Computational design of target-specific linear peptide binders with TransformerBeta

Haowen Zhao[1,†,*], Francesco A. Aprile[2], and Barbara Bravi[1,*]

[1] *Department of Mathematics, Imperial College London, London SW7 2AZ, United Kingdom*
[2] *Department of Chemistry and Institute of Chemical Biology, Molecular Sciences Research Hub, Imperial College London, London W12 0BZ, United Kingdom*
[†] *Current affiliation: Department of Chemistry, University of Cambridge, Cambridge CB2 1EW, United Kingdom*
[*] *For correspondence: hz362@cam.ac.uk, b.bravi21@imperial.ac.uk*

Abstract

The computational prediction and design of peptide binders targeting specific linear epitopes is crucial in biological and biomedical research, yet it remains challenging due to their highly dynamic nature and the scarcity of experimentally solved binding data. To address this problem, we built an unprecedentedly large-scale library of peptide pairs within stable secondary structures (beta sheets), leveraging newly available AlphaFold predicted structures. We then developed a machine learning method based on the Transformer architecture for the design of specific linear binders, in analogy to a language translation task. Our method, TransformerBeta, accurately predicts specific beta strand interactions and samples sequences with beta-sheet-like molecular properties, while capturing interpretable physico-chemical interaction patterns. As such, it can propose specific candidate binders targeting linear epitope for experimental validation to inform protein design.

1 Introduction

Peptides have emerged as important tools for fundamental and applied research in protein science and therapeutics. Linear peptide-like binding interfaces are found in many protein-protein interactions involved in cell signalling and regulation [1, 2, 3]. Targeting these regions is of great biomedical interest in the treatment of numerous human pathologies, including neurodegenerative diseases [4, 5, 6] and cancer [7]. Peptides can bind to epitopes with high affinity and specificity, while exhibiting lower immunogenicity and more cost-effective production than large biologics like proteins and antibodies [8, 9, 10].

The design of high-affinity linear peptides can be pivotal for epitope-specific antibody design [11]. Fragment-based approaches, which design these linear peptides purely by joining interacting protein-protein fragments and grafting them onto antibody scaffolds, have achieved low-nanomolar affinity binders without *in vitro* affinity maturation [11]. However, the applicability of fragment-based methods to new or less well-characterized targets is limited, as the epitope must already exist in the database, or interacting partners for its short subfragments must be identified and joined using custom rules. Machine learning-based generative models have also shown remarkable potential in designing binding peptides [12, 13, 14] and antibodies [15, 16, 17, 18, 19]. The state-of-the-art method, RFdiffusion, has achieved considerable success in designing medium-sized binding peptides, as validated by a large array of experiments [13]. However, RFdiffusion does not focus on designing short linear peptides, and there is room for improvement in the success rate at directly designing binding antibodies [16].

Motivated by recent advances in natural language processing, in this paper we propose that training generative language models on protein fragments to learn their complex dependencies could be an effective tool for modelling linear epitope-specific peptide binding, and thus for guiding the rational design of peptide binders and antibodies. These models, first designed for machine translation, have demonstrated broad applicability in generating sequences across many

domains [20, 21, 22] and in various protein-related tasks, including identifying epitopes [23, 24], learning interaction motifs at the paratope-epitope interface [25], designing functional proteins [26, 27] and peptides with desired biological activities [28, 29, 30, 31, 32].

A major challenge in machine learning-based peptide design is data acquisition, particularly in obtaining a training dataset representative of the complex interaction motif space. The release of the Alphafold Protein Structure Database (Alphafold DB) [33], containing 214 million structures predicted by AlphaFold2 [34], has provided an unprecedented, large-scale new set of protein structures that can serve many research purposes, from protein design [35, 36] and characterization [37, 14] to training dataset augmentation [38, 39, 40].

In this work, we created AlphaFold 2 Beta Strand Database, hosting 488 million high-quality beta strand interaction motifs collected from Alphafold DB for training data augmentation, since large datasets are crucial to train language models. We then trained a Transformer-based model, that we refer to as TransformerBeta, to predict probabilistic scores of peptide binding to linear epitopes. Based on these scores, TransformerBeta efficiently generates putative peptide binders, with the associated scores useful in candidate selection. We demonstrated that the designed peptides are highly similar to natural interaction motifs, both statistically and physico-chemically, and that the embedding of our model captures biologically meaningful representations.

2 Results

2.1 AlphaFold 2 Beta Strand Database

We constructed a large-scale database of sequence pairs sampling diverse beta-strand interaction motifs, building upon the release of Alphafold DB [33] (Methods). We call this database AlphaFold 2 Beta Strand Database, which stores 488 million distinct beta strand pairs (Fig. 1A). It shows a higher occurrence of antiparallel pairs, three times as many as parallel pairs (Fig. 1B), which are potentially relevant to peptide design strategies due to stronger inter-strand stability and geometric planarity. The database contains data of significant diversity, with 97.7% pairs of target-binder sequences showing less than 20% similarity, as calculated by the normalized Hamming distance (Figs. 1D, 6, Supp. Methods 5.1). Constructing large and diverse dataset is essential for training a deep language model, as machine translation performance was found to improve consistently with increasing dataset size [41]. This database is potentially useful for studying beta strand interactions in general and setting benchmarks for new prediction methods.

Despite the abundance and diversity of motifs in this database, especially at shorter lengths (Fig. 1B), directly searching for matched epitope sequences and using the interaction partner as a binder candidate is often not a viable strategy. For a typical length 8 target, fewer than an average of 1.5 potential binders are available per target (Fig. 1C). This highlights the need for extrapolating a general probability distribution over peptide sequence binding pairs through machine learning to deliver a broadly applicable design strategy and retrieve molecular patterns.

2.2 Deep learning model – TransformerBeta

To develop our machine learning design strategy, we formulated the peptide binder prediction problem as a machine translation task of beta strand interaction pairings, where one strand mimics a linear epitope target, while the other acts as a potential interacting binder, and we performed it through the Transformer architecture [42], which is the foundation for the majority of current language models. This choice builds upon the success of recent generative approaches for protein domain sequences [43], protein-specific drug molecules [44] and signal peptide generation [32] based on casting the problem as a machine translation task and on the Transformer architecture.

For computational feasibility, we implemented our strategy on a curated dataset of 2.1 million length 8 antiparallel beta strand pairs from the AlphaFold 2 Beta Strand Database (Methods). Length 8 is the typical epitope length, and the designed peptide of this length is suitable for grafting onto a majority of antibody scaffolds by replacing the amino acids of CDR3 regions.

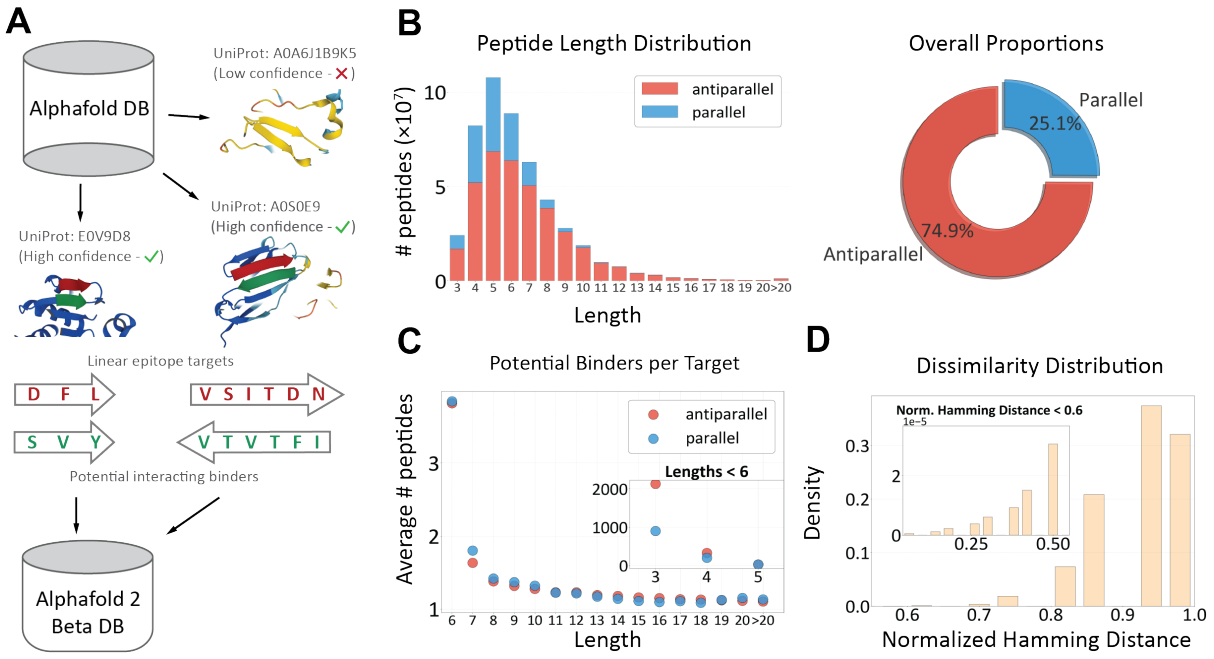

Figure 1: **AlphaFold 2 Beta Strand Database.** (**A**) Illustration of the collection of high-confidence beta strand pairs from Alphafold Protein Structure Database. Example structures (viewed with Mol* viewer [45]) show two high-confidence pairs (anti-parallel and parallel) and one pair that did not meet the high-confidence criteria. (**B**) Peptide length distribution of beta strand pairs with overall proportion of anti-parallel (74.9%) to parallel (25.1%) pairs. (**C**) Average number of potential binders available for each distinct target sequence. For clarity of visualization, lengths < 6 are plotted as an inset. (**D**) Pairwise dissimilarity distribution for anti-parallel length 8 data (a subset of 1,933,932 pairs) using the normalized Hamming distance. In **B-C**, pairs with lengths longer than 20 are grouped together for clarity.

However, our implementation is not constrained to peptides of length 8, allowing alignment-free training on varied input lengths. The Transformer model learns a probability distribution formulated autoregressively, which means that the probability of an amino acid at position i (y_i, with $i = 1, ..., m$) depends on the previous amino acids ($y_{<i}$) and the input target sequence (X) (Methods, Supp. Methods 5.2, Fig. 6). The probability of a target X-specific binder (Y) is the product of the likelihood of the individual amino acids y_i:

$$P_\theta(Y|X) = P_\theta(y_1, ..., y_m|X) = \prod_{i=1}^{m} P_\theta(y_i|X, y_{<i}) \tag{1}$$

We performed training on 90% of data by maximum likelihood, searched for optimal hyperparameters for a high-quality model on 5% and evaluated its performance on the remaining 5%. We then retrained the model with best hyperparameter settings on 100% data for optimal performance (Supp. Methods 5.3). As a result, we have TransformerBeta, a 6-layer encoder-decoder model with 44 million parameters (Fig. 2). Once evaluated on new peptides for a given target, TransformerBeta predicts a probability score, $P_\theta(Y|X)$, which represents the likelihood of the binder adopting a natural beta-strand conformation with the target, thus reflecting the peptide's quality as a target-specific binder. The designed binder could be used as a peptide candidate to carry on to additional computational and experimental tests in peptide and antibody design.

