Supporting Information

TopDomain: Exhaustive Protein Domain Boundary Meta-Prediction Combining Multi-Source Information and Deep Learning

Daniel Mulnaes[1], Pegah Golchin[1], Filip Koenig[1], and Holger Gohlke[1,2]*

[1]Institut für Pharmazeutische und Medizinische Chemie, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany

[2]John von Neumann Institute for Computing (NIC), Jülich Supercomputing Centre (JSC), Institute of Biological Information Processing (IBI-7: Structural Biochemistry) & Institute of Bio- and Geosciences (IBG-4: Bioinformatics), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany

Author ORCID

Daniel Mulnaes: 0000-0003-2162-5918

Pegah Golchin: 0000-0002-8686-113X

Filip Koenig: 0000-0003-0852-440X

Holger Gohlke: 0000-0001-8613-1447

*Address: Universitätsstr. 1, 40225 Düsseldorf, Germany.

Phone: (+49) 211 81 13662; Fax: (+49) 211 81 13847

E-mail: gohlke@uni-duesseldorf.de

Text T1: TopDomain$_{TMC}$

For TopDomain$_{TMC}$, the final goal is to classify if a sequence needs to be cut into domains at the domain boundaries or not, given that a specific template was identified. Cutting the sequence is deemed unnecessary if the template covers each inter-boundary sequence segment by at least 80% (good coverage of all domains) and has a TM-Score larger than 0.5 (good over-all arrangement of domains relative to each other). If these criteria are not met, cutting the sequence into domains is deemed necessary for proper modeling.

For training TopDomain$_{TMC}$, we extract templates from the primary threaders used by TopDomain, and for each template, we calculate the agreement between the template and the linear features predicted from the target sequence. The calculated agreements include secondary structure (Q3 agreement with MuFoldSS [1]), ϕ/ψ torsion angles (MAE agreement with SPIDER3 [2]), solvent accessibility (Pearsons R^2 agreement with SANN [3]), and contact prediction (Precision, Recall and F1 score agreement with PCONSC4 [4]) as well as the agreement with the PSSM. These agreement features are combined with three basic features: Sequence identity, sequence similarity, and sequence coverage. These features are used as input for a fully connected feed-forward DNN with 15 layers, determined by a grid search, to predict the TM-Score between the template and the native structure, which is trained on the TopDomain training dataset. For each target, all identified templates are evaluated and ranked according to their predicted TM-Score. For each template found by multiple different threaders, the maximum predicted TM-Score, coverage, and sequence identity is used for that template.

For each threading alignment generated for each template, the coverage of each inter-boundary segment is calculated based on the predicted boundary positions from TopDomain. To make the final decision, the minimum, maximum, and mean inter-boundary segment coverage is calculated. These are used as features, along with sequence identity, coverage, and predicted TM-Score for a fully connected feed-forward DNN with 16 layers (determined by a grid search) trained to perform binary classification for that template ("Parse" or "Don't Parse"). For each template, the true decision is calculated based on pairwise alignment with TM-Align, the true TM-Score based on TM-Align, and the true boundaries. If any template is identified that gets the label "Don't Parse", then the overall prediction for the protein is "Don't Parse", otherwise it is "Parse".

The overall workflow of TopDomainTMC can thus be summarized in three steps:

I. Extract templates and agreement with sequence features from primary threaders and primary predictors. Use these features to predict a TM-Score for each template.

II. Calculate the inter-segment coverage for each template based on the boundaries predicted by TopDomain and calculate minimum, maximum, and mean coverage of segments.

III. Calculate the "Parse/Don't Parse" prediction based on inter-segment coverages, sequence identity, coverage, and predicted TM-Score for each template. Assign the final prediction as "Don't Parse" if any template gets this classification, otherwise assign the final prediction as "Parse".

Text T2: TopDomain Dataset

Data basis. For domain boundary prediction, one of the key issues is data availability. Resolving large protein structures, especially those with disordered regions or multiple domains, is experimentally challenging [5-8]. Therefore, the number of large multi-domain proteins that have an experimentally verified structure is limited. Databases such as CATH [9] and Astral SCOPe [10] seek to annotate domain boundaries accurately. Both databases annotate domain boundaries in released structures in the PDB [11] using a combination of automated and manual annotation. The main difference between the two is that CATH contains both domains annotated from structures and annotations from sequences with no available structure, but for which domains can be predicted by the Gene3D [12] software. By contrast, Astral SCOPe contains only annotations from structures. Gene3D uses HMM comparison to match structure-annotated domains to sequences with no annotation. Despite these efforts, there are limitations to these databases.

First, there is a large degree of redundancy, and the number of genuinely non-redundant multi-domain proteins in the PDB is much smaller than the CATH or ASTRAL-SCOPe database sizes (122727 and 59514, respectively) would suggest. This is evident because once clustered to 70% sequence identity, the number of multi-domain proteins in, e.g., Astral SCOPe is less than 2700 as of December 2019, a number that shrinks even more if a more stringent criterion for redundancy is used. Second, because of the limitations of tools used for domain annotation (HMM comparison and structure-based tools such as DDOMAIN [13]), discontinuous domains are generally poorly annotated despite making up a significant portion of proteins. An estimated 18% of structures in the PDB have at least one discontinuous domain according to DomainParser2 [14], and 15% of CATH domains are annotated as discontinuous [15]. Yet, many ASTRAL-SCOPe annotations do not contain discontinuous domains, and DDOMAIN, which is parameterized on ASTRAL-SCOPe, predicts only continuous domains. Because the CATH database contains automated annotations, human annotations, and sequence-based predicted

domains, we decided to use the ASTRAL-SCOPe database as a starting point. This is because we do not wish for TopDomain to predict the output of Gene3D (Gene3D is a primary predictor for InterProScan and, therefore, also a primary predictor for TopDomain).

TopDomain dataset. Because of the large set of primary predictors used in TopDomain, the training dataset has to be of limited size but high quality. Therefore, all multi-domain proteins from the Astral SCOPe database were downloaded and clustered at 70% pairwise sequence identity using MMSeqs2 [16]. These proteins were then re-annotated by manually inspecting the Astral SCOPe annotations as well as structure-based domain predictions from DomainParser2 [14] and DDOMAIN [13]. During this inspection, particular care was taken when annotating discontinuous domains. A reasonable topology of discontinuous domains requires that the domain insertion site is biologically feasible, thus, the two attachment points should be nearby in the discontinuous domain [17]. Furthermore, many of the multi-domain structures were resolved by modification of the target sequence, *i.e.*, the removal of domains or disordered regions to make crystallization easier [6]. Therefore, the genomic sequence was aligned to the sequence of the resolved structure using MAFFT7 [18], and large gaps in the genomic sequence (corresponding to missing regions in the structure) were annotated as follows:

If a piece of the genomic sequence larger than 40 residues was missing in the structure, this region was annotated with boundaries on either side of the missing region. This effectively assigns large missing regions as putative, potentially disordered domains, which were likely removed to enable crystallization or not visible in the crystal. Regions smaller than 40 residues were annotated depending on the protein topology as either a disordered loop (no boundary between domains) or a linker between two domains (one boundary in the middle of the missing region). In this alignment, gaps in the target sequence correspond to the artificial fusion of proteins in order to enable crystallization (such as fusion of T4 lysozyme with GPCRs [19]) and were therefore ignored.

Furthermore, because TopDomain should learn the location of domain boundaries, after initial training of the stage 2 DNN for TopDomain with a window size of 40 residues once on the entire dataset, the predictions were used to validate the human annotations. After the DNN annotation, each of the proteins in the dataset was manually re-inspected to rectify mistakes in the initial annotations. Upon careful inspection, false negatives (when the DNN predicted a true boundary that human annotation had initially missed) and false positives (when the human annotation had incorrectly placed a boundary) in the initial manual annotations were manually rectified. This was done to ensure the high quality of the domain boundary annotations in the dataset.

