

Supplementary Information: Protein Sequence Modelling with Bayesian Flow Networks

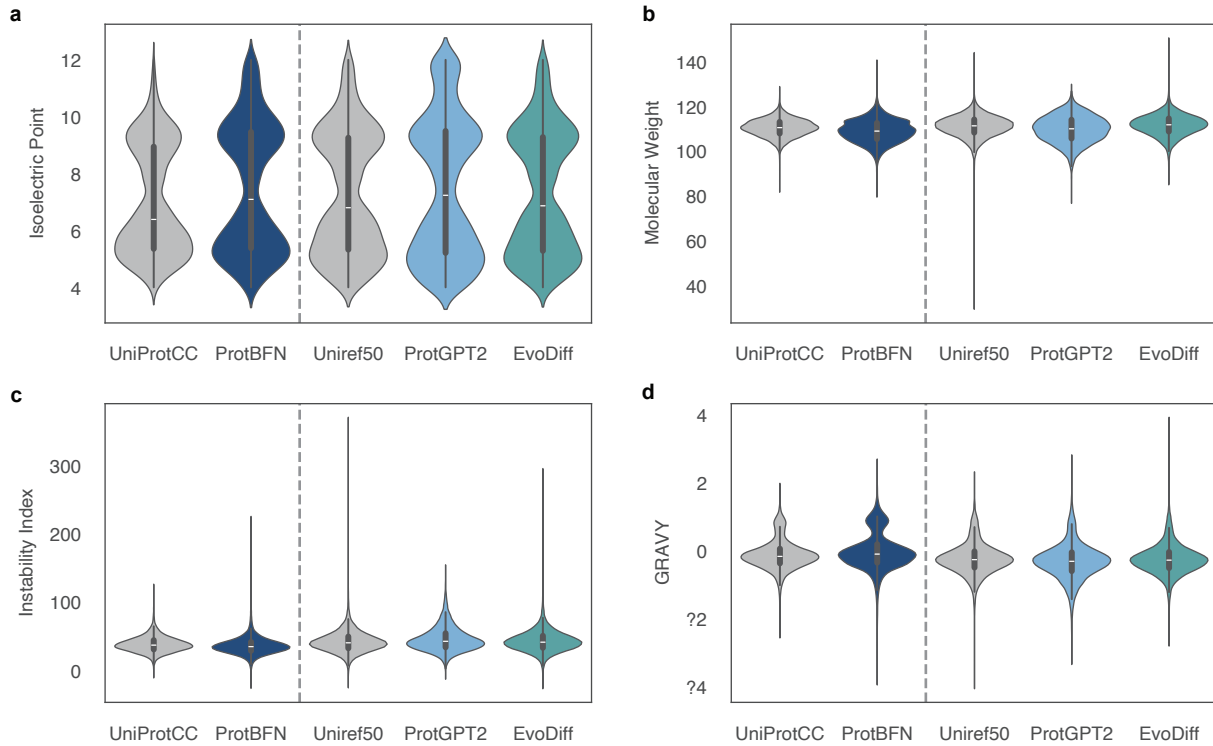
Timothy Atkinson^{*,1}, Thomas D. Barrett^{*,1},
Scott Cameron¹, Bora Guloglu¹, Matthew Greenig¹, Charlie B. Tan¹, Louis Robinson¹,
Alex Graves¹, Liviu Copoiu¹ and Alexandre Laterre¹
^{*}Equal contributions, ¹InstaDeep, 5 Merchant Square, London, England, W2 1AY

Contents

Supplementary Note 1 Additional naturalness measures for generated proteins	2
Supplementary Note 2 Additional coverage measures for generated proteins	4
Supplementary Note 3 Additional structural analysis of generated proteins	6
Supplementary Note 4 Ablation of unconditional sampling methods	8
Supplementary Note 5 UR50BFN: Ablation of UniProtCC and UniRef50	9
Supplementary Note 6 AbBFN generates natural heavy chains	11
Supplementary Note 7 Comparison of training resources	12

