A Domain-Token Transformer for Protein Representation Learning

PCXQ3[1]

MSc Data Science and Machine Learning

Dr Daniel Buchan

Submission date: 7 September 2025

Abstract

Today protein LLMs are widely used to solve tasks like structure and function prediction. With their utility depending on the quality of the learned embedding space. To build such spaces, transformer-based masked-token models are typically trained on amino-acid tokenisations. This study asks whether biological structure could be similarly captured if tokens are protein domains instead. We train an encoder on domain-level sequences and obtain an embedding space covering 77 million proteins. The resulting space shows some organisation: embeddings cluster by similarity, embedding proximity correlates with multi-domain architecture order, and neighbours display Gene Ontology consistency, indicating that the model captures both domain composition and ordering. Despite limits in training scale and embedding dimensionality, these results provide proof of concept that domain-based tokenisation is biologically informative. We conclude that protein LLMs trained using domain tokens can complement residue-level models and may support downstream applications such as protein function prediction and design.

Acknowledgements

Many thanks to my supervisor Dr Daniel Buchan for his direction and advice throughout the dissertation. I also wish to express my gratitude to the government of my country for funding my MSc degree through the "State Scholarship Program on Education of Youth Abroad".

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

By encoding huge numbers of protein sequences, protein language models uncover links between sequence, shape, and function. This lets us predict structures, as in AlphaFold achieving near-experimental accuracy at scale [Jumper et al., 2021], design proteins, as shown by ProtGPT2 generating novel functional proteins [Ferruz et al., 2022], and judge the impact of variants, for example through ESM-1v predicting functional effects of mutations [Meier et al., 2021]. In turn, that speeds up medicine, such as the use of protein language models for enzyme and drug target design [Hie et al., 2022], clarifies genetic diagnoses, for instance by enabling rare disease variant interpretation [Frazer et al., 2021], contributes to outbreak response, as shown by early SARS-CoV-2 protein structure predictions [Callaway et al., 2020], and enables cleaner biotech, for example through machine learning–guided enzyme engineering for plastic biodegradation [Lu et al., 2022].

Most current protein language models use attention-based Transformers, with BERT-style masked encoders now the mainstream, as seen in TAPE [Rao et al., 2019] and ProtBERT [Elnaggar et al., 2022]. These models learn general-purpose sequence representations that transfer across tasks, for example enabling secondary structure and contact prediction [Rao et al., 2019]. Alignment-aware variants add evolutionary signal, such as MSA Transformer using multiple sequence alignments [Rao et al., 2021], and generative models focus on design, as in ProtGPT2 generating novel functional proteins [Ferruz et al., 2022]. Large pretraining efforts show clear gains from scale, demonstrated by ESM-1b/ESM-2 trained on hundreds of millions of sequences [Rives et al., 2021, Lin et al., 2023]. Across these approaches, inputs are still usually represented at the amino-acid level. However, an alternative that we explore in this study is to move beyond individual residues and instead treat protein domains as fundamental tokens, capturing higher-order structural and functional units.

A protein structural domain is a compact, semi-independently folding region of a protein that often carries a distinct structural or functional role. In contrast, sequence domains are identified by conserved sequence motifs, which may or may not correspond directly to structural domains.

Unlike the models that use amino-acids as units, we could instead tokenise protein domains. If we view proteins as sentences in a language, then protein domains would play the role of words, while amino acids would be on the level of letters in the words. This analogy illustrates

our motivation for using domains as tokens. With this approach the number of tokens for a given protein significantly reduces, and approximates the number of tokens to the number of words in an average sentence in English. Thus, the adoption of language model architectures such as BERT, which were originally designed for natural languages, becomes more appropriate for modelling protein sequences [Devlin et al., 2019].

In this project we trained a model using a dataset of 77 million protein sequences by using protein domains as tokens and adopting the BERT architecture. BERT is a neural network that masks certain words in the text and then learns to fill in what is missing, so it can understand context more deeply [Devlin et al., 2019]. In our case, the model learns to understand the relationships between different domains by masking and guessing the protein domains instead. After training, the model understands the "language" of protein - it knows which domains commonly appear together, in what order, and what patterns exist. This knowledge gets stored in the 256-dimensional embeddings that represent each protein sequence.

We further evaluate those embeddings to validate whether domain-based tokenisation yields a model that is feasible and that represents some useful biological meaning. In particular, we test whether the embeddings capture functional relationships between proteins with shared domains. More broadly, we examine whether the learned representations reflect meaningful biological signal that could support downstream applications such as protein function prediction or guiding protein design.

1.2 Overview

Below is an outline of this paper. In Chapter 2 we provide the necessary background. We begin by outlining the modular nature of proteins, including domains, linkers, and intrinsically disordered regions. We then present the principles of Transformer encoders and masked language modelling, emphasising how tokenisation shapes both the learning signal and computational efficiency. We review existing language models such as BERT and protein language models such as ProteinBERT and TCR-BERT, situating our approach in relation to them.

Chapter 3 describes the data sources and methodology. We first detail the resources used to construct the training corpus, including CATH, AlphaFold DB, UniProt, Pfam, InterPro, and TED. We then present the processing pipeline and the construction of domain-based and linker-based vocabularies. The methodology section specifies the featurisation, tokenisation, and masking strategies used in pretraining, before describing the model architecture, pretraining loop, and quality checks.

In Chapter 4 we outline our evaluation strategy and present results. We begin with unsupervised analyses such as retrieval, clustering, and community detection. We then explore whether the learned embeddings capture biologically meaningful signal, including architecture-level regularities and functional relationships, alongside GO consistency test and summarise the outcomes of these experiments.

Chapter 5 is devoted to discussion. We reflecting on limitations that were noted throughout our work. We consider the implications for downstream tasks such as function prediction, and discuss potential steps that are required for the practical implementation.

Finally, in Chapter 6 we conclude the report. We summarise the main contributions of the

work and suggest directions for future research, including further evaluation frameworks and the integration of domain-token embeddings into broader biological applications.

Chapter 2

Background

2.1 Protein domains and structural elements

2.1.1 Proteins as modular sequences

Proteins are linear polymers that are formed from 20-letter amino-acid alphabet. The chain folds via local interactions such as hydrogen bonding and side-chain packing into secondary-structure motifs (α-helices, β-strands), which pack into tertiary and, in many cases, quaternary structures. Primary structure is simply the amino-acid sequence (N→C order), which fixes backbone directionality and side-chain chemistry.

Secondary structure consists mainly of α-helices and β-sheets formed by regular backbone hydrogen bonds. An alpha-helix is a tight spiral where each backbone C=O makes a hydrogen bond to an N–H four residues ahead, forming a stable coil with side chains pointing outward. A beta-sheet is built from stretched strands lying side by side and joined by hydrogen bonds; the strands can run parallel or anti-parallel, and their side chains alternate above and below the sheet.

Tertiary structure is the three-dimensional fold of one polypeptide chain-largely set by its sequence and stabilised by the hydrophobic effect, hydrogen bonds, ionic interactions and tight packing-whereas quaternary structure is the assembly of several chains into a single functional complex.

There is a shared vocabulary - Gene Ontology (GO) that is widely used to describe what proteins do where they act, and which broader processes they contribute to. It has three parts: Molecular Function (the specific activity a protein can perform, such as “ATP binding” or “DNA helicase”), Biological Process which models Molecular Biological Processes (the larger goal or pathway the protein helps achieve, such as “DNA replication” or “signal transduction”), and Cellular Component (the place in the cell where the protein works, such as “nucleus” or “ribosome”). Each GO term has a stable identifier and is linked to related terms in a directed acyclic graph, so annotations can be as general or as specific as the available evidence allows. [Ashburner et al., 2000]

2.1.2 Protein domains

Beyond the structural hierarchy, many proteins are modular. Most of them are built from one or more compact substructures - domains. This compact, evolutionarily conserved unit tends

to fold independently and recur in different protein. Typically protein domains are associated with a discrete biochemical or regulatory role. For example, the protein kinase catalytic domain, which catalyses ATP-dependent phosphorylation of substrates, or the SH2 domain, which mediates phosphotyrosine-dependent protein-protein binding.

In practice, domains are defined and maintained by community resources. While **Pfam** and **InterPro** provide definitions of sequence domains, **CATH** and **SCOP** focus on structural domains.

Pfam defines conserved domain families from curated seed alignments using profile HMMs and applies family-specific gathering thresholds, grouping related families into higher-level clans. This yields stable identifiers and boundaries that make domain calls reproducible across proteomes [Mistry et al., 2021].

InterPro integrates Pfam with other signature databases into non-redundant entries with stable accessions, curated abstracts, hierarchical relationships and GO links, and collocates complementary features to contextualise domain organisation [Paysan-Lafosse et al., 2022].

SCOP (Structural Classification of Proteins) also provides a hierarchical classification of protein structural domains, organised into Classes, Folds, Superfamilies, and Families, with emphasis on evolutionary relationships inferred from structural similarities. SCOP is highly curated and offers fine-grained annotations of protein fold space, but it is less frequently updated than CATH and relies heavily on expert manual curation [Andreeva et al., 2020].

CATH defines domains based on experimentally determined 3D structures, decomposing proteins into compact, evolutionarily conserved units of independent folding. It classifies these structural domains hierarchically into Class (C), Architecture (A), Topology (T), and Homologous superfamily (H), thereby capturing both broad structural similarities and detailed evolutionary relationships. This allows consistent mapping of structural domains across proteins, which is why most of the TED (The Encyclopedia of Domains) data we worked with comes in via CATH [Sillitoe et al., 2021].

2.1.3 Intrinsic disorder and linker segments

Domains in multi-domain proteins are commonly separated by linker segments, many of which are intrinsically disordered regions (IDRs) that do not adopt a single stable 3D structure under physiological conditions. IDRs are common and functional: they can carry short linear motifs that act as docking or switch sites, can fold upon binding to partners, and support context-dependent and often partner-specific recognition. Their length, flexibility, and composition influence how neighbouring domains communicate by setting effective distance and by exposing sites for post-translational regulation [Wright et al., 2015, Robin van der Lee et al., 2014].

Curation resources treat disorder alongside folded domains. Pfam provides type labels (e.g., domain, repeat, disordered) and curated boundaries for conserved regions, while InterPro integrates these calls and collocates disorder tracks (e.g., MobiDB-lite) with domain annotations so that folded modules and flexible linkers can be viewed on the same coordinate system [Mistry et al., 2021, Paysan-Lafosse et al., 2022].

In this thesis we therefore model both structured domains and IDR (linkers) as first-class elements. Domains supply modular, reusable units grounded in Pfam/InterPro and CATH hi-

erarchies, and IDRs capture the flexible, regulation-rich connectors that organise those units [Wright et al., 2015, Robin van der Lee et al., 2014]. When these elements are laid out along a sequence, we will refer to the resulting ordered list of modules as the protein's multi-domain architecture (MDA) [Yu et al., 2019, Geer et al., 2002].

2.2 Large language models (LLMs) relevant to sequence analysis

2.2.1 Transformer encoders

Classical sequence models (RNNs/LSTMs) process tokens step by step, which makes it hard to parallelise training and to capture long-range dependencies without very deep recurrence. The Transformer replaces recurrence with self-attention: at each layer, every token can directly "look at" every other token to decide how much they matter for its updated representation. Because these lookups are done with matrix multiplications, the whole sequence can be processed in parallel on modern hardware, which is one reason Transformers trained faster and reached higher accuracy than contemporary RNN/CNN models in the original work.

Encoder-only configuration A Transformer encoder is a stack of identical layers. Each layer has two sublayers: (i) multi-head self-attention, which mixes information across positions, and (ii) a position-wise feed-forward network (a two-layer MLP applied to each position independently). Each sublayer is wrapped with a residual connection and layer normalisation, so the output is $\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$. Stacking these blocks refines a contextual representation for every token using the entire sequence as context [Vaswani et al., 2017].

Self-attention Self-attention computes a weighted sum of value vectors V for each position by comparing a query Q (derived from that position) with keys K (derived from all positions). The comparison score is a dot product, scaled by $1/\sqrt{d_k}$ to keep gradients well-behaved when the key/query dimension d_k is large, followed by a softmax to obtain attention weights:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V.$$

This scaled dot-product attention maps to efficient matrix multiplications and lets every token condition on every other token within a single layer [Vaswani et al., 2017].

Multi-head Using a single attention map mixes all relationships into one set of weights. Multi-head attention instead linearly projects Q, K, V into h lower-dimensional subspaces, runs attention in parallel in each subspace, and then concatenates the results:

$$\mathrm{MHA}(X) = \mathrm{Concat}\big(\mathrm{head}^{(1)}, \ldots, \mathrm{head}^{(h)}\big)\, W_O.$$

Different heads can specialise (for example, local versus long-range relations). Because each head operates at reduced dimension, the total cost remains comparable to a single wide head, while

quality improves empirically [Vaswani et al., 2017].

