# GREATER THAN THE SUM OF ITS PARTS: BUILDING SUBSTRUCTURE INTO PROTEIN ENCODING MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Protein representation learning has achieved major advances using large sequence and structure datasets, yet current models primarily operate at the level of individual residues or entire proteins. This overlooks a critical aspect of protein biology: proteins are composed of recurrent, evolutionarily conserved substructures that mediate core molecular functions. Despite decades of curated biological knowledge, these substructures remain largely unexploited in modern protein models. We introduce Magneton, an integrated environment for developing substructure-aware protein models. Magneton provides (1) a large-scale dataset of 530,601 proteins annotated with over 1.7 million substructures spanning 13,075 types, (2) a training framework for incorporating substructures into existing models, and (3) a benchmark suite of 13 tasks probing residue-, substructure-, and protein-level representations. Using Magneton, we develop substructure-tuning, a supervised finetuning method that distills substructural knowledge into pretrained protein models. Across state-of-the-art sequence- and structure-based models, substructure-tuning improves function-related tasks while revealing that substructural signals are complementary to global structural information. The Magneton environment, datasets, and substructure-tuned models are all openly available[1].

## 1 INTRODUCTION

Protein representation learning has progressed from models trained on large sequence databases (Rives et al., 2021; Elnaggar et al., 2022) to models incorporating experimentally determined or predicted structures (Gligorijević et al., 2021; Zhang et al., 2022b), enabling advances in folding (Lin et al., 2023), function prediction (Rao et al., 2019), and variant effect prediction (Meier et al., 2021; Brandes et al., 2023). However, these models have largely ignored the recurrent and modular composition of proteins, which introduces substantial technical challenges. Protein substructures occur at multiple spatial and functional scales, from local motifs spanning only a handful of residues to domains that cover large fractions of a protein. They are often non-contiguous in sequence space, making them difficult to encode with standard sequential architectures. A single residue can belong to several overlapping substructures, inducing hierarchical and context-dependent relationships that are not naturally handled by flat representations. Finally, annotated substructures are distributed in a long-tailed fashion, with abundant secondary structure elements but scarce examples of specialized motifs, complicating the design of training objectives and evaluation protocols.

These challenges arise because proteins are not uniform chains but are organized into recurrent, modular substructures that provide a natural multiscale vocabulary for representation. At the finest level are amino acids, which assemble into secondary structure elements such as alpha helices and beta sheets; these in turn combine into higher-order motifs and domains such as beta barrels and zinc fingers (Figure 1A). These substructures are responsible for core molecular functions of proteins, such as coordinating metal ions for reaction catalysis or binding to other proteins as parts of cellular signaling networks, and their importance is underscored by their occurrence in proteins sampled from across the tree of life. Decades of biological research has led to the categorization of these recurrent substructures, resulting in large databases that exhaustively annotate these elements

[1] https://anonymous.4open.science/r/magneton-14F2/README.md

across proteins (Sonnhammer et al., 1997; Paysan-Lafosse et al., 2025; Blum et al., 2025). However, prevailing protein representation learning methods still rely on self-supervised objectives that operate at the scale of single amino acids, such as masked language modeling or structural denoising, or occasionally operate on full proteins (Yu et al., 2023). This is despite abundant evidence that evolutionarily conserved substructures are key components of protein function (Rossman & Liljas, 1974). In this work, we ask, *how should we systematically incorporate decades of biological knowledge about protein substructures into protein encoding models?*

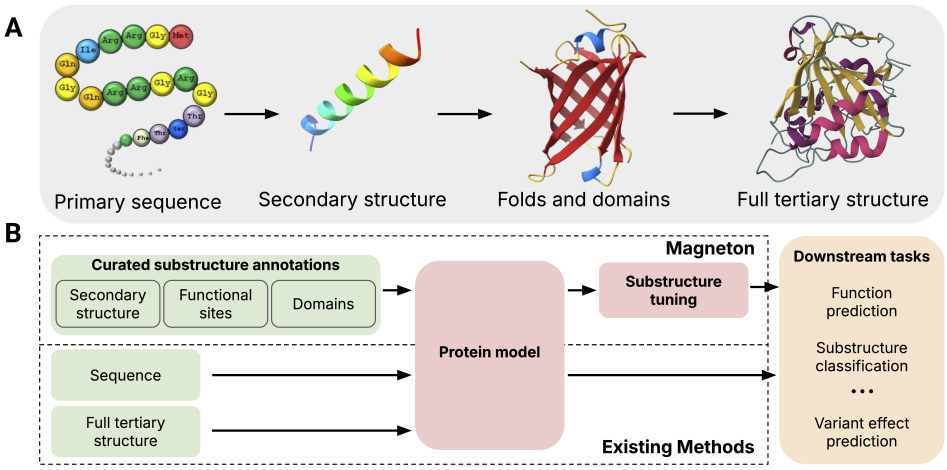



**Figure 1: Overview of protein structure and the Magneton environment.** (A) Proteins are built from modular substructures that assemble into full structures. (B) Magneton leverages decades of substructure research to provide an environment for developing and evaluating substructure-aware models.

While there exists a growing body of work exploring how to best integrate protein sequence and structure into a single model, either via direct incorporation of structural tokens (Su et al., 2023; Li et al., 2024; Hayes et al., 2025; Lu et al., 2025; Yuan et al., 2025) or finetuning of sequence models to better align with structural representations (Zhang et al., 2024b; Ouyang-Zhang et al., 2025), there are few examples of incorporating substructure information into protein encoding models. Models such as GearNet (Zhang et al., 2022b) use a multi-view contrastive objective and cite recurrent substructures as motivation, but use multiple views of subsets of the same protein rather than considering recurrent substrucutures across proteins. The Functional Community Invariance approach (Wang et al., 2025b) employs secondary structure annotations to guide graph augmentations but ignores higher-order substructures. Other threads of work seek to construct hierarchical representations of proteins, either by connecting residues to their exposed surface areas (Somnath et al., 2022; Zhang et al., 2024c; Mallet et al., 2025), or in models such as ProNet (Wang et al., 2023), by progressing from all-atom graphs to residue graphs, but these works pass over protein substructure as a valuable part of the structural hierarchy.

**Present work.** To close this gap, we first create a new environment for developing substructure-aware protein models, which we call *Magneton*. Magneton has three main components: (1) a dataset of proteins with curated substructures in an ML-ready format; (2) a framework for using these substructures to train or finetune protein encoding models; and (3) a benchmark of evaluation tasks that probe the learned representations at the residue, substructure, and protein levels (Figure 1B). By curating data from Pfam, InterPro, and DSSP, we create a dataset of 530,601 proteins with over 1.7 million substructural annotations (37 million when including secondary structure) across six substructure classes with 13,075 distinct substructure types.

Using Magneton, we next explore *substructure-tuning*, a supervised fine-tuning strategy that explicitly distills substructural information into protein encoders. Concretely, we formulate substructure-tuning as classification of evolutionarily conserved substructures, where residue-level embeddings produced by a base encoder are pooled to construct substructure representations and optimized with a cross-entropy loss. This objective is model-agnostic, requiring only residue-level embeddings, and naturally extends to multiple structural scales through a multi-task formulation in which each substructure class is assigned its own prediction head and the total loss is the sum across scales.

We systematically vary the substructures used for tuning, exploring configurations ranging from small, highly local elements (e.g., active sites spanning $< 10$ residues) to larger domains, as well as joint training over multiple scales. Substructure-tuning is evaluated on 13 benchmarking tasks using 6 state-of-the-art base models, including both sequence-only and sequence–structure encoders. Substructure-tuned representations yield consistent improvements of 5% on function-related prediction tasks (e.g., EC and GO terms), while effects on localization and residue-level tasks are neutral or negative. Improvements persist even when base models already incorporate global structural inputs, underscoring that substructural signals are distinct and complementary to global protein structure.

Our key contributions are: ① We present Magneton, a benchmark that combines large-scale curated substructural annotations with an associated Python library and a suite of 13 evaluation tasks spanning residue, substructure, protein, and interaction levels. This environment enables exploration of how substructural priors can be integrated into protein models. ② We introduce substructure-tuning, a supervised fine-tuning method for distilling substructural information into pretrained models. We exhaustively evaluate its design space across six state-of-the-art encoders, covering both sequence-only and sequence–structure models and ranging from 150M to 650M parameters. ③ We show that substructure-tuning improves models' ability to represent protein function: for example, Enzyme Commission (EC) prediction with ESM-C 300M improves from 0.688 to 0.815, and Gene Ontology molecular function prediction increases from 0.429 to 0.525. These results demonstrate that substructural information is complementary to global structure, yielding consistent gains in functional prediction tasks across architectures. We envision that this work will catalyze closer integration of machine learning and biology, motivating new approaches and inductive biases that incorporate decades of knowledge about protein structure across scales.

## 2 Related Work

**Integrating structure- and function-based inductive biases into sequence-based protein models.** A large body of work has explored distilling auxiliary modalities into sequence-based protein models. Some methods incorporate free-text descriptions, such as Gene Ontology terms (Zhang et al., 2022a) or SwissProt annotations (Xu et al., 2023). The majority, however, focus on structural information. Explicit approaches integrate structure directly, either through structure graphs (**?**) or structural tokenization (Su et al., 2023; Li et al., 2024). Structural distillation methods instead use structure only at training time, preserving sequence-only inference. For example, Implicit Structure Model (ISM) (Ouyang-Zhang et al., 2025) trains residue-level predictors on tokens from a structural autoencoder, while ESM-S (Zhang et al., 2024b) distills global structural information via fold classification. S-PLM (Wang et al., 2025a) employs contrastive learning to align representations of an ESM encoder with those of a contact-map encoder. Magneton differs by focusing on protein *substructures* rather than only residue-level or global structural signals. It provides large-scale curated annotations of conserved substructures and a framework for supervised fine-tuning on these elements to encode modular, recurrent units of protein organization. This is orthogonal to existing sequence-structure integration and structural distillation approaches.

**Substructure-aware training and hierarchical models.** Protein substructure admits a hierarchical view, but most hierarchical modeling approaches focus on geometric relations rather than functional substructures. Some methods connect residues to exposed surface areas (Somnath et al., 2022; Zhang et al., 2024c; Mallet et al., 2025), while others connect residues to constituent atoms (Wang et al., 2023). Few approaches incorporate substructural information directly. GearNet (Zhang et al., 2022b) uses a multiview contrastive objective that samples local regions within a protein, but supervision is restricted to intra-protein partitions rather than conserved substructures across proteins. SES-Adapter (Tan et al., 2024) augments sequence models with cross-attention to DSSP-derived secondary structure tokens, but does not extend beyond this single level of annotation. Protein language models such as xTrimoPGLM (Chen et al., 2025) use span-masking, but the masked spans are random residue segments rather than biologically defined substructures. ESM3 (Hayes et al., 2025) introduces multi-track tokenization, including secondary structure and function tracks, where the function track is derived from ontology terms often correlated with substructural annotations. However, the learning remains self-supervised and intra-protein, without supervision on conserved substructures across proteins. Magneton differs by providing annotations of conserved substructures across proteins and by defining supervised training objectives that operate directly on these annotations. This design moves beyond local partitions, random spans, or ontology proxies, enabling

systematic study of substructure-aware modeling across residue-, motif-, domain-, and protein-level representations.

**Geometric protein models.** Geometric deep learning has been widely applied to proteins, with models developed for folding (Jumper et al., 2021; Abramson et al., 2024), structure design (Passaro et al., 2025; Watson et al., 2023; Huang et al., 2024), and representation learning (Jing et al., 2020; Fang et al., 2025). These approaches operate at the atom scale (Qu et al., 2025; Widatalla et al., 2025) and encode spatial coordinates of all atoms to model global protein geometry. Magneton addresses a complementary problem: representing recurrent substructures that span residues, motifs, and domains, and recur across proteins. Rather than optimizing directly on atomic coordinates, Magneton introduces supervised objectives on conserved substructures, providing functional supervision across structural scales. This supervision captures signals relevant to tasks such as Enzyme Commission classification, Gene Ontology function prediction, and thermostability, where global geometry alone is insufficient. Substructural objectives can also be integrated with atom-scale geometric encoders to yield models that capture fine-grained geometry and functional modularity.

# 3 METHODS

**Preliminaries.** Two possible views of a protein $P$ are the residue-level, $P = (a_1, \ldots, a_l)$ where $a_i$ is the $i$'th residue in the primary sequence, and the substructure-level, $P = (s_1, \ldots, s_n)$ where each $s_i$ represents a substructure contained within a protein. Other views are possible (*e.g.* atom-level), but these two views are the most relevant for our work. In the substructure view, each substructure is a subset of $k$ residues, $s_i = \{a_j\}_{j=1}^{j=k}$, where the residues $a_j$ may or may not be contiguous in the primary sequence. Since substructures exist at multiple scales, a given residue may be a member of multiple, possibly overlapping substructures, *e.g.* a residue may be part of a secondary structure element, such as a beta strand, that is itself part of a larger fold, such as a beta barrel. It is also possible for a given residue to not be included in any annotated substructure. While the substructure view of a protein is common in the biological community, there is a lack of curated datasets for exploring it in the context of protein modeling.

## 3.1 MAGNETON DEVELOPMENT ENVIRONMENT

Magneton is an environment for developing substructure-aware protein models, and consists of three main parts: (1) a curated dataset of proteins with annotated substructures, (2) a framework for using this dataset for substructure-aware training, and (3) an integrated benchmark of evaluation tasks that probe a model's learned representations at multiple structural scales.

**Dataset.** We use the `2024_06` release of UniProtKB/TrEMBL (The UniProt Consortium, 2025) as our core protein dataset, containing roughly 254M proteins. We obtain annotations of 8-class secondary structure from DSSP (Kabsch & Sander, 1983; Hekkelman et al., 2025) and annotations of higher-order structures (Homologous superfamilies, domains, conserved sites, active sites, binding sites) from the 103.0 release of InterPro (Blum et al., 2025). We process these raw releases into Magneton's core datatypes representing a protein and its associated substructures, and store these as compressed, binary files which we shard to enable parallel processing and file-level shuffling for large-scale training runs. Due to the scale of the dataset at this stage and the size of protein structure data, we focus our further exploration on the manually curated SwissProt subset of UniProtKB, but make the processed version of the full UniProtKB/TrEMBL dataset available to the community. For each protein, we obtain amino acid sequences from UniProtKB and predicted structures from AlphaFold DB (Varadi et al., 2022). To ensure consistent training and evaluation across sequence-based and structure-based models, we subset the SwissProt dataset to only proteins with calculated structures in the current (Nov 2022) release of AlphaFold DB, leaving 530,601 proteins. Additional details on the dataset and processing can be found in Appendix A.1.1.