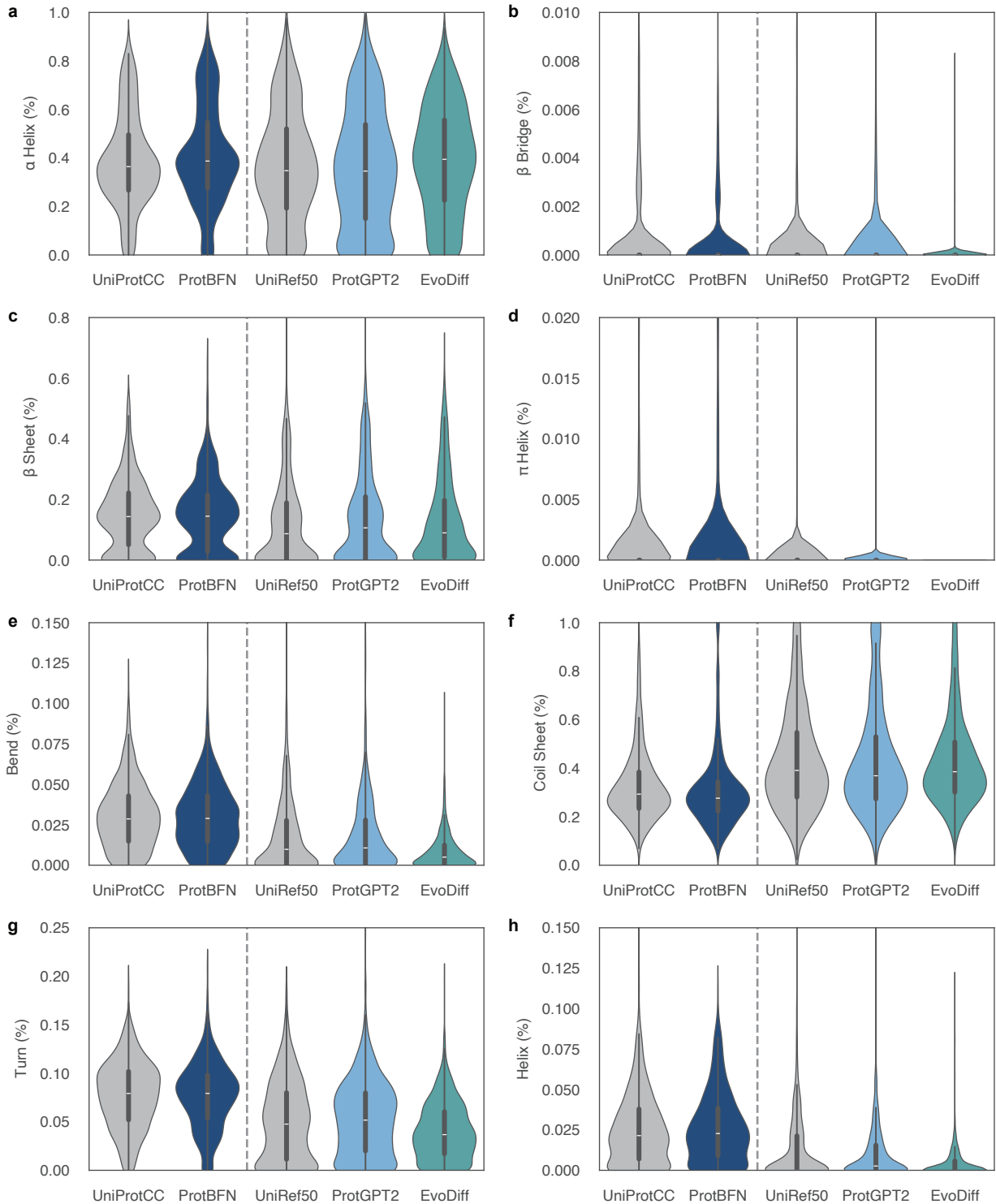
Supplementary Note 1. Additional naturalness measures for generated proteins

ProtParams We compute a number of additional biochemical properties of naturalness of *de novo* generated proteins using BioPython [2], as shown in Supplementary Figure 1. Consistent with our other findings, we observe that ProtBFN is highly consistent with UniProtCC, whereas ProtGPT2 and EvoDiff are instead consistent with UniRef50. As UniProtCC is restricted to consider only those proteins observed in nature, the high coherence with ProtBFN implies that our generated *de novo* proteins are highly natural with respect to these important properties.



Supplementary Figure 1 | ProtBFN effectively *de novo* generates protein sequences that are in distribution with respect to various biochemical properties. (a) The isoelectric point is the pH at which a given protein has a charge of zero, and typically represents the pH at which solubility is minimal. (b) The molecular weight of a given protein sequence can be used for further assessment of many functionalities, such as gene regulation and immunological responses. (c) The instability index of a protein sequence is an estimate for the likelihood that it will be stable in a test tube. (d) The grand average of hydropathicity (GRAVY) is the average hydropathy value of a protein, where negative GRAVY values are hydrophilic and positive values mean they are hydrophobic. In all cases, ProtBFN is seen to well match the natural training distribution UniProtCC, whereas ProtGPT2 and EvoDiff are substantially closer to UniRef50. These plots show a KDE of the data along with a box plot that displays the three quartile values of the distributions (median, upper and lower quartile); with whiskers extending to points within 1.5 IQRs (interquartile range) of the lower and upper quartile. Statistics are calculated using 10 000 samples for each method or data distribution. Source data are provided as a Source Data file.

NetSurfP-3.0 We compute a number of additional structural measures of naturalness of *de novo* generated proteins using NetSurfP-3.0 [8]. These are shown in Supplementary Figure 2. Each residue in each protein is annotated as one of 8 possible secondary structures, or intrinsically disordered. Once again, we observe that ProtBFN is highly consistent with UniProtCC, whereas ProtGPT2 is instead consistent with UniRef50. We note that EvoDiff substantially under-represents a number of secondary structure types, particularly those which occur with lower frequency within UniRef50.



