Evaluating out of distribution generalization of protein language models

Anonymous Authors[1]

Abstract

Protein language models (pLMs) are increasingly used for protein function prediction tasks such as detecting and annotating homologous domains in sequences. However, because they are pretrained on such a broad sample of known protein sequence space, it becomes difficult to construct downstream train/test splits that are truly independent of pretraining data and therefore to assess whether downstream performance reflects genuine generalization. We study this question in a controlled setting by constructing large-scale train, validation, and test splits with no detectable cross-split homology and using them for both pLM pretraining and downstream evaluation. Holding architecture, compute, and downstream training fixed, we vary only the amount of overlap between pretraining and test data and measure its effect on domain annotation sensitivity. Pretraining overlap substantially increases test-set sensitivity even when training loss, validation perplexity, and validation-set downstream performance remain nearly unchanged, showing that overlap with pretraining data can inflate apparent performance without improving broader generalization. We also find that domain-relevant signal emerges early during masked language model pretraining.

1. Introduction

Protein sequence databases now contain hundreds of millions to billions of sequences, yet much of this sequence space remains functionally uncharacterized. Less than 0.5% of known proteins have experimental functional annotations in the Gene Ontology (Littmann et al., 2021). Since experimental characterization has been outpaced by the rapid growth of sequence databases, protein function is often inferred computationally. In practice, inference typically proceeds through domain annotation: the identification of conserved structural and functional units whose identity and arrangement provide some of the clearest clues to a protein's biological role. At the same time, these expanding sequence databases contain a vast record of evolutionary information that, if exploited effectively, could extend function prediction far beyond the reach of current methods.

Protein language models (pLMs) offer one promising way to exploit evolutionary information at this scale. Trained on massive corpora of unlabeled protein sequences, pLMs have become an increasingly common approach to protein representation learning (Rives et al., 2021; Brandes et al., 2022; Elnaggar et al., 2021). pLM-based methods have supported progress across several protein modeling tasks, including variant effect prediction (Meier et al., 2021), structure prediction (Lin et al., 2023; Hayes et al., 2025) and domain annotation (Sarkar et al., 2026). However, an important open question is whether these gains generalize to remote homologs, where evolutionary relationships are typically weakly reflected in sequence. This question matters for predictions on proteins that are evolutionarily distant from the training data, for large-scale annotation pipelines in which misannotations can propagate through growing databases to become training signal for future models, and for downstream AI systems that use pLM-derived annotations or embeddings as part of larger retrieval and reasoning pipelines (Huang et al., 2025; Su et al., 2025; Fallahpour et al., 2026).

Assessing generalization to remote homologs is difficult because pLMs are pretrained on very large sequence corpora. As a result, apparently held-out evaluation sequences may still be close homologs of proteins already present in the pretraining data. This creates a risk of *data leakage*: information from evaluation sequences enters pretraining through identical or homologous sequences, inflating apparent downstream performance (Bernett et al., 2024; Hermann et al., 2024) as shown in Figure 1A. A rigorous test of generalization to remote homologs therefore requires constructing splits at the scale of language-model pretraining itself, not only at the level of downstream task training and evaluation.

Assessing generalization to remote homologs is difficult because pLMs are often pretrained on such a broad sample of known protein sequence space that apparently held-out evaluation sequences may still be close homologs of proteins

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.

already present in the pretraining data. This creates a risk of *data leakage*: information from evaluation sequences can enter pretraining through identical or homologous sequences, inflating apparent downstream performance (Bernett et al., 2024; Hermann et al., 2024), as illustrated in Figure 1A. Proper evaluation therefore requires train/test separation not only for the downstream task, but also with respect to pLM pretraining itself. In practice, however, the computational cost of pretraining pLMs at scale makes such evaluations difficult, and most downstream studies rely on pretrained models whose training corpora already span much of known sequence space.

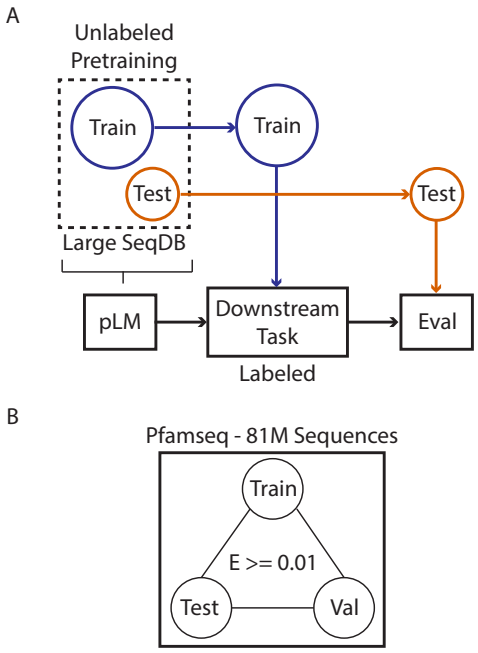

Figure 1. **Data leakage overview** A. Schematic showing pLMs pretrained on large databases used in a downstream task and inducing leakage despite having a train/test split B. Our controlled data split on Pfamseq allowing for non leaky pLM training where a pairwise sequence comparison E-value of less than 0.01 across splits is disallowed.

