Merizo: A rapid and accurate protein domain segmentation method using Invariant Point Attention

Supplementary Document

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

5 Supplementary Methods

5.1 Learning to cluster residues via embedding affinity

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 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 end up sharing the same label [1, 2]. Forcing a domain to use a particular label increases the difficulty of the segmentation task, as similar domains may inadvertently have conflicting labels. Several methods have been devised to encourage label-agnostic assignment, one of which is to use an index-invariant learning objective such as via affinity learning [1]. Under this learning regime, and in the context of protein domain segmentation, the embeddings 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 different when part of different domains [2, 3]. Calculating the pairwise similarity (i.e. affinity) between all pairs of positions, yields an affinity map, A, of shape $[N, N]$ (where N is the length of the protein). The ground truth domain arrangement can also be represented by $[N, N]$ domain map R which can be constructed given pairs of residues r_i and r_j as,

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

During training, the goal of the network is to minimise the difference between objective R and the predicted map A, by maximising or minimising the similarity of residue embeddings such that $A_{ij} \approx R_{ij}$. Residues with similar embeddings will naturally produce similar probability distributions and by extension, result in the same domain indices being assigned, thus allowing residues to be clustered into domains in a label-agnostic manner.

5.2 Loss functions

The loss of *positive* and *negative* residue pairs (where R_{ij} is equal to 1 and 0, respectively; Equation 1), can be calculated as,

$$\mathcal{L}^{+}_{ij} = 1 - \mathcal{S}_{ij} \tag{2}$$

$$\mathcal{L}^{-}_{ij} = \mathcal{S}_{ij} \tag{3}$$

where $\mathcal{S}_{ij}$ is the cosine similarity between the embeddings of residues r_i and r_j. To guide affinity map A towards domain map R, we calculate the overall loss of a single input protein using a balanced squared-affinity (BSA) loss, formulated as,

$$\mathcal{L}_{BSA} = \mathcal{L}^{+2} + \mathcal{L}^{-2} \tag{4}$$

The equal contributions of $\mathcal{L}^{+}$ and $\mathcal{L}^{-}$ terms attempt to balance the push and pull forces on the embeddings which leads to faster convergence, while squared terms emphasise the loss when deviations between A_{ij} and R_{ij} are large. The loss of each minibatch is calculated as the sum of four components,

$$\begin{aligned} \mathcal{L}_{total} = \beta_1 \cdot \mathcal{L}_{IPA,BSA} + \beta_2 \cdot \mathcal{L}_{decoder,BSA} \\ + \mathcal{L}_{bg,CE} + \mathcal{L}_{conf,MSE} \end{aligned} \tag{5}$$

where $\mathcal{L}_{IPA,BSA}$ and $\mathcal{L}_{decoder,BSA}$ correspond to the BSA loss applied to the post-IPA single representation of shape $[N, 512]$ and post-decoder domain mask tensor of shape $[N, k]$ and β_1 and β_2 are their respective weights (a hyperparameter set to 1 during initial training and 2 during fine-tuning). $\mathcal{L}_{bg,CE}$ is a cross-entropy loss term applied to the non-domain residue predictions of shape $[N, 2]$. $\mathcal{L}_{conf,MSE}$ is the mean squared error (MSE) loss applied to the confidence score predictions and is trained to return values close to 1 when a residue is assigned to the correct domain, and 0 otherwise.

5.3 Evaluation Metrics

5.3.1 Domain pairing

As our network is trained in a label-agnostic manner, the predicted domain indices may not match the ground truth values. Hence, the first exercise is to determine the correspondence between the ground truth and predicted domains (Algorithm 1). The domain pairing is determined from the ground truth domain labels $\mathbf{k}_{gt}$, and the predicted domain labels $\mathbf{k}_{pr}$, both of which are vectors of length N, where N is the number of residues in the input protein. Each element $\mathbf{k}_{x,i}$ (where x is gt or pr) describes the ground truth and predicted domain index of residue i. The unique set of non-zero labels in $\mathbf{k}_{gt}$ and $\mathbf{k}_{pr}$ are given by K_{gt} and K_{pr}, and the number of domains in each, given by $|K_{gt}|$ and $|K_{pr}|$. Note that $|K_{pr}|$ may not necessarily be equal to $|K_{gt}|$.

For each ground truth domain g in K_{gt}, we isolate all residue indices belonging to domain g, and determine p, the non-zero modal value of $\mathbf{k}_{pr}$ at these residue positions (by taking the mode of the array values). If p is not already assigned to a ground truth domain, g and p are paired domain indices. This procedure maximises the overlap between a predicted domain to its corresponding ground truth. From this pairing, we calculate the binary arrays $domain\_g$ and $domain\_p$ of length N, which contain 1 where a residue is in domains g and p respectively, and 0 at all other positions. The IoU and MCC scores are calculated from these arrays. If g cannot be matched to a value of p, which can occur for example, when $|K_{pr}| < |K_{gt}|$, no predicted domain index can be paired, and a score of 0 is given. For each ground truth domain, we record the domain-level IoU, MCC and the length of domain g. The final IoU and MCC scores W are given by taking the domain-length weighted average of the per-domain scores (Equation 6):

$$W = \frac{\sum_{i=1} w_i, J_i}{\sum_{i=1} w_i} \tag{6}$$

Where w represents an array of domain lengths and J represents the unweighted per-domain scores (either IoU or MCC).

5.3.2 Intersect-over-union

The IoU measures the degree of overlap between a predicted domain segment and its equivalent ground truth, where a score of 1 indicates that the two segments perfectly overlap, and 0 indicates no overlap. The overlap between two segments can be treated as a binary prediction task which allows the equation,

$$IoU = \frac{TP}{TP + FP + FN}, \quad \in [0, 1] \tag{7}$$

where TP (true positives) are the number of overlapping residues between the two segments, FP (false positives) is the number of residues in the predicted domain that are not in the ground truth, and FN (false negatives) are the number of residues in the ground truth that are not in the predicted segment.

5.3.3 Matthews Correlation Coefficient

The IoU is useful for evaluating how well a true domain is represented by a predicted domain, however, does little to quantify the exact boundary positions. For the latter, we use an MCC score applied directly to the boundaries of each domain (Equation 8). A predicted boundary is deemed correct if it lies within a range of $\pm m$ residues from an actual boundary. For our assessment, we set the value of m to 20 residues. This choice prevents an excessive penalty on boundaries that may be slightly less precise, particularly when the broader topology of the domain assignment remains similar. For all paired ground truth and predicted domains, we take the expected and predicted boundary indices and use a linear sum assignment algorithm to obtain an optimal 1:1 pairing between the two, while respecting $\pm m$.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad \in [-1, 1] \tag{8}$$

where TP and FN is the number of paired and unpaired ground truth boundaries, FP is the number of unpaired predicted boundaries, and TN are all other residue positions.