Supplementary Figure 2 | ProtBFN effectively *de novo* generates protein sequences that are in distribution with respect to various per-residue structural annotations. Each plot shows the distribution of secondary structures elements – (a) α helix, (b) β bridge, (c) β sheet, (d) π helix, (e) bend, (f) coil sheet, (g) turn and (h) helix – as predicted by NetSurfP-3.0 [8], measured as a fraction of each overall protein's residues. In all cases, ProtBFN is seen to well match the natural training distribution UniProtCC, whereas ProtGPT2 and EvoDiff are substantially closer to UniRef50. These plots show a KDE of the data along with a box plot that displays the three quartile values of the distributions (median, upper and lower quartile); with whiskers extending to points within 1.5 IQRs (interquartile range) of the lower and upper quartile. Statistics are calculated using 10 000 samples for each method or data distribution. Source data are provided as a Source Data file.

Supplementary Note 2. Additional coverage measures for generated proteins

ESM-MMD estimator Maximum mean discrepancy (MMD) is a metric defined over a space of probability distributions. It measures how different any two distributions are by embedding their support into a reproducing kernel Hilbert space. An unbiased estimator for the squared MMD can be calculated using a kernel as follows [7]

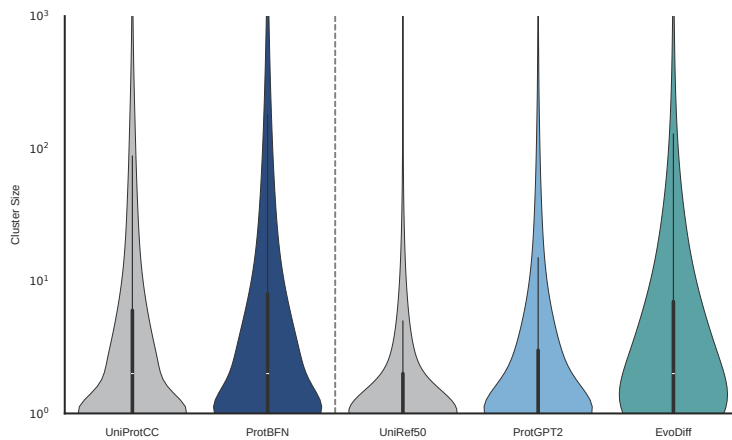
$$\widehat{\text{MMD}}^2 = \frac{1}{N(N-1)} \sum_{i \neq j} (k(x_i, x_j) + k(y_i, y_j) - 2k(x_i, y_j)), \quad (1)$$

where $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ are i.i.d. samples from each distribution, and k is a symmetric positive-definite kernel function. We use the radial basis function kernel to estimate the squared MMD between mean ESM embeddings of each sequence dataset. Supplementary Table 1 shows the estimated MMD^2 between each set of sequences. Note that this estimator is unbiased and so can be negative, despite estimating a squared quantity. We observe that ProtBFN is substantially closer to UniProtCC than any other dataset, even including UniRef50. In contrast, ProtGPT2 is the closest model to UniRef50.

Source	Target	$\widehat{\text{MMD}}^2 (\times 10^{-3})$		
		Mean	Median	Spread
UniProtCC	UniProtCC	-0.295	-0.295	± 0.034
	ProtBFN	6.512	6.512	± 0.031
	ProtGPT2	18.832	18.832	± 0.017
	EvoDiff	62.941	62.940	± 0.014
UniRef50	UniRef50	-0.277	-0.277	± 0.028
	UniProtCC	10.555	10.555	± 0.019
	ProtBFN	16.487	16.487	± 0.015
	ProtGPT2	4.076	4.076	± 0.025
	EvoDiff	32.848	32.847	± 0.027

Supplementary Table 1 | MMD^2 estimates between embeddings. Statistics are calculated using 10 000 samples for each method or data distribution. The best matching distribution (other than the reference distribution itself) under this metric is highlighted in bold.

Cluster size coverage We compare the distribution of the sizes of UniRef50 clusters covered by each model. We run `mmseqs2` against the UniRef50 cluster centers and record all clusters to which at least one generated sequence matches. We then take all of these clusters and plot the distribution of the sizes of these clusters as defined by UniRef50. UniRef50 has many clusters of size 1, and many of the sequences we removed when

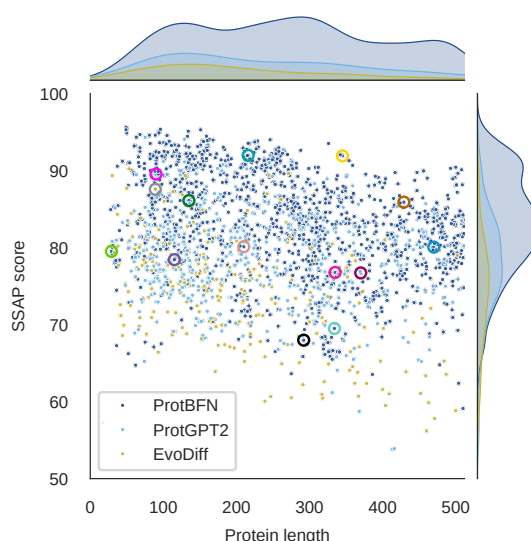


Supplementary Figure 3 | Sizes of unique clusters hit. These plots show a KDE of the data along with a box plot that displays the three quartile values of the distributions (median, upper and lower quartile); with whiskers extending to points within 1.5 IQRs (interquartile range) of the lower and upper quartile. Statistics are calculated using 10 000 samples for each method or data distribution. Source data are provided as a Source Data file.