To focus learning efforts on substructures where sufficient data is present, we create a restricted label set of more frequently occurring substructures. We restrict to substructures that occur at least 75 times in the SwissProt dataset, corresponding to to retaining only the top 10% most frequently occurring domains. While this may seem stringent, we find that this retains the vast majority of actual substructure occurrences across types, since many substructures have very few occurrences. We additionally generate versions of our dataset using more permissive cutoffs (minimum counts of 25 or 10) (Appendix A.1.3). Moreover, our published datasets retain all substructure annotations,

| Substructure class | Unique types (pre-filter) | Total occurrences (pre-filter) | Unique types (post-filter) | Total occurrences (post-filter) | Median protein span |
|---|---|---|---|---|---|
| **Homologous superfamily** | 2978 | 1.09M | 1133 | 1.05M | 50% (137 AA) |
| **Domain** | 9133 | 389K | 917 | 301K | 34.8% (127 AA) |
| **Conserved site** | 739 | 175K | 356 | 162K | 5.18% (16 AA) |
| **Binding site** | 67 | 20.1K | 48 | 19.0K | 4.28% (16 AA) |
| **Active site** | 132 | 31.1K | 82 | 29.2K | 3.47% (12 AA) |
| **Secondary structure** | 8 | 35.2M | 8 | 35.2M | 0.94% (3.4 AA) |
| **Total** w/o secondary structure | 13075 | 1.71M | 2542 | 1.56M | — |

**Table 1: Summary of Magneton substructure dataset (SwissProt subset).** Before and after refers to filtering out rare substructures. Median protein span is the median length of a type of substructure, expressed as a percentage of the protein and as absolute amino acid count.

to enable future research by the community. Table 1 summarizes the different classes of substructures, their counts, number of types, and typical span on the protein. As expected for substructural elements, the majority of the substructures span less than 10% of the annotated protein, with the scale varying by the class of substructure. We then split this dataset into training, validation, and test sets using the AFDB50 sequence-based clusters (Barrio-Hernandez et al., 2023), ensuring that sequences sharing more than 50% identity and 90% overlap are assigned to the same split.

**Evaluation benchmark.** To provide a holistic evaluation of substructure-focused protein modeling within Magneton, we integrate numerous evaluation tasks from the community. These tasks probe a model's learned representations at multiple scales: individual residues, substructures, proteins, and protein interactions (Table 2). At the residue-level, we include contact prediction (Rao et al., 2019), zero-shot prediction of variant effects (Notin et al., 2023), and multiple types of functional residue prediction tasks (Dallago et al., 2021; Yuan et al., 2025); at the substructure-level, we include multiclass substructure classification problems derived from the Magneton dataset itself; at the protein-level, we include function prediction (GO and EC terms) (Gligorijević et al., 2021), subcellular localization (Almagro Armenteros et al., 2017), and fitness prediction (Rao et al., 2019). Finally, we include a human PPI prediction task (Pan et al., 2010; Xu et al., 2022). Full details of evaluation datasets can be found in Appendix A.1.5.

| Scale | Task | Task type | Metric | Data source |
|---|---|---|---|---|
| Interaction | Human PPI prediction | Binary | Accuracy | Pan et al. |
| Protein | Gene Ontology prediction | Multilabel | $F_{\max}$ | Gligorijević et al. |
| | Enzyme Commission prediction | Multilabel | $F_{\max}$ | |
| | Subcellular localization | Multiclass | Accuracy | Almagro Armenteros et al. |
| | Binary localization | Binary | Accuracy | |
| | Thermostability prediction | Regression | Spearman's $\rho$ | Rao et al. |
| Substructure | Substructure classification | Multiclass | Macro accuracy | Ours |
| Residue | Contact prediction | Binary | Precision@L | Rao et al. |
| | Variant effect prediction | Regression | Spearman's $\rho$ | Notin et al. |
| | Binding residue categorization | Multilabel | $F_{\max}$ | Dallago et al. |
| | Functional site prediction | Binary | AUROC | Yuan et al. |

**Table 2: Evaluation tasks contained within Magneton.** Grouped by the scale of structural representation they interrogate.

### 3.2 SUBSTRUCTURE REPRESENTATION AND TUNING

Given the dataset in Magneton, we now have a large collection of proteins $\mathcal{P}$, where each protein has curated substructural annotations, $P = (s_1, \ldots, s_k); P \in \mathcal{P}$. We first use this dataset to assess whether existing protein models can generate meaningful representations of substructures. Specifically, for a protein model $f$, we construct a representation of each substructure $s_j \in P$ by calculating residue-level embeddings, $f(P) = (v_1, \ldots, v_l), v_l \in \mathbb{R}^d$ where $v_i$ is the embedding of residue $a_i$. We then perform a substructure pooling operation over the constituent residues of $s$, $f(s) = \texttt{pool}(\{v_i : a_i \in s\}, f(s) \in \mathbb{R}^d$, where $\texttt{pool}$ can be any arbitrary pooling operation. These substructure-level representations are then input to a classifier over the possible substructure labels for the final substructure classification task. Since a substructure's constituent residues are given to the model, this is a *diagnostic task* meant to probe each model's ability to represent substructures, not a task meant to measure the ability to identify previously unannotated substructures. For the purposes of this diagnostic assessment, we freeze the parameters of the underlying protein model and train only the substructure classification head.

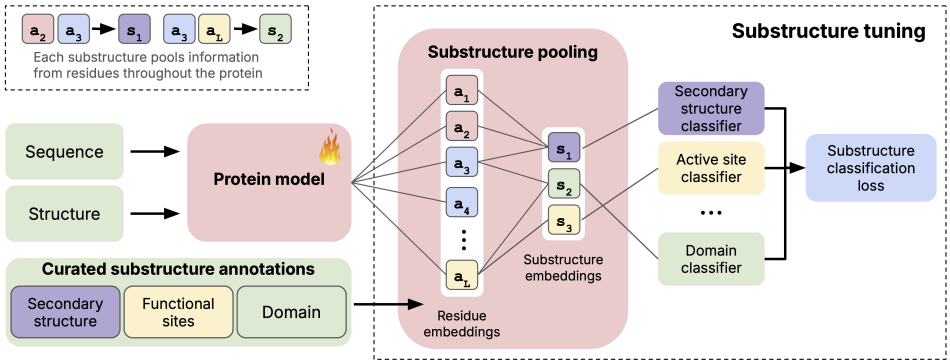



**Figure 2: Overview of using Magneton for substructure-tuning.** Given a pre-trained protein model, substructure-tuning first pools residue-level embedding to create substructure representations, which are then used for supervised finetuning via substructure type-specific classifier heads.

We next explore imbuing existing protein models with substructural information. In a process we refer to as *substructure-tuning*, we again perform the substructure classification task outlined above, but with finetuning of the original protein model's parameters (Figure 2) to encourage the model to distinguish between the many different types of biologically-relevant substructures in our dataset. Although we use supervised finetuning, other losses, such as a contrastive objective (van den Oord et al., 2019), could also be used. The substructure-tuning process is compatible with any finetuning method, including parameter-efficient methods such as LoRA (Hu et al., 2021) for larger base models. We perform substructure-tuning using the Magneton training set and explore tuning with different substructure types as well as their combinations. When finetuning with multiple substructure classes, each class uses its own predictor module with the cross entropy loss across all types summed to form the final substructure classification loss.

### 3.3 IMPLEMENTATION DETAILS

For our experiments, we select base protein models that represent state-of-the-art models across a range of model sizes and modality inputs. For sequence-based models, we use ESM2-150M and -650M (Lin et al., 2023) and ESM-C 300M and 600M (ESM Team, 2024). For models that incorporate protein structure, we use SaProt (Su et al., 2023) and ProSST-2048 (Li et al., 2024), both of which use both protein sequence and structure. We opt to exclude purely structural models such as GearNet (Zhang et al., 2022b) as their performance is generally below that of the sequence-structure models we've included.

For substructure classification and tuning, we use single-hidden layer MLPs where the hidden dimension size matches that of the base model as our prediction modules, mean pool for the substructure pooling operation. For substructure-tuning, we perform full finetuning of the base model. To regularize the substructure-tuning process and avoid catastrophic forgetting of the base model's

original objective, we use elastic weight consolidation (EWC) (Kirkpatrick et al., 2017). Detailed training methodology is available in Appendix A.2.1.

For supervised downstream evaluations, we train head models on top of either the original base model or the substructure-tuned base model. For these evaluations, we freeze the base model to focus on evaluating the representations learned during substructure-tuning. Results across all tasks and models were generated within the Magneton environment and use identical datasets and splits. We unfortunately exclude ProSST from the functional site prediction and contact prediction tasks due to it's incompatibility with experimental structures from PDB. Full training details for all models and tasks are available in Appendix A.2.2.

# 4 EXPERIMENTS

## 4.1 SUBSTRUCTURE REPRESENTATION ASSESSMENT

Table 3 shows that base models are readily able to produce effective representations of substructures across scales, with structure-based models generally outperforming sequence-only models. We also find that models are able to correctly classify substructures within proteins that contain multiple substructures (*e.g.* accurately classifying all domains within a single protein containing multiple domains), indicating that classification relies on local structural cues rather than global structural similarity (Figure 3A). While performance degrades for some rarer substructures, we generally see high accuracy even for rare substructures (Figure 3B).

| Model | Homologous superfamily | Domain | Conserved site | Binding site | Active site | Secondary structure |
|---|---|---|---|---|---|---|
| ESM2-150M | 0.899 | 0.969 | 0.988 | 1.000 | 0.995 | 0.827 |
| +ST | 0.925 | 0.983 | 0.991 | 0.999 | 0.994 | 0.916 |
| ESM2-650M | 0.926 | 0.982 | 0.986 | 1.000 | 0.995 | 0.892 |
| +ST | 0.902 | 0.967 | 0.986 | 1.000 | 0.996 | 0.938 |
| ESM-C 300M | 0.913 | 0.962 | 0.990 | 0.998 | 0.994 | 0.863 |
| +ST | 0.946 | 0.982 | 0.983 | 0.999 | 0.996 | 0.757 |
| ESM-C 600M | 0.919 | 0.975 | 0.992 | 0.977 | 0.994 | 0.891 |
| +ST | 0.907 | 0.966 | 0.993 | 0.997 | 0.996 | 0.927 |
| ProSST-2048 | 0.888 | 0.945 | 0.995 | 0.996 | 0.993 | 0.927 |
| +ST | 0.879 | 0.976 | 0.991 | 0.991 | 0.995 | 0.961 |
| SaProt | 0.916 | 0.967 | 0.992 | 0.999 | 0.996 | 0.955 |
| +ST | 0.925 | 0.980 | 0.993 | 0.999 | 0.996 | 0.972 |

**Table 3: Comparison of substructure classification performance.** Model performance on the *diagnostic task* of classifying substructures given their annotated residues. All values are macro-averaged accuracy.

## 4.2 SUBSTRUCTURE-TUNING

**Substructure-tuning configurations.** Table 4 shows the results of substructure-tuning with a range of different substructure classes, both individually and their combinations, as measured by downstream evaluation tasks. Due to the large number of possible configurations, we restricted this initial exploration to a single model (ESM-C 300M), a subset of evaluation tasks, and a selection of the $2^6$ possible substructure class combinations aimed at exploring combinations of substructure classes across scales.

Our exploration of substructure configurations revealed the following: 1) The effects of substructure-tuning are largely consistent across the selected substructure types used, with large performance boosts in tasks related to protein function (GO:MF, GO:BP, EC, Thermostability) and neutral to negative effects on localization tasks (GO:CC, Binary localization, Subcellular localization) and residue-level variant-effect prediction. 2) These effects are present even when tuning with very small substructures, such as active sites, which typically consist of only 12 amino acids (median protein span of 3.47%). Based on these results, we selected the combination of active site, binding

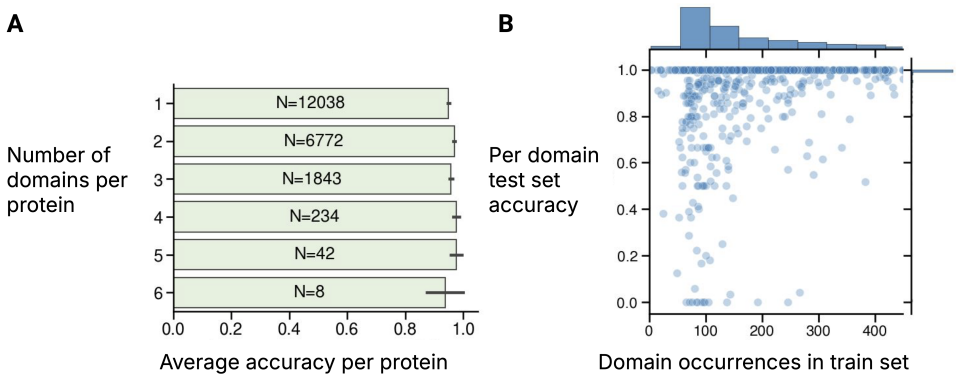



**Figure 3: (A) Domain classification uses local cues.** Even within proteins containing multiple domains, classification accuracy remains high for all contained domains. Labels within bars show the number of test set proteins containing that number of domains. **(B) Domain classification accuracy as a function of training set representation.** Results shown for ESM-C 300M.

