Protein generation with evolutionary diffusion: sequence is all you need

Sarah Alamdari Microsoft Research
Nitya Thakkar Brown University
Rianne van den Berg Microsoft Research
Alex X. Lu Microsoft Research
Nicolo Fusi Microsoft Research
Ava P. Amini Microsoft Research
Kevin K. Yang Microsoft Research

Abstract

Diffusion models have demonstrated the ability to generate biologically plausible proteins that are dissimilar to any proteins seen in nature, enabling unprecedented capability and control in *de novo* protein design. However, current state-of-the-art diffusion models generate protein structures, which limits the scope of their training data and restricts generations to a small and biased subset of protein space. We introduce a general-purpose diffusion framework, EvoDiff, that combines evolutionary-scale data with the conditioning capabilities of diffusion models for controllable protein generation in sequence space. EvoDiff generates high-fidelity, diverse, structurally-plausible proteins that cover natural sequence and functional space. Critically, EvoDiff can generate proteins inaccessible to structure-based models, such as those with disordered regions, and design scaffolds for functional structural motifs, demonstrating the universality of our sequence-based formulation. We envision that EvoDiff will expand capabilities in protein engineering beyond the structure-function paradigm toward programmable, sequence-first design.

1 Introduction

Evolution has yielded a diversity of functional proteins. Deep generative models learn from this diversity to generate valid and novel proteins, with the ultimate goal of tailoring function to solve outstanding modern-day challenges, such as the development of targeted therapeutics or engineered enzymes (**Fig. 1A**) [1, 2]. Diffusion models provide a particularly powerful generative modeling framework that generates high-quality, diverse samples that can be conditioned on a variety of design objectives [3–6]. Indeed, today's most biologically-plausible *in silico*-designed proteins come from diffusion models of protein structure [7–15]. These models fit in the structure-based design paradigm of first generating a structure that fulfills desired constraints and then designing a sequence that folds to that structure. However, *sequence*, not structure, is the universal protein design space. Every protein is completely defined by and synthesized as an amino-acid sequence, which then determines function through an ensemble of structural conformations and amino-acid chemistry. However, not every protein folds into a static structure [16–18]. Further, structural data (ca. 200k structures in PDB) is available for a sparse, biased subset of proteins (ca. billions of unique natural proteins; **Fig. 1A**), limiting the capacity of structure-based models to learn the full diversity of protein functional space.

We combine evolutionary-scale datasets with diffusion models to develop a powerful new generative modeling framework, EvoDiff, for controllable protein design from sequence data alone (**Fig. 1**). Given the natural framing of proteins as sequences of tokens, we use a *discrete* diffusion framework in which a forward process iteratively corrupts a protein sequence by changing its amino acid identities, and a learned reverse process, parameterized by a neural network, predicts the changes made at each iteration (**Fig. 1B**) and can be used to generate new sequences starting from random noise (**Fig. 1C**).

NeurIPS 2023 Generative AI and Biology Workshop.

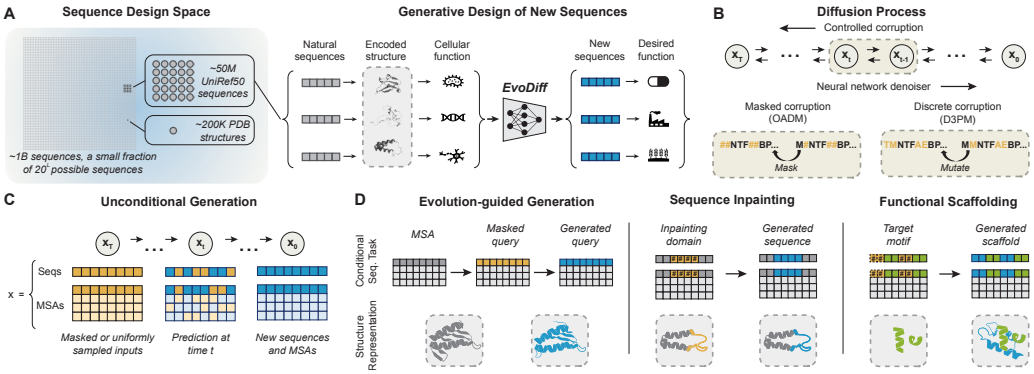

Figure 1: **Protein sequence generation with evolutionary diffusion. (A)** (Left) Evolution has sampled a tiny fraction of possible protein sequences. Experimental structures exist for even fewer proteins. (Right) EvoDiff is a generative discrete diffusion model trained on natural protein sequences. **(B)** EvoDiff's controlled corruption and learned denoising processes. In masked corruption, input tokens are masked in an order-agnostic fashion (bottom, left). In discrete corruption, inputs are corrupted via a Markov process controlled by a transition matrix of amino acid mutation frequencies (bottom, right). **(C)** EvoDiff enables unconditional generation of sequences or MSAs. **(D)** Controllable design with EvoDiff by conditioning on MSAs (left); inpainting (middle); or scaffolding (right).

Because of its grounding in a universal design space, EvoDiff unconditionally generates high-quality, diverse proteins and, via conditioning, enables evolution-guided design of novel sequences, inpainting of functional motifs, and motif scaffolding without any explicit structural information (**Fig. 1D**).

2 Discrete diffusion models of protein sequence

EvoDiff is the first generative diffusion model for protein design trained on evolutionary-scale sequence data. We investigated two types of forward processes for diffusion over discrete data modalities: order-agnostic autoregressive diffusion (EvoDiff-OADM, Methods) [19] and discrete denoising diffusion probabilistic models (EvoDiff-D3PM, Methods) [20] (**Fig. 1B**). In EvoDiff-OADM, one amino acid is converted to a special mask token at each step in the forward process (**Fig. 1B**). In EvoDiff-D3PM, the forward process corrupts sequences by sampling mutations according to a transition matrix, such that after T steps the sequence is indistinguishable from a uniform sample over the amino acids (**Fig. 1B**). In the reverse process for both, a neural network model is trained to undo the previous corruption. The trained model can then generate new sequences starting from sequences of masked tokens (EvoDiff-OADM) or of uniformly-sampled amino acids (EvoDiff-D3PM) (**Fig. 1C**).

To facilitate direct and quantitative model comparisons, we trained all EvoDiff sequence models on 42M sequences from UniRef50 [21] using a dilated convolutional neural network architecture of either 38M- or 640M-parameters [22]. We calculated each model's test-set perplexity, which reflects its ability to capture the distribution of natural sequences and generalize to unseen sequences (Methods). EvoDiff-OADM learns to reconstruct the test set more accurately than two tested EvoDiff-D3PM variants employing uniform and BLOSUM62-based transition matrices and is the only model variant where performance scales with increased model size (**Table S1; Fig. S1**).

3 Results

Structural plausibility of generated sequences. We investigated whether EvoDiff could generate new protein sequences that were individually valid and structurally plausible, via a workflow that evaluates the foldability of sequences generated by EvoDiff (**Fig. S2**). We generated 1000 sequences from each EvoDiff sequence model with lengths drawn from the empirical distribution of lengths in the training set. We compared EvoDiff's generations to sequences generated from a left-to-right autoregressive language model (LRAR) with the same architecture and training set as EvoDiff and from protein masked language models such as ESM-2 [23] (**Figs. 2A-B, S3, S4; Table S2**).

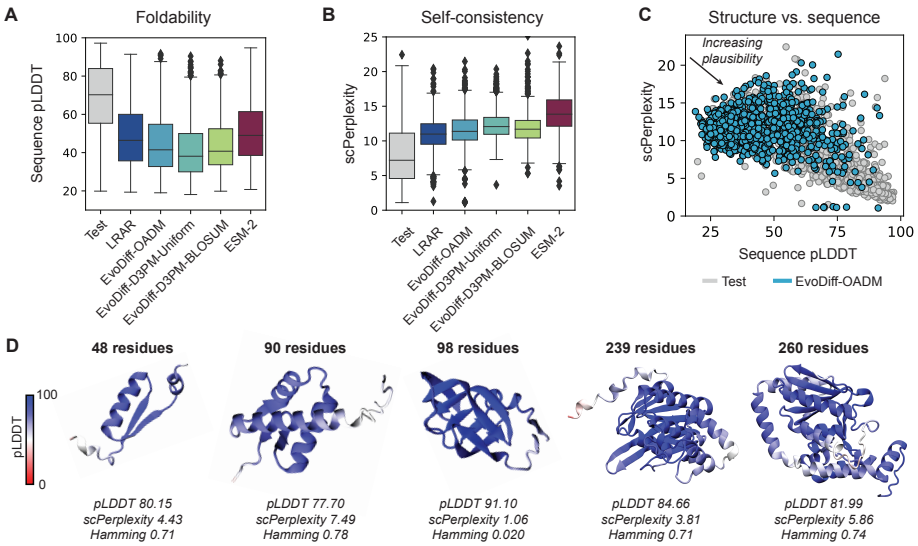

Figure 2: **EvoDiff generates realistic and structurally-plausible protein sequences**. **(A-B)** Distributions of foldability (A) and self-consistency (B) metrics for sequences from the test set, EvoDiff models, and baselines (n=1000 per model). **(C)** Sequence pLDDT vs. scPerplexity for sequences from the test set (grey, n=1000) and the 640M-parameter OADM model EvoDiff-Seq (blue, n=1000). **(D)** Predicted structures and metrics for select structurally plausible generations from EvoDiff-Seq.

We assessed the foldability of individual sequences by predicting their corresponding structures using OmegaFold [24] and computing the average predicted local distance difference test (pLDDT) across the whole structure (**Fig. 2A, S5**). pLDDT reflects the confidence in the structure prediction for each residue. While pLDDT scores above 70 are often considered to be high confidence, low pLDDT can be consistent with intrinsically disordered regions (IDRs) of proteins [25]. As an additional metric of structural plausibility, we computed a self-consistency perplexity (scPerplexity) by redesigning each predicted structure with the inverse folding algorithm ESM-IF [26] and computing the perplexity against the original generated sequence (**Fig. 2B; Table S2**). While no generative model approaches the test set values for foldability and self-consistency, EvoDiff-OADM outperforms EvoDiff-D3PM and improves when increasing the model size (**Fig. 2A-C**; **Table S2**). We thus select the 640M-parameter EvoDiff-OADM model for downstream analysis and hereafter refer to it as EvoDiff-Seq. Analysis of samples from EvoDiff-Seq illustrates their structural plausibility and novelty, demonstrating that EvoDiff generates protein sequences that are individually valid (**Fig. 2D**).

Biological properties of generated sequence distributions. We next evaluated how well the *distribution* of designed protein sequences covered natural protein space. To assess coverage over sequence and functional properties, we embedded each generated sequence using ProtT5 [27], a protein language model benchmarked for imputing GO annotations [28], and calculated the embedding space Fréchet distance between a set of generated sequences and the test set, where lower distance reflects better coverage (**Figs. S6-S8; Table S1**). Both qualitatively and quantitatively, EvoDiff-Seq generates proteins that better recapitulate natural sequence and functional diversity than sampling from a state-of-the-art protein masked language model (ESM-2) or predicting sequences from structures generated by a state-of-the-art structure diffusion model (RFdiffusion) (**Fig. S6A**).

We evaluated structural properties by computing 3-state secondary structures [29] for each residue in generated sequences and examining the resulting distributions of structural properties (**Figs. S6B, S9**). EvoDiff-Seq generates proportions of strands and disordered regions similar to those in natural sequences, while ESM-2 and RFdiffusion both generate proteins enriched in helices (**Fig. S6B**). We assessed the novelty of our generations and found that, on average, a sequence from EvoDiff-Seq has a Hamming distance of 0.83 from the most similar training sequence of the same length (**Table S1**). These results show that EvoDiff's diffusion objective and evolutionary-scale training data are both necessary to generate novel sequences that cover protein sequence, functional, and structural space.

Conditional sequence generation for controllable design. EvoDiff's OADM diffusion framework induces a natural method for conditional generation by fixing some subsequences and inpainting the remainder. Because the model is trained to generate proteins with an arbitrary decoding order, this is accomplished by simply masking and decoding the desired portions. We applied EvoDiff's power for controllable protein design across three scenarios (**Fig. 1D**): conditioning on evolutionary information from MSAs (**Fig. S10-S12**), inpainting functional domains such as IDRs (**Fig. S13-S15**), and scaffolding structural motifs (**Fig. 3**). To condition on evolutionary information, we designed and trained EvoDiff MSA models using the MSA Transformer [30] architecture on the OpenFold dataset [31] (**Table S3; Fig. S11**). By performance we select the OADM-Max model as EvoDiff-MSA and demonstrate its ability to generate sequences conditioned on an MSA (**Fig. S10; Table S4**).

Scaffolding functional motifs. Thus far, the primary application of deep generative models of protein structure is their ability to scaffold binding and catalytic motifs: given 3D motif coordinates, these models can often generate a structural scaffold that holds the motif in precisely the 3D geometry needed for function [10, 14, 32]. Given that the fixed motif includes information on its residue identities, we investigated whether a structural model is actually necessary for motif scaffolding.

We used conditional generation with EvoDiff to generate scaffolds for 17 motif-scaffolding problems [10] by fixing the functional motif, supplying only the motif's amino-acid sequence as conditioning information, and then decoding the remainder of the sequence (**Fig. 1D, 3A-C**). The problems include simple "inpainting", viral epitopes, receptor traps, small molecule and protein binding sites, and enzyme active sites. We compared the performance of EvoDiff, which uses only sequence information, to the state-of-the-art structure model RFdiffusion, by using OmegaFold to predict structures for our generated sequences as well as for sequences inverse-folded from RFdiffusion structures (**Fig. 3A-B**). We find almost no correlation between the problem-specific success rates of EvoDiff and RFdiffusion, showing that the two may have orthogonal strengths (**Fig. 3A-B**). EvoDiff-Seq and EvoDiff-MSA generate successful scaffolds for 8 and 13 of the 17 problems, respectively (**Table S5, S6**). Successful scaffolds from EvoDiff demonstrate the qualitative and quantitative quality of generated proteins and predicted structures for select functional motifs (**Fig. 3D-E**). These results show that EvoDiff can scaffold structural motifs via conditional generation in sequence space alone.