In this work, we study this problem through controlled experiments on pretraining leakage in downstream domain annotation. We construct large-scale train, validation, and test splits with no detectable cross-split homology (Figure 1B) and use them for both language-model pretraining and downstream evaluation. We then vary only the amount of test-set or test-related sequence overlap present in the pretraining data while keeping architecture, compute, and downstream evaluation fixed. This design lets us distinguish between two possibilities: pretraining on test sequences or related sequences may inflate downstream performance on the test set, or it may improve representations in a way that transfers to other still out-of-distribution sequences. We find that increasing leakage increases test-set domain annotation sensitivity, while training loss, validation perplexity, and validation-set downstream performance remain nearly unchanged. This pattern suggests that pretraining on test sequences or close homologs inflates performance on the test set without improving broader generalization. We also find that domain-relevant signal emerges early during masked language model pretraining.

2. Related work

Large-scale protein domain annotation has long relied on profile hidden Markov models (profile HMMs). In practice, annotation is performed by scanning sequences against libraries of curated domain models, each representing a homologous domain family. These models are typically built and searched with HMMER (Eddy, 2011), which underlies domain annotation in most of the protein family databases integrated by InterPro, a widely used resource that brings together domain models from multiple sources (Blum et al., 2025). More recent work has begun to explore deep-learning-based alternatives that replace large libraries of family-specific models with learned sequence representations (Bileschi et al., 2022; Dohan et al., 2021; Shaw et al., 2024). The most directly relevant example is PSALM, which combines a pretrained protein language model with a per-residue classifier and structured decoder to identify domains and reports domain-detection performance competitive with HMMER (Sarkar et al., 2026). This line of work suggests that pLMs can support domain annotation directly, rather than only sequence-level representation learning or downstream property prediction, but it does not by itself resolve how such methods behave when evaluation proteins are evolutionarily distant from the data available during pretraining.

A related literature has focused on benchmark construction and appropriate splits for biological sequence modeling. Work on independent-set benchmark construction showed that random partitioning is not enough for homologous biological sequences and introduced procedures that enforce explicit dissimilarity constraints between training and test sets (Petti & Eddy, 2022). Related methods such as GraphPart (Teufel et al., 2023) and tools such as sledgeSeq (Krishnan et al., 2026) likewise aim to construct sequence dataset partitions enforcing dissimilarity thresholds between splits, rather than relying on random partitioning or ordinary clustering alone. In parallel, benchmark efforts such as TAPE (Rao et al., 2019) and FLIP (Dallago et al., 2021) emphasized curated downstream-task splits designed to probe more realistic forms of biological generalization than random partitioning. In practice, many such controlled splits are built with clustering or similarity-thresholding pipelines, often using tools such as MMseqs2 (Steinegger & Söding, 2017). These studies established that random splits can substan-

tially overstate downstream performance, but they generally operate at the level of the supervised benchmark itself. They do not control similarity between a downstream test set and the much larger corpus used for pLM pretraining.

More recent work has addressed data leakage from pretraining more directly. In pLM-based thermostability prediction, train/test splits constructed to account for similarity to the pretraining data were shown to materially change measured performance relative to downstream task-only split strategies, demonstrating that leakage from pLM pretraining can inflate estimates of generalization (Hermann et al., 2024). Related work on protein-protein interaction prediction likewise found that pretrained pLMs can introduce leakage but studied this effect by training and comparing small, efficient pLMs on strict splits and splits with leakage rather than in a setting matched to the scale of modern pLM pretraining (Szymborski & Emad, 2026). Similar concerns have also been reported for genomic language models, trained on DNA sequences, where homology-based leakage has been shown to inflate performance in gene expression prediction tasks (Rafi et al., 2025). Together, these studies make clear that leakage is a widespread problem in biological sequence modeling, but they do not yet characterize its effects on downstream performance in a controlled setting at a scale comparable to modern pLM pretraining.

3. Methods

3.1. Protein language model pretraining

3.1.1. Data curation

We use sledgeSeq (Krishnan et al., 2026) to partition the 81M-sequence Pfamseq database (Paysan-Lafosse et al., 2025) into train, validation, and test sets with no statistically significant homology between sequences across the splits. To construct these splits, sledgeSeq sequentially applies pHMMER (Eddy, 2011), MMseqs2 (Steinegger & Söding, 2017), BLASTp (Altschul et al., 1997), and accelerated Smith-Waterman alignment (Pearson & Lipman, 1988; Smith et al., 1981) to detect homology between sequences across splits and remove sequences involved in statistically significant cross-split matches. We treat homology with an E-value of 0.01 or lower as significant, where the E-value denotes the expected number of false positives for 1 query sequence searched against an 81M sequence database. Under this criterion, the train, validation, and test sets are out of distribution with respect to one another. Because the validation split is constructed by the same homology-filtering procedure and remains non-homologous to both train and test, it serves as a negative control in our controlled leakage experiments.