Feed-forward sublayer (two-layer MLP) Between attention blocks, each position independently passes through the same MLP:

$$\mathrm{FFN}(x) = W_2\,\sigma(W_1 x + b_1) + b_2,$$

where $x \in \mathbb{R}^{d_{\text{model}}}$, $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$, $W_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$, and σ is a nonlinearity. The original Transformer used ReLU with $d_{\text{ff}} \approx 4\,d_{\text{model}}$; BERT keeps the same structure but typically uses GELU [Vaswani et al., 2017, Devlin et al., 2019]. This sublayer does not mix positions; it applies a nonlinear channel transformation that increases capacity between attention blocks.

Positional information Self-attention has no built-in notion of order, so the model adds positional encodings to token embeddings at the bottom of the stack. The original work used fixed sinusoidal encodings with multiple frequencies, while learned positional embeddings perform similarly in-distribution. The position vectors are summed with token embeddings and then processed by the encoder [Vaswani et al., 2017].

Complexity and why sequence length matters Self-attention gives a maximum path length of $O(1)$ between any two positions. The trade-off is computational: an attention map over L tokens has L^2 entries, so the per-layer cost scales as $O(L^2 d)$ with representation size d. By contrast, RNNs require $O(L)$ sequential steps, and CNNs need $O(\log_k L)$ layers to connect distant positions with kernel size k. This comparison motivates exploring shorter tokenisations (for example, domains versus residues) to reduce effective L while keeping biological meaning [Vaswani et al., 2017].

Regularisation and optimisation The reference Transformer applies dropout after sublayers and on token-plus-position sums, and uses label smoothing in the training loss; optimisation uses Adam with warm-up followed by inverse square-root decay [Vaswani et al., 2017]. For encoder-only pretraining in the BERT style, it is standard to use AdamW with warm-up and decay schedules and the masked-token objective; BERT also uses GELU in the FFN [Devlin et al., 2019].

What the encoder learns and why it suits representation learning By stacking attention and FFN layers, the encoder builds contextual embeddings: each token's representation reflects its entire context. Parallel processing enables efficient mini-batching, and increasing depth, width, and head count typically improves quality under sufficient data and compute [Vaswani et al., 2017]. In encoder-only mode, there is no auto-regressive decoder; each layer refines token embeddings with bidirectional context, which is ideal for masked-token pretraining and for producing general-purpose embeddings that lightweight heads can reuse across tasks [Devlin et al., 2019].

2.2.2 Masked language modelling (MLM)

Masked language modelling (MLM) pretrains an encoder by hiding a random subset of tokens and training the model to recover their original identities from bidirectional context. This produces contextual embeddings that transfer well across tasks and matches the conditioning style

of encoder-only Transformers [Devlin et al., 2019]. In protein sequence work, MLM at scale has been shown to yield representations with structure- and function-relevant signal that improve with more data and larger models [Rives et al., 2021, Lin et al., 2023].

Objective and notation Let $x = (x_1, \ldots, x_L)$ be a tokenised sequence and $M \subset \{1, \ldots, L\}$ the set of masked positions. Given a corrupted input $x_{\setminus M}$ after mask-time replacements described below, the loss is the sum of cross-entropies over masked positions only:

$$\mathcal{L}_{\mathrm{MLM}}(\theta) \;=\; -\sum_{i \in M} \log p_\theta\big(x_i \,\big|\, x_{\setminus M}\big).$$

Special symbols (for example, [CLS], [SEP], [PAD]) are never masked and are excluded from the loss [Devlin et al., 2019]. This objective forces the encoder to model dependencies among tokens strongly enough to recover the hidden items [Devlin et al., 2019].

Mask selection and replacement policy BERT masks approximately 15% of positions and applies the 80/10/10 replacement rule to the selected indices: with probability 0.8 replace the token by [MASK]; with probability 0.1 replace it by a random vocabulary token; and with probability 0.1 leave it unchanged. The latter two cases reduce overfitting to the literal mask symbol and improve robustness [Devlin et al., 2019].

Why token choice shapes what is learned MLM is defined on tokens, therefore changing the unit changes the inductive bias of the task [Devlin et al., 2019]. With residue tokens, the model emphasises local chemistry and short motifs; with domain tokens and explicit IDR/linker tokens, the prediction problem becomes "which module fits here given its neighbours and position?", exposing architecture-level co-occurrence and ordering regularities directly to the loss [Yu et al., 2019, Mistry et al., 2021, Paysan-Lafosse et al., 2022]. Empirical studies in protein LMs show that performance is sensitive to tokenisation choices (for example, reduced alphabets or subword-like units), underscoring the need to justify the modelling unit [Dotan et al., 2024, Ieremie et al., 2024].

Optimisation and regularisation Following established practice, we train with AdamW and a warm-up then decay schedule, apply dropout to embeddings and sublayer outputs, and may use label smoothing at the prediction head. These choices are standard in BERT-style pretraining and stabilise large-scale encoder training. Mini-batching is straightforward because the encoder processes all positions in parallel [Vaswani et al., 2017].

What the encoder learns under MLM Stacked attention and feed-forward layers produce contextual embeddings whose token representations reflect the entire surrounding context. In proteins, MLM-pretrained encoders capture signals related to structure and function, and benefit from scale in both model size and data volume; evaluation typically uses frozen embeddings with light probes to isolate representation quality from large task-specific heads [Rives et al., 2021, Lin et al., 2023].

Connections to sequence length and compute Self-attention in each encoder layer has $O(L^2)$ cost in sequence length L. Domain-level tokenisation shortens L relative to residues, which directly lowers attention cost or, under a fixed budget, permits longer contexts or larger batches. This aligns computational considerations with the biological motivation for modelling at the domain level [Vaswani et al., 2017].

2.2.3 Tokenisation effects and design space

Tokenisation is not neutral. In language modelling and in biological sequence modelling, changing the unit of analysis changes both what the model is asked to predict and the computational profile of training. In proteins, alternatives to single-residue tokens have been shown to retain strong predictive performance on some tasks while altering inductive bias and efficiency [Dotan et al., 2024, Ieremie et al., 2024]. Moving from residues to domain-level tokens modifies (i) the granularity of the prediction target, (ii) the effective sequence length L, and (iii) the kinds of regularities directly exposed to the objective (for example, domain collocations and positional preferences versus residue-level motifs) [Devlin et al., 2019, Yu et al., 2019]. This section lays out the design space and the consequences that follow from each choice.

Unit of tokenisation Options include residues, reduced alphabets or subword-like units, fixed-length k-mers, and curated biological modules such as domains and explicit IDR segments. Subword and reduced-alphabet schemes compress the vocabulary and can preserve task performance in some settings [Dotan et al., 2024, Ieremie et al., 2024].

Vocabulary definition Open vocabularies learned from data adapt to frequency but may blur biological categories. On the other hand, closed curated vocabularies (for example, Pfam and CATH accessions and IDR tokens) provide stable identifiers and enable hierarchy-aware interpretation [Mistry et al., 2021, Paysan-Lafosse et al., 2022, Sillitoe et al., 2021].

Segmentation method Statistical segmentation (for example, BERT WordPiece) follows token frequency [Devlin et al., 2019], whereas curation-based segmentation follows domain boundaries and disorder annotations.

Special symbols and masking set The set of maskable tokens and masking rate interact with the unit. While masking residues yields many supervised positions per sequence masking domains yields fewer but semantically richer targets per sequence [Devlin et al., 2019].

Effects on compute and capacity Self-attention scales quadratically with sequence length L per layer Vaswani2017. Residue-level tokenisation produces long inputs for large proteins, increasing the $O(L^2)$ cost. Domain+IDR tokenisation shortens L, reducing cost or, at fixed budget, permitting longer contexts or larger batches. Empirically, larger models and more context yield better protein representations under MLM [Rives et al., 2021, Lin et al., 2023]. Thus, tokenisations that reduce L can be traded for increased width/depth or context length, holding total compute approximately constant.

Effects on the learning signal Under MLM the loss sums cross-entropies over masked positions [Devlin et al., 2019]. Token choice sets how much supervision is available per sequence and what dependencies the model must capture to succeed. With residues, the model sees dense supervision and focuses on local chemical patterns and short motifs. With domain+IDR tokens, supervision is sparser but semantically aligned to module-level co-occurrence and ordering. Because domain families exhibit a long-tailed size distribution [Mistry et al., 2021], masking at the module level naturally probes both frequent and rare families; reporting accuracy by frequency bucket helps characterise long-tail behaviour [Rives et al., 2021, Lin et al., 2023].

Effects on the types of regularities captured Different tokenisations surface different regularities to the objective. Residues expose substitution patterns, local motifs, and secondary-structure signatures. Domain tokens expose architecture-level statistics such as characteristic neighbours, copy-number preferences, and positional biases that have been described as grammar-like at the architecture level [Yu et al., 2019]. Because many comparative tools already operate on ordered domain lists when inferring relatedness [Geer et al., 2002, Yang et al., 2020], a domain-level tokenisation aligns model inputs with existing analysis units curated by Pfam/InterPro and CATH [Mistry et al., 2021, Paysan-Lafosse et al., 2022, Sillitoe et al., 2021].

Effects on interpretability Curated domain identifiers and their clan or superfamily relations enable error analysis at known biological levels, for example, within-clan versus across-family confusions [Mistry et al., 2021, Paysan-Lafosse et al., 2022]. Explicit IDR tokens make it possible to separate mistakes on flexible connectors from mistakes on folded modules and to analyse how flexible regions influence neighbouring module predictions [Wright et al., 2015, Robin van der Lee et al., 2014]. Such analyses are harder to express when tokens are purely statistical subwords without a biological ontology.

Effects on evaluation design Token choice interacts with data splits. Leakage-aware benchmarking recommends family- or taxon-held-out partitions and deduplication to avoid overestimating performance through close-homologue overlap [Rao et al., 2019]. When tokens are domains, architecture-aware deduplication and architecture-level held-out splits become natural, complementing standard sequence-identity filtering [Geer et al., 2002, Yang et al., 2020]. Consistency across tokenisations requires matched model capacity, training steps, and data to attribute differences to the unit rather than scale [Rives et al., 2021, Lin et al., 2023].

Trade-offs in summary Residues maximise supervision density and preserve fine-grained chemistry, at the cost of long sequences and higher attention cost. Subwords or reduced alphabets compress sequences and vocabulary but introduce statistical units whose biological meaning may be task dependent [Dotan et al., 2024, Ieremie et al., 2024]. Domains shorten sequences and align with curated biological modules and hierarchies, improving interpretability and directly exposing architecture-level patterns, while shifting supervision to sparser, higher-level targets [Yu et al., 2019, Mistry et al., 2021, Paysan-Lafosse et al., 2022]. The best choice depends on task goals and compute: for architecture-centric questions and hierarchy-aware interpretation, domain-level tokens are attractive; for local biophysics and residue-level labelling, residues

or hybrid schemes may be preferable. In all cases, MLM remains the same objective, but the tokenisation determines which dependencies are easiest to learn and how efficiently the model can allocate capacity [Devlin et al., 2019, Vaswani et al., 2017, Rives et al., 2021, Lin et al., 2023].

2.3 Protein LLMs: architectures, objectives, and evaluation

2.3.1 ProteinBERT and multi-task pretraining

ProteinBERT is an encoder-style protein language model that augments BERT's general recipe with protein-specific choices in both architecture and pretraining objective. The two central ideas are: (i) to maintain two coupled representation paths, a token-local path for per-position information and a global path for whole-protein information, and (ii) to jointly pretrain on sequence reconstruction and Gene Ontology (GO) annotation prediction, so that functional supervision shapes the learned representations during pretraining rather than only at fine-tuning time [Brandes et al., 2022].

Architecture in brief: coupled local and global streams ProteinBERT maintains local and global hidden states throughout the network and allows controlled information exchange between them in each block. Concretely, the model keeps a local tensor of shape $B \times L \times d_{\text{local}}$ (per-position features along the sequence) and a global vector of shape $B \times d_{\text{global}}$ (whole-protein features). Each of six transformer-like blocks updates the local stream with 1D convolutions (narrow and dilated wide layers, kernel size 9; stacking six blocks yields a receptive field of 241 positions), followed by a position-wise fully connected layer, while the global stream is updated by a small multilayer perceptron. The two streams communicate bidirectionally: broadcast fully connected layers send the global vector to all positions, and a global attention mechanism pools information from positions back into the global vector [Brandes et al., 2022].

Global attention with linear complexity ProteinBERT replaces quadratic self-attention over L positions with a global attention module that maps (local sequence, global vector) to an updated global vector. For one head, keys k_i and values v_i are computed from local states s_i, and a query q is computed from the global vector. Attention weights and pooled output are

$$z_1, \ldots, z_L = \text{softmax}\Big(\frac{q^\top k_i}{\sqrt{d_{\text{key}}}}\Big)_{i=1}^{L}, \qquad y = \sum_{i=1}^{L} z_i \, v_i.$$

Multi-head variants concatenate per-head outputs. Because pooling is from L positions into a single global vector, the cost grows linearly with L, enabling very long sequences without the $O(L^2)$ cost of standard self-attention [Brandes et al., 2022]. For comparison, the Transformer encoder's self-attention is $O(L^2)$ [Vaswani et al., 2017].

