invariant point attention · AlphaFold Protein Structure Database · CATH

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 can be found and adopted by a variety of different sequences. Structural 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 1973,[6] but even the denominations of what constitutes a domain are 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 the 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 its 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 detecting 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 UniDoc,[15] SWORD[16] and DomBPred,[17] which decompose the input protein into fragments that are then progressively aggregated into domains. SWORD, in particular, proposes several alternative configurations as well as an optimal one, which can be reviewed by users to identify a suitable partitioning.

The AlphaFold Protein Structure Database (AFDB) contains predicted models for over 200 million protein sequences and constitutes a valuable expansion of protein space.[18] An obvious use case for domain segmentation is in the high-throughput identification of domains from the AFDB, facilitating their sorting into structure databases such as CATH. Compared to experimentally-derived PDB structures, AFDB models may exhibit less optimal packing or folding, particularly for rare folds with limited known sequence homology. Furthermore, unlike experimental structures, the generation of *in silico* models is not constrained by the same factors (such as crystallisation success, quality, etc.), enabling the modelling of the entire sequences, including previously difficult-to-resolve regions. As such, many AFDB models also feature long stretches of unstructured regions that may hinder the performance of methods that are not prepared to operate on such models. More recently, methods such as DPAM have been developed to specifically operate on AFDB models, leveraging a combination of inter-residue distances, structural similarity to ECOD domains and the predicted aligned error (PAE) map produced by AlphaFold2 to inform domain assignment.[19]

Our approach to the domain segmentation problem, called Merizo, is based on a deep neural network which conducts bottom-up domain assignment by learning to directly cluster residues into domains, based on a combination of its sequence and structure. Notably, our method makes use of the Invariant Point Attention (IPA) module introduced in AlphaFold2,[20] leveraging its ability to mix together sequence and coordinate information to directly encode a protein structure into a latent representation (Figure 1). Residue embeddings are clustered together by using an affinity learning approach whereby the ground truth domain map is used directly as an objective. Residues that are part of the same domain are encouraged towards the same embedding while those that are not, are encouraged to have different embeddings. Merizo is trained on CATH domain annotations and fine-tuned on a subset of AFDB models using a self-knowledge distillation approach, allowing the network to be equally applied to experimental structures as well as those generated by AlphaFold2. Furthermore, we show how fast and accurate methods such as Merizo can be applied to the human genome, identifying 40,818 putative domains that can be matched to existing structures in CATH at various levels of homology.

2 Results

2.1 Benchmark against existing state-of-the-art methods

Merizo was trained on 17,287 multi-domain proteins with annotations sourced from CATH 4.2, and another 663 chains were held out to be used as a testing set (referred to as CATH-663). Targets in CATH-663 do not share any domains from the same homologous superfamily as the training set in order to better gauge performance on folds that the network has not seen before. How well a predicted assignment agrees with the ground truth can be quantified via a number of different measures. Here, we score predictions based on: 1) how well the residues in a predicted domain overlap with a true domain, measured via the intersect-over-union (IoU) between residues in the predicted and ground truth domain, and 2) how precise the predicted domain boundaries are, when assessed using Matthew's correlation coefficient (MCC). The MCC describes the correlation between the predicted and ground truth boundary positions, and a boundary is deemed correct if it is predicted within $\pm m$ residues of a ground truth domain boundary (where m is evaluated at tolerances of 5, 10 and 20 residues). Both scores are calculated at the domain level, and we report the domain length-weighted average for each target.

Our benchmark compares the accuracy of domain assignments by Merizo against those produced by four recently published methods including DeepDom,[12] a CNN-based method from Eguchi et al.[21] (referred to as Eguchi-CNN), SWORD[16] and UniDoc.[15] Both DeepDom and Eguchi-CNN are machine learning (ML)-based methods and operate on primary sequence and distance map inputs respectively. In contrast, SWORD and UniDoc are non-ML-based and conduct segmentation on coordinates in a bottom-up fashion by clustering low-level structural elements into domains, in a manner similar to Merizo. In addition to the four published methods, we include four baseline measures, including scoring ECOD assignments against CATH (where ECOD assignments are treated as a prediction result), and three random assignment methods prefixed with 'Random', where the domain count is estimated according to the Domain Guess by Size method.[22] Targets are then divided into either equally or unequally sized segments ('Random equal/unequal') or each residue is assigned into a domain at random ('Random assigned').