Multi-domain proteins are the most important source of information for a predictor of protein domain boundaries. Hence, multi-domain proteins were clustered at 70% sequence identity to retain a sufficient number of structures for training. However, it is essential to include single-domain proteins in the training dataset to avoid that the neural network learns to predict boundaries in all proteins. Different estimates of the ratio between single-domain and multi-domain have been calculated in the past, ranging from single-domain proteins comprising 20-35% in prokaryotes and 35-60% in eukaryotes [20, 21]. However, due to single-domain proteins being easier to crystallize, these comprise a much higher fraction of the proteins in structural databases like the PDB and Astral SCOPe. To balance the dataset and include a reasonable amount of diverse single-domain proteins, we, therefore, analyzed proteins annotated as single-domain in the Astral SCOPe dataset and pre-processed them in the same way as the multi-domain proteins, with two main differences: (1) Only proteins with a perfect agreement between the resolved structure and the protein sequence were selected. (2) Proteins were clustered at 20% pairwise sequence identity instead of 70% in order to retain a small but very different number of single-domain proteins. We then manually inspected these single-domain proteins in the same way as the multi-domain proteins. Surprisingly, we found that about 10% of the proteins annotated as single-domain in Astral SCOPe were multi-domain proteins incorrectly annotated as a single domain. We, therefore, added these multi-domain proteins as well as 1035 verified single domain proteins to the already annotated multi-domain proteins. The final TopDomain dataset thus consists of 3105 multi-domain proteins and 1035 single-domain proteins.

Training and test set separation. A pairwise sequence identity of 70% between multi-domain proteins still leaves a significant redundancy in the TopDomain dataset. Therefore, we clustered the dataset at 20% pairwise sequence identity using MMSeqs2 [16] and split it into training and test dataset according to clusters. This split was made using an in-house multiple steepest descent algorithm that optimizes the similarity between the two data splits while enforcing no two proteins from the training and test set to share more than 20% identity. The parameters that were optimized were protein size distribution, distribution of the number of effective sequences (N_{eff}) [4], and the distribution of the number of domain boundaries per protein. The latter also balances the number of single-domain proteins since this keeps the number of proteins with zero boundaries similar in the two datasets. That way, the training and test set are similar in terms of protein size distribution, the difficulty of predictions (N_{eff} indicates available sequence information), and boundary number distribution. Moreover, the training and test sets are very dissimilar in terms of homology between the two datasets, as no two proteins between the

datasets share more than 20% sequence identity. By contrast, within each split, the sequence identity for multi-domain proteins may be as high as 70%. These two datasets are termed the TopDomain training set and TopDomain test set, respectively.

Since TopDomain predicts domain boundaries using a sliding window, in which each window is a separate sample during training, each residue is effectively one data point for training. This differs from methods that train on the entire protein at once and use zero padding to ensure that the input vectors have the same size despite different protein sizes. When using a sliding window, all residues in the dataset are predicted in a random order, and no information from residues outside of the sliding window is used. That way, although the data set is split according to proteins, no whole-protein patterns can be memorized by the DNNs, which prevents over-fitting. This also helps the DNNs generalize to domain architectures not seen during training (*e.g.*, to consider different combinations of different domains in different orders). The distribution of the number of boundaries in a protein for different protein lengths across the TopDomain dataset is shown in Figure S1.

Figure S1. TopDomain Dataset Boundary Distribution.

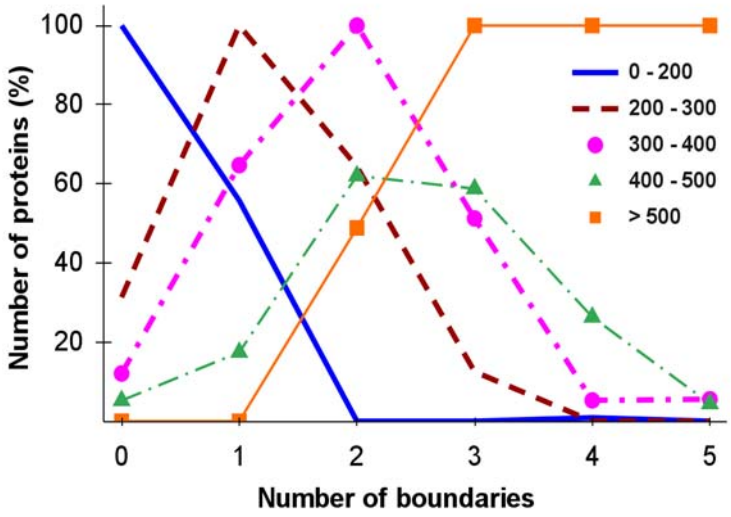

Figure S1. The x-axis shows the number of true boundaries in a protein. The y-axis illustrates the normalized number of proteins available in the training set, which have various sequence lengths.

Text T3: Homology Features

One group of primary predictor features includes template information from threading results obtained by running RPS-BLAST [22] on the conserved domain database [23], running DELTA-BLAST [24] on the ASTRAL-SCOPe database [10], as well as running pGenThreader [25], pDomThreader [25], FFAS03 [26], RAPTORX [27], SPARKSX [28], and HHSEARCH [29] against the PDB [11]. These threading programs are used to identify potential conserved domains in the target sequence based on template matches found in structure databases. The identified templates are used as input for structure-based boundary prediction using DDOMAIN [13], DomainParser2 [14], and SWORD [30], from which the resulting boundary predictions are mapped back to the target sequence using the threading alignment. Similar mapping of other structural features of the templates is also used as primary features. These include 3-state secondary structure (α-helix, β-strand, and coil), ϕ/ψ angles, and relative solvent accessibility calculated by DSSP [31]. The structural features from DSSP are combined into weighted averages across all templates using the sequence identity of each template as a weight.

For each of the threading methods mentioned above, the identified templates are clustered at 90% sequence identity using CD-Hit [32] to reduce redundancy. The distribution of threading scores is then analyzed to select the top-ranking templates based on the second derivative of the score vs. rank curve. This scoring, termed inflection-point filtering, removes poorly scoring templates if much higher scored ones are found, but does nothing if all identified templates have similar scores or if the scores decrease gradually without any sudden drop in the score at a certain rank. Additionally, the templates are ranked according to how much the template provides new coverage compared to higher ranked templates. This is done to ensure that lower-ranked templates are not removed if they cover a part of the target sequence that is not covered by any of the higher-ranked templates. Further filtering includes removing templates for which sequence identity to the target is less than 10%, where the e-value is larger than 0.01, or where the size of the match is less than 40 residues long. After filtering and re-ranking, the top 20 templates from each threading program are kept for feature extraction.

Template-based Features. For each template-based primary predictor (ThreaDom [33], InterProScan [34], DomPred [35], FIEFDom [36], RPS-BLAST [22], DELTA-BLAST [24], pGenThreader [25], pDomThreader [25], FFAS03 [26], RAPTORX [27], SPARKSX [28], and HHSEARCH [29]), care is taken to ensure that an upper limit to the sequence identity between the target sequence and any identified template structure or template-based database match can be imposed. This is a critical feature for benchmarking to simulate the absence of closely homologous templates for a *de novo* prediction. Without this feature, template-based methods would simply find the target

structure in a database and infer the boundaries from the known structure. Imposing different cut-offs to sequence identity during benchmarking enables us to train the DNNs to balance the importance of template-based and *ab initio* features, depending on the availability and similarity of template information. In short, the DNN should learn how much to trust template-based information and how much to trust sequence-based information depending on the availability and quality of the template-based information.