Sequence length handling and positional information To generalise across a wide range of lengths, ProteinBERT does not use learned or sinusoidal positional embeddings. Instead, it

relies on start/end tokens, convolutional context, and global–local exchange to convey positional cues. During pretraining, the model cycles through sequence lengths (128, 512, 1024 tokens) to avoid length overfitting and shows modest degradation on much longer sequences at test time [Brandes et al., 2022].

Parameter count and efficiency The published model has approximately 16 million parameters-much smaller than many contemporary protein LMs - yet attains competitive performance on diverse benchmarks. Through linear-in-L global attention and a convolutional local path, the model scales to long inputs while preserving end-to-end differentiability of both local and global outputs [Brandes et al., 2022].

Pretraining objective: dual denoising on sequence and GO ProteinBERT couples sequence-level denoising with GO supervision during pretraining. The input consists of a corrupted amino-acid sequence and a corrupted GO-annotation vector (dimension 8,943 for the most frequent GO terms). Sequence corruption replaces tokens with probability 0.05 (not the BERT 15%/80–10–10 scheme); GO corruption randomly drops true annotations with probability 0.25 and adds false annotations at very low rate (0.01% per missing term). For half of the examples, GO inputs are zeroed to force prediction from sequence alone. The total loss is the sum of categorical cross-entropy over sequence tokens and binary cross-entropy over GO terms:

$$\mathcal{L} \;=\; -\sum_{i=1}^{\ell} \log \hat{S}_{i,\,S_i} \;-\; \sum_{j=1}^{8943} \Big[A_j \log \hat{A}_j + (1 - A_j) \log(1 - \hat{A}_j) \Big],$$

where S_i is the true token at position i, $\hat{S}_{i,\cdot}$ the predicted token distribution, and $A_j \in \{0, 1\}$, $\hat{A}_j \in [0, 1]$ are the true and predicted indicators for GO term j [Brandes et al., 2022].

Pretraining scale and data hygiene Pretraining uses about 10^6 UniRef90 representatives; approximately 46 million sequences carry at least one of the 8,943 GO terms. To reduce label leakage into downstream benchmarks, GO annotations are stripped from pretraining for any protein with at least 40% BLAST similarity to a benchmark test sequence. The schedule alternates sequence lengths and continues to improve sequence-model loss across multiple passes, with GO loss saturating earlier [Brandes et al., 2022].

Fine-tuning protocol and outputs Fine-tuning adds a small task head on top of either the global state (sequence-level tasks) or the local states (residue-level tasks). A two-stage schedule is used: first freeze the pretrained body and train the head; then unfreeze and fine-tune end-to-end with early stopping and learning-rate reduction on plateau. Outputs use task-appropriate activations (softmax, sigmoid, linear). During fine-tuning and evaluation, GO inputs are set to zeros so that no privileged information beyond sequence is used [Brandes et al., 2022].

Results at a glance On the TAPE suite (secondary structure, remote homology, fluorescence, stability), ProteinBERT is competitive with or close to larger models despite its smaller size; longer pretraining improves downstream performance on several tasks without clear saturation. Ablation indicates that the GO objective contributes to gains on structure-related tasks such

as secondary structure and remote homology. The model also shows robustness across sequence lengths, with modest degradation for very long sequences and occasional improvements for specific tasks [Brandes et al., 2022].

Inductive bias and relation to this thesis Relative to a vanilla Transformer encoder, the inductive bias in ProteinBERT arises from three sources: global–local coupling, GO supervision during pretraining, and a length-agnostic design (no positional embeddings; linear-cost global attention; convolutional context) that targets realistic sequence-length variability [Brandes et al., 2022]. In this thesis, by contrast, the bias is injected primarily through token choice (domain and IDR tokens) and evaluation design, while keeping pretraining purely self-supervised (MLM). The two strategies—objective-level supervision versus tokenisation-level bias-offer complementary routes to protein-aware representations [Brandes et al., 2022, Devlin et al., 2019, Vaswani et al., 2017].

2.3.2 BERT in T-cell receptor (TCR) modelling: a usage pattern

A recurring pattern in immune-repertoire modelling of T-cell receptors mirrors language pretraining: (i) pretrain a BERT-style encoder on unlabeled receptor sequences using masked-token prediction, (ii) freeze the encoder and extract per-sequence embeddings, and (iii) evaluate with lightweight classifiers or unsupervised clustering. This pipeline has been applied to TCRs and now serves as a reference workflow for representation quality in short, function-rich sequences [Devlin et al., 2019, Rives et al., 2021].

Data and pretraining objective Typical TCR corpora comprise CDR3-centric amino-acid sequences drawn from curated repositories, optionally stratified by chain (TRA/TRB). Pretraining follows masked amino-acid modelling: a random subset of residues is hidden and the model is trained to predict them from bidirectional context, yielding contextual embeddings that transfer across tasks [Devlin et al., 2019]. This setup aligns with encoder-only Transformers and requires no labels at pretraining time.

Frozen embeddings for classification After pretraining, the encoder is used as a fixed feature extractor that maps each sequence to a vector, for example by mean pooling hidden states or by using a special classification token. Simple classifiers such as linear models or support-vector machines trained on these frozen embeddings can predict antigen specificity; dimensionality reduction (for example, PCA) is often applied before the classifier to keep the downstream head intentionally small so that gains are attributable to the encoder [Rives et al., 2021].

Unsupervised structure: clustering and retrieval Frozen embeddings also support label-free analysis. Nearest-neighbour retrieval and graph-based clustering reveal specificity groups that are visible in low-dimensional projections such as UMAP. In antigen-specific cohorts, clustering quality is summarised by repertoire-focused metrics introduced in earlier TCR studies, where groups of sequences sharing antigen reactivity were identified from sequence features
[Glanville et al., 2017]. These analyses provide an additional, label-free check on representation quality.

Evaluation design and leakage control Immune-repertoire datasets contain many near-duplicate sequences, so leakage-aware evaluation is essential. Family- or taxon-held-out splits and explicit deduplication are recommended to avoid inflated performance on random splits that place closely related sequences in both train and test [Rao et al., 2019]. Cross-patient generalisation is a further safeguard that reflects realistic use-cases.

What this usage pattern establishes First, it validates the pretrain-then-probe paradigm for short immune receptors: with limited labels, frozen embeddings already support meaningful classification and clustering, consistent with observations in larger protein corpora [Rives et al., 2021]. Second, it motivates simple, transparent probes and geometry-based diagnostics that complement supervised metrics.

Relevance and limits for domain-level modelling CDR3 sequences are short and do not test multi-domain composition, so TCR studies are not a direct probe of architecture-level reasoning. Nevertheless, the pattern of masked-token pretraining, frozen-embedding extraction, simple probes, and unsupervised geometry checks directly informs our methodology for protein LMs. In this thesis we retain the same objective and evaluation style while changing the unit of tokenisation from residues to curated protein modules (domains and IDR), isolating tokenisation effects under a well-understood encoder-and-probe workflow [Devlin et al., 2019, Rives et al., 2021].

2.4 Data Sources

Before detailing the construction of the final corpus in the next chapter, this section reviews the external resources from which the component annotations originate.

CATH is a hierarchical classification of protein domains derived primarily from three-dimensional structures. It groups domains into four nested levels: **Class** - overall secondary-structure content, **Architecture** - spatial arrangement of secondary-structure elements, **Topology/Fold** - connectivity of the architecture, and **Homologous superfamily** - inferred common ancestry, hence the acronym CATH [Sillitoe et al., 2021]. For a simple demonstration we can use Myoglobin. It is a single all-α domain with CATH 1.10.490.10 (Mainly Alpha, Orthogonal Bundle, Globin-like, Globins). In our corpus, such domains become discrete tokens (e.g. `3.40.50.300`), and the linker segment is represented explicitly, aligning tokenisation with structure-aware domain parsing.

The resource is built by extracting domains from experimentally determined structures (PDB - Protein Data Bank) and organising them by automated structure comparison and expert curation. Sequence models are then generated to propagate superfamily membership to related sequences without solved structures. CATH is regularly updated with new PDB structures and increasingly with structure predictions to widen coverage. Its pipelines combine automated structure comparison, clustering, and manual validation to maintain stability of the hierarchy while incorporating growth in structural data .

CATH is complementary to Pfam and InterPro. The later two provide **sequence-driven** families and integrative entries, whereas CATH provides a **structure-driven** hierarchy. Using both allows us to analyse whether domain-level embeddings align with sequence homology, structural

fold, or both, and to perform **hierarchy-aware error analysis** (for example, within-superfamily versus across-fold confusions) under a unified evaluation protocol [Sillitoe et al., 2021].

AlphaFold Protein Structure Database (AFDB) provides predicted three-dimensional structures for the vast majority of known protein sequences, together with per-residue confidence scores and quality diagnostics [Varadi et al., 2022]. Entries are keyed by UniProt accessions, enabling straightforward joins to sequence-centric resources. For each protein, AFDB distributes a coordinate model and confidence summaries that allow downstream pipelines to decide which regions are reliable and which are uncertain.

AFDB publishes two complementary confidence measures. First, the per-residue predicted Local Distance Difference Test (pLDDT) score reports the model's confidence in local backbone geometry on a 0–100 scale. Scores above roughly 70–80 usually indicate well-defined local structure, while very low scores often correlate with flexibility, intrinsic disorder or the model being incorrect [Varadi et al., 2022]. Second, the predicted aligned error (PAE) matrix captures the expected positional error between any pair of residues when the structure is aligned on one residue or the other. Low off-diagonal PAE blocks suggest confidently placed domain–domain relationships, whereas high PAE between blocks indicates uncertain inter-domain orientation even if within-domain pLDDT is high [Varadi et al., 2022]. These two signals together distinguish confidently folded domains from flexible linkers and from uncertain domain packing.

AFDB spans essentially all UniProt reference proteomes, providing structures and confidence metrics at proteome scale with regular releases that track UniProt identifiers and versioning [Varadi et al., 2022]. Because AFDB mirrors UniProt accessions, models can be joined losslessly to sequence annotations (for example, Pfam domains, InterPro entries) and to external structural taxonomies. This identifier consistency underpins corpus construction, where AFDB-derived information is merged with sequence-defined domain calls.

We use AFDB in three roles. (i) As a structural field that supports domain boundary inference: pLDDT and PAE patterns delineate compact, confidently folded regions from flexible or uncertain segments, which is a prerequisite for creating domain-level tokens when sequence-only methods are incomplete. (ii) As a source of quality metadata: per-domain statistics such as mean or median pLDDT are propagated into the corpus so that models and evaluations can condition on confidence (for example, analysing performance separately on high- versus low-confidence regions). (iii) As a bridge to structure-based labels: AFDB models are the starting point for mapping domains to CATH categories when structural similarity is detected, enabling hierarchy-aware evaluation alongside sequence-based Pfam and InterPro annotations.

In summary, AFDB supplies the proteome-scale structural context and confidence signals required to segment long proteins into putative domains and linkers, to attach per-segment quality attributes, and to cross-reference structure-derived units with sequence-defined annotations. This makes AFDB a central component of the domain - and linker-aware corpus used for this study.

UniProt is the central knowledgebase for protein sequences and annotations, designed to provide a comprehensive, high-quality, freely accessible set of protein records with functional information. It comprises a reviewed section (UniProtKB/Swiss-Prot) curated by experts and an unreviewed section (UniProtKB/TrEMBL) annotated automatically, alongside companion resources for clus-

tering (UniRef) and historical sequence tracking (UniParc). UniProt serves as a hub that cross-links to hundreds of external databases and integrates large community datasets, making it the standard entry point for proteome-scale analyses [UniProt].

Recent releases report more than 253 million sequence records in UniProtKB, reflecting continuous growth driven by genome projects and metagenomics. UniProt publishes releases approximately every eight weeks and provides both interactive access and programmatic interfaces for retrieval at scale [UniProt].

Swiss-Prot entries are manually curated from the literature with structured feature annotations (for example, domains, active sites, variants, binding sites). TrEMBL entries are annotated by automated systems informed by expert rules and integrated signatures. UniProt increasingly leverages machine-learning assisted triage for curation while preserving explicit evidence tracking in records [UniProt].

UniProt integrates AlphaFold structure predictions for a large majority of entries, exposing per-residue pLDDT confidence on entry pages and through programmatic access. Feature-rich visualisations (for example, ProtVista) map structural and sequence features such as variants, PTMs, and ligand sites, enabling direct alignment of sequence annotations with 3D context [UniProt].

UniProt provides browsable entry pages, bulk downloads, and multiple APIs to support large-scale queries, reproducible pipelines, and integration with external analytics [UniProt].

In our corpus construction, UniProt is the authoritative source of canonical and isoform sequences and stable accessions to anchor all other annotations and taxonomic metadata for stratified sampling and evaluation. Furthermore, it cross-references to Pfam, InterPro, CATH, and AlphaFold to join identifiers consistently across resources. Reviewed status and evidence codes are retained for quality control. When multiple isoforms exist, the canonical sequence is selected unless a specific isoform has stronger supporting evidence for a given annotation. These choices ensure that domain and linker tokens are attached to stable sequence records with traceable provenance, and that downstream evaluation can leverage UniProt's cross-resource links and structural integration [UniProt].

Pfam was introduced in Background Section as the canonical sequence-defined resource for protein families and domains. Here we summarise the aspects that matter operationally for building the corpus.