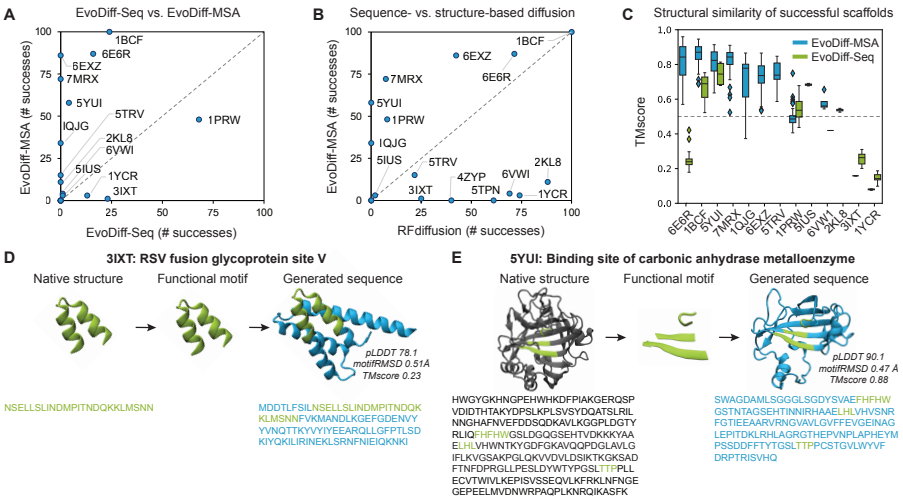

Figure 3: **Scaffolding functional motifs without explicit structural information**. **(A)** Successes from n=100 trials for EvoDiff-Seq (x-axis) versus EvoDiff-MSA (y-axis) across scaffolding problems in which at least one method succeeds. **(B)** Successes of sequence-based scaffolding via EvoDiff-MSA (y-axis) vs. structure-based scaffolding via RFdiffusion (x-axis) (n=100 trials per problem). **(C)** Distributions of TM-scores of successful scaffolds from EvoDiff models relative to the true structures (dashed line at 0.5). **(D-E)** Generated sequences, predicted structures, and metrics for generations from EvoDiff-Seq (D) and EvoDiff-MSA (E). Motif: green, original: black, generated scaffold: blue.

4 Conclusion

We present EvoDiff, a diffusion modeling framework capable of generating high-fidelity, diverse, and novel proteins with the ability to condition on sequence constraints. Because it operates in the universal protein design space, EvoDiff can sample diverse and plausible proteins, generate intrinsically disordered regions, and scaffold structural motifs using only sequence information, challenging a paradigm in structure-based protein design. EvoDiff is the first framework to demonstrate the power of diffusion modeling over evolutionary-scale protein sequence space. EvoDiff may enable new abilities in controllable protein design by reading and writing function in the language of proteins.

References

[1] Zachary Wu et al. "Protein sequence design with deep generative models". In: *Current Opinion in Chemical Biology* 65 (2021), pp. 18–27.

[2] Sarah L Lovelock et al. "The road to fully programmable protein catalysis". In: *Nature* 606.7912 (2022), pp. 49–58.

[3] Jascha Sohl-Dickstein et al. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International Conference on Machine Learning*. PMLR. 2015, pp. 2256–2265.

[4] Prafulla Dhariwal and Alexander Nichol. "Diffusion models beat GANs on image synthesis". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 8780–8794.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[6] Robin Rombach et al. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 10684–10695.

[7] Namrata Anand and Tudor Achim. "Protein structure and sequence generation with equivariant denoising diffusion probabilistic models". In: *arXiv* 2205.15019 (2022).

[8] Kevin E. Wu et al. "Protein structure generation via folding diffusion". In: *arXiv* 2209.15611 (2022).

[9] Brian L Trippe et al. "Diffusion Probabilistic Modeling of Protein Backbones in 3D for the motif-scaffolding problem". In: *The Eleventh International Conference on Learning Representations* 11 (2023).

[10] Joseph L Watson et al. "De novo design of protein structure and function with RFdiffusion". In: *Nature* 620 (2023), pp. 1089–1100.

[11] John Ingraham et al. "Illuminating protein space with a programmable generative model". In: *bioRxiv* 2022.12.01.518682 (2022).

[12] Yeqing Lin and Mohammed AlQuraishi. "Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds". In: *Proceedings of the 40th International Conference on Machine Learning* (2023).

[13] Jason Yim et al. "SE (3) diffusion model with application to protein backbone generation". In: *arXiv preprint arXiv:2302.02277* (2023).

[14] Jin Sub Lee, Jisun Kim, and Philip M Kim. "Score-based generative modeling for de novo protein design". In: *Nature Computational Science* 3 (2023), pp. 382–392.

[15] Alexander E Chu et al. "An all-atom protein generative model". In: *bioRxiv* (2023).

[16] Nobuhiko Tokuriki and Dan S Tawfik. "Protein dynamism and evolvability". In: *Science* 324.5924 (2009), pp. 203–207.

[17] Philip A Romero and Frances H Arnold. "Exploring protein fitness landscapes by directed evolution". In: *Nature Reviews Molecular Cell Biology* 10.12 (2009), pp. 866–876.

[18] Jiali Gao and Donald G Truhlar. "Quantum mechanical methods for enzyme kinetics". In: *Annual Review of Physical Chemistry* 53.1 (2002), pp. 467–505.

[19] Emiel Hoogeboom et al. "Autoregressive Diffusion Models". In: *The Eleventh International Conference on Learning Representations* 11 (2022).

[20] Jacob Austin et al. "Structured Denoising Diffusion Models in Discrete State-Spaces". In: *Advances in Neural Information Processing Systems* 34 (2021).

[21] Baris E Suzek et al. "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches". In: *Bioinformatics* 31.6 (2015), pp. 926–932.

[22] Kevin K. Yang, Nicolo Fusi, and Alex X. Lu. "Convolutions are competitive with transformers for protein sequence pretraining". In: *bioRxiv* (2022). DOI: 10.1101/2022.05.19.492714.

[23] Robert Verkuil et al. "Language models generalize beyond natural proteins". In: *bioRxiv* (2022). DOI: 10.1101/2022.12.21.521521.

[24] Ruidong Wu et al. "High-resolution de novo structure prediction from primary sequence". In: *bioRxiv* (2022).

[25] Kiersten M Ruff and Rohit V Pappu. "AlphaFold and implications for intrinsically disordered proteins". In: *Journal of Molecular Biology* 433.20 (2021), p. 167208.

[26] Chloe Hsu et al. "Learning inverse folding from millions of predicted structures". In: *Proceedings of the 39th International Conference on Machine Learning* 162 (2022), pp. 8946–8970.

[27] Ahmed Elnaggar et al. "ProtTrans: Toward understanding the language of life through self-supervised learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2021), pp. 7112–7127.

[28] Maria Littmann et al. "Embeddings from deep learning transfer GO annotations beyond homology". In: *Scientific Reports* 11.1 (2021), p. 1160.

[29] Wolfgang Kabsch and Christian Sander. "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features". In: *Biopolymers: Original Research on Biomolecules* 22.12 (1983), pp. 2577–2637.

[30] Roshan M Rao et al. "MSA Transformer". In: *Proceedings of the 38th International Conference on Machine Learning* 139 (2021), pp. 8844–8856.

[31] Gustaf Ahdritz et al. "OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization". In: *bioRxiv* (2022). DOI: 10.1101/2022.11.20.517210. eprint: https://www.biorxiv.org/content/early/2022/11/24/2022.11.20.517210.full.pdf. URL: https://www.biorxiv.org/content/early/2022/11/24/2022.11.20.517210.

[32] Jue Wang et al. "Scaffolding protein functional sites using deep learning". In: *Science* 377.6604 (2022), pp. 387–394.

[33] Emiel Hoogeboom et al. "Argmax Flows and Multinomial Diffusion: Learning Categorical Distributions". In: *arXiv* 2102.05379 (2021).

[34] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *arXiv* 2010.02502 (2020).

[35] Steven Henikoff and Jorja G Henikoff. "Amino acid substitution matrices from protein blocks." In: *Proceedings of the National Academy of Sciences* 89.22 (1992), pp. 10915–10919.

[36] Nal Kalchbrenner et al. "Neural Machine Translation in Linear Time". In: *arXiv* 1610.10099 (2017).

[37] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[38] Ashish Vaswani et al. "Attention Is All You Need". In: *arXiv* 1706.03762 (2017).

[39] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv* 1412.6980 (2017).

[40] Alexander Rives et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118.

[41] Susann Vorberg, Stefan Seemayer, and Johannes Söding. "Synthetic protein alignments by CCMgen quantify noise in residue-residue contact prediction". In: *PLOS Computational Biology* 14.11 (2018), e1006526.

[42] Noelia Ferruz et al. "From sequence to function through structure: deep learning for protein design". In: *Computational and Structural Biotechnology Journal* 21 (2023), pp. 238–250.

[43] Yang Zhang and Jeffrey Skolnick. "Scoring function for automated assessment of protein structure template quality". In: *Proteins: Structure, Function, and Bioinformatics* 57.4 (2004), pp. 702–710.

[44] Alex X Lu et al. "Discovering molecular features of intrinsically disordered regions by using evolution for contrastive learning". In: *PLOS Computational Biology* 18.6 (2022), e1010238.

[45] Jack Hanson et al. “Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks”. In: *Bioinformatics* 33.5 (2017), pp. 685–692.

[46] Adrian M Altenhoff et al. “OMA orthology in 2021: website overhaul, conserved isoforms, ancestral gene order and more”. In: *Nucleic Acids Research* 49.D1 (2021), pp. D373–D379.

[47] Ananthan Nambiar et al. “DR-BERT: A Protein Language Model to Annotate Disordered Regions”. In: *bioRxiv* (2023).

[48] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589.

Appendix: Methods and Supplementary Information

Diffusion models Diffusion models are a class of generative models that learn to generate data from noise. They consist of a forward corruption process and a learned reverse denoising process. The forward process is a Markov chain of diffusion steps $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ that corrupts an input ($\boldsymbol{x}_0$) over T timesteps such that $\boldsymbol{x}_T$ is indistinguishable from random noise. The learned reverse denoising process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is parameterized by a model such as a neural network and generates new data from noise. Discrete diffusion models have previously been developed over binary random variables [3], developed over categorical random variables with uniform transition matrices [33, 34], linked to autoregressive models [19], and optimized for use with transition matrices [20].

This work presents models from two different discrete diffusion frameworks – order-agnostic autoregressive diffusion models (OADMs) and discrete denoising diffusion probabilistic models (D3PMs) – on protein sequences and multiple sequence alignments (MSAs).

Discrete Denoising Diffusion Probabilistic Models (D3PMs) Discrete denoising diffusion probabilistic models (D3PMs) operate by defining a transition matrix Q such that, over T timesteps, discrete inputs (i.e. protein amino-acid sequences for EvoDiff) are iteratively corrupted via a controlled Markov process until they constitute samples from a uniform stationary distribution at time T. This section describes the D3PM process and loss for a single categorical variable x in one-hot format. The forward corruption process is described by:

$$q(x_t|x_{t-1}) = \mathrm{Cat}(x_t; p = x_{t-1}Q_t). \tag{1}$$

This allows for efficient training via efficient computation of $q(x_t|x_0)$ and $q(x_{t-1}|x_t)$. The D3PM approach can emulate a masked modeling process by choosing a transition matrix with an absorbing state (e.g., [MASK]; [20]). However, in this work, the D3PM formulation is only used for discrete corruption because masking corruption via OADM generally outperforms absorbing-state D3PM [19]. EvoDiff includes two discrete corruption schemes: one based on a uniform transition matrix (D3PM-Uniform) and one based on a biologically-informed transition matrix (D3PM-BLOSUM).

EvoDiff-D3PM models are trained via a hybrid loss function

$$\mathcal{L}_\lambda = \mathcal{L}_{vb} + \lambda\mathcal{L}_{ce}. \tag{2}$$

This loss combines a variational lower bound $\mathcal{L}_{vb}$ on the negative log likelihood

$$\mathcal{L}_{vb} = \mathbb{E}_{q(x_0)}[\underbrace{D_{KL}\left[q(x_T|x_0)\|p(x_T)\right]}_{\mathcal{L}_T} + \sum_{t=2}^{T}\underbrace{\mathbb{E}_{q(x_t|x_0)}\left[D_{KL}\left[q(x_{t-1}|x_t,x_0)\|p_\theta(x_{t-1}|x_t)\right]\right]}_{\mathcal{L}_{t-1}} + \underbrace{-\mathbb{E}_{q(x_1|x_0)}\left[\log(p_\theta(x_0|x_1)\right]]}_{\mathcal{L}_0}. \tag{3}$$

and a cross-entropy loss $\mathcal{L}_{ce}$ on $p_\theta(x_0|x_t)$. Investigation of the impact of λ on model performance revealed minimal improvement to sample generation quality when $\lambda > 0$, consistent with the findings of the original D3PM paper [20]. Thus λ=0 and T=500 were used in all D3PM experiments.