2.3 TransformerBeta accurately predicts target-specific binders

We tested the TransformerBeta's ability to accurately recover known binders for given target sequences. We assigned the TransformerBeta's probability score to target-binder pairs from the test set and from two negative-control datasets. In one of them, we coupled each target to random binders by concatenating amino acids uniformly at random (random set) to test the

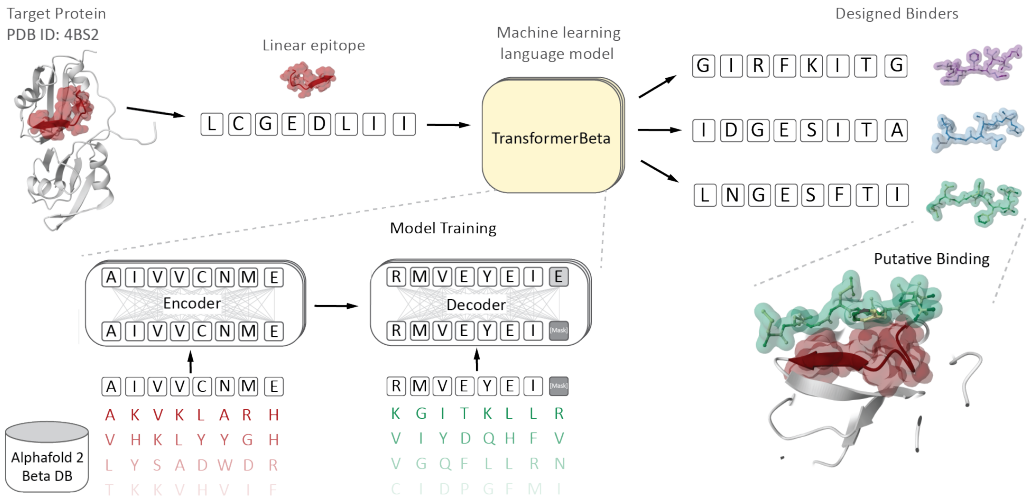

Figure 2: **Strategy for designing binders to target a linear epitope using TransformerBeta.** The target protein is shown in silver, with the epitope of interest highlighted in red. TransformerBeta takes as input the target sequence (from N to C terminus) and generates diverse target-specific linear peptides that are putative binders. The putative bound structure is a simulated docking pose using HPEPDOCK [46] and all protein structures are viewed using Mol* viewer [45].

model's ability to capture the typical amino acid usage of beta-strand binders. In the second one, we shuffled the target-binder pairings, obtaining a randomly shuffled set that maintains all statistical properties but loses its target-specific nature (shuffled set), to test whether the model's predictions preserve information on specificity (Methods). The scores' distribution (Fig. 3A) shows the model's ability to discriminate the test data of genuine binders from both the shuffled and random data sets, assigning to the former a higher probability of target specificity. We quantified the model's performance using the Receiver Operating Characteristic (ROC) curve, the Precision-Recall (PR) curve, and the corresponding areas under the curve (ROC-AUC and PR-AUC respectively). The model achieved a ROC-AUC of 0.98 and PR-AUC of 0.98 for random peptides and ROC-AUC of 0.95 and PR-AUC of 0.96 for shuffled peptides, as shown in Fig. 3B-C, confirming that the model accurately distinguishes binders from non-binders.

We further tested if TransformerBeta can well predict binders for *unseen* target sequences. When constructing the test set, we specifically ensured that the test set contained no target-binder pair identical to those in the training set. Fig. 3D shows the performance on this test set, stratified by the closest Hamming distance of the test target relative to the training set target sequences: it stays high for unseen targets with distance 1 from training targets, with a decrease beyond this distance that however kept ROC-AUC>0.8 for distance 2 and ROC-AUC>0.65 for distances 3-4. The performance showed a similar pattern for distances between training and test binders (Fig. 3E). Generalization performance is expected to become increasingly difficult with higher distances between training and test set, as is well documented and typically controlled for in applications of ML to protein data [47, 40, 48, 49, 50]. The prediction of immune receptors specific binding for unseen linear peptide targets is an extremely challenging and far from being solved [51, 52, 25]. In this context, for instance, it is already useful to achieve good predictive power for targets harbouring one mutation compared to the available data (like in the case of cancer neoantigens or viral single-point mutant epitopes); similarly, on the paratope side, 2-3 mutations can be sufficient to determine target specificity [53, 51, 54]. For a fixed target, we also observed a trade-off performance-wise in terms of the heterogeneity of the corresponding binders in the training data (Fig. 8J-L), as it has been assessed in other contexts of statistical modelling of protein sequences [55, 56]: target-specific training binders should be heterogeneous enough to sample their potential diversity, but not too divergent to preserve the necessary functionally-relevant statistical information. Such a result could help optimize training dataset construction from our AlphaFold 2 Beta Strand Database for specific targets. Finally, the model

prioritizes binding pairs that appear frequently in beta strands and that have the potential to be promiscuous binders (Supp. Methods 5.4, Fig. 8A-C, G-I).

2.4 TransformerBeta generates peptides similar to natural peptides

For the peptide design task, it is key that the model generated sequences exhibit biophysical and functional properties akin to natural ones, ensuring *e.g.* high binding affinity and stability. We generated peptides with TransformerBeta (generated set) and compared them to a set of natural binders (test data binders of Section 4.2, referred to as the natural set) and a set of randomly concatenated amino acids (random set). First, to ensure that our generated peptides were not simply replicates of natural ones, we measured the Hamming distances between them for the same targets. We found an average Hamming distance of 4.6 with length 8 sequences, suggesting that the generated sequences were mostly novel.

We projected a set of natural, generated and random peptides on a two-dimensional space using t-SNE [57](Fig. 4A). We observed a clear similarity in overall distribution between generated and natural sequences, but not with random ones, indicating that the model captures the organization of natural sequences in sequence space. Furthermore we found that generated sequences accurately reproduce the amino acid frequencies and correlations of natural sequences (Supp. Methods 5.5, Fig. 9), which is a fundamental test of the model's generative capacity [58, 59, 60, 61].

Finally, we focused on the physicochemical properties known to influence peptide stability and molecular interactions [62, 63]. By comparing the cumulative distribution functions (CDF) of five physicochemical properties (Net charge, Hydrophobicity, Molecular weight, Isoelectric point and Aromaticity) across different data groups, we observed that generated and natural peptides have a substantial overlap in terms of these properties, while being clearly distinct from random sequences (Fig. 4B). The differences in these properties' distribution between natural and generated peptides are not statistically significant in four out of five cases (p-value > 0.05, Kolmogorov's D-statistic tests, Supp. Methods 5.6), with a slight deviation only in aromaticity. We specifically noted that TransformerBeta emulates the hydrophobicity distribution of natural peptides, which is substantially higher than that of random sequences (Fig. 4B). This is expected since beta sheets often form the hydrophobic core of proteins, but it implies the model would design binders with hydrophobic tendencies potentially compromising their solubility, which should be additionally screened by tailored computational methods [64, 65, 66].

2.5 Model's biological interpretability

Language models have been shown to learn representations that recapitulate interpretable biological information and capture key protein features, including structure, function, interactions and evolution [67, 68, 69, 19]. To check whether the learnt representations of TransformerBeta are biologically sensible, we first studied its input embedding layer, a dictionary of 20 learnable amino acid vectors shared across the encoder and decoder (Fig. 5A). We projected the high-dimensional embedding vectors onto a two-dimensional space using t-SNE to visualize the distribution of amino acids, annotating them by the main physico-chemical property, including charge, polarity, hydrophobicity and aromaticity (Fig. 5A). The trained embedding displayed clear clustering patterns corresponding to such groups, demonstrating the model has captured biologically meaningful representations in the embedding space, similar to other language models [67, 68]. We compared the TransformerBeta's embedding layer with the amino acid substitution propensities summarized by the BLOSUM62 matrix [70] (reflecting physico-chemical similarities), and with the learnt embeddings of established pretrained protein language models [68, 71], finding a relatively high degree of correlation with all of them (Figs. 5B, 10, Table 3, Supp. Methods 5.7).

Finally, we assessed if our model has learned features specific to our beta strand data, beyond general amino acid properties. We extracted all cross-attention maps for a set of 1,000 target-binder pairs randomly sampled from the training data (Supp. Methods 5.2). These attention values reflect the relevance of each amino acid in the target for the prediction, and hence the

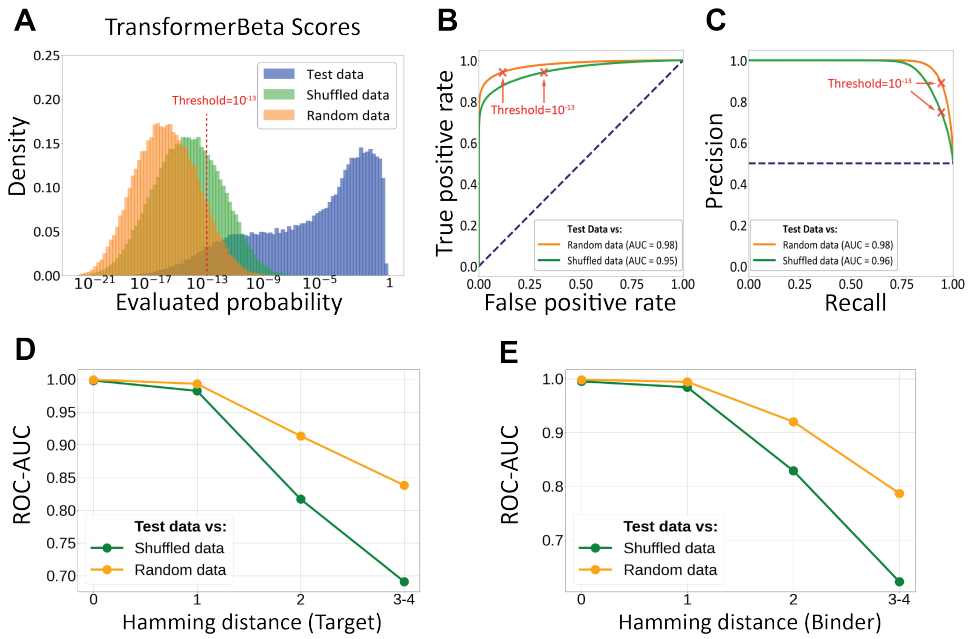

Figure 3: **Model's prediction accuracy.** (**A**) Distribution of the TransformerBeta predicted probability assigned to binders in the test, shuffled, and random sets, each containing 107,441 sequence pairs. (**B**) Receiver Operating Characteristic (ROC) curve and corresponding Area Under the Curve (ROC-AUC). (**C**) Precision-Recall (PR) curve and corresponding Area Under the Curve (PR-AUC). The dashed line gives the performance of a random classifier (ROC-AUC=PR-AUC=0.50). ROC-AUC as a function of the closest Hamming distance between test and training targets (**D**) and between test and training binders (**E**). Hamming distances of 4, having fewer than 5 data points, are grouped with distance 3.

design, of each amino acid in the binder. In average, attention values are mostly concentrated along the diagonal (Fig. 5C): when predicting the next amino acid of the binder, the model leverages maximally the information from the facing residue along the target, in line with the fact that hydrogen bonds between facing residues are the key interactions in beta strands [72].