| Substructures used | EC | GO:BP | GO:CC | GO:MF | Localization (Accuracy) | | Thermostability (Spearman's $\rho$) | Zero-shot DMS ( Spearman's $\rho$) |
|---|---|---|---|---|---|---|---|---|
| | $F_{max}$ | | | | Binary | Subcellular | | |
| None | 0.688 | 0.307 | **0.416** | 0.429 | 0.871 | 0.703 | 0.648 | **0.432** |
| H_____ | 0.805 | 0.312 | 0.395 | 0.518 | 0.851 | 0.632 | 0.662 | 0.308 |
| _D____ | 0.776 | 0.307 | 0.403 | 0.501 | 0.811 | 0.640 | **0.666** | 0.340 |
| __C___ | 0.749 | 0.318 | 0.398 | 0.491 | 0.870 | **0.706** | 0.661 | 0.402 |
| ___B__ | 0.745 | 0.315 | 0.415 | 0.478 | 0.852 | 0.686 | 0.663 | 0.423 |
| ____A_ | 0.794 | 0.318 | 0.403 | 0.518 | 0.851 | 0.639 | 0.663 | 0.340 |
| _____S | 0.618 | 0.297 | 0.379 | 0.381 | 0.823 | 0.587 | 0.612 | 0.264 |
| HD____ | 0.774 | 0.316 | 0.388 | 0.500 | 0.847 | 0.606 | 0.639 | 0.302 |
| H____S | 0.765 | 0.297 | 0.395 | 0.466 | **0.883** | 0.651 | 0.644 | 0.346 |
| HD___S | 0.754 | 0.318 | 0.413 | 0.473 | 0.868 | 0.633 | 0.658 | 0.350 |
| H_CBA_ | 0.800 | 0.322 | 0.389 | 0.515 | 0.857 | 0.611 | 0.663 | 0.340 |
| _D___S | 0.751 | 0.308 | 0.384 | 0.462 | 0.872 | 0.646 | 0.643 | 0.369 |
| _DCBA_ | **0.815** | **0.329** | 0.395 | **0.525** | 0.851 | 0.662 | 0.659 | 0.369 |
| __CBA_ | 0.761 | 0.325 | 0.403 | 0.488 | 0.879 | 0.681 | 0.660 | 0.410 |
| ___BA_ | 0.740 | 0.319 | 0.406 | 0.467 | 0.841 | 0.677 | 0.656 | 0.418 |
| __CBAS | 0.719 | 0.313 | 0.393 | 0.453 | 0.839 | 0.666 | 0.636 | 0.379 |
| HDCBAS | 0.760 | 0.315 | 0.383 | 0.457 | 0.832 | 0.624 | 0.640 | 0.359 |

**Table 4: Comparison of substructure-tuning configurations.** Performance across tasks for ESM-C 300M with a range of substructure-tuning configurations. For each configuration, the substructures used are indicated by the presence of that substructure type's single-letter code: H=Homologous superfamily, D=Domain, C=Conserved site, B=Binding site, A=Active site, S=Secondary structure; an underscore (_) means that substructure type was not used.

site, and conserved site as the substructure-tuning configuration for use on the full set of models and benchmarks, as this configuration represented a balance of positive gains on function-related tasks and neutral effects on localization and residue-level variant-effect tasks.

**Substructure-tuning across models.** Tables 5 and 6 show how the selected substructure-tuning configuration affects the downstream performance of the full set of base protein models across protein-level and residue-level tasks, respectively. The full evaluation across models and benchmarks led to the following conclusions: 1) Results across models are consistent with the initial exploration: performance boosts in function-related tasks and neutral to negative effects on localization and residue-level tasks. 2) Importantly, these results hold true for models that already incorporate protein structure as an input (ProSST-2048 and SaProt), suggesting complementarity between structural and substructural information. Due to the close relationship between substructures and protein function, we additionally verify that performance increases from substructure-tuning are not trivially attributed to leakage between the Magneton substructure training set and the test sets of the evaluation tasks (Appendix A.1.4). We also perform an ablation of EWC, finding that it moderates the performance improvements of substructure-tuning, while reducing the amount of degradation in tasks where substructure-tuning has negative effects (Appendix A.2.1).

| Model | EC | GO:BP | GO:CC | GO:MF | Localization (Accuracy) | | Thermostability (Spearman's $\rho$) | Human PPI (AUROC) |
|---|---|---|---|---|---|---|---|---|
| | $F_{max}$ | | | | Binary | Subcellular | | |
| ESM2-150M | 0.727 | 0.316 | 0.416 | 0.441 | 0.869 | 0.694 | 0.627 | 0.933 |
| +ST | 0.742 | 0.324 | 0.415 | 0.473 | 0.866 | 0.679 | 0.582 | 0.919 |
| ESM2-650M | 0.755 | 0.319 | 0.431 | 0.486 | 0.876 | 0.710 | 0.643 | 0.939 |
| +ST | 0.745 | 0.321 | 0.440 | 0.534 | 0.895 | 0.749 | 0.655 | 0.935 |
| ESM-C 300M | 0.688 | 0.307 | 0.416 | 0.429 | 0.871 | 0.703 | 0.648 | 0.917 |
| +ST | 0.761 | 0.325 | 0.403 | 0.488 | 0.879 | 0.681 | 0.660 | 0.933 |
| ESM-C 600M | 0.701 | 0.312 | 0.403 | 0.436 | 0.863 | 0.713 | 0.668 | 0.927 |
| +ST | 0.780 | 0.319 | 0.385 | 0.527 | 0.872 | 0.635 | 0.667 | 0.902 |
| SaProt (650M) | 0.778 | 0.326 | 0.453 | 0.538 | 0.887 | 0.784 | 0.692 | 0.952 |
| +ST | 0.839 | 0.339 | 0.446 | 0.584 | 0.896 | 0.741 | 0.697 | 0.932 |
| ProSST-2048 | 0.778 | 0.317 | 0.426 | 0.522 | 0.878 | 0.693 | 0.686 | 0.925 |
| +ST | 0.791 | 0.314 | 0.420 | 0.567 | 0.853 | 0.683 | 0.648 | 0.883 |

**Table 5: Protein-level task performance for models with and without substructure-tuning.**

| Model | Binding residue ($F_{max}$) | Functional site prediction | | Contact Prediction | | | Variant Effect (Spearman's $\rho$) |
|---|---|---|---|---|---|---|---|
| | | Binding (AUROC) | Catalytic (AUROC) | Short (Precision@L) | Medium (Precision@L) | Long (Precision@L) | |
| ESM2-150M | 0.379 | 0.871 | 0.910 | 0.487 | 0.452 | 0.289 | 0.342 |
| +ST | 0.327 | 0.852 | 0.890 | 0.460 | 0.445 | 0.285 | 0.262 |
| ESM2-650M | 0.366 | 0.849 | 0.912 | 0.551 | 0.528 | 0.372 | 0.359 |
| +ST | 0.362 | 0.851 | 0.927 | 0.532 | 0.518 | 0.367 | 0.317 |
| ESM-C 300M | 0.367 | 0.851 | 0.923 | 0.339 | 0.364 | 0.174 | 0.432 |
| +ST | 0.411 | 0.866 | 0.910 | 0.350 | 0.374 | 0.180 | 0.410 |
| ESM-C 600M | 0.357 | 0.850 | 0.921 | 0.329 | 0.362 | 0.161 | 0.434 |
| +ST | 0.368 | 0.852 | 0.906 | 0.313 | 0.315 | 0.141 | 0.381 |
| SaProt (650M) | 0.423 | 0.891 | 0.923 | 0.788 | 0.747 | 0.697 | 0.457 |
| +ST | 0.400 | 0.871 | 0.924 | 0.765 | 0.726 | 0.647 | 0.405 |
| ProSST-2048 | 0.375 | N/A | N/A | N/A | N/A | N/A | 0.507 |
| +ST | 0.342 | N/A | N/A | N/A | N/A | N/A | 0.356 |

**Table 6: Residue-level task performance for models with and without substructure-tuning.**

We additionally explored how substructure-tuning interacts with task-specific finetuning by repeating the evaluations above for a subset of models and tasks with full finetuning of the protein model for each task (Appendix A.2.3). We found that task-specific finetuning results in similar performance across models trained with and without substructure-tuning, indicating that aggressive task-specific finetuning may dominate the substructural information imbued during the substructure-tuning process.

**Mechanistic exploration of substructure-tuning.** We next investigated the effects of substructure-tuning on the learned embeddings of the underlying protein models (Table 7, Appendix Figure A.13). By comparing substructure embeddings before and after substructure-tuning, we found that substructure-tuning improves a model's ability to group substructures of the same type. Furthermore, by restricting our analysis to the rare substructure types that were excluded from the Magneton training set, we find that substructure-tuning results in more consistent representations of even substructure types that were never seen during training. This indicates that substructure-tuning encourages models to learn general features of functional substructures, rather than just signatures of specific substructure types. These experiments focused on ESM-C and SaProt as representative sequence-only and sequence-structure models.

To understand the task-specific effects of substructure-tuning, we performed a gradient conflict analysis, in which we compared the gradient updates for ESM-C 300M for the substructure classification task and for a set of evaluation tasks, including protein-level function prediction and residue-level classification tasks (Appendix A.3.2). We found that gradients for evaluation tasks were highly

| Model | Homologous superfamily | | Domain | | Conserved site | | Binding site | | Active site | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Seen | Unseen | Seen | Unseen | Seen | Unseen | Seen | Unseen | Seen | Unseen |
| ESM-C 300M | -0.183 | 0.180 | -0.184 | 0.201 | 0.279 | 0.466 | 0.378 | 0.641 | 0.490 | 0.476 |
| +ST | 0.339 | 0.584 | 0.486 | 0.652 | 0.830 | 0.747 | 0.882 | 0.894 | 0.933 | 0.816 |
| SaProt (650M) | 0.079 | 0.301 | 0.122 | 0.412 | 0.534 | 0.623 | 0.613 | 0.796 | 0.714 | 0.701 |
| +ST | 0.478 | 0.684 | 0.554 | 0.717 | 0.796 | 0.764 | 0.843 | 0.938 | 0.912 | 0.866 |

**Table 7: Silhouette scores for substructure types included ("seen") and excluded ("unseen") from training.** Higher silhouette scores indicate tighter clustering of substructures within a type.

consistent across batches, gradients for substructure classification had lower, although still positive, within-task similarity and were close to orthogonal to gradients for the evaluation tasks. While these results do not fully explain the task-specific effects of substructure-tuning, they suggest that the behavior is not due to a simple misalignment between the substructure objective and certain downstream tasks. Instead, we hypothesize that our current instantiation of substructure-tuning biases the model against fine-grained residue-level distinctions, because it explicitly encourages residues within the same substructure to share similar representations.

Finally, we performed an explainability analysis to understand if substructure-tuning increases a model's utilization of substructural information. For the subset of GO:MF terms that can be mapped to domain annotations, we found that substructure-tuning resulted in increased attribution of predictions to residues within domains by an average of 17% over the untuned base model (Appendix A.3.3).

## 5 Conclusion and Future Work

Our study has several limitations and directions for future work. We focused on an intuitive substructure-tuning approach applied to existing state-of-the-art models, which yielded mixed gains across tasks and proved brittle under task-specific finetuning. These results suggest that exploring alternate methods for incorporating substructural information may be fruitful. Modifications at the architectural level, such as hierarchical or graph-based encoders, or training objectives that operate across multiple scales simultaneously may provide a more stable integration strategy. Our current exploration of substructure-tuning focused which substructure types to use for tuning, but their representation varies greatly across the dataset (e.g. millions of secondary structures, tens of thousands of active sites. Exploring how to best balance or weight these different types is another avenue of future exploration. Finally, our experiments restricted to SwissProt proteins. Extending to the full UniProtKB and incorporating the long tail of infrequent substructures could enable deeper insights into poorly characterized aspects of protein modularity.

In this work, we've presented the open problem: *how to best incorporate decades of research on protein substructures into protein models?* To this end, we introduced Magneton, an integrated environment for developing substructure-aware protein models that provides (1) large-scale datasets of proteins with curated substructure annotations, (2) a framework for using these processed datasets for training and finetuning protein models using sequence, structure, and substructure inputs, and (3) a suite of benchmarking tasks that evaluate models across a range of structural granularities. Using Magneton, we explored both how well existing models are able to represent protein substructures and whether a supervised finetuning paradigm can be used to effectively imbue those models with substructural information. We found that while this direct, intuitive substructure-tuning approach improves model performance on molecular function-related tasks, it has a neutral to negative effect on others. Our work lays the foundation for development of substructure-aware protein models.

## 6 Ethics Statement

This work involves the analysis of publicly available protein sequence and structure data from established databases (UniProtKB/SwissProt, AlphaFold DB, InterPro, and Pfam). All data used in this study is derived from previously published sources and does not involve human subjects, animal experiments, or the generation of new biological data requiring ethical oversight. Our work improves computational methods for understanding protein function, which could contribute to ad-

vances in drug discovery and biotechnology. We encourage responsible use of our methods and datasets, which are publicly available to promote scientific reproducibility and advancement.

## 7 REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide the following:

1. All code for Magneton, including data processing pipelines, model training scripts, and evaluation benchmarks, is available at `https://anonymous.4open.science/r/magneton-14F2/`. The processed datasets will be made publicly available following the anonymous review period.
2. We provide comprehensive implementation details including model architectures and hyperparameters, training procedures, optimization details, and data splitting procedures (Methods 3.3).
3. We specify all experimental details including dataset statistics and preprocessing steps such as substructure filtering criteria and thresholds (Table 1, Appendix **??**), as well as evaluation metrics and protocols for all benchmark tasks (Table 2).
4. All experiments can be reproduced using 1-4 NVIDIA A100 GPUs.

The modular design of Magneton facilitates easy plug-and-play usability of our benchmark suite, supporting not only reproducibility but also future research in this area.

## REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w.

José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx431.

Inigo Barrio-Hernandez, Jingi Yeo, Jürgen Jänes, Milot Mirdita, Cameron L. M. Gilchrist, Tanita Wein, Mihaly Varadi, Sameer Velankar, Pedro Beltrao, and Martin Steinegger. Clustering predicted structures at the scale of the known protein universe. *Nature*, 622(7983):637–645, 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06510-w.

