Chainsaw: protein domain segmentation with fully convolutional neural networks

Jude Wells, Alex Hawkins-Hooker, Nicola Bordin, Ian Sillitoe, Brooks Paige, Christine Orengo

https://academic.oup.com/bioinformatics/article/40/5/btae296/7667299

Burak Suyunu

03.10.2024

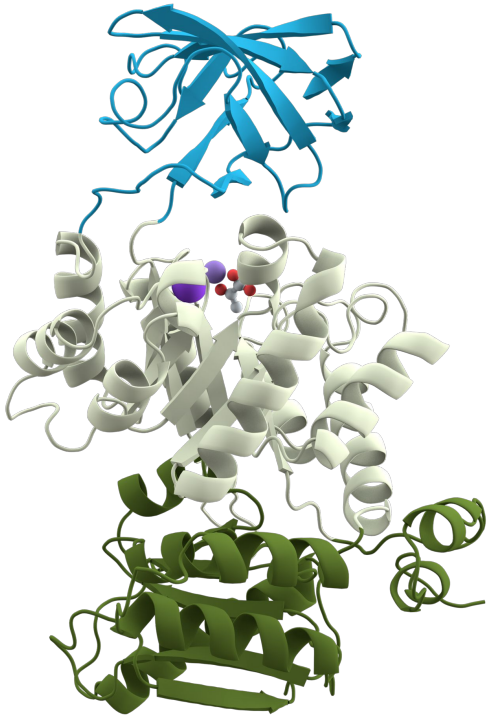
Protein Domains

What are Protein Domains?

- Structural units within proteins that fold independently.
- Comprised of several secondary structures like alpha helices and beta strands.
- Play a key role in protein function and stability.
- Many proteins consist of several domains, and a domain may appear in a variety of different proteins.
- Vary in length from between about 50 amino acids up to 250 amino acids in length.

Why are Protein Domains Important?

- Understanding domains aids in protein classification, evolutionary analysis, and the design of new proteins.
- Protein domains are often associated with specific biological functions.



Pyruvate kinase, a protein with three domains (PDB: 1PKN).

The Scale of the Protein Structure Problem

Explosion of Structural Data

- Over 200 million protein structures predicted by AlphaFold (DeepMind, 2022).
- Structural databases like CATH, SCOP, and ECOD are crucial for annotating these structures.
- Challenge in Domain Segmentation

Many protein structures in databases are unannotated.

- Domain segmentation is essential for mapping evolutionary links and understanding protein functions.

Existing Approaches to Domain Segmentation

Sequence-Based: Predicts domain boundaries using only the amino acid sequence.

Limitation: Less accurate for complex or novel proteins.

Structure-Based: Uses 3D structural information to predict domains.

Uses contact maps or pairwise residue distances.

Density of contacts is higher within domains than between domains

Limitation: Existing methods often rely on unsupervised or heuristic algorithms.

EguchiCNN trains CNN to do image segmentation on protein structures represented by 2D distance maps.

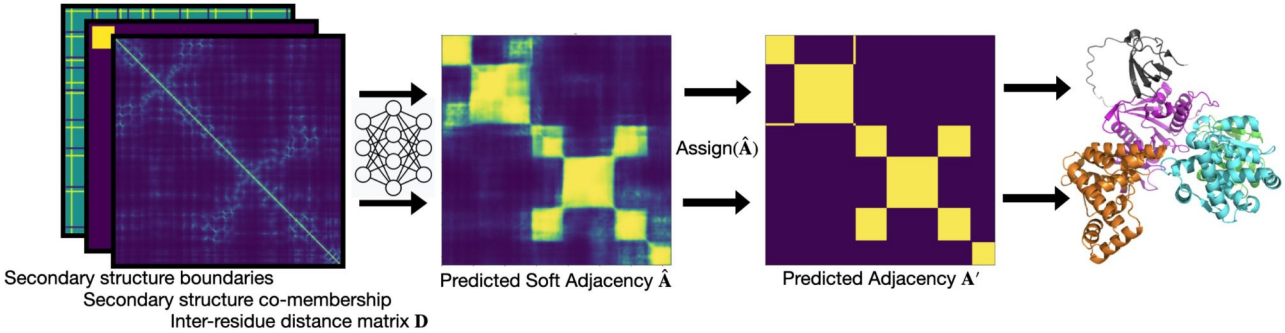
Merizo uses a transformer architecture with invariant point attention to directly cluster residues into domains based on both sequence and structure inputs. However, Merizo was not trained on any single-domain CATH proteins, as such we find that it tends to over-split single-domain proteins

Chainsaw Overview

Chainsaw relies on a 2D CNN trained to estimate the probability that pairs of residues belong in the same domain

Domain boundaries are derived from these pairwise co-membership probabilities using a greedy algorithm that searches for the most likely assignment of residues to domains given the predicted probabilities

Trained using CATH-annotated PDB files with a specific focus on proteins from CATH classes 1, 2, and 3.



Chainsaw Advantages

Formulating the supervised learning problem as a **classification** task at the level of **pairs of residues** rather than as a **boundary prediction** task at the level of **individual residues** has notable advantages.