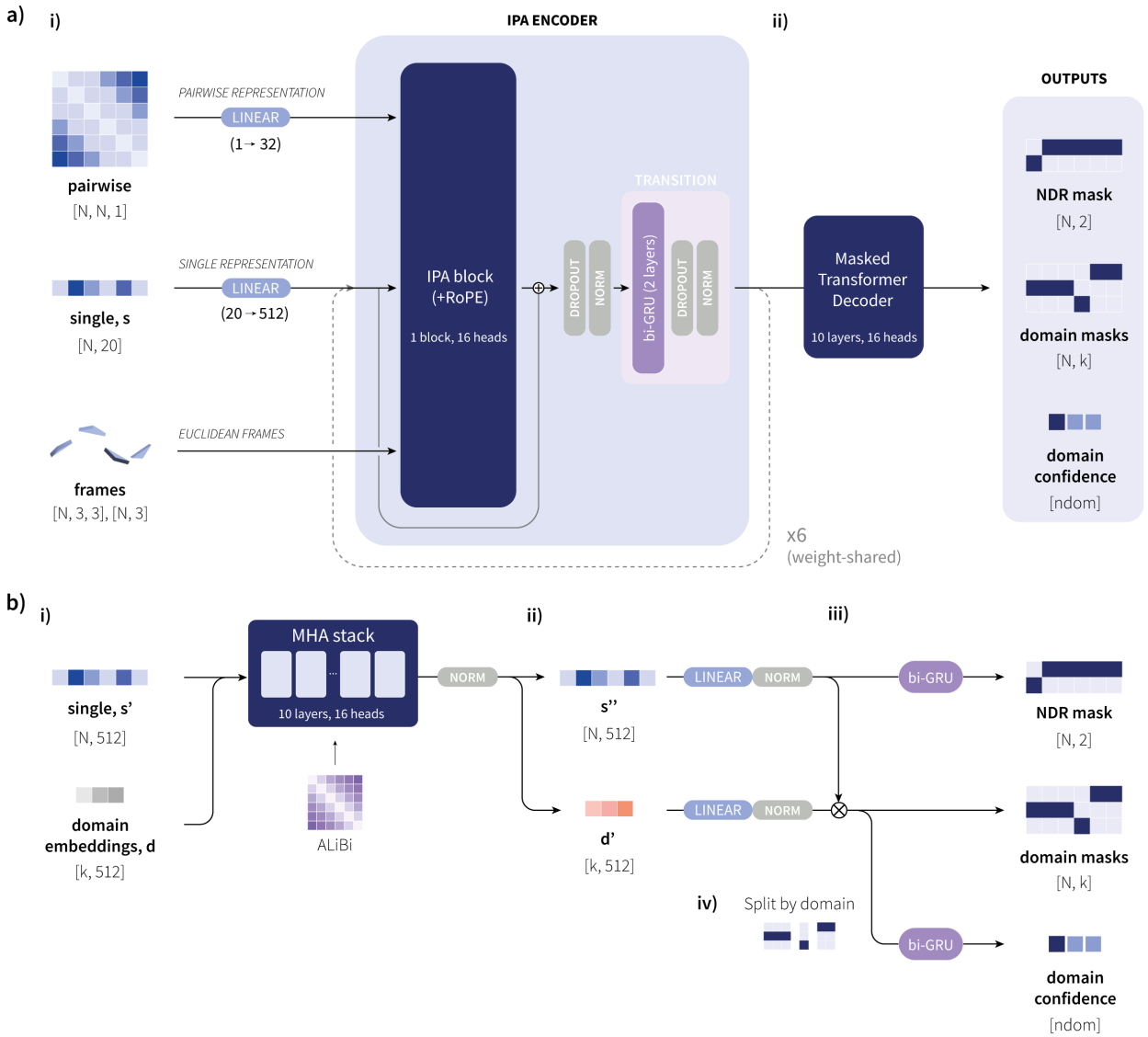

Figure 1: Overview of the Merizo network. a) Summary of the network architecture. **i**) Network inputs to the IPA encoder are the single and pairwise representations and backbone frames in the style of AlphaFold2. The IPA encoder comprises six weight-shared blocks, each containing a single IPA block with RoPE positional encoding, and a bi-GRU transition block. N denotes the residue count in a given target, and k denotes the maximum number of assignable classes, set to 20. **ii**) The encoder returns an updated single representation which is decoded by a masked transformer decoder. **b**) Summary of the masked transformer decoder. **i**) In the decoder, learnable domain mask embeddings d are concatenated to the single representation and passed through a 10-layer MHA stack with ALiBi positional encoding. The attention-treated output is split to recover updated single and domain mask embeddings, and each are passed through a linear layer followed by normalisation. Domain mask predictions are made via calculating the inner product between the updated single representation s'' and the conditioned domain embeddings d'. **iii**) The positions of NDRs are predicted by passing s'' through a two-layer bi-GRU followed by projection into two dimensions. **iv**) To make per-domain pIoU predictions, the predicted domain mask tensor is split according to the predicted domain and is passed through a two-layer biGRU, followed by projection into one dimension to produce a single pIoU value for each domain. $ndom$ represents the number of predicted domains.

A summary of the benchmark is shown in Figure 2. Overall, it can be seen that Merizo is the most performant on the CATH-663 set when scoring by IoU, achieving a similar median IoU to the ECOD baseline. Merizo is followed closely by UniDoc which exhibits a similar median IoU, albeit with a wider distribution. As domain assignments can change drastically depending on the classification scheme used, we further divided CATH-663 into two sets, depending on whether there is consensus between the definitions from CATH and ECOD (a consensus set with 313 targets and a dissensus set with 350 targets) (Figure 2a). Based on this split, most methods perform more strongly on the consensus set, where targets may be more obvious in their domain arrangement and are easier to both classify (for CATH and ECOD) as well as predict. The opposite is true for the dissensus set, where the gap between Merizo and UniDoc widens, indicating that where CATH and ECOD disagree on an assignment, Merizo is more likely than other methods to produce a CATH-like result.

Where Merizo does not produce a well-scoring result based on CATH, the assignment may not be wrong *per se*, but may represent an alternative assignment that the network has negotiated given its internal knowledge of domain packing. When the possibility of an alternative ground truth is considered (such as scoring Merizo against ECOD), it can be seen that Merizo does produce ECOD-like assignments for a subset of targets despite not being trained to do so (Figure 2b). Points on the scatterplot where a data point falls in the upper triangle indicate targets where Merizo's domain assignment matches that of ECOD over CATH, while the lower triangle represents the opposite. For some targets, the domain annotation of CATH may contain errors, or where the assignment was made from the culmination of other priors that our method does not have access to. Indeed, several cases were identified in the CATH-663 set where pairs of chains shared similar structures but were inconsistently parsed by CATH, leading to Merizo underperforming against the conflicting ground truth labels (Supp. Figure 3).

When CATH or ECOD are individually used as the ground truth for scoring Merizo, it is both expected and observed that Merizo is attuned to producing CATH-like assignments (Figure 2c). However, scoring Merizo dynamically to either CATH or ECOD (whichever ground truth scores highest), yields a much stronger performance in terms of both IoU and MCC scores, as well as the number of correctly predicted domains which increases to nearly 75%, from 65% when scoring by only CATH (Figure 2c).

Another important facet of domain segmentation is correctly predicting the number of domains within a given target. On this task, domain count predictions by Merizo were the most accurate, scoring a mean absolute error (MAE) of 0.332 (Figure 2d). All ML-based methods including Merizo, Eguchi-CNN and DeepDom also have a tendency to under-estimate rather than over-estimate the number of domains, whereas the opposite is true for all other methods (Figure 2d).