5.3.4 Domain count

The deviation between the expected $|K_{gt}|$ and predicted number of domains $|K_{pr}|$ in a single target can be quantified by calculating the absolute error between them. Averaging across a set of targets $\{X\}$, provides the mean absolute error

Algorithm 1 Domain Pairing Algorithm

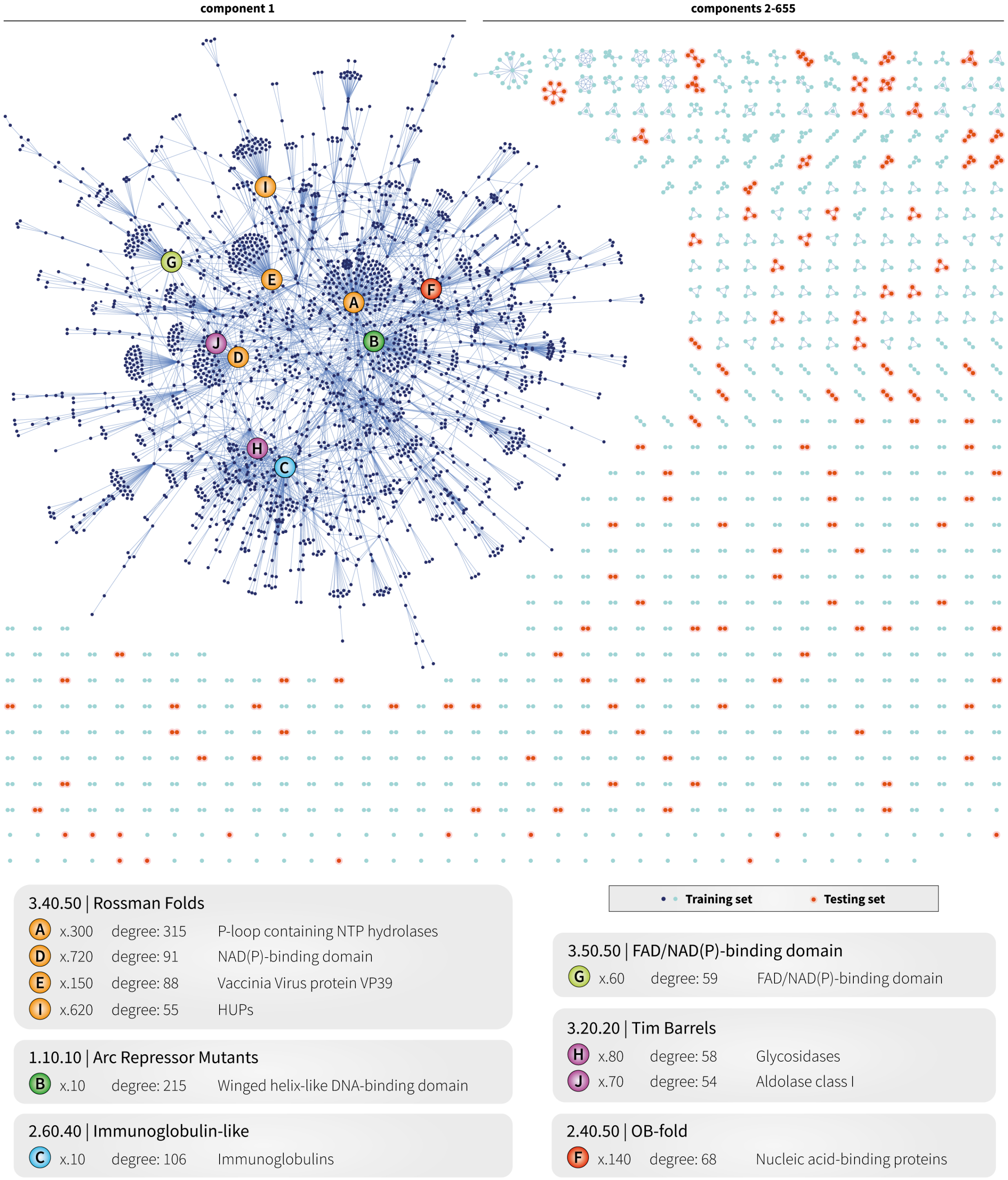
```
function DOMAINPAIRING($\mathbf{k}_{pr}, \mathbf{k}_{gt}$)    ▷ $\mathbf{k}_{pr}, \mathbf{k}_{gt} \in \mathbb{Z}^N$
    # Get the unique domain indices in $K_{gt}$ and $K_{pr}$
    $K_{gt} \leftarrow$ unique(nonzero($\mathbf{k}_{gt}$))    ▷ Ground truth domain indices, $K_{gt} = \{g_1, g_2, \ldots, g_G\}$
    $K_{pr} \leftarrow$ unique(nonzero($\mathbf{k}_{pr}$))    ▷ Predicted domain indices, $K_{pr} = \{p_1, p_2, \ldots, p_P\}$

    $G \leftarrow |K_{gt}|$    ▷ No. ground truth domains
    $P \leftarrow |K_{pr}|$    ▷ No. predicted domains

    # Iterate over each ground truth domain index $g$ in $K_{gt}$
    used, domain_lengths, all_iou, all_mcc $\leftarrow$ list(), list(), list(), list()

    for $g$ in $K_{gt}$ do
        domain_g $\leftarrow$ int($\mathbf{k}_{gt} = g$)    ▷ Get mask for domain $g$, $\{0, 1\}^N$
        nres_domain $\leftarrow$ sum(domain_g)    ▷ Get length of domain

        $p \leftarrow$ mode(nonzero($\mathbf{k}_{pr}$[domain_g]))    ▷ Apply mask to $\mathbf{k}_{pr}$ and get modal value of $p$

        if $p \notin$ used and $p \neq 0$ then
            used $\leftarrow$ append($p$)    ▷ Keep track of already assigned values of $p$
            domain_p $\leftarrow$ int($\mathbf{k}_{pr} = p$)    ▷ Get mask for domain $p$, $\{0, 1\}^N$

            domain_iou $\leftarrow$ calculate_iou(domain_g, domain_p)    ▷ Equation 7
            domain_mcc $\leftarrow$ calculate_mcc(domain_g, domain_p)    ▷ Equation 8
        else
            # Assign scores of 0 if no value of $p$ can be assigned
            domain_iou $\leftarrow 0$
            domain_mcc $\leftarrow 0$

        domain_lengths $\leftarrow$ append(nres_domain)
        all_iou $\leftarrow$ append(domain_iou)
        all_mcc $\leftarrow$ append(domain_mcc)

    # Calculate the domain-length weighted average scores
    iou $\leftarrow$ get_length_weighted_average(all_iou, domain_lengths)    ▷ Equation 6
    mcc $\leftarrow$ get_length_weighted_average(all_mcc, domain_lengths)    ▷ Equation 6

    return iou, mcc
```

(MAE) metric which summarises the average deviation of the predicted domain count against the ground truth and is given by,