Text T4: Sequence Features

In addition to template-based features, sequence features predicted from the target sequence are used. The rationale behind this is that domain boundaries and protein domains are highly diverse. Therefore, information about the target protein will help predict domain boundaries, even if this information is not initially intended for boundary prediction. To account for differences and inaccuracies in predicted features from a single method, multiple different methods are used for each feature to allow the neural network to learn from a diverse set of predictions. The predicted features can be divided into the following categories:

1. Solvent accessibility predictions from SANN [3] and SPIDER3 [2]. These features are useful since many domain boundaries are located in solvent-exposed linkers between domains. Thus, the prediction of exposed residues should improve boundary identification.
2. Secondary structure predictions from MuFOLD-SS[1], SPIDER3 [2], and DeepCNF-SS [37], and dihedral angle predictions from SPIDER3. These features are useful since domains often belong to different fold classes such as α-fold, β-fold, α/β-fold, and α+β-fold. Furthermore, many domain boundaries are located between secondary structure elements. Thus, the prediction of secondary structure and dihedral angles should improve boundary identification.
3. Residue disorder prediction from DISOPRED [38], DeepCNF-D [39], GlobPlot [40], and MobiDBLite [41]. These features are useful since many domain boundaries are located in disordered regions. Furthermore, some protein and protein domains are intrinsically disordered. Thus, the prediction of disordered regions should improve boundary identification.
4. Transmembrane topology predictions from PHOBIUS [42], TMHMM [43], and BOCTOPUS [44] and signal peptide predictions from SignalP [45]. These features are useful since many trans-membrane proteins include both trans-membrane- and globular

domains. Thus, the prediction of transmembrane regions should improve the separation of transmembrane domains from globular domains.

5. Protein repeat predictions from TRUST [46], T-REKS [47], and HHrep [48] These features are useful since many proteins contain repeating domains with the same fold. Thus, identifying repeating sequence units should help to identify the boundaries between repeating domains. Furthermore, solenoid protein domains are highly repetitive. Thus, the identification of highly repetitive regions should improve the identification of solenoid domains.
6. Coiled-coil predictions from COILS2 [49] and DeepCoil [50]. These features are useful since many proteins contain coiled-coil domains. Thus, the identification of coiled-coil regions should improve the identification of such domains.
7. PSSM calculated from the combined alignment of MetaPSICOV [51] (HHBLITS [52] against the UniClust30 [53] database), DNCON2 [54] (HMMER3 [55] against the UniRef90 [56] database), and CONDO [57] (HMMER3 against the non-redundant sequence database). The PSSM is calculated using Henikoff-Henikoff re-weighting [58]. The PSSM features are useful since different domains experience different evolutionary pressure to conserve different patterns of residues. Thus, the PSSM should improve the separation of different domains.
8. From the combined alignment, sequence termini propensities and gap propensities are calculated, and weighted propensities are calculated using the sequence identity between each sequence and the target as a weight. These features are useful since domain linkers often vary in length, which causes linker regions in the MSA to contain more gaps than domain regions. Furthermore, the location of sequence termini in partial matches may indicate single domain matches. Thus, identifying regions with high gap content and high termini content should improve domain boundary identification.
9. Residue contact predictions from MetaPSICOV [51] (including the primary predictors PSICOV [59], CCMPRED [60], and EVFOLD [61]), DNCON2 [54], DeepCov [62] , and PCONSC4 [4]. DeepCov predictions are calculated from alignments generated by MetaPSICOV [51], DNCON2 [54], and ConDo [57]. PCONSC4 predictions are calculated from the combined alignment. These features are useful since domains have high intra-domain coevolution compared to inter-domain coevolution. Thus, the coevolution signal should help detect sequence regions with low coevolutionary signals between residues on either side as potential domain boundary regions. Three sliding windows are used to decompose a contact map into a 1D vector. This decomposition was done by summing the

coevolution scores between residues on one side of the sliding window and the other. The sliding window sizes are ± 20, ± 40**,** and ± 80 residues to cover intra-domain coevolution for varying domain sizes.

10. The number of effective sequences (N_{eff}) and the normalized number of effective sequences for the target sequence. This is calculated from the number of sequences in the combined alignment (from step 9) and the total sequence length as in [63] These features seek to quantify the amount of sequence information available and, thus, indicate the degree of difficulty for the prediction.

While some primary features are relevant for all types of proteins (such as secondary structure, solvent accessibility, and residue contacts), others may only be relevant for specific types of proteins (protein repeats, residue disorder, coiled-coil regions, or transmembrane topology).

Text T5: Feature Conversion

For many primary predictors, particularly template-based ones, the output of a predictor is a set of boundaries. To turn these predictions into a feature vector with the length of the target sequence, each primary predictor output is used to calculate a feature score S_r for each residue r in the target sequence. S_r is defined as 0 when no boundaries are found within ±20 residues of r and is otherwise a double-weighted sum of distances normalized as given in Equation S1.

$$S_r = 1 - \frac{1}{N \cdot D_{min}} \sum_{D_i \leq D_{min}} D_i \cdot C_i^2 \cdot e^{-\frac{1}{2}\left(\frac{D_i \cdot k}{D_{min}}\right)^2} \qquad \text{Equation S1}$$

S_r is 1 when r perfectly overlaps with all N boundaries within ± 20 residues, corresponding to a weighted mean boundary distance of zero. D_i is the residue distance from residue r to boundary i. D_{min} is 20 residues; hence, the sum is over boundaries within ± 20 residues of r. C_i is the confidence of boundary i. The value k determines the relative importance of distance vs. confidence. Boundaries further away from r have a larger impact on S_r the lower k is. For TopContact, k is set to 5.

The weighting of S_r has two components: The first component is C_i, which is squared to put a high weight on confident boundaries (e.g., boundaries of templates with high sequence identities). For template-based predictions, C_i is the sequence identity of the given template; otherwise, it is the confidence given by the primary predictor. If no confidence score is provided by a predictor, C_i is set to 1 for all boundaries of that predictor. The second component is a distance-based weight, which follows a Gaussian distribution. This weight is 1 when D_i is 0 and

0.04 when D_i is 10. Thus, given equal C_i, only close-by boundaries (± 10 residues) contribute markedly to S_r.

Applying Equation S1 to a set of boundaries predicted by a given primary predictor for a given input protein gives a feature vector with the same length as the protein irrespective of how many boundaries were predicted (*e.g.*, due to a different number of templates identified by a threader or alternate boundary predictions from the same predictor). Furthermore, the boundary score considers the confidence of the boundaries and the agreement between different predictions.

Text T6: Stage 2 DNNs

Each of the Stage 2 DNNs is a residual neural network (ResNet) as implemented in [64] with 18 layers. Further information about each layer is available in Table S2.

As the number of boundary residues (distance to true boundary = 0 residues) and putative boundary residues (distance to true boundary ≤ 20 residues) is considerably lower than that of non-boundary residues (distance to true boundary > 20 residues) in the TopDomain training set, we face a severe class imbalance. To resolve this issue in Stage 2 (see section TopDomain Stage 2 in the main text), we use oversampling on the training split of the TopDomain training dataset to train on a balanced set of classes in each batch. Each batch consists of 1200 input images (200 images for each of the six distance bin classes), which were chosen randomly and uniformly (see section TopDomain Stage 2 in the main text). However, we do not use oversampling for the validation split of the TopDomain training set, which, therefore, still has the class imbalance seen in the raw data. This prevents overfitting and learning the statistics of the training dataset.

Early stopping is used to choose the number of epochs to not lead the network to over- or under-fit. The training is stopped once the model performance does not improve for 15 epochs on the validation set.

To increase the performance of optimization and decrease the training time, we use a learning rate schedule that decreases the learning rate by a factor of 0.2 once the validation loss stops improving for five epochs. The minimum learning rate is 10^{-5}.

The model uses categorical cross-entropy as a loss function [65] to decide to which of the six possible classes (see section TopDomain Stage 2 in the main text) each input image (see section TopDomain Stage 2 in the main text) belongs. Softmax [65] is the proper activation

function to be used with a categorical cross-entropy loss function because it transforms the model output into a vector with ranges from 0 to 1 and makes the output categorical. In turn, this allows the loss function to compare the probability distributions of model output and true targets [66]

The Adam optimization algorithm [67] is used to update network weights iteratively based on the training data. The exponential decay rate for the first-moment estimate is 0.9 and the second-moment estimate is 0.999. To prevent division by zero, epsilon is chosen as 10^{-8}.