We further cluster the training set at 90% sequence identity with MMseqs2 to remove closely related sequences and construct the Pfamseq90 training set. The resulting splits contain 24M sequences in train, 17K in validation, and 97K in test. In all data-leakage experiments, we use these same splits for both language-model pretraining and downstream-task training.

To compare against ESM-2, we pretrain a protein language model on UniRef50 (50M protein sequences) (Suzek et al., 2015) to match the ESM-2 training setup, while retaining the Pfamseq90 train, validation, and test sets for downstream training and evaluation. The various datasets are characterized in Table 1 showing the number of sequences, total residues, domains and unique families (defined by the Pfam database (Paysan-Lafosse et al., 2025), see section 3.2.1).

Table 1. Dataset characterization

Attribute	Train	Validation	Test	UniRef50
Sequences	24M	17K	97K	50M
Residues	6.6B	7M	17M	14.1B
Total domains	15.4M	26K	41.3K	37.6M
Unique families	22.3K	5K	8K	24K

3.1.2. Pretraining

We train all protein language models with the masked language modeling objective (Devlin et al., 2019) and use the 650M-parameter ESM-2 architecture (Lin et al., 2023). We crop sequences to a maximum length of 1024 residues and train with an effective batch size of $\sim$1.1M tokens using Adam with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.95, 10^{-8})$, gradient clipping at 0.5, and weight decay of 0.01. Inspired by the compute-optimal pLM training recipe given by Cheng et al. (2024), we mask 15% of tokens and use a cosine learning-rate schedule over the first 100K steps with 2,500 warmup steps, reaching a peak learning rate of 4×10^{-4} and then decaying to 4×10^{-5} by step 100K. We hold the learning rate constant thereafter.

The training recommendations in Cheng et al. (2024) are designed for approximately one epoch and the authors note diminishing returns, along with a greater risk of overfitting, with longer training. In our setting, one epoch corresponds to roughly 15K steps on UniRef50 and 8K steps on Pfamseq90 train. We train for multiple epochs to examine how downstream task performance changes with additional pretraining, and monitor validation perplexity throughout to guard against overfitting.

3.2. Downstream task

We use domain annotation as the downstream task for evaluating pLMs. We focus on this task for two reasons:

1. We hypothesize that protein domains, which are often identified through patterns or motifs of conserved (sub)sequences, are among the earliest biologically

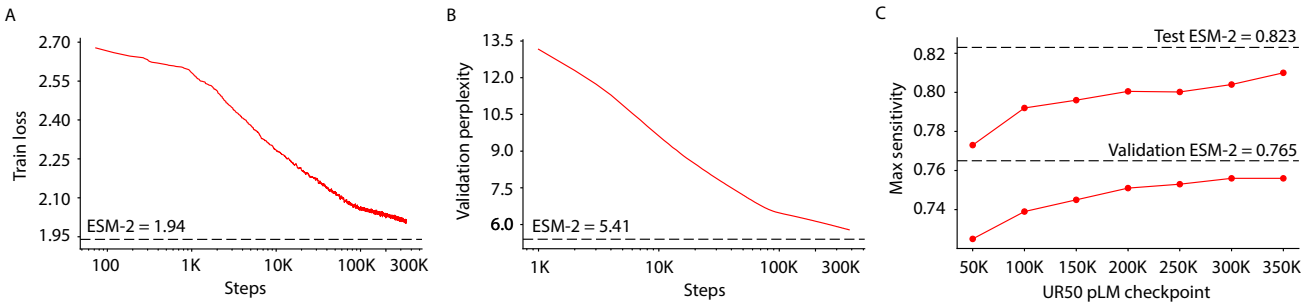

Figure 2. **Compute-limited training recovers ESM-2 performance**. A. Training loss for UR50 model with ESM-2 training loss as dashed line B. Validation perplexity against steps with ESM-2 perplexity in dashed line C. Max sensitivity for downstream domain annotation on test and validation set for UR50 model with ESM-2 values in dashed line

meaningful features learned during masked language model pretraining.

2. Domain annotation depends directly on sequence homology, so constructing train, validation, and test splits through homology-based sequence filtering provides a natural way to assess generalization for this task.

3.2.1. Ground truth labels

We annotate the Pfamseq90 train, validation, and test sets using Pfam 37.2 (Paysan-Lafosse et al., 2025), a manually curated collection of 24,076 protein domain families, each represented by a profile hidden Markov model. For each sequence in these splits, we define the ground-truth domains as the Pfam hits reported by HMMER 3.4 (Eddy, 2011) when run with Pfam's curated family-specific gathering thresholds (`--cut_ga`). These calls correspond to Pfam's curator-specified, high-confidence domain annotations. Additional true domains may still be present in a sequence but fall below these thresholds and remain unannotated in the groundtruth. Throughout this work, we evaluate sensitivity only with respect to this ground-truth set.