3 Conclusion

In this paper, we built upon recent progress in machine learning architectures for language translation and text generation to train a generative model of linear peptide binders (TransformerBeta) on a dictionary of beta strand pairs. To obtain a more exhaustive sampling of such interaction motifs for the model's training, we leveraged the predicted structures made available by the Alphafold Protein Structure Database [33]. We have shown that TransformerBeta recovers with high accuracy complementary beta strands, and that by sampling from the learnt distribution one can generate novel candidate peptides with beta strand characteristics that resemble natural ones statistically and physico-chemically. The generative power of TransformerBeta could be exploited to design high-affinity beta-strand-like binders specific to pre-selected linear epitopes. We found that TransformerBeta learns representations recapitulating beta strand-specific binding modes (like the presence of hydrophobic face-to-face bonds) and general amino acid properties.

We provided proof-of-concept of our TransformerBeta design strategy for peptide pairs of fixed length, nevertheless future efforts could improve its ability to generalize to varying lengths by exploiting more data from our AlphaFold 2 Beta Strand Database, albeit at the cost of increased computational power. Systematic comparisons with existing peptide design methods and computational pipelines [11, 35, 56] will be needed in connection to specific design applications. Our beta strand database could also be further expanded with new predicted structures becoming available, *e.g.*, following the release of AlphaFold3 [73].

Converting structural motifs into a library of interacting sequence fragments gave the advantage of building an approach that is sequence-based, hence computationally cheaper, making it

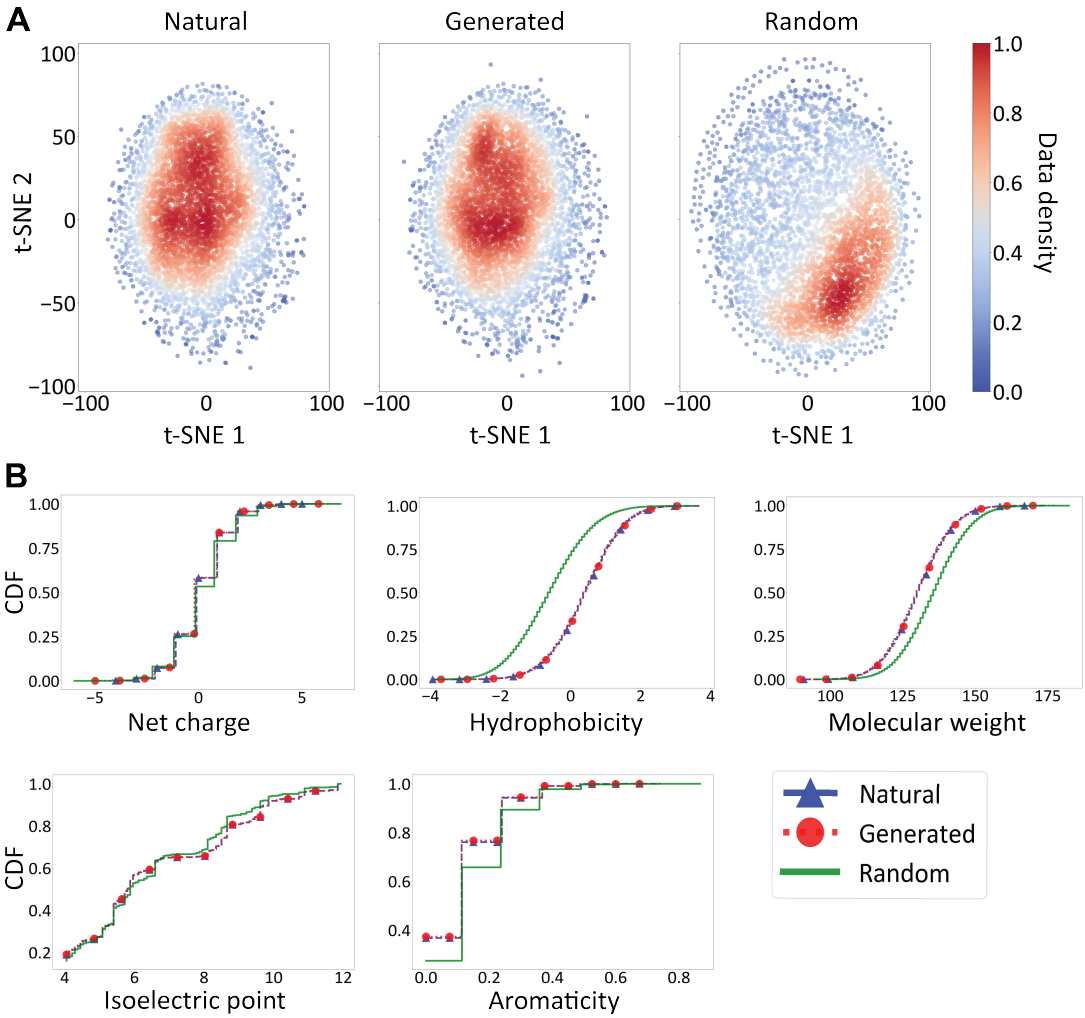

Figure 4: **Properties of generated data.** (**A**) t-SNE projected distributions of 5,000 randomly sampled binders from natural set, the model generated set and random set. (**B**) Cumulative Distribution Function (CDF) of various physicochemical properties (Net charge, Hydrophobicity, Molecular weight, Isoelectric point, Aromaticity) for the same natural, model generated, and random binders as in (**A**).

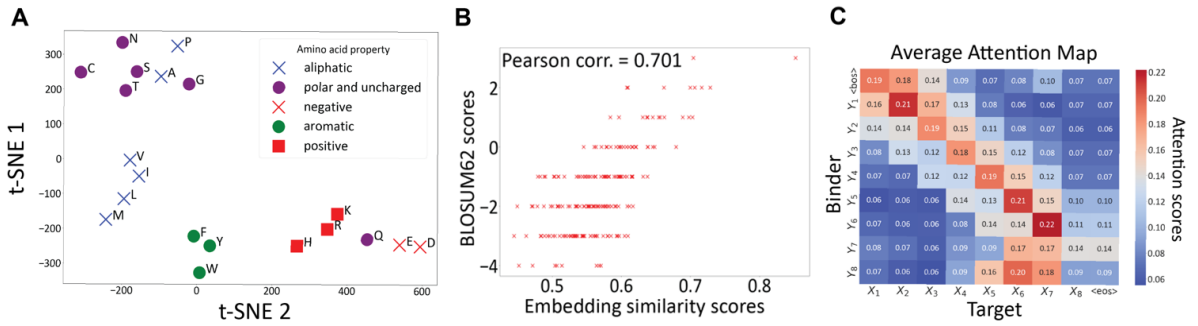

Figure 5: **Interpretability of TransformerBeta.** (**A**) Input embedding shared across encoder and decoder. (**B**) Scatter plot comparing embedding cosine similarity scores and BLOSUM62 substitution scores (Supp. Methods 5.7). (**C**) Average cross-attention map. X_i and Y_i represents the i^{th} amino acid of target and binder respectively. <bos> and <eos> are two special tokens.

useful to score or generate large numbers of putative binders in a short time. TransformerBeta could be used to select a few high-quality starting sequences for in vitro optimization of ligands, or to generate epitope-specific peptide libraries for experimental screening to accelerate drug discovery. It could be flexibly coupled with functional motif scaffolding methods for the design of a binding protein, *e.g.* in antibody engineering, where designed peptides are grafted onto the antigen-binding region of an antibody scaffold. Finally, it could serve as baseline model of generic beta-sheet-type interactions to train via transfer learning models of specific types of interactions with linear targets and ligands, for instance some immune epitope-paratope interactions [74, 16].

4 Methods

4.1 Creation of AlphaFold 2 Beta Strand Database

We developed the AlphaFold 2 (AF2) Beta Strand Database to store high-confidence scored beta strand pairs predicted by Alphafold 2 [34]. We downloaded all 214 million protein structures from the Alphafold Protein Structure Database (Alphafold DB) [33] with data access dated 07/10/2022, available at `https://alphafold.ebi.ac.uk/`. We extracted the equal-length pairs of amino acid sequences that face each other in a beta-strand conformation with the following criteria: (i) Both sequence lengths are at least 3 residues; (ii) All residues have a pLDDT of at least 70; (iii) All residues that face each other have a PAE of less than or equal to 10. We obtained a total of 1,532,767,459 beta strand pairs from the Alphafold DB and reduced them to 488,153,783 unique beta strand pairs, each with a count value indicating its frequency in the original 1.5 billion dataset. The database is available at `https://huggingface.co/datasets/hz3519/AF2_Beta_Strand_Database`.

4.2 Dataset preparation for model training and evaluation

We retrieved 38,616,564 length 8 antiparallel beta strands from the AF2 Beta Strand Database. We removed data with counts less than 5, resulting in a final dataset with 2,148,813 pairs. The dataset was randomly split into training data, comprising 90% of the data (1,933,932 pairs), validation data, with 5% (107,440 pairs) and test data, with the remaining 5% (107,441 pairs). For evaluation, we generated additional control datasets, which we refer to as 'random' or 'shuffled'. Random binders were generated by uniformly sampling from the 20 amino acids at each position for each test target. Shuffled binders were obtained by permuting the order of true binders for the test set. Samples of 'generated' binders were generated by a probabilistic sampling strategy (Supp. Methods 5.3) for each test target. We constructed the final random, shuffled and generated sets, each containing 107,441 pairs, by pairing the binders with corresponding test targets.

4.3 Generative model for peptide design – TransformerBeta

Our aim was to learn the conditional probability distribution $P(Y|X)$ of natural beta strand space, where X is the target and Y is the binder. We treated the amino acids as discrete tokens and represented X and Y as sequences of tokens $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_m)$, respectively. Given that AF2 Beta Strand Database contains target and binder pairs of equal length, the lengths of the target and binder sequences are equal ($n = m$). We trained a standard Transformer [42] (Supp. Methods 5.2, 5.8), an autoregressive model that learns the probability $P_\theta(Y|X)$ given by Equation 1, then used for the beta-strand-like binders generation task (Supp. Methods 5.3). The parameters θ specifying the conditional probability distribution $P_\theta(Y|X)$ are learnt by maximizing the log-likelihood over the training data $\mathcal{D}_\mathcal{T}$, i.e., by finding θ^* such that:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{(X,Y)\in\mathcal{D}_\mathcal{T}} \sum_{i=1}^{m} \log P_\theta(y_i|X, y_{<i}) \tag{2}$$

The code is available at `https://github.com/HZ3519/TransformerBeta_project`.