Table S1. Stage 3 Filtering Score Cut-offs [a]

Predictor[b]	**Stage 2 Cut-off (Window ± 10)**	**Stage 2 Cut-off (Window ± 20)**	**Stage 2 Cut-off (Window ± 40)**
TopDomain	0.01	0.04	0.16
ThreaDom	0.51	0.38	0.43
InterProScan	0.32	0.31	0.73
DOMPRED	0.36	0.31	0.91
FIEFDom	0.18	0.20	0.18
$\text{TopDomain}_{\text{Seq}}$	0.16	0.03	0.09
ConDo	0.63	0.32	0.35
DOBO	0.23	0.23	0.16
DeepDom	0.79	0.80	0.76
DROP	0.20	0.26	0.22
PPRODO	0.80	0.99	0.96
DOMCUT	0.71	0.49	0.38
Scooby-Domain	0.34	0.97	0.15
DOMpro	0.19	0.23	0.18
$\text{TopDomain}_{\text{Parse}}$	0.44	0.55	0.30
DDOMAIN	0.29	0.30	0.32
DomainParser2	0.26	0.34	0.29
SWORD	0.10	0.30	0.68

[a] Stage 2 cut-offs for the non-contact probability used to calculate the Stage 3 filtering score. If a residue has a non-contact probability lower than the respective cut-off for each of the three window sizes (see section TopDomain Stage 2 in the main text) for the respective Stage 2 DNNs, the residue is assigned as a probable boundary residue with a Stage 3 filtering score of one; otherwise, it is given a Stage 3 filtering score of zero. Low values indicate a strong ability to separate putative boundaries (distance ≤ 20 residues from a true boundary) from non-boundary residues (distance > 20 residues from a true boundary). Each cut-off is calculated by maximizing the harmonic mean of the fraction of non-boundary residues above the cut-off and the fraction of putative boundary residues below it (see section TopDomain Stage 2 in the main text).

[b] The first section lists homology-based predictors, the second sequence-based predictors, and the third structure-based domain parsers.

Text T7: Stage 3 DNNs

The input image size of each TopDomain Stage 3 DNN (For TopDomain and TopDomain$_{Seq}$) is 81 × 19. This stems from the fact that Stage 3 uses a sliding window of ± 40 residues (40 × 2 + 1 = 81) and has input features consisting of six class probabilities from Stage 2 (see section TopDomain Stage 2 in the main text) for three window sizes (± 10, ± 20, and ± 40 residues) and a binary feature, called the Stage 3 filter (6 × 3 + 1 = 19) (see section TopDomain Stage 3 in the main text). The target value in Stage 3 is a number between 0 and 1 calculated by Equation 2 (see section TopDomain Stage 3 in the main text).

Each Stage 3 DNN (For TopDomain and TopDomain$_{Seq}$) is a regression ResNet with 50 layers. Further information about each layer is available in Table S2. The learning rate schedule, early stopping, and oversampling are the same as in Stage 2. The Adam optimizer with the same hyper-parameters as in Stage 2 is selected for Stage 3. As this is a regression problem, no activation function is used for the last layer. To be sensitive to outliers, the mean squared error (MSE) is used as a loss function [65].

Table S2. Stage 2 and Stage 3 DNN architectures [a]

Layer Name	Stage 2 DNN [b]	Stage 3 DNN [b]
Layer 1	Convolution, 3×3, 64, batch normalization, 3×3 max pooling	Convolution, 3×3, 64, batch normalization, 3×3 max pooling
Layer 2	Convolution, $\begin{bmatrix}3\times3, & 64\\ 3\times3, & 64\end{bmatrix}\times2$, batch normalization, ReLU activation	Convolution, $\begin{bmatrix}1\times1, 64\\ 3\times3, 64\\ 1\times1, 256\end{bmatrix}\times3$, batch normalization, ReLU activation
Layer 3	Convolution, $\begin{bmatrix}3\times3, & 128\\ 3\times3, & 128\end{bmatrix}\times2$, batch normalization, ReLU activation	Convolution, $\begin{bmatrix}1\times1, 128\\ 3\times3, 128\\ 1\times1, 512\end{bmatrix}\times4$, batch normalization, ReLU activation
Layer 4	Convolution, $\begin{bmatrix}3\times3, & 256\\ 3\times3, & 256\end{bmatrix}\times2$, batch normalization, ReLU activation	Convolution, $\begin{bmatrix}1\times1, 256\\ 3\times3, 256\\ 1\times1, 1024\end{bmatrix}\times6$, batch normalization, ReLU activation
Layer 5	Convolution, $\begin{bmatrix}3\times3, & 512\\ 3\times3, & 512\end{bmatrix}\times2$, batch normalization, ReLU activation	Convolution, $\begin{bmatrix}1\times1, 512\\ 3\times3, 512\\ 1\times1, 2048\end{bmatrix}\times3$, batch normalization, ReLU activation
Layer 6	Average pooling, 6-dimension fully connected, softmaxactivation	Average pooling, 1-dimension fully connected

[a] Every layer of Stage 2 and Stage 3 ResNet is composed of sequential operations such as convolutions, batch normalizations, max pooling , average pooling, and activation functions.

[b] Convolution operations are represented with its kernel size and feature map size, as implemented in [64]. There is a batch normalization and activation function after each convolution operation as implemented in [64].

Text T8: Peak Detection & Confidence Estimation

To detect the peaks of the smoothened boundary score, the find-peak function of the Scipy package is used to assign binary boundary predictions. The peak height and prominence (the distance between a peak and the closest other peak) are chosen as peak detection parameters. To estimate the optimal peak detection parameters for each DBP, we optimize the F1 score according to the strict quality criterion (see section Quality Criteria in the main text). First, we choose a list of height and prominence values between 0 and 1 with a step size of 0.1 and perform a grid search on the training set for each DBP. Then we use a step size of 0.01 between the two values with the highest F1 score and repeat the grid search. The final height and prominence parameters of each DBP are shown in Table S3.

Table S3. Optimal height and parameters for peak detection of each DBP [a]

Predictor[b]	Height	Prominence
TopDomain	0.500	0.580
ThreaDom	0.480	0.570
InterProScan	0.540	0.350
DOMPRED	0.330	0.120
FIEFDom	0.490	0.100
$\text{TopDomain}_{\text{Seq}}$	0.560	0.580
ConDo	0.500	0.550
DOBO	0.500	0.280
DeepDom	0.390	0.120
DROP	0.760	0.200
PPRODO	0.100	0.050
DOMCUT	0.060	0.003
Scooby-Domain	0.150	0.360
DOMpro	0.360	0.040
$\text{TopDomain}_{\text{Parse}}$	0.690	0.550
DDOMAIN	0.530	0.290
DomainParser2	0.530	0.400
SWORD	0.540	0.400

[a] The height and prominence parameters were obtained by optimizing the boundary detection F1 score according to the strict correctness criterion (see section Quality Criteria in the main text). A small prominence value indicates that the boundary score does not change much in the training data set.

[b] The first section lists homology-based DBPs. The second section lists sequence-based DBPs. The third section

lists structure-based DBPs.

The boundary region is predicted as a function of peak height. This prediction is based on a logistic function fitted on the 1σ (68%) confidence interval of the distance between predicted and true boundary. To do so, the peak height of all predicted boundaries in the TopDomain training dataset is collected and binned with a bin size of 0.001. For each bin, the 68% confidence interval for the distance between the peak and the nearest true boundary is calculated. Finally, a logistic function is fit to these values. The fitted logistic functions are then used as models to predict the boundary confidence interval for future predictions. The results of the logistic fit for each DBP are presented in Figure S2

We expect that when a DBP detects a boundary with a high peak height, the confidence interval of the distance between the predicted and the true boundary is smaller since we are more confident in the location of the boundary. TopDomain, TopDomain$_{Seq}$, and TopDomain$_{Parse}$ meet these expectations (Figure S2), and their high Pearson's coefficients of determination (Pearson's R^2) indicate that there are only a few noisy data points. Some predictors such as DOBO and DROP have flat curves indicating that their confidence is independent of peak height. Finally, for DOMCUT and PPRODO, the confidence interval of the distance between the predicted and the true boundary becomes the larger, the higher the peak is, but because they have small height and prominence values (Table S3) their boundary score is mostly flat. Therefore, they consider most peaks as a predicted boundary with correspondingly large confidence intervals due to the high false-positive rate of these predictors.