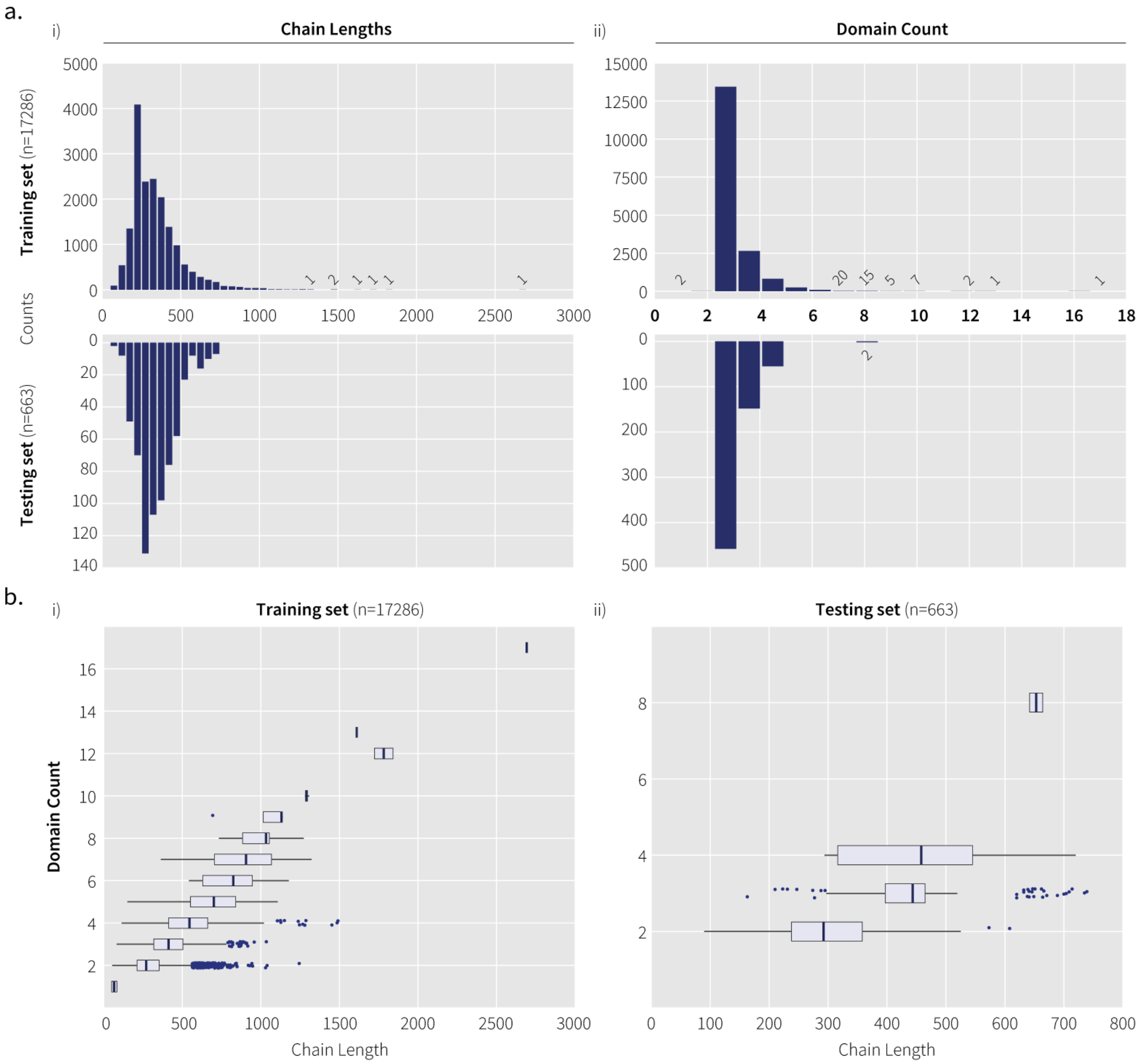
$$MAE = \frac{1}{|\{X\}|} \sum_{j=1} ||K_{pr,j}| - |K_{gt,j}||, \quad \in [0, \infty] \tag{9}$$

Supplementary Table 1: Training and test set statistics.

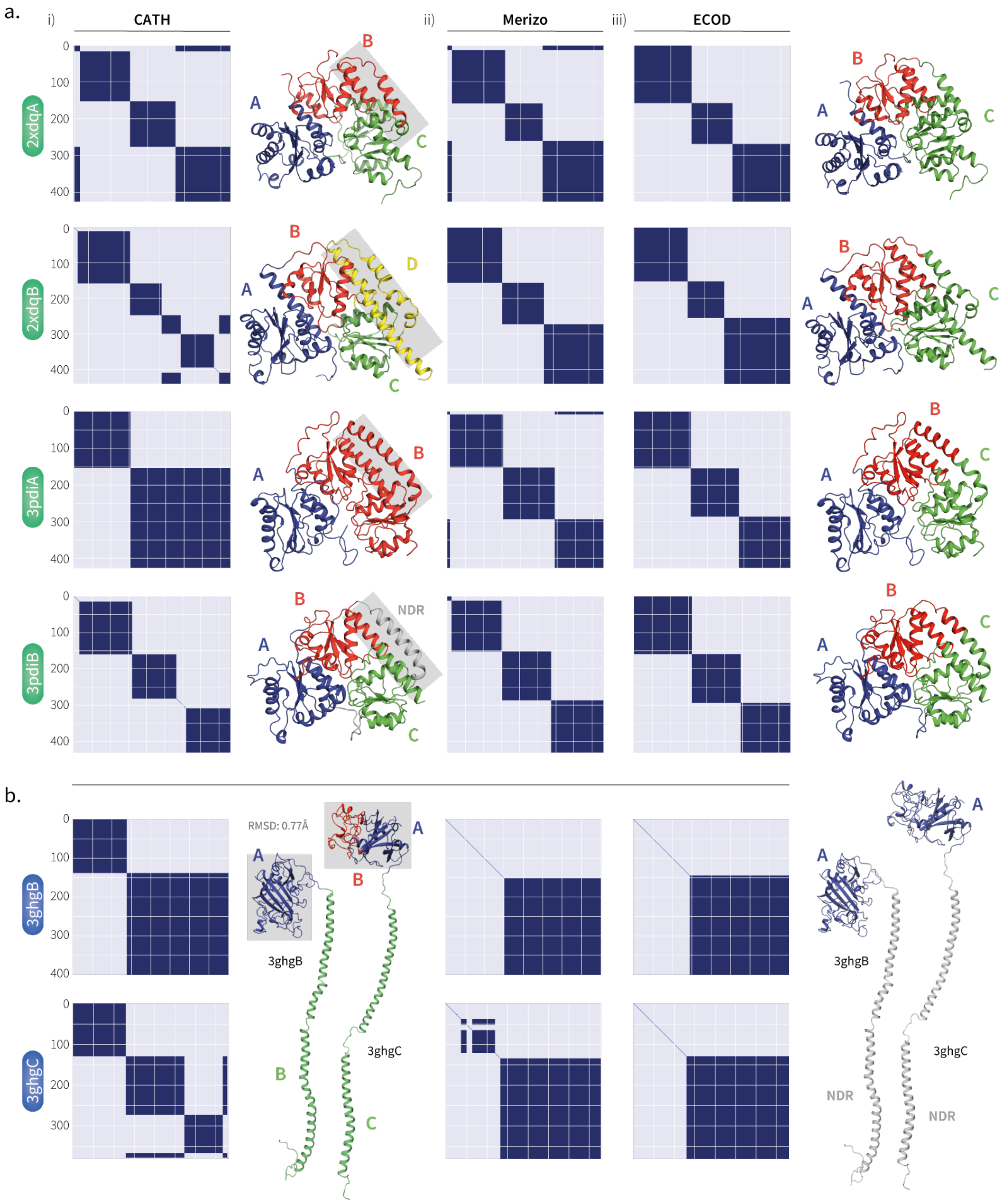
	Training	**% of total**	**Testing**	**% of total**	**Total**
No. Components	537	82.0	118	18.0	655
of which contain a single superfamily	62	83.8	12	16.2	74
of which contain >1 superfamily	475	81.8	106	18.2	581
No. Topologies	974	86.0	158	14.0	1132
No. Superfamilies	3587	92.4	293	7.6	3880
No. Domains	43214	96.0	1780	4.0	44994
No. Chains	18231	96.2	725	3.8	18956
Redundant chains (seqid > 0.99)	944	93.8	62	6.2	1006
Final No. Chains	17287	96.3	663	3.7	17950


component 1
components 2-655
A
B
C
D
E
F
G
H
I
J
Training set
Testing set
3.40.50 | Rossman Folds
A x.300 degree: 315 P-loop containing NTP hydrolases
D x.720 degree: 91 NAD(P)-binding domain
E x.150 degree: 88 Vaccinia Virus protein VP39
I x.620 degree: 55 HUPs
1.10.10 | Arc Repressor Mutants
B x.10 degree: 215 Winged helix-like DNA-binding domain
2.60.40 | Immunoglobulin-like
C x.10 degree: 106 Immunoglobulins
3.50.50 | FAD/NAD(P)-binding domain
G x.60 degree: 59 FAD/NAD(P)-binding domain
3.20.20 | Tim Barrels
H x.80 degree: 58 Glycosidases
J x.70 degree: 54 Aldolase class I
2.40.50 | OB-fold
F x.140 degree: 68 Nucleic acid-binding proteins