Acknowledgments

FAA thanks UK Research and Innovation (Future Leaders Fellowship MR/S033947/1 and MR/Y003616/1) for support.

References

[1] Kim Van Roey, Bora Uyar, Robert J Weatheritt, Holger Dinkel, Markus Seiler, Aidan Budd, Toby J Gibson, and Norman E Davey. Short linear motifs: ubiquitous and functionally diverse protein interaction modules directing cell regulation. *Chemical reviews*, 114(13):6733–6778, 2014.

[2] Peter E Wright and H Jane Dyson. Linking folding and binding. *Current opinion in structural biology*, 19(1):31–38, 2009.

[3] Alexander Cumberworth, Guillaume Lamour, M Madan Babu, and Jörg Gsponer. Promiscuity as a functional trait: intrinsically disordered regions as central players of interactomes. *Biochemical Journal*, 454(3):361–369, 2013.

[4] Olaf Schweers, E Schönbrunn-Hanebeck, Alexander Marx, and Eckhard Mandelkow. Structural studies of tau protein and Alzheimer paired helical filaments show no evidence for beta-structure. *Journal of Biological Chemistry*, 269(39):24290–24297, 1994.

[5] Paul H Weinreb, Weiguo Zhen, Anna W Poon, Kelly A Conway, and Peter T Lansbury. NACP, a protein implicated in alzheimer's disease and learning, is natively unfolded. *Biochemistry*, 35(43):13709–13715, 1996.

[6] Vladimir N Uversky. A protein-chameleon: conformational plasticity of α-synuclein, a disordered protein involved in neurodegenerative disorders. *Journal of Biomolecular Structure and Dynamics*, 21(2):211–234, 2003.

[7] Bora Uyar, Robert J Weatheritt, Holger Dinkel, Norman E Davey, and Toby J Gibson. Proteome-wide analysis of human disease mutations in short linear motifs: neglected players in cancer? *Molecular BioSystems*, 10(10):2626–2642, 2014.

[8] Komal Sharma, Krishna K Sharma, Anku Sharma, and Rahul Jain. Peptide-based drug discovery: Current status and recent advances. *Drug Discovery Today*, page 103464, 2022.

[9] Lei Wang, Nanxi Wang, Wenping Zhang, Xurui Cheng, Zhibin Yan, Gang Shao, Xi Wang, Rui Wang, and Caiyun Fu. Therapeutic peptides: Current applications and future directions. *Signal Transduction and Targeted Therapy*, 7(1):48, 2022.

[10] Jasmine Davda, Paul Declerck, Siwen Hu-Lieskovan, Timothy P Hickling, Ira A Jacobs, Jeffrey Chou, Shahram Salek-Ardakani, and Eugenia Kraynov. Immunogenicity of immunomodulatory, antibody-based, oncology therapeutics. *Journal for immunotherapy of cancer*, 7:1–9, 2019.

[11] Pietro Sormanni, Francesco A Aprile, and Michele Vendruscolo. Rational design of antibodies targeting specific epitopes within intrinsically disordered proteins. *Proceedings of the National Academy of Sciences*, 112(32):9902–9907, 2015.

[12] Jiahan Li, Chaoran Cheng, Zuofan Wu, Ruihan Guo, Shitong Luo, Zhizhou Ren, Jian Peng, and Jianzhu Ma. Full-atom peptide design based on multi-modal flow matching. *arXiv preprint arXiv:2406.00735*, 2024.

[13] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, pages 1–3, 2023.

[14] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022.

[15] John Boom, Matthew Greenig, Pietro Sormanni, and Pietro Liò. Score-based generative models for designing binding peptide backbones. *arXiv preprint arXiv:2310.07051*, 2023.

[16] Nathaniel R Bennett, Joseph L Watson, Robert J Ragotte, Andrew J Borst, Déjenaé L See, Connor Weidle, Riti Biswas, Ellen L Shrock, Philip JY Leung, Buwei Huang, et al. Atomically accurate de novo design of single-domain antibodies. *bioRxiv*, 2024.

[17] Rahmad Akbar, Philippe A Robert, Cédric R Weber, Michael Widrich, Robert Frank, Milena Pavlović, Lonneke Scheffer, Maria Chernigovskaya, Igor Snapkov, Andrei Slabodkin, et al. In silico proof of principle of machine learning-based antibody design at unconstrained scale. In *MAbs*, volume 14, page 2031482. Taylor & Francis, 2022.

[18] Koichiro Saka, Taro Kakuzaki, Shoichi Metsugi, Daiki Kashiwagi, Kenji Yoshida, Manabu Wada, Hiroyuki Tsunoda, and Reiji Teramoto. Antibody design using LSTM based deep generative model from phage display library for affinity maturation. *Scientific reports*, 11(1):1–13, 2021.

[19] Jinwoo Leem, Laura S Mitchell, James HR Farmery, Justin Barton, and Jacob D Galson. Deciphering the language of antibodies using self-supervised learning. *Patterns*, 3(7), 2022.

[20] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[21] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[22] Viraj Bagal, Rishal Aggarwal, PK Vinod, and U Deva Priyakumar. MolGPT: molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9):2064–2076, 2021.

[23] Joakim Nøddeskov Clifford, Magnus Haraldson Høie, Sebastian Deleuran, Bjoern Peters, Morten Nielsen, and Paolo Marcatili. BepiPred-3.0: Improved b-cell epitope prediction using protein language models. *Protein Science*, 31(12):e4497, 2022.

[24] Yanyi Chu, Yan Zhang, Qiankun Wang, Lingfeng Zhang, Xuhong Wang, Yanjing Wang, Dennis Russell Salahub, Qin Xu, Jianmin Wang, Xue Jiang, et al. A transformer-based model to predict peptide–HLA class i binding and optimize mutated peptides for vaccine design. *Nature Machine Intelligence*, 4(3):300–311, 2022.

[25] Rahmad Akbar, Philippe A Robert, Milena Pavlović, Jeliazko R Jeliazkov, Igor Snapkov, Andrei Slabodkin, Cédric R Weber, Lonneke Scheffer, Enkelejda Miho, Ingrid Hobæk Haff, et al. A compact vocabulary of paratope-epitope interactions enables predictability of antibody-antigen binding. *Cell Reports*, 34(11), 2021.

[26] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[27] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[28] Francesca Grisoni, Claudia S Neuhaus, Gisela Gabernet, Alex T Müller, Jan A Hiss, and Gisbert Schneider. Designing anticancer peptides by constructive machine learning. *ChemMedChem*, 13(13):1300–1302, 2018.

[29] Alice Capecchi, Xingguang Cai, Hippolyte Personne, Thilo Köhler, Christian van Delden, and Jean-Louis Reymond. Machine learning designs non-hemolytic antimicrobial peptides. *Chemical science*, 12(26):9221–9232, 2021.

[30] Christina Wang, Sam Garlick, and Mire Zloh. Deep learning for novel antimicrobial peptide design. *Biomolecules*, 11(3):471, 2021.

[31] Javier Caceres-Delpiano, Roberto Ibañez, Patricio Alegre, Cynthia Sanhueza, Romualdo Paz-Fiblas, Simon Correa, Pedro Retamal, Juan Cristóbal Jiménez, and Leonardo Álvarez. Deep learning enables the design of functional de novo antimicrobial proteins. *BioRxiv*, pages 2020–08, 2020.

[32] Zachary Wu, Kevin K Yang, Michael J Liszka, Alycia Lee, Alina Batzilla, David Wernick, David P Weiner, and Frances H Arnold. Signal peptides generated by attention-based neural networks. *ACS Synthetic Biology*, 9(8):2154–2161, 2020.

[33] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

[34] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[35] Patrick Bryant and Arne Elofsson. EvoBind: in silico directed evolution of peptide binders with AlphaFold. *bioRxiv*, pages 2022–07, 2022.

[36] Casper Goverde, Benedict Wolf, Hamed Khakzad, Stéphane Rosset, and Bruno E Correia. De novo protein design by inversion of the AlphaFold structure prediction network. *Protein Science*, page e4653, 2023.

[37] You Yu, Shibai Li, Zheng Ser, Huihui Kuang, Thane Than, Danying Guan, Xiaolan Zhao, and Dinshaw J Patel. Cryo-EM structure of dna-bound smc5/6 reveals dna clamping enabled by multi-subunit conformational changes. *Proceedings of the National Academy of Sciences*, 119(23):e2202799119, 2022.

[38] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International Conference on Machine Learning*, pages 8946–8970. PMLR, 2022.

[39] Jeffrey A Ruffolo and Jeffrey J Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Biophysical Journal*, 121(3):155a–156a, 2022.

[40] Magnus Haraldson Høie, Frederik Steensgaard Gade, Julie Maria Johansen, Charlotte Würtzen, Ole Winther, Morten Nielsen, and Paolo Marcatili. DiscoTope-3.0: improved b-cell epitope prediction using inverse folding latent representations. *Frontiers in Immunology*, 15:1322712, 2024.

[41] Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, 2021.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[43] Barthelemy Meynard-Piganeau, Caterina Fabbri, Martin Weigt, Andrea Pagnani, and Christoph Feinauer. Generating interacting protein sequences using domain-to-domain translation. *Bioinformatics*, 39(7):btad401, 2023.

[44] Daria Grechishnikova. Transformer neural network for protein-specific de novo drug generation as a machine translation problem. *Scientific reports*, 11(1):321, 2021.

[45] David Sehnal, Sebastian Bittrich, Mandar Deshpande, Radka Svobodová, Karel Berka, Václav Bazgier, Sameer Velankar, Stephen K Burley, Jaroslav Koča, and Alexander S Rose. Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic acids research*, 49(W1):W431–W437, 2021.

[46] Pei Zhou, Bowen Jin, Hao Li, and Sheng-You Huang. HPEPDOCK: a web server for blind peptide–protein docking based on a hierarchical algorithm. *Nucleic acids research*, 46(W1):W443–W450, 2018.

[47] Jérôme Tubiana, Dina Schneidman-Duhovny, and Haim J Wolfson. ScanNet: an interpretable geometric deep learning model for structure-based protein binding site prediction. *Nature Methods*, 19(6):730–739, 2022.

[48] Xiangxin Zhou, Dongyu Xue, Ruizhe Chen, Zaixiang Zheng, Liang Wang, and Quanquan Gu. Antigen-specific antibody design via direct energy-based preference optimization. *arXiv preprint arXiv:2403.16576*, 2024.