Pfam classifies protein sequences into families (often corresponding to domains) using profile hidden Markov models (HMMs) built from manually curated seed multiple sequence alignments. Each entry provides a seed alignment, a profile HMM, and a curated gathering threshold to standardise inclusion in HMMER searches. Sequences scoring above this threshold are added to the full alignment. Related entries are grouped into higher-level clans. This curation gives stable identifiers, consistent boundaries, and a clear hierarchy for large-scale annotation [Mistry et al., 2021].

In recent releases, the database contains on the order of 1.8×10^4 families and about 6.3×10^2 clans, with sequence and residue coverage of UniProtKB around 77% and 53%, respectively. Release notes document proteome coverage and the long-tailed distribution of family sizes and taxonomic ranges, both important for downstream sampling and frequency analyses (Mistry et al., 2021).

Entries are created by iteratively searching the HMM against a target database (pfamseq,

based on UniProtKB reference proteomes), adding distant homologues, and refining the seed; profiles are then searched proteome-wide and distributed via the website and flat files. A key practical point is to use the per-family gathering thresholds rather than a single global E-value cut-off; a global cut-off either increases false positives or reduces sensitivity depending on the choice. Pfam historically enforced a non-overlap rule between distinct families, later relaxed to allow small overlaps and to permit overlaps within a clan, with competition rules for displaying the best-scoring match [Mistry et al., 2021].

Pfam annotates entry types (Family, Domain, Repeat, Coiled-coil, Motif, Disordered) to aid downstream use, and has audited repeat and disordered types using RepeatsDB and MobiDB-lite. In parallel, Pfam-B provides automatically generated multiple sequence alignments for regions not covered by Pfam-A (now built by clustering with MMseqs2), serving as a substrate for future family building [Mistry et al., 2021].

We run Pfam against UniProt using each family's trusted (gathering) thresholds and, for every match that passes, we store the domain's accession (the token), its start–end positions, and the match strength (bit score and E-value). If a family sits inside a broader clan (superfamily), we also record that ID so we can analyse results at the clan level. We keep each entry's type (e.g., Repeat, Disordered) to enable filtering and to distinguish structured domains from linkers or intrinsically disordered regions. When domains overlap, we follow Pfam's rules: small overlaps are allowed; if two hits come from the same clan, we keep the higher-scoring one; any remaining clashes are broken in a fixed, deterministic way so each protein ends up with a clean, ordered, non-overlapping domain list. This gives the language model stable, community-curated units with consistent boundaries and a usable hierarchy, while the saved scores and types support quality control and interpretation [Mistry et al., 2021].

InterPro was also introduced in the Background section. It is an integrative resource that unifies protein "signatures" from many specialist member databases into non-redundant entries with stable accessions, curated abstracts, entry types (family, domain, repeat, site, homologous superfamily), cross-references, and Gene Ontology (GO) mappings. By consolidating overlapping signatures that describe the same biological entity, InterPro provides a single, stable handle per concept and a coherent hierarchy that is consistent across releases. InterPro is updated on an eight-week cycle, tracks UniProtKB growth, and maintains high coverage at proteome scale; for example, version 90.0 reported roughly 82% UniProtKB coverage and more than forty thousand entries built from over fifty thousand integrated signatures [Paysan-Lafosse et al., 2022]

Member databases emphasise complementary aspects of sequence space: Pfam profile HMMs for conserved regions, SMART domain models, CDD curated models, PROSITE patterns and profiles, PANTHER families, CATH-Gene3D structure-informed families, TIGRFAMs/NCBIFAMs, SUPERFAMILY HMMs, PRINTS fingerprints, PIRSF families, SFLD enzyme superfamilies, and HAMAP rules, among others. InterPro merges these signatures when they describe the same family or domain and attaches a single entry with a stable accession, name, abstract, entry type, and optional GO terms that can be projected to all matched proteins [Paysan-Lafosse et al., 2022] This reduces redundancy, harmonises nomenclature, and provides a consistent layer for downstream analyses that would otherwise need to reconcile multiple, partly overlapping calls.

InterPro also aggregates complementary sequence features that are not strictly families or

domains but are essential for large-scale annotation: consensus intrinsic disorder (MobiDB-lite), signal peptides (SignalP), transmembrane helices (TMHMM), coiled coils, and AntiFam profiles for filtering common spurious ORFs (Paysan-Lafosse et al., 2023) These features appear in the same viewer and API as family/domain hits, enabling pipelines to reason about structured segments and IDRs in a single place-important for domain-level tokenisation.

For this study it serves in three main ways. First, as an integration layer: when member databases (e.g., Pfam, SMART) annotate the same region, the InterPro entry defines a single token, avoiding redundancy. Second, as a source of hierarchy and interpretability. Its stable, versioned structure and GO links enable hierarchy-aware evaluation (e.g., rolling up near-misses within the same entry or function class) and clearer semantics in error analysis. Third, as feature context. Through MobiDB-lite, SignalP and TMHMM, InterPro separates structured domains from predicted disordered linkers and explicitly flags signal peptides and transmembrane segments, aligning the token stream with widely used community annotations [Paysan-Lafosse et al., 2022].

TED The Encyclopedia of Domains is a large, structure-first catalogue of protein domains parsed from the AlphaFold Protein Structure Database at proteome scale. TED was built to turn the full-length AlphaFold models into domain-level units with identifiers, quality measures, and cross-links to existing structure taxonomies. Thereby making the 3D information in AlphaFold searchable and analysable at the same granularity as sequence-based domain resources. In its initial release, TED processes the entire AFDB v4 and reports more than 371 million domain segments across 214 million UniProt sequences, after deriving a non-redundant working set of 188.9 million unique sequences ("TED-100") for most analyses. Only 5% of proteins in TED-100 lack any domain assignment, versus 26–34% when using sequence-only methods (Pfam, Gene3D), indicating that structure-based parsing substantially increases detectable domain coverage. Most TED domains fall in the "high or very high" AlphaFold confidence bins by pLDDT (with only 2% in the lowest bin).

TED processes AlphaFold models by applying three domain parsers (Merizo, Chainsaw, UniDoc) and keeping only regions where at least two agree, removing low-confidence or conflicting segments to reduce artefacts and produce clean domain spans. These domains are then labelled using Foldseek and Merizo-search against CATH representatives, followed by an embedding-based method (Foldclass-search) with TM-align checks, assigning about 194 million domains to CATH superfamilies and 36 million to folds, with 171 million superfamily labels independently validated; this expands CATH coverage by 15%. Domains are sequence-clustered with MMseqs2 into 121 million groups, supporting novelty detection. Unmatched clusters are filtered and re-analysed, leading to 6,433 repeat-rich domains (e.g. new 11-bladed propellers) and 7,427 putative novel folds lacking structural matches, some with predicted functions like zinc or nucleic acid binding. Working from full chains, TED also maps domain–domain interactions, reporting 27.3 million instances across 13,771 interacting superfamily pairs-about double CATH's catalogue-revealing many new hubs. Its taxonomic breadth shows both widespread fold reuse and lineage-specific enrichments, such as transport-related families overrepresented in bacteria. While some anomalies occur due to AlphaFold variability, TED manages quality through consensus chopping, filtering, and cross-validation, though it misses certain cases like viral proteins or engineered constructs; nonetheless, it reveals many new plausible folds and interactions beyond sequence-only approaches.

TED matters because it turns AlphaFold's proteome-scale structures into high-coverage domain boundaries, even when sequence signatures are weak, so proteins without reliable sequence-defined domains still receive usable tokens. Through structural links to CATH, it enables hierarchy-aware roll-ups at the superfamily and fold levels and supports interpretable error analysis. Its consensus segmentation also flags candidate novel domains beyond known Pfam and CATH space, and its architecture context lets one test whether embeddings respect domain co-occurrence patterns. In short, TED implements AFDB predictions into domain units with rich labels and confidence metadata that plug directly into a domain-token language model and support principled corpus building and evaluation.

Chapter 3

Data and Methodology

As it was stated in the chapters above this project implements a self-supervised learning approach using Masked Language Modeling with transformer architecture to learn protein domain representations, following the BERT paradigm but adapted for protein domain sequences. The training corpus used in this thesis was provided by Dr. Daniel Buchan.

3.1 Corpus construction

This section describes the end-to-end procedure used to assemble the final, domain–linker token corpus from the component resources reviewed above. The corpus was built by first extracting UniProtKB accessions from the TED assignments to define a stable protein set used as keys for all joins. InterPro match exports were then scanned to recover MobiDB-lite disorder segments as per-sequence intervals, providing candidate IDR regions consistent with disorder literature and InterPro's integrated tracks. Intervals were merged with TED consensus domains (AFDB-derived) and InterPro-assigned Pfam domains, giving precedence so that TED overrides Pfam and Pfam overrides disorder, ensuring contiguity and yielding an ordered list of labelled protein segments. Segments labelled "unknown" were marked as UNK, chain lengths were taken from AFDB FASTA sequences keyed by accession, and the corresponding amino-acid subsequences were extracted. Unknown subsequences were clustered with MMseqs2 Linclust at a chosen identity threshold under coverage constraints, and each UNK span was relabelled by its cluster UNKκ to create stable tokens without exploding the vocabulary. Standard summaries were computed across proteins to sanity-check token counts and label frequencies, and corpus variants were generated to support ablations-keeping all tokens, collapsing all UNKs to a single token, removing single-domain chains, or combining both filters-followed by pruning chains with discontiguous labels or no recognised annotations. This design aligns with community resources and their intended use: UniProt for identifiers, Pfam gathering thresholds for family calls, InterPro (MobiDB-lite) for disorder, AFDB for accession-aligned sequences, TED for consensus spans, and CATH for structure-based hierarchy.

3.2 Data Processing

This section sets out how protein domain sequences are prepared for masked-language pre-training with Transformers. The overall process has three stages: first a domain-level vocabulary is built, then domain sequences are tokenised into numerical form, and finally masked sequences are generated so that the model can be trained to predict hidden elements. Each stage ensures that the raw data is systematically shaped into inputs and outputs that a Transformer can handle.

3.2.1 Domain Vocabulary Construction

The pipeline begins by creating a vocabulary of domain tokens. To do this, the full corpus is scanned and the occurrences of each domain identifier are counted. Only those domains that appear often enough to pass a minimum frequency threshold are kept. If the vocabulary risks becoming too large, it can be restricted to the most frequent domains, for example the top five. Any identifier that is not retained is treated as out-of-vocabulary and represented by the special placeholder `[UNK]`. In addition to these identifiers, several special tokens are reserved for technical purposes: `[PAD]` is used to equalise sequence lengths during batching, `[MASK]` hides tokens during training, `[UNK]` stands in for unseen domains, `[CLS]` represents whole-sequence information, and `[SEP]` marks sequence boundaries. Each token-whether a domain identifier or one of these reserved symbols-is then assigned a unique index number. This mapping is defined as a function from the set of tokens to integer values between zero and the vocabulary size minus one:

$$\mathrm{idx} : \mathcal{D} \to \{0, \ldots, V-1\}.$$

The mapping is stored in a dictionary structure for efficient lookup, such as:

```
domain2idx = {"PF00001": 0, "PF00002": 1, ...}
```

3.2.2 Sequence Tokenisation

With the vocabulary in place, domain sequences can be turned into model inputs. Starting from an original sequence of domains, each domain identifier is first replaced with its corresponding index number. A `[CLS]` token is then placed at the beginning of the sequence and a `[SEP]` token at the end, extending the sequence length by two. Because models expect fixed-length inputs, the sequence is either cut short if it exceeds the maximum allowed length $L_{\max}$ or extended with `[PAD]` tokens if it falls short. To help the model distinguish genuine tokens from padding, an attention mask is created in parallel: every real token, including the special ones, is marked with a 1, while each padding position is marked with a 0.

For example, the short sequence [`PF00001`, `PF00002`] becomes $[3, 0, 1, 4]$ after mapping to indices and adding the special tokens, with the mask marking all four positions as real tokens.

3.2.3 Masked Sequence Generation (MLM)

The final stage prepares training examples for the masked language modelling objective. After padding and adding special tokens, the valid (non-padding) positions in the sequence are identified. Each position is then considered for masking with probability $p = 0.15$, meaning that on average 15 per cent of the tokens are replaced by the special `[MASK]` symbol.

To supervise the model, a target label vector y is created: the label at a masked position is set to the original token index $x_i^{(\text{orig})}$, while all other positions are filled with the ignore value -100, which ensures that they do not contribute to the loss calculation:

$$y_i = \begin{cases} x_i^{(\text{orig})} & \text{if } i \in M, \\ -100 & \text{otherwise.} \end{cases}$$

Training uses token-level cross-entropy loss, but it is only applied at the masked positions:

$$\mathcal{L}(\theta) \;=\; -\frac{1}{|M|} \sum_{i \in M} \log p_\theta(y_i \,|\, \text{masked } x_{1:L_{\max}},\, m)\,.$$

In practice, this means the model is updated only where information has been deliberately hidden. For instance, the input sequence $[3, 0, 1, 4]$ might become $[3, 1, 1, 4]$ after masking, with the label vector $[-100, -100, 1, -100]$. In this example, the model is required to recover the correct token "1" at the third position, while ignoring the others.