$\mathcal{L}_{vb}$ has three terms. $\mathcal{L}_T$ measures whether the corruption reaches the stationary distribution $p(x_T)$ at time T and does not depend on θ. Next consider the remaining two terms $\mathcal{L}_{t-1}$ and $\mathcal{L}_0$, which depend on θ. Following the original D3PM paper, $\tilde{p}_\theta(\tilde{x}_0|x_t)$ is directly predicted by the neural network. To compute the loss at timesteps $0 < t < T$, the terms $q(x_{t-1}|x_t, x_0)$ and $p_\theta(x_{t-1}|x_t)$ must be computed from x_t, x_0, and $\tilde{p}_\theta(\tilde{x}_0|x_t)$ using Markov properties

Defining $\overline{Q}_t = Q_1Q_2\cdots Q_t$:

$$q(x_{t-1}|x_t, x_0) = \mathrm{Cat}\left(x_{t-1}; p = \frac{x_tQ_t^\top \odot x_0\overline{Q}_{t-1}}{x_0\overline{Q}_tx_t^\top}\right) \tag{4}$$

$$p_\theta(x_{t-1}|x_t) \propto \sum_{\tilde{x}_0} q(x_{t-1}, x_t|\tilde{x}_0)\tilde{p}_\theta(\tilde{x}_0|x_t) \tag{5}$$

where $\odot$ represents an element-wise product. For Equation 5 rules of conditional probability and Markov properties are used to define $q(x_{t-1}, x_t|\tilde{x}_0)$ in terms of x_t and $\tilde{x}_0$:

$$q(x_{t-1}, x_t|\tilde{x}_0) = \text{Cat}\left(x_{t-1}; p = x_t Q_t^\top \odot \tilde{x}_0 \overline{Q}_{t-1}\right) \tag{6}$$

Putting everything together, at each step of training a corruption timestep is sampled according to $t \sim \mathcal{U}(1, \ldots, T-1)$. x_t is then sampled via $q(x_t|x_0) \sim \text{Cat}\left(x_t; p = x_0 \overline{Q}_t\right)$ for every residue in the input protein, and the neural network predicts $\tilde{p}_\theta(\tilde{x}_0|x_t)$. Note that, while the corruption and loss are computed independently over each residue, the neural network predicts $\tilde{p}$ in the context of the entire sequence. If $t = 1$, only the loss $\mathcal{L}_0$ is used, reflecting a standard negative log likelihood. Otherwise, Equations 4 and 5 are used to compute the loss $\mathcal{L}_{t-1}$.

Sampling from a trained model begins with the noised $\boldsymbol{x}_T$, where each residue is randomly sampled from a uniform distribution over amino acids. x_{t-1} is then iteratively sampled via $p_\theta(x_{t-1}|x_t)$ as described in Equation 5. For all models, generated sequences are sampled to match the distribution of sequence lengths in the training set, going up to 2048 residues as the maximum length.

EvoDiff-D3PM-Uniform Many strategies exists to schedule corruption in D3PMs. EvoDiff-D3PM-Uniform employs the simplest case – a uniform corruption scheme. Specifically, EvoDiff-D3PM-Uniform models implement a doubly stochastic, uniform transition matrix Q_t with a corruption schedule $(T - t + 1)^{-1}$ from Sohl-Dickstein et al. [3], so that information is linearly corrupted between x_t and x_0 for all $t < T$.

EvoDiff-D3PM-BLOSUM EvoDiff-D3PM-BLOSUM implements a transition matrix derived from BLOSUM62 matrices of amino acid substitution frequencies [35]. BLOSUM matrices are derived from observed alignments across highly conserved regions of protein families and thus provide the relative frequencies of amino acids and their substitution probabilities.

Rows that represent uniform transition probabilities for non-standard amino acid codes (J, O, U) and for $<$ GAP $>$ tokens in the MSA input case are included in addition to standard amino acids. BLOSUM substitution frequencies are converted to a matrix of transition probabilities by performing a Softmax over the frequencies and then normalizing over rows and columns via the Sinkhorn-Knopp algorithm to obtain a doubly stochastic matrix. In this scheme, the gradual corruption of a single sequence to random noise is simulated in a way that prioritizes conserved evolutionary relationships of amino acid mutations. A β-schedule was implemented to taper the number of mutations over time for timesteps up to T=500, specifically via an empirical schedule that corrupts half the sequence content by half of T (t=250) (**Fig. S16**). This schedule was chosen to approximate the linear rate of mutations observed over 500 timesteps in the uniform transition matrix case, shown in Fig. S16b.

Order-Agnostic Autoregressive Diffusion Models (OADMs) Order-agnostic autoregressive diffusion models (OADMs) generalize absorbing-state D3PM and left-to-right autoregressive models (LRARs) [19]. This section describes the OADM process and loss for a sequence $\boldsymbol{x}$ of L categorical variables. In the case of EvoDiff, L is the sequence length.

LRARs factorize a high-dimensional joint distribution $p(\boldsymbol{x})$ into the product of L univariate distributions using the probability chain rule:

$$\log p(\boldsymbol{x}) = \sum_{t=1}^{L} \log p(x_t|\boldsymbol{x}_{<t}) \tag{7}$$

where $\boldsymbol{x}_{<t} = x_1, x_2, \ldots x_{t-1}$. LRARs are typically parametrized using a triangular dependency structure, such as causal masking in a transformer or CNN, in order to allow parallelized computation of all the conditional distributions in the likelihood during training. LRARs learn to generate sequences in a pre-specified left-to-right decoding order, which may be non-obvious for modalities such as proteins and does not allow conditioning on arbitrary fixed subsequences.

LRARs can be expanded into a diffusion framework via two subtle changes. Following the exposition in Hoogeboom *et al.,* [19], the first change is to allow order-agnostic decoding. In an order-agnostic autoregressive model, a decoding order σ is first sampled uniformly from all possible decoding orders S_L. At time step t in the forward process, $x_{\sigma(L-t)}$ is masked. The log-likelihood for an

order-agnostic autoregressive model is derived using Jensen's inequality:

$$\log p(\boldsymbol{x}) = \log \mathbb{E}_{\sigma \sim \mathcal{U}(S_L)} p(\boldsymbol{x}|\sigma) \geq \mathbb{E}_{\sigma \sim \mathcal{U}(S_L)} \log p(\boldsymbol{x}|\sigma)$$
$$\geq \mathbb{E}_{\sigma \sim \mathcal{U}(S_L)} \sum_{t=1}^{L} \log p(x_{\sigma(t)}|\boldsymbol{x}_{\sigma(<t)})$$

The next change involves an objective that optimizes over arbitrary decoding orders one timestep at a time in the style of modern diffusion models, without requiring a neural network that enforces a triangular or causal dependency structure. This is accomplished by replacing the summation over t by an expectation that is appropriately re-weighted.

$$\begin{aligned}
\log p(\boldsymbol{x}) &\geq \mathbb{E}_{\sigma \sim \mathcal{U}(S_L)} \sum_{p=1}^{L} \log p(x_{\sigma(t)}|\boldsymbol{x}_{\sigma(<t)}) \\
&= \mathbb{E}_{\sigma \sim \mathcal{U}(S_L)} L \cdot \mathbb{E}_{t \sim \mathcal{U}(1,\ldots,L)} \log p(x_{\sigma(t)}|\boldsymbol{x}_{\sigma(<t)}) \\
&= L \cdot \mathbb{E}_{t \sim \mathcal{U}(1,\ldots,L)} \mathbb{E}_{\sigma \sim \mathcal{U}(S_L)} \frac{1}{L-t+1} \sum_{k \in \sigma(\geq t)} \log p(x_k|\boldsymbol{x}_{\sigma(<t)})
\end{aligned}$$

The overall expected log likelihood $\log p(\boldsymbol{x})$ can be thought of according to a series of likelihoods, each captured in the loss at step t, $\mathcal{L}_t$:

$$\mathcal{L}_t = \frac{1}{L-t+1} \mathbb{E}_{\sigma \sim \mathcal{U}(S_L)} \sum_{k \in \sigma(\geq t)} \log p(x_k|\boldsymbol{x}_{\sigma(<t)}). \tag{8}$$

Thus, the overall expected log likelihood is lower bounded as:

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{t \sim \mathcal{U}(1,\ldots,L)} [L \cdot \mathcal{L}_t] \tag{9}$$

A neural network can be efficiently trained to learn the reverse process $p_\theta(x_{\sigma(t)}|\boldsymbol{x}_{\sigma(<t)})$ by randomly masking a set of t tokens at each iteration and minimizing the reweighted loss, allowing the model to learn from predictions of all masked positions at each timestep. By learning one model over all possible decoding orders, OADM allows for conditioning by fixing arbitrary subsequences at generation time. Sequences were generated unconditionally from OADM models by beginning with an all-mask sequence as input, randomly sampling a decoding order, and sampling each token from the predicted probability distribution.

Left-to-right autoregressive and masked language models are diffusion models The connection between autoregressive models and diffusion models has been described previously [19, 20]. Left-to-right autoregressive (LRAR) diffusion models implement a masked modeling process that is akin to a process which iteratively and deterministically masks all tokens to the right of the sampled token x_t, where the current diffusion timestep t is equivalent to the number of tokens masked over the entire sequence length, with all tokens masked at the final timestep $T = L$.

Likewise, masked language models (MLMs) are equivalent to only learning one step t of OADM:

$$\mathcal{L}_{\text{MLM}} = \frac{1}{L-t+1} \mathbb{E}_{\sigma \sim \mathcal{U}(S_L)} \sum_{k \in \sigma(\geq t)} \log p(x_k|\boldsymbol{x}_{\sigma(<t)}). \tag{10}$$

Thus, the OADM setup generalizes LRAR models by considering all possible decoding orders rather than left-to-right decoding, while the MLM learning task is equivalent to only training on one step of the OADM diffusion process.

Datasets Sequence-only EvoDiff models were trained on UniRef50 [21] which contains approximately 45 million protein sequences. The UniRef50 release and train/validation/testing splits from CARP [22] were used to facilitate comparisons between models. Sequences longer than 1024 residues were randomly subsampled to 1024 residues. Multiple sequence alignment (MSA) EvoDiff models were trained on OpenFold [31], which contains 401,381 MSAs for 140,000 unique Protein Data Bank (PDB) chains and 16,000,000 UniClust30 clusters. To construct the MSAs used to train EvoDiff,

lowercase characters were removed to restore the alignments, as the queries do not contain gap characters. Next, MSAs that contained sequences with more than 512 consecutive $<$ GAP $>$ tokens as well as MSAs that contained fewer than 64 sequences per alignment were filtered out. This filtering resulted in 382,296 total MSAs, which were then randomly split into 372,296 training and 10,000 validation MSAs.

MSA subsampling for training EvoDiff-MSA models To optimize for memory constraints during training, MSAs were subsampled to 64 sequences and a maximum sequence length of 512. MSAs shorter than 512 sequences were padded to a sequence length of 512, but MSAs containing fewer than 64 sequences were excluded from training. For MSAs with more than 64 sequences, two subsampling schemes were implemented: random ("Rand.") and MaxHamming ("Max"). The random subsampling scheme ("Rand.") randomly samples 64 sequences from the MSA, making sure that the reference/query sequence (i.e. the first sequence) is always included. The "Max" subsampling scheme greedily selects for sequence diversity in the 64 sequence subset by iteratively selecting the sequence that maximizes the minimum Hamming distance to the sequences already selected. The Hamming distance measures the distance between two sequences, denoted by the number of amino acids that differ between aligned sequences. Subsampling to maximize the Hamming distance enabled input of an MSA rich with evolutionarily diverse sequences to EvoDiff-MSA models.

Modeling, architecture, and training details For sequences, the EvoDiff denoising model adopts a ByteNet-style CNN architecture [36] previously shown to perform similarly to transformers for protein sequence masked language modeling tasks [22]. All models are implemented in PyTorch [37]. In EvoDiff-OADM models, the diffusion timestep is implicitly encoded in the number of masked positions. EvoDiff-D3PM models use a 1D sinusoidal encoding [38] to denote the timestep for each input. All sequence models were trained with the Adam optimizer [39], a learning rate of 1e-4 with linear warmup over 16,000 steps, and dynamic batching to optimize GPU usage. EvoDiff's small sequence models implement a ByteNet-style architecture with ca. 38M parameters. Large models were scaled to a ByteNet architecture of ca. 640M parameters by increasing the model dimension d from 1020 to 1280, increasing the encoder hidden dimension from $d/2$ to d, and increasing the number of layers from 16 to 56.

38M parameter models were trained on 8 32GB NVIDIA V100 GPUs; 640M parameter models were trained on 32 (2x16) 32GB NVIDIA V100 GPUs. The maximum number of tokens per GPU in each batch was reduced from 40,000 to 6,000 to accommodate training the larger 640M parameter models. 38M parameter models were trained for approximately 2 weeks and saw ca. 3e14 tokens over 700,000 training steps. 640M parameter models were trained for as long as computationally feasible to achieve the best results possible; models saw between ca. 1e10 and 1e17 tokens over ca. 400,000-2,000,000 training steps. The D3PM-BLOSUM model stopped improving after approximately 12 days of training. The D3PM-Uniform and OADM models were trained for 23 days without reaching convergence.

For MSAs, the EvoDiff denoising model adopts a 100M parameter MSA Transformer architecture [30]. As with the single sequence models, EvoDiff-MSA-OADM models implicitly encode the diffusion timestep; EvoDiff-MSA-D3PM models include an additional sinusoidal timestep embedding. All MSA models were trained with the Adam optimizer with a learning rate of 1e-4 and linear warmup over 15,000 steps. EvoDiff MSA models were trained on 16 32GB NVIDIA V100 GPUs for 10 days and saw ca. 3e9 tokens over 55,000 training steps.

Baseline models To enable direct comparison, the left-to-right autoregressive (LRAR) and CARP baselines were trained with the same CNN architectures on the same dataset as EvoDiff sequence models. For LRAR, the convolution modules have a causal mask to prevent information leakage. For additional MLM baselines, sequences were sampled from the protein MLMs ESM-1b [40] and ESM-2 [23], which were trained on different releases of UniRef50. ESM-1b and ESM-2 both generated many "unknown" amino acids (`X`); performance was improved results by manually setting the logits for `X` to `inf`. Sequences were sampled from MLMs by treating the MLM as an OADMs and beginning from an all-mask state. For the structure-based diffusion baselines, sequences were obtained from FoldingDiff [8] and RFdiffusion [10] by first unconditionally generating structures and then using ESM-IF [26] to design their sequences. For MSA baselines, new query sequences were generated from ESM-MSA [30] by treating it as an OADM and sampling from an all-mask starting

query sequence. CCMgen [41] with default parameters was used to train and generate from Potts models of validation MSAs from OpenFold.