Matthias Blum, Antonina Andreeva, Laise Cavalcanti Florentino, Sara Rocio Chuguransky, Tiago Grego, Emma Hobbs, Beatriz Lazaro Pinto, Ailsa Orr, Typhaine Paysan-Lafosse, Irina Ponamareva, Gustavo A Salazar, Nicola Bordin, Peer Bork, Alan Bridge, Lucy Colwell, Julian Gough, Daniel H Haft, Ivica Letunic, Felipe Llinares-López, Aron Marchler-Bauer, Laetitia Meng-Papaxanthos, Huaiyu Mi, Darren A Natale, Christine A Orengo, Arun P Pandurangan, Damiano Piovesan, Catherine Rivoire, Christian J A Sigrist, Narmada Thanki, Françoise Thibaud-Nissen, Paul D Thomas, Silvio C E Tosatto, Cathy H Wu, and Alex Bateman. InterPro: the protein sequence classification resource in 2025. *Nucleic Acids Research*, 53(D1):D444–D456, 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae1082.

Nadav Brandes, Grant Goldman, Charlotte H. Wang, Chun Jimmie Ye, and Vasilis Ntranos. Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, 55(9):1512–1522, 2023. ISSN 1546-1718. doi: 10.1038/s41588-023-01465-0.

Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, Chiming Liu, Aohan Zeng, Yuxiao Dong, Jie Tang, and Le Song. xTrimoPGLM: unified 100-billion-parameter pretrained transformer for deciphering the language of proteins. *Nature Methods*, 22(5):1028–1039, 2025. ISSN 1548-7105. doi: 10.1038/s41592-025-02636-z.

Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce Wittmann, Nick Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. FLIP: Benchmark tasks in fitness landscape inference for proteins. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3095381.

ESM Team. ESM Cambrian: Revealing the mysteries of proteins with unsupervised learning. https://www.evolutionaryscale.ai/blog/esm-cambrian, 2024.

Ada Fang, Michael Desgagné, Zaixi Zhang, Andrew Zhou, Joseph Loscalzo, Bradley L. Pentelute, and Marinka Zitnik. Learning Universal Representations of Intermolecular Interactions with ATOMICA, 2025. ISSN 2692-8205.

Vladimir Gligorijević, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12(1):3168, 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-23303-9.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018.

Maarten L. Hekkelman, Daniel Álvarez Salmoral, Anastassis Perrakis, and Robbie P. Joosten. DSSP 4: FAIR annotation of protein secondary structure. *Protein Science*, 34(8):e70208, 2025. ISSN 1469-896X. doi: 10.1002/pro.70208.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, 2021.

Yufei Huang, Yunshu Liu, Lirong Wu, Haitao Lin, Cheng Tan, Odin Zhang, Zhangyang Gao, Siyuan Li, Zicheng Liu, Yunfan Liu, Tailin Wu, and Stan Z. Li. EVA: Geometric Inverse Design for Fast Protein Motif-Scaffolding with Coupled Flow. In *The Thirteenth International Conference on Learning Representations*, 2024.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from Protein Structure with Geometric Vector Perceptrons. In *International Conference on Learning Representations*, 2020.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983. ISSN 1097-0282. doi: 10.1002/bip.360221211.

Felix Kallenborn, Alejandro Chacon, Christian Hundt, Hassan Sirelkhatim, Kieran Didi, Sooyoung Cha, Christian Dallago, Milot Mirdita, Bertil Schmidt, and Martin Steinegger. GPU-accelerated homology search with MMseqs2. *Nature Methods*, 22(10):2024–2027, 2025. ISSN 1548-7105. doi: 10.1038/s41592-025-02819-8.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Mingchen Li, Yang Tan, Xinzhu Ma, Bozitao Zhong, Huiqun Yu, Ziyi Zhou, Wanli Ouyang, Bingxin Zhou, Pan Tan, and Liang Hong. ProSST: Protein Language Modeling with Quantized Structure and Disentangled Attention. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, 2019.

Jiarui Lu, Xiaoyin Chen, Stephen Zhewen Lu, Chence Shi, Hongyu Guo, Yoshua Bengio, and Jian Tang. Structure Language Models for Protein Conformation Generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Vincent Mallet, Yangyang Miao, Souhaib Attaiki, Bruno Correia, and Maks Ovsjanikov. AtomSurf: Surface Representation for Learning on Protein Structures. In *The Thirteenth International Conference on Learning Representations*, 2025.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, 2021.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. *Advances in Neural Information Processing Systems*, 36:64331–64379, 2023.

Jeffrey Ouyang-Zhang, Chengyue Gong, Yue Zhao, Philipp Kraehenbuehl, Adam Klivans, and Daniel Jesus Diaz. Distilling Structural Representations into Protein Sequence Models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xiao-Yong Pan, Ya-Nan Zhang, and Hong-Bin Shen. Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features. *Journal of Proteome Research*, 9(10):4992–5001, 2010. ISSN 1535-3907. doi: 10.1021/pr100618t.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards Accurate and Efficient Binding Affinity Prediction, 2025.

Typhaine Paysan-Lafosse, Antonina Andreeva, Matthias Blum, Sara Rocio Chuguransky, Tiago Grego, Beatriz Lazaro Pinto, Gustavo A Salazar, Maxwell L Bileschi, Felipe Llinares-López, Laetitia Meng-Papaxanthos, Lucy J Colwell, Nick V Grishin, R Dustin Schaeffer, Damiano Clementel, Silvio C E Tosatto, Erik Sonnhammer, Valerie Wood, and Alex Bateman. The Pfam

protein families database: embracing AI/ML. *Nucleic Acids Research*, 53(D1):D523–D534, 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae997.

Wei Qu, Jiawei Guan, Rui Ma, kezhai, Weikun.Wu, and Haobo Wang. P(all-atom) Is Unlocking New Path For Protein Design. In *Forty-second International Conference on Machine Learning*, 2025.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating Protein Transfer Learning with TAPE, 2019.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118.

Michael G. Rossman and Anders Liljas. Recognition of structural domains in globular proteins. *Journal of Molecular Biology*, 85(1):177–181, 1974. ISSN 0022-2836. doi: 10.1016/0022-2836(74)90136-3.

Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-Scale Representation Learning on Proteins, 2022.

Erik L.L. Sonnhammer, Sean R. Eddy, and Richard Durbin. Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins: Structure, Function, and Bioinformatics*, 28(3):405–420, 1997. ISSN 1097-0134. doi: 10.1002/(SICI)1097-0134(199707)28:3⟨405::AID-PROT10⟩3.0.CO;2-L.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein Language Modeling with Structure-aware Vocabulary, 2023.

Yang Tan, Mingchen Li, Bingxin Zhou, Bozitao Zhong, Lirong Zheng, Pan Tan, Ziyi Zhou, Huiqun Yu, Guisheng Fan, and Liang Hong. Simple, Efficient, and Scalable Structure-Aware Adapter Boosts Protein Language Models. *Journal of Chemical Information and Modeling*, 64(16):6338–6349, 2024. ISSN 1549-9596. doi: 10.1021/acs.jcim.4c00689.

The UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 2025. ISSN 1362-4962. doi: 10.1093/nar/gkae1010.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, 2019.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, Augustin Žídek, Tim Green, Kathryn Tunyasuvunakool, Stig Petersen, John Jumper, Ellen Clancy, Richard Green, Ankur Vora, Mira Lutfi, Michael Figurnov, Andrew Cowie, Nicole Hobbs, Pushmeet Kohli, Gerard Kleywegt, Ewan Birney, Demis Hassabis, and Sameer Velankar. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444, 2022. ISSN 0305-1048. doi: 10.1093/nar/gkab1061.

Duolin Wang, Mahdi Pourmirzaei, Usman L. Abbas, Shuai Zeng, Negin Manshour, Farzaneh Esmaili, Biplab Poudel, Yuexu Jiang, Qing Shao, Jin Chen, and Dong Xu. S-PLM: Structure-Aware Protein Language Model via Contrastive Learning Between Sequence and Structure. *Advanced Science*, 12(5):2404212, 2025a. ISSN 2198-3844. doi: 10.1002/advs.202404212.

Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. Learning Hierarchical Protein Representations via Complete 3D Graph Networks, 2023.

Yusong Wang, Shiyin Tan, Jialun Shen, Yicheng Xu, Haobo Song, Qi Xu, Prayag Tiwari, and Mingkun Xu. Enhancing Graph Contrastive Learning for Protein Graphs from Perspective of Invariance. In *Forty-second International Conference on Machine Learning*, 2025b.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8.

Talal Widatalla, Richard W. Shuai, Brian Hie, and Possu Huang. Sidechain conditioning and modeling for full-atom protein sequence design with FAMPNN. In *Forty-second International Conference on Machine Learning*, 2025.

Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Chang Ma, Runcheng Liu, and Jian Tang. PEER: A Comprehensive and Multi-Task Benchmark for Protein Sequence Understanding. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. ProtST: Multi-Modality Learning of Protein Sequences and Biomedical Texts. https://arxiv.org/abs/2301.12040v2, 2023.

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023. doi: 10.1126/science.adf2465.

Xinyu Yuan, Zichen Wang, Marcus D. Collins, and Huzefa Rangwala. Protein Structure Tokenization: Benchmarking and New Recipe. In *Forty-second International Conference on Machine Learning*, 2025.

Chengxin Zhang, Xi Zhang, Lydia Freddolino, and Yang Zhang. BioLiP2: an updated structure database for biologically relevant ligand-protein interactions. *Nucleic Acids Research*, 52(D1): D404–D412, 2024a. ISSN 1362-4962. doi: 10.1093/nar/gkad630.

Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Jiazhang Lian, Qiang Zhang, and Huajun Chen. OntoProtein: Protein Pretraining With Gene Ontology Embedding, 2022a.

Zuobai Zhang, Minghao Xu, Arian Rokkum Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein Representation Learning by Geometric Structure Pretraining. In *The Eleventh International Conference on Learning Representations*, 2022b.

Zuobai Zhang, Jiarui Lu, Vijil Chenthamarakshan, Aurélie Lozano, Payel Das, and Jian Tang. Structure-Informed Protein Language Model, 2024b.

Zuobai Zhang, Pascal Notin, Yining Huang, Aurelie Lozano, Vijil Chenthamarakshan, Debora Susan Marks, Payel Das, and Jian Tang. Multi-Scale Representation Learning for Protein Fitness Prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024c.

# A Appendix

## A.1 Datasets

### A.1.1 Magneton substructure dataset

Here we provide additional details on the curated protein substructure dataset that makes up a core part of the Magneton environment.

**Dataset processing.** Below we outline the steps to create the protein substructure dataset:

- We start from the full XML file of all InterPro annotations (`match_complete.xml.gz`) for the 103.0 release, downloaded from the InterPro FTP server [1].
- We parse each XML entry, which each correspond to a single protein, by extracting all `<match>` elements that contain at least one `<ipr>` element. The presence of the `<ipr>` element indicates that the annotation is integrated into InterPro and has been assigned a unique InterPro accession ID. No additional filtering is performed at this stage.
- At this point, the dataset covers all of the approximately 254 million proteins in the `2024_06` release of UniProtKB/TrEMBL. To subset to the SwissProt set, we obtain the set of all SwissProt proteins (UniProt IDs and amino acid sequences) from the `uniprot_sprot-only2024_06.tar.gz` file on the UniProt FTP server [2].
- To ensure consistency between models trained using only sequence data and models trained using sequence and structure data, we further subset the SwissProt set to only proteins contained within the v4 release of AlphaFold DB (`swissprot_cif_v4.tar`)[3].
- We use the CIF files from AlphaFold DB to source secondary structure annotations as calculated using DSSP.
- To select for a non-redundant set of substructural annotations, we select only annotations marked as "representative" within InterPro, when such annotations are available (as of the 103.0 release, these were only available for the "Repeat", "Family", and "Domain" types) [4].

The full set of scripts used for the above steps alongside a detailed `README` file are available in our GitHub repository in the `scripts/dataset_processing` directory.

**Substructure type descriptions.** Here we provide additional information about the various types of substructures contained within the Magneton dataset.

- **Homologous superfamily** - a group of proteins that share a common evolutionary origin, reflected by similarity in their structure, even if sequence similarity is low. Examples include alpha-helical portion of some viral capsid proteins (IPR008935) and a group of single-stranded DNA-binding transcriptional regulator proteins (IPR009044).
- **Domain** - distinct functional, structural, or sequence units that may exist in a variety of biological contexts. Examples include zinc finger binding domains which serve as binding sites for various types of ligands (one example type is IPR000058) and various types of phosphatase domains which enable regulation of protein phosphorylation (such as IPR000242).
- **Conserved site** - a short sequence that contains one or more conserved residues. Examples include the helix-turn-helix motif found in all known DNA binding proteins that regulate gene expression (IPR000047) and the helix-hairpin-helix motif found in proteins that exhibit non-specific DNA binding activity (IPR000445).
- **Binding site** - a short sequence that contains one or more conserved residues, which form a protein interaction site. Examples include sites for binding copper in various enzymes (IPR001505) and sites for binding proteins with other well-characterized motifs (for example, the IQ motif which binds the EF-hand domain, IPR000048).

[1] https://ftp.ebi.ac.uk/pub/databases/interpro/releases/103.0/

[2] https://ftp.uniprot.org/pub/databases/uniprot/previous_releases/release-2024_06/knowledgebase/

[3] https://ftp.ebi.ac.uk/pub/databases/alphafold/v4/

[4] https://interpro-documentation.readthedocs.io/en/latest/represent_dom.html

- **Active site** - a short sequence that contains one or more conserved residues, which allow the protein to bind a ligand. Examples include active sites for catalyzing hydrolysis of DNA and RNA (IPR002071) and active sites for catalyzing the breakdown of lipids (IPR008265).
- **Secondary structure** - conserved local spatial arrangements of a span of amino acids in a protein. Canonical examples are alpha helices and beta sheets. For our work, we use 8-class secondary structure definitions from DSSP [5].