3.3 Model Architecture

The model we employ is a lightweight Transformer equipped with a multi-scale attention block. Unless specified otherwise, the model width is fixed at $d_{\text{model}} = 256$, with a dropout rate of 0.1, and the network consists of $L = 2$ Transformer layers. Multi-head attention is used with $h = 8$ heads, meaning that each head has a dimension of $d_h = d_{\text{model}}/h = 32$.

3.3.1 Multi-Scale Attention

Let $X \in \mathbb{R}^{T \times d_{\text{model}}}$ denote the token embeddings after positional encoding has been applied. The attention mechanism combines both local and global contexts. Local attention follows the standard multi-head self-attention (MHSA) approach, which produces

$$\text{MHSA}(X) = \text{Concat}\big(H_1, \ldots, H_h\big) W^O,$$

where each head is defined as $H_j = \text{softmax}\left(\frac{Q_j K_j^\top}{\sqrt{d_h}}\right) V_j$ with $Q_j = XW_j^Q,\ K_j = XW_j^K,\ V_j = XW_j^V$. Alongside this, global attention is obtained by computing a sequence-level context. This is achieved by applying global average pooling over the sequence, giving

$$\bar{x} = \frac{1}{T} \sum_{t=1}^{T} X_t, \qquad g = \bar{x} W^G + b^G \in \mathbb{R}^{d_{\text{model}}},$$

and the resulting vector g is then broadcast to all positions in the sequence. To combine both sources of information, the local and global contexts are concatenated and passed through a linear transformation with a pointwise nonlinearity, producing

$$Z = \phi\big(\big[\mathrm{MHSA}(X) \parallel g\big]W^F + b^F\big) \in \mathbb{R}^{T\times d_{\text{model}}},$$

where $\phi(\cdot)$ is the ReLU activation function.

3.3.2 Transformer Layer

Each Transformer layer processes the input in several steps. Discrete domain indices are first mapped to continuous vectors in $\mathbb{R}^{256}$ to form embeddings, and positional encodings are added so that the model can incorporate order information. The embeddings are then passed through the multi-scale attention block described above. After this, a feed-forward network is applied, which consists of two linear transformations with a ReLU activation between them, written as

$$\mathrm{FFN}(x) = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2,$$

with the hidden layer typically set to size $4d_{\text{model}}$. Throughout the process, layer normalisation is applied around sublayers and dropout of 0.1 is used to reduce overfitting.

3.3.3 Dual Output Modes

The model operates in two modes depending on the task. In the masked language modelling mode, the goal is to predict hidden tokens. To do this, a classifier is applied independently at each sequence position:

$$\ell_t = Z_t W^O + b^O \in \mathbb{R}^{|\mathcal{V}|}, \qquad t = 1, \ldots, T,$$

which produces logits over the vocabulary $\mathcal{V}$.

In the embedding mode, the model generates a representation for the entire sequence. This is done by applying mean pooling over all valid positions that are not padding tokens, yielding a 256-dimensional vector:

$$e \;=\; \frac{1}{|\mathcal{S}|}\sum_{t\in\mathcal{S}} Z_t \in \mathbb{R}^{256},$$

where $\mathcal{S}$ is the set of indices corresponding to valid tokens.

Below we provide a chart that visualises the process described in this section.

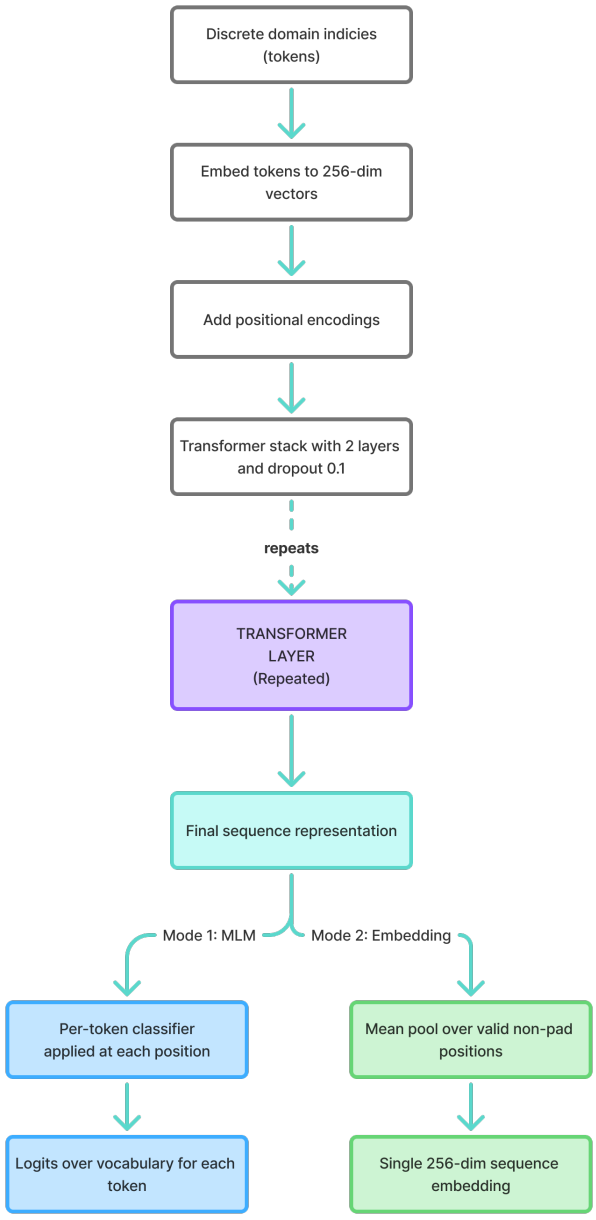

Figure 3.1: Overall architecture of our Transformer-based model, showing token embedding, positional encoding, stacked Transformer layers, and dual output modes for masked language modelling (MLM) and sequence embedding.

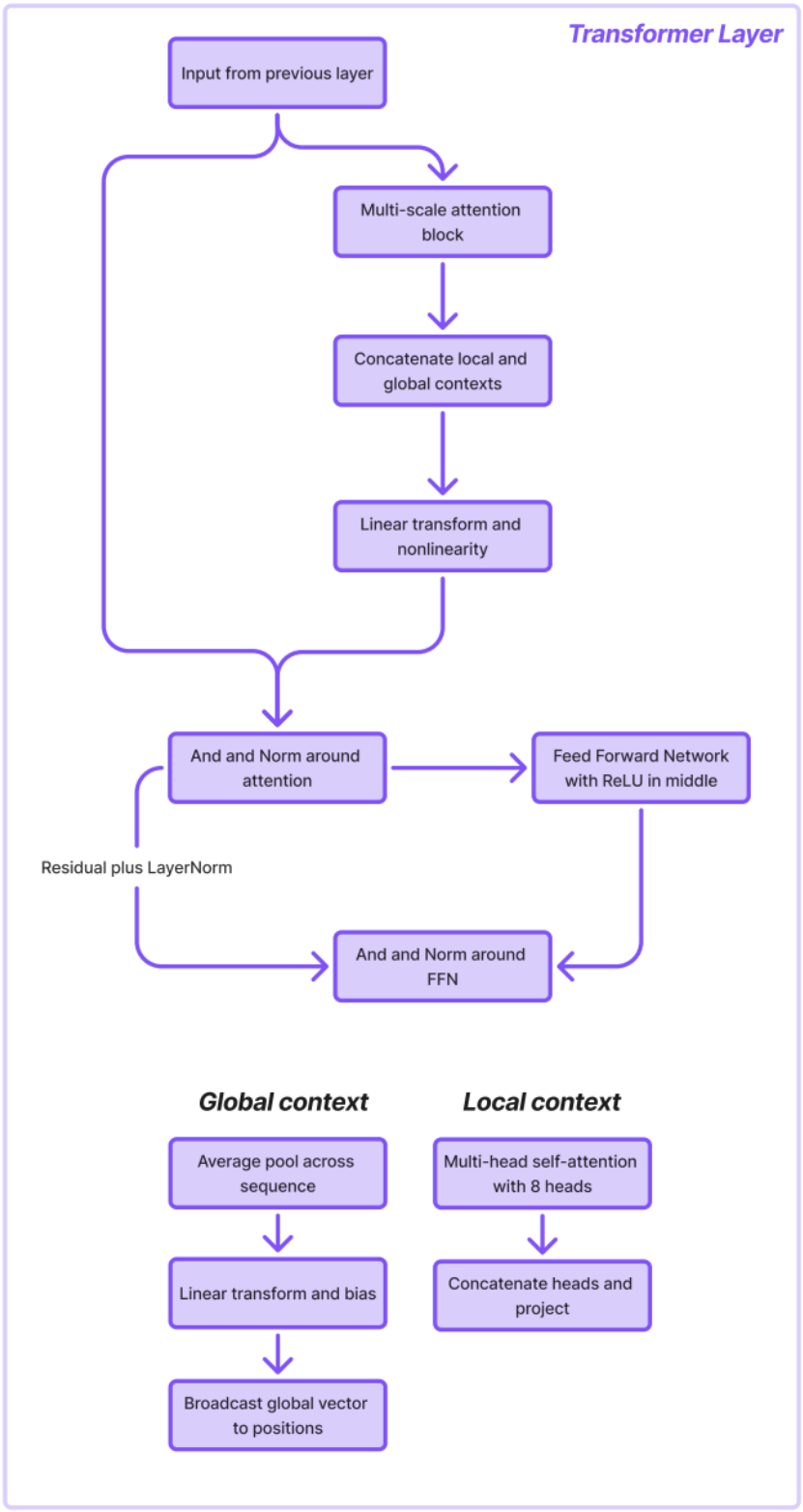

Figure 3.2: Detailed structure of a single Transformer layer, illustrating the multi-scale attention block, residual connections, feed-forward network, and integration of local and global contexts.

3.4 Training Process

We train the model using a masked language modelling (MLM) objective in a self-supervised fashion. At a high level, each mini-batch contains tokenised sequences of domains, a version of those sequences where some tokens are masked, and a set of labels in which only the masked positions are marked as supervised. This setup allows the model to learn directly from raw sequences without the need for external annotations. We trained the main model on NVIDIA A100 SXM GPU (80GB VRAM, 125GB RAM, 16 vCPU). 57% of RAM was consumed when the model was loaded to the GPU. The process took around 15 hours.

3.4.1 Self-Supervised Learning Setup

The training objective is to recover masked tokens from their surrounding context. In practice, the model operates over the domain vocabulary and is asked to predict the original domain identifier at positions that have been deliberately hidden. Masking is applied independently to $p = 0.15$ of the tokens that are not `[PAD]`, meaning that roughly fifteen percent of the input sequence is replaced by a special mask symbol. The targets for learning are constructed so that masked positions take the value of the original token id, while all other positions are filled with the ignore index -100, which ensures that they are excluded from the loss computation.

3.4.2 Training Configuration

The model is trained with the Adam optimiser, using a learning rate of 1×10^{-4}. The loss function is token-level cross-entropy, but it is only evaluated at the masked positions, with all other positions ignored by virtue of the -100 label. Each training batch contains 32 sequences, and the model is trained for 4 epochs in total. Sequences are capped at a maximum length of 10 domains, meaning that longer sequences are truncated while shorter ones are padded to this fixed size.

3.4.3 Objective Function

Let $Z \in \mathbb{R}^{T \times d_{\text{model}}}$ represent the final hidden states of the Transformer, and let $W^O \in \mathbb{R}^{d_{\text{model}} \times |\mathcal{V}|}$ with b^O be the output classifier parameters. Denote by $M \subset \{1, \ldots, T\}$ the set of indices that were masked, and by y_i the ground-truth token at position i. The per-batch loss is then defined as

$$\mathcal{L} = -\frac{1}{|M|} \sum_{i \in M} \log \operatorname{softmax}\big(Z_i W^O + b^O\big)_{y_i}.$$

This expression shows that the model is penalised only for predictions at the masked positions, while all others are ignored.

3.4.4 Training Loop

The training process follows a simple but structured loop. Protein domain sequences are first loaded from CSV files, after which the vocabulary is built and each sequence is tokenised. Masked versions of the sequences and their corresponding labels are then generated, with non-masked

positions assigned the ignore index -100. During the forward pass, the masked sequences are fed into the model to produce logits. The loss is computed against the labels using the defined criterion, and backpropagation is performed. Updates to the parameters are applied using Adam. This cycle is repeated for all mini-batches across the dataset. Once training is complete, the model is saved along with the vocabulary mapping and the final embedding matrix.

3.4.5 Output Files

The training pipeline produces a compact but important set of artefacts for reuse, analysis, and reproducibility. The first is `domain_model.pth`, which contains the encoder weights of the trained transformer. This checkpoint can be used directly for inference (e.g., embedding extraction). The second file, `domain2idx.pkl`, stores the vocabulary mapping from domain names to integer IDs, ensuring that preprocessing during inference exactly matches what was used in training and guaranteeing consistent round-tripping across runs and machines. The third artefact is `embeddings.npy`, which contains 256-dimensional per-protein embeddings stored as a NumPy array; these vectors serve as the basis for downstream analyses such as PCA/UMAP and clustering, and they also support frozen-encoder probes. Finally, `train_losses.npy` provides a concise summary of the learning trajectory by recording MLM prediction accuracy across epochs.