1. It makes the prediction of **discontinuous domains** more straightforward.
2. It sets **no limit on the number of domains** that can be predicted.
3. It improves the **class imbalance** of boundary to non-boundary residues.
4. It solves **sensitivity to small changes in the boundary**.
    a. multiple adjacent residues could equally be considered to be the ‘correct’ boundary

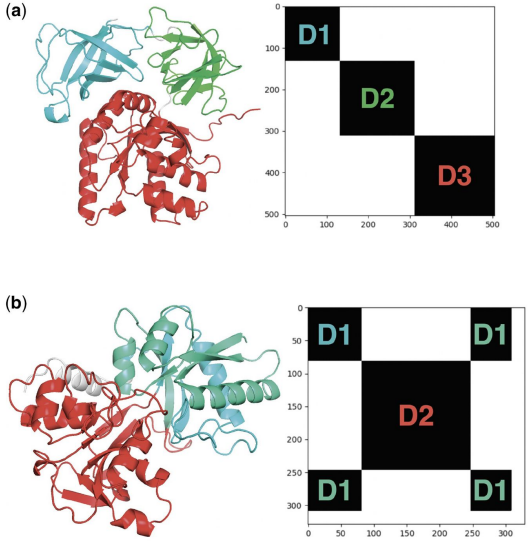
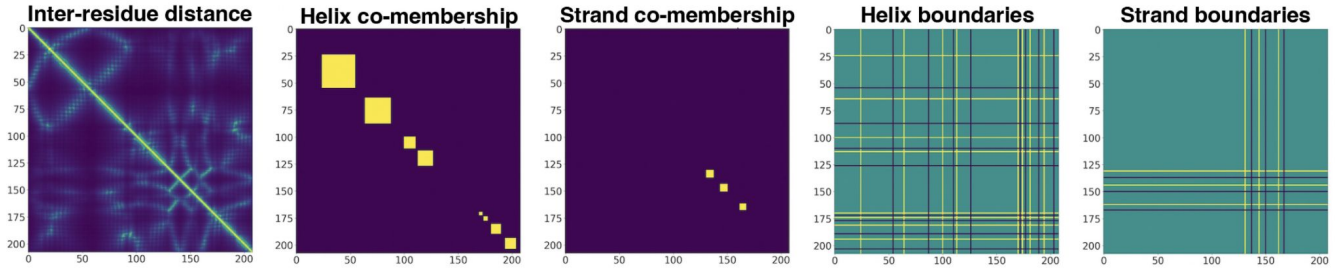


Figure 5. Domain assignments represented with binary pairwise co-membership matrices. (a) PDB 1b23P with continuous domains and corresponding pairwise domain label representation. (b) PDB 1a8eA with a discontinuous first domain

Input Features

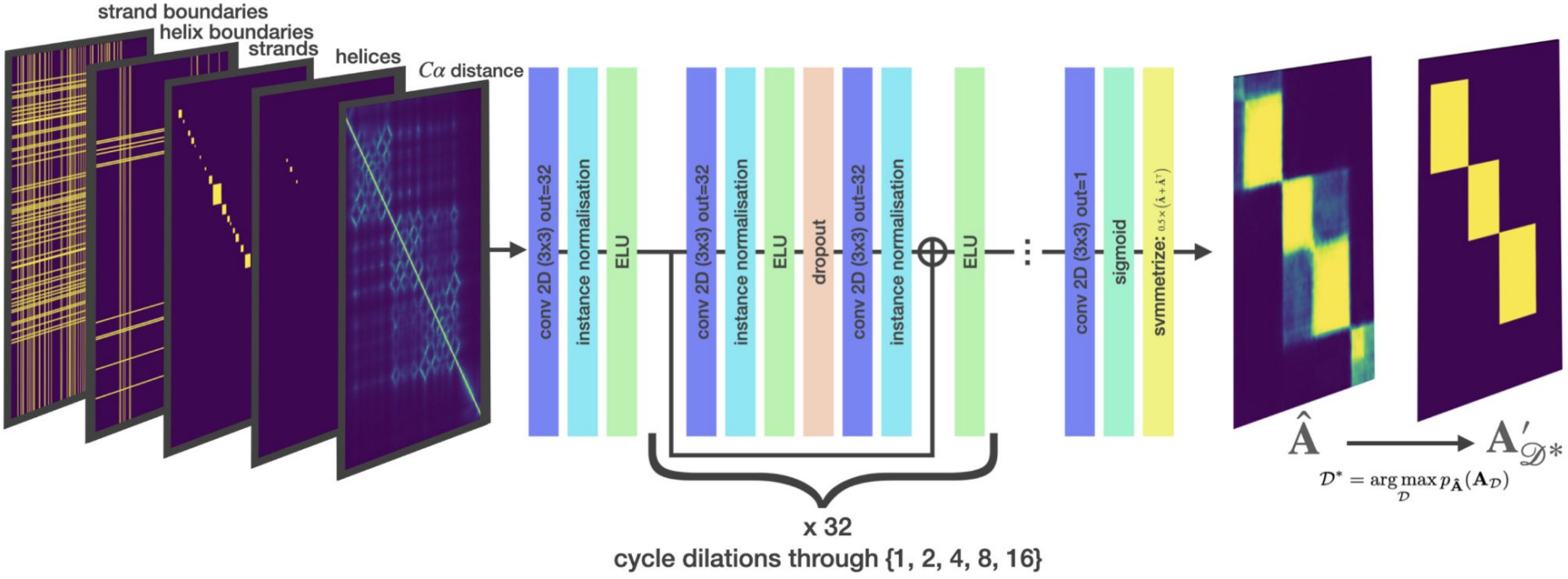
1. **The pairwise residue distance matrix** is an L×L matrix D where element dij is the distance, in angstroms, between the α-carbon atoms of residues i and j.
2. **Co-membership matrix** C where element cij is 1 if residues i, j are in the same secondary structure component, 0 otherwise.
3. **Boundaries matrix** indicates which residues occur at the start and end of secondary structure components with the first residue of a secondary structure component indicated with 1 and the last residue -1.