constructing our training data fall into these isolated clusters. For each value of cluster size, this metric provides an estimate of the number of UniRef50 clusters of that size which lie within the support of the model's learned distribution. Supplementary Figure 3 shows the distribution of cluster sizes for each set of sequences. Since we consider only unique clusters, the distributions shown are not the relative frequency of the model sampling clusters of a given size, but rather the relative coverage of clusters of a given size. In other words, if a given model tends to sample a specific cluster more frequently than others, that is not shown here; what is shown, is the portion of clusters of each size that the model was able to sample from. A closer match between a model and its training dataset would indicate that sequences generated by the model have a broad coverage of the modes present in the dataset. Both ProtBFN and ProtGPT2 show good match in cluster size distribution to their respective training datasets, whilst the sequences from EvoDiff which did match clusters seem to over-represent larger clusters.

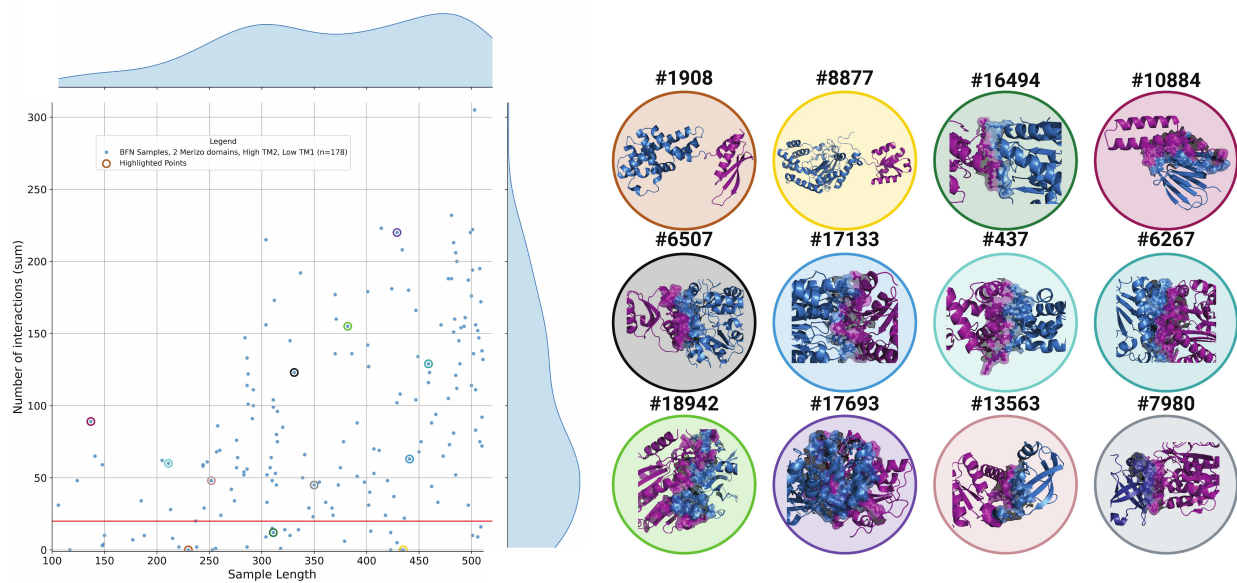
Supplementary Note 3. Additional structural analysis of generated proteins

The sequential structure alignment program SSAP is a metric used to generate the CATH hierarchical classification (Class, Architecture, Topology, Homology) and it calculates the pairwise similarity of structures. When comparing two structures SSAP takes into account beta carbon vectors, the order of residues and secondary structure elements. The algorithm returns a normalised score between 0-100. Pairs of proteins with a score below 60 typically don't show any degree of similarity. Scores between 60-70 correspond to the Architecture level in CATH, and usually have similar local structural elements and belong to the same protein class. Scores between 70-80 correspond to the Topology level in CATH and structures with these scores usually present the same overall fold but with variations in coils and other secondary structure elements. Lastly, scores between 80-100 correspond to the Homologous Superfamily level in CATH, and structures with these scores display extremely similar overall folds [11].



Supplementary Figure 4 | Sample length vs SSAP score for ProtBFN (dark blue), ProtGPT2 (light blue) and EvoDiff (yellow). The SSAP score is calculated between the sample and the best CATH hit. Source data are provided as a Source Data file.