Computation of test-set perplexities Perplexity was calculated by uniformly sampling a timestep for each test sequence, corrupting the sequence according to each diffusion model, predicting the sequence x_0 at $t = 0$ by passing inputs once through each trained model, and then computing the perplexity. For D3PM models, the perplexity is:

$$\text{Perp}_{\text{D3PM}} = \mathbb{E}_{t\sim\mathcal{U}(1,\ldots,T)} \exp\left(-\frac{1}{L}\sum_{i}^{L} \log p_\theta(x_0|x_t)\right) \tag{11}$$

For OADMs, the perplexity is:

$$\text{Perp}_{\text{OADM}} = \cdot\mathbb{E}_{t\sim\mathcal{U}(1,\ldots,L)}\mathbb{E}_{\sigma\sim\mathcal{U}(S_D)} \exp\left[\frac{-1}{L-t+1}\sum_{k\in\sigma(\geq t)} \log p(x_k|\boldsymbol{x}_{\sigma(<t)})\right] \tag{12}$$

To enable model comparison, perplexities for MLMs (CARP, ESM-1b, ESM-2) were computed as if they are OADMs.

And for LRAR models, the perplexity is:

$$\text{Perp}_{\text{LRAR}} = \exp\left[-\sum_{t=1}^{L} \log p(x_t|\boldsymbol{x}_{<t})\right] \tag{13}$$

Calculated D3PM perplexities were on average higher as $t \to T$ and lower as $t \to 1$, and masked perplexities were similarly higher for a greater number of masked tokens per sequence, i.e., as $t \to L_{masked}$ (**Fig. S1, S11**). Lower perplexities indicated improved performance and generalization capacity.

Evaluation of structural plausibility The structural plausibility pipeline (**Fig. 2A**) evaluates both the foldability and self-consistency of a given sequence. Foldability was evaluated by averaging the per-residue confidence score, reported as pLDDT by OmegaFold, across the entire sequence. Sequence self-consistency, denoted scPerpelxity, describes how likely the generated sequence is to correspond to the predicted structure. Self-consistency was measured by taking structures predicted for a sequence from OmegaFold, running them through ESM-IF, and calculating the perplexity between the ESM-IF predicted-sequence and the original generated sequence.

The novelty of generated sequences was evaluated relative to training data seen by the model, by computing the Hamming distance between each generated sequence and every training-set sequence of the same sequence length. The minimum of these Hamming distances, representing the closest sequence seen by the model during training, was reported for each sequence.

Computation of functional and structural features To evaluate sequence coverage, ProtT5 embeddings were computed for each of 1,000 generated protein sequences and 10,000 sequences sampled from the test set using the Tools from Protein Prediction for Interpretation of Hallucinated Proteins (PPIHP) package [42]. The resulting distributions of sequence embeddings (i.e., representing the corresponding distributions of sequences) were compared via the Fréchet ProtT5 distance (FPD),

$$\text{FPD} = ||\mu_{test} - \mu_{gen}||^2 + \text{Tr}(C_{test} + C_{gen} - 2\sqrt{C_{test}C_{gen}}) \tag{14}$$

where, given the embedding space feature vectors for the test and generated distributions, μ is the feature-wise mean for each set of sequences, C is the respective covariance matrix, and Tr refers to the trace linear algebra operation, defined as the sum of the elements along the main diagonal of a square matrix. Embeddings were visualized in 2D via uniform manifold approximation and projection (UMAP), fit to the test data and with `n_neighbors`=25. The number of neighbors hyperparameter was selected to favor local similarities in place of global ones, in order to appropriately visualize the corresponding differences in embedding space and FPD measured for each model.

Structural features of generated sequences were evaluated via the ProtTrans [27] CNN predictor model to assign a 3-state secondary structure definition from DSSP (helix, strand, or other) to each residue in a protein. The fraction of predicted 'helix', 'strand', or 'other' was computed (the three values sum to 1 per sequence). The resulting multivariate distributions of secondary structure features (computed over 1000 generated or natural sequences) were visualized via kernel density estimation. The KL divergence between the mean values across the 3-state predictions for the generated and test sets was used to quantitatively evaluate the distribution of secondary-structures assigned for each model.

Evolution-guided generation with EvoDiff-MSA Starting with either a random or MaxHamming subsampled MSA, new query sequences were generated by sampling from an all-mask starting query sequence. The generated query sequence was evaluated relative to the corresponding original query sequence using the same tools and workflow described in *Evaluation of structural plausibility*. Each generated sequence was additionally evaluated for similarity relative to its reference MSA, which is comprised of a query sequence and alignment sequences. The % similarity of each generated sequence relative to its parent MSA was computed as the maximum % similarity over all sequences in the original MSA. Specifically, for a pair of sequences, the % similarity was computed by calculating the number of shared residue identities (accounting for both amino-acid identity and position index in the sequence), and for a given generated sequence the maximum value of these % similarities was determined. Across generated sequences both the CDF and mean of maximum % similarity were reported. Generated sequences were additionally evaluated for structural similarity relative to their original query sequences. Structures were predicted for each of the generated query sequences and the original query sequences using OmegaFold. Structural similarity was measured via the template modeling score (TM-score) [43] for the two predicted structures following structural alignment:

$$\text{TM-score} = \max\left[\frac{1}{L_{\text{gen}}}\sum_{i}^{L_{\text{common}}}\frac{1}{1+\left(\frac{d_i}{d_0(L_{\text{true}})}\right)^2}\right] \tag{15}$$

where L_{gen} is the length of the generated query sequence; L_{common} is the number of shared residues; d_i is the distance between the i^{th} pair of residues; L_{true} is the length of the true query sequence; and $d_0(L_{\text{true}}) = 1.24\sqrt[3]{L_{\text{true}} - 15} - 1.8$ is a distance scale for normalization.

Generation of intrinsically disordered regions (IDRs) IDR generation and analysis leveraged a publicly available dataset of 15,996 human IDRs and their orthologs [44]. This dataset was generated by running SPOT-Disorder v1 [45] on the human proteome and applying the predicted IDR positions to an MSA of likely-similar-function orthologs (determined using an evolutionary distance heuristic), curated from the larger set of orthologs contained in the OMA database [46]. The resulting dataset only contained IDRs, not full protein sequences, and thus IDR sequences were mapped back to the MSAs of full protein sequences in OMA in order to provide context about the sequence regions surrounding the IDRs.

For input to EvoDiff models, the full sequence of an IDR-containing human protein was treated as the query sequence, and a corresponding MSA was constructed by subsampling 63 other sequences from all the query's orthologs. All sequences were subsampled to 512 residues in length, with the following criteria maintained. Subsampling criteria were that the subsampled query sequence contain at least 1 IDR, and that the total IDR region was less than half the total length of the subsampled sequence ($L_{\text{IDR}} \leq 256$). For IDR generation from EvoDiff-Seq, the query sequence with the IDR region masked was provided as the only input to EvoDiff-Seq, which then generated new residues for the masked region (i.e., the region corresponding to the true IDR). For IDR generation from EvoDiff-MSA, the query sequence with the IDR region masked, aligned to the rest of the MSA, was provided as input to EvoDiff-MSA, which then generated new residues for the masked region.

The resulting generations, containing putative IDRs, were input to DR-BERT, a protein language model fine-tuned for disorder prediction [47], to obtain per-residue disorder scores ranging from 0-1 (less to more disordered). A single-sequence IDR predictor (DR-BERT) was used in place of MSA-based IDR scoring methods, because of an observed bias towards higher disorder scores with MSA-based methods – e.g., random uniform sampling of residues in the masked query positions still resulted in a prediction of disorder given the presence of the orthologs in the alignment. Disorder scores for true IDRs, generated IDRs, scrambled IDRs, and randomly generated IDRs were computed to

evaluate the performance of DR-BERT predictions. The randomly-sampled baseline was constructed by randomly sampling amino acids over an IDR region; the scrambled baseline was constructed by shuffling the existing amino acids over an IDR region into a scrambled permutation. In all cases (true IDRs, generated IDRs, scrambled and random baselines), the entire protein sequence was input to DR-BERT for scoring. Since DR-BERT is for single-sequences, for putative IDRs generated by EvoDiff-MSA, the entire query sequence was inputted into DR-BERT, with $<$ GAP $>$ tokens eliminated, to obtain per-residue disorder scores. Lastly, a direct comparison between the original IDR and the generated putative IDR was conducted by calculating the % sequence similarity between the fraction of shared residues between the two IDR regions.

Motif scaffolding Scaffolding performance was evaluated on a recently published benchmark [10] of 25 scaffolding problems across 17 unique proteins.

In our scaffolding benchmark, each unique protein was treated as 1 example, for a total of 17 unique scaffolding examples. 100 samples were generated for each unique scaffolding example. For proteins 6E6R, 6EXZ, 7MRX, and 5TRV, which were the 4 examples evaluated at 3 different scaffolding lengths in RFdiffusion [10], the number of successes across these three different scaffolding lengths were averaged to facilitate comparisons between RFDiffusion and EvoDiff.

To generate a scaffold with EvoDiff-Seq, a scaffold length between 50-100 residues (exclusive of the motif) was sampled uniformly; the motif was placed randomly within the length; and scaffold residues were generated from EvoDiff-Seq conditioned on the provided motif residues. In this approach, on average, protein sequences generated by EvoDiff-Seq were longer (between 45 and 194 residues in length) than those inverse-folded from structures generated by RFdiffusion, which range from 30-152 residues in total length inclusive of the length of the motif.

For scaffolding with EvoDiff-MSA, MSAs for each sequence corresponding to the original PDB structure were generated using the tools from AlphaFold [48] and then subsampled to 64 sequences, and a maximum of 150 residues in length, where the original sequence obtained from the PDB crystal structure was assigned as the query sequence. In cases where the scaffolding examples were shorter than 150 residues, sequences were padded with a $<$ GAP $>$ token, to allow EvoDiff to generate longer-scaffolds. Sequences generated by EvoDiff-MSA were between 56 and 150 residues in length, inclusive of the motif and scafffold. For each scaffolding example, a common set of 100 subsampled MSAs, where 50 were randomly subsampled and 50 were subsampled via MaxHamming, was used commonly across EvoDiff-MSA (Max), EvoDiff-MSA (Random), and ESM-MSA. That is, an individual generation trial for each model corresponded to a unique MSA from the common set of 100 MSAs constructed for a scaffolding example. At inference time, all non-motif residues in the query sequence were masked, and new residues in these locations were generated by EvoDiff-MSA.

OmegaFold was used to predict structures corresponding to sequences generated by EvoDiff. A generation was counted as 'successful' if its predicted structure had a pLDDT ≥ 70 and a motifRMSD ≤ 1.0Å relative to the original motif crystal structure. Note that these success criteria are cutoffs proposed by structure-based models [10] and adopted here to facilitate comparison. The motifRMSD was computed as the RMSD between the alpha-carbons of the motif in the original crystal structure and the predicted structure for the scaffolded motif.

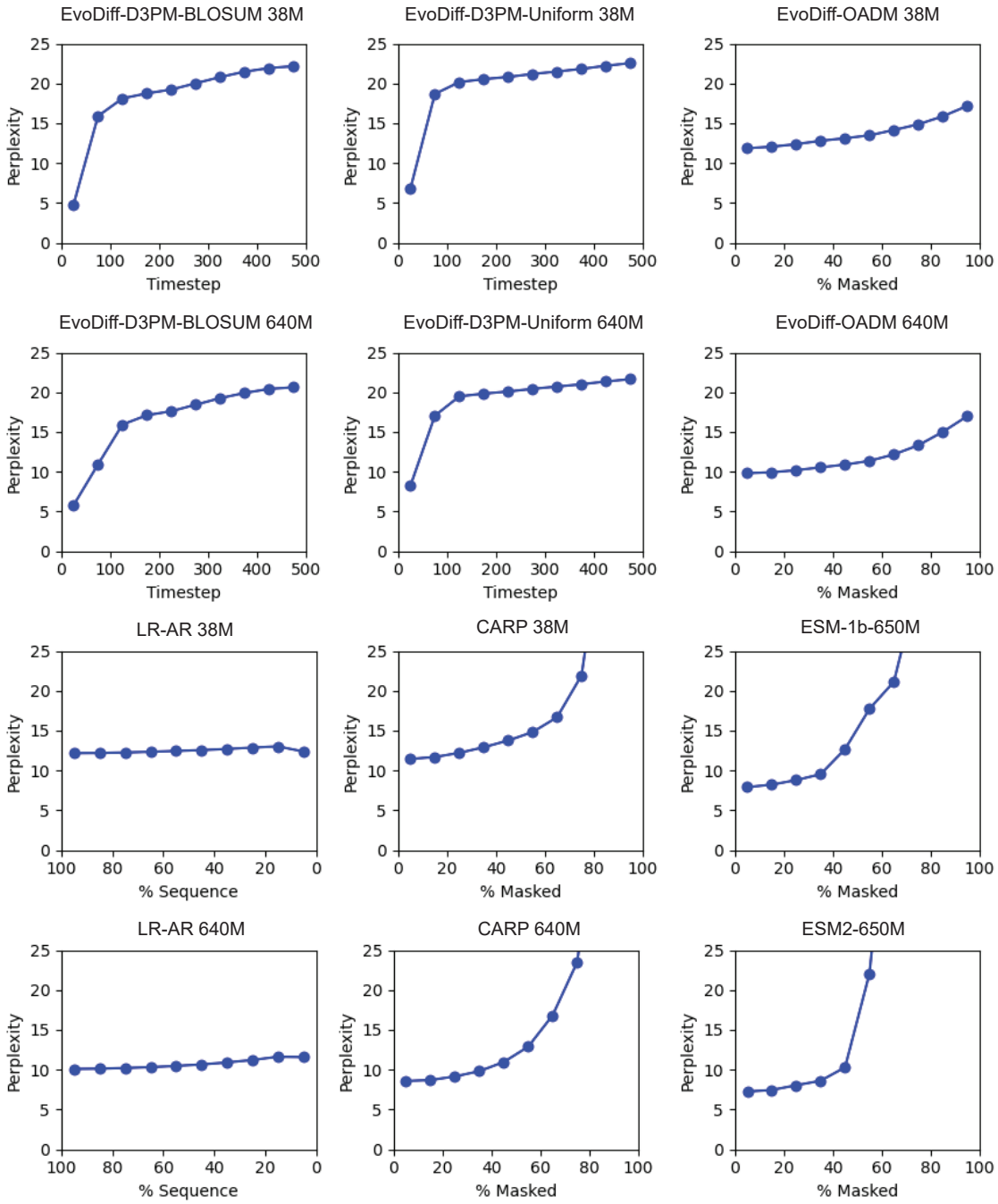

Figure S1: **Perplexity as a function of corruption step for EvoDiff sequence models.** Test-set perplexities at sampled intervals of the degree of corruption, specifically the diffusion timestep for D3PM models, the fraction of masked residues for OADM and masked language models, and the fraction of evaluated sequence for LRAR models. Intervals reflect evenly spaced windows of 50 timesteps for D3PM models or 10% masking for masked models.