Chapter 4

Results and Evaluation

After we train our final model on the full dataset of 77 million proteins for 4 epochs we get the following losses:

Epoch	**1**	**2**	**3**	**4**
Value	0.46005491	0.45867695	0.45699780	0.44964467

Figure 4.1: Training loss values over 4 epochs, showing gradual decrease during optimisation.

A plot below clearly shows a healthy pattern of how our model is learning. Although loss is constantly decreasing per each epoch the improvements except for the last step are modest. In addition it is very essential to note that 4 epochs could not be enough to capture perfectly the full distribution of domain tokens and their long-range co-occurrence patterns in such a large, long-tailed corpus.

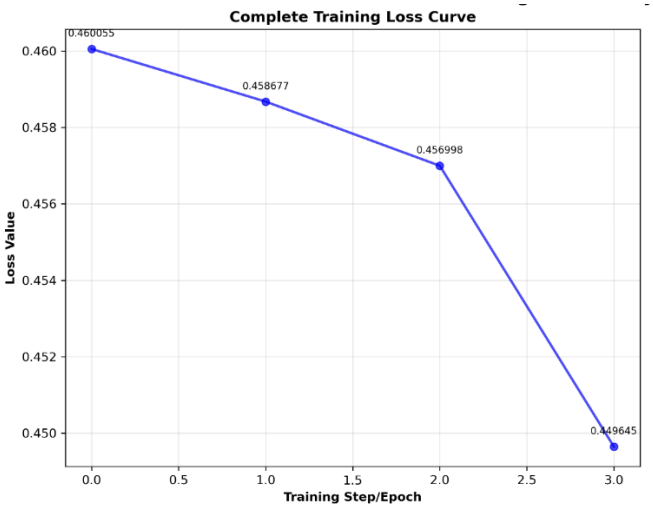

Figure 4.2: Training loss curve over 4 epochs, illustrating steady improvement with a sharper drop in the final step.

To further analise obtained we adopt a three-step evaluation that keeps the encoder frozen and focuses on what the learned embeddings encode:

1. Unsupervised retrieval and clustering to test whether the embedding space exhibits meaningful structure without labels.
2. A simple biological sanity check that relates embedding distance to multi–domain architecture (MDA) similarity, reflecting the long–standing use of domain order/co–occurrence as a proxy for related function.
3. A consistency test that checks if neighbours in our embedding space share GO annotations.

4.1 Unsupervised retrieval and clustering

We start by reducing dimensionality of our embeddings for denoising and neighbourhood construction. Then we build a k-nearest neighbour graph in the reduced space (for k = 50), detect communities with the Leiden algorithm, and visualise the result in two dimensions with UMAP. This pipeline groups proteins by similarity in the embedding space without using labels, and provides an interpretable 2D map to inspect cluster structure. Next, we detail the community-detection stage. After constructing the k-NN graph, we apply the Leiden algorithm, which partitions the graph into well-connected communities. It proceeds in three steps:

1. local moving, where nodes are reassigned to neighbouring communities to improve a quality function
2. refinement, which enforces that each community is internally well connected
3. aggregation, which contracts communities into super-nodes and repeats the process on the coarsened graph.

Furthermore, we proceed with Uniform Manifold Approximation and Projection. UMAP is a dimension-reduction algorithm that produces a 2D embedding in which nearby points in the original space tend to remain nearby, enabling faithful visual inspection of local structure. Conceptually, UMAP builds a weighted k-NN graph that encodes local neighbourhoods of the high-dimensional data, then learns a low-dimensional layout whose graph has similar fuzzy-connectivity. The layout is obtained by optimising a cross-entropy objective with attractive forces for edges and repulsive forces for non-edges. Compared to alternatives, UMAP typically runs faster, preserves more of the global organisation, and scales to larger datasets, while still providing high-quality cluster separation for visual analysis. Key hyperparameters are `n_neighbors` (how local the notion of neighbourhood is) and `min_dist` (how tightly points pack within a cluster).

From the resulting plot we can conclude that proteins as projected points in 2D form well-separated clouds indicating that the embedding space captures meaningful variation in domain composition and organisation, while the cluster assignments provide a concrete partition for further downstream summaries and sanity checks.

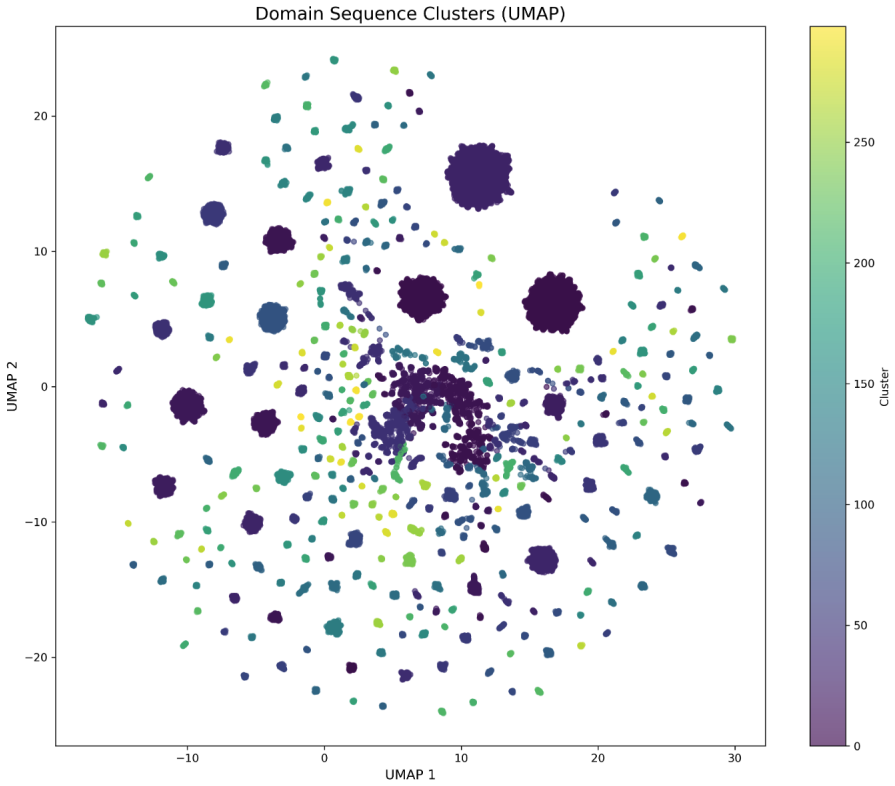

Figure 4.3: 2D embedding showing protein clusters formed with k = 50

We have additionally tried clustering with with a significantly higher value for $k = 200$. The resulting visualisation did not prove to be useful as the clustering produced based on the embedding space was overly fragmented, making it difficult to extract any meaningful structure or interpretability from the results.

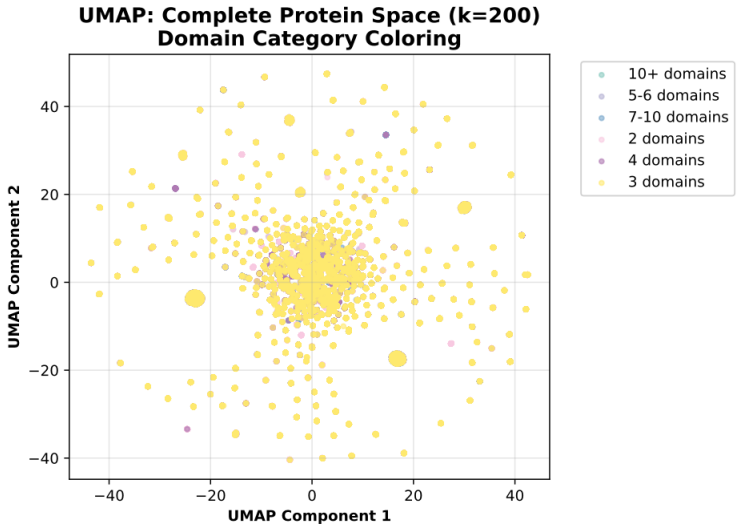

Figure 4.4: 2D embedding showing protein clusters formed with k = 200

4.2 Biological meaning

Unsupervised retrieval and clustering show that proteins form groups in the learned embedding space, a pattern also reported for protein language models trained with masked-token objectives. The next question is whether these clusters make biological sense. A common route would be to test overlap with Gene Ontology (GO) functions, either via probes on frozen embeddings or multi-task pretraining. We deliberately do not pursue a GO benchmark here because it introduces evaluation confounds that are outside the scope of this project: (i) label incompleteness - many UniProt entries carry only electronic or no GO annotations, which complicates ground truth and calibration [Paysan-Lafosse et al., 2022]; (ii) ontology and thresholding issues GO is hierarchical and multi-label, so metric choice and decision thresholds strongly affect scores and (iii) the need for carefully homology-controlled splits to avoid leakage from near-duplicate sequences, which requires substantial task engineering [Rao et al., 2019].

Instead, we adopt a simple, transparent biological sanity check directly aligned with our representation. We quantify whether embedding proximity correlates with similarity of multi-domain architectures (MDAs) ordered lists of domains known to display grammar-like reuse and to track function across proteins. Concretely, we compute a bigram Jaccard similarity between the sets of consecutive domain pairs in two proteins and plot this against cosine distance of their embeddings.

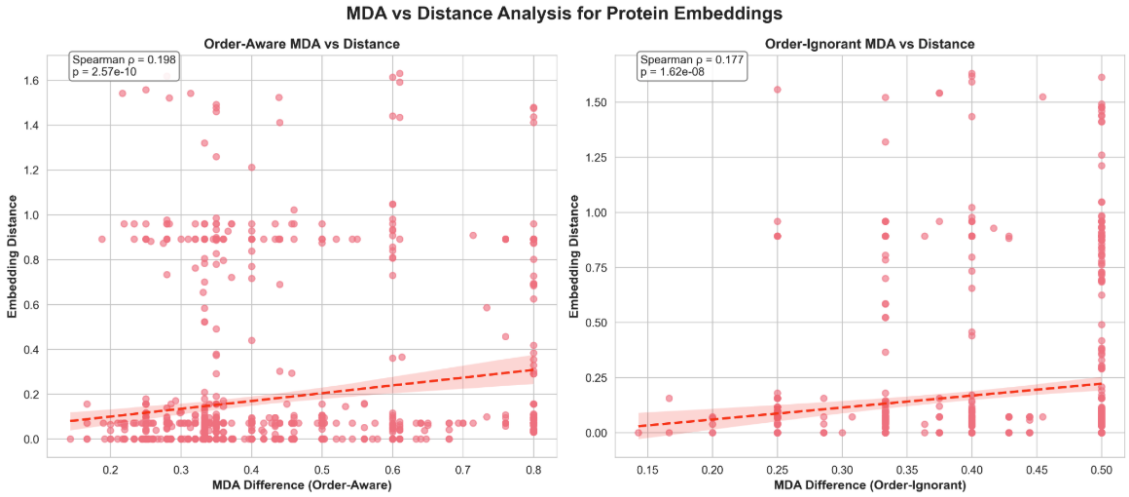

Figure 4.5: Scatter plot showing correlation between protein embedding proximity and multi-domain architecture similarity.

In this study, we examined a balanced sample of 1,000 protein pairs (a detailed list is available in the GitHub repository of this project). The analysis focused on the relationship between the proximity of protein embeddings and the similarity of their multi–domain architectures (MDA). To measure MDA similarity, we used an order-aware bigram Jaccard score, which takes into account not only the presence of domains but also the order in which they appear. When comparing this score with the cosine distance between embeddings, we observed a Spearman rank correlation of $\rho = 0.198$.

For comparison, we also tested a simpler, order–ignorant measure that treats domains as a "bag-of-domains" without considering their sequence. This approach produced a weaker correlation of $\rho = 0.177$. The reduction in correlation strength demonstrates that the embeddings are sensitive to the order of domains, rather than to domain composition alone.

Although the size of the effect is modest - something expected from a label-free, self-supervised encoder - the correlations are both highly significant and consistently positive. This provides some evidence that the model is able to capture meaningful regularities at the level of protein architecture. In particular, proteins that share more of the same domain pairs in the same order tend to appear closer together in the embedding space.

4.2.1 GO consistency test

The purpose of this analysis was to examine whether the closeness of proteins in the embedding space that we constructed corresponds to similarities in their biological function. In other words, we wanted to test if proteins that are positioned near each other in this space also tend to share the same Gene Ontology (GO) annotations. To do this, we measured a quantity we call neighbour GO consistency, which allowed us to evaluate functional similarity directly from the embeddings, without requiring any further training.

For each protein i that was annotated in a given ontology O (with O being one of Molecular Function (MF), Biological Process (BP), or Cellular Component (CC)), we identified its top-k nearest neighbours in the embedding space. Nearest neighbours were determined using cosine similarity, so proteins with more similar embeddings were ranked closer together.