[49] Xiangzhe Kong, Wenbing Huang, and Yang Liu. Conditional antibody design as 3d equivariant graph translation. *arXiv preprint arXiv:2208.06073*, 2022.

[50] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models. *bioRxiv*, pages 2022–07, 2022.

[51] Mathias Fynbo Jensen and Morten Nielsen. Enhancing tcr specificity predictions by combined pan-and peptide-specific training, loss-scaling, and sequence similarity integration. *Elife*, 12:RP93934, 2024.

[52] Pieter Meysman, Justin Barton, Barbara Bravi, Liel Cohen-Lavi, Vadim Karnaukhov, Elias Lilleskov, Alessandro Montemurro, Morten Nielsen, Thierry Mora, Paul Pereira, et al. Benchmarking solutions to the t-cell receptor epitope prediction problem: Immrep22 workshop report. *ImmunoInformatics*, 9:100024, 2023.

[53] Andreas Mayer and Curtis G Callan Jr. Measures of epitope binding degeneracy from t cell receptor repertoires. *Proceedings of the National Academy of Sciences*, 120(4):e2213264120, 2023.

[54] Chang Liu, Hong Lin, Limin Cao, Kaiqiang Wang, and Jianxin Sui. Research progress on unique paratope structure, antigen binding modes, and systematic mutagenesis strategies of single-domain antibodies. *Frontiers in Immunology*, 13:1059771, 2022.

[55] Lorenzo Posani, Francesca Rizzato, Rémi Monasson, and Simona Cocco. Infer global, predict local: quantity-quality trade-off in protein fitness predictions from sequence data. *bioRxiv*, pages 2022–12, 2022.

[56] Duccio Malinverni and M Madan Babu. Data-driven design of orthogonal protein-protein interactions. *Science signaling*, 16(774):eabm4484, 2023.

[57] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[58] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S. Marks, Chris Sander, Riccardo Zecchina, José N. Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[59] William P. Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, and Rama Ranganathan. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, 2020.

[60] Francisco McGee, Sandro Hauri, Quentin Novinger, Slobodan Vucetic, Ronald M Levy, Vincenzo Carnevale, and Allan Haldane. The generative capacity of probabilistic protein sequence models. *Nature communications*, 12(1):6302, 2021.

[61] Damiano Sgarbossa, Umberto Lupo, and Anne-Florence Bitbol. Generative power of a protein language model trained on multiple sequence alignments. *Elife*, 12:e79854, 2023.

[62] Michelle Felicia Lee and Chit Laa Poh. Strategies to improve the physicochemical properties of peptide-based drugs. *Pharmaceutical Research*, 40(3):617–632, 2023.

[63] Annette Bak, Dennis Leung, Stephanie E Barrett, Seth Forster, Ellen C Minnihan, Andrew W Leithead, James Cunningham, Nathalie Toussaint, and Louis S Crocker. Physicochemical and formulation developability assessment for therapeutic peptide delivery—a primer. *The AAPS journal*, 17:144–155, 2015.

[64] Christophe N. Magnan, Arlo Randall, and Pierre Baldi. SOLpro: Accurate sequence-based prediction of protein solubility. *Bioinformatics*, 25(17):2200–2207, 2009.

[65] Pietro Sormanni, Francesco A. Aprile, and Michele Vendruscolo. The CamSol Method of Rational Design of Protein Mutants with Enhanced Solubility. *Journal of Molecular Biology*, 427(2):478–490, 2015.

[66] Sameer Khurana, Reda Rawi, Khalid Kunji, Gwo-Yu Chuang, Halima Bensmail, and Raghvendra Mall. DeepSol: A deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 34(15):2605–2613, 2018.

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

[68] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[69] Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. BERTology meets biology: interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.

[70] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[71] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with TAPE. *Advances in neural information processing systems*, 32, 2019.

[72] Merridee A Wouters and Paul MG Curmi. An analysis of side chain interactions and pair correlations within antiparallel β-sheets: The differences between backbone hydrogen-bonded and non-hydrogen-bonded residue pairs. *Proteins: Structure, Function, and Bioinformatics*, 22(2):119–131, 1995.

[73] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, pages 1–3, 2024.

[74] Anna Weber, Jannis Born, and María Rodriguez Martínez. TITAN: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics*, 37(Supplement_1):i237–i244, 2021.

[75] Mary Phuong and Marcus Hutter. Formal algorithms for transformers. *arXiv preprint arXiv:2207.09238*, 2022.

[76] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 2022.

[77] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.

[78] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[79] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[80] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[81] Martin Popel and Ondřej Bojar. Training tips for the transformer model. *arXiv preprint arXiv:1804.00247*, 2018.

[82] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[83] Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. Subformer: Exploring weight sharing for parameter efficiency in generative transformers. *arXiv preprint arXiv:2101.00234*, 2021.

[84] Jack Kyte and Russell F Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology*, 157(1):105–132, 1982.

[85] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[86] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[87] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

[88] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704*, 2020.

[89] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[90] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[91] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature communications*, 9(1):2542, 2018.

[92] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[93] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

5 Supplementary Methods

5.1 Database dissimilarity measures

To estimate the dissimilarity in peptide composition of our dataset, we concatenated the sequence pairs and calculated the pairwise normalized Hamming distance between different concatenated sequences (each of length 16). The Hamming distance, denoted as $H(x, y)$, is defined as the number of positions at which the corresponding amino acids in sequences x and y are different. The normalized Hamming distance is given by the formula:

$$\text{Normalized Hamming distance} = \frac{H(x,y)}{16} \tag{3}$$

The sequence pairs within all datasets (training, validation, and test) were sufficiently diverse for model training and evaluation as the majority of sequence pairs exhibited less than 20% similarity (Figs. 1, 6).

5.2 Model architecture

To learn the autoregressive model of Equation 1, we employed the encoder-decoder neural network architecture known as Transformer, as illustrated in Fig. 7. We adopted the same architecture in the original paper [42], where both the encoder and the decoder are composed of a stack of N_L identical layers. The Transformer architecture, its training and performance have been revised in detail in several studies [75, 76, 77]. For the sake of a brief illustration, we describe the structure of a typical Transformer encoder layer containing two key components: a multi-head self-attention module followed by a position-wise fully connected feed-forward neural network. We omit the details of the decoder, as it closely resembles the encoder.

Before entering the encoder, the target sequence of amino acids $X = (x_1, ..., x_n)$ is first preprocessed by an input embedding transformation and positional encoding. Each amino acid is independently transformed by a learnable input embedding layer into a vector of dimension d_{model}. The embedded vectors are added by the sinusoidal positional encoding as in [42], which produces a matrix E of dimension $n \times d_{model}$. The transformation of input embedding layer is learnt during the training process and gives a representation of each amino acid as vector of continuous values. We quantify the correlation between two amino acids using embedding similarity, calculated as the cosine similarity between their embedding representations $\vec{a}$ and $\vec{b}$, given by the following formula:

$$\text{Embedding similarity}(\vec{a}, \vec{b}) = \text{Cosine similarity}(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}||_2 \cdot ||\vec{b}||_2} \tag{4}$$

The multi-head attention module then projects in parallel the embedded input E into h sets (or heads) of query (Q), key (K), and value (V) representations through learnable linear transformations. For each head $i = 1, \ldots, h$, the matrices Q, K, and V have dimensions $d_{model} \times d_q$, $d_{model} \times d_k$, and $d_{model} \times d_v$, respectively; here we chose $d_q = d_k = d_v = d_{model}/h$ as in [42]. The attention function for each of the h heads is computed as follows:

$$\text{Attention}^{(i)}(Q^{(i)}, K^{(i)}, V^{(i)}) = \text{softmax}\left(\frac{Q^{(i)}K^{(i)T}}{\sqrt{d_{model}/h}}\right)V^{(i)}, \qquad i = 1, \ldots, h \tag{5}$$

The $n \times n$ matrix $\text{softmax}\left(\frac{Q^{(i)}K^{(i)T}}{\sqrt{d_{model}/h}}\right)$ provides, for a given input sequence, the weights by which each token attends to the other tokens along the sequence, quantifying its relevance to their representation and prediction and hence reflecting the degree of statistical interdependence among tokens. The h attention outputs 5 (each of dimension $n \times d_{model}/h$) are concatenated to produce a multi-head attention output with dimension $n \times d_{model}$.

Next the position-wise feed-forward neural network applies, independently to each position of this multi-head attention output, two linear transformations with a ReLU activation in the middle. The first transformation projects the output to dimension $n \times d_{ff}$, while the second transformation projects it back to the original dimension $n \times d_{model}$, producing the final output of a Transformer encoder layer - a high-level protein feature representation vector $Z = (z_1, \ldots, z_n)$. The encoded representations Z then enter the cross attention module in the decoder, which operates similarly to Equation 5. In the cross attention module, the value (V) projections are derived from Z, while query (Q) and key (K) projections are derived from binder sequences.

During training, the model outputs a predicted probability distribution $P_\theta(y_i|X, y_{<i})$ over 20 amino acids at each decoding position i. This predicted probability is compared to the true amino acid at the same decoding position, which is represented by a categorical distribution $q(y_i|X, y_{<i})$ over the 20 amino acids, with probability 1 at the correct amino acid and 0 at others. The Transformer learns the parameter set θ^* that minimizes the categorical cross-entropy loss across all decoding positions in the training dataset $\mathcal{D}_\mathcal{T}$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \left(- \sum_{(X,Y)\in\mathcal{D}_\mathcal{T}} \sum_{i=1}^{m} \sum_{k\in\mathcal{S}} q(y_i = k|X, y_{<i}) \log P_\theta(y_i = k|X, y_{<i}) \right) \tag{6}$$

where k represents a specific amino acid within the 20-amino-acid set $\mathcal{S}$. The minimization of the cross-entropy loss aims to align the model's predicted amino acid frequencies (via P_θ) with the empirical ones (via q). This process is equivalent to maximizing the log-likelihood (Equation 2). As in [42], we used two regularization techniques to prevent overfitting during training: applying a dropout with probability P_{drop} to the output of each sub-layer and label smoothing ϵ_{ls} to the cross-entropy loss [78].

5.3 Model selection

We evaluated the performance of Transformer models with various hyperparameter configurations by changing the number of layers (N_L), the dimension of embedding (d_{model}), the dimension of the feed-forward layer (d_{ff}), the number of heads for the multi-head attention module (h), the dropout probability (P_{drop}), label smoothing (ϵ_{ls}), and the number of training steps (Table 1). We used Adam optimizer [79] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$ (the default choices by [42]). A cosine warmup schedule was utilized for the Adam optimizer, incorporating a linear increase warmup phase over 10,000 steps, for training stabilization [80, 81]. We implemented embedding sharing for encoding input embedding and decoder input embedding, which was shown to achieve similar (or better) performance while effectively reducing the model's parameters [82, 83]. We maintained a consistent batch size of 4096 and a learning rate of 0.004 across the hyper-parametric search.