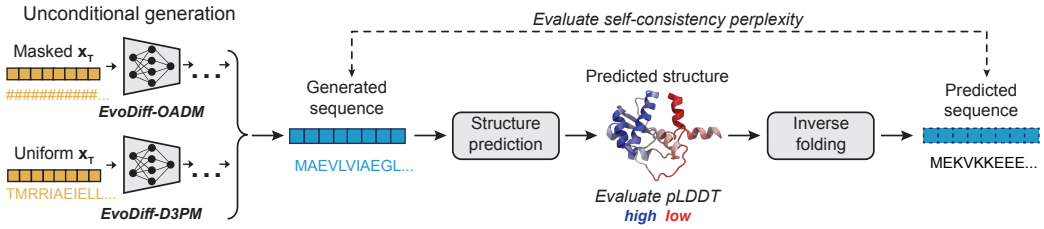

Figure S2: **Workflow for evaluating the foldability and self-consistency of sequences generated by EvoDiff sequence models.**

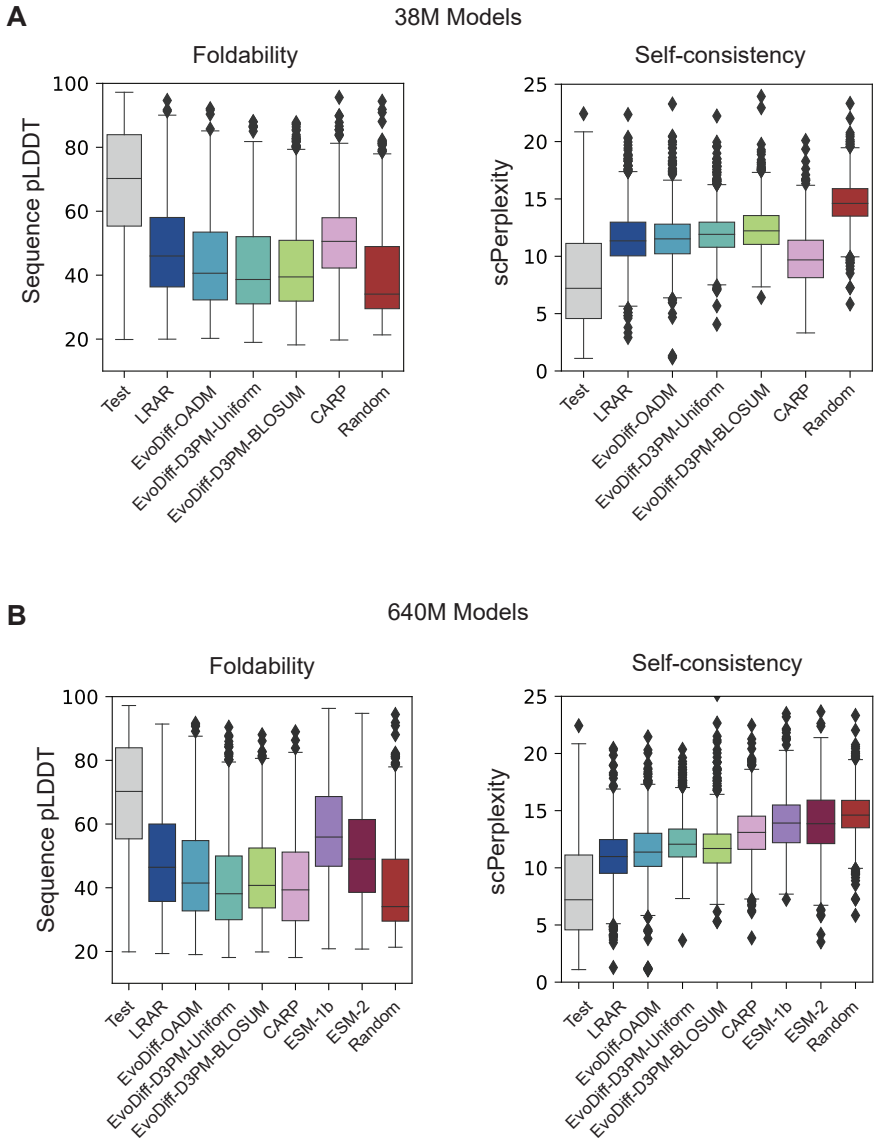

Figure S3: **Summary statistics for structural plausibility metrics for sequence models. (A-B)** Distribution of pLDDT and scPerplexity metrics for sequences from the test set, 38M parameter EvoDiff and baseline models (A), and 640M parameter EvoDiff and baseline models (B) (n=1000 sequences per model). Test and Random baselines are reproduced in (A) and (B) for reference.

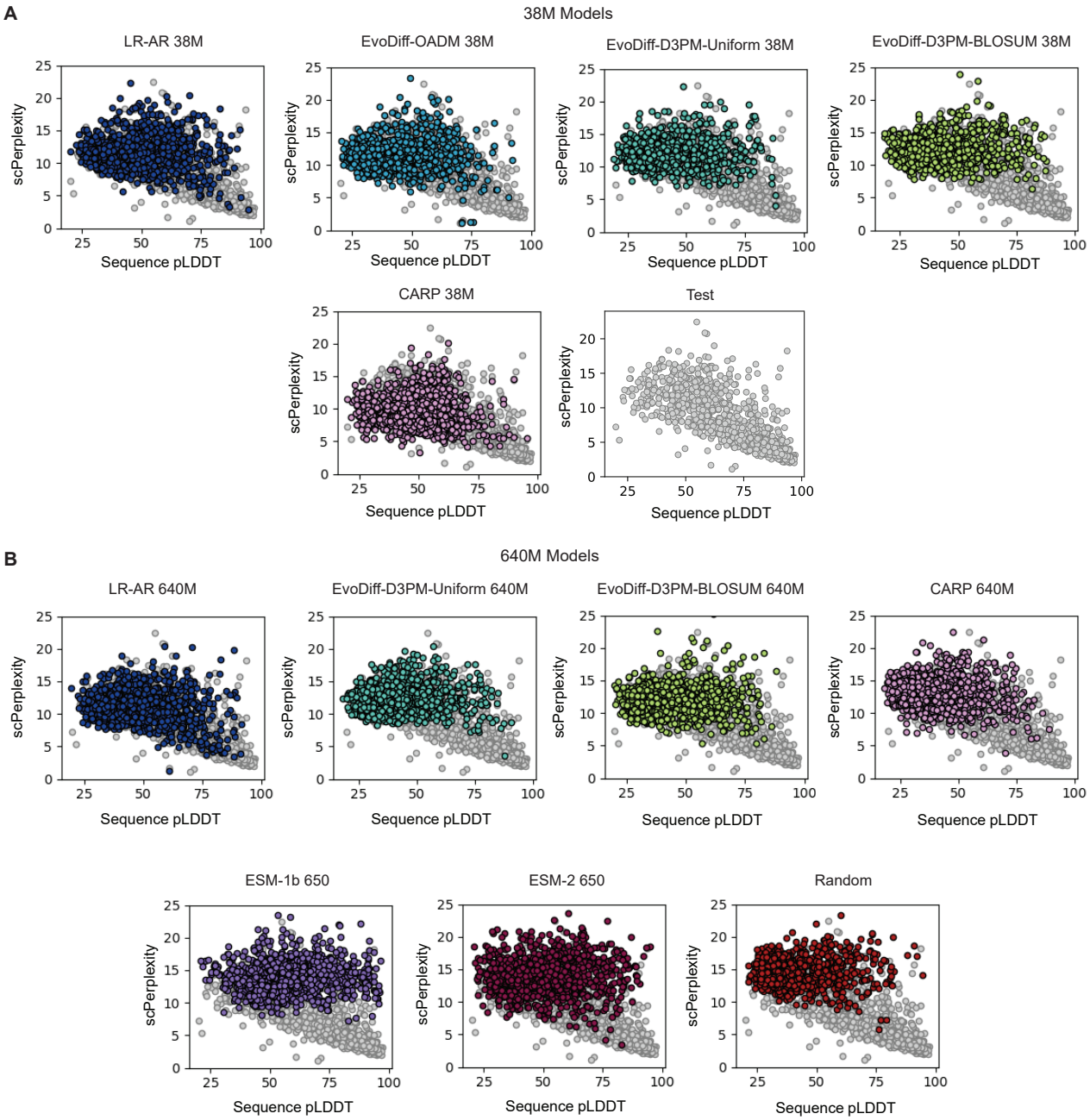

Figure S4: **Sequence pLDDT versus self-consistency perplexity for EvoDiff sequence models. (A-B)** Results for sequences from 38M parameter EvoDiff, baseline models, and test data plotted alone (A), and 640M parameter EvoDiff and baseline models (B) (various colors, n=1000), except for EvoDiff-OADM-640M (EvoDiff-Seq, shown in Fig. 2C), relative to sequences from the test set (grey, n=1000). The self-consistency perplexity (ESM-IF Perplexity) is computed using sequences inverse-folded by ESM-IF.

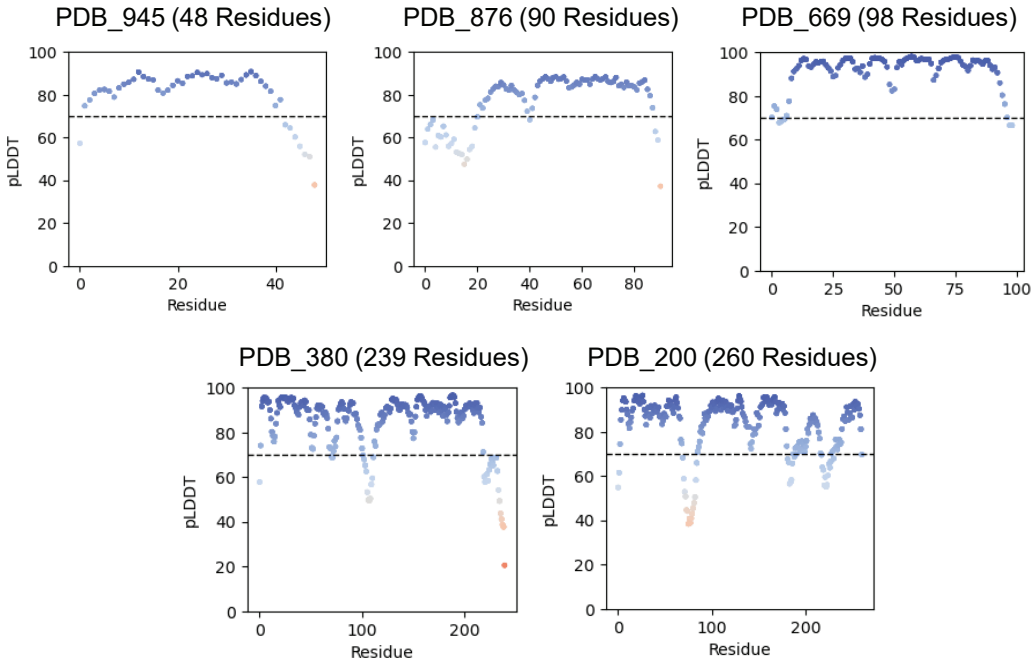

Figure S5: **Per-residue pLDDT for representative proteins generated by EvoDiff-Seq.** pLDDT scores computed based on the OmegaFold predicted structures, for individual residues in representative high-fidelity generations from EvoDiff-Seq (Fig. 2E). Points are colored by pLDDT (0-100, red to blue).

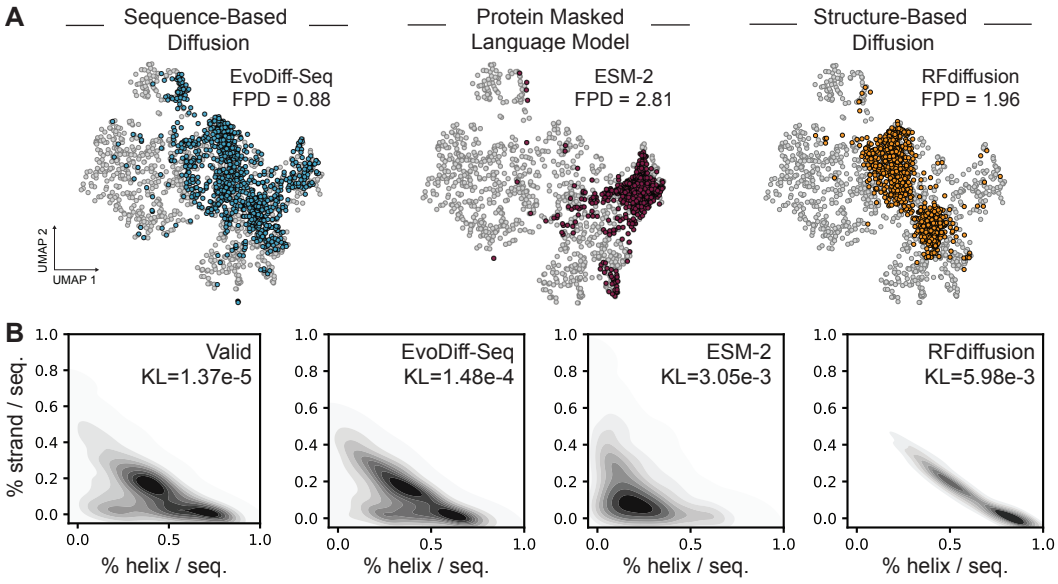

Figure S6: **Generated protein sequences capture natural distributions of protein functional and structural features.** **(A)** UMAP of ProtT5 embeddings, with Fréchet ProtT5 distance (FPD), of natural sequences from the test set (grey, n=1000) and of generated sequences from EvoDiff-Seq (blue, n=1000) and ESM-2 (red, n=1000), and inferred sequences inverse-folded from structures from RFdiffusion (orange, n=1000). **(B)** Distributions of structural features from DSSP 3-state predictions of generations (n=1000), and their Kullback-Leibler (KL) divergence relative to the test set.