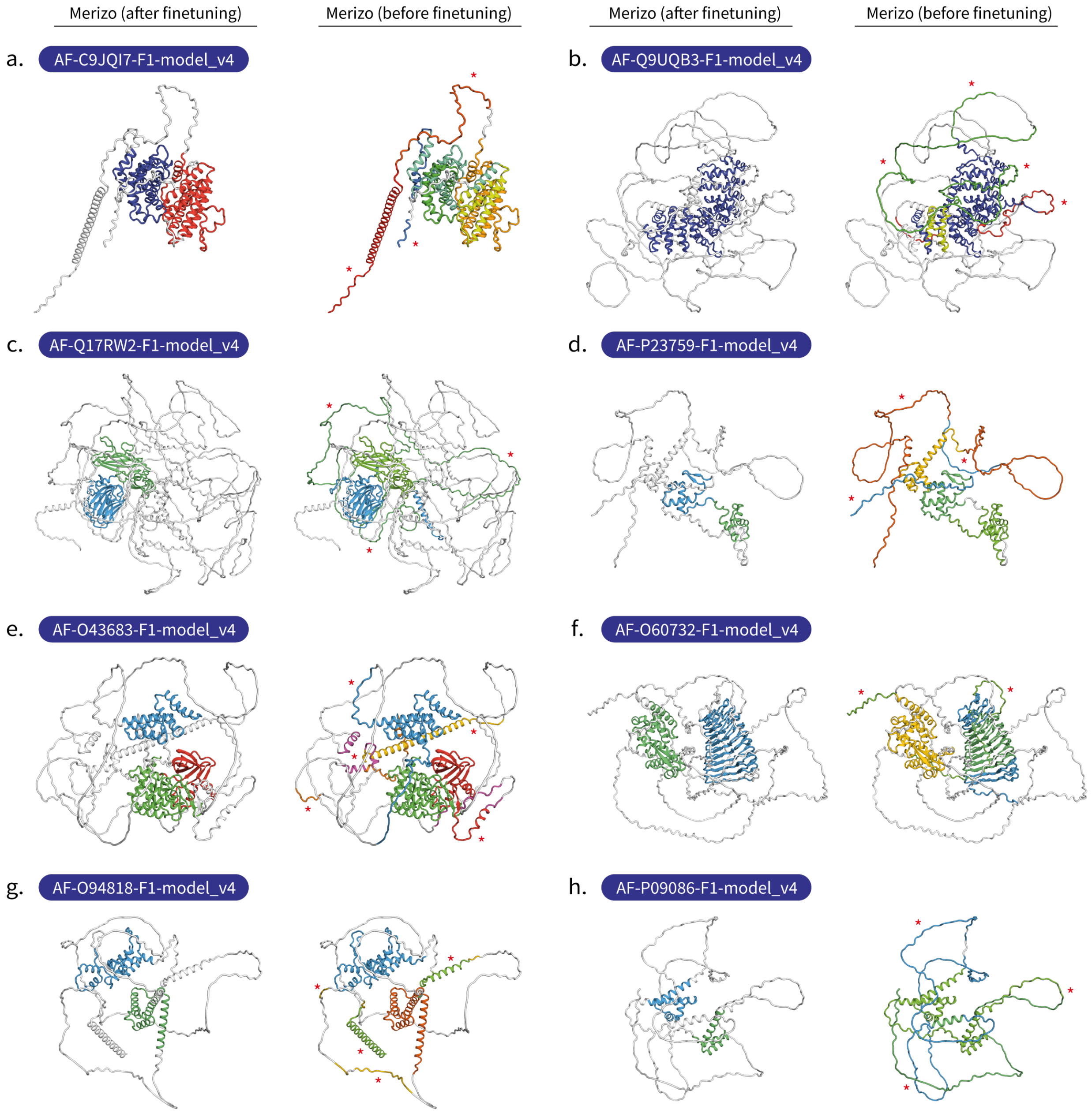
Supplementary Figure 1: CATH superfamily graph. The graph is composed of 3880 nodes across 655 components (i.e. sub-graphs) where each node represents a CATH superfamily. Two nodes are connected when a PDB chain containing domains from both superfamilies can be found. The 10 most connected superfamilies are highlighted by labels A-J. A summary of each hub superfamily is shown in the legend at the bottom of the figure. Blue and red components represent those assigned to either the training or test set. The lack of components sharing both green and red nodes indicates that both sets do not overlap at the superfamily level. Additional training and testing set statistics are reported in Supp. Table 1.



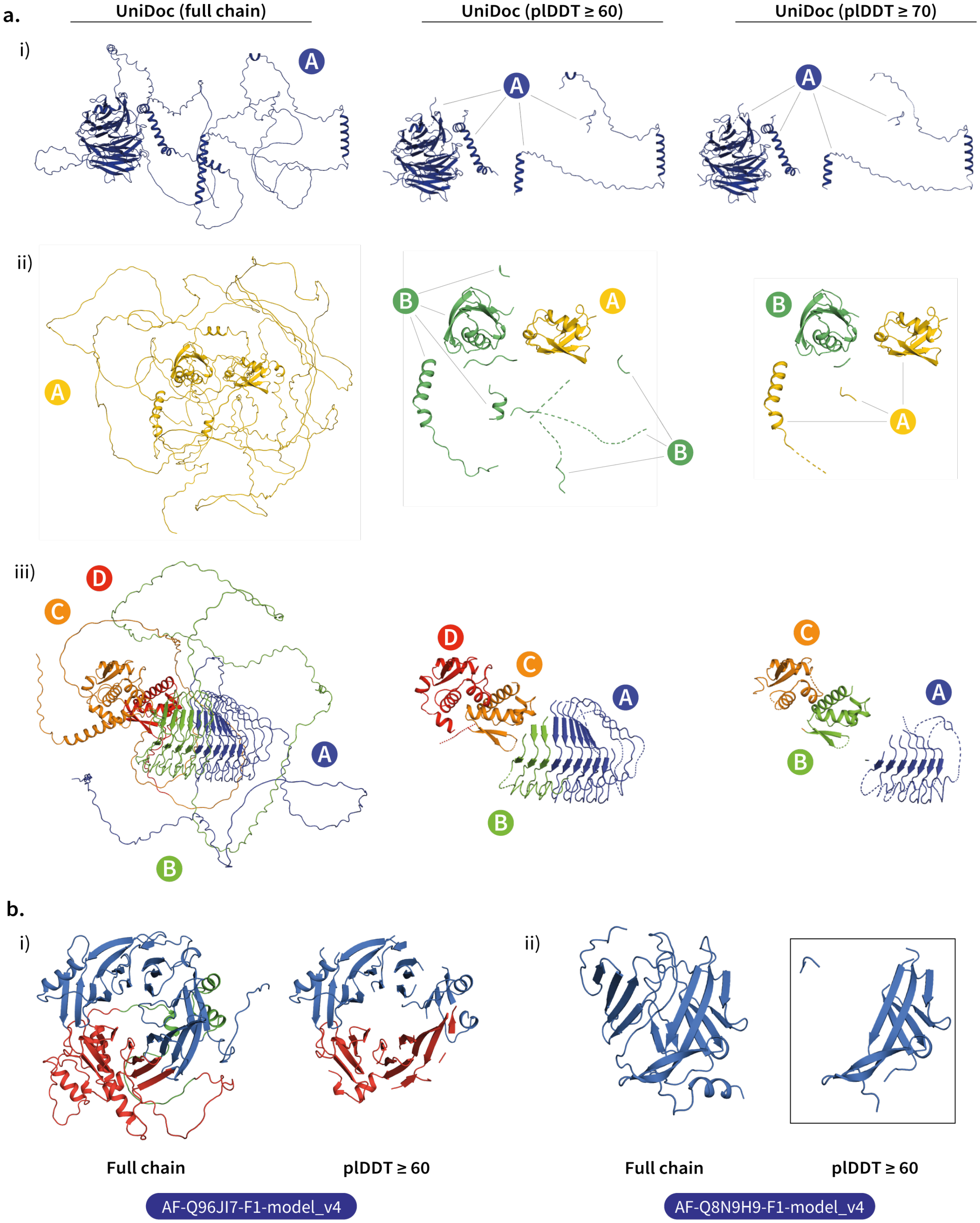
Supplementary Figure 2: Training and test set statistics. **a**) Chain lengths and domain count distributions for targets in the training and test sets. Bins with very few members are labelled with the bin count for clarity. **b**) Chain length distributions delimited by domain count for i) training and ii) test sets. The bar within each box represents the distribution median. Outliers are defined as chains with a length exceeding 1.5xIQR. The two targets in the training set with a domain count of 1 result from holding pen domains being removed from multi-domain chains.