Each of the secondary structure representations is instantiated independently for helices and strands resulting in four secondary structure feature channels in total.



Network Architecture and Training Objective

- Fully **convolutional architecture with skip connections** (modified **trRosetta**).
- **Supervised learning problem as a 2D to 2D task:** transforming the 2D input features into a pairwise probability matrix which expresses the probability that pairs of residues are in the same domain.
- The learning objective is to minimize the **binary-cross entropy** of the predicted residue pairwise domain co-membership matrix (soft adjacency matrix) and the true adjacency matrix.
- Utilizes a **greedy algorithm** to find the most likely **domain assignments** based on the predicted co-membership probabilities.
- One advantage of this pairwise representation in both inputs and outputs lies in its **SE (3) invariance**, signifying that the representation remains unaltered under rigid **transformations of the 3D structure**, which consists of **rotations**, **translations**, and **reflections** within the original coordinate space.


strand boundaries
helix boundaries
strands
helices
Cα distance
conv 2D (3x3) out=32
instance normalisation
ELU
conv 2D (3x3) out=32
instance normalisation
ELU
dropout
conv 2D (3x3) out=32
instance normalisation
ELU
conv 2D (3x3) out=1
sigmoid
symmetrize: $0.5 \times (\hat{A} + \hat{A}^T)$
$\hat{\mathbf{A}}$
$\mathbf{A}'_{\mathcal{D}^*}$
$\mathcal{D}^* = \arg\max_{\mathcal{D}} p_{\hat{\mathbf{A}}}(\mathbf{A}_{\mathcal{D}})$
x 32
cycle dilations through {1, 2, 4, 8, 16}

Domain Assignment Algorithm

Chainsaw predicts an L×L matrix where each element represents the probability that residues i and j belong to the same domain.

The algorithm assigns residues to the domain that maximizes probability, iterating through all residues until optimal assignment is achieved.

Let $\mathbf{A}_{\mathcal{D}}$ be the binary matrix associated with a given domain assignment $\mathcal{D}$. To identify the most likely domain assignment $\mathcal{D}^*$ given a predicted set of co-membership probabilities $\hat{\mathbf{A}}$, we seek to find the $\mathcal{D}$ with maximum probability under $\hat{\mathbf{A}}$,

$$\mathcal{D}^* = \arg\max_{\mathcal{D}} p_{\hat{\mathbf{A}}}(\mathbf{A}_{\mathcal{D}}). \quad (1)$$

Let $\mathbf{A}_{\mathcal{D}}$ be the binary matrix associated with a given domain assignment $\mathcal{D}$. To identify the most likely domain assignment $\mathcal{D}^*$ given a predicted set of co-membership probabilities $\hat{\mathbf{A}}$, we seek to find the $\mathcal{D}$ with maximum probability under $\hat{\mathbf{A}}$,

$$\mathcal{D}^* = \arg,\max_{\mathcal{D}} p_{\hat{\mathbf{A}}}(\mathbf{A}_{\mathcal{D}}). \tag{1}$$

$$p_{\hat{\mathbf{A}}}(\mathbf{A}) = \prod_{i<j} p_{\hat{a}_{ij}}(a_{ij}) = \prod_{i<j} \hat{a}_{ij}^{a_{ij}}(1-\hat{a}_{ij})^{(1-a_{ij})} \tag{2}$$

Let $\mathbf{v}_1, \ldots, \mathbf{v}_K$ be a sequence of binary indicator vectors for each of the K (predicted) domains, let $\mathbf{V}_{\mathcal{D}}$ be the matrix whose columns are the $\mathbf{v}_i$, hence $\mathbf{V}_{\mathcal{D}}$ is a binary matrix with dimensions $L \times K$. The requirement that no residue can be assigned to more than one domain ensures that the rows of this matrix are K-dimensional one-hot vectors indicating domain assignments for each residue. Given this matrix of domain assignments, $\mathbf{V}_{\mathcal{D}}$, the elements a_{ij} of the adjacency matrix $\mathbf{A}_{\mathcal{D}}$ are generated as

$$\mathbf{A}_{\mathcal{D}} = \sum_{k=1}^{K} \mathbf{v}_k \mathbf{v}_k^T = \mathbf{V}_{\mathcal{D}} \mathbf{V}_{\mathcal{D}}^T. \tag{3}$$

Thus our maximization problem becomes to find a set of K vectors, such that the probability of domain assignment induced by the vectors is maximized

$$\mathbf{V}_{\mathcal{D}}^* = \arg,\max_{\mathbf{V}_{\mathcal{D}}} p_{\hat{\mathbf{A}}}(\mathbf{V}_{\mathcal{D}} \mathbf{V}_{\mathcal{D}}^T) \tag{4}$$

$$\mathcal{L}(\mathbf{V}, \hat{\mathbf{A}}) = \log p_{\hat{\mathbf{A}}}(\mathbf{V}_{\mathcal{D}} \mathbf{V}_{\mathcal{D}}^T) = \sum_{i<j} \log p_{\hat{a}_{ij}}(\mathbf{v}_i^\top \mathbf{v}_j), \tag{5}$$

$$\mathcal{L}_{j \to k}(\mathbf{V}, \hat{\mathbf{A}}) = \mathcal{L}(V', \hat{\mathbf{A}}) - \mathcal{L}(\mathbf{V}, \hat{\mathbf{A}}) \tag{6}$$