The test split from our Pfamseq partition contains 97K sequences, but many belong to groups of closely related proteins. To reduce redundancy and limit over-representation of highly similar examples, we further cluster the test set with sledgeSeq at 50% sequence identity and retain one representative sequence per cluster. This yields a non-redundant test set of 57K sequences, which we use for all reported sensitivity measurements.

3.2.2. Training

We use the PSALM framework for downstream domain annotation, training a multilayer perceptron (MLP) together with a structured probabilistic decoder on top of frozen pLM embeddings (Sarkar et al., 2026). For downstream training, we augment the Pfamseq90 training set as described in PSALM: each sequence contributes either an original or shuffled-outside example with equal probability, 10% of examples are domain slices, and fully shuffled negatives are added to increase the amount of non-domain background. This augmentation increases training diversity and exposes the classifier to both realistic domain context and non-domain background.

We train the domain annotation MLP for a single epoch ($\sim$140K steps) with a learning rate of 5×10^{-4} using Adam with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.999, 10^{-8})$, gradient clipping at 1.0, and weight decay 0.01.

In this work we depart from the full PSALM training procedure by restricting training to its first phase, where the pLM is frozen and only the classifier head is optimized. We omit the second phase, which fine-tunes the pLM to improve specificity, so that differences in downstream performance can be attributed to pretrained embeddings rather than differences introduced during downstream fine-tuning. We therefore assess pLM quality through the maximum sensitivity achieved in this PSALM-style domain annotation task.

3.2.3. Statistical analysis

To compare domain annotation sensitivity between models, we use nonparametric bootstrap resampling over sequences. For each model and evaluation split, we sample sequences with replacement and recompute sensitivity over 5,000 bootstrap replicates. For each pairwise comparison, we compute a two-sided bootstrap p-value from the distribution of sensitivity differences and report significance using the standard notation (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$).

3.3. Compute infrastructure

All experiments were run on a single 4 x NVIDIA H200 141GB node with 64 available CPU cores. Language model training for 200K steps completed in approximately 21 days and PSALM training completed in approximately 12 hours.

4. Results

We first show that our reduced-compute pretraining setup recovers most of the downstream performance of ESM-2, making it a suitable foundation for the controlled leakage experiments that follow. We then show that increasing pretraining leakage increases test-set domain annotation sensitivity, while leaving validation-set performance essentially unchanged. Finally, we compare our controlled exact-sequence leakage setting to the broader homologous leakage present in realistic pretraining corpora.

4.1. Compute-limited training largely recovers ESM-2 performance

As a positive-control experiment, we first test whether our compute-efficient training setup, inspired by Cheng et al. (2024) (see section 3.1.2), can recover the performance of ESM-2 in a lower-compute regime. To do this, we train a pLM, which we refer to as UR50, on UniRef50 using the ESM-2 architecture and the training procedure described in Methods. ESM-2 was trained for roughly 1M steps at our effective batch size of ∼1.1M tokens, whereas UR50 is trained for 350K steps.

Since UR50 matches ESM-2 in both architecture and pretraining dataset, the main remaining sources of variability are differences in optimization settings and sequence sampling during training. By tracking training loss, validation perplexity, and downstream task performance as compute increases, we assess how closely UR50 approaches ESM-2 under this reduced-compute setup. This experiment establishes that our reduced-compute training setup is strong enough to recover most of ESM-2 performance, making it a reasonable foundation for the controlled leakage comparisons that follow.

Training dynamics. Figure 2A shows the training loss as a function of optimization steps. We report the target ESM-2 loss evaluated on a randomly-selected held-out shard of our dataset (approximately 100 batches). The loss curve steadily decreases and approaches the ESM-2 reference loss computed on a subset of Pfamseq90 train (∼100K sequences).

Validation perplexity. In Figure 2B, we track the validation perplexity on a held-out dataset (see section 3.1.1). Similar to the training loss, validation perplexity decreases consistently and approaches the ESM-2 reference value of 5.41.

Downstream domain annotation. We next evaluate the quality of learned representations from the language model on a domain annotation task. Protein language models are known to encode sequence homology information that can be used for domain annotation (Sarkar et al., 2026).

Figure 2C reports the maximum sensitivity of domain detection across checkpoints during UR50 training. Sensitivity increases with longer pretraining and approaches the ESM-2 baseline of 0.82. Despite the reduced-compute regime, UR50 at 200K steps recovers most of ESM-2 performance in both validation perplexity and downstream domain annotation. Even after 50K steps, UR50 reaches 77% domain-detection sensitivity, indicating that substantial domain-relevant signal is already present well before the model reaches its best overall performance. This supports our hypothesis that protein domains are learned early during masked language model pretraining and supports our use of domain annotation as a probe of representations learned during pretraining. We therefore use 200K steps as the training budget in the controlled leakage experiments, while training UR50 to 350K steps only to characterize its approach to ESM-2 performance.