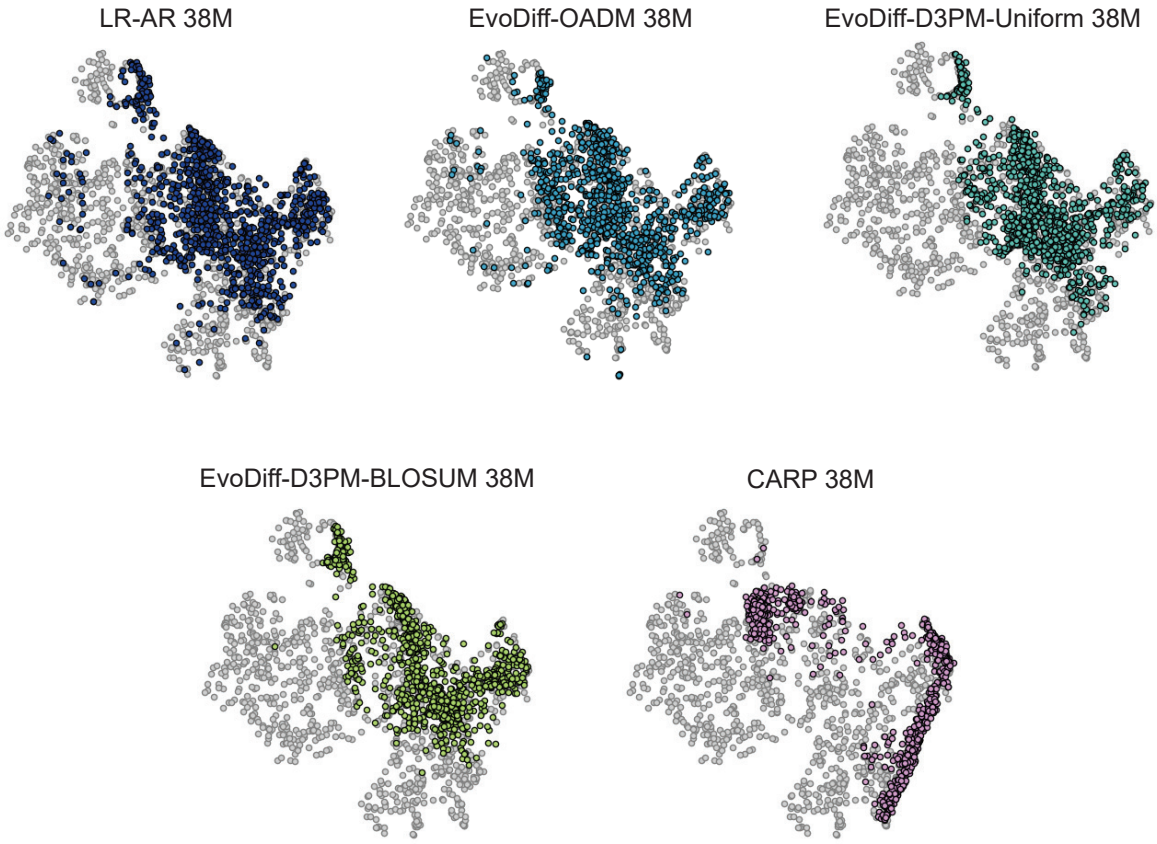

Figure S7: **Coverage of sequence and functional space for generated distributions from 38M parameter EvoDiff sequence models and baselines.** UMAP of ProtT5 embeddings, annotated with FPD, of natural sequences from test set (grey, n= 1000 plotted) and of generated sequences from EvoDiff 38M parameter models and baselines (various colors, n=1000).

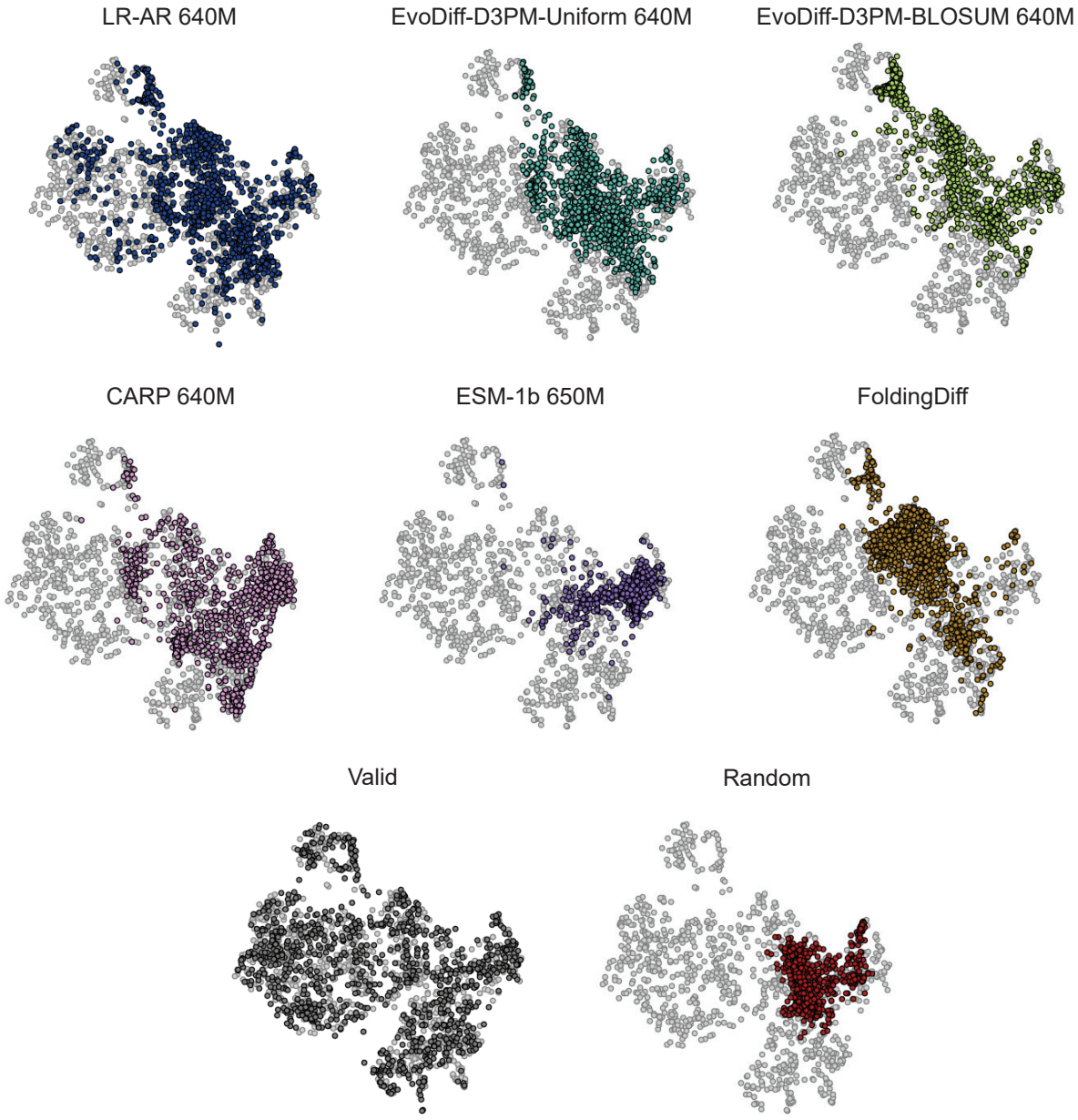

Figure S8: **Coverage of sequence and functional space for generated distributions from 640M parameter EvoDiff sequence models and baselines.** UMAP of ProtT5 embeddings, annotated with FPD, of natural sequences from test set (grey, n= 1000) and of generated sequences from EvoDiff 640M parameter models and baselines (various colors, n=1000). A visualization of sequences from the validation set (dark grey, n=1000) is included for reference. The visualization for the 640M OADM model is excluded due to inclusion in Fig. S6A.

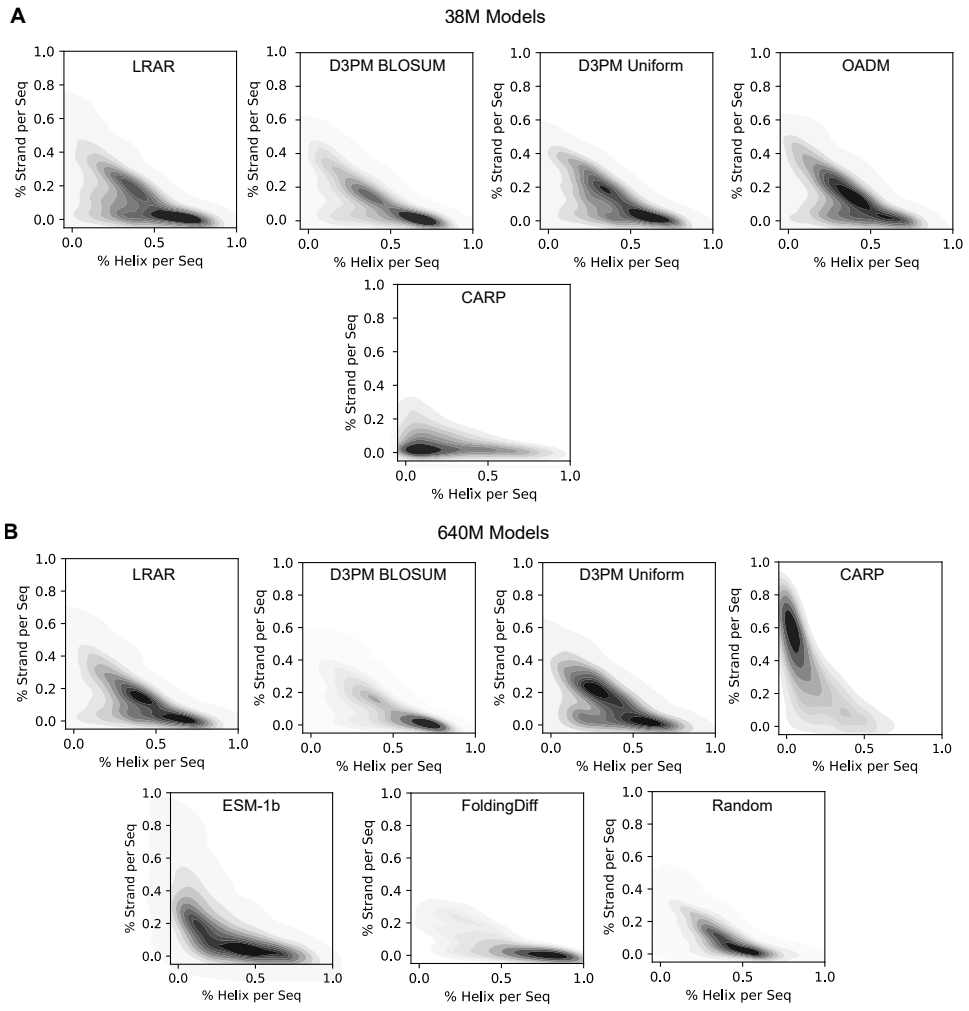

Figure S9: **Structural features in generated sequences from all sequence models. (A-B)** Multivariate distributions of helix and strand features in sequences from 38M (A) and 640M (B) parameter models, and baselines based on DSSP 3-state predictions and annotated with the KL divergence relative to the test set (n=1000 samples from each model). In (B), the distribution for the 640M OADM model is excluded (see Fig. S6B); the distribution for random sequences (n=1000) is provided as reference.

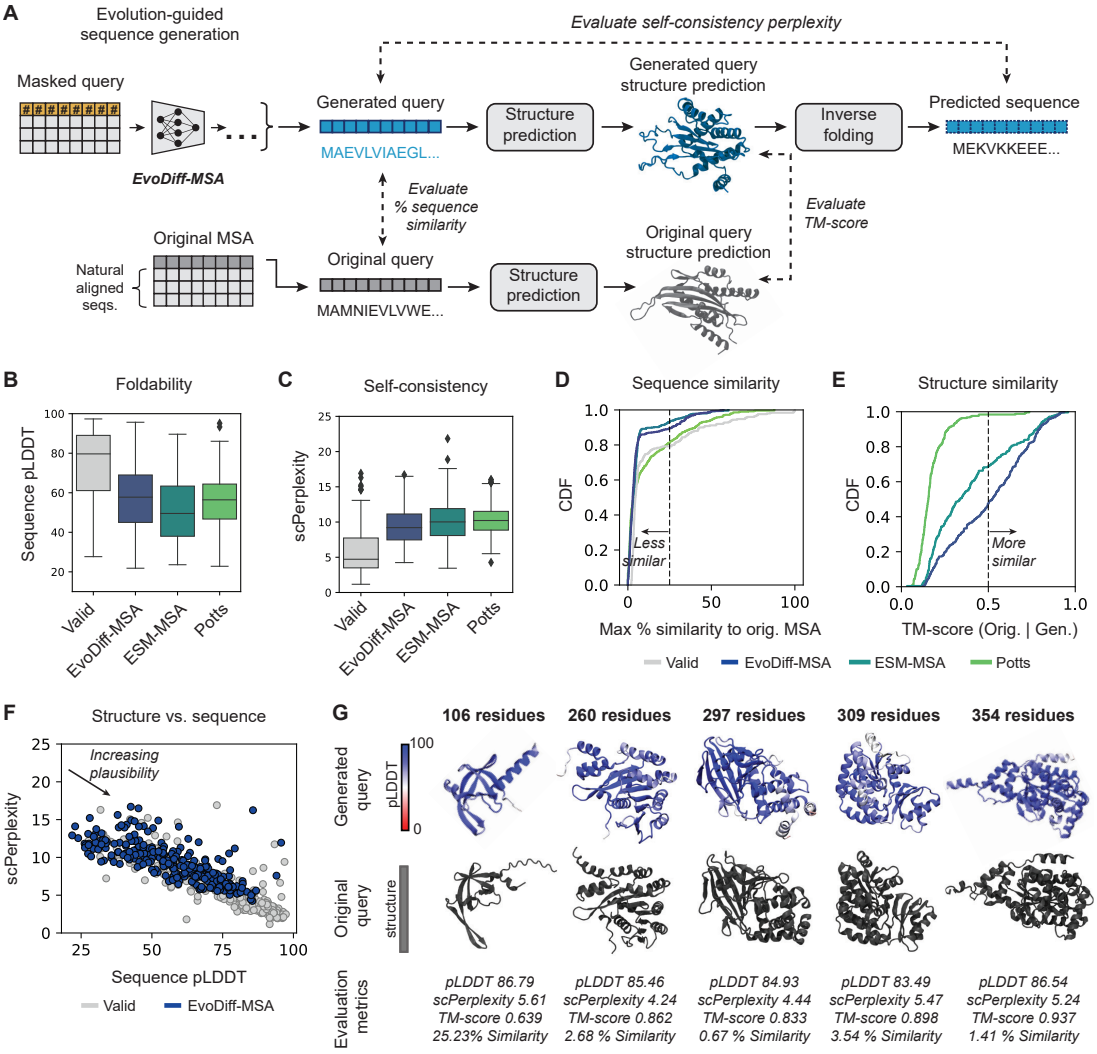

Figure S10: **EvoDiff-MSA enables evolution-guided sequence generation**. **(A)** A new sequence is generated from EvoDiff-MSA via diffusion over only the query component. Generations are evaluated for diversity and self-consistency and for the quality and consistency of their predicted structures. **(B-E)** Distributions of pLDDT (B), scPerplexity (C), sequence similarity (D; dashed line at 25%), and TM-score (E; dashed line at 0.5) for sequences from the validation set, EvoDiff-MSA, ESM-MSA, and a Potts model (n=250 sequences per model; box plots show median and interquartile range). **(F)** Sequence pLDDT versus scPerplexity for sequences from the validation set (grey, n=250) and EvoDiff-MSA (blue, n=250). **(G)** Predicted structures and metrics for structurally plausible generations from EvoDiff-MSA.