2.2 Fine-tuning Merizo on AlphaFold2 models

As some AFDB models may contain large stretches of unstructured regions (which we refer to as non-domain residues or NDRs), we finetuned Merizo on a subset of the AFDB human proteome (AFDB-human) to encourage the network to become performant on these models. Domains from AFDB-human have been classified in ECOD, describing 47,577 domains across 18,038 proteins.[23] Since Merizo is a CATH-specific domain segmentation method, we opted not to train on the ECOD classifications, but to instead utilise a self-knowledge distillation approach which was conducted in two stages (see Methods sections 4.1.2 and 4.4).

A comparison of before and after fine-tuning Merizo is shown in Figure 3. On the task of NDR detection, we compared the number of NDRs predicted by Merizo before and after fine-tuning, as well as to three baselines: 1) predictions by UniDoc, 2) inferring from PAE/plDDT and 3) predictions by DPAM. Results show that after fine-tuning, the ability of Merizo to detect NDRs drastically increases, and is highly correlated with NDR counts inferred from PAE/plDDT and those reported by DPAM (Figure 3a). Performance on domain prediction on the CATH-663 set changes little after fine-tuning but noticeably results in a small drop in median IoU and MCC scores, but with narrower overall distributions, suggesting that self-distillation leads to more consistent assignments (Figure 3b).

On the set of AFDB-1195, where ground truth domain assignments are unavailable, Merizo identifies a total of 3,109 domains, with 40% of these domains matching DPAM annotations (Figure 3c). Despite DPAM identifying nearly twice as many domains as Merizo, only 3,749 (73%) were later classified into ECOD domains. One possible explanation for this discrepancy is that DPAM has detected domains that are novel to ECOD and cannot be easily assigned to an ECOD class. However, a closer examination of targets where DPAM has predicted a large number of domains reveals that DPAM tends to identify domains within the unstructured regions of AFDB models (Figure 3d). The examples depicted in Figure 3d illustrate cases where DPAM has over-segmented NDRs, leading to inflated estimates of domain counts. Both SWORD and UniDoc are incapable of differentiating domains from NDRs, resulting in the inclusion of the latter in domain predictions. In models with a significant proportion of NDRs, this limitation reduces the effectiveness of both methods when applied to AFDB models, as NDRs must be addressed separately in order to accurately segment domains. Taken together, these observations suggest that utilizing a fine-tuning strategy, even in the absence of ground truth labels, can effectively improve the performance of Merizo on AFDB models.

2.3 Application of Merizo to the human proteome

As a demonstration of our method, we applied Merizo to the entire AFDB-human set containing 23,391 models