**Example dataset entry.** For illustrative purposes, here we provide an abbreviated example of a single entry in the Magneton dataset in JSONL format:


```
{
  "uniprot_id": "A0A009IHW8",
  "name": "ABTIR_ACIB9",
  "length": 269,
  "entries": [
    {
      "id": "IPR035897",
      "element_type": "Homologous_superfamily",
      "match_id": "G3DSA:3.40.50.10140",
      "element_name": "Toll/interleukin-1 receptor homology (TIR) domain superfamily",
      "representative": false,
      "positions": [
        [
          80,
          266
        ]
      ]
    },
    {
      "id": "IPR000157",
      "element_type": "Domain",
      "match_id": "PF13676",
      "element_name": "Toll/interleukin-1 receptor homology (TIR) domain",
      "representative": false,
      "positions": [
        [
          138,
          231
        ]
      ]
    },
    {
      "id": "IPR000157",
      "element_type": "Domain",
      "match_id": "PS50104",
      "element_name": "Toll/interleukin-1 receptor homology (TIR) domain",
      "representative": true,
      "positions": [
        [
          133,
          266
        ]
      ]
    },
    ...
  ],
  "secondary_structs": [
    {
      "dssp_type": "Alphahelix",
      "start": 3,
      "end": 21
    },
    {
      "dssp_type": "Turn",
      "start": 21,
      "end": 22
    },
    {
      "dssp_type": "Turn",
      "start": 24,
      "end": 26
    },
    ...
  ]
}
```


[5] https://pdb-redo.eu/dssp/about

### A.1.2 SUBSTRUCTURE COMPOSITION ANALYSIS

Here we provide exploratory plots to give a sense of the overall size and composition of the different types of substructures contained within the Magneton dataset. These plots show data collected from the SwissProt dataset prior to filtering.

As shown in Table 1, we find that the typical length of a substructure varies widely, with domains spanning hundreds of residues and various functional sites spanning tens of residues (Appendix Figures A.1, A.2). We also find some binding site outliers in terms of length, which may be indicative of annotation artifacts. Despite the wide variance in length for substructures, we find that the amount of the protein they cover is relatively consistent for domains and functional sites (Appendix Figure A.4). We additionally inspect the amino acid composition of the various substructure types, both at the individual amino acid level (Appendix Figures A.5, A.5), and with amino acids grouped by chemical characteristics of their sidechains (Appendix Figure A.7).

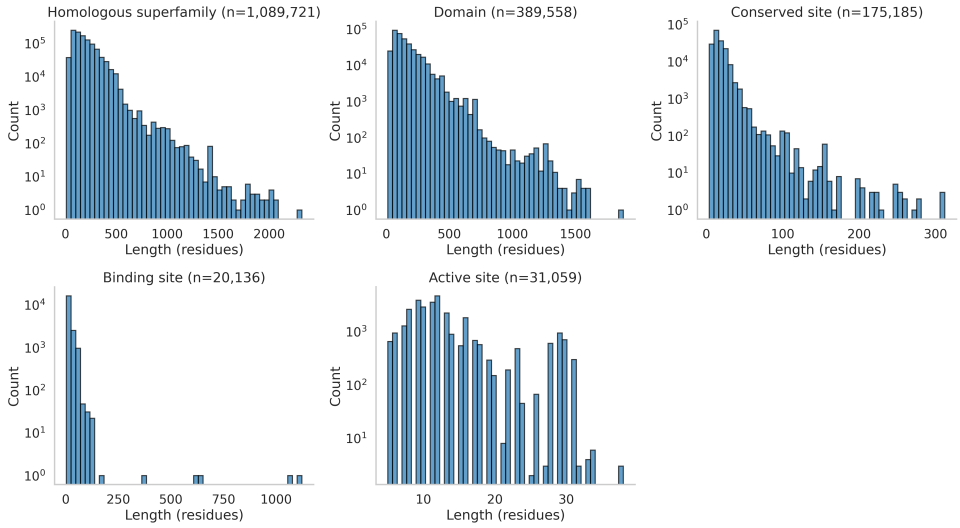



**Figure A.1:** Distribution of substructure lengths by type. Here, length is defined as the total number of residues contained within the substructure, regardless of whether they are contiguous within the sequence.

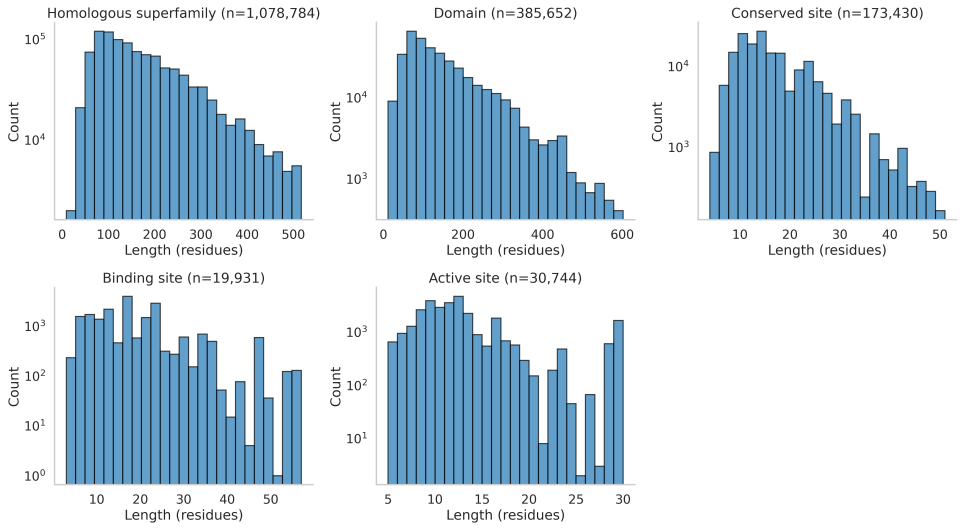



**Figure A.2:** Distribution of substructure lengths by type. Same as Figure A.1, but filtered to remove outliers (greater than 99th percentile of length within that type).

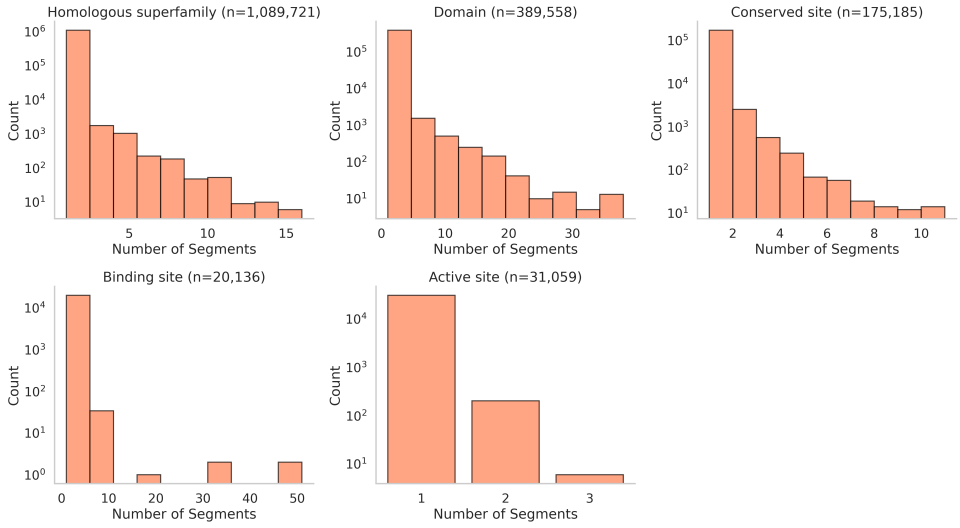



**Figure A.3:** Distribution of number of contiguous segments per substructure, by type.

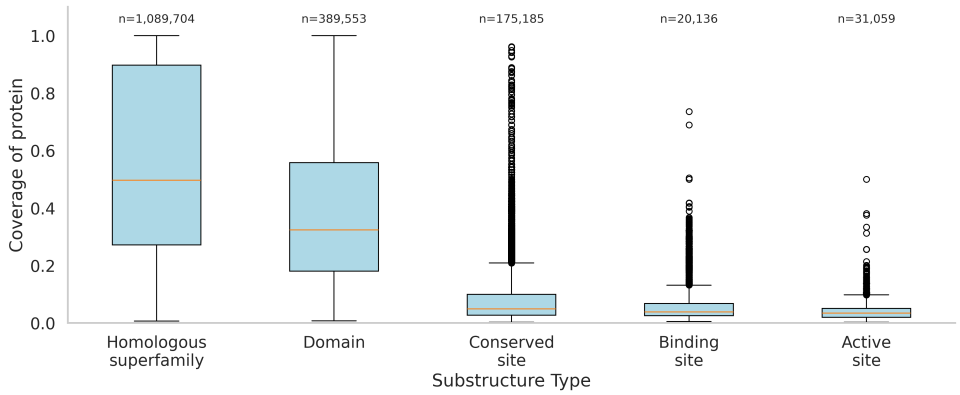



**Figure A.4:** Distribution of protein coverage per substructure. Protein coverage is defined as the total number of residues contained within the substructure divided by the length of the protein.

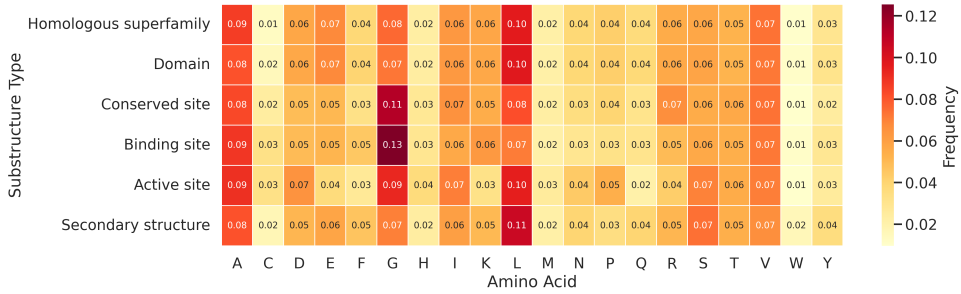



**Figure A.5:** Amino acid composition of substructure types. Each row shows the amino acid composition of that substructure type. Rows sum to 1.

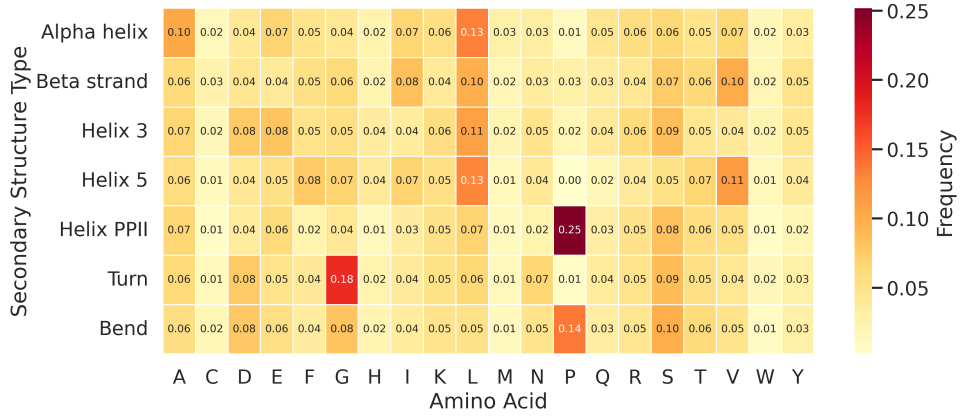



**Figure A.6:** Amino acid composition of granular secondary structure types. Each row shows the amino acid composition of that secondary substructure type. Rows sum to 1.

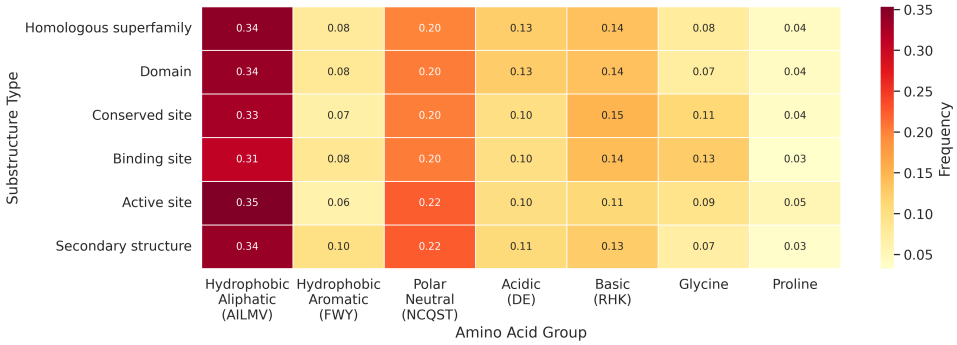



**Figure A.7:** Amino acid composition of substructure types, with amino acids grouped by side chain chemical properties. Each row shows the amino acid composition of that substructure type. Rows sum to 1.

### A.1.3 SUBSTRUCTURE DATASET FILTERING

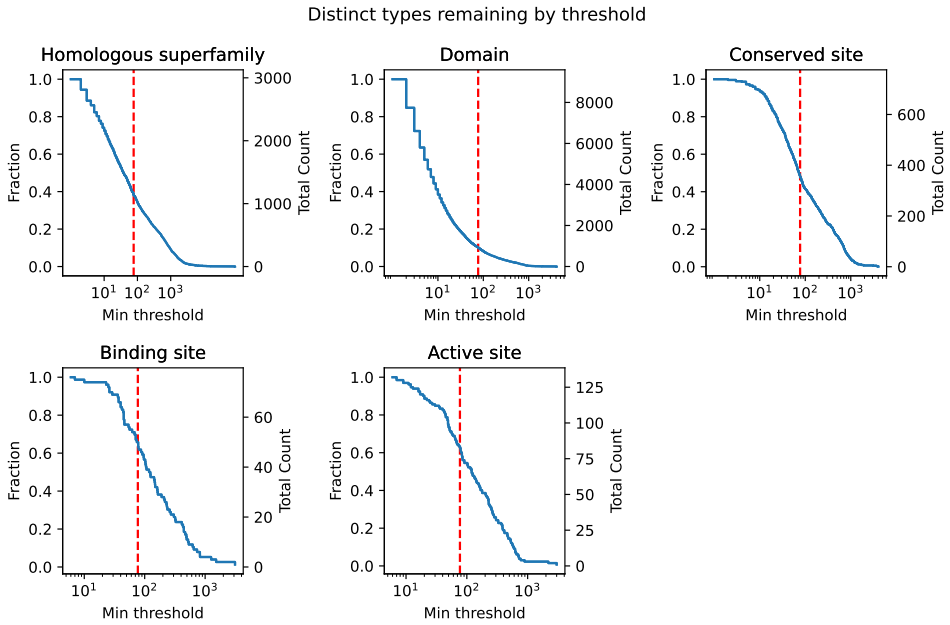



**Figure A.8:** Inverse CDF of **unique types** retained at a given count threshold. The $x$-axis specifies the minimum count for a substructure type to be retained, the left $y$-axis shows the fraction of all unique types retained at the given threshold, and the right $y$-axis shows the absolute count of unique types retained. Facets show different classes of substructural elements. The vertical dashed red lines show the threshold selected for downstream substructure-tuning.