Algorithm 1 DomainAssigner($\hat{\mathbf{A}}$)

$K = K_{init}$ ▷ Set the initial number of domains (4 in our case)
$\mathbf{V} = \text{zeros}(L, K)$ ▷ Initialize domain assignments matrix with zeros
for $i < N_{iter}$ **do**
  for $j < L$ **do**
    $k^* = \text{argmax}_k \mathcal{L}_{j \to k}(\mathbf{V}, \hat{\mathbf{A}})$ ▷ Get highest scoring domain assignment for residue j.
    $\mathbf{v}_j \leftarrow 0$ ▷ Remove any previous assignment for residue j
    if $\mathcal{L}_{j \to k^*}(\mathbf{V}, \hat{\mathbf{A}}) > \mathcal{L}_{j \to \varnothing}(\mathbf{V}, \hat{\mathbf{A}})$ **then** ▷ Compare the proposed assignment k^* to assigning to no domain, $\varnothing$
      $v_{jk^*} \leftarrow 1$ ▷ Assign residue j to domain k^*
    end if
    if $k^* = \text{K}$ **then** ▷ If the best domain is the last one …
      $K \leftarrow K + 1$ ▷ … increase the total number of domains
      $\mathbf{V} \leftarrow \text{concat}(\mathbf{V},\ \text{zeros(L,\ 1)})$ ▷ Extend the assignment matrix with zeros
    end if
  end for
end for

Uncertainty Quantification

A natural approach to **uncertainty quantification** is to consider the **output** of the neural network **Â** as an **L×L multivariate Bernoulli distribution**. Then we can consider the output of the **final assignment A'** as an observation from the **Â distribution** and calculate the **likelihood** (normalized by the number of residues).

Chainsaw's confidence score has a good correlation with the ground truth accuracy.
Using a **confidence score cutoff of 0.85** will increase the probability that a domain is predicted correctly **from 0.78 to 0.9** at the expense of introducing a **5%** chance that a correctly predicted domain is **discarded**.

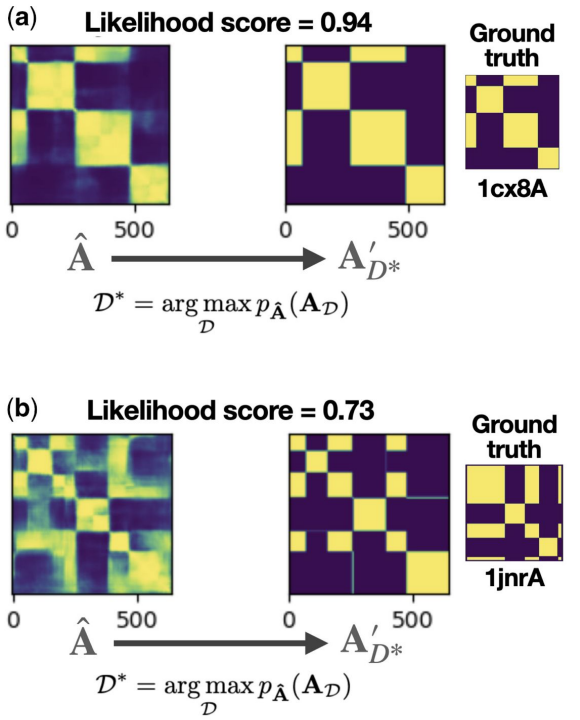


Figure 4. Chainsaw generates a confidence score which is the residue-averaged likelihood of A' under $\hat{\mathbf{A}}$ we find that this confidence measure has a good correlation with ground-truth accuracy. (a) High confidence model prediction where $\hat{\mathbf{A}}$ is very close to A'. (b) Low confidence prediction implies alternative assignments