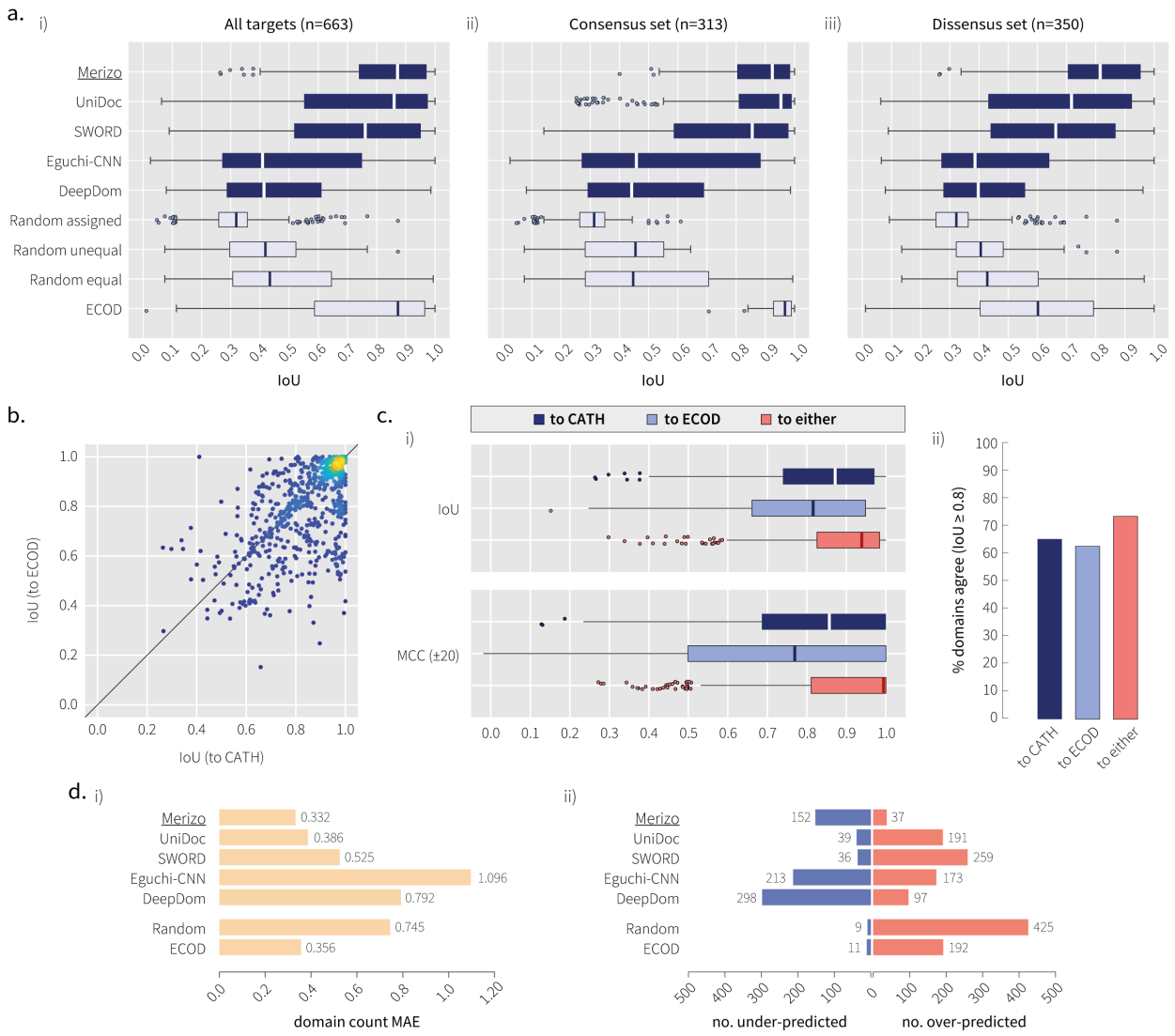

Figure 2: Benchmark against existing methods on the CATH-663 set. **a**) IoU distributions for each method for i) all CATH-663 targets, and targets where there is ii) consensus or iii) no consensus between CATH and ECOD assignments. Solid bars inside each box represent the distribution median. Outliers are defined as data points exceeding 1.5xIQR. **b** Comparison of IoU achieved when scoring Merizo against CATH or ECOD domain assignments. The colour gradient indicates the density of data points, where yellow and dark blue are high and low respectively. **c**) i) Box plots showing IoU and MCC (boundary±20) distributions and ii) the percentage of domains that agree between Merizo and either CATH, ECOD or the maximum of either. Correct domains are defined as those with an IoU of at least 0.8 to the ground truth domain. **d**) Performance of each method on i) domain count prediction, and ii) the number of under and over-predicted domains across all targets.

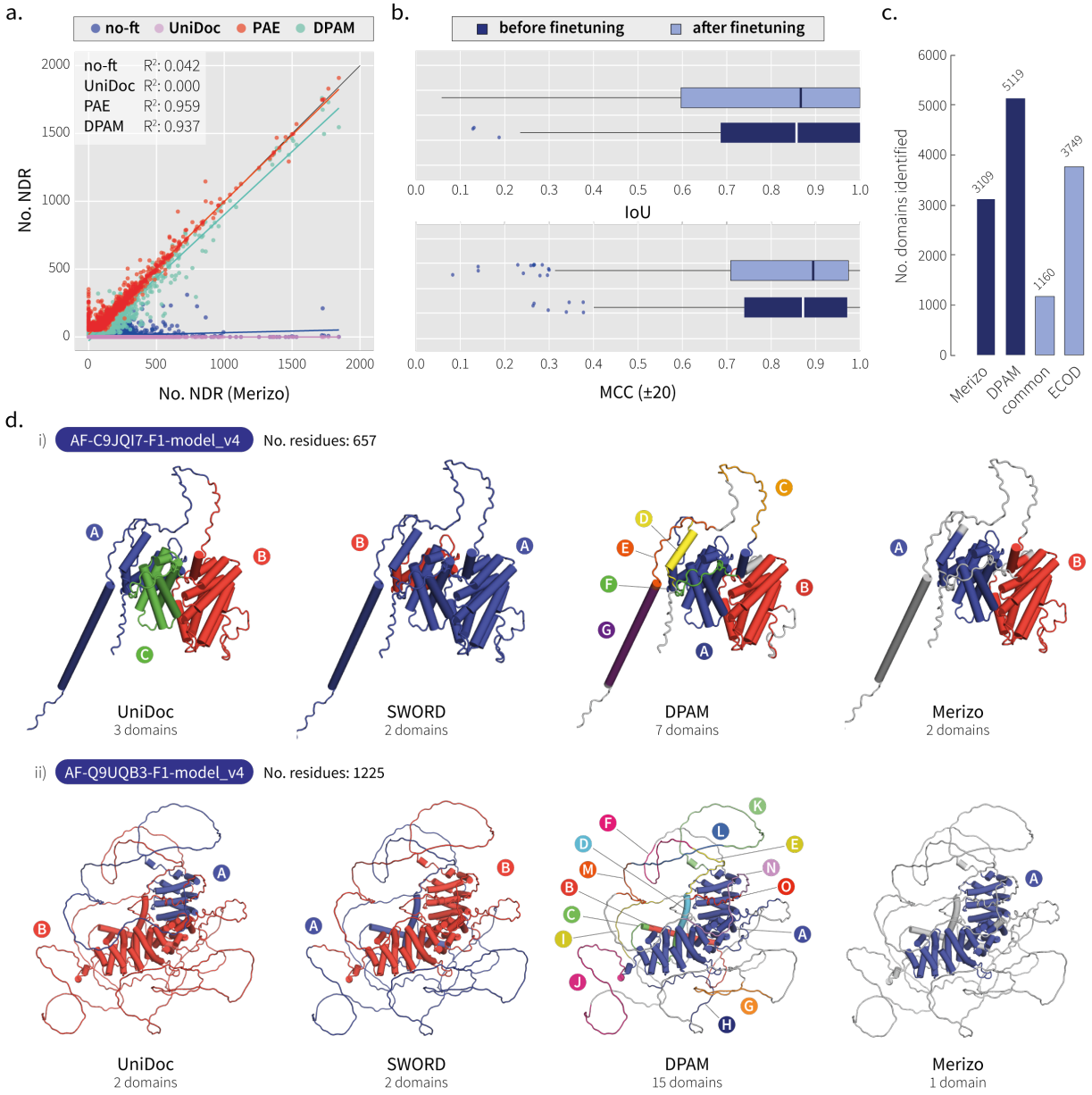

Figure 3: Performance of Merizo on the AFDB-1195 set. **a**) Agreement between the number of NDRs identified by Merizo after fine-tuning, compared to before fine-tuning (blue), UniDoc (purple), inferring from PAE and plDDT data (red) and as classified by DPAM (teal). The inset shows the coefficient of determination (R^2) for each correlation. **b**) IoU and MCC (boundary$\pm$20) distributions on the CATH-663 set before and after fine-tuning. **c**) The number of domains identified by Merizo (after fine-tuning), DPAM, common domains identified by both methods and the number of DPAM domains classified into ECOD families. **d**) Examples of domain assignments by UniDoc, SWORD, DPAM and Merizo. Each identified domain has been shown in a different colour as well as by the text labels from A to O. NDRs are shown in white. In both examples, NDR detection is the most robust in Merizo, while UniDoc and SWORD do not classify these regions entirely, and DPAM over-segments NDRs into additional domains.

generated by AlphaFold2.[24] The first observation is that approximately 37% of analysed residues were classified by Merizo as NDRs (Figure 4a). Overall, across 23,391 AFDB-human models, 74,250 candidate domains were identified, with most of these having high domain-level plDDT scores (residue plDDT averaged across the domain). 96.4% of these domains fall within the 'confident' to 'very high' plDDT bins demarcated by AlphaFold2,[24] indicating that most domains identified by Merizo are segmented from well-folded regions where AlphaFold2 is confident (Figure 4b).