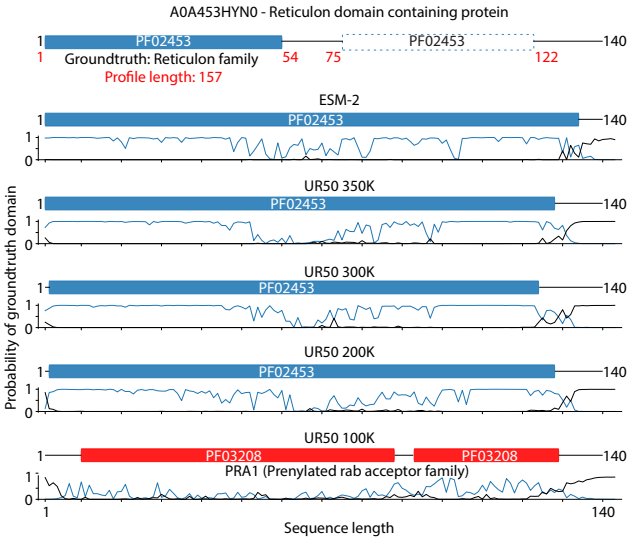

Figure 3. **Representative domain annotation across UR50 checkpoints.** Colored bars show the decoded domain calls for each model, with the ground-truth Pfam annotation shown at top. Red numbers indicate the profile coordinates of the Pfam hits, suggesting that the annotated region covers only part of the full profile. Blue curves show the per-residue probabilities assigned by the PSALM MLP to the correct family, PF02453. Black curves show the corresponding per-residue probabilities assigned to the non-domain state.

Domain annotation case study across varying compute. Figure 3 shows the reticulon-domain-containing protein A0A453HYN0, selected from the test set as a representative example of how downstream domain annotation improves with additional pretraining. The 100K UR50 checkpoint fails to recover the ground-truth reticulon family and instead predicts an incorrect domain, whereas UR50 at 200K steps and beyond recovers the same family-level call as ESM-2. These later calls extend beyond the ground-truth annotation, but the Pfam profile coordinates indicate that

the annotated hit may cover only part of the full domain. Consistent with this, HMMER run in maximum-sensitivity (`--max`) mode identifies an additional hit in the C-terminal half of the sequence, suggesting that the longer UR50 and ESM-2 calls may reflect more complete recovery of the underlying domain signal rather than simple overextension. The residue-level probabilities support this interpretation: from 200K to 350K steps, UR50 assigns increasingly strong probability to the regions supported by the HMMER evidence, and ESM-2 shows the clearest support across the full span. One possible explanation is that this sequence contains substantial insertions relative to the Pfam profile, which may make the full domain more difficult to recover with profile-based matching alone.

4.2. Pretraining leakage inflates domain annotation sensitivity

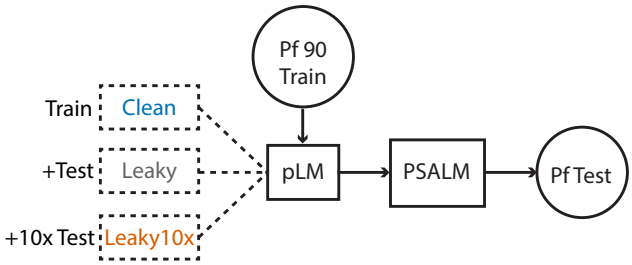

Figure 4. **Workflow for controlled leakage experiments**

To isolate the effect of pretraining leakage on domain annotation, we train three compute-matched pLMs for 200K steps each that differ only in the amount of test-set leakage present during pretraining, as shown in Figure 4:

(i) Clean: trained on Pfamseq90 train only.
(ii) Leaky: trained on Pfamseq90 train + Pfamseq test.
(iii) Leaky 10x: trained on Pfamseq90 train + 10 copies of Pfamseq test.

Because the test set contains only 97K sequences, or 0.4% of the size of Pfamseq90 train, a single copy of the test set may provide too weak a signal to produce a clear downstream effect. We therefore include the leaky 10x condition to increase the amount of leakage while keeping the downstream task fixed. In all cases, the downstream PSALM model is trained only on Pfamseq90 train and evaluated on both the test set and the validation set. Since the validation set remains out of distribution with respect to pretraining for all three models, it serves as a negative control for leakage-specific effects: if leakage improves performance only through contamination of the train set with test set sequences, the gain should not transfer to validation. Because our controlled manipulation introduces leakage only through exact test-set sequences, it provides a conservative estimate of how much performance inflation can arise in realistic pretraining corpora that contain additional homologous sequences beyond exact test set matches.