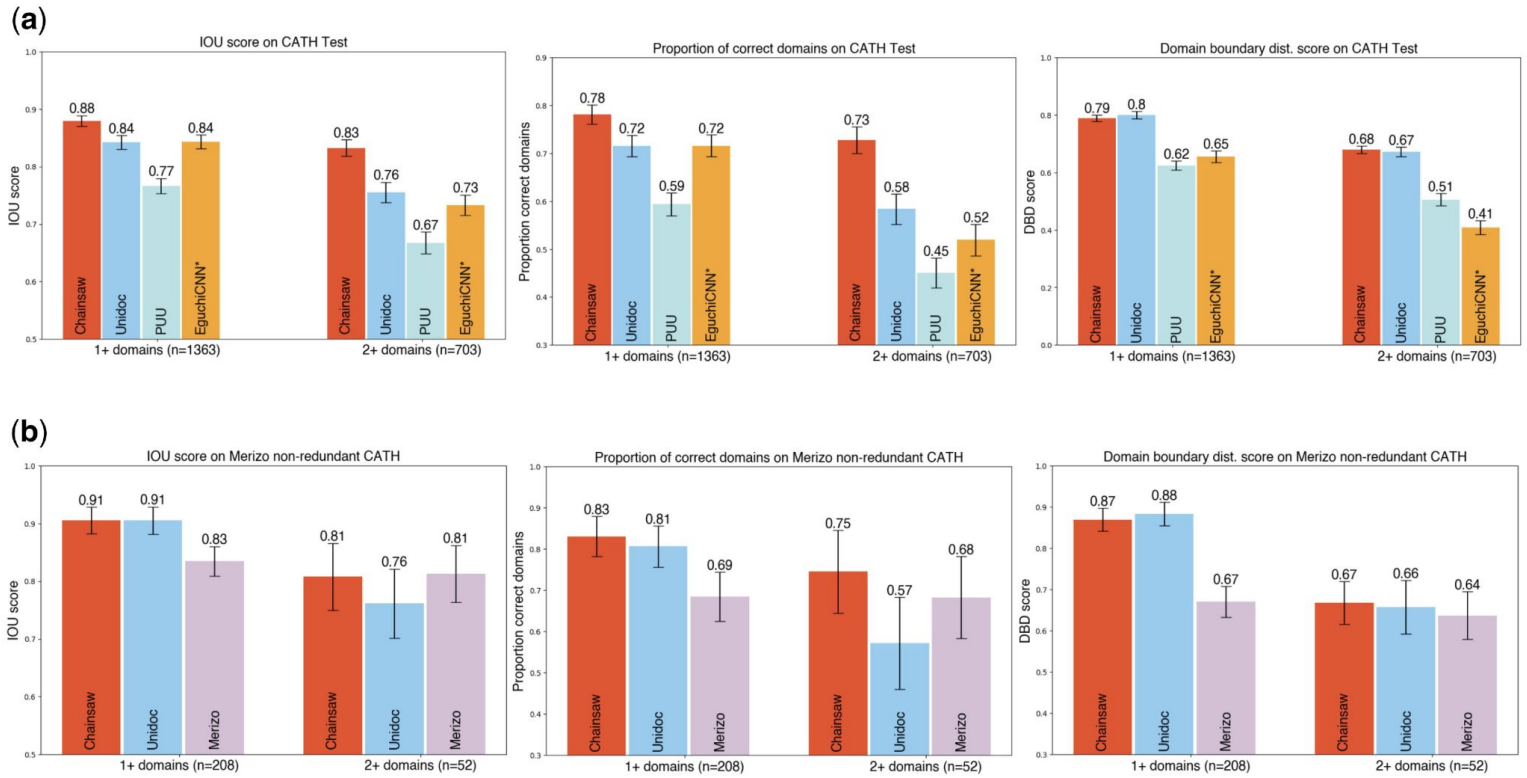
Results on PDB Structures and AlphaFold Models

Benchmarked against three unsupervised methods, **UniDoc**, PUU, and SWORD and two supervised methods, **Merizo** and EguchiCNN.

Benchmark dataset of **1365** protein chains from the PDB using **CATH** domain annotations as ground truth.

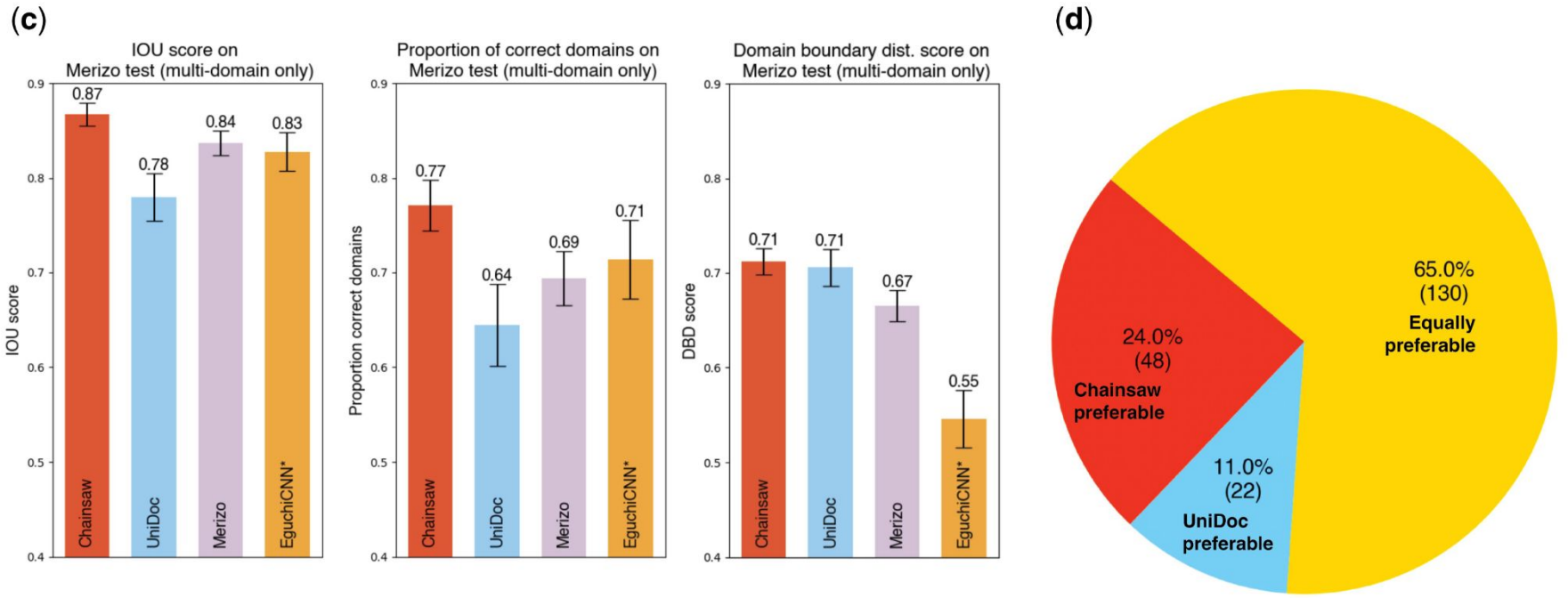
Metrics

- **Average intersection over union (IoU)** between paired sets of predicted and ground-truth residues assigned to each domain.
- **Proportion of correctly parsed domains** (domain-level IoU ≥ 0.8)
- **Domain boundary distance score**
    - Each boundary is scored independently. Predicting within 1 residue of the true boundary scores 8 points, within 2 residues scores 7 and so on until the distance is 9 residues or more, at which point the score is 0. Each boundary score is divided by 8 so that scores per boundary are between 0 and 1. The final boundary distance score for the entire chain is then calculated as the sum of individual predicted boundary scores divided by the total number of domain boundaries.