In addition to predictions of domains, Merizo also outputs estimates of confidence in the predictions, expressed as predicted IoU (pIoU). The pIoU distribution produced by Merizo also illustrates that the network is confident in most of its predictions, as most fall into the 0.95-1 pIoU bin (Figure 4b). From our analysis of the CATH-663 set, we determined that a pIoU cutoff of 0.75 can be used to group domain predictions into high- (pIoU ≥ 0.75) and low-quality (pIoU < 0.75) predictions (Supp. Figure 4).

Next, to verify the validity of the identified domains, we extracted a subset of high-confidence domains by applying cutoffs to the domain-level plDDT (plDDT ≥ 70) and pIoU scores (pIoU ≥ 0.75). This process yielded 50,175 high-confidence domains, which were then putatively assigned to CATH superfamilies using the sequential structure alignment program (SSAP) score[25] against the CATH S40 non-redundant set.[2] The SSAP score quantifies the similarity between two structures, taking into account the order of residues as well as secondary structure elements and motifs. SSAP scores in the ranges of 60-70, 70-80, and 80-100 correspond to similarity at roughly the architecture, topology, and homologous superfamily levels, respectively. The distribution of SSAP scores, depicted in Figure 4c, illustrates the similarity between each high-confidence domain and the best-matching representative CATH domain. Overall, 40,818 domains (81.3%) were successfully aligned to an existing CATH class, with 49.7%, 19.2%, and 12.5% matching at the superfamily, topology, and architecture levels, respectively.

To analyze the subset of models that did not align straightforwardly to a CATH representative domain, we employed FoldSeek's *easy-cluster* algorithm[26] for clustering these structures. Applying a criterion of 50% coverage and a TM-score threshold of 0.5, we identified 5281 distinct clusters. On closer examination, a significant number of clusters corresponded to domain fragments, including segmented blades from propeller folds and variable-sized fragments from repetitive domains like HEAT repeats. However, several clusters corresponded to well-packed domains exhibiting high regional plDDT, which did not find a match among CATH representatives. We provide examples of these identified folds in Supp. Figure 5.

The superfamily distribution of the putatively assigned domains is shown in Figure 4d. CATH classes 1 to 3 (1: mainly alpha, 2: mainly beta, and 3: alpha beta domains) comprised most of the assignments, with only roughly 1% finding matches to classes 4 and 6 (4: few secondary structures and 6: special). Like the ECOD classification of the AFDB-human set,[23] the most abundant domain families included immunoglobulin-like folds such as canonical immunoglobulins (2.60.40.10) and cadherins (2.60.40.60). Domains in this topology are found in a range of proteins related to cell adhesion and immune response and are formed as long tandem repeats, including titin (Q8WZ42), sialoadhesin (Q9BZZ2), hemicentin-1 (Q96RW7), and FRAS1 (Q86XX4; Figure 4e). The most abundant superfolds identified included Rossman folds (3.40.50), jelly rolls (2.60.120), alpha/beta plaits (3.30.70), transferases (1.10.510) and helix-hairpins (1.10.287) (Supp. Figure 6).

2.4 Benchmarking the runtime of Merizo against other segmentation methods

Besides accuracy, another important consideration of a method is its speed. Although accuracy should be the primary concern, the enormous number of models that have been made available in different databases including the AFDB, means that the speed and efficiency of methods are increasingly important factors that should not be overlooked. As such, we conducted a benchmark study of runtimes, comparing Merizo against other methods on two sets of proteins. The first is the CATH-663 benchmark set which contains proteins from 90 to 739 residues long, while the second is a small set of 27 proteins (referred to as AFDB-27) selected from the AFDB-human set, chosen to encompass the full range of lengths (up to a maximum of 2700 residues) and to test runtimes on longer models.

Table 1 shows the measured runtimes for Merizo against UniDoc, SWORD, DeepDom and Eguchi-CNN and on different hardware. For several methods including Merizo, the average runtime per target is less than a second, with the fastest being DeepDom, which can process inputs in batches to allow them to be segmented concurrently. The slowest method in our benchmark was SWORD, which required 6366 seconds (1.7 hours) to process the CATH-663 set. As expected, runtimes on CPU are in general slower than on GPU hardware, however an exception to this is the UniDoc method which was 30% faster than Merizo (GPU). The fast runtime of UniDoc appears to be attributable to the partitioning algorithm which only considers residues that are not part of a secondary structure element, as potential boundaries. While this decreases the number of possible cut points, it also means that UniDoc intrinsically cannot produce precise domain assignments for structures where this assumption is not true. Examples of domain boundaries falling on secondary structure elements can be seen in both CATH and ECOD classifications and several cases from the CATH-663 set are shown in Supp. Figure 7.

Similar results were obtained from the AFDB-27 set which compared Merizo against other high-accuracy methods (including UniDoc, SWORD and DPAM) on AFDB models. Results are shown in Supp. Figure 8, where it can be seen that Merizo (GPU) and UniDoc are overall the fastest meth-