The primary metric used for model selection was the average log-likelihood of the validation dataset $\mathcal{D}_\mathcal{V}$, defined as:

$$\mathcal{L}_\mathcal{V} = \frac{1}{|\mathcal{D}_\mathcal{V}|} \sum_{(X,Y)\in\mathcal{D}_\mathcal{V}} \log P_{\theta^*}(Y|X) \tag{7}$$

which measures the model's capability to assign high probabilities to unseen data. We supplemented this primary evaluation metric with additional metrics to ensure the quality of the selected model. To ensure the selected model would capture the statistical properties of beta strands, we calculated the single-site frequency and 2-point connected correlations for both generated and validation binders, monitoring the Mean Absolute Errors (MAE_1, MAE_2) on these quantities for each model (Supp. Methods 5.5, Fig. 9). To confirm the selected model's ability to capture the physicochemical properties of beta strands, we looked at the distributions of various physicochemical properties (charge, hydrophobicity, molecular weight, isoelectric point and aromaticity) for both generated and validation binders, and calculated the Kolmogorov's

D-statistic, representing the maximum absolute difference between two empirical distributions (Supp. Methods 5.6).

As a final model to generate results, we have selected Model M in Table 1, as it demonstrated the best performance in terms of $\mathcal{L}_{\mathcal{V}}$. Across two additional statistical metrics, this model proves to be the best performing in terms of MAE_1 and third best performance in terms of MAE_2. Furthermore, 4 out of 5 Kolmogorov-Smirnov property tests cannot be rejected at the 0.05 significance level, *i.e.*, the distribution of the corresponding physicochemical properties across the generated sequences is not significantly distinguishable from the one across the sequences of the validation set. We used Model M trained on 90% of the training data for subsequent validation analyses. For the generation of peptides, we retrained the parameter settings of Model M using the full dataset (2,148,813 pairs, Methods 4.2) for optimized performance.

To generate peptides for evaluation, we adopted a random sampling strategy that sequentially generating amino acids for the binder given a target. Amino acids are sampled from the learnt conditional distribution (1) until both the binder and target sequence attain equal length.

5.4 Additional assessment of model performance

In this section, we define the conditions informative about model performance assessed in Figs. 3 and 8.

1. "Hamming distance (Target)" represents the minimum Hamming distance target sequences in the test data and the closest training data.

2. "Hamming distance (Binder)" represents the minimum Hamming distance binder in the test data and the closest training data.

3. "Count" represents the frequency of occurrence of each test data point in the AF2 Beta strand database.

4. "Hamming distance (Concat.)" represents the minimum Hamming distance between concatenated sequences, by joining target sequences and binder sequences, in the test data and the closest training data.

5. "Promiscuity (Target)" represents the number of binders for a test target in the AF2 Beta Strand database.

6. "Promiscuity (Binder)" represents the number of targets for a test binders in the AF2 Beta Strand database.

7. "Average within distance (Target)" represents the average pairwise Hamming distance of binders for a test target in AF2 Beta Strand database.

8. "Average within distance (Binder)" represents the average pairwise Hamming distance of targets for a test binder in AF2 Beta Strand database.

We additionally monitored model performance against other metrics: 'Count' represents the frequency of target-binder pairs; 'Promiscuity' is given by the number of binders per target (and vice versa) and is a proxy for a peptide's versatility in binding; 'Average within distance' quantifies the sequence diversity of binding partners to the same peptide. In conclusion, we found that TransformerBeta predictions have higher accuracy for higher count, higher promiscuity and medium average within distance, as shown in Fig. 8, essentially reflecting the level of sampling of target-binder pairs in the training set. Beyond sampling, a space of binding partners too narrow or too heterogeneous leads to slightly less accurate predictions (Figs. 8J-L).

5.5 Validation on statistical properties

The statistical covariation between sequence positions encodes key evolutionary protein information and is highly relevant in generating functional synthetic sequences, as experimentally demonstrated by previous research [59]. We evaluated three statistical properties, namely the single-site amino acid frequency $f_i(a)$, *i.e.* the frequency of the amino acid a at position i along the sequence in the sample under consideration, the 2-point connected correlation $C_{ij}(a,b)$, and the 3-point correlation $C_{ijk}(a,b,c)$. The definitions are as follows. The two-point connected correlation of amino acids a and b at distinct positions i and j, respectively, is given by:

$$C_{ij}(a,b) = f_{ij}(a,b) - f_i(a)f_j(b) \tag{8}$$

where $f_{ij}(a,b)$ denotes the joint frequency of amino acids a and b occurring at positions i and j (where $i \neq j$), respectively. $C_{ij}(a,b)$ measures the degree to which the observed frequency of a and b appearing together deviates from what would be expected if their occurrences were independent.

The three-point correlation of amino acids a, b, and c at distinct positions i, j, and k, respectively, is given by:

$$C_{ijk}(a,b,c) = f_{ijk}(a,b,c) - f_i(a)f_{jk}(b,c) - f_j(b)f_{ik}(a,c) - f_k(c)f_{ij}(a,b) + 2f_i(a)f_j(b)f_k(c) \tag{9}$$

where $f_{ijk}(a,b,c)$ denotes the joint frequency of amino acids a, b, and c occurring at positions i, j, and k (where $i \neq j$, $i \neq k$, and $j \neq k$), respectively. It measures the degree to which the observed frequency of amino acids a, b, and c appearing together at positions i, j, and k deviates from what is expected if their occurrences were independent or only pairwise dependent. The three-point correlation provides a higher-order statistical description of the beta strand sequence space, capturing the intricate interdependence among the amino acid residues.

The Mean Absolute Errors (MAEs) between the original data statistics ($f_i(a)$, $C_{ij}(a,b)$, and $C_{ijk}(a,b,c)$) and the model predicted ones ($f'_i(a)$, $C'_{ij}(a,b)$, and $C'_{ijk}(a,b,c)$) are computed as follows:

$$MAE_1 = \frac{1}{m|\mathcal{S}|}\sum_{i=1}^{m}\sum_{a\in\mathcal{S}}|f_i(a) - f'_i(a)| \tag{10}$$

$$MAE_2 = \frac{2}{m(m-1)|\mathcal{S}|^2}\sum_{\substack{i=1,j=1\\ i\neq j}}^{m}\sum_{a,b\in\mathcal{S}}|C_{ij}(a,b) - C'_{ij}(a,b)| \tag{11}$$

$$MAE_3 = \frac{6}{m(m-1)(m-2)|\mathcal{S}|^3}\sum_{\substack{i=1,j=1,k=1\\ i\neq j, i\neq k, j\neq k}}^{m}\sum_{a,b,c\in\mathcal{S}}|C_{ijk}(a,b,c) - C'_{ijk}(a,b,c)| \tag{12}$$

where m is the total length of the sequence and $\mathcal{S}$ stands for the set of 20 amino acids. During model selection (Supp. Methods 5.3), we kept track of MAE_1 and MAE_2 between the statistics of the model generated set and that of the validation set of natural binders for each model (Methods 4.2). For the best-selected model M, we extended the examination to include MAE_3, verifying its effectiveness in capturing high-order beta strand statistics. As presented in Figs. 9B, D, F, the generated data exhibited a strong correlation with validation data at single-site frequency (MAE_1=0.00093, R^2=0.99871, Pearson correlation coefficient=0.99935), two-point correlation (MAE_2=0.00013, R^2=0.90624, Pearson correlation coefficient=0.95197) and three-point correlation (MAE_3=0.00003, R^2=0.46089, Pearson correlation coefficient=0.67889). We performed an additional correlation analysis comparing generated data with training data, which produced similar results (Figs. 9A, C, E). These outcomes imply that the chosen Model M has the capacity to generate samples that closely mirror the statistical properties of natural beta strands. It is worth noting that the training of Transformers does not rely on a moment-matching

procedure that constrains the moments of the learnt distribution to match the empirical moments (as is the case for single-site and pairwise frequencies in protein sequence modelling methods like Direct Coupling Analysis [58]). On the other hand, the Transformer learns probabilities of single sequence sites conditioned on the sequence context, which enables it to capture global correlations and hence to well reproduce high-order statistics [61].

5.6 Validation on physicochemical properties

We evaluated five physicochemical properties of peptide sequences:

1. Net charge ($Char$), which reflects the overall electric charge of a peptide. The net charge of each peptide was calculated by summing the charge of all amino acids in a peptide.

2. Hydrophobicity ($Hydro$), which indicates the preference of a peptide for nonpolar environments based on its constituent amino acids' reluctance to interact with water. The hydrophobicity of each peptide was calculated by averaging the hydrophobicity (using the Kyte-Doolittle scale [84]) of all amino acids in a peptide.

3. Molecular weight (MW), which denotes the aggregate mass of a peptide's constituent atoms. The molecular weight of each peptide was calculated by averaging the molecular weight of all amino acids in a peptide.

4. Isoelectric point (IP), which identifies the pH value at which a peptide exhibits a net charge of zero. The isoelectric point of each peptide was estimated using Bio.SeqUtils.IsoelectricPoint module in Biopython package [85].

5. Aromaticity ($Arom$), which measures the proportion of aromatic residues within a peptide. The aromaticity of each peptide was calculated by the relative frequency of aromatic amino acids (Phenylalanine, Tryptophan, Tyrosine) in a peptide.

During model selection (Supp. Methods 5.3), we computed the cumulative distribution functions (CDF) of these physicochemical properties of generated binders, comparing them to those in the validation and random sets for each model (Methods 4.2, Fig. 4). We first calculated the Kolmogorov's D-statistics (D_{Char}, D_{Hydro}, D_{MW}, D_{IP}, D_{Arom}) for each respective property between the natural and random sets. The respective computed D-statistics were 0.048427, 0.39769, 0.20989, 0.05868 and 0.10316, while the respective p-values were 6.35304×10^{-110}, 0 (less than representable positive number in python), 0, 2.96223×10^{-161} and 0, demonstrating a clear distinction between natural and random binders. We then calculated the Kolmogorov's D-statistic for each respective property between natural and generated sets (see Table 2). For the best selected model M, the computed D-statistics were 0.00338, 0.00300, 0.00332, 0.00443 and 0.00684, while the respective p-values were 0.571, 0.719, 0.592, 0.242 and 0.013. Four of the five tests can not be rejected at a 0.05 significant level, which implies no significant difference between these properties for generated and the natural binders. Based on these results, we can conclude that our selected model M accurately captures the physicochemical properties of beta strands, with a minor deviation in the case of aromaticity.

5.7 Embedding comparisons with protein language models

We calculated the embedding similarity between amino acids as in Equation 4, resulting in a 20 by 20 matrix. To avoid repeated values, we used the upper triangle of this matrix and the BLOSUM62 scores, then computed their Pearson correlation (Fig. 10, Fig. 5B).

By comparing the models' embeddings, we found that our model can learn amino acid representations comparable to protein language models with millions of parameters pretrained on millions of sequences, whose amino acid embedding similarity scores have a Pearson correlation with the BLOSUM62 scores ranging between ~ 0.42 (for ProtBert) and ~ 0.86 (for ProtXLNet), see Fig. 10.