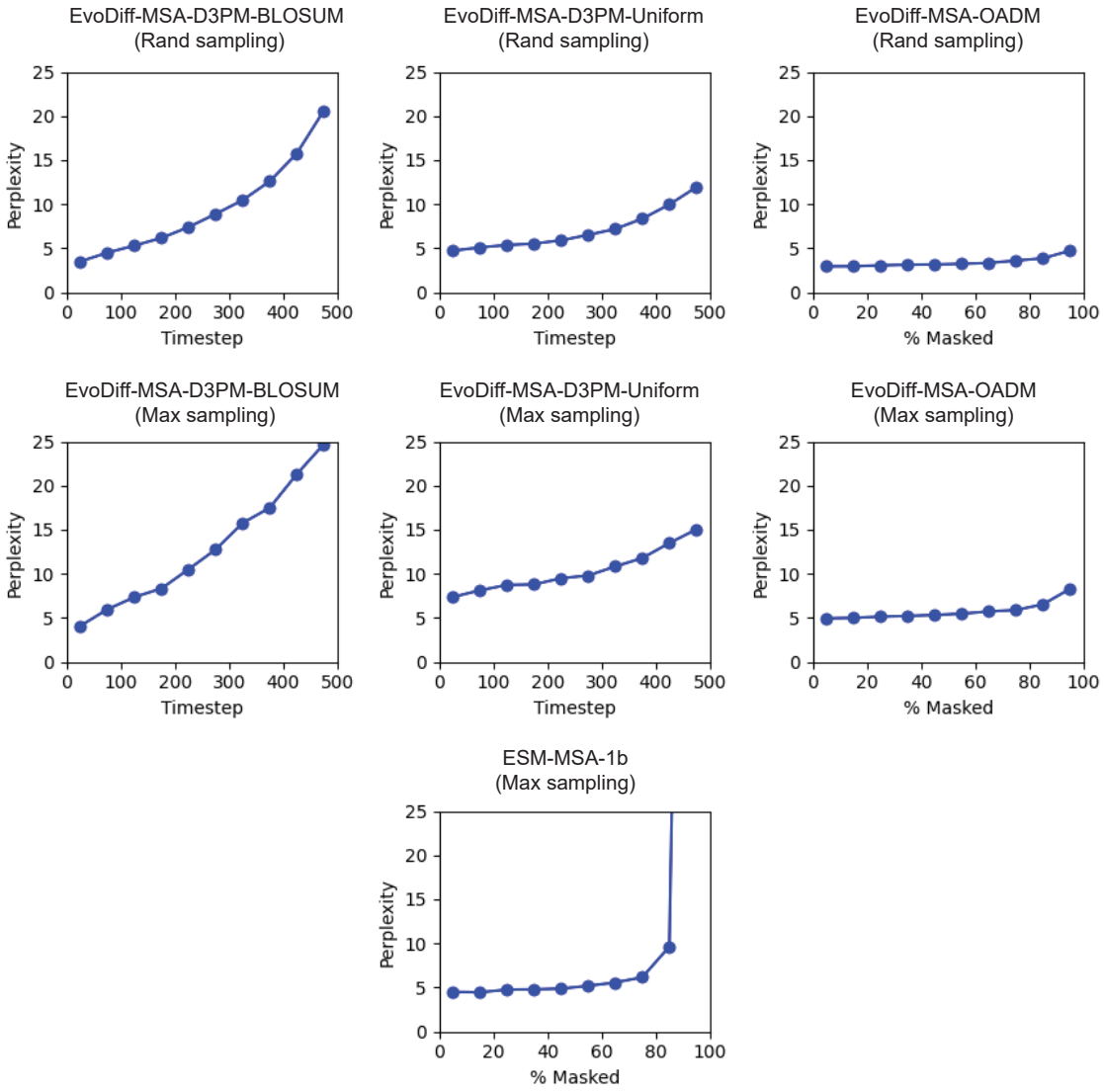

Figure S11: **Perplexity as a function of corruption step for EvoDiff MSA models.** Test-set MSA perplexities at sampled intervals of the degree of corruption, specifically the diffusion timestep for D3PM models and the fraction of masked residues for OADM and ESM models. The test-set evaluated for each model was sampled using the same sampling scheme assigned during training. Intervals reflect evenly spaced windows of 50 timesteps for D3PM models or 10% masking for masked models.

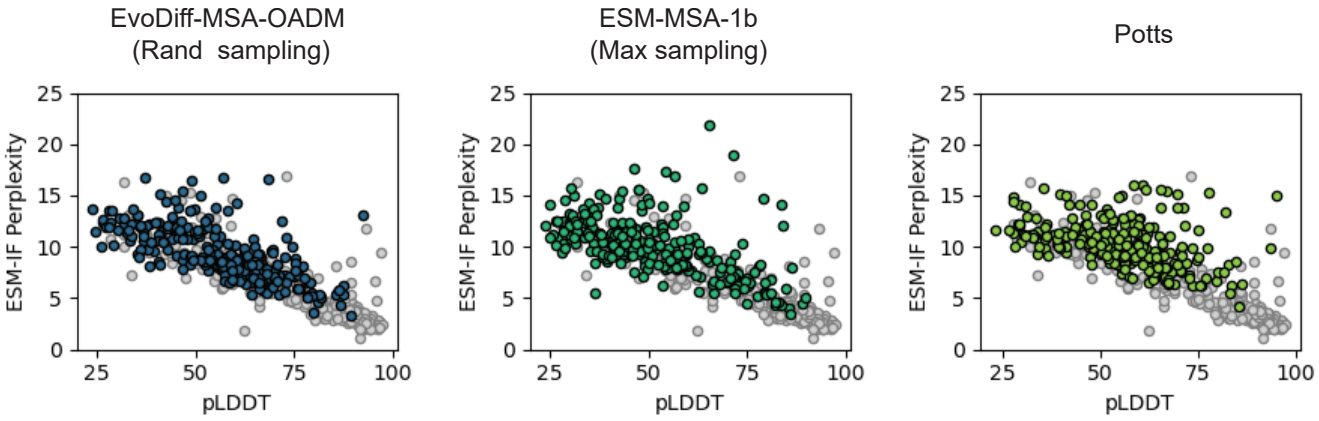

Figure S12: **Sequence pLDDT versus scPerplexity for EvoDiff MSA models,** for sequences from the validation set (grey, n=250) and evaluated MSA models (various colors, n=250), except for EvoDiff-OADM-MSA-Max (EvoDiff-MSA, shown in Fig. S10F).

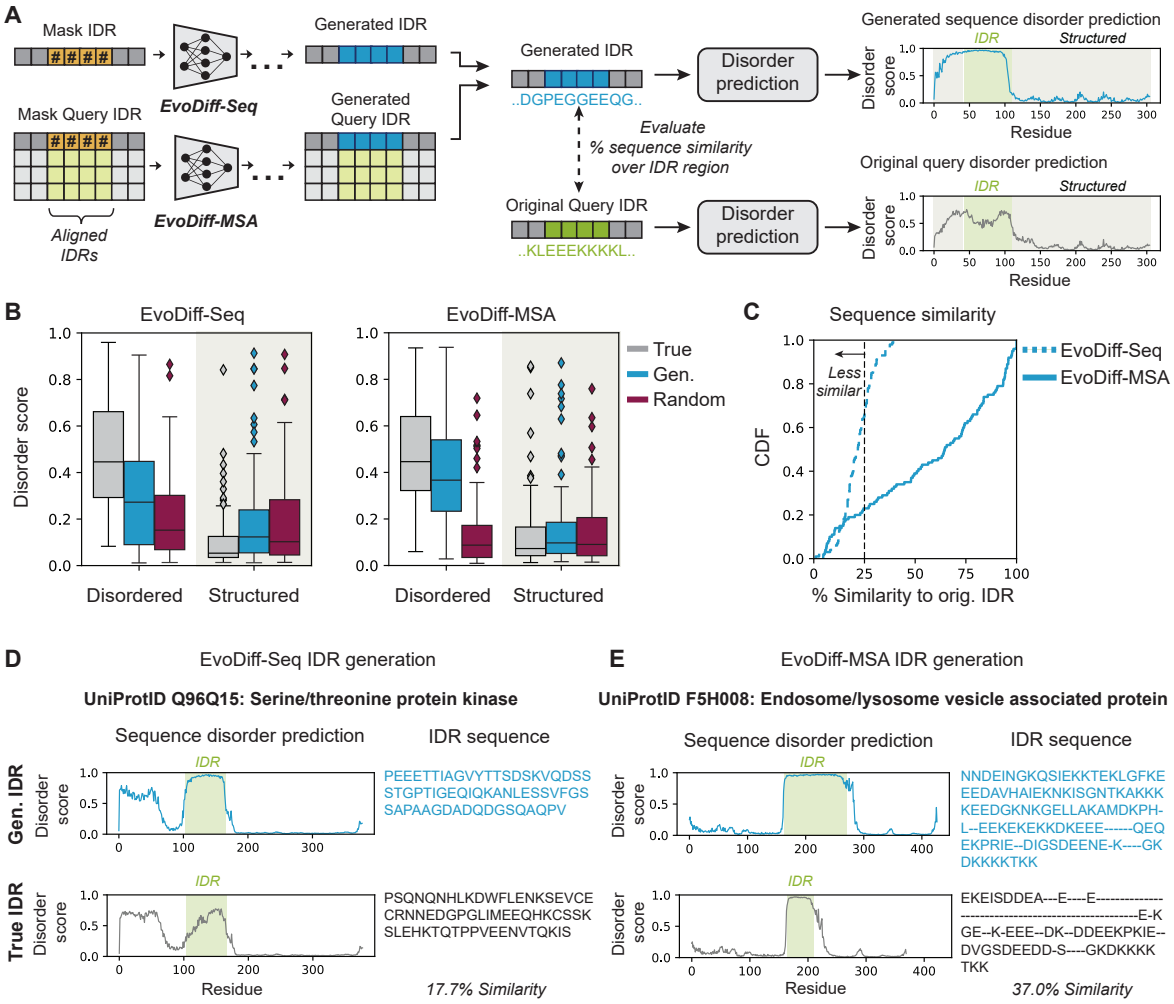

Figure S13: **EvoDiff generates intrinsically disordered regions. (A)** A new IDR sequence is generated from EvoDiff-Seq or EvoDiff-MSA by inpainting disordered residues in the query sequence. DR-BERT is then used to predict disorder scores for the original and regenerated sequences. **(B)** Distributions of disorder scores over disordered and structured regions for sequences with true (grey), inpainted (blue), and randomly-sampled (red) IDRs (n=100 sequences per condition; box plots show median and interquartile range). **(C)** Distribution of sequence similarity relative to the original IDR for generated IDRs from EvoDiff-Seq (blue, dashed) and EvoDiff-MSA (blue, solid) (n=100; dashed line at 25%). **(D-E)** Predicted disorder scores and corresponding sequences for representative generated (top row) and true (bottom row) IDRs from EvoDiff-Seq (D) and EvoDiff-MSA (E).

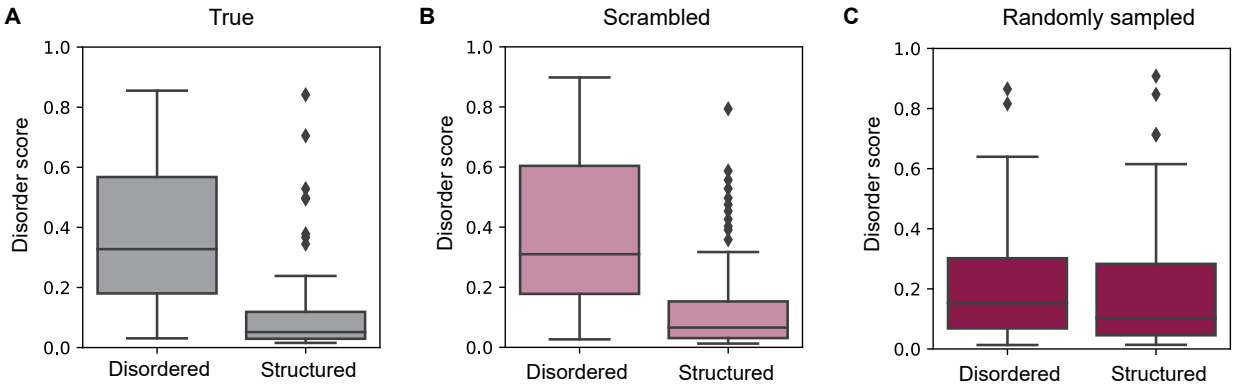

Figure S14: **Baseline performance of DR-BERT evaluator. (A-C)** Distributions of DR-BERT predicted disorder scores across disordered and structured regions for sequences with true (A), scrambled (B), and randomly sampled (C) IDRs (n=100).