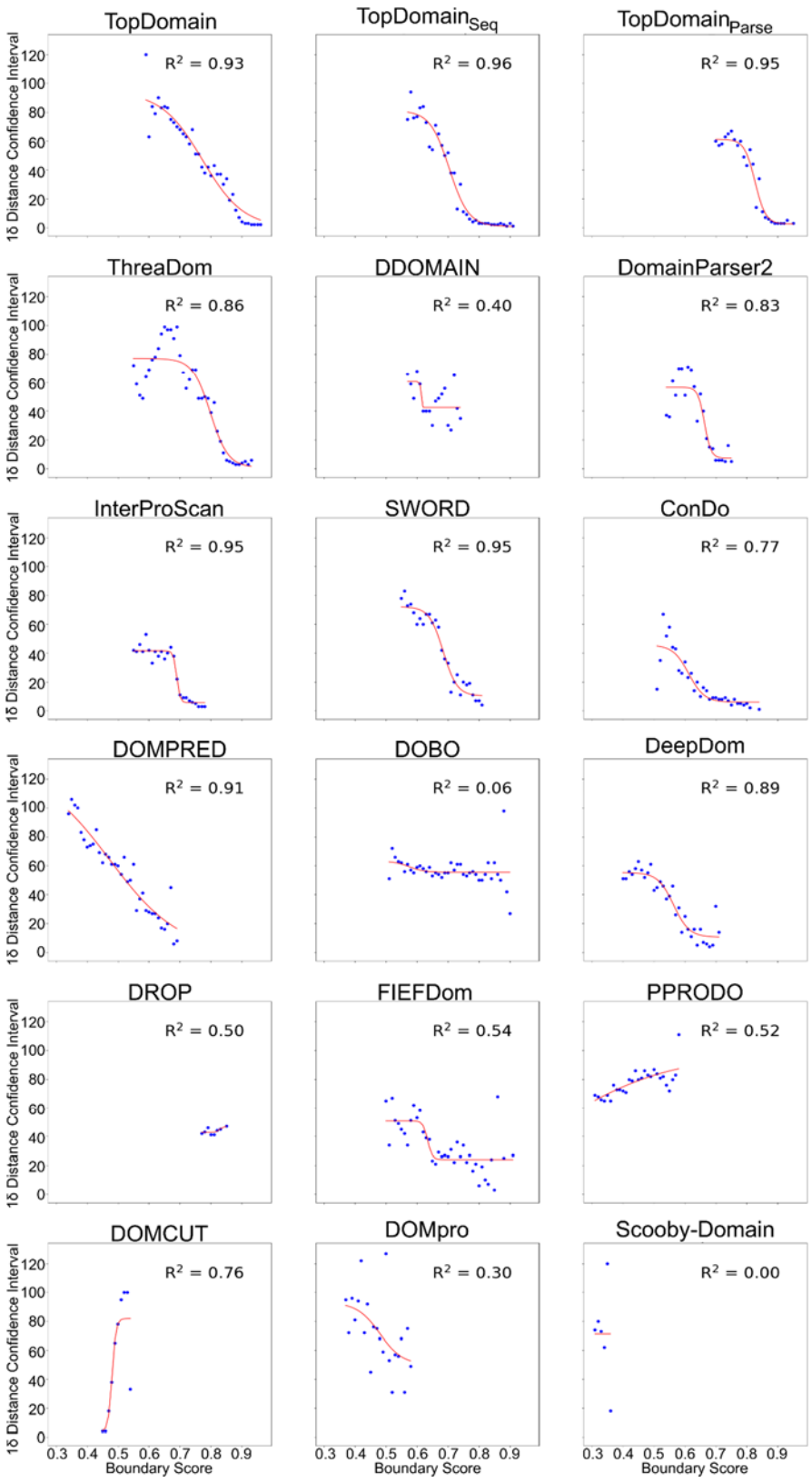

Figure S2. Confidence Interval Estimations. This figure shows the 1σ confidence intervals of the distance between predicted and true boundary vs. the boundary score of each DBP. The blue points were obtained by calculating 68% confidence interval for the distance between the peak and the nearest true boundary . The red lines are logistic functions fit to the data points to avoid getting negative or infinite values as a boundary region. The fitted functions are used as models to predict the boundary confidence interval for future predictions. Higher Pearson's R^2 values represent less noisy data points for a given DBP.

Table S4. Literature correctness criterion performance of boundary predictors

Predictor[a]	TopDomain Test dataset[b]				CASP domain dataset[b]			
	Precision	**Recall**	**F1**	**MCC**	**Precision**	**Recall**	**F1**	**MCC**
TopDomain	**75.1 %**	**81.9 %**	**78.4 %**	64 %	**48.3 %**	**46.7 %**	**47.5 %**	**40 %**
ThreaDom	71.2 %	69.0 %	70.1 %	**67 %**	43.8 %	25.7 %	32.4 %	36 %
InterProScan	67.1 %	52.9 %	59.2 %	49 %	40.3 %	17.8 %	24.7 %	22 %
DOMPRED	52.8 %	39.8 %	45.4 %	39 %	39.0 %	21.1 %	27.4 %	37 %
FIEFDom	50.3 %	19.3 %	27.9 %	26 %	33.3 %	4.6 %	8.0 %	4 %
$TopDomain_{Seq}$	71.8 %	63.7 %	**67.5 %**	59 %	43.6 %	33.6 %	37.9 %	33 %
ConDo	**79.3 %**	*39.1 %*	52.4 %	**60 %**	**60.7 %**	22.4 %	32.7 %	43 %
DOBO	39.8 %	*41.4 %*	40.6 %	35 %	33.1 %	29.6 %	31.2 %	32 %
DeepDom	49.4 %	*31.3 %*	38.3 %	37 %	51.5 %	32.9 %	**40.2 %**	**51 %**
DROP	42.4 %	*22.6 %*	29.5 %	28 %	48.6 %	23.7 %	31.9 %	6 %
PPRODO	22.5 %	72.2 %	34.3 %	0 %	22.7 %	69.2 %	34.2 %	0 %
DOMCUT	20.4 %	**86.5 %**	33.0 %	0 %	24.0 %	**91.8 %**	38.1 %	0 %
Scooby-Domain	28.3 %	*15.4 %*	*20 %*	36 %	19.5 %	10.5 %	13.7 %	31.4 %
DOMpro	40.7 %	*12.8 %*	*19.7 %*	28 %	21.7 %	6.6 %	10.1 %	18.8 %
$TopDomain_{Parse}$	69.0 %	**55.8 %**	**61.7 %**	54 %	64.4 %	**30.9 %**	**41.8 %**	7 %
DDOMAIN	**72.4 %**	49.1 %	58.5 %	64 %	**64.8 %**	23.0 %	34.0 %	**31 %**
DomainParser2	68.5 %	50.1 %	57.9 %	**68 %**	59.4 %	25.0 %	35.2 %	31 %
SWORD	56.9 %	50.0%	53.2%	33%	40.6 %	27.0 %	32.4 %	17 %
RanDom	19.8 %	44.4 %	27.4 %	-	-	-	-	-

[a] The first section lists homology-based DBPs. The second section lists sequence-based DBPs. The third section lists structure-based DBPs. The fourth section shows the performance of RanDom as a base-line reference.