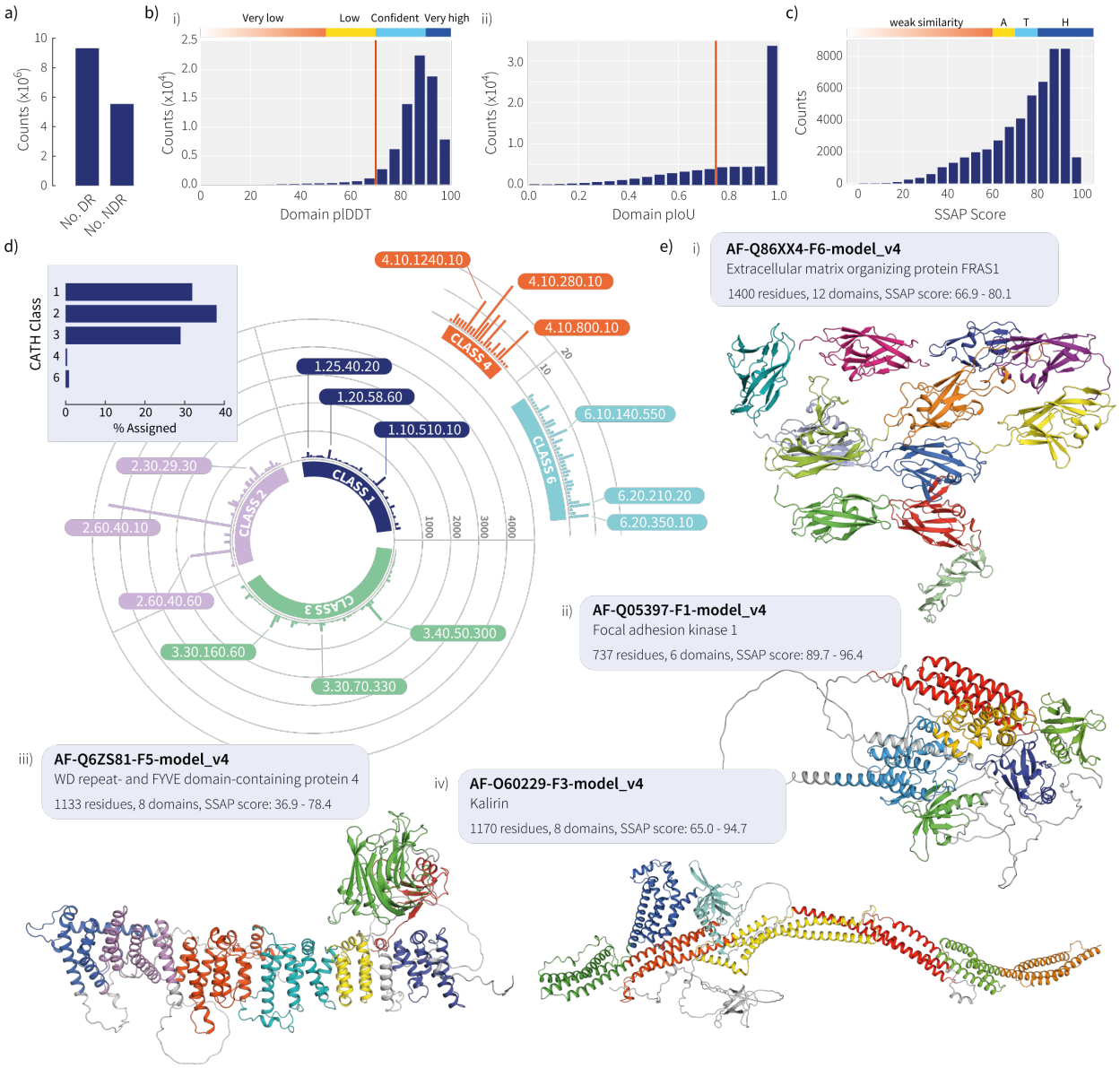

Figure 4: Putative domains identified in the entire human proteome. **a**) The number of residues identified as domain residues (DR) and NDR in the AFDB-human set. **b**) Distributions showing the i) average domain plDDT and ii) domain pIoU for 74,250 domains identified by Merizo in 23,391 models of the AFDB-human set. The red lines indicate thresholds applied to the plDDT and pIoU scores which were used to dichotomise a subset for further analysis. **c**) Distribution of SSAP scores for 50,175 confident domains, aligned to the CATH S40 non-redundant set. SSAP score bins demarcated above the histogram represent similarity at the CATH architecture (A; $70 > \text{SSAP} \geq 60$), topology (T; $80 > \text{SSAP} \geq 70$) and homologous superfamily (H; SSAP$\geq$80) levels. 40,818 identified domains align to CATH domains with a SSAP score of at least 60. **d**) The most abundant superfamilies identified in AFDB-human by Merizo. The inset shows the distribution of domains assigned to each CATH class. **e**) Examples of AFDB-human models segmented by Merizo, where each colour represents a different predicted domain. NDRs are shown in white.

ods. On targets below 1500 residues in length, UniDoc achieves lower runtimes than Merizo, however, the difference becomes smaller as models approach 2000 residues in length. The maximum model size that Merizo can process is limited by GPU memory, where on a NVIDIA 1080Ti with 11GB of memory, this maximum is roughly 2100 residues. Models longer than this cutoff can instead be processed either on a GPU with larger memory capacity, or by CPU, albeit at an 8-10x increase in runtime. Even on CPU however, Merizo compares very favourably to SWORD and DPAM which on the longer models, can be up to three orders of magnitude slower than Merizo. This difference is especially prominent when comparing the total runtime of each method, as SWORD and DPAM require approximately 40 hours to process the set of 27 models on a single CPU core while requiring only 14 minutes from Merizo on the same hardware. When the accuracy of each method is also taken into consideration, as well as applicability to AFDB models, the performance of Merizo compares extremely favourably, being able to produce accurate domain assignments even on AFDB models, at a fraction of the runtime of other methods.

3 Discussion and conclusions

In this study, we have developed a fast and accurate domain segmentation method which can be applied to both experimentally-derived PDBs as well as *in silico* models such as those generated by AlphaFold2. AlphaFold2 models differ considerably from the former, particularly in the abundance of NDRs seen in some models (which we estimate to be around 30% of residues in the AFDB-human set). The presence of these residues can preclude the ability of some methods (including UniDoc and SWORD) from operating successfully, however, is overcome in Merizo by explicitly predicting their locations through a fine-tuning regime that familiarises the network with these types of residues. While UniDoc offers slightly faster runtimes than Merizo, it is constrained by its idiosyncratic treatment of domain boundaries and its inability to accurately predict NDRs, rendering it generally unsuitable for certain domain structures (Supp. Figure 7) and AFDB models (Figure 3d).