We then checked whether protein i shared at least one GO term in ontology O with each of its k neighbours. This was summarised mathematically as:

$$\text{consistency@}k = \frac{1}{N_O} \sum_{i \in O} \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \mathbb{1}\{\text{GO}_O(i) \cap \text{GO}_O(j) \neq \emptyset\}\,, \tag{4.1}$$

where N_O is the number of proteins annotated in ontology O, $\mathcal{N}_k(i)$ is the set of k nearest neighbours of protein i, and the indicator function checks whether there is at least one GO annotation in common. The measure therefore represents the fraction of neighbours that share at least one GO term with the query protein, averaged across all proteins in the ontology.

We report values of this metric for several neighbourhood sizes, specifically $k \in \{1, 5, 10, 20, 50\}$.

As expected, we observed that GO consistency values decrease as k increases. This is natural, since larger values of k include more distant neighbours, which are less likely to share the same function. Among the three ontologies, the Molecular Function (MF) results showed the strongest signal, meaning that proteins with similar embeddings were more likely to share molecular functions. By contrast, Biological Process (BP) and Cellular Component (CC) showed considerably weaker signals (Fig. 4.6, Table 4.1).

The numerical results are as follows:

- **Molecular Function (MF):** Consistency was relatively high at the closest neighbour level, with 0.462 for k=1, and gradually decreased to 0.241 at k=50 (n=973).
- **Biological Process (BP):** The values were much lower, starting from 0.105 at k=1 and dropping to 0.009 at k=50 (n=76).

- **Cellular Component (CC):** Similarly, CC showed weak but slightly more stable results than BP, beginning with 0.096 at k=1 and decreasing to 0.022 at k=50 (n=52).

Overall, these results indicate that our embedding space is particularly effective at capturing local molecular functionality, whereas capturing broader biological processes or cellular localisation appears to be more difficult. This is consistent with earlier findings in the literature, which have also reported that simple nearest-neighbour transfer methods work best for Molecular Function, while Biological Process and Cellular Component are more challenging to predict reliably.

The evaluation was carried out using Swiss-Prot proteins with Gene Ontology Annotation (GOA) records. We also sampled 1000 proteins to carry out this experiment (a detailed list of these proteins and their GO annotations is also available in the GitHub repository of this project).

Table 4.1: Consistency@k by ontology (sample sizes in parentheses).

	k=1	k=5	k=10	k=20	k=50
MF (n=973)	0.462	0.386	0.336	0.288	0.241
BP (n=76)	0.105	0.055	0.030	0.015	0.009
CC (n=52)	0.096	0.092	0.056	0.036	0.022

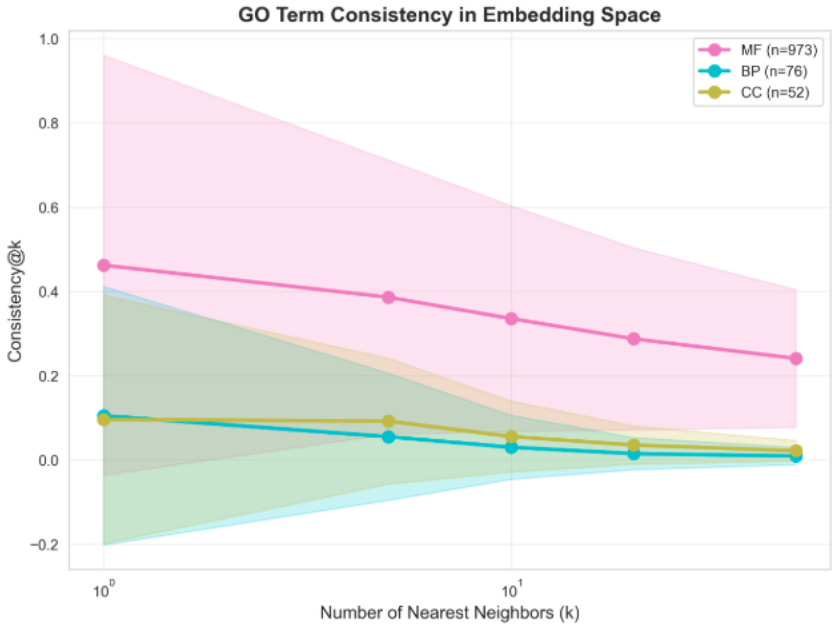

Figure 4.6: Consistency@k results for Molecular Function, Biological Process, and Cellular Component across different neighbour sizes.

4.3 Small summary example

In the example provided in this section, we compare 5 pairs of proteins with each other. The distances between points in the embedding space indicate relative similarity, with selected examples highlighting how closer embeddings correspond to higher similarity levels between sequences in some cases while in the other cases the correlation is weak.

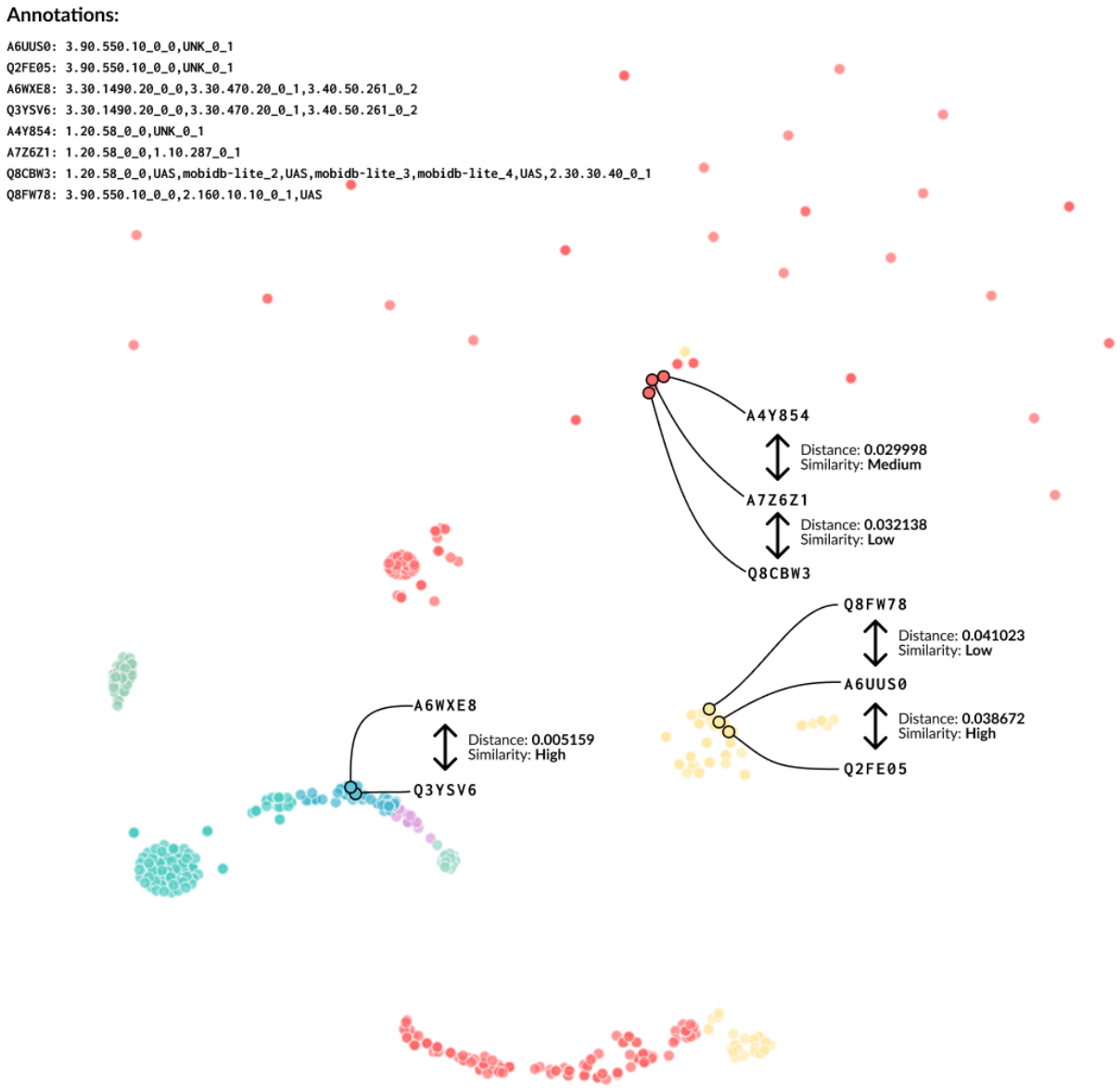

Figure 4.7: Example of a small summary visualisation of sequence embeddings. Points are projected into two dimensions to highlight cluster structure, with annotated examples showing distances and similarity levels between selected sequences.

Chapter 5

Discussion

5.1 Limitations

This project faced several limitations due to resource and scope constraints. The main factors are described below.

5.1.1 Limited training epochs

The model was trained for only a small number of epochs. Protein language models generally require lengthy training schedules in order to stabilise their embeddings and capture subtle co-occurrence patterns across millions of sequences. Here, training was carried out for only four epochs, which is not sufficient to fully learn long-range dependencies or rare combinations of protein domains. Extending the training schedule would almost certainly have improved performance, but it would also have required considerably more computational resources and time than were available for this MSc project. Large-scale efforts, such as ESM-2, have shown that substantial gains are only realised after very long training on massive compute budgets [Lin et al., 2023].

5.1.2 Restricted embedding dimensionality

The dimensionality of the embedding space was deliberately restricted to 256. By contrast, modern protein language models often employ much larger embedding sizes, typically ranging from 512 to over 1280, which have been shown to capture richer biological features and improve transferability [Bepler et al., 2021, Brandes et al., 2022]. A higher-dimensional embedding space might have reduced the risk of compressing diverse biological signals into too few variables. However, increasing dimensionality also increases memory usage and computational cost, both during training and in downstream applications. Given the resource limitations of this project, a smaller embedding size was chosen as a necessary compromise.

5.1.3 No embedding compression strategies

No embedding compression techniques were explored in this work. Even embeddings of 256 dimensions can become challenging to store and process at proteome scale. Methods such as principal

component analysis (PCA), pooling, or quantisation could have improved scalability. For example, [Johnson et al., 2017] demonstrated that product quantisation in FAISS [Douze et al., 2024] enables efficient similarity search across very large datasets, while other studies like [Sun et al., 2019] have shown that dimensionality reduction can preserve biological signal while reducing storage demands. Such approaches were not implemented here, as the project's primary aim was to demonstrate the feasibility of domain-level tokenisation rather than to optimise deployment efficiency.

5.1.4 Lack of interpretability mechanisms

The model does not incorporate interpretability methods, and the resulting embeddings therefore function largely as black boxes. Previous research has shown that techniques such as attention visualisation and attribution analyses can provide insight by aligning model representations with structural or functional properties of proteins [Vig et al., 2020]. Including such methods would have increased transparency but would also have required dedicated evaluation pipelines and substantial additional experimentation. Within the limited timeframe of the MSc, priority was placed on establishing and validating the modelling framework rather than on interpretability.

5.1.5 Narrow evaluation scope

Finally, the evaluation carried out in this study was intentionally narrow. The experiments focused on clustering and Gene Ontology consistency as proof-of-concept validation. Stronger evidence of utility could have been obtained by including established downstream benchmarks, such as ProteinGym for mutational fitness prediction [Notin, Pascal et al., 2023]. However, applying such benchmarks would have required extensive preparation, including task-specific data handling, the implementation of baseline comparisons, and more comprehensive evaluation pipelines. These requirements were judged to be outside the practical scope of the MSc project, which instead prioritised establishing and validating the tokenisation strategy.

5.2 Potential evaluation: GO function prediction

A natural way to evaluate the learned embeddings would be through protein function prediction using Gene Ontology (GO) annotations. This task is widely recognised as a benchmark because it provides a biologically meaningful measure of representation quality and allows direct comparison with existing approaches [Radivojac et al., 2013]. Carrying out such an evaluation would involve several stages, from preparing the data to interpreting the results, each of which plays an important role.

First, the data would need to be carefully prepared and annotated. This would involve collecting GO annotations from curated resources such as UniProt-GOA or QuickGO [Huntley et al., 2015], filtering entries by evidence codes, and propagating terms through the GO hierarchy to maintain consistency [Ashburner et al., 2000]. Equally important would be designing data splits that avoid leakage, for example by separating proteins from the same family or species, and addressing the imbalance between frequent and rare terms by using class weighting or applying frequency cut-offs. These steps are crucial to ensure that the training targets are reliable and that evaluation results are not artificially inflated.

Second, an appropriate model architecture would be required. A multi-label classification head could be added to the embeddings, with a sigmoid activation function to allow multiple terms per protein. Loss functions would need to reflect the multi-label and hierarchical nature of GO, with options such as binary cross-entropy, focal loss, or hierarchical losses designed for directed acyclic graphs [Lin et al., 2017, Giunchiglia et al., 2020]. One approach would be to begin with frozen embeddings in order to test their representational quality directly, followed by selective fine-tuning of the encoder to explore whether additional task-specific adjustment improves performance.

Third, the training process would need to be staged and carefully controlled. A common strategy is to begin by training only the classification head, later unfreezing parts of the encoder if required. Further techniques such as label smoothing and curriculum learning could help to stabilise training, with curriculum learning gradually introducing rarer GO terms. This approach helps to avoid catastrophic forgetting, maintains focus on the quality of the embeddings, and encourages more robust generalisation.