Validation set domain sensitivity as a negative control. Figures 5A and B show that all three pLMs have nearly identical training loss and validation perplexity. We also find no meaningful difference in downstream domain annotation sensitivity on the validation set (Figure 5C). Since the validation set is non-homologous to the pretraining data for all three models, this lack of change is expected if leakage acts specifically through contamination of the train set with test set sequences. Conversely, if leakage were improving representation quality in a more general way, we would expect leaky and leaky 10x to improve on the validation set as well. We do not observe such an improvement.

Leakage increases test-set sensitivity. In contrast, sensitivity on the test set increases clearly with the amount of leakage present during pretraining (Figure 5C). Maximum sensitivity increases by 2.5% from clean to leaky and by 5% from clean to leaky 10x; bootstrap resampling confirms that these differences are statistically significant (see section 3.2.3). The fact that these gains appear only on the test set but not on the non-homologous validation set indicates that pretraining leakage inflates test-set performance rather than improving broader out-of-distribution generalization.

Exact-sequence leakage vs. homologous leakage. Test-set domain annotation sensitivity increases with the amount of leakage present in the pLM pretraining data. In our controlled leakage experiments, this leakage consists only of exact test-set sequences. By construction, the clean model is trained on sequences with no detectable homology to the test set. The leaky model includes one copy of the 97K-sequence test set during pretraining, whereas leaky 10x includes ten copies of that same test set, for a total of 970K leaky sequences. Thus, the difference between leaky and leaky 10x is not greater diversity of leaked sequences, but greater replication of the same exact test-set contamination.

UniRef50 is leakier in a broader sense. In addition to containing exact test-set sequences, it also contains many homologous sequences related to the test set. Using sledgeSeq, we estimate that UniRef50 contains approximately 11M sequences with significant similarity to the test set, compared with 970K exact leaked sequences in the leaky 10x setting. We hypothesize that this larger and more diverse source of leakage helps explain the higher test-set sensitivity of ESM-2 and UR50 (even in a compute matched setting as shown in Figure 5C) relative to the clean, leaky, and leaky 10x models, because homologous sequences may partially bridge the gap between the pretraining and test data distributions. The difference in domain sensitivity observed between the clean, leaky, and leaky 10x models therefore provides a lower bound on the performance inflation that leakage can induce.

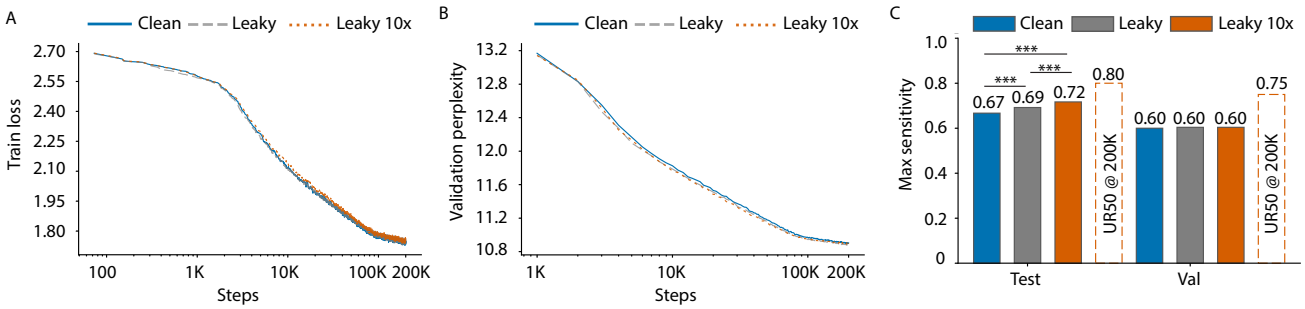

Figure 5. **Effect of pLM data leakage on downstream performance.** A. Train loss for clean, leaky and leaky 10x models B. Validation perplexity for clean, leaky and leaky 10x models C. Max sensitivity for downstream domain annotation of each model on test and validation set along with corresponding compute matched UR50 model sensitivity shown as a dashed bar

Because realistic pLM pretraining corpora contain both exact and homologous leakage, their inflation may be larger than what we measure in this controlled exact-sequence setting.

Domain annotation case study across varying leakage. We selected the protein kinase domain-containing protein A0A150GE86 from the test set to illustrate how data leakage in pretraining affects domain annotation (Figure 6). The ground-truth annotation contains a protein kinase family domain (PF07714). The compute-matched UR50 model at 200K steps identifies this family correctly across nearly the full annotated region, with high per-residue probability assigned to the ground-truth family and low probability assigned to the non-domain state. The clean model fails to recover the correct family and instead predicts two incorrect domains. The leaky model recovers a kinase-family prediction, but assigns the sequence to the wrong Pfam family within the same clan, where a clan is a curated group of homologous Pfam domain families (Finn et al., 2006). In contrast, the leaky 10x model recovers the correct family-level annotation, closely matching the UR50 prediction. This example is consistent with the aggregate results in Figure 5: increasing leakage improves test-set domain annotation, and in some cases repeated exposure to leaked sequences is sufficient to shift the model from an incorrect family prediction to the correct one.