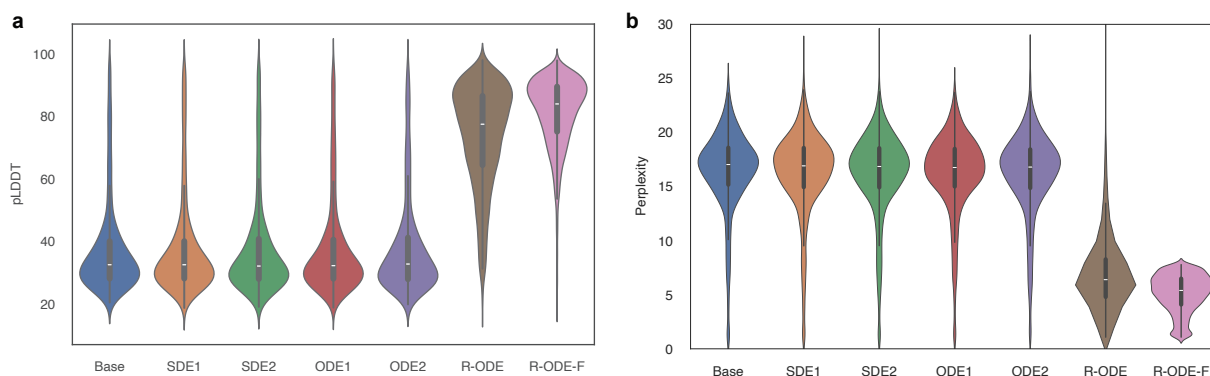
Interacting domains We used the Merizo algorithm to segment BFN samples into individual domains. For our multi-domain samples we wanted to understand if they are simply chained together by flexible loops or if there is some type of global coherence. To this end we used a popular method, called getcontacts [4], to calculate the number of interactions between domains annotated by Merizo. The interactions considered are: hydrogen, salt bridges, van der Waals, pi stacking as well as hydrophobic events. Supplementary Figure 5 displays high level statistics as well as individual examples drawn from the distribution. Because proteins are not static molecules it is hard to place a threshold on how many interactions are required between domains, especially for samples generated unconditionally. As a rule of thumb for setting a threshold we identified PDB ID: 3DH3 which contains two functional domains: binding and catalytic [1], while getcontacts identifies just under 20 interactions. Overall two domain BFN samples display good overall coherence with some examples even exhibiting discontinuous domains (17693, 16494, 7980, 437). A more in-depth structural analysis is beyond the scope of this research article.



Supplementary Figure 5 | Global coherence of two domain BFN samples. Joint plot of BFN samples length vs Number of Interactions. The interactions considered are: hydrogen, salt bridges, van der Waals, pi as well as hydrophobic events. Highlighted points are displayed in. Each domain is coloured differently, and the overall structure is presented in a cartoon representation. The interface is presented using a surface representations while each individual interface residues is displayed using the ball and stick representation. Black dashed lines represent all the interactions identified by PyMOL [3] between the domains. Note that interactions at domain boundaries have been excluded. The right panel is created in BioRender, Copoiu, L. (2025) <https://BioRender.com/q73d282>. Source data are provided as a Source Data file.

Supplementary Note 4. Ablation of unconditional sampling methods

We evaluate the influence of the Reconstructed ODE (R-ODE) sample method introduced in Methods by comparing this sampler to the original BFN method [6] (“Base”) and the first and second order methods introduced in [13] (“ODE1”, “ODE2”, “SDE1”, “SDE2”). Figure 6 presents results for 2048 samples generated from ProtBFN using each method with 10,000 sampling steps. In Supplementary Figure 6a shows violin plots of pLDDT scores, as predicted by ESMFold, for generated samples, where it is seen that pLDDT scores are systematically higher when using the R-ODE method. Correspondingly in Supplementary Figure 6b, sample perplexity, as measured according to the ProtBFN model, is substantially reduced when using R-ODE. Additionally, in both cases the influence of the perplexity and repetitiveness filters described in Methods is presented as R-ODE-F. These filtering methods unsurprisingly reduce the perplexity of generated samples, and yield a corresponding boost in pLDDT scores. Overall, these results provide compelling empirical evidence for the R-ODE sampling method.



Supplementary Figure 6 | The R-ODE sampling method is essential for generating globally coherent proteins. (a) using the R-ODE method substantially improves pLDDT scores (as predicted by ESMFold) of samples in comparison to those generated using the base sampling method for BFNs [6] or the SDE and ODE methods presented in [13]. Correspondingly, in (b), sample perplexities are systematically reduced when using the R-ODE method. In both cases, the incorporation of perplexity and repetition filters (R-ODE-F) further improve over the performance of R-ODE. These plots show a KDE of the data along with a box plot that displays the three quartile values of the distributions (median, upper and lower quartile); with whiskers extending to points within 1.5 IQRs (interquartile range) of the lower and upper quartile. Statistics are calculated using 2048 samples for each sampling method with the exceptions of R-ODE and R-ODE-F which use 8192 and 10 000 samples, respectively. Source data are provided as a Source Data file.

Supplementary Note 5. UR50BFN: Ablation of UniProtCC and UniRef50

In this section, we compare the influence of UniProtCC as a training set in comparison to UniRef50, to better isolate the origins of the empirical advantages of ProtBFN. To this end, a new model, UR50BFN, is trained according to exactly the same methodology as ProtBFN as described in Methodology, except that the exact UniRef50 training data from [5] is used as the training set. UR50BFN is trained for 100,000 steps.