[b] TopDomain performance is compared to primary predictors using the literature correctness criterion (boundary distance ≤ 20 residues). The boundary performance metrics are Precision, Recall, and F1 score calculated on multi-domain proteins of the TopDomain Test dataset (no. of proteins: 1857, no. of boundaries: 3354) and the CASP domain dataset (no. of proteins: 82, no. of boundaries: 304), respectively. The MCC column reflects the Matthews Correlation Coefficient for classifying single-domain proteins on each of the respective datasets. Overall, TopDomain methods show a better Precision, Recall, and F1 score than primary DBPs in each category, as well as an equivalent ability to predict single-domain proteins. An exception is DDOMAIN and DomainParser2, which show markedly higher MCCs for the classification of single-domain proteins than $TopDomain_{Parse}$ for both the TopDomain Test dataset and the CASP domain dataset. For each category, the best performance is highlighted in bold. Performances worse than RanDom are highlighted in italics.

Table S5. Strict correctness criterion performance of boundary predictors

Predictor[a]	TopDomain Test dataset [b]				CASP domain dataset [b]			
	Precision	**Recall**	**F1**	**MCC**	**Precision**	**Recall**	**F1**	**MCC**
TopDomain	**70.8 %**	**77.2 %**	**73.8 %**	64 %	**43.5 %**	**42.1 %**	**42.8 %**	**40 %**
ThreaDom	65.3 %	63.2 %	64.2 %	**67 %**	40.4 %	23.7 %	29.9 %	36 %
InterProScan	56.2 %	44.3 %	49.5 %	49 %	31.3 %	13.8 %	19.2 %	22 %
DOMPRED	37.8 %	28.6 %	32.6 %	39 %	24.4 %	13.2 %	17.1 %	37 %
FIEFDom	37.4 %	*14.6 %*	21.0 %	26 %	14.3 %	2.0 %	3.5 %	4 %
$TopDomain_{Seq}$	65.5 %	**58.1 %**	**61.6 %**	59 %	41.9 %	32.2 %	**36.4 %**	33 %
ConDo	**67.3 %**	33.2 %	44.5 %	**60 %**	**51.8 %**	19.1 %	27.9 %	43 %
DOBO	30.6 %	31.8 %	31.2 %	35 %	24.3 %	21.7 %	22.9 %	32 %
DeepDom	32.8 %	*20.8 %*	25.5 %	37 %	35.1 %	22.4 %	27.3 %	**51 %**
DROP	32.7 %	*17.4 %*	22.8 %	28 %	32.4 %	15.8 %	21.2 %	6 %
PPRODO	13.1 %	43.5 %	20.1 %	0 %	14.5 %	45.4 %	22.0 %	0 %
DOMCUT	11.0 %	53.4 %	18.3 %	0 %	14.1 %	**62.9 %**	23.1%	0 %
Scooby-Domain	14.7 %	*8.1 %*	*10.4 %*	36 %	12.2 %	6.6 %	8.5 %	31.4 %
DOMpro	20.8 %	*6.5 %*	*9.9 %*	28 %	10.9 %	3.3 %	5.1 %	18.8 %
$TopDomain_{Parse}$	63.2 %	**51.1 %**	**56.5 %**	54 %	56.2 %	**27.0 %**	**36.4 %**	7 %
DDOMAIN	**65.5 %**	44.4 %	52.9 %	64 %	**59.3 %**	21.1 %	31.1 %	**31 %**
DomainParser2	60.4 %	44.2 %	51.1 %	**68 %**	56.2 %	23.7 %	33.3 %	31 %
SWORD	49.3 %	43.3 %	46.1 %	33%	35.6%	24.3 %	29.2 %	17 %
RanDom	10.2 %	22.8 %	14.1 %	-	-	-	-	-

[a] The first section lists homology-based DBPs. The second section lists sequence-based DBPs. The third section lists structure-based DBPs. The fourth section shows the performance of RanDom as a base-line reference.

[b] TopDomain performance is compared to primary predictors using the strict correctness criterion (boundary distance ≤ 10 residues). The boundary performance metrics are Precision, Recall, and F1 score calculated on multi-domain proteins of the TopDomain Test dataset (no. of proteins: 1857, no. of boundaries: 3354) and the CASP domain dataset (no. of proteins: 82, no. of boundaries: 304), respectively. The MCC column reflects the Matthews Correlation Coefficient for classifying single-domain proteins on each of the respective datasets. Overall, TopDomain methods show a better Precision, Recall, and F1 score than primary DBPs in each category, as well as an equivalent ability to predict single-domain proteins. An exception is DDOMAIN and DomainParser2, which show markedly higher MCCs for the classification of single-domain proteins than $TopDomain_{Parse}$ for both the TopDomain Test dataset and the CASP domain dataset. For each category, the best performance is highlighted in bold. Performances worse than RanDom are highlighted in italics.

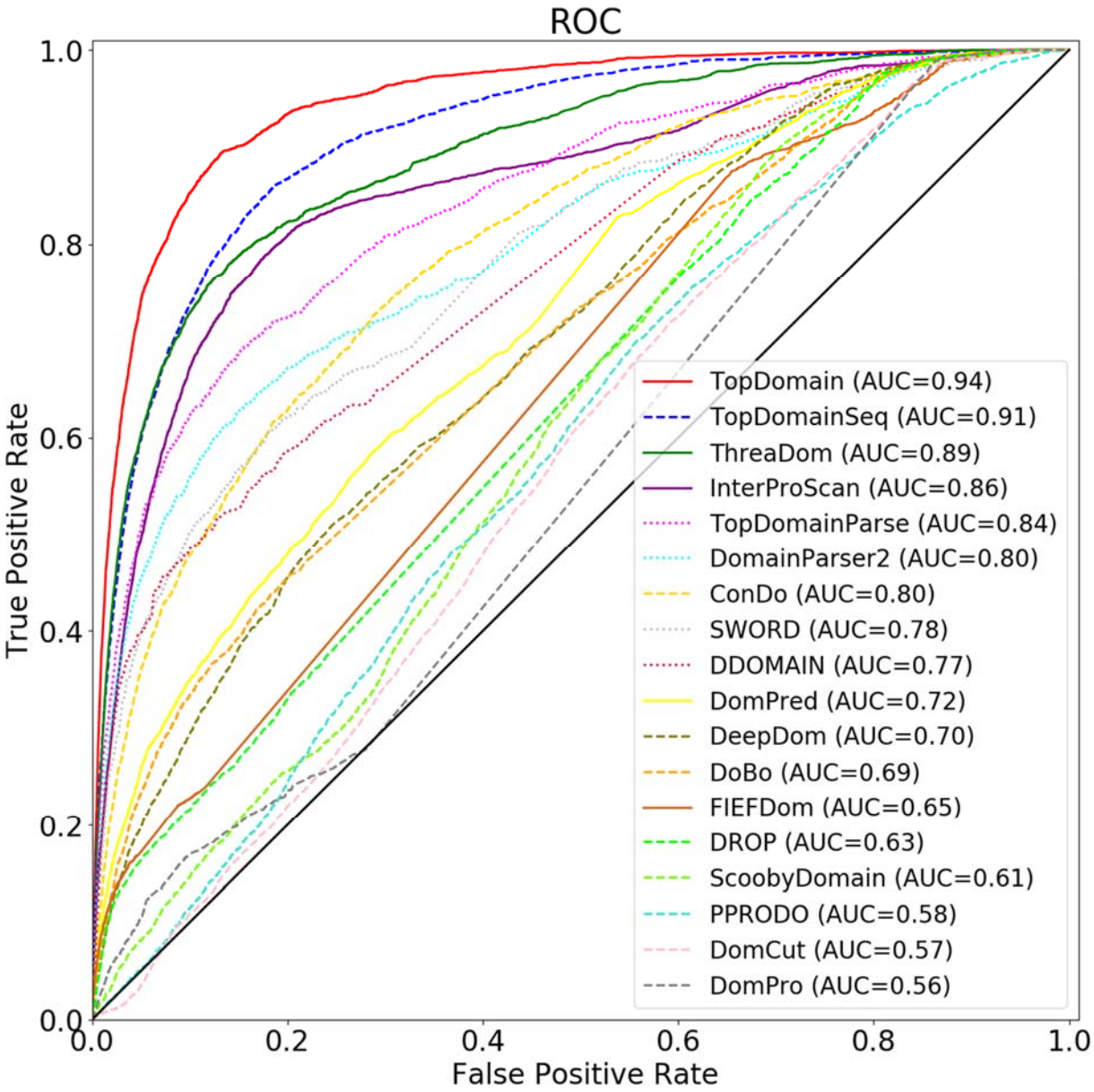

Figure S3. ROC of TopDomain and Primary DBP scores. This figure shows the receiver-operator characteristic curves for all primary DBPs and TopDomain methods and the area under the curve (AUC) for each predictor. These scores reflect the ability of each DBP score to separate non-boundary residues from boundary residues. They do not, however, reflect how boundaries are assigned, since boundary assignment depends not only on the score of an individual residue, but on the height and prominence of the entire boundary peak. Homology-based predictors are shown in solid lines, sequence-based predictors are shown in dashed lines, and structure-based domain parsers are shown in dotted lines. The black diagonal line reflects a random boundary score, which has equal probability of assigning a residue as boundary and non-boundary. Performance is calculated for the TopDomain test set (1857 multi-domain targets and 627 single-domain targets, 3354 boundaries).