(a) Benchmarking on the Chainsaw CATH test set.

(b) Benchmarking on the subset of the Chainsaw CATH test set that is non-redundant with the Merizo training data.



(c) Performance comparison on the Merizo test data. For these results, we trained a Chainsaw model from scratch using the Merizo training data and subsequently evaluated on the same test data as Merizo.

(d) Results from visually assessing Chainsaw and UniDoc domain parsings on 200 randomly selected AlphaFold models from the human proteome

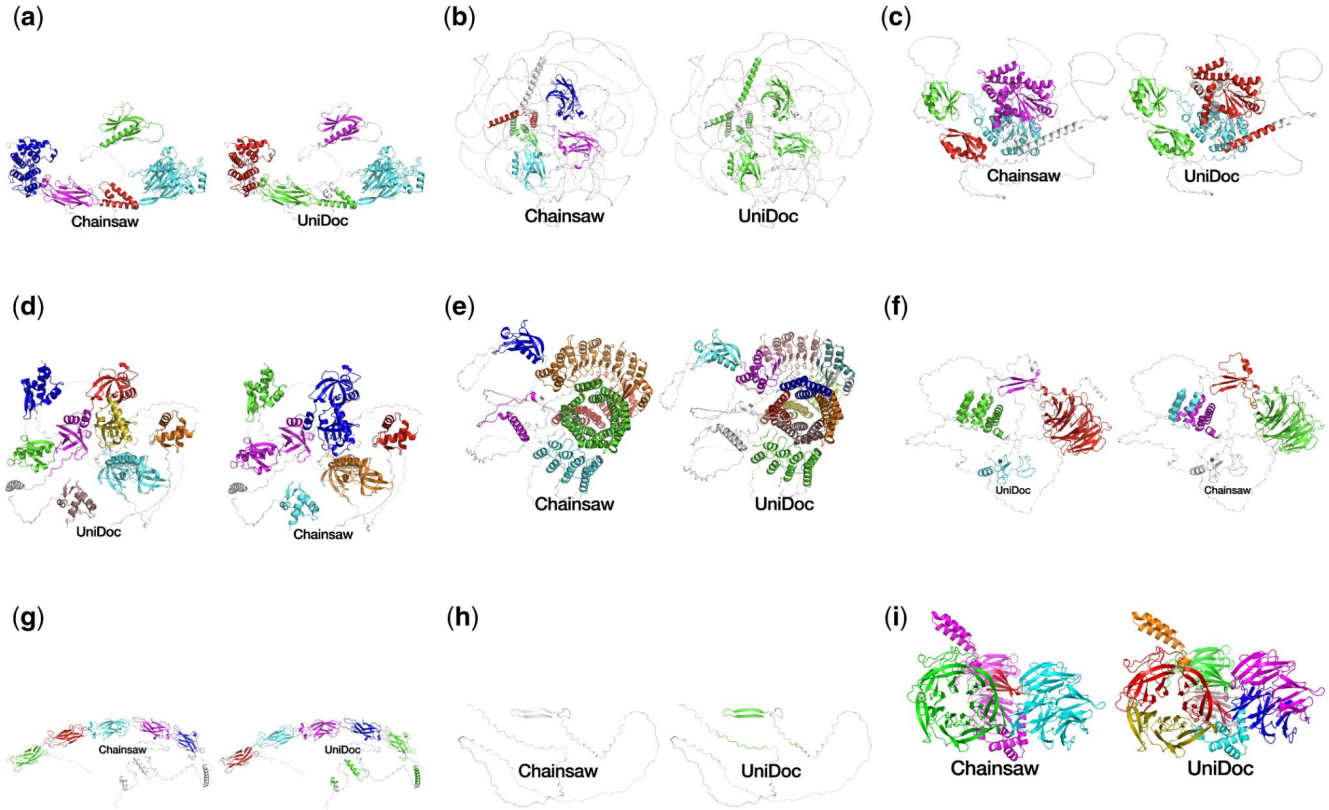


Figure 7. A selection of judgements from the blind comparison of UniDoc and Chainsaw on 200 AlphaFold human structures. Domain segmentations and judgements for all 200 proteins can be found in the Supplementary Material. (a) Under-splitting by UniDoc (right) in green domain. (b) Under-splitting by UniDoc (right) in green domain. (c) UniDoc (right) shows under-splitting in the green domain and an under-trimmed boundary in the red domain. (d) Chainsaw (right) shows under-splitting in the blue and magenta domains. (e) UniDoc (right) shows over-splitting of the α/β horseshoe domain identified by Chainsaw (left) in orange. (f) Chainsaw (right) shows over-splitting of the single domain which has been split into cyan and magenta. (g) Under-trimmed boundary by UniDoc (right) in the green domain. (h) UniDoc (right) identifies a domain where there should be none. (i) UniDoc (right) over-splits two propeller domains identified by Chainsaw (left) in green and cyan.