5.8 Hardware

We conducted database construction, data preparation, training and analysis work on the Imperial High Performance Computing cluster. Our model implementation was mainly built on the open-source Pytorch library [86] and d2l package [87]. We employed parallel computing [88] to train the model (300K steps) on two RTX6000 GPUs over a duration of 50 hours.

6 Supplementary Tables and Figures

Model	N_L	d_{model}	d_{ff}	h	P_{drop}	ϵ_{ls}	train steps
base	6	512	2048	8	0.1	0.1	100K
A	6	512	2048	4	0.1	0.1	100K
B	6	512	2048	16	0.1	0.1	100K
C	2	512	2048	8	0.1	0.1	100K
D	4	512	2048	8	0.1	0.1	100K
E	8	512	2048	8	0.1	0.1	100K
F	6	256	2048	8	0.1	0.1	100K
G	6	768	2048	8	0.1	0.1	100K
H	6	512	2048	8	0.0	0.1	100K
I	6	512	2048	8	0.2	0.1	100K
J	6	512	2048	8	0.1	0.0	100K
K	6	512	2048	8	0.1	0.2	100k
L	6	512	2048	8	0.1	0.1	300K
M	6	512	2048	8	0.2	0.1	300K

Table 1: **Transformer architecture hyperparameters scanned for model selection.** Deviations from the original Transformer architecture from [42] (Model base) are emphasized in red.

Model	$\mathcal{L}_\mathcal{V}$	MAE_1 ($\times 10^{-5}$)	MAE_2 ($\times 10^{-5}$)	D_{Char} ($\times 10^{-5}$)	D_{Hydro} ($\times 10^{-5}$)	D_{MW} ($\times 10^{-5}$)	D_{IP} ($\times 10^{-5}$)	D_{Arom} ($\times 10^{-5}$)
base	-12.26	109.97	13.73	370	1246	890	501	506
A	-12.20	121.06	13.50	388	1075	677	483	496
B	-12.32	117.68	13.62	451	1269	876	587	533
C	-13.36	122.35	14.11	655	1617	759	746	320
D	-12.19	115.48	13.74	356	1484	552	407	706
E	-12.60	126.99	13.35	224	1239	760	454	533
F	-12.58	119.64	14.03	260	1561	944	385	257
G	-12.69	116.09	13.20	231	837	638	398	795
H	-16.70	209.84	15.38	243	2045	1151	810	252
I	-11.98	95.20	13.51	312	665	320	380	741
J	-13.20	128.11	13.28	356	1318	925	423	1354
K	-12.29	294.22	16.02	638	4090	2377	866	629
L	-12.65	111.71	13.34	396	934	648	581	679
M	-11.95	92.52	13.31	338	300	332	443	684

Table 2: **Transformer model selection**. Metrics for model selection are evaluated on the validation set (Methods 4.2). Best results according to log-likelihood, and quality of reproduction of statistical properties are emphasized in blue. We highlighted in red when the D-statistics of the Kolmogorov-Smirnov test suggests that the null hypothesis (H_0) cannot be rejected at a 0.05 significance level. H_0 posits that the validation and generated samples distributions are identical, while the alternative hypothesis (H_1) asserts that they differ. The selected Model M is enclosed within a red dashed box.

Model name	**Architecture**	**Number of Parameters**	**Dataset**	**Dataset Size**
ProtBert	BERT [89]	420M	UniRef100 [90]	216M
ProtBert-BFD	BERT	420M	BFD-100 [91]	2.1B
Protт5XL-BFD	T5 [92]	2.8B	BFD-100	2.1B
ProtXLNet	XLNET [93]	409M	UniRef100	216M
ProtAlbert	ALBERT [82]	224M	UniRef100	216M
TapeBert	BERT	92M	Pfam	31M

Table 3: Summary of pretrained protein language models used for input embedding comparison.

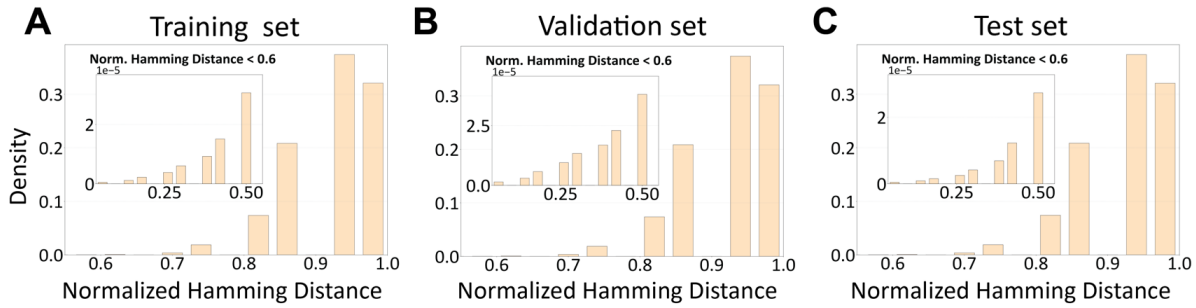

Figure 6: **Pairwise dissimilarity distributions for concatenated sequences in training, validation and test sets.** Distribution of normalized Hamming distance within: (**A**) a randomly sampled portion of the training set (10%, 193,393 sequences); (**B**) the validation set (107,440 sequences); (**C**) the test set (107,441 sequences). The score ranges between 0 and 1, with higher values indicating greater dissimilarity. For clarity of visualization, a zoom on the density for normalized Hamming distance < 0.6 is provided in the insets. Panel **D** of Fig. 1 is the training set distribution in panel **A** here.

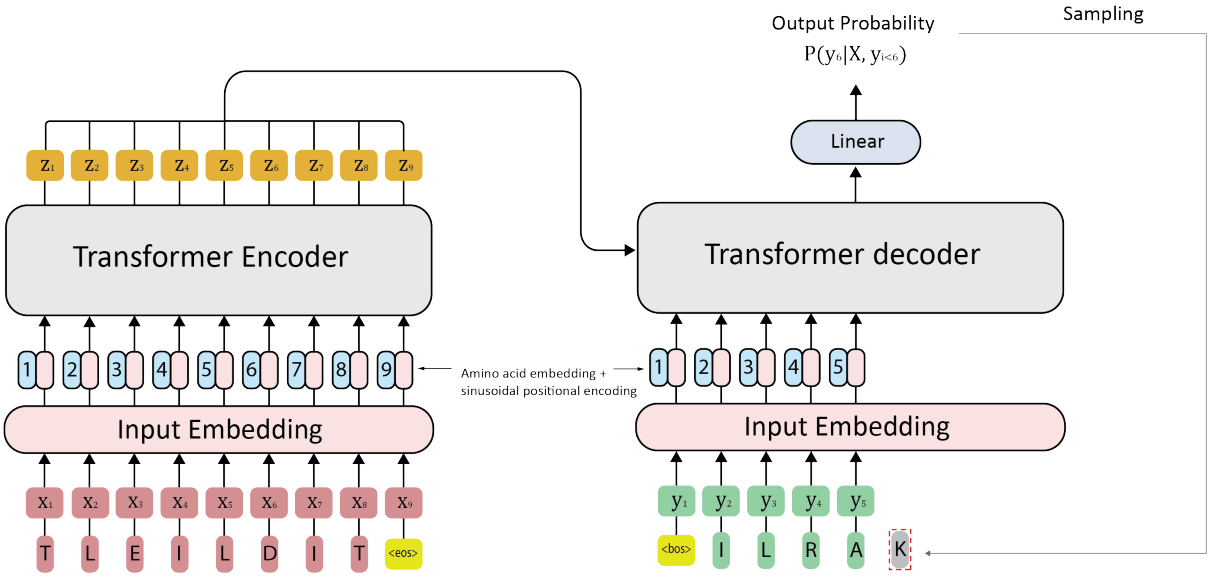

Figure 7: **TransformerBeta architecture and sampling strategy.** The figure depicts the generation of a binder "ILRAKVIL" for the target sequence "TLEILDIT" at the step of decoding symbol in position 6 along the sequence. Red tokens represent the target sequence, green tokens represent the decoded sequence for previous steps and gray token represents the next decoded token in this example. <eos> and <bos> are 2 special learnable tokens such that the decoding process starts with <bos> token and continues until the binder with a length equal to the target sequence is generated.