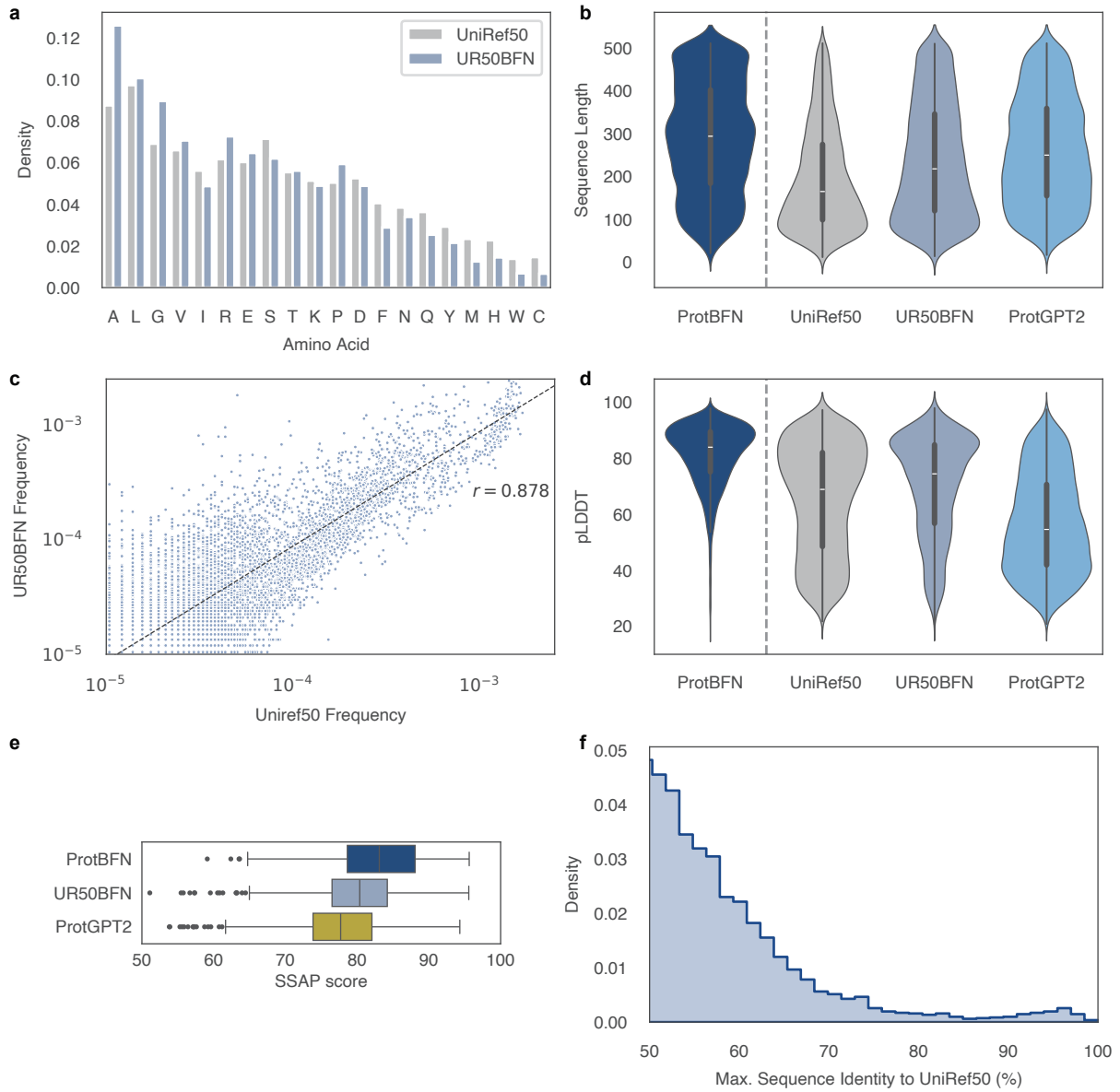
Supplementary Figure 7 shows various metrics of interest for 10,000 samples drawn from the UR50BFN model in comparison to those of ProtBFN, UniRef50 and ProtGPT2. In a similar manner to ProtBFN, UR50BFN is quite consistent with its training data with respect to Amino Acid distribution (Supplementary Figure 7a), sequence length, (Supplementary Figure 7b) and oligomer frequency (Supplementary Figure 7c). With respect to structural analysis, the pLDDT scores shown in Figure Supplementary Figure 7d (as predicted by ESMFold) are substantially higher for UR50BFN in comparison to ProtGPT2 despite the fact that both models are trained on the same training data. Additionally, SSAP score analysis (Supplementary Figure 7e) demonstrates that UR50BFN generates sequences which have predicted structures that are more structurally similar to the CATH S40 representatives. Nevertheless, in both cases, UR50BFN generates sequences with lower pLDDT and SSAP scores than ProtBFN, indicating that our choice of training on UniProtCC has a substantial influence on overall performance. As with ProtBFN, Supplementary Figure 7f shows that the vast majority of the sequences that UR50BFN generates are novel.

Supplementary Table 2 presents results for cluster hits and coverage score of UR50BFN in comparison to ProtBFN and ProtGPT2. These results are consistent with those of Supplementary Figure 7; UR50BFN generally covers the known protein space more completely than ProtGPT2, but generates less diverse proteins in comparison to ProtBFN.

From this extended analysis, we observe that there are two key components to the empirical performance of ProtBFN; firstly, the choice of data (comparing ProtBFN and UR50BFN), and secondly the choice of model (comparing UR50BFN and ProtGPT2).