Chainsaw on 3 Downstream Tasks

1. **Identifying domains** in the **PDB** that have **not yet been annotated** in the **CATH** protein domain database.
2. **Identifying domains** in the **AFDB** that can be **matched to representative domains** found in **CATH**
3. Showing that Chainsaw can be used to **infer novel functional annotations** for previously uncharacterized proteins.

In each of these cases, we first generate domain predictions using Chainsaw and subsequently use the Foldseek structure and sequence matching algorithm to match predicted domains against a library of known CATH domains (clustered at 60% sequence identity).

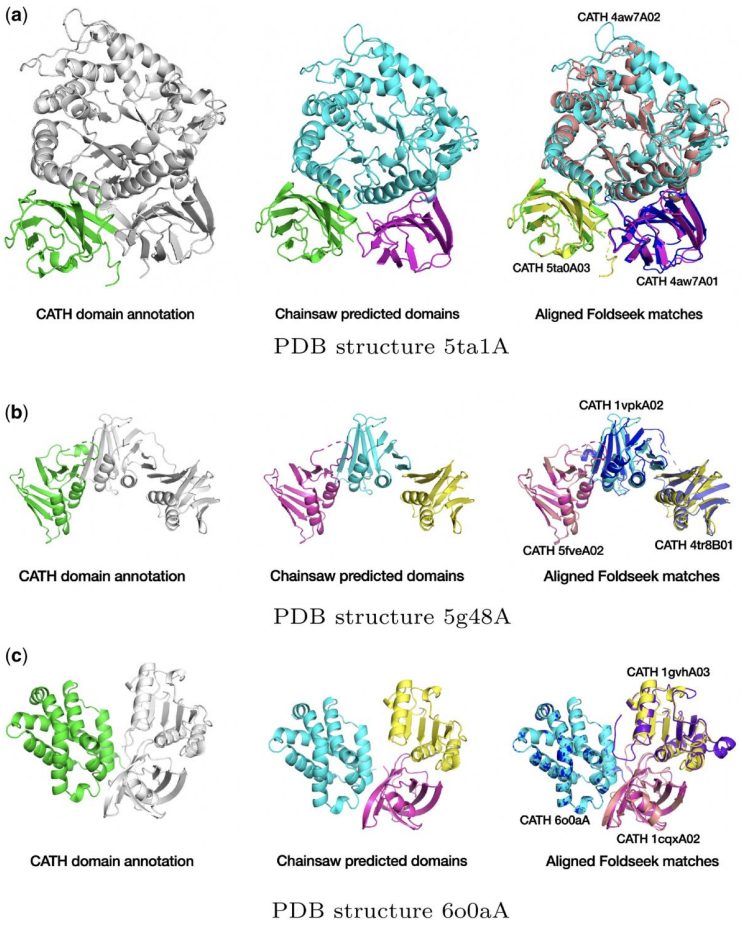


Figure 8. Example PDB structures where Chainsaw combined with FoldSeek can identify two additional domains which have not yet been annotated by CATH. In each case, the single CATH-identified domain is shown on the left, Chainsaw's domain prediction is shown in the middle, and the matched CATH domains aligned with the Chainsaw predictions are shown on the right. (a) PDB structure 5ta1A. (b) PDB structure 5g48A. (c) PDB structure 6o0aA