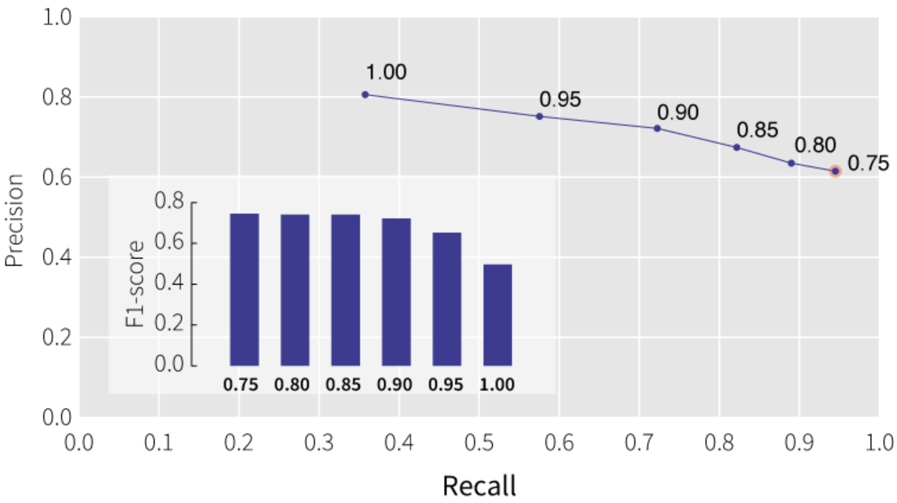
Supplementary Figure 3: Examples of targets where Merizo agrees with ECOD but not CATH. The i) CATH domain map and structure, ii) Merizo-predicted domain map and iii) ECOD domain map and structure are shown for two sets of multidomain targets from the CATH-663 set. Set (a) contains the subunit structures of Protochlorophyllide Oxidoreductase (2xdqA and 2xdqB) and precursor-bound NifEN (3pdiA and 3pdiB). All four chains are classified as three domains by ECOD and Merizo but vary in domain assignment by CATH despite having similar structures. The grey box highlights a region of three helices which differ in assignment in each chain. In example (b), the assignment of the C-terminal domain in human fibrinogen (3ghgB and 3ghgC) differs in CATH despite having only a 0.77Å RMSD difference (only residues highlighted by the grey box are aligned). Domain assignments by Merizo and ECOD are mostly identical with the exception of Merizo classifying a portion of the N-terminal tail as a separate domain.



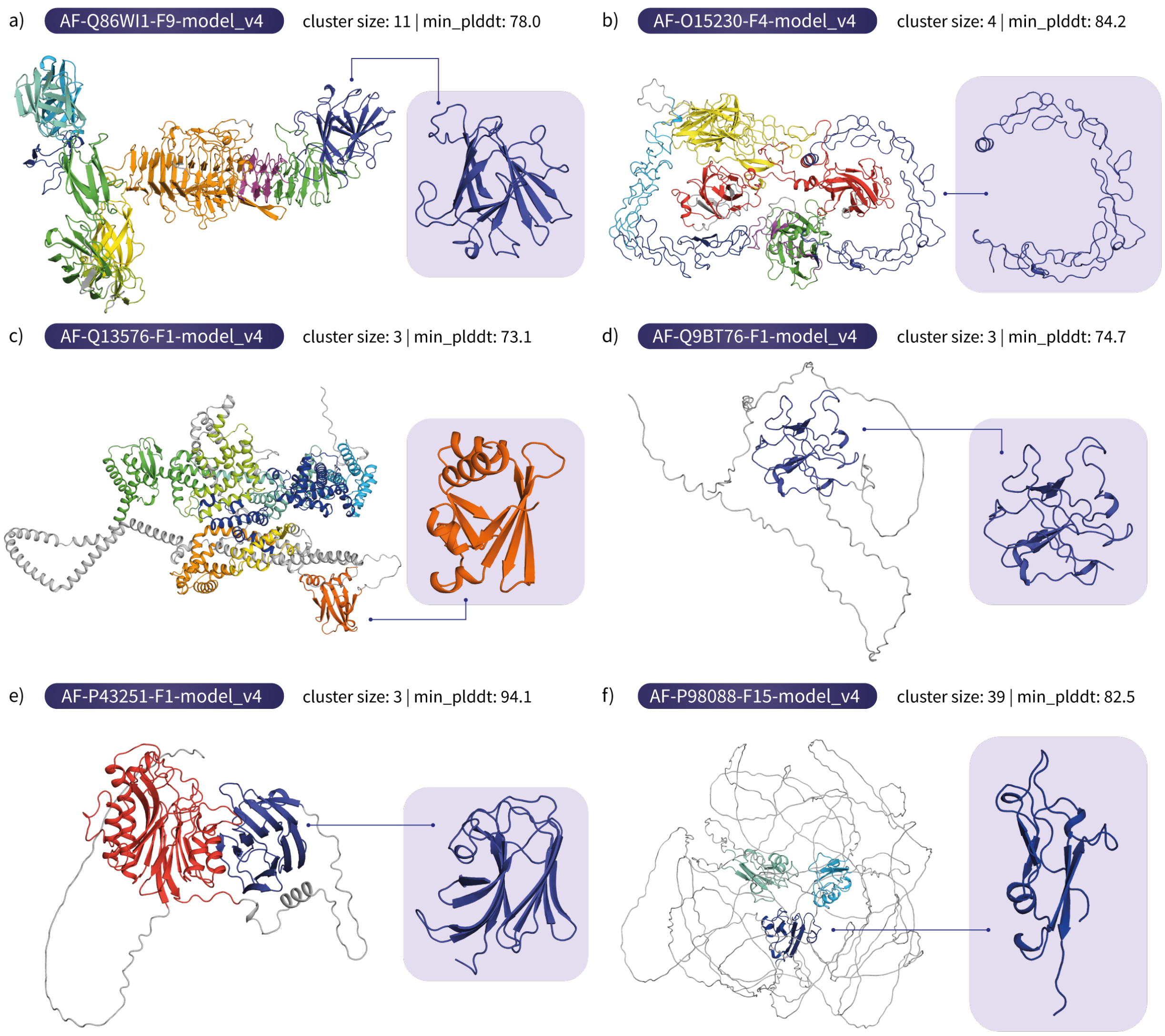
Supplementary Figure 4: Examples of domain assignments from Merizo before and after finetuning. Panel **a**) and **b**) correspond to the same models as shown in Figure 3d. In each example, the segmentation result after (left) and before (right) finetuning on NDRs is shown. Asterisks (*) highlight areas where domain assignments have been made to NDR regions. Domains have been demarcated using contrasting colours.