Supplemental References

1. Fang, C.; Shang, Y.; Xu, D., Mufold-Ss: New Deep Inception-inside-Inception Networks for Protein Secondary Structure Prediction. *Proteins: Struct. Funct. Bioinform.* **2018**, *86*, 592-598.
2. Heffernan, R.; Yang, Y.; Paliwal, K.; Zhou, Y., Capturing Non-Local Interactions by Long Short-Term Memory Bidirectional Recurrent Neural Networks for Improving Prediction of Protein Secondary Structure, Backbone Angles, Contact Numbers and Solvent Accessibility. *Bioinformatics* **2017**, *33*, 2842-2849.
3. Joo, K.; Lee, S. J.; Lee, J., Sann: Solvent Accessibility Prediction of Proteins by Nearest Neighbor Method. *Proteins: Struct. Funct. Bioinform.* **2012**, *80*, 1791-1797.
4. Michel, M.; Menéndez Hurtado, D.; Elofsson, A., Pconsc4: Fast, Accurate and Hassle-Free Contact Predictions. *Bioinformatics* **2018**.
5. Durbin, S.; Feher, G., Protein Crystallization. *Annu. Rev. Phys. Chem.* **1996**, *47*, 171-204.
6. Dale, G. E.; Oefner, C.; D'Arcy, A., The Protein as a Variable in Protein Crystallization. *J. Struct. Biol.* **2003**, *142*, 88-97.
7. Derewenda, Z. S.; Vekilov, P. G., Entropy and Surface Engineering in Protein Crystallization. *Acta Crystallogr. Sect. D. Biol. Crystallogr.* **2006**, *62*, 116-124.
8. Derewenda, Z. S., The Use of Recombinant Methods and Molecular Engineering in Protein Crystallization. *Methods* **2004**, *34*, 354-363.
9. Pearl, F. M.; Bennett, C.; Bray, J. E.; Harrison, A. P.; Martin, N.; Shepherd, A.; Sillitoe, I.; Thornton, J.; Orengo, C. A., The Cath Database: An Extended Protein Family Resource for Structural and Functional Genomics. *Nucleic Acids Res.* **2003**, *31*, 452-455.
10. Fox, N. K.; Brenner, S. E.; Chandonia, J.-M., Scope: Structural Classification of Proteins—Extended, Integrating Scop and Astral Data and Classification of New Structures. *Nucleic Acids Res.* **2013**, *42*, D304-D309.
11. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E., The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235-242.
12. Yeats, C.; Lees, J.; Reid, A.; Kellam, P.; Martin, N.; Liu, X.; Orengo, C., Gene3d: Comprehensive Structural and Functional Annotation of Genomes. *Nucleic Acids Res.* **2007**, *36*, D414-D418.
13. Zhou, H.; Xue, B.; Zhou, Y., Ddomain: Dividing Structures into Domains Using a Normalized Domain–Domain Interaction Profile. *Protein Science* **2007**, *16*, 947-955.
14. Xu, Y.; Xu, D.; Gabow, H. N., Protein Domain Decomposition Using a Graph-Theoretic Approach. *Bioinformatics* **2000**, *16*, 1091-1104.
15. Xue, Z.; Jang, R.; Govindarajoo, B.; Huang, Y.; Wang, Y., Extending Protein Domain Boundary Predictors to Detect Discontinuous Domains. *PLoS One* **2015**, *10*, e0141541.
16. Steinegger, M.; Söding, J., Clustering Huge Protein Sequence Sets in Linear Time. *Nature communications* **2018**, *9*, 1-8.
17. Aroul-Selvam, R.; Hubbard, T.; Sasidharan, R., Domain Insertions in Protein Structures. *J. Mol. Biol.* **2004**, *338*, 633-641.
18. Katoh, K.; Standley, D. M., Mafft Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution* **2013**, *30*, 772-780.
19. Chun, E.; Thompson, A. A.; Liu, W.; Roth, C. B.; Griffith, M. T.; Katritch, V.; Kunken, J.; Xu, F.; Cherezov, V.; Hanson, M. A., Fusion Partner Toolchest for the Stabilization and Crystallization of G Protein-Coupled Receptors. *Structure* **2012**, *20*, 967-976.
20. Ekman, D.; Björklund, Å. K.; Frey-Skött, J.; Elofsson, A., Multi-Domain Proteins in the Three Kingdoms of Life: Orphan Domains and Other Unassigned Regions. *J. Mol. Biol.* **2005**, *348*, 231-243.