Model	Cluster hits	Coverage score
ProtBFN	69.7 %	0.544
UR50BFN	33.9 %	0.205
ProtGPT2	15.7 %	0.095

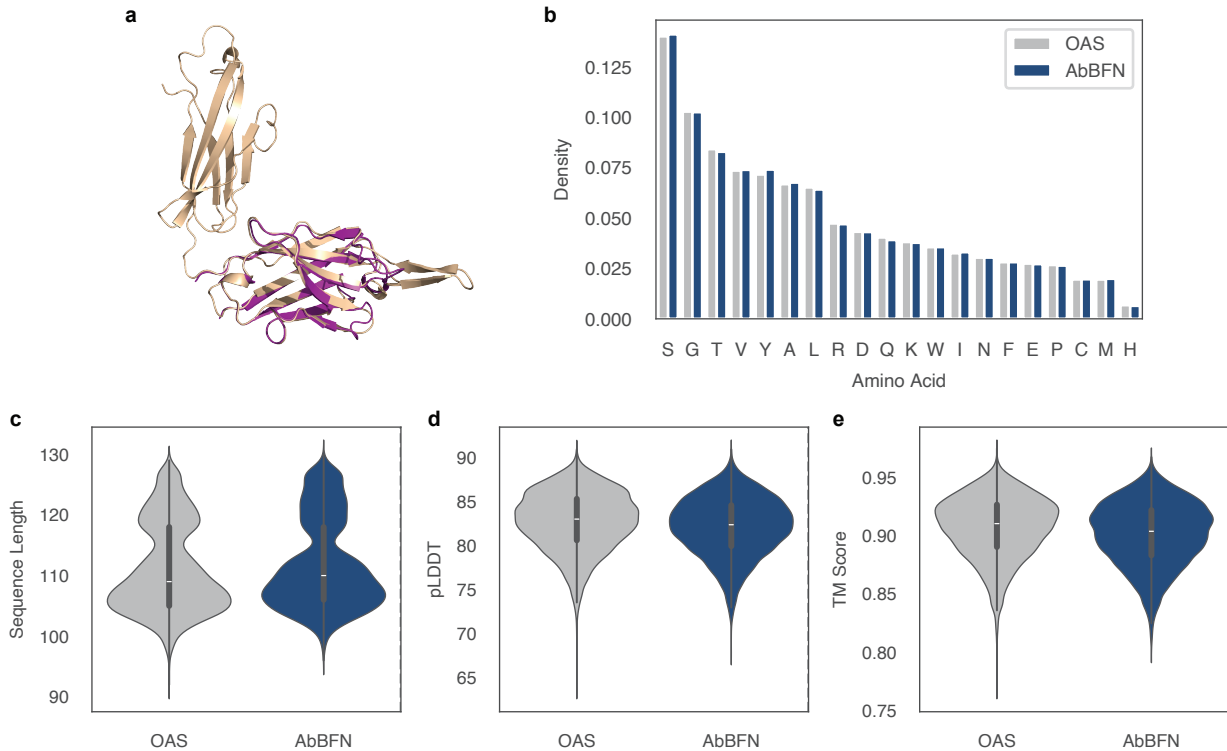
Supplementary Table 2 | UR50BFN generates natural and diverse proteins. 10 000 generated sequences from ProtBFN and ProtGPT2, 8680 sequences from UR50BFN, are aligned against UniRef50 with above 50 % sequence identity between a sequence and cluster counted as a ‘hit’. ProtBFN sequences are found to align with known clusters significantly more often than those from UR50BFN or ProtGPT2. Nevertheless, UR50BFN, which is trained on the same training data as ProtGPT2, achieves higher cluster hits and coverage scores. Higher scores corresponding to broader coverage of the target clusters. The best performing method under each metric is highlighted in bold.



Supplementary Figure 7 | UR50BFN is an effective *de novo* generator of protein sequences, but underperforms in comparison to ProtBFN. The local metrics considered are amino acid (a) and (b) the sequence lengths (b); both of which show a reasonable match to the training distribution of the model. Both oligomer frequencies (c) and predicted mean local distance difference test (pLDDT) scores from ESMFold [9] (d) are shown. Predictions with pLDDT > 70 are generally considered high confidence. The 50 256 oligomers used are taken from the vocabulary of ProtGPT2 [5] and the plot has overlaid a linear correlation fit and the associated Pearson correlation coefficient. Also included are ProtBFN and a baseline autoregressive, ProtGPT2 [5], model. A more granular analysis of the structural hits is given by sequential structure alignment program (SSAP) scores [10, 12] (e). SSAP scores of 60–70, 70–80, and 80–100 correspond to similarity at the architecture, topology, and homologous superfamily levels, respectively, in the CATH nomenclature. Finally, the maximum sequence identity of the ProtBFN generated sequences to the UniRef50 training data is plotted (f) to demonstrate that the training data is not being memorised. Violin plots (b, d) show a KDE of the data along with a box plot. These box plots; and that in e, displays the three quartile values of the distributions (median, upper and lower quartile); with whiskers extending to points within 1.5 IQRs (interquartile range) of the lower and upper quartile. Statistics are calculated using 10 000 samples for each method or data distribution, except UR50BFN which uses 8680 samples. Source data are provided as a Source Data file.