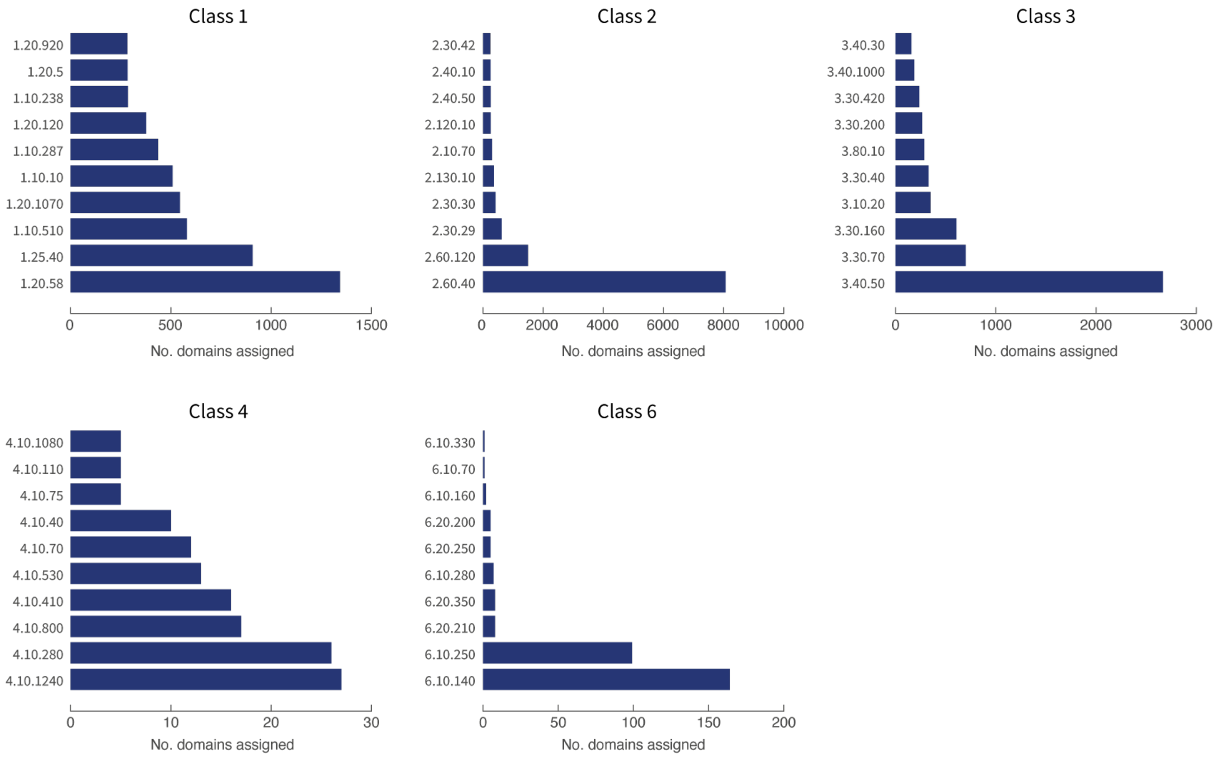
Supplementary Figure 5: Domain assignments by UniDoc before and after plDDT filtering. a) Example models and UniDoc assignments for i) AF-Q5VTH9-F1-model_v4, ii) AF-Q53SF7-F1-model_v4 and iii) AF-O60732-F1-model_v4. Left column shows the UniDoc assignment for the full chain model. Centre column shows the UniDoc assignment on the same structure, after a residue plDDT of 60 is applied (residues with a plDDT of less than 60 are removed). Right column shows the UniDoc assignment at a plDDT threshold of 70. Domains are shown in distinct colours and demarcated using letters A-D. **b**) Examples of domain damage by applying a plDDT threshold of 60. Some residues are omitted in **b**) for clarity.



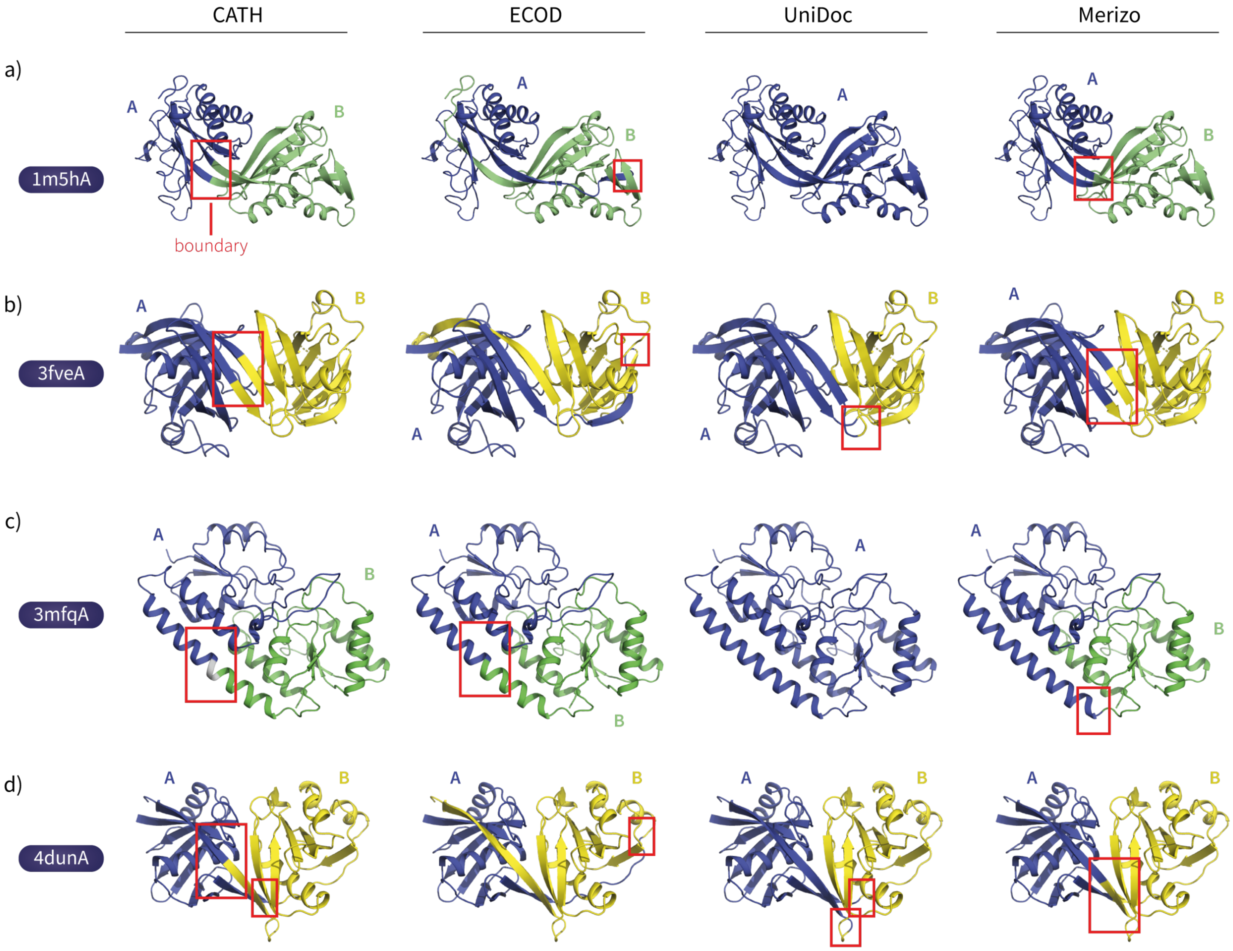
Supplementary Figure 6: Determining an optimal pIoU threshold Precision-recall curve showing the precision, recall and F1-score obtained when using the pIoU as a binary measure of domain assignment quality. The highest F1-score obtained is from using a pIoU cutoff of 0.75. Domains with a pIoU of 0.75 or greater are likely true positives, while those below this value are likely false positives.