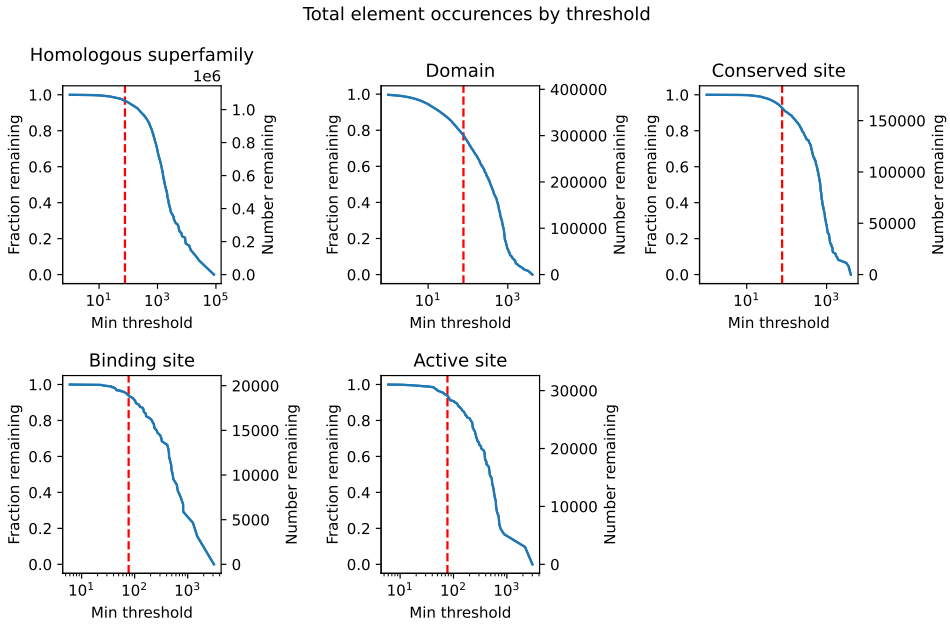



**Figure A.9:** Inverse CDF of **total occurrences** retained at a given count cutoff. This is analogous to Figure A.8 above, but showing total occurrences of substructures rather than unique types, demonstrating that for classes like domains, the majority of annotations come from a small number of domain types.

| Model | EC | GO:BP | GO:CC | GO:MF | Localization (Accuracy) | | Thermostability (Spearman's $\rho$) | Human PPI (AUROC) |
|---|---|---|---|---|---|---|---|---|
| | $F_{\max}$ | | | | Binary | Subcellular | | |
| ESM-C 300M | 0.688 | 0.307 | 0.416 | 0.429 | 0.871 | 0.703 | 0.648 | 0.917 |
| +ST (original, $\geq 75$) | 0.761 | 0.325 | 0.403 | 0.488 | 0.879 | 0.681 | 0.660 | 0.933 |
| +ST ($\geq 25$) | 0.802 | 0.32 | 0.399 | 0.526 | 0.851 | 0.693 | 0.64 | 0.917 |
| +ST ($\geq 10$) | 0.792 | 0.327 | 0.401 | 0.514 | 0.89 | 0.728 | 0.64 | 0.852 |

**Table A.1: Effect of substructure frequency cutoff on protein-level task performance.** The performance of substructure-tuning is robust to the presence of rare substructures.

| Substructure type | Full SwissProt set | Count $\geq 75$ | Count $\geq 25$ | Count $\geq 10$ |
|---|---|---|---|---|
| Homologous superfamily | 3511 | 1133 | 1685 | 2159 |
| Domain | 15868 | 917 | 1964 | 3506 |
| Conserved site | 748 | 356 | 573 | 691 |
| Binding site | 76 | 48 | 72 | 74 |
| Active site | 133 | 82 | 114 | 127 |

**Table A.2: Number of substructure types included in the dataset at different count cutoffs.** Reducing the cutoff down to 10 greatly expands the number of substructure types included in the dataset.

| Model | EC | GO:BP | GO:CC | GO:MF | Localization (Accuracy) | | Thermostability (Spearman's $\rho$) | Human PPI (AUROC) | FLIP_bind ($F_{\max}$) | Biolip binding (AUROC) | Biolip catalytic (AUROC) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $F_{\max}$ | | | | Binary | Subcellular | | | | | |
| ESM-C 300M | 0.688 | 0.307 | 0.416 | 0.429 | 0.871 | 0.703 | 0.648 | 0.917 | 0.367 | 0.851 | 0.923 |
| +ST (original) | 0.761 | 0.325 | 0.403 | 0.488 | 0.879 | 0.681 | 0.660 | 0.933 | 0.411 | 0.866 | 0.910 |
| +ST (exact match) | 0.789 | 0.317 | 0.403 | 0.5 | 0.88 | 0.687 | 0.647 | 0.892 | 0.372 | 0.847 | 0.879 |
| +ST (similar seq) | 0.777 | 0.317 | 0.385 | 0.507 | 0.83 | 0.686 | 0.645 | 0.889 | 0.375 | 0.849 | 0.912 |

**Table A.3: Effect of stringent data split on substructure-tuning performance.**

### A.1.4 Dataset leakage analyses

It's generally standard practice when training protein models to consider the dataset used for self-supervised learning as distinct from the downstream evaluation sets, presumably this is because the sequence or structure data is considered sufficiently far removed from the labels used in downstream evaluations (e.g. GO terms, experimental fitness values, etc.). However, substructural annotations are different from sequence or global structure in that they can be more directly tied to protein function. For this reason, we performed a series of analyses to understand whether any of the effects of substructure-tuning could be attributed to data leakage between the substructure-tuning training set and the test sets of any of the evaluation benchmarks.

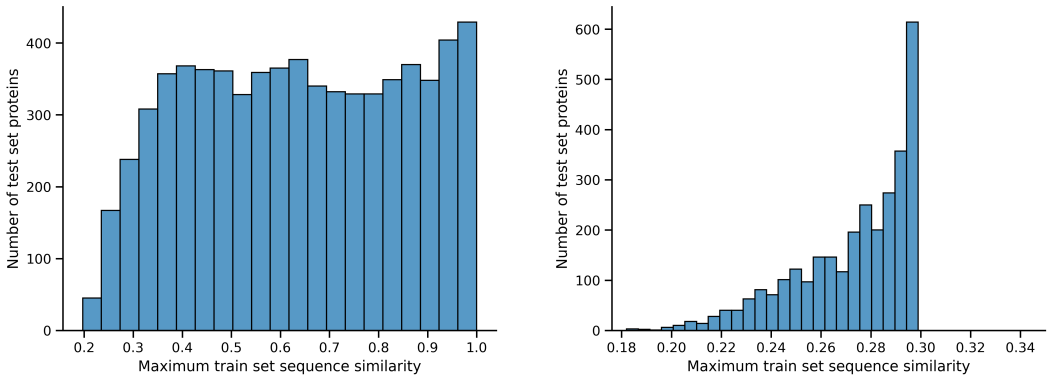
First, we performed a direct experiment by constructing two versions of more stringent dataset splits:

- "Exact match": To construct this split, we identified all proteins in the Magneton train set that were also contained in the test split of any evaluation benchmark, based on UniProt ID. We then removed all such proteins and all of their corresponding AFDB50 clusters from the Magneton train set. This removed 31,191 of the 423,885 proteins in the Magneton training split.
- "Similar seq": To construct an even more stringent split, we collected the amino acid sequences of all proteins in the test split of any evaluation benchmark, and aligned these sequences to the proteins in the Magneton train set. We then removed from the Magneton train set any protein with greater than 30% sequence similarity and 80% overlap with any evaluation test set protein. Alignments were performed using MMseqs2 (Steinegger & Söding, 2017; Kallenborn et al., 2025) with sequence similarity defined using the `fident` output (command: `mmseqs easy-search eval_seqs.fa train_seqsDB tmp output.tsv --cov-mode 1 -c 0.8 --alignment-mode 3`). By constructing the dataset split in this manner, we ensure that no protein contained in the Magneton train set has more than 30% sequence similarity to any protein in any of the evaluation benchmark test sets (Figure A.10). This process removed 131,440 of the 423,885 proteins in the Magneton training split.

We then performed substructure-tuning of ESM-C 300M using active, binding, and conserved site annotations with each of these stringent training splits and evaluated the resulting models on our evaluation suite (Table A.3). We found that benchmark performance was largely similar to that achieved with the original training, providing evidence that data leakage does not drive any of the effects of substructure-tuning.

As a further step, we leveraged mappings from Gene Ontology terms to InterPro entries [6] to check whether substructure-tuning's improvement on GO MF prediction could be attributed to this mapping. In particular, we checked if the improvement in predicting a GO MF term after substructure-tuning correlated with the representation of its corresponding domains in the Magneton train set. We found that the performance improvement for predicting a given GO MF does not depend on its corresponding domain's representation in the Magneton train set, again supporting the lack of data leakage (Figure A.11, Table A.4).

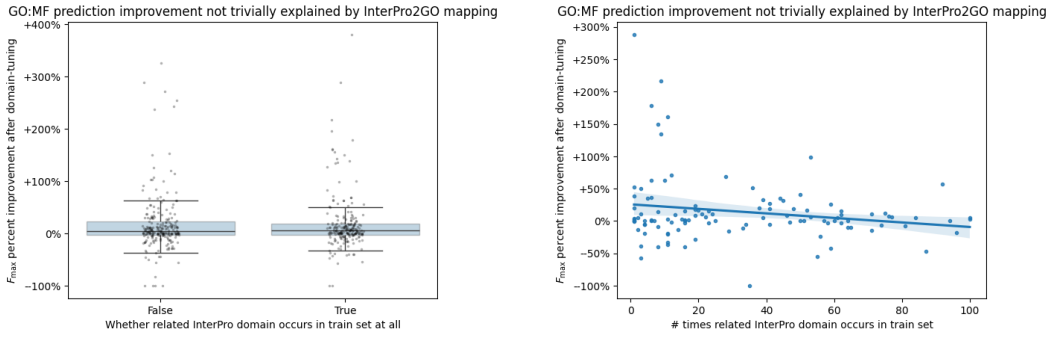
[6] https://www.ebi.ac.uk/GOA/InterPro2GO




**(a)** Sequence similarities before filtering.

**(b)** Sequence similarities after filtering

**Figure A.10:** Distribution of sequence similarities (`fident` from MMseqs2) between proteins in the test set of any evaluation benchmark and proteins in the Magneton training set. Plotted values are the maximum similarity to any Magneton training set protein.




**(a)** Performance improvement split by inclusion only.

**(b)** Performance by number of occurrences in Magneton train set.

**Figure A.11:** GO:MF prediction improvement by representation of associated domain in the Magneton train set.

| | % $F_{\max}$ improvement after substructure-tuning | | | |
|---|---|---|---|---|
| | Mean | 25th percentile | Median | 75th percentile |
| Domains in train set | +17.82% | -2.36% | +5.11% | +18.66% |
| Domains not in train set | +16.84% | -3.40% | +4.88% | +23.35% |

**Table A.4: GO:MF prediction improvement by domain presence in training set.** Improvement in predicting a GO:MF term after substructure-tuning does not correlate with the representation of its corresponding domains in the train set.

### A.1.5 MAGNETON EVALUATION DATASETS

Magneton contains eleven different benchmarking datasets, comprising 14 evaluation tasks. These evaluation tasks represent years of work from the scientific community, both in their original generation and later processing that we build upon, and we acknowledge and thank all of those involved. The included tasks are:

- **Human PPI prediction.** The goal of this task is to predict whether or not two proteins form an interacting pair. This is a binary classification task where the input is two proteins and the output is a binary label indicating interaction or no interaction. The evaluation metric is accuracy. The original dataset is sourced from Pan et al. (2010). We build off of processed data files from Su et al. (2023).
- **Gene Ontology and Enzyme Commission prediction.** The goal of these tasks are to predict the Gene Ontology (GO) or Enzyme Commission (EC) annotations for a protein. There are three categories of GO annotations: Molecular Function (MF), Cellular Component (CC), and Biological Process (BP), each of which captures a different aspect of protein biology and is treated as a separate benchmarking task, giving a total of four tasks. These are multilabel classification tasks where a protein can have multiple annotations. The evaluation metric is $F_{\max}$, the maximum $F_1$ score over possible thresholds. We source original data from Gligorijević et al. (2021).
- **Subcellular and binary localization.** The goal of these tasks is to predict a protein's localization either within multiple cellular compartments (subcellular localization) or whether the protein is membrane-bound or soluble (binary localization). The input is a single protein and this is either a multiclass or binary classification task. The evaluation metric is accuracy. The original dataset is sourced from Almagro Armenteros et al. (2017). We build off of processed data files from Su et al. (2023).
- **Thermostability.** The goal of this task is to predict the stability of a protein under extreme temperatures. The output is a continuous value indicating the thermostability, and the goal is to rank-order proteins according to their experimental values. The evaluation metric is Spearman rank correlation (Spearman's $\rho$) calculated against the experimental values. The original dataset is sourced from Rao et al. (2019).
- **Binding residue categorization.** The goal of this task is to predict whether a given residue binds three different types of ligands: metal ions, small molecules, or nucleic acids. This is a residue-level multilabel classification task. The evaluation metric is $F_{\max}$. The original dataset is sourced from Dallago et al. (2021).
- **Binding and catalytic site prediction.** The goal of these tasks are to predict whether a given residue is part of an annotated binding or catalytic site. This is a residue-level binary classification task. The evaluation metric is AUROC. The original datasets are sourced from experimentally determined structures curated by Zhang et al. (2024a). We build off of processed data files from Yuan et al. (2025).
- **Contact prediction.** The goal of this task is to predict whether two residues within the same protein are "in contact" with each other, which is defined as having alpha-carbon atoms within 8 angstroms of each other in the tertiary structure. The input is a single protein of length $L$ and the output is a $L \times L$ contact map, where element $i, j$ of the contact map is the predicted probability that residues $i$ and $j$ are in contact. The evaluation metric is Precision@L, which calculates precision over the top $L$ most confident contact predictions where $L$ is the protein length. This metric is further stratified into short, medium, and long-range contacts in which the possible residue pairs considered are those whose pairwise separation along the primary sequence is either in $[6, 10]$, $[12, 22]$, or $[24, L]$, respectively. The underlying data are experimentally determined structures from PDB, originally curated by Rao et al. (2019).
- **Zero-shot DMS variant effect prediction**. The goal of this task is to predict the effect of a single or multiple amino acid mutations on a protein's function. The input is the mutated sequence and the output is a continuous value representing fitness. The evaluation metric is Spearman's $\rho$ against the experimentally-determined fitness values from deep mutational scans (DMS) experiments. These tasks are zero-shot in that no supervised training for

variant effect prediction is performed. For each model, we use the author's recommended methods for VEP. The data is sourced from Notin et al. (2023).