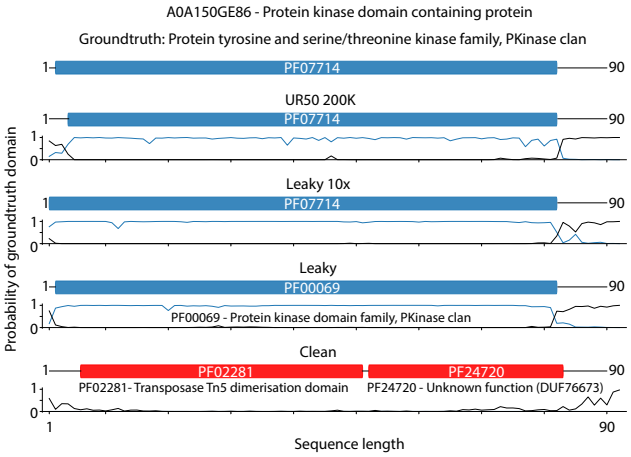

Figure 6. **Representative domain annotation across controlled leakage settings.** Predictions from PSALM for the protein kinase domain-containing protein A0A150GE86 using a compute-matched UR50 model at 200K steps and the clean, leaky, and leaky 10x pLMs. Colored bars show the decoded domain calls for each model, with the ground-truth Pfam annotation shown at top. Blue curves show the per-residue probabilities assigned directly by the PSALM MLP to the ground-truth family PF07714. Black curves show the corresponding per-residue probabilities assigned to the non-domain state.

5. Discussion

Our results show that pretraining leakage can materially inflate downstream domain annotation sensitivity, even when standard pretraining metrics remain nearly unchanged. In our controlled-leakage experiments, increasing the amount of leaked test-set data in pretraining improves performance on the test set, while leaving training loss and validation perplexity almost identical across models. This makes leakage difficult to detect from pretraining metrics alone and suggests that downstream evaluation can substantially overstate performance when pretraining corpora are not controlled.

The contrast between test and validation performance is especially important for interpretation. The validation set serves as a negative control in our experiments because it remains out-of-distribution (non homologous) to the pretraining data for all models. If leakage were improving representation quality in a way that helped generalization, we would expect those gains to transfer to validation performance as well. Instead, validation-set downstream performance remains essentially unchanged. This suggests that the improvement in test-set performance arises from leakage in pretraining rather than from improved out-of-distribution generalization.

Our results also support the use of domain annotation as a probe of what pLMs learn during pretraining. Even in the

compute-limited UR50 setting, domain-detection sensitivity rises quickly, and useful domain-level signal is already detectable at a small fraction of the compute required for ESM-2. This is consistent with our hypothesis that protein domains are learned early in masked language model pretraining. At the same time, the leakage experiments show that strong downstream domain-annotation performance is not by itself evidence of remote-homology generalization, since some of that performance can be explained by contamination of the pretraining data with test set or test-related sequences.

The comparison between our controlled leakage settings and UniRef50 also suggests that exact-sequence leakage is only one part of the problem. In the clean, leaky, and leaky 10x models, leakage is limited to repeated copies of exact test-set sequences, whereas UniRef50 contains both exact test sequences and many homologous sequences related to the test set. We hypothesize that this broader and more diverse form of leakage contributes to the stronger performance of ESM-2 and UR50 pLMs by partially bridging the gap between pretraining and test distributions. At the same time, our study focuses on downstream domain annotation and on a controlled exact-sequence leakage setting, so the magnitude and form of leakage effects may differ for other downstream tasks, as suggested by Szymborski & Emad (2026), and for more realistic mixtures of homologous overlap. More generally, these results suggest that leakage in biological sequence modeling should be understood not only in terms of exact duplicates, but also in terms of homologous sequence relationships that can blur the boundary between interpolation and true generalization.

Future work. Two directions for future work seem especially important. First, it will be useful to study how leakage effects vary across downstream tasks, especially tasks that depend less directly on sequence homology than domain annotation. Second, a more detailed characterization of homologous leakage across sequence-similarity ranges could clarify when pLM performance reflects transfer to genuinely novel proteins and when it reflects partial overlap with pretraining data. It will also be important to test whether the magnitude of these effects changes with larger-scale pretraining. More broadly, leakage-controlled evaluation that accounts for the pretraining stage may be necessary for making reliable claims about biological generalization in large sequence models.

Code and Data Availability

All code and data will be made public after acceptance to the workshop.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

Bernett, J., Blumenthal, D. B., Grimm, D. G., Haselbeck, F., Joeres, R., Kalinina, O. V., and List, M. Guiding questions to avoid data leakage in biological machine learning applications. *Nature Methods*, 21(8):1444–1453, 2024.