a)
AF-Q86WI1-F9-model_v4
cluster size: 11 | min_plddt: 78.0
b)
AF-O15230-F4-model_v4
cluster size: 4 | min_plddt: 84.2
c)
AF-Q13576-F1-model_v4
cluster size: 3 | min_plddt: 73.1
d)
AF-Q9BT76-F1-model_v4
cluster size: 3 | min_plddt: 74.7
e)
AF-P43251-F1-model_v4
cluster size: 3 | min_plddt: 94.1
f)
AF-P98088-F15-model_v4
cluster size: 39 | min_plddt: 82.5

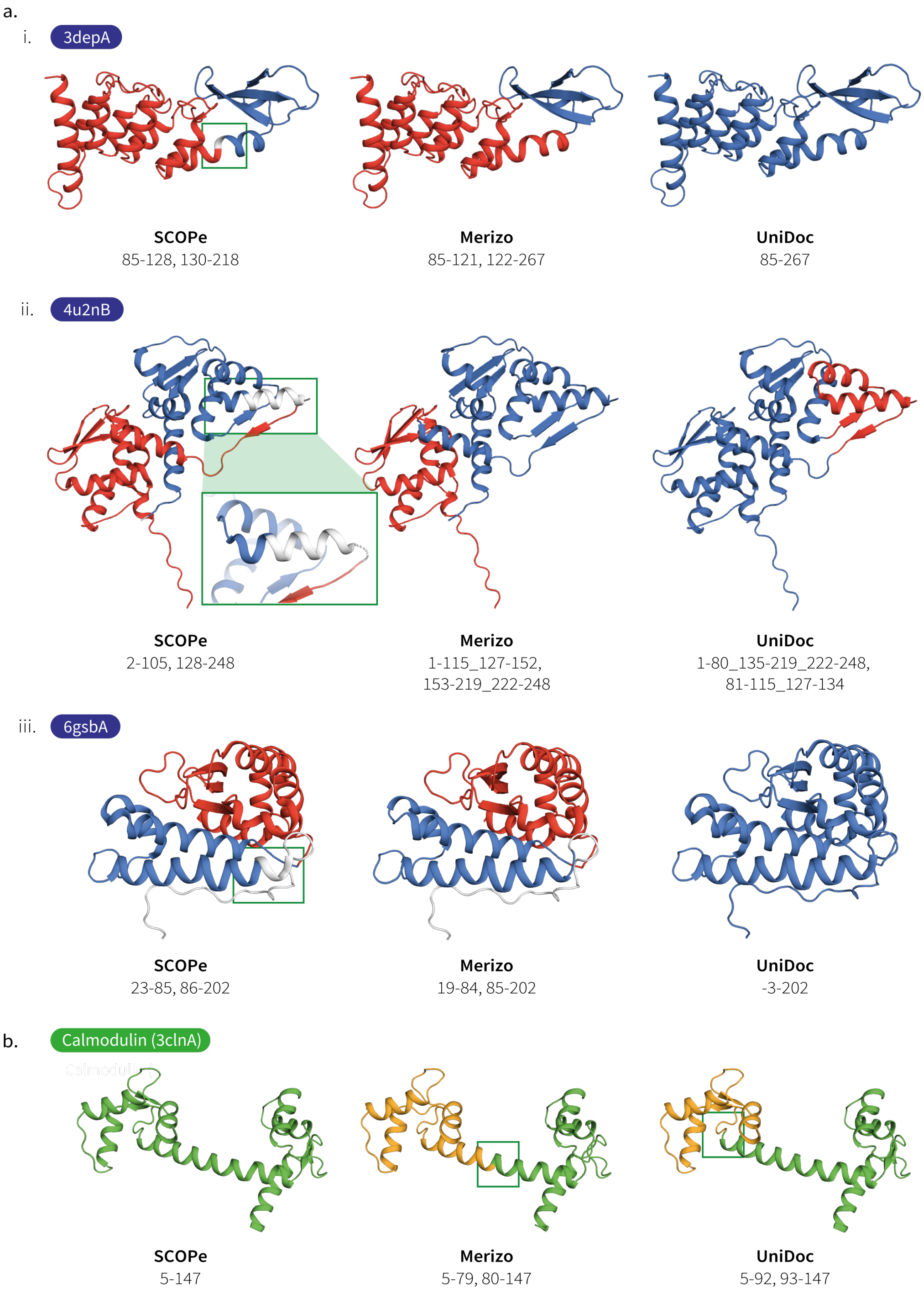
Supplementary Figure 7: Examples of folds identified in the AFDB-human set not matched to CATH representative domains. In each panel a-f, the full AFDB model as well as the non-CATH-matched domain is shown. Models shown are the cluster representatives of the subset of domains identified in the AFDB-human set that cannot be aligned to a CATH S40 representative domain based on the SSAP score. The cluster size as well as the lowest domain pldd (average plDDT across the domain) is shown for each example.



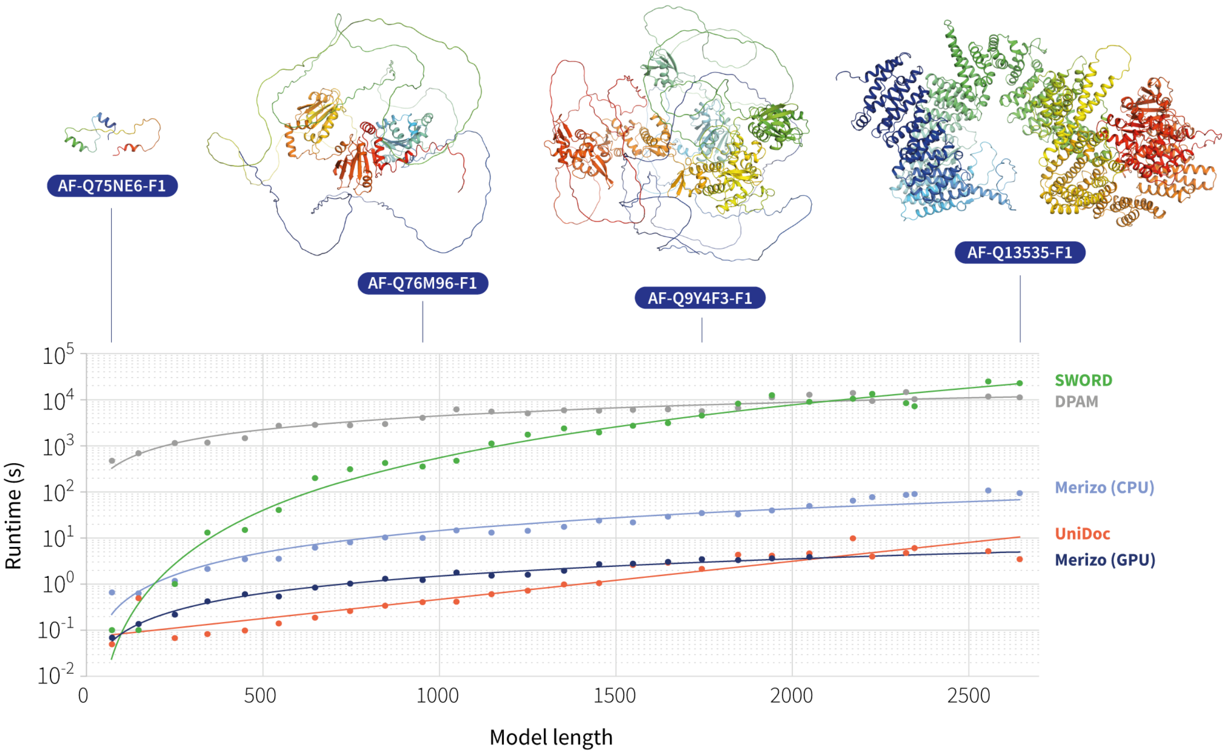
Supplementary Figure 8: Number of domains putatively assigned to CATH topologies in the AFDB-human set. The 10 most abundant CATH topologies identified in each CATH class are shown. Data encompasses 34,564 domains that could be matched to CATH S40 representative domains with a SSAP score of at least 70, indicating topology-level similarity.