Supplementary Note 6. AbBFN generates natural heavy chains

Similarly to ProtBFN, an important goal of AbBFN is to learn the distribution of natural VH chains. We employ various measures of naturalness for unconditional heavy chains, considering the distribution of sequence lengths, amino acids, ESM pLDDT. To this end, Supplementary Figure 8 presents a selection of such measures for 10,000 generated sequences in comparison to 10,000 natural sequences taken from OAS, from which we can infer that AbBFN closely matches the natural OAS distribution on which it was trained. As an additional measure, we align generated structures, as predicted by ESMFold, against reference chain 6vi2_chain_B using TM-Align [14] to obtain a TM-score. We again observe strong concordance between AbBFN generated chains and natural OAS chains, and consistently high TM scores indicating that the generated sequences structurally resemble a canonical VH fold.



Supplementary Figure 8 | AbBFN is an effective *de novo* generator of VH chain sequences; generating novel VH chain sequences that in distribution under both local and global metrics. In (a) a generated VH chain is compared to reference chain 6vi2_chain_b, showing clear structural consistency. The amino acid (b) frequencies show a strong match to the training distribution of the model. Both the sequence lengths (c) and predicted mean local distance difference test (pLDDT) scores from ESMFold [9] (d) are shown. Predictions with pLDDT>70 are generally considered high confidence. By computing the TM-score of generated and natural VH chains, against reference chain 6vi2_chain_b, we again observe strong consistency between AbBFN and OAS (e). In all cases, AbBFN is seen to well match the natural training distribution OAS. Violin plots show a KDE of the data along with a box plot that displays the three quartile values of the distributions (median, upper and lower quartile); with whiskers extending to points within 1.5 IQRs (interquartile range) of the lower and upper quartile. Statistics are calculated using 10 000 samples for each data distribution. Source data are provided as a Source Data file.

Supplementary Note 7. Comparison of training resources

We provide detailed information regarding the model size, dataset size and training hardware of various models evaluated within the main manuscript in Supplementary Table 3.

Model	Params	Training Data	Training Steps	Training Hardware
ProtBFN	650M	71M sequences (representative of 60M clusters)	500k steps	128 TPU v4 chips (~26 NVIDIA A100s), 2 weeks
UR50BFN	650M	745M (Uniref50) sequences (representative of 45M clusters)	100k steps	64 TPU v4 chips (~13 NVIDIA A100s), 1 week
ProtGPT2	738M	45M (Uniref50) sequences (representative of 45M clusters)	50 epochs	128 NVIDIA A100s, 4 days
EvoDiff	640M	45M (Uniref50) sequences (representative of 45M clusters)	400k-2M steps	32 NVIDIA V100, 12-23 days
AbBFN	650M	195M sequences	100k steps	64 TPU v4 chips (~13 NVIDIA A100s), 3 days
AntiBERTy	26M	558M sequences	8 epochs	4 NVIDIA A100s, 10 days
AbLang2	50-60M	28.8M sequences	200k steps	N/A

Supplementary Table 3 | Comparison of training data and resources for discussed methods and models.

Supplementary References

- [1] A. Alian, A. DeGiovanni, S. L. Griner, J. S. Finer-Moore, and R. M. Stroud. Crystal structure of an RluF–RNA complex: a base-pair rearrangement is the key to selectivity of RluF for U2604 of the ribosome. *Journal of molecular biology*, 388(4):785–800, 2009.
- [2] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009.
- [3] W. L. DeLano et al. PyMOL: An open-source molecular graphics tool. *CCP4 Newsl. Protein Crystallogr*, 40(1):82–92, 2002.
- [4] G. Developers. Getcontacts: Library for computing dynamic non-covalent contact networks in proteins throughout md simulation. <https://github.com/getcontacts/getcontacts>, 2024. Accessed: 2024-07-10.
- [5] N. Ferruz, S. Schmidt, and B. Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.
- [6] A. Graves, R. K. Srivastava, T. Atkinson, and F. Gomez. Bayesian flow networks. *arXiv*, 2308.07037, 2023.
- [7] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.
- [8] M. H. Høie, E. N. Kiehl, B. Petersen, M. Nielsen, O. Winther, H. Nielsen, J. Hallgren, and P. Marcatili. NetSurfP-3.0: accurate and fast prediction of protein structural features by protein language models and deep learning. *Nucleic Acids Research*, 50(W1):W510–W515, 06 2022.
- [9] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
- [10] C. A. Orengo and W. R. Taylor. SSAP: sequential structure alignment program for protein structure comparison. In *Methods in enzymology*, volume 266, pages 617–635. Elsevier, 1996.
- [11] G. Porwal, S. Jain, S. D. Babu, D. Singh, H. Nanavati, and S. Noronha. Protein structure prediction aided by geometrical and probabilistic constraints. *Journal of Computational Chemistry*, 28(12):1943–1952, 2007.
- [12] W. R. Taylor and C. A. Orengo. Protein structure alignment. *Journal of molecular biology*, 208(1):1–22, 1989.
- [13] K. Xue, Y. Zhou, S. Nie, X. Min, X. Zhang, J. Zhou, and C. Li. Unifying bayesian flow networks and diffusion models through stochastic differential equations, 2024.
- [14] Y. Zhang and J. Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.