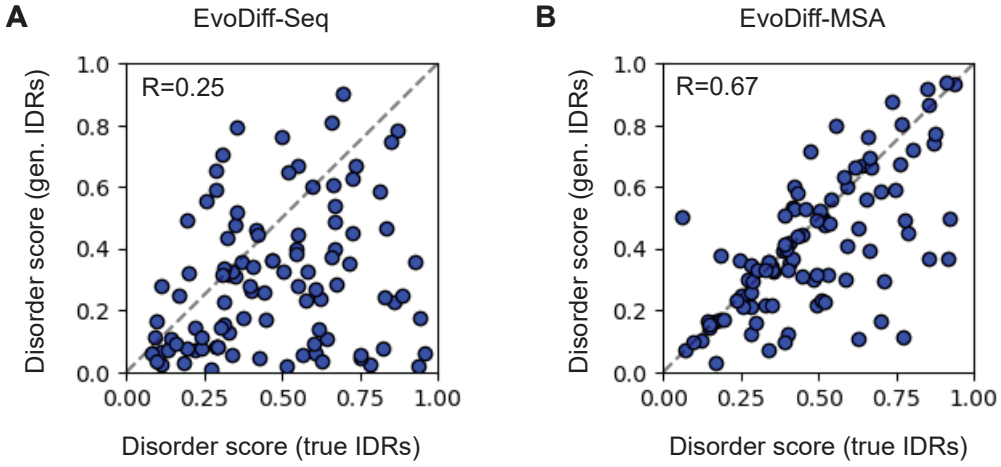

Figure S15: **Performance of DR-BERT evaluator on disorder regions. (A-B)** Disorder scores predicted by DR-BERT for true (x-axis) vs. generated (y-axis) IDRs for the same given sequence, for generations from EvoDiff-Seq (A) and EvoDiff-MSA (B). Each dot represents an individual IDR (n=100). The Pearson R is given for each of EvoDiff-Seq and EvoDiff-MSA.

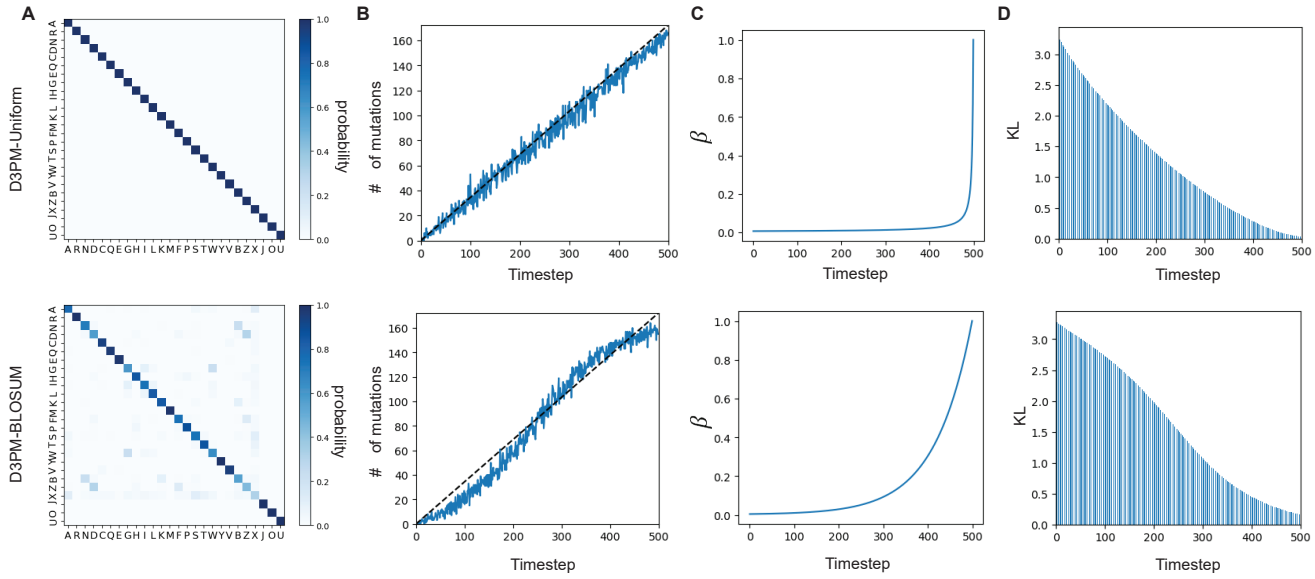

Figure S16: **Details of EvoDiff-D3PM corruption schemes.** The top and bottom rows correspond to EvoDiff-D3PM-Uniform and EvoDiff-D3PM-BLOSUM, respectively. **(A)** Visualization of EvoDiff-D3PM transition matrices. **(B)** Evolution of the number of mutations accrued as a function of the diffusion timestep t for a sample input. **(C)** Evolution of β as a function of the diffusion timestep t. **(D)** Evolution of $D_{KL}\left[q(x_t|x_0)\|p(x_T)\right]$ as a function of the diffusion timestep t, indicating convergence to a uniform stationary distribution at $t = 500$ as D_{KL} approaches zero.

Table S1: **Performance of EvoDiff sequence models**. The reconstruction KL (Recon KL) was calculated between the distribution of amino acids in the test set and in generated samples (n=1000). The perplexity was computed on 25k samples from the test set. The minimum Hamming distance to any train sequence of the same length (Hamming) is reported for each model as the mean $\pm$ standard deviation over the generated samples. Fréchet ProtT5 distance (FPD) was calculated between the test set and generated samples. The secondary structure KL (SS KL) was calculated between the means of the predicted secondary structures of the test and generated samples.

Model	parameters	Recon KL	perplexity	Hamming	FPD	SS KL
Test	-	9.92e-4[1]	-	0.0039[2]	0.10[1]	1.37e-5[1]
D3PM BLOSUM	38M	1.77e-2	17.16	0.83 ± 0.05	1.42	3.30e-5
D3PM Uniform	38M	1.48e-3	18.82	0.83 ± 0.05	1.31	3.73e-5
OADM	38M	1.11e-3	14.61	0.83 ± 0.07	0.92	1.61e-4
D3PM BLOSUM	640M	3.73e-2	15.74	0.83 ± 0.05	1.53	4.96e-4
D3PM Uniform	640M	2.90e-3	18.47	0.83 ± 0.05	1.35	2.13e-4
OADM	640M	1.26e-3	13.05	0.83 ± 0.08	0.88	1.48e-4
LRAR	38M	7.90e-4	12.38	0.82 ± 0.06	0.86	1.61e-4
CARP	38M	5.71e-1	25.13	0.74 ± 0.07	6.30	2.72e-3
LRAR	640M	7.01e-4	10.41	0.83 ± 0.06	0.63	1.76e-5
CARP	640M	3.56e-1	31.77	0.84 ± 0.05	1.78	5.03e-3
ESM-1b [3]	650M	4.91e-1	53.49	0.83 ± 0.06	6.67	5.48e-4
ESM-2 [3]	650M	5.00e-1	68.39	0.84 ± 0.06	6.79	3.05e-3
FoldingDiff [4]	14M	5.49e-2	-	-	1.64	1.76e-3
RFdiffusion [5]	60M	7.19e-2	-	-	1.96	5.98e-3
Random	-	1.65e-1	20	0.85 ± 0.04	3.16	1.90e-4

[1] Calculated between the test set and validation set.
[2] Reported value is the minimum Hamming distance between any two natural sequences of the same length in UniRef50.
[3] Due to model constraints, the maximum sequence length sampled was 1022.
[4] For the FoldingDiff baseline, 1000 structures generated by FoldingDiff were randomly selected, and the corresponding 1000 inferred sequences were inverse-folded using ESM-IF. These sequences are between lengths of 50 and 128 residues.
[5] For the RFdiffusion baseline, 1000 structures were generated corresponding to the UniRef train distribution length, and 1000 corresponding sequences were inverse-folded using ESM-IF.

Table S2: **Structural plausibility metrics for EvoDiff sequence models and baselines.** Metrics are reported as the mean ± standard deviation for 1000 generated samples for each model.

Model	Params	ESM-IF scPerplexity	ProteinMPNN scPerplexity	OmegaFold pLDDT
Test	-	8.04 ± 4.04	3.09 ± 0.63	68.25 ± 17.85
D3PM Blosum	38M	12.38 ± 2.06	3.80 ± 0.49	42.76 ± 14.55
D3PM Uniform	38M	12.03 ± 2.04	3.77 ± 0.50	42.37 ± 14.39
OADM	38M	11.61 ± 2.38	3.72 ± 0.50	43.78 ± 14.18
D3PM Blosum	640M	11.86 ± 2.21	3.73 ± 0.48	44.14 ± 13.80
D3PM Uniform	640M	12.29 ± 2.05	3.78 ± 0.49	41.65 ± 14.32
OADM	640M	11.53 ± 2.50	3.71 ± 0.52	44.46 ± 14.62
LRAR	38M	11.61 ± 2.38	3.64 ± 0.56	48.26 ± 14.87
CARP	38M	9.68 ± 2.56	3.66 ± 0.62	50.79 ± 12.06
LRAR	640M	10.99 ± 2.63	3.59 ± 0.54	48.71 ± 15.47
CARP	640M	14.13 ± 2.42	4.05 ± 0.52	41.56 ± 14.35
ESM-1b	650M	13.90 ± 2.44	3.47 ± 0.68	58.07 ± 15.64
ESM-2	650M	14.02 ± 2.87	3.58 ± 0.69	50.70 ± 15.67
Random	-	14.68 ± 1.97	3.96 ± 0.50	39.97 ± 14.05

Table S3: **Validation-set perplexities for EvoDiff MSA models.** The perplexity is calculated based on the ability of each model to reconstruct a subsampled MSA from the validation set. "Max Perplexity" and "Rand. Perplexity" indicate MaxHamming and Random subsampling, respectively, for construction of the validation MSA.

Corruption	Subsampling	Params	Max Perplexity	Rand. Perplexity
D3PM BLOSUM	Random	100M	11.35	8.31
D3PM BLOSUM	Max	100M	10.98	7.61
D3PM Uniform	Random	100M	10.14	6.77
D3PM Uniform	Max	100M	10.06	6.66
OADM	Random	100M	6.05	3.64
OADM	Max	100M	6.14	3.60
ESM-MSA-1b	Max	100M	11.20	5.89

Table S4: **Performance of EvoDiff MSA models in generating query sequences conditioned on MSAs.** Metrics are reported as the mean ± standard deviation over 250 generated samples for each model.

Model	scPerplexity	pLDDT	Seq. similarity	TM score
Valid	5.93 ± 3.19	73.99 ± 17.80	14.58 ± 21.64 [1]	-
OADM (Rand) - Rand MSA	9.41 ± 2.61	55.99 ± 14.75	6.13 ± 9.88	0.49 ± 0.23
OADM (Max) - Max MSA	9.38 ± 2.57	57.08 ± 16.01	6.74 ± 11.00	0.50 ± 0.23
OADM (Max) - Rand MSA	9.59 ± 2.69	54.95 ± 16.83	6.55 ± 10.49	0.46 ± 0.23
ESM-MSA-1b	10.05 ± 2.92	51.64 ± 16.54	7.13 ± 11.60	0.40 ± 0.23
Potts	10.34 ± 2.26	55.46 ± 13.82	12.01 ± 17.19	0.17 ± 0.10

[1] Sequence similarity is calculated between the original query sequence and all the sequences in the MSA.

Table S5: **Scaffolding performance of EvoDiff-Seq.** Number of scaffolding successes out of 100 generations for RFdiffusion, EvoDiff-Seq, the LRAR baseline, the CARP baseline, and randomly sampled scaffolds (Random), for each of 17 scaffolding problems. The bottom row contains the total number of successful scaffolds generated per model.

PDB	RFdiffusion	EvoDiff-Seq	LRAR	CARP	Random
1BCF	100	24	0	4	0
6E6R	71	16	7	3	1
2KL8	88	0	1	1	0
6EXZ	42	0	0	0	0
1YCR	74	13	12	10	7
6VW1	69	1	0	0	0
4JHW	0	0	0	0	0
5TPN	61	0	0	0	0
4ZYP	40	0	0	0	0
3IXT	25	23	22	13	7
7MRX	7	0	0	0	0
1PRW	8	68	70	54	5
5IUS	2	0	0	0	0
5YUI	0	4	0	0	0
5WN9	0	0	0	0	2
1QJG	0	0	0	0	0
5TRV	22	0	0	0	0
Total	610	149	112	85	22

Table S6: **Scaffolding performance of EvoDiff-MSA.** Number of scaffolding successes out of 100 generations for RFdiffusion, EvoDiff-MSA (Max), EvoDiff-MSA (Random), and the ESM-MSA baseline, for each of 17 scaffolding problems. The bottom row contains the total number of successful scaffolds generated per model.

PDB	RFdiffusion	EvoDiff-MSA (Max)	EvoDiff-MSA (Random)	ESM-MSA
1BCF	100	100	98	99
6E6R	71	87	63	96
2KL8	88	11	31	42
6EXZ	42	86	87	73
1YCR	74	3	0	0
6VW1	69	4	3	4
4JHW	0	0	0	0
5TPN	61	0	0	0
4ZYP	40	0	0	0
3IXT	25	1	0	5
7MRX	7	72	68	66
1PRW	8	48	46	92
5IUS	2	3	1	7
5YUI	0	58	44	70
5WN9	0	0	0	0
1QJG	0	34	22	38
5TRV	22	15	12	12
Total	610	522	475	604