Besides its speed, the major advantage of Merizo over other AFDB-centric methods such as DPAM is that Merizo requires only a PDB structure to operate, while DPAM makes use of several tools and databases (including HH-suite,[27] DALI,[28] Foldseek[26] and databases UniRef30,[29] PDB70[30] and ECOD[3]) as well as the PAE map from AlphaFold2. In a high-throughput setting, the minimal dependencies of Merizo make it particularly well-suited to operate on a large number of models, reducing both the amount of time spent on computation as well as on file management.

On an individual basis, one may surmise that identifying the boundaries of a domain within a single structure may not be difficult even by eye, however, the segmentation problem rapidly becomes intractable when a large number of structures are concerned. As the basic structural and functional units of protein structures, expanding our coverage of domain annotations across protein space can improve our understanding of their functions and how they interact with one another. In drug discovery, an expanded description of domains may facilitate the identification of potential targets as well as aid in repurposing existing drugs to new targets. A combination of Merizo together with structure searching or comparison tools such as FoldSeek[26] or Progres[31] would be well suited for identifying the structural homologs of a protein of interest by limiting the search space to the domains that matter most.

More recently, developments in single-sequence and language model-based prediction methods[32–34] have also been accompanied by faster run times over traditional sequence alignment-based methods (including AlphaFold2), which will undoubtedly boost the rate at which models will be made available to the scientific community. As these methods continue to improve in predictive accuracy, it may become commonplace for predictions to be made at genome scale or above, necessitating that any downstream analysis such as domain segmentation or function prediction be prepared to process or even re-process large amounts of data on a regular basis.

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 their 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.[16] In the context of ML, it may be advantageous to confine the labels used for training on a single classification scheme (at least in cases where assignments by different databases are at odds with one another) in order to avoid inadvertently introducing conflicting ground truths. That being said, as shown in Supp. Figure 3, there are cases where Merizo has produced an assignment that matches that of ECOD but not CATH, and these cases illustrate that the network's definition of domain packing was confident enough to correct the ground truth CATH assignment.

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 homologs in the Protein Data Bank. Supp. 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 organis-

Table 1: Comparison of runtimes on CATH-663 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
UniDoc	CPU	0.078	51.75	0.70
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	CPU	4.475	2966.77	39.92

ers 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.

Acknowledgements

We would like to thank Ian Sillitoe for helpful discussions on CATH domain annotations and Daniel Buchan for helpful feedback. This research was funded in whole, or in part, by the UKRI [Grant number BB/T019409/1]. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

4 Methods

4.1 Datasets used for training Merizo

4.1.1 CATH training dataset

The PDB chains and domain annotations used for training were accessed from version 4.2 of the CATH database.[2] 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. Edges were added between superfamilies if a PDB chain can be found that contains domains from two superfamilies (Supp. Figure 1). The resulting graph contains 655 components and is highly disproportionate, with the first and largest component containing roughly 60% of all superfamilies (2295), while the rest are spread across the other 654 components (1585 superfamilies). Additional statistics are summarised in Supp. Table 1.

Each graph component represents a subset of PDB chains which only contain domains from an isolated set of superfamilies. Thus, by iterating over the list of components, each can be assigned to either the training or the test set without PDB chains overlapping at the homologous superfamily level. As the largest component contains the majority of superfamilies and domains, it is naturally assigned to the training set. Of the remaining components, roughly 1 in 20 were held out to comprise the test set. Further redundancy filtering with CD-hit[35] was performed to cluster targets which had a sequence identity of greater than 99%. The final training and testing set contained 17,287 and 663 chains respectively.

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. As such, they are masked out during training to avoid polluting the network by learning these regions as either single domains or NDRs.

4.1.2 AFDB models used for fine-tuning

After training our network initially on the CATH dataset, Merizo was fine-tuned on models from the AFDB-human set in order to improve predictive performance on these types of models. The AFDB-human set contains 23,391 models, however, not all models could be used for fine-tuning for several reasons. First, some AFDB-human models may contain domains that are homologous to those in the CATH-663 set and such models should be avoided. Determining the non-overlapping subset of models from the AFDB-human set requires the consultation of ECOD annotations that were recently made available in.[23] From the set of 18,038 models which ECOD annotations were available for, we were able to ascertain a non-overlapping set of 8510 models suitable for use in fine-tuning.

Next, our method of generating ground truth NDR labels makes use of the PAE output of AlphaFold2. To generate ground truth labels, residues were dichotomised into either NDR (class 0) or non-NDR (class 1) categories based on two criteria: 1) the residue plDDT is less than 60, and 2) the standard deviation of PAE values for the residue is less than 0.4. All residues meeting both criteria are assigned as NDR, and all other residues are domain residues. As PAE data is not available for all targets, including any fragment model not designated with 'F1' in its filename, the final number of models that were used for fine-tuning was reduced to 7052. A smaller set of 1195 models were sequestered for testing Merizo and is referred to as AFDB-1195.

4.2 Network architecture

Merizo is a small encoder-decoder network with approximately 37M parameters (20.4M in the encoder and 16.8M in the decoder; 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 channels, while iteratively organising the backbone frames towards the ground truth structure. In our usage, we repurpose the IPA module in an input-reversed fashion to instead read a folded structure into a latent representation which can then be decoded to provide the segmentation map. The single representation is produced by one-hot encoding the primary sequence into 20 amino acid classes and then projected into 512 feature dimensions. The dimensionality of the single representation is maintained throughout the network. For the pairwise representation, we use the pairwise distance map derived from alpha carbons, directly embedded into 32 feature dimensions as continuous values using a linear layer. The final input to the IPA encoder are the Euclidean backbone frames which are calculated from each residue "frame" (N-CA-C atoms) via Gram-Schmidt orthogonalisation as per AlphaFold2.

The IPA encoder is composed of six weight-shared blocks, each with 16 attention heads and employs rotary positional encoding (RoPE[36]). In place of the typical feed-forward network that processes the attention outputs in the original Transformer model (and in AlphaFold2), we instead utilise a two-layer bi-directional gated recurrent unit (bi-GRU) which processes the post-attention single representation and introduces sequential dependency to the residue embeddings. The output of the IPA encoder is an updated single representation which is conditioned on structural information present in the pairwise and frame channels.