21. Apic, G.; Gough, J.; Teichmann, S. A., Domain Combinations in Archaeal, Eubacterial and Eukaryotic Proteomes. *J. Mol. Biol.* **2001**, *310*, 311-325.
22. Marchler-Bauer, A.; Lu, S.; Anderson, J. B.; Chitsaz, F.; Derbyshire, M. K.; DeWeese-Scott, C.; Fong, J. H.; Geer, L. Y.; Geer, R. C.; Gonzales, N. R., Cdd: A Conserved Domain Database for the Functional Annotation of Proteins. *Nucleic Acids Res.* **2010**, *39*, D225-D229.
23. Marchler-Bauer, A.; Derbyshire, M. K.; Gonzales, N. R.; Lu, S.; Chitsaz, F.; Geer, L. Y.; Geer, R. C.; He, J.; Gwadz, M.; Hurwitz, D. I., Cdd: Ncbi's Conserved Domain Database. *Nucleic Acids Res.* **2014**, *43*, D222-D226.
24. Boratyn, G. M.; Schaffer, A.; Agarwala, R.; Altschul, S. F.; Lipman, D. J.; Madden, T. L., Domain Enhanced Lookup Time Accelerated Blast. *Biol. Direct* **2012**, *7*, 12.
25. Lobley, A.; Sadowski, M. I.; Jones, D. T., Pgenthreader and Pdomthreader: New Methods for Improved Protein Fold Recognition and Superfamily Discrimination. *Bioinformatics* **2009**, *25*, 1761-1767.
26. Jaroszewski, L.; Rychlewski, L.; Li, Z.; Li, W.; Godzik, A., Ffas03: A Server for Profile–Profile Sequence Alignments. *Nucleic Acids Res.* **2005**, *33*, W284-W288.
27. Peng, J.; Xu, J., Raptorx: Exploiting Structure Information for Protein Alignment by Statistical Inference. *Proteins: Struct. Funct. Bioinform.* **2011**, *79*, 161-171.
28. Yang, Y.; Faraggi, E.; Zhao, H.; Zhou, Y., Improving Protein Fold Recognition and Template-Based Modeling by Employing Probabilistic-Based Matching between Predicted One-Dimensional Structural Properties of Query and Corresponding Native Properties of Templates. *Bioinformatics* **2011**, *27*, 2076-2082.
29. Söding, J.; Biegert, A.; Lupas, A. N., The Hhpred Interactive Server for Protein Homology Detection and Structure Prediction. *Nucleic Acids Res.* **2005**, *33*, W244-W248.
30. Postic, G.; Ghouzam, Y.; Chebrek, R.; Gelly, J.-C., An Ambiguity Principle for Assigning Protein Structural Domains. *Science Advances* **2017**, *3*, e1600552.
31. Kabsch, W.; Sander, C., Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. *Biopolymers* **1983**, *22*, 2577-2637.
32. Li, W.; Godzik, A., Cd-Hit: A Fast Program for Clustering and Comparing Large Sets of Protein or Nucleotide Sequences. *Bioinformatics* **2006**, *22*, 1658-1659.
33. Xue, Z.; Xu, D.; Wang, Y.; Zhang, Y., Threadom: Extracting Protein Domain Boundary Information from Multiple Threading Alignments. *Bioinformatics* **2013**, *29*, i247-i256.
34. Jones, P.; Binns, D.; Chang, H.-Y.; Fraser, M.; Li, W.; McAnulla, C.; McWilliam, H.; Maslen, J.; Mitchell, A.; Nuka, G., Interproscan 5: Genome-Scale Protein Function Classification. *Bioinformatics* **2014**, *30*, 1236-1240.
35. Bryson, K.; Cozzetto, D.; Jones, D. T., Computer-Assisted Protein Domain Boundary Prediction Using the Dom-Pred Server. *Current Protein and Peptide Science* **2007**, *8*, 181-188.
36. Bondugula, R.; Lee, M. S.; Wallqvist, A., Fiefdom: A Transparent Domain Boundary Recognition System Using a Fuzzy Mean Operator. *Nucleic Acids Res.* **2008**, *37*, 452-462.
37. Wang, S.; Peng, J.; Ma, J.; Xu, J., Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. *Sci. Rep.* **2016**, *6*, 18962.
38. Ward, J. J.; McGuffin, L. J.; Bryson, K.; Buxton, B. F.; Jones, D. T., The Disopred Server for the Prediction of Protein Disorder. *Bioinformatics* **2004**, *20*, 2138-2139.
39. Wang, S.; Weng, S.; Ma, J.; Tang, Q., Deepcnf-D: Predicting Protein Order/Disorder Regions by Weighted Deep Convolutional Neural Fields. *Int. J. Mol. Sci.* **2015**, *16*, 17315-17330.
40. Linding, R.; Russell, R. B.; Neduva, V.; Gibson, T. J., Globplot: Exploring Protein Sequences for Globularity and Disorder. *Nucleic Acids Res.* **2003**, *31*, 3701-3708.
41. Necci, M.; Piovesan, D.; Dosztányi, Z.; Tosatto, S. C., Mobidb-Lite: Fast and Highly Specific Consensus Prediction of Intrinsic Disorder in Proteins. *Bioinformatics* **2017**, *33*, 1402-1404.

42. Käll, L.; Krogh, A.; Sonnhammer, E. L., A Combined Transmembrane Topology and Signal Peptide Prediction Method. *J. Mol. Biol.* **2004**, *338*, 1027-1036.
43. Krogh, A.; Larsson, B.; Von Heijne, G.; Sonnhammer, E. L., Predicting Transmembrane Protein Topology with a Hidden Markov Model: Application to Complete Genomes. *J. Mol. Biol.* **2001**, *305*, 567-580.
44. Hayat, S.; Elofsson, A., Boctopus: Improved Topology Prediction of Transmembrane B Barrel Proteins. *Bioinformatics* **2012**, *28*, 516-522.
45. Petersen, T. N.; Brunak, S.; Von Heijne, G.; Nielsen, H., Signalp 4.0: Discriminating Signal Peptides from Transmembrane Regions. *Nature Methods* **2011**, *8*, 785.
46. Szklarczyk, R.; Heringa, J., Tracking Repeats Using Significance and Transitivity. *Bioinformatics* **2004**, *20*, i311-i317.
47. Jorda, J.; Kajava, A. V., T-Reks: Identification of Tandem Repeats in Sequences with a K-Means Based Algorithm. *Bioinformatics* **2009**, *25*, 2632-2638.
48. Söding, J.; Remmert, M.; Biegert, A., Hhrep: De Novo Protein Repeat Detection and the Origin of Tim Barrels. *Nucleic Acids Res.* **2006**, *34*, W137-W142.
49. Lupas, A., Predicting Coiled-Coil Regions in Proteins. *Current Opinion in Structural Biology* **1997**, *7*, 388-393.
50. Ludwiczak, J.; Winski, A.; Szczepaniak, K.; Alva, V.; Dunin-Horkawicz, S., Deepcoil—a Fast and Accurate Prediction of Coiled-Coil Domains in Protein Sequences. *Bioinformatics* **2019**.
51. Jones, D. T.; Singh, T.; Kosciolek, T.; Tetchner, S., Metapsicov: Combining Coevolution Methods for Accurate Prediction of Contacts and Long Range Hydrogen Bonding in Proteins. *Bioinformatics* **2014**, *31*, 999-1006.
52. Remmert, M.; Biegert, A.; Hauser, A.; Söding, J., Hhblits: Lightning-Fast Iterative Protein Sequence Searching by Hmm-Hmm Alignment. *Nature Methods* **2012**, *9*, 173-175.
53. Mirdita, M.; von den Driesch, L.; Galiez, C.; Martin, M. J.; Söding, J.; Steinegger, M., Uniclust Databases of Clustered and Deeply Annotated Protein Sequences and Alignments. *Nucleic Acids Res.* **2017**, *45*, D170-D176.
54. Adhikari, B.; Hou, J.; Cheng, J., Dncon2: Improved Protein Contact Prediction Using Two-Level Deep Convolutional Neural Networks. *Bioinformatics* **2017**, *34*, 1466-1472.
55. Eddy, S. R., Accelerated Profile Hmm Searches. *PLoS Computational Biology* **2011**, *7*, e1002195.
56. Consortium, U., Uniprot: A Hub for Protein Information. *Nucleic Acids Res.* **2014**, *43*, D204-D212.
57. Hong, S. H.; Joo, K.; Lee, J., Condo: Protein Domain Boundary Prediction Using Coevolutionary Information. *Bioinformatics* **2018**.
58. Henikoff, S.; Henikoff, J. G., Position-Based Sequence Weights. *J. Mol. Biol.* **1994**, *243*, 574-578.
59. Jones, D. T.; Buchan, D. W.; Cozzetto, D.; Pontil, M., Psicov: Precise Structural Contact Prediction Using Sparse Inverse Covariance Estimation on Large Multiple Sequence Alignments. *Bioinformatics* **2011**, *28*, 184-190.
60. Seemayer, S.; Gruber, M.; Söding, J., Ccmpred—Fast and Precise Prediction of Protein Residue–Residue Contacts from Correlated Mutations. *Bioinformatics* **2014**, *30*, 3128-3130.
61. Kaján, L.; Hopf, T. A.; Kalaš, M.; Marks, D. S.; Rost, B., Freecontact: Fast and Free Software for Protein Contact Prediction from Residue Co-Evolution. *BMC Bioinformatics* **2014**, *15*, 85.
62. Jones, D. T.; Kandathil, S. M., High Precision in Protein Contact Prediction Using Fully Convolutional Neural Networks and Minimal Sequence Features. *Bioinformatics* **2018**, *1*, 8.
63. Marks, D. S.; Colwell, L. J.; Sheridan, R.; Hopf, T. A.; Pagnani, A.; Zecchina, R.; Sander, C., Protein 3d Structure Computed from Evolutionary Sequence Variation. *PLoS One* **2011**, *6*, e28766.

64. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; 2016; pp 770-778.
65. Bishop, C. M., *Pattern Recognition and Machine Learning*. springer: 2006.
66. Qin, Z.; Kim, D.; Gedeon, T., Rethinking Softmax with Cross-Entropy: Neural Network Classifier as Mutual Information Estimator. *arXiv preprint arXiv:1911.10688* **2019**.
67. Kingma, D. P.; Ba, J., Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* **2014**.