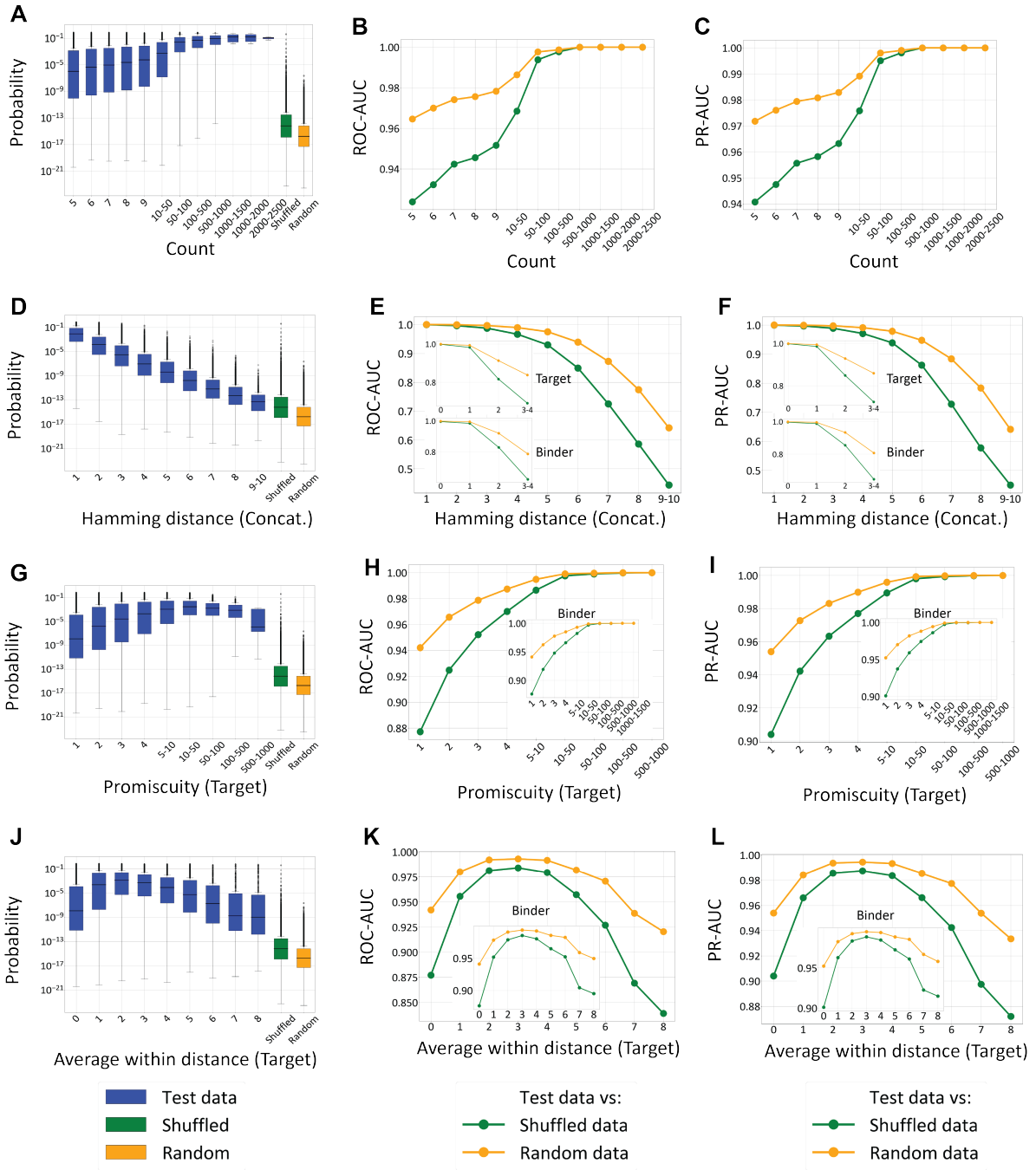

Figure 8: **TransformerBeta's ability to discriminate test data from shuffled and random data with varying factors**. The average Area Under the Receiver Operating Characteristic Curve (ROC-AUC) is plotted as a function of: (**B**) occurrence count of each test data point in AF2 Beta Strand database; (**E**) minimum Hamming distance between concatenated sequences (targets and binders as insets) in the test data and closest training data; (**H**) promiscuity score of test targets (binders as inset) when searched in AF2 Beta Strand database; (**K**) average pairwise Hamming distance of binders for each test target (targets for each binder as inset) when searched in AF2 Beta Strand database. Similar Area Under the Precision-Recall Curves (PR-AUC) are monitored in (**C**), (**F**), (**I**) and (**L**). The probability distributions generating the AUC plots are shown in (**A**), (**D**), (**G**) and (**J**).

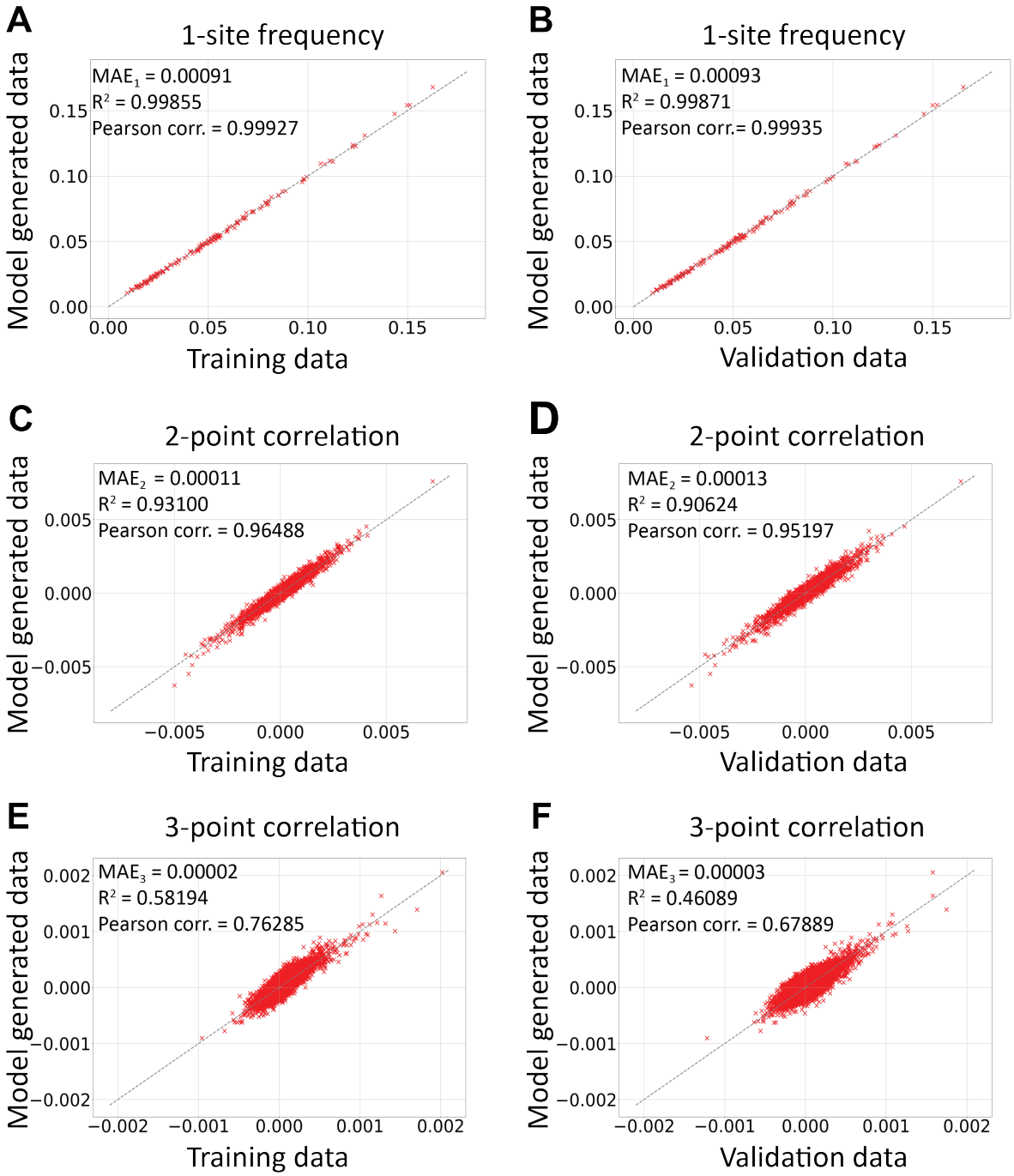

Figure 9: **Model's validation on statistical properties.** Comparison of 1-site frequency (**A-B**), 2-point connected correlations (**C-D**) and 3-point connected correlations (**E-F**) between model generated binders (107,440 sequences) and: training binders (1,933,932 sequences, **A**, **C**, **E**); validation binders (107,440 sequences, **B**, **D**, **F**). The model used to generate new samples is Model M (Table 1).

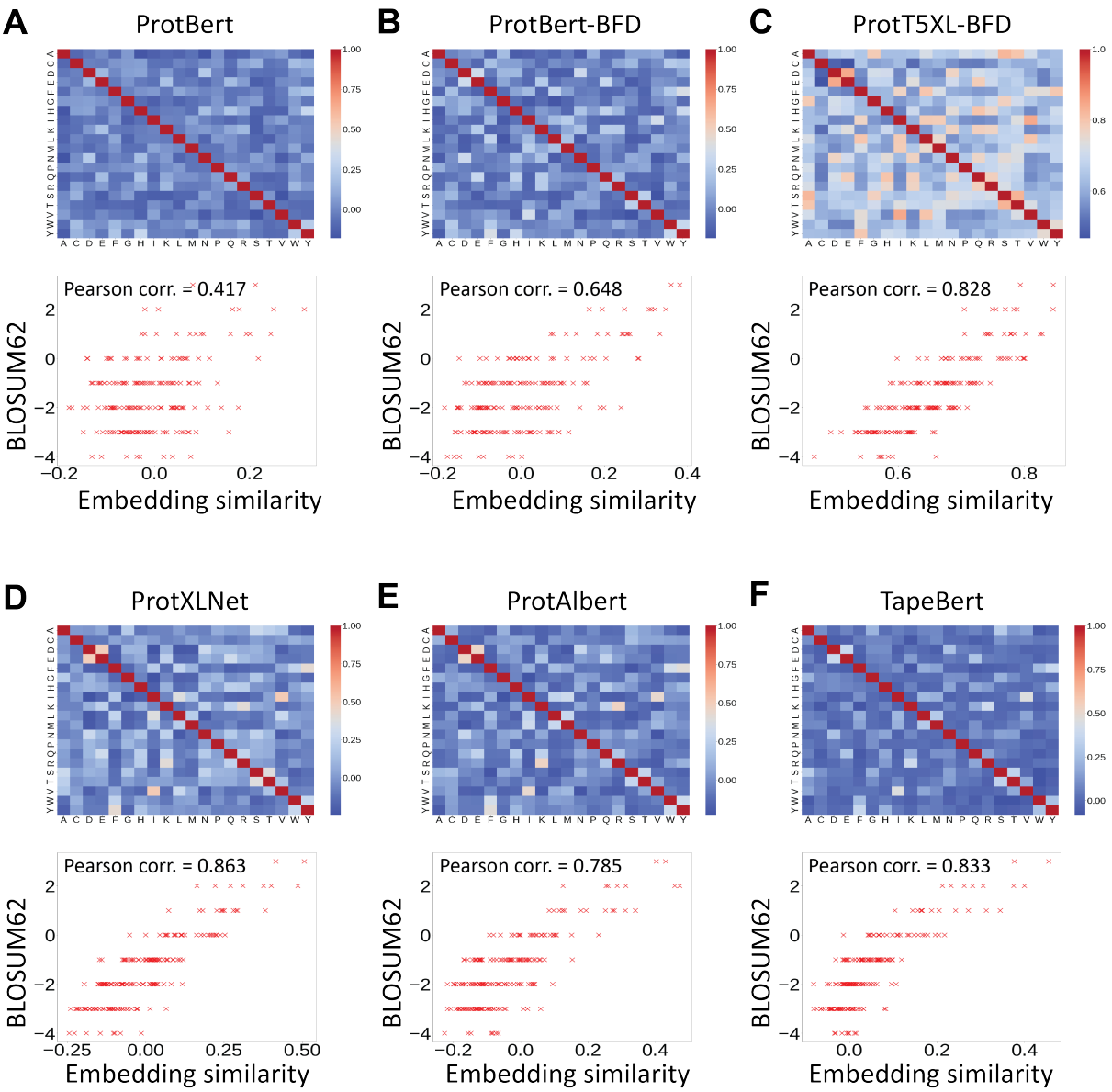

Figure 10: **Correlation between cosine similarity of input embedding and BLOSUM62 substitution matrix scores across 6 pretrained protein language models** (see Table 3). (**A**) ProtBert: a BERT architecture model trained on UniRef100 dataset. (**B**) ProtBert-BFD: a BERT architecture model trained on BFD-100 dataset. (**C**) ProtT5XL-BFD: a T5 architecture model trained on BFD-100 dataset. (**D**) ProtT5XLNet: a XLNET architecture model trained on UniRef100 dataset. (**E**) ProtAlbert: a ALBERT architecture model trained on UniRef100 dataset. (**F**) TapeBert: a BERT architecture trained on Pfam dataset.