For all datasets, protein structures are sourced from AlphaFold DB (Varadi et al., 2022) unless otherwise specified. Accounting of samples per split in each dataset are available in Table A.5.

| Task | Train | Validation | Test | Number of classes | Task type |
|---|---|---|---|---|---|
| EC | 14,466 | 1,599 | 1,715 | 538 | Multilabel |
| GO:BP | 21,470 | 2,393 | 3,394 | 1,943 | Multilabel |
| GO:CC | 9,793 | 1,118 | 3,394 | 320 | Multilabel |
| GO:MF | 22,621 | 2,495 | 3,394 | 489 | Multilabel |
| Subcellular localization | 8,741 | 2,190 | 2,744 | 10 | Multiclass |
| Binary localization | 5,473 | 1,335 | 1,728 | 2 | Binary |
| Thermostability | 5,020 | 636 | 1,329 | N/A | Regression |
| Binding residue categorization | 890 | 102 | 286 | 3 | Multilabel |
| Binding site prediction | 8,231 | 2,389 | 5,182 | 2 | Binary |
| Catalytic site prediction | 2,856 | 603 | 1,165 | 2 | Binary |
| Contact prediction | 20,653 | 209 | 40 | 2 | Binary |
| Variant effect prediction[1] | N/A | N/A | 217 | N/A | Regression |
| Human PPI prediction[2] | 26,313 | 234 | 180 | 2 | Binary |

**Table A.5:** Dataset sizes (in proteins) and number of classes for each benchmarking task.

[1] This is a zero-shot task, hence the lack of training and validation data. Samples correspond to assays, covering 2.3M mutations.

[2] Samples correspond to *pairs* of proteins rather than individual proteins.

## A.2 Training details

### A.2.1 Substructure classification and tuning

For our substructure prediction modules, we use single hidden-layer MLPs where the dimensionality of the hidden layer matches that of the underlying base model. We extract residue-level embeddings from the final hidden layer of the base model, and use mean pooling across a substructure's constituent residues to construct a single embedding per substructure. When training with multiple categories of substructures, we use a separate prediction module for each category. The training loss is the sum of the classification losses across all categories.

We train using AdamW (Loshchilov & Hutter, 2019) ($\beta_1 = 0.9, \beta_2 = 0.999$), learning rates of $10^{-3}$ and dropout rate of 0.1 for the prediction heads, learning rate of $10^{-5}$ for the base model, and EWC weight of 400 (as used by original authors). Training proceeded until convergence of validation loss. All runs used batches of 32 proteins with a variable number of substructures per protein. All training was performed using `bfloat16` on one to four NVIDIA A100 GPUs.

**Elastic weight consolidation.** Briefly, EWC uses the diagonal of the Fisher information matrix $\mathcal{F}$ as weights on a loss that regularizes towards the original model parameters, $\theta_0$:

$$L = L_c(\theta) + \sum_i \frac{\lambda}{2} F_i \left(\theta_i - \theta_{0_i}\right)^2$$

where $L_c$ is the substructure classification loss. $\mathcal{F}$ can be estimated at the beginning of training as the squared gradients of the original loss with respect to the model parameters using the training set. In our case, the original loss corresponds to the training objective of the underlying model (*e.g.* masked amino acid prediction for ESM models, masked amino acid prediction in presence of structure tokens for ProSST or SaProt). In practice, the $\mathcal{F}$ is estimated by making a single pass over the training set, running backwards passes using the original loss, and averaging the squared gradients over minibatches.

While similar to a $L_2$ loss, EWC has two advantages over a simple $L_2$ or weight decay regularization: (1) weights are decayed towards the original weights of the base model, (2) per-parameter weights are applied which correspond to the importance of that parameter for the original task. We

| | Frozen base model | Full finetuning | EWC finetuning |
|---|---|---|---|
| Validation MLM loss | 1.24 | 2.30 | 1.75 |

**Table A.6: Masked language modeling loss under different finetuning strategies.**

| Model | Substructures used for tuning | EC | GO:BP | GO:CC | GO:MF | Zero-shot DMS (Spearman's $\rho$) | Binary Localization (Accuracy) | Subcellular Localization (Accuracy) |
|---|---|---|---|---|---|---|---|---|
| | | $F_{\max}$ | | | | | | |
| ESM-C (300M) | N/A | 0.688 | 0.307 | 0.416 | 0.429 | 0.432 | 0.871 | 0.703 |
| EWC | Domain | 0.776 | 0.307 | 0.403 | 0.501 | 0.340 | 0.811 | 0.640 |
| No EWC | Domain | 0.831 | 0.311 | 0.386 | 0.531 | TBD | 0.802 | 0.643 |

**Table A.7: Effect of EWC on substructure-tuning performance.** Training with EWC offers a good balance of maintaining improvement on function-related tasks while mitigating performance decreases on other tasks.

selected EWC due to its simplicity and ease of use, as the estimate of $\mathcal{F}$ can be calculated a single time and used for the remainder of training or for other training runs using the same base model, as opposed to alternate methods like replay buffers. For more details, please refer to Kirkpatrick et al. (2017).

Empirically, use of EWC is motivated by an increase in the masked language modeling loss and validation perplexity when performing substructure-tuning as shown in Table A.6 and Figure A.12. We additionally observe that EWC offers a good balance of maintaining improvement on function-related tasks while mitigating performance decreases on other tasks (Table A.7).

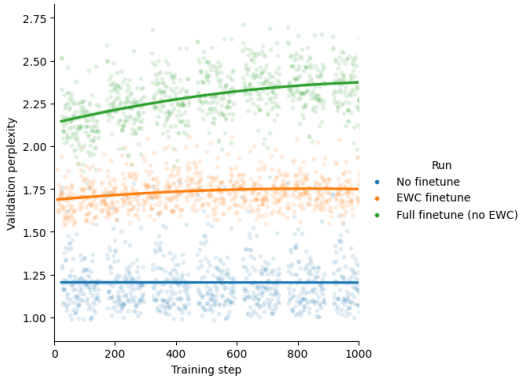



**Figure A.12: Validation perplexity for finetuning ESM-C 300M on domain annotations with and without EWC.**

### A.2.2 Downstream task training details

We use different head models for different scales of supervised downstream tasks:

- **Protein-level.** For protein-level tasks such as GO term prediction, we construct a protein-level representation for each protein following author's recommendations. For ESM-2, we use the final embedding of the `CLS` token as the protein-level embedding. For all other models, we mean pool the final hidden layer representations of all residue tokens (*i.e.* excluding `CLS`, `EOS` , and `PAD` tokens). Prediction heads are then single hidden-layer MLPs with hidden dimensionality matching the hidden dimension of the underlying model.
- **Residue-level.** For residue-level tasks such as binding site prediction, we use a head model consisting of a single 1-dimensional convolutional layer with zero-padding and filter width 5, followed by a nonlinearity and linear layer to the final output.
- **Protein-protein interaction.** For protein-protein interaction prediction, we extract protein-level embeddings as above, concatenate the embeddings for the two input proteins, and pass into a single hidden-layer MLP.

- **Contact prediction**. For contact prediction, we use the `EsmContactPredictionHead` from the `transformers` Python package, which trains a linear regression on top of attention weights from all attention heads in the underlying model.

To train the models, we used AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$) with a learning rate of $10^{-2}$, weight decay of $10^{-2}$, dropout rate of 0.1, and a batch size of 32. Training proceeded for a maximum of 20 epochs, selecting the best model based on validation set performance. When performing full task-specific finetuning, we use a learning rate of $2 \times 10^{-5}$ for the base model and scale the number of GPUs and gradient accumulation steps accordingly to maintain a batch size of 32. All training was performed using `bfloat16` on one to four NVIDIA A100 GPUs.

### A.2.3 TASK-SPECIFIC FINETUNING

| Model | EC | GO:BP | GO:CC | GO:MF | Localization (Accuracy) | | Thermostability (Spearman's $\rho$) |
|---|---|---|---|---|---|---|---|
| | $F_{\max}$ | | | | Binary | Subcellular | |
| ESM2-150M | 0.911 | 0.352 | 0.451 | 0.658 | 0.928 | 0.810 | 0.694 |
| +ST | 0.910 | 0.349 | 0.444 | 0.658 | 0.936 | 0.791 | 0.702 |
| ESM2-650M | 0.910 | 0.356 | 0.446 | 0.662 | 0.933 | 0.824 | 0.703 |
| +ST | 0.914 | 0.360 | 0.457 | 0.665 | 0.930 | 0.793 | 0.703 |
| ESM-C 300M | 0.916 | 0.368 | 0.470 | 0.667 | 0.932 | 0.806 | 0.693 |
| +ST | 0.920 | 0.355 | 0.454 | 0.669 | 0.941 | 0.804 | 0.693 |
| ESM-C 600M | 0.920 | 0.374 | 0.469 | 0.669 | 0.917 | 0.812 | 0.705 |
| +ST | 0.924 | 0.372 | 0.468 | 0.667 | 0.931 | 0.828 | 0.699 |
| ProSST-2048 | 0.911 | 0.319 | 0.439 | 0.631 | 0.912 | 0.760 | 0.673 |
| +ST | 0.901 | 0.336 | 0.427 | 0.638 | 0.923 | 0.744 | 0.670 |

**Table A.8:** Evaluation task performance for models with and without substructure-tuning, and **full finetuning for the downstream task.** In this regime, we find that task-specific finetuning largely results in similar models, showing that imbuing substructural information via supervised finetuning may be brittle in the face of aggressive task-specific finetuning.

## A.3 MECHANISTIC ANALYSES

### A.3.1 EMBEDDING ANALYSES

To explore cross-scale consistency of different substructure classes and directly assess the utility of a shared bottleneck, we performed substructure-tuning ESM-C 300M using a single classification head operating over all of the active, binding, and conserved site substructure types. We found that using a single shared head is consistent with using separate heads in terms of performance boosts in function-centric tasks over the base model, but with varying effects on other tasks (Table A.9).

### A.3.2 GRADIENT CONFLICT ANALYSIS

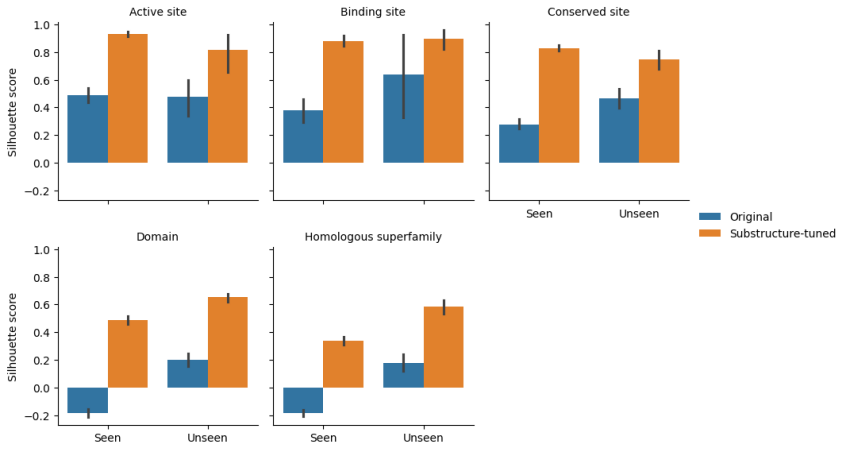
The gradient conflict analysis was performed as follows:

- Begin with a base ESM-C 300M model (i.e. no substructure-tuning)
- For each evaluation dataset listed in Table A.10, compute the gradient for the associated loss function for 1000 batches of batch size 32.
- For substructure-tuning with active, binding, and conserved-site annotations, compute the gradient for the substructure classification objective for 1000 batches of batch size 32 (where batch size refers to the number of proteins per batch, each of which must contain at least one substructure).
- Compute pairwise similarities between all gradient vectors.

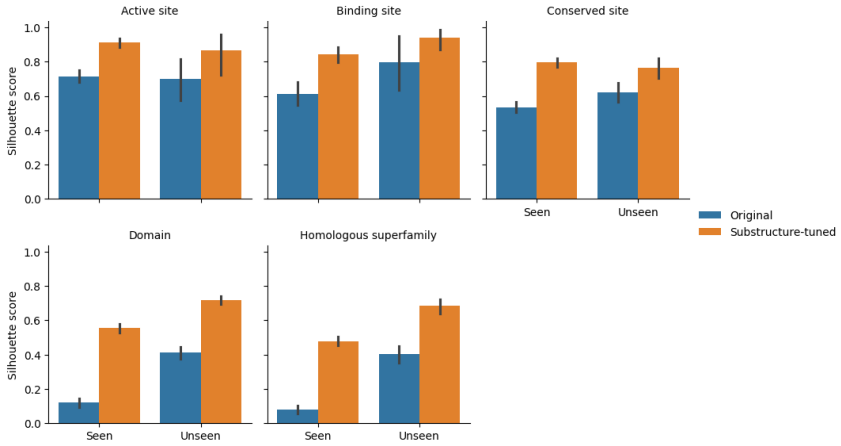
The final results of this analysis are shown for comparison of gradient vectors derived from batches from the same task (Table A.10a) and from batches from different tasks (Table A.10b).