Bileschi, M. L., Belanger, D., Bryant, D. H., Sanderson, T., Carter, B., Sculley, D., Bateman, A., DePristo, M. A., and Colwell, L. J. Using deep learning to annotate the protein universe. *Nature Biotechnology*, 40:932–937, 2022.

Blum, M., Andreeva, A., Florentino, L. C., Chuguransky, S. R., Grego, T., Hobbs, E., Pinto, B. L., Orr, A., Paysan-Lafosse, T., Ponamareva, I., et al. InterPro: the protein sequence classification resource in 2025. *Nucleic Acids Research*, 53:D444–D456, 2025.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8): 2102–2110, 2022.

Cheng, X., Chen, B., Li, P., Gong, J., Tang, J., and Song, L. Training compute-optimal protein language models. *Advances in Neural Information Processing Systems*, 37: 69386–69418, 2024.

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. FLIP: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pp. 2021–11, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Dohan, D., Gane, A., Bileschi, M. L., Belanger, D., and Colwell, L. Improving protein function annotation via unsupervised pre-training: Robustness, efficiency, and

insights. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2782–2791, 2021.

Eddy, S. R. Accelerated Profile HMM Searches. *PLOS Computational Biology*, 7:1–16, 2011. doi: 10.1371/journal.pcbi.1002195.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. ProtTrans: toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Fallahpour, A., Seyed-Ahmadi, A., Idehpour, P., Ibrahim, O., Gupta, P., Naimer, J., Zhu, K., Shah, A., Ma, S., Adduri, A., et al. BioReason-Pro: advancing protein function prediction with multimodal biological reasoning. *bioRxiv*, pp. 2026–03, 2026.

Finn, R. D., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., et al. Pfam: clans, web tools and services. *Nucleic Acids Research*, 34(suppl_1):D247–D251, 2006.

Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

Hermann, L., Fiedler, T., Nguyen, H. A., Nowicka, M., and Bartoszewicz, J. M. Beware of data leakage from protein LLM pretraining. *bioRxiv*, pp. 2024–07, 2024.

Huang, K., Zhang, S., Wang, H., Qu, Y., Lu, Y., Roohani, Y., Li, R., Qiu, L., Li, G., Zhang, J., et al. Biomni: A general-purpose biomedical ai agent. *bioRxiv*, 2025.

Krishnan, K., Sarkar, A., and Eddy, S. R. sledgeseq: partitioning protein sequence databases for llm training. github.com/sledgeseq/sledge, 2026. GitHub repository, accessed 2026-04-17.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T., and Rost, B. Embeddings from deep learning transfer GO annotations beyond homology. *Scientific Reports*, 11(1): 1160, 2021.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

Paysan-Lafosse, T., Andreeva, A., Blum, M., Chuguransky, S. R., Grego, T., Pinto, B. L., Salazar, G. A., Bileschi, M. L., Llinares-López, F., Meng-Papaxanthos, L., et al. The Pfam protein families database: embracing AI/ML. *Nucleic Acids Research*, 53(D1):D523–D534, 2025.

Pearson, W. R. and Lipman, D. J. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988. doi: 10.1073/pnas.85.8.2444.

Petti, S. and Eddy, S. R. Constructing benchmark test sets for biological sequence analysis using independent set algorithms. *PLOS Computational Biology*, 18(3):e1009492, 2022.

Rafi, A. M., Kiyota, B., Yachie, N., and de Boer, C. Detecting and avoiding homology-based data leakage in genome-trained sequence models. *bioRxiv*, pp. 2025–01, 2025.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating protein transfer learning with TAPE. *Advances in neural information processing systems*, 32, 2019.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. 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.

Sarkar, A., Krishnan, K., and Eddy, S. R. Protein sequence domain annotation using a language model. *bioRxiv*, pp. 2026–04, 2026.

Shaw, P., Gurram, B., Belanger, D., Gane, A., Bileschi, M. L., Colwell, L. J., Toutanova, K., and Parikh, A. P. ProtEx: a retrieval-augmented approach for protein function prediction. *bioRxiv*, 2024. doi: https://doi.org/10.1101/2024.05.30.596539.

Smith, T. F., Waterman, M. S., et al. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

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

Su, J., He, Y., You, S., Jiang, S., Zhou, X., Zhang, X., Wang, Y., Su, X., Tolstoy, I., Chang, X., et al. A trimodal protein language model enables advanced protein searches. *Nature Biotechnology*, pp. 1–7, 2025.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and UniProt Consortium, t. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

Szymborski, J. and Emad, A. A flaw in using pretrained protein language models in protein–protein interaction inference models. *Nature Machine Intelligence*, pp. 1–12, 2026.

Teufel, F., Gíslason, M. H., Almagro Armenteros, J. J., Johansen, A. R., Winther, O., and Nielsen, H. GraphPart: homology partitioning for biological sequence analysis. *NAR Genomics and Bioinformatics*, 5(4):lqad088, 2023.