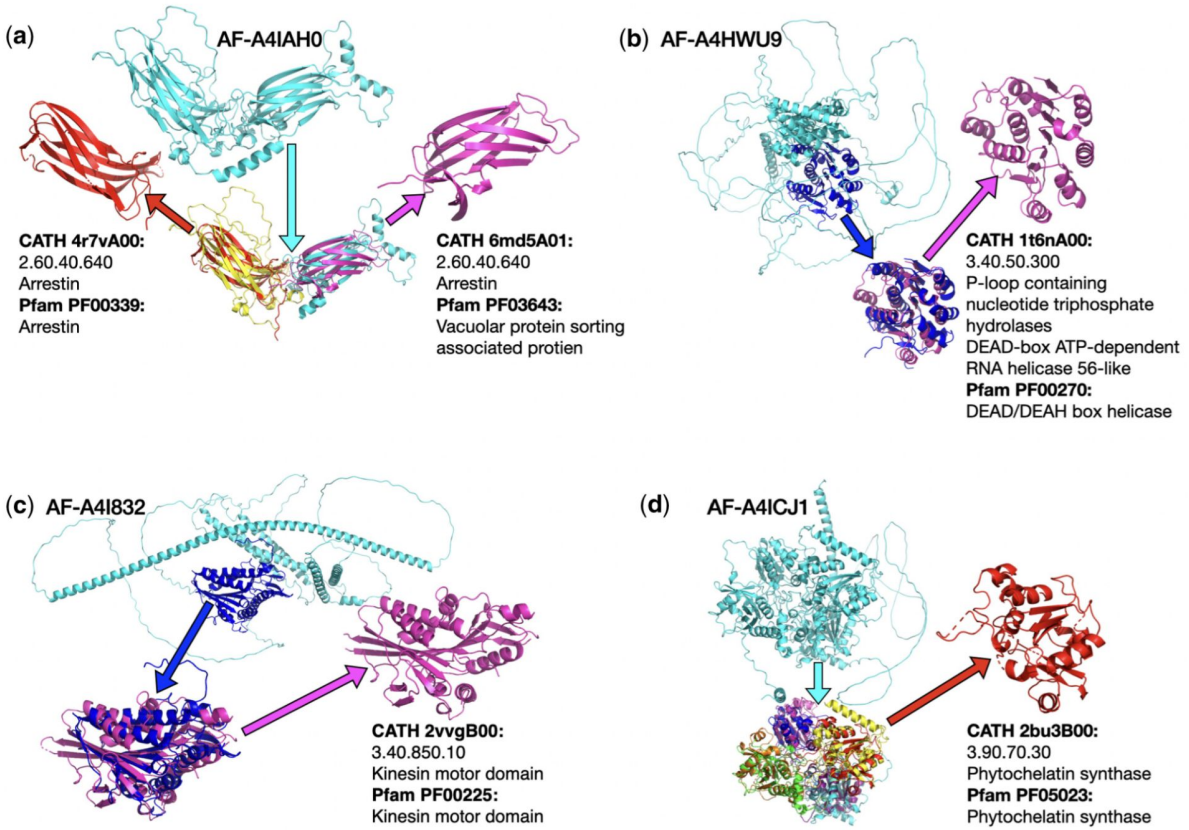


Figure 9. Uncharacterized proteins (no Pfam or Go annotations) from the *L.infantum* proteome were parsed with Chainsaw. The predicted domains were subsequently searched against the CATH S60 domain representative structures using Foldseek. We show four examples (a, b, c, d) where we can infer novel functional annotations via structural homology with representative CATH domains. Sequence identity for the matches above ranges from 11% to 16%, which indicates why these homologous relationships were not detected with sequence-only methods. Figure d Shows that Chainsaw correctly parses the structure into four phytochelatin synthase domain repeats. This protein is common to multiple pathogenic organisms and has been considered a potential drug target due to the fact it has no human homolog (Ray and Williams 2011)

Ablation Studies

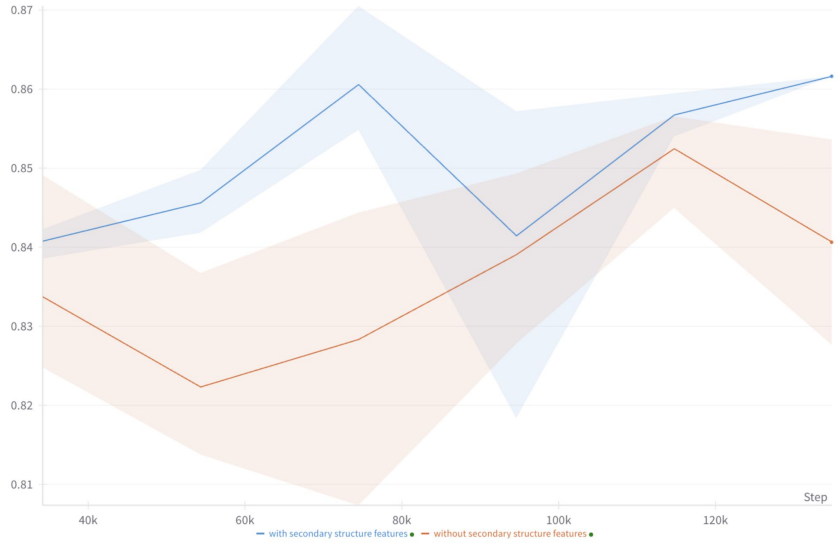


Figure 10. Results from training six Chainsaw models, three with secondary structure features included and three without (distance matrix only). We show the grouped mean IoU score on the validation data. The shaded range covers the minimum and maximum values at each epoch.

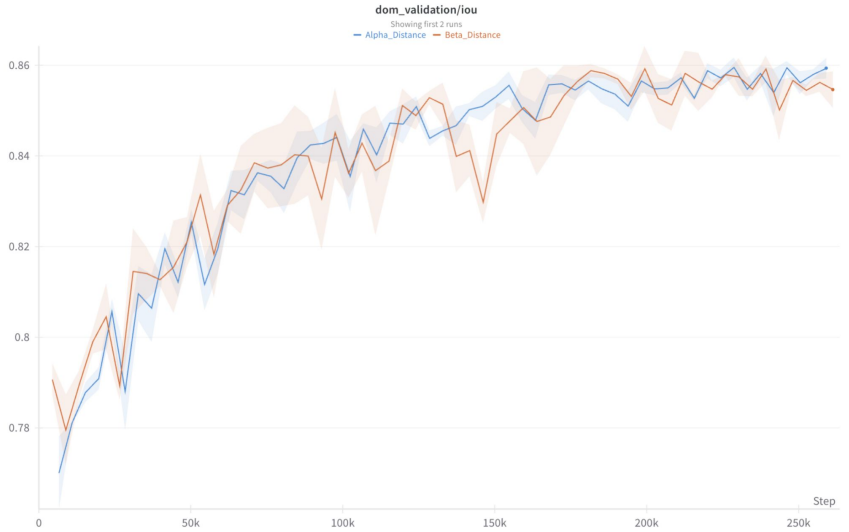


Figure 11. Results from training six Chainsaw models, three with α-carbon distances and three with β-carbon distances. The shaded range covers the minimum and maximum values at each epoch.

Conclusion

- Chainsaw is a powerful new method for **domain segmentation**, outperforming state-of-the-art methods.
- Its ability to handle **discontinuous domains** and offer **flexible assignments** makes it ideal for future structural bioinformatics applications.
- Chainsaw's integration with existing databases and structure matching tools like Foldseek opens new possibilities for **discovering novel protein functions**.
- Evaluations revealed cases where **UniDoc** outperformed Chainsaw, suggesting the benefit of an **ensemble approach** that combines predictions from multiple models.
- Incorporating **confidence measures** to highlight **ambiguous cases**.

Thanks for listening!

Questions?