### A.3.3 SUBSTRUCTURE ATTRIBUTION ANALYSIS

To understand correlation of substructure-tuned embeddings with known functional motifs, we performed an explainability analysis by first identifying all proteins in the GO:MF test set whose GO:MF annotations could be attributed to a specific domain (accomplished via InterPro2GO mappings), yielding 81 (protein, domain, GO:MF) triplets. For each protein, we computed a saliency map to calculate residue contributions to the prediction of the associated GO:MF term. We then calculated the total contribution of residues contained within the associated domain and normalized this by the total contribution across all residues in the protein. For ESM-C 300M before and after substructure-tuning, we find that attributions to domain residues increased by an average of




**(a)** Average silhouette scores before and after substructure-tuning for ESM-C 300M




**(b)** Average silhouette scores before and after substructure-tuning for SaProt 650M

**Figure A.13: Average silhouette scores of each substructure type class, before and after substructure-tuning, for both seen and unseen classes.** Despite never training on any examples of the unseen substructure types, substructure-tuning results in more consistent representations of these substructures.

| Model | EC | GO:BP | GO:CC | GO:MF | Localization (Accuracy) | | Thermostability (Spearman's $\rho$) | Human PPI (AUROC) | FLIP_bind ($F_{\max}$) | Biolip binding (AUROC) | Biolip catalytic (AUROC) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $F_{\max}$ | | | | Binary | Subcellular | | | | | |
| ESM-C 300M | 0.688 | 0.307 | 0.416 | 0.429 | 0.871 | 0.703 | 0.648 | 0.917 | 0.367 | 0.851 | 0.923 |
| +ST (original, separate heads) | 0.761 | 0.325 | 0.403 | 0.488 | 0.879 | 0.681 | 0.660 | 0.933 | 0.411 | 0.866 | 0.910 |
| +ST (single head) | 0.777 | 0.304 | 0.392 | 0.513 | 0.872 | 0.704 | 0.666 | 0.887 | 0.39 | 0.85 | 0.888 |

**Table A.9: Effect of using a single shared classification head for substructure-tuning.** The effect of using a single shared head is consistent with using separate heads in terms of performance boosts in function-centric tasks over the base model, but with varying effects on other tasks.

17%, indicating increased usage of substructures for function prediction after substructure-tuning (Appendix Figure A.15).

As a concrete example, we provide a saliency map for the protein `Q15436` and the prediction of its annotated `GO:0008270` term. The `GO:0008270` GO term can be directly mapped to the InterPro domain `IPR006895` contained at positions 58 to 98 in the protein. We can observe that the untuned model exhibits less attribution to the relevant domain (Appendix Figure A.16) than the substructure-tuned model (Appendix Figure A.17).

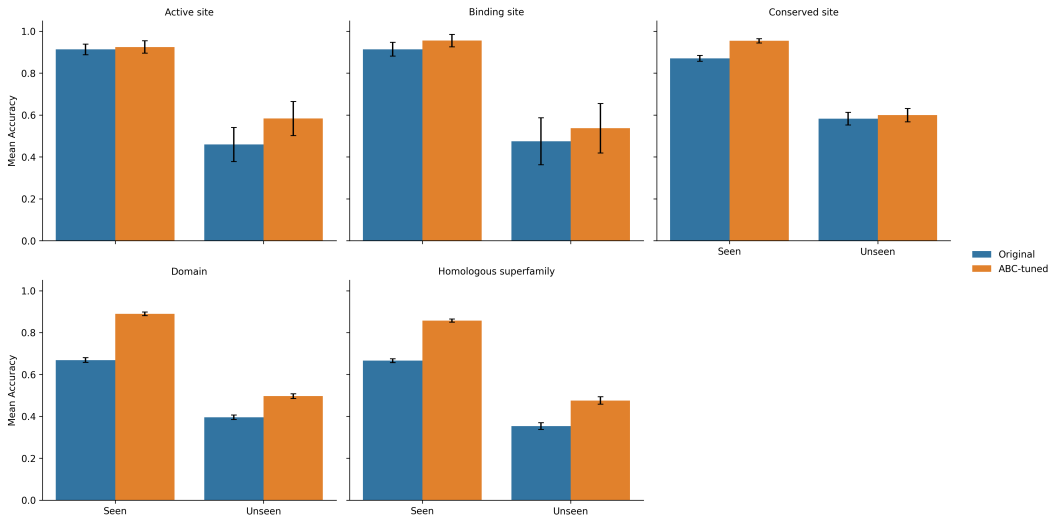



**Figure A.14: Average accuracy of kNN-based classification of each substructure class using embeddings, before and after substructure-tuning, for both seen and unseen classes.**

| Task | GO:MF | EC | Binding residue | Functional site prediction (binding) | Functional site prediction (catalytic) | Substructure classification |
|---|---|---|---|---|---|---|
| Gradient cosine similarity | 0.957 ± 0.004 | 0.956 ± 0.004 | 0.961 ± 0.007 | 0.964 ± 0.004 | 0.973 ± 0.022 | 0.095 ± 0.077 |

**(a)** Within task gradient similarity (mean ± std deviation)

| Task | GO:MF | EC | Binding residue | Functional site prediction (binding) | Functional site prediction (catalytic) |
|---|---|---|---|---|---|
| Gradient cosine similarity | -0.006 ± 0.047 | 0.003 ± 0.027 | 0.022 ± 0.056 | 0.002 ± 0.055 | -0.005 ± 0.038 |

**(b)** Task gradient similarity with substructure classification gradient (mean ± std deviation)

**Table A.10: Gradient conflict analysis for substructure-tuning.** These results suggest that the task-specific effects of substructure-tuning is not due to a simple global misalignment between the substructure objective and certain downstream tasks.

## A.4 Ablations

### A.4.1 Alternate pooling techniques

We initially developed our approach using mean pooling to construct substructure representations due to its simplicity, but also explore additional aggregation techniques to help expand the completeness of our work. We note that the choice of a simple pooling over a learnable aggregation was intentional: since the goal of substructure-tuning is to use the supervised substructure classification problem to tune the parameters of the underlying protein model, we wanted to encourage learning to occur within the base model rather than in a parameterized pooling module.

To investigate this, we performed substructure-tuning using max pooling and a learnable attention pooling. Both of these experiments performed substructure-tuning on ESM-C 300M using active, binding, and conserved site annotations. We find that substructure-tuning is robust to the exact method used for pooling residue-level embeddings to substructural embeddings (Appendix Table A.11).

### A.4.2 Additional models

We additionally conducted a direct comparison of substructure-tuning to methods that use global structure to tune sequence models would help contextualize our work. To ensure a fair comparison,

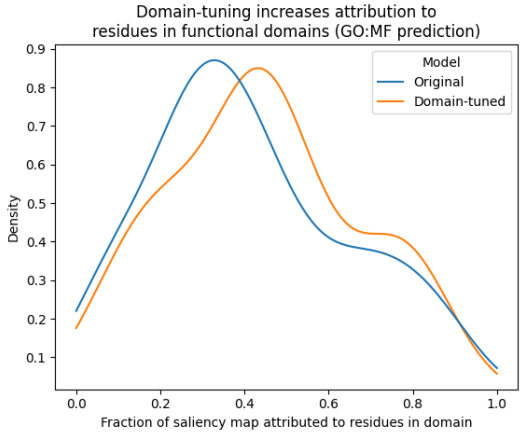



**Figure A.15: Saliency map indicating attributions to domain residues during GO:MF predictions before and after domain-tuning.**

M T T Y L E F I Q Q N E E R D G V R F S W N V W P S S R L E A T R M V V P V A
A L F T P L K E R P D L P P I Q Y E P [DOM_START] V L C S R T T C R A V L N P L
C Q V D Y R A K L W A C N F C Y Q R N Q F P P S Y [DOM_END] A G I S E L N Q P
A E L L P Q F S S I E Y V V L R G P Q M P L I F L Y V V D T C M E D E D L Q A L
K E S M Q M S L S L L P P T A L V G L I T F G R M V Q V H E L G C E G I S K S Y
V F R G T K D L S A K Q L Q E M L G L S K V P L T Q A T R G P Q V Q Q P P P S
N R F L Q P V Q K I D M N L T D L L G E L Q R D P W P V P Q G K R P L R S S G
V A L S I A V G L L E C T F P N T G A R I M M F I G G P A T Q G P G M V V G D
E L K T P I R S W H D I D K D N A K Y V K K G T K H F E A L A N R A A T T G H
V I D I Y A C A L D Q T G L L E M K C C P N L T G G Y M V M G D S F N T S L F
K Q T F Q R V F T K D M H G Q F K M G F G G T L E I K T S R E I K I S G A I G P
C V S L N S K G P C V S E N E I G T G G T C Q W K I C G L S P T T T L A I Y F E
V V N Q H N A P I P Q G G R G A I Q F V T Q Y Q H S S G Q R R I R V T T I A R N
W A D A Q T Q I Q N I A A S F D Q E A A A I L M A R L A I Y R A E T E E G P D
V L R W L D R Q L I R L C Q K F G E Y H K D D P S S F R F S E T F S L Y P Q F M
F H L R R S S F L Q V F N N S P D E S S Y Y R H H F M R Q D L T Q S L I M I Q P
I L Y A Y S F S G P P E P V L L D S S S I L A D R I L L M D T F F Q I L I Y H G E
T I A Q W R K S G Y Q D M P E Y E N F R H L L Q A P V D D A Q E I L H S R F P
M P R Y I D T E H G G S Q A R F L L S K V N P S Q T H N N M Y A W G Q E S G A
P I L T D D V S L Q V F M D H L K K L A V S S A A

**Figure A.16: Example saliency map before substructure-tuning.** `[DOM_START]` and `[DOM_END]` denote the beginning and end of the domain within the sequence respectively.

we have integrated ESM-S into Magneton and evaluated it on our benchmark suite (Appendix Table A.12).

We find that substructure-tuning compares favorably to ESM-S, with generally larger improvements on function-related tasks and smaller performance drop-offs in other tasks. We also note that substructure-tuning is also useful for models that already incorporate global structure information, something which has not been shown for ESM-S's methodology.

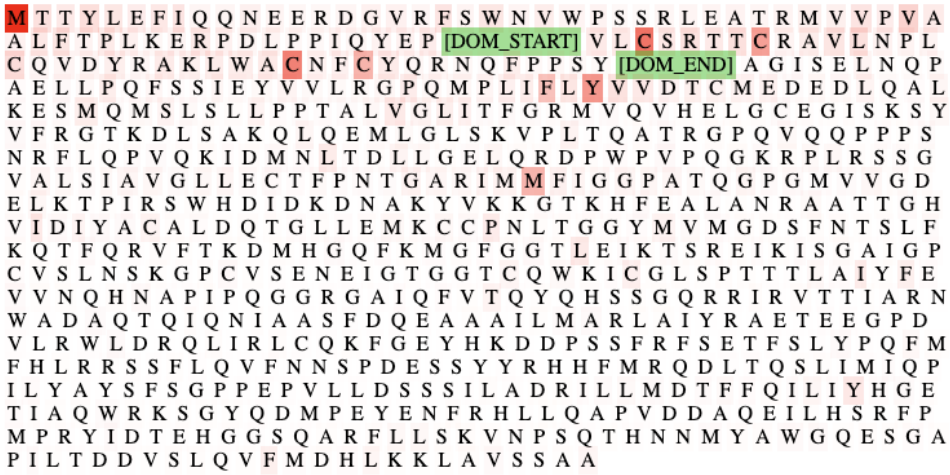



**Figure A.17: Example saliency map after substructure-tuning.** `[DOM_START]` and `[DOM_END]` denote the beginning and end of the domain within the sequence respectively.

| Model | EC | GO:BP | GO:CC | GO:MF | Localization (Accuracy) | | Thermostability (Spearman's $\rho$) | Human PPI (AUROC) |
|---|---|---|---|---|---|---|---|---|
| | $F_{max}$ | | | | Binary | Subcellular | | |
| ESM-C 300M | 0.688 | 0.307 | 0.416 | 0.429 | 0.871 | 0.703 | 0.648 | 0.917 |
| +ST (original, mean) | 0.761 | 0.325 | 0.403 | 0.488 | 0.879 | 0.681 | 0.660 | 0.933 |
| +ST (max) | 0.777 | 0.304 | 0.392 | 0.513 | 0.872 | 0.704 | 0.666 | 0.887 |
| +ST (attention) | 0.778 | 0.317 | 0.392 | 0.505 | 0.815 | 0.675 | 0.643 | 0.89 |

**Table A.11: Effect of pooling method on substructure-tuning performance.** Substructure-tuning is robust to the exact method used for pooling residue-level embeddings to substructural embeddings.

| Model | EC | GO:BP | GO:CC | GO:MF | Localization (Accuracy) | | Thermostability (Spearman's $\rho$) | Human PPI (AUROC) | Binding residue ($F_{max}$) | Functional site prediction | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $F_{max}$ | | | | Binary | Subcellular | | | | Binding | Catalytic (AUROC) |
| ESM2-150M | 0.727 | 0.316 | 0.416 | 0.441 | 0.869 | 0.694 | 0.627 | 0.933 | 0.379 | 0.871 | 0.910 |
| +ST | 0.742 | 0.324 | 0.415 | 0.473 | 0.866 | 0.679 | 0.582 | 0.919 | 0.327 | 0.852 | 0.890 |
| ESM-S 150M | 0.760 | 0.315 | 0.373 | 0.460 | 0.866 | 0.677 | 0.631 | 0.917 | 0.352 | 0.871 | 0.885 |
| ESM2-650M | 0.755 | 0.319 | 0.431 | 0.486 | 0.876 | 0.710 | 0.643 | 0.939 | 0.366 | 0.849 | 0.912 |
| +ST | 0.745 | 0.321 | 0.440 | 0.534 | 0.895 | 0.749 | 0.655 | 0.935 | 0.362 | 0.851 | 0.927 |
| ESM-S 650M | 0.794 | 0.327 | 0.422 | 0.497 | 0.896 | 0.679 | 0.653 | 0.912 | 0.363 | 0.876 | 0.902 |
| S-PLM (650M) | 0.566 | 0.284 | 0.375 | 0.482 | 0.910 | 0.622 | 0.682 | 0.911 | 0.358 | 0.500 | 0.500 |

**Table A.12: Comparison of substructure-tuning with global structure tuning methods.** Substructure-tuning compares favorably to ESM-S and S-PLM, with generally larger improvements on function-related tasks and larger performance drop-offs in other tasks.