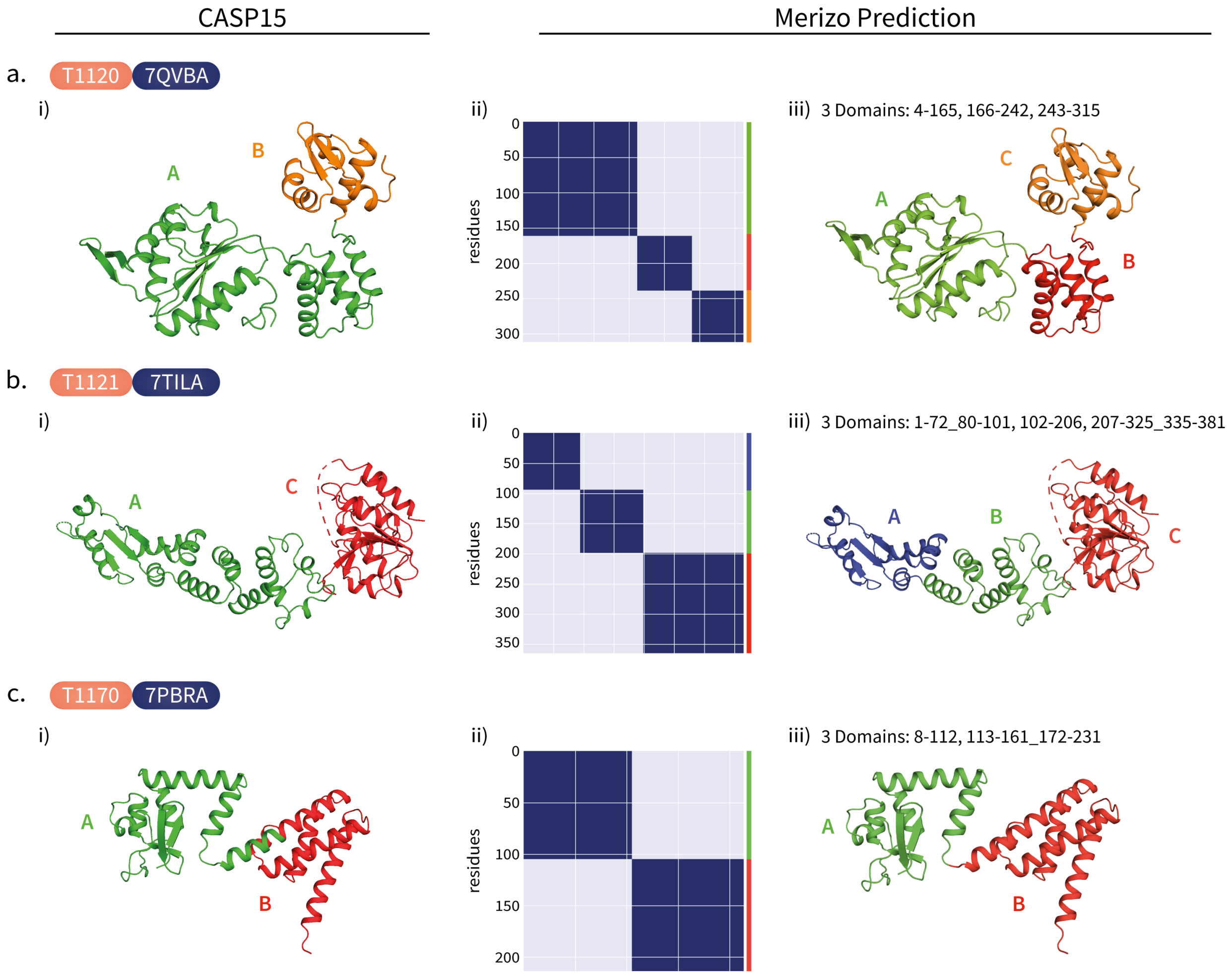
Supplementary Figure 9: Examples of CATH and ECOD domain annotations for multi-domain proteins with domain boundaries on secondary structure elements. Four examples are shown where in each case, the CATH domain boundary assignment is within a secondary structure element (red box). In c) both CATH and ECOD assign a boundary to the middle of a long helix which runs adjacent to the two domains. Domain boundary assignments by UniDoc are restricted to residues not part of secondary structure elements, while Merizo does not have this restriction. In examples b and d, UniDoc assigns the boundary to the end of the nearest β-strand, while the assignment by Merizo is more precise and is made to the strands themselves.



Supplementary Figure 10: Examples of SCOPe domain annotations and assignments by Merizo and UniDoc. **a**) Examples of SCOPe domain boundaries which are part of secondary structure elements. The domain boundary is highlighted by a green box in each example. The inset in panel ii), highlights an internal chain break in the structure proceeding the boundary SSE. **b**) The domain assignment of calmodulin from SCOPe, Merizo and UniDoc. In all examples shown, domains are coloured in distinct colours and NDRs are shown in white.



Supplementary Figure 11: Comparison of runtimes for the AFDB-27 set The model lengths of the AFDB-human set were divided into 100-residue bins (27 in total) and one model was selected from each bin at random, to form a set of 27 models encompassing the full range of AFDB model lengths. Each method was timed on how long it took to segment models. Merizo timings were conducted on either a single GPU (NVIDIA GTX 1080Ti with 11GB of memory) or a single CPU (Intel Xeon E5-1620 v3 @ 3.50GHz). Line functions of the form $y = bx^k$ were fit to each set of data points with the exception of UniDoc data which was fit to an exponential function. Note that 6 of the longest models were not processed by Merizo (GPU) due to memory limitations.



Supplementary Figure 12: Segmentation maps for CASP15 multi-domain targets. i) Domain classifications as provided by CASP organisers (left) vs ii) Merizo predictions, are shown for three multidomain targets T1120, T1121 and T1170. Targets T1121 and T1170 are annotated as two-domain proteins in CASP15 but are predicted as three domains by Merizo.

References

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

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

[3] Wei Huang, Shiyu Deng, Chang Chen, Xueyang Fu, and Zhiwei Xiong. Learning to model pixel-embedded affinity for homogeneous instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1007–1015, 2022.