Fourth, evaluation would need to follow community standards so that results are directly comparable with prior studies. The Critical Assessment of Function Annotation (CAFA) challenges have established widely used metrics, including Fmax (the maximum F1 score) and Smin (semantic distance), along with precision, recall, and coverage [Zhou et al., 2019]. It would also be valuable to compute semantic similarity measures, such as those proposed by Resnik [Resnik et al., 1999], since predictions close to the correct GO term still carry biological meaning. Together, these metrics provide a comprehensive view of both predictive accuracy and biological plausibility.

Fifth, the evaluation would require baselines for comparison. Traditional sequence-similarity approaches such as BLAST or DIAMOND remain strong performers in CAFA challenges, while methods like DeepGOPlus [Kulmanov et al., 2020] provide widely used neural baselines. Embeddings from large-scale protein language models such as ESM or ProtT5 would also form important points of reference [Lin et al., 2023, Elnaggar et al., 2022]. Comparing against these methods would clarify whether domain-level tokenisation offers a tangible advantage over residue-level representations.

Finally, any rigorous evaluation would need to include biological validation. This would involve stratifying results across GO domains (molecular function, biological process, and cellular component), across label frequencies, and across different protein families. Detailed case studies of well-characterised proteins could also be performed, together with comparisons against outputs from established annotation tools. Such validation would not only highlight systematic strengths and weaknesses but would also provide interpretability and biological credibility for the predictions.

Chapter 6

Conclusion

This study investigated a method for protein representation learning based on domain-level tokenisation. Although, we cannot deny that the model was able to capture regularities in multi-domain architectures and to preserve aspects of biological function within its embeddings, further work will be needed to assess its performance on more demanding downstream tasks, such as protein function prediction or protein design, and to compare it rigorously against state-of-the-art baselines. Overall, the findings suggest that domain-aware tokenisation is a feasible and biologically meaningful alternative to residue-level modelling. The code is publicly available at: https://github.com/melogranoo/domainbert.

Bibliography

[Devlin et al., 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. https://aclanthology.org/N19-1423/

[Vaswani et al., 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010. https://doi.org/10.48550/arXiv.1706.03762

[Rives et al., 2021] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C.L. Zitnick, J. Ma, R. Fergus, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, Proc. Natl. Acad. Sci. U.S.A. 118 (15) e2016239118, https://doi.org/10.1073/pnas.2016239118 (2021).

[Brandes et al., 2022] Brandes, Nadav et al. "ProteinBERT: a universal deep-learning model of protein sequence and function." Bioinformatics (Oxford, England) vol. 38,8 (2022): 2102-2110. doi:10.1093/bioinformatics/btac020

[Paysan-Lafosse et al., 2022] Paysan-Lafosse T, Blum M, Chuguransky S, et al. InterPro in 2022. Nucleic Acids Research. 2023 Jan;51(D1):D418-D427. DOI: 10.1093/nar/gkac993. PMID: 36350672; PMCID: PMC9825450.

[UniProt] The UniProt Consortium , UniProt: the Universal Protein Knowledgebase in 2023, Nucleic Acids Research, Volume 51, Issue D1, 6 January 2023, Pages D523–D531, https://doi.org/10.1093/nar/gkac1052

[Sillitoe et al., 2021] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla S M Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, Mahnaz Abbasian, Sean Le Cornu, Su Datt Lam, Karel Berka, Ivana Hutařová Varekova, Radka Svobodova, Jon Lees, Christine A Orengo, CATH: increased structural coverage of functional space, Nucleic Acids Research, Volume 49, Issue D1, 8 January 2021, Pages D266–D273, https://doi.org/10.1093/nar/gkaa1079

[Andreeva et al., 2020] Antonina Andreeva, Eugene Kulesha, Julian Gough, Alexey G Murzin, The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures, Nucleic Acids Research, Volume 48, Issue D1, 08 January 2020, Pages D376–D382, https://doi.org/10.1093/nar/gkz1064

[Varadi et al., 2022] 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, Sameer Velankar, AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models, Nucleic Acids Research, Volume 50, Issue D1, 7 January 2022, Pages D439–D444, https://doi.org/10.1093/nar/gkab1061

[Andy M. Lau et al., 2024] Andy M. Lau et al. ,Exploring structural diversity across the protein universe with The Encyclopedia of Domains.Science386,eadq4946(2024).DOI:10.1126/science.adq4946

[Mistry et al., 2021] Mistry, Jaina et al. "Pfam: The protein families database in 2021." Nucleic acids research vol. 49,D1 (2021): D412-D419. doi:10.1093/nar/gkaa913

[Dotan et al., 2024] Edo Dotan, Gal Jaschek, Tal Pupko, Yonatan Belinkov, Effect of tokenization on transformers for biological sequences, Bioinformatics, Volume 40, Issue 4, April 2024, btae196, https://doi.org/10.1093/bioinformatics/btae196

[Ieremie et al., 2024] Ieremie I, Ewing RM, Niranjan M. Protein language models meet reduced amino acid alphabets. Bioinformatics. 2024 Feb 1;40(2):btae061. doi: 10.1093/bioinformatics/btae061. PMID: 38310333; PMCID: PMC10872054.

[Yu et al., 2019] L. Yu, D.K. Tanwar, E.D.S. Penha, Y.I. Wolf, E.V. Koonin, M.K. Basu, Grammar of protein domain architectures, Proc. Natl. Acad. Sci. U.S.A. 116 (9) 3636-3645, https://doi.org/10.1073/pnas.1814684116 (2019).

[Rao et al., 2021] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, Alexander Rives Proceedings of the 38th International Conference on Machine Learning, PMLR 139:8844-8856, 2021.

[Wright et al., 2015] Wright, Peter E, and H Jane Dyson. "Intrinsically disordered proteins in cellular signalling and regulation." Nature reviews. Molecular cell biology vol. 16,1 (2015): 18-29. doi:10.1038/nrm3920

[Rao et al., 2019] Rao, Roshan et al. "Evaluating Protein Transfer Learning with TAPE." Advances in neural information processing systems vol. 32 (2019): 9689-9701.

[Vig et al., 2020] Vig, Jesse Madani, Ali Varshney, Lav Xiong, Caiming Socher, Richard Rajani, Nazneen. (2020). BERTology Meets Biology: Interpreting Attention in Protein Language Models. 10.1101/2020.06.26.174417.

[Robin van der Lee et al., 2014] Robin van der Lee, Marija Buljan, Benjamin Lang, Robert J. Weatheritt, Gary W. Daughdrill, A. Keith Dunker, Monika Fuxreiter, Julian Gough, Joerg Gsponer, David T. Jones, Philip M. Kim, Richard W. Kriwacki, Christopher J. Oldfield, Rohit V. Pappu, Peter Tompa, Vladimir N. Uversky, Peter E. Wright, and M. Madan Babu Chemical Reviews 2014 114 (13), 6589-6631 DOI: 10.1021/cr400525m

[Lin et al., 2023] Zeming Lin et al. ,Evolutionary-scale prediction of atomic-level protein structure with a language model.Science379,1123-s1130(2023).DOI:10.1126/science.ade2574

[Geer et al., 2002] Geer LY, Domrachev M, Lipman DJ, Bryant SH. CDART: protein homology by domain architecture. Genome Res. 2002 Oct;12(10):1619-23. doi: 10.1101/gr.278202. PMID: 12368255; PMCID: PMC187533.

[Ashburner et al., 2000] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet. 2000 May;25(1):25-9. doi: 10.1038/75556. PMID: 10802651; PMCID: PMC3037419.

[Jumper et al., 2021] Jumper, J., Evans, R., Pritzel, A. et al. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589 (2021). https://doi.org/10.1038/s41586-021-03819-2

[Ferruz et al., 2022] Ferruz, N., Schmidt, S. Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. Nat Commun 13, 4348 (2022). https://doi.org/10.1038/s41467-022-32007-7

[Meier et al., 2021] Language models enable zero-shot prediction of the effects of mutations on protein function Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, Alexander Rives bioRxiv 2021.07.09.450648; doi: https://doi.org/10.1101/2021.07.09.450648

[Frazer et al., 2021] Frazer, Jonathan et al. "Disease variant prediction with deep generative models of evolutionary data." Nature vol. 599,7883 (2021): 91-95. doi:10.1038/s41586-021-04043-8

[Callaway et al., 2020] Callaway, Ewen. "The race for coronavirus vaccines: a graphical guide." Nature vol. 580,7805 (2020): 576-577. doi:10.1038/d41586-020-01221-y

[Lu et al., 2022] Lu, H., Diaz, D.J., Czarnecki, N.J. et al. Machine learning-aided engineering of hydrolases for PET depolymerization. Nature 604, 662–667 (2022). https://doi.org/10.1038/s41586-022-04599-z

[Ponting and Russell et al., 2002] Ponting, Chris P, and Robert R Russell. "The natural history of protein domains." Annual review of biophysics and biomolecular structure vol. 31 (2002): 45-71. doi:10.1146/annurev.biophys.31.082901.134314

[Yang et al., 2020] Lu, Y., Nelson, M. R. (2020). The Conserved Domain Database (CDD) is a freely available resource for the annotation of sequences with the locations of conserved protein domain footprints.

[Glanville et al., 2017] Glanville, Jacob et al. “Identifying specificity groups in the T cell receptor repertoire.” Nature vol. 547,7661 (2017): 94-98. doi:10.1038/nature22976

[Varadi et al., 2022] Varadi, Mihaly et al. “AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models.” Nucleic acids research vol. 50,D1 (2022): D439-D444. doi:10.1093/nar/gkab1061

[Bepler et al., 2021] Bepler, T., Berger, B. (2021). Learning the protein language: Evolution, structure, and function. Cell Systems, 12(6), 654–669. https://www.cell.com/cell-systems/fulltext/S2405-4712%2821%2900203-9?utm_source=chatgpt.com

[Radivojac et al., 2013] Radivojac, P., Clark, W., Oron, T. et al. A large-scale evaluation of computational protein function prediction. Nat Methods 10, 221–227 (2013). https://doi.org/10.1038/nmeth.2340

[Huntley et al., 2015] Rachael P. Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J. Martin, Claire O’Donovan, The GOA database: Gene Ontology annotation updates for 2015, Nucleic Acids Research, Volume 43, Issue D1, 28 January 2015, Pages D1057–D1063, https://doi.org/10.1093/nar/gku1113Rachael P. Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J. Martin, Claire O’Donovan, The GOA database: Gene Ontology annotation updates for 2015, Nucleic Acids Research, Volume 43, Issue D1, 28 January 2015, Pages D1057–D1063, https://doi.org/10.1093/nar/gku1113

[Resnik et al., 1999] Resnik, O. (1999) Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity and Natural Language. Journal of Artificial Intelligence Research, 11, 95-130. https://doi.org/10.1613/jair.514

[Kulmanov et al., 2020] Maxat Kulmanov, Robert Hoehndorf, DeepGOPlus: improved protein function prediction from sequence, Bioinformatics, Volume 36, Issue 2, January 2020, Pages 422–429, https://doi.org/10.1093/bioinformatics/btz595

[Elnaggar et al., 2022] Elnaggar, Ahmed et al. “ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning.” IEEE transactions on pattern analysis and machine intelligence vol. 44,10 (2022): 7112-7127. doi:10.1109/TPAMI.2021.3095381

[Hie et al., 2022] Hie, Brian L et al. “Evolutionary velocity with protein language models predicts evolutionary dynamics of diverse proteins.” Cell systems vol. 13,4 (2022): 274-285.e6. doi:10.1016/j.cels.2022.01.003

[Johnson et al., 2017] Johnson, Jeff Douze, Matthijs Jégou, Hervé. (2017). Billion-Scale Similarity Search with GPUs. IEEE Transactions on Big Data. 7. 10.1109/TBDATA.2019.2921572.

[Lin et al., 2017] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, ”Focal Loss for Dense Object Detection,” 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324. keywords: Detectors;Training;Entropy;Object detection;Proposals;Robustness;Computer vision,

[Giunchiglia et al., 2020] Giunchiglia, Eleonora Lukasiewicz, Thomas. (2020). Coherent Hierarchical Multi-Label Classification Networks. 10.48550/arXiv.2010.10151.

[Zhou et al., 2019] Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., ... Radivojac, P. (2019). The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. Genome Biology, 20(1), 244. https://doi.org/10.1186/s13059-019-1835-8

[Sun et al., 2019] Sun, S., Zhu, J., Ma, Y. et al. Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. Genome Biol 20, 269 (2019). https://doi.org/10.1186/s13059-019-1898-6

[Douze et al., 2024] Douze, Matthijs, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazar'e, Maria Lomeli, Lucas Hosseini and Herv'e J'egou. "The Faiss library." https://doi.org/10.48550/arXiv.2401.08281

[Notin, Pascal et al., 2023] Notin, Pascal et al. "ProteinGym: Large-Scale Benchmarks for Protein Design and Fitness Prediction." bioRxiv : the preprint server for biology 2023.12.07.570727. 8 Dec. 2023, doi:10.1101/2023.12.07.570727.