To make predictions from the single representation, the masked transformer decoder adapted from the Segmenter model[37] is used to predict domain masks (Figure 1b). Learnable embeddings corresponding to domain masks are concatenated to the single representation (as if they were extra residues) and processed by a stack of 10 multi-head attention (MHA) blocks, each with 16 heads. Each MHA block employs Attention with Linear Biases (ALiBi) positional encoding[38] which applies a penalty to the attention score between pairs of residues, according to the separation between their residue indices. The output of the MHA stack is divided to recover the original dimensions of the single representation and learned domain mask embeddings. The two tensors are matrix multiplied to produce the $[N, k]$ per-residue domain probability distributions, which can be converted into predicted domain masks by passing the probability distributions through the $argmax$ function.

In addition to domain masks, our network predicts two additional outputs: the positions of NDRs and per-domain pIoU predictions. NDR positions are made by passing the decoder-updated single representation through a two-layer biGRU with 256 hidden dimensions and projecting the output features of the final layer into two dimensions (with indices 0 and 1 signifying whether a residue is an NDR or not, respectively). The $argmax$ of the output generates a binary array of length $[N]$ which can be multiplied with the domain ID assignments to mask out the positions of predicted NDRs. To make pIoU predictions, the domain ID probability distributions are divided according to the predicted domain masks, and the set of residue probability distributions corresponding to each domain are passed through a two-layer biGRU with 512 hidden dimensions. In each case, the final timestep of the biGRU is projected into one dimension to predict a pIoU value between 0-1 which is trained to match the calculated IoU of the assigned domain against the ground truth assignment.

In a final stage, the predicted domain assignments are post-processed in a cleaning step which coalesces any domain with fewer than 30 residues or any segment fewer than 10 residues, with the domain preceding it. This step is performed outside of the neural network and reduces instances where only several residues are assigned into a domain segment. When two domains that are distal to one another are assigned to the same domain index by the network (which for example, can occur when a target contains a large number of domains), we separate these domains if the minimum distance between them is greater than 10Å.

4.3 Training procedures

Our method is trained fully end-to-end in PyTorch, with all input features calculated directly from PDB files. Training was conducted in two phases: initial and fine-tuning (not to be confused with fine-tuning on AFDB models, described in Section 4.4). Initial training was carried out for approximately 30 epochs using the Rectified Adam (RAdam) optimiser with a learning rate of 1e-4. During training, each target chain was randomly cropped to a window of 512 residues. was carried out for approximately 10 epochs in which the contribution of both the IPA and decoder loss terms were multiplied by a factor of 2. A minibatch of size 1 was used throughout, and gradients were accumulated and back-propagated every 32 mini-batches. All training was conducted using up to 6 NVIDIA GTX 1080Ti GPUs with 11GB of memory.

4.4 Fine-tuning on AFDB models

The fine-tuning of Merizo on AFDB models was performed in two stages. First, Merizo was finetuned to detect NDRs in the AFDB models. NDR tuning needs to occur first before self-distillation since the predicted NDR mask overrides the domain mask. Poor performance in predicting NDRs would naturally lead to poor domain boundary prediction. As ground truth labels for NDRs are not available, we inferred these positions via an empirically determined proxy based on residue-level PAE and plDDT scores (see Section 4.1.2). All network parameters are frozen with the exception of the bi-GRU that predicts the NDR masks (Figure 1b-iii). The dataset used for this exercise consists

of the set of 7052 AFDB models described in Section 4.1.2. The $\mathcal{L}_{bg,CE}$ loss component on the AFDB-1195 set is monitored throughout to measure network performance on the NDR task.

The second stage is conducted when the loss on the NDR task converges. All network weights are unfrozen and training is conducted as described in Section 4.3. As ground truth labels for the AFDB-human models are not available, we adopted a self-distillation approach whereby the predicted domain assignment is taken as the ground truth, following a cleaning function that removes any domains smaller than 30 residues as well as coalesces any segments that are fewer than 10 residues with the domain preceding it. During fine-tuning, the network is trained on the 17,287 chains with CATH annotations and 7052 AFDB-human models. Network performance on the CATH-663 set is monitored to ensure that performance on this set does not degrade. Losses on AFDB models were also scaled by a factor of 0.2. Hyperparameters were the same as those described in Section 4.3.

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] Kun Zhu, Hong Su, Zhenling Peng, and Jianyi Yang. A unified approach to protein domain parsing with inter-residue distance matrix. *Bioinformatics*, 39(2), 02 2023. btad070.

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

[17] 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.

[18] 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.

[19] Jing Zhang, R. Dustin Schaeffer, Jesse Durham, Qian Cong, and Nick V. Grishin. Dpam: A domain parser for alphafold models. *Protein Science*, 32(2):e4548, 2023.

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

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

[23] R. Dustin Schaeffer, Jing Zhang, Lisa N. Kinch, Jimin Pei, Qian Cong, and Nick V. Grishin. Classification of domains in predicted structures of the human proteome. *Proceedings of the National Academy of Sciences*, 120(12):e2214069120, 2023.

[24] Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, et al. Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596, 2021.

[25] William R Taylor and Christine A Orengo. Protein structure alignment. *Journal of molecular biology*, 208(1):1–22, 1989.

[26] Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, pages 1–4, 2023.

[27] Martin Steinegger, Markus Meier, Milot Mirdita, Harald Vöhringer, Stephan J Haunsberger, and Johannes Söding. Hh-suite3 for fast remote homology detection and deep protein annotation. *BMC bioinformatics*, 20(1):1–15, 2019.

[28] Liisa Holm. Dali and the persistence of protein shape. *Protein Science*, 29(1):128–140, 2020.

[29] Milot Mirdita, Lars Von Den Driesch, Clovis Galiez, Maria J Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2017.

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

[31] Joe G Greener and Kiarash Jamali. Fast protein structure searching using structure graph embeddings. *bioRxiv*, 2022.

[32] 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.

[33] 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.

[34] 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.

[35] Weizhong Li, Lukasz Jaroszewski, and Adam Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases . *Bioinformatics*, 17(3):282–283, 03 2001.

[36] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

[37] 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.

[38] 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.