No Pseudolikelihood for Me: Training Potts Models with Contrastive Divergence for Protein Design

Tianyu Lu[*]
Department of Computer Science
Department of Cell and Systems Biology
University of Toronto

Syed Hussain Ather[*]
Institute of Medical Science
University of Toronto

April 24, 2021

Abstract

We train a Potts model for protein sequence families with contrastive divergence and compare its ability for out-of-distribution detection and ability to model a distribution over plastic-degrading protein sequences with alternative training methods such as Boltzmann ML and pseudolikelihood. Contrastive divergence has the best out-of-distribution detection performance while Boltzmann ML achieves the best performance in capturing second order correlations and protein structure prediction.

***K*eywords** Potts Model · Protein Design · Energy-based Model

Introduction

The current 250 million known protein sequences are the result of the generative process evolution [26]. Similar sequences that have likely evolved from the same ancestral sequence, also known as homologs, are clustered in families stored in the Pfam database [19]. Each family is represented by a multiple sequence alignment, a $B \times L$ matrix where each entry can be one of 20 amino acids or the gap character [15]. B is the number of sequences in a family and L is the length of each sequence in the alignment. Potts models trained on homologous sequences can predict the common protein 3D structure to these sequences [16], predict the beneficial or detrimental effect of mutations [13], and be sampled to generate novel sequences for protein design [22].

A protein is a string of amino acids, also called residues, that can adopt a 3D fold. These folds mean that two residues far apart in the linear sequence can be nearby in 3D space. An underlying assumption of protein science is that proteins structures prefer to adopt low energy states [3]. Interactions with neighbouring atoms should be locally favourable. Thus, for mutations to not disrupt protein folds, residues nearby in 3D space tend to co-evolve, i.e. a residue's evolutionary trajectory is dependent on nearby residues [16, 9]. Potts models are attractive for protein modelling because of their inductive bias which explicitly captures the tendencies for a residue-residue pair to co-evolve [16, 12].

Methods

Potts Model

Before discussing the Potts model, we describe the notation used in [12].

- $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \cdots, \sigma_N)$ is a protein sequence of length N
- $\sigma_i \in \{1, 2, \cdots, q\}, \quad q = 21$ specifies the amino acid (or gap) at position i
- B: the number of sequences in the multiple sequence alignment (MSA)

[*] Equal contribution.

- $f_i(k) = \frac{1}{B}\sum_{b=1}^{B}\mathbb{I}[\sigma_i^{(b)} = k]$ is the empirical frequency (in the MSA) of observing amino acid k at position i and $\mathbb{I}$ is the indicator function.
- $f_{ij}(k,l) = \frac{1}{B}\sum_{b=1}^{B}\mathbb{I}[\sigma_i^{(b)} = k] \cdot \mathbb{I}[\sigma_j(b) = l]$ is the empirical frequency (in the MSA) of obsserving amino acid k at position i and amino acid l at position j.
- $c_{ij}(k,l) = f_{ij}(k,l) - f_i(k)f_j(l)$ captures how frequent k, l co-occur at positions i, j that expected if they were to be independent.

A Potts model is an energy-based model where the likelihood of a sequence $\boldsymbol{\sigma}$ is proportional to its energy.

$$P(\boldsymbol{\sigma}) = \frac{1}{Z}\exp\left(\underbrace{\sum_{i=1}^{N} h_i(\sigma_i)}_{\text{fields}} + \underbrace{\sum_{1\le i<j\le N} J_{ij}(\sigma_i,\sigma_j)}_{\text{couplings}}\right) \tag{1}$$

where $h_i(\cdot)$ and $J_{ij}(\cdot,\cdot)$ are the $Nq + \frac{N(N-1)}{2}q^2$ parameters of the Hamiltonian to be learned by minimizing $-\log P(\boldsymbol{\sigma})$. The fields term model the sitewise frequencies of amino acids and the couplings term model the pairwise frequencies.

The form of a Potts model is derived from imposing two conditions on the desired probability distribution: (1) reproduce sitewise and pairwise frequencies observed in the data and (2) maximize entropy [16]. Empirically, the a coupling term J_{ij} is directly related to spatial proximity of residues i and j, and reduces the number of false contacts due to chaining effects (i.e. if A and B are correlated, B and C are correlated, then A and C are correlated despite not being in contact) [16].

However, good performance in contact prediction does not necessarily imply good performance in function prediction, nor the biological validity of generated samples from a trained model [17]. In particular for protein design, robust out-of-distribution detection is critical for success, where in-distribution is defined as those sequences that can fold and function at least at the wildtype level. Without good out-of-distribution detection, generated sequences may fail despite having high likelihood under a generative model or high predicted function under a discriminative model, leading to wasteful experiments [8]. Thus, inspired by [14], we train Potts models with a loss that explicitly penalizes low likelihoods for in-distribution samples and high likelihoods for out-of-distribution samples. Our motivation for choosing to test Boltzmann Machine Learning and Pseudolikelihood training methods arise from the fact that they have been used to infer generative enzyme models and protein fold/function prediction models respectively [22, 16, 13].

Boltzmann Machine Learning

The update equations for the Potts model parameters using Boltzmann Machine Learning are:

$$h_i(k) \leftarrow h_i(k) + \epsilon(f_i(k) - \langle\delta_{\sigma_i,k}\rangle_P) \tag{2}$$

$$J_{ij}(k,l) \leftarrow J_{ij}(k,l) + \epsilon(f_{ij}(k,l) - \langle\delta_{\sigma_i,k}\cdot\delta_{\sigma_j,l}\rangle_P) \tag{3}$$

where $\langle\delta_{\sigma_i,k}\rangle_P$ is the estimated frequency of amino acid k at position i and $\langle\delta_{\sigma_i,k}\cdot\delta_{\sigma_j,l}\rangle_P$ is the estimated co-occurring frequencies of amino acid k at position i and amino acid l at position j, both estimated by MCMC [1]. As an efficient implementation is crucial, we use the C implementation developed by the Ranganathan lab.

Pseudolikelihood

A fast but not exact training method to overcome the intractable normalizer Z is to train via the pseudolikelihood [12].

$$P(\sigma_r^{(b)}|\boldsymbol{\sigma}_{\backslash r}^{(b)}) = \frac{\exp\left(h_r(\sigma_r) + \sum\limits_{\substack{i=1\\i\neq r}}^{N} J_{ri}(\sigma_r^{(b)},\sigma_i^{(b)})\right)}{\sum\limits_{l=1}^{q}\exp\left(h_r(l) + \sum\limits_{\substack{i=1\\i\neq r}}^{N} J_{ri}(l,\sigma_i^{(b)}))\right)} \tag{4}$$

such that the final objective to minimize over a batch of B sequences is

$$\mathcal{L}(\mathbf{h}, \mathbf{J}) = -\frac{1}{B} \sum_{r=1}^{N} \sum_{b=1}^{B} \log P(\sigma_r^{(b)} | \boldsymbol{\sigma}_{\setminus r}^{(b)}) \tag{5}$$

In NLP terms, this is the masked language modelling loss [10]. We use the GREMLIN implementation for this training strategy [5].

Contrastive divergence

The contrastive divergence loss to minimize is

$$\mathcal{L}(\mathbf{h}, \mathbf{J}) = \mathbb{E}_{\mathbf{x}_{\text{in}} \sim \mathcal{D}_{\text{in}}}[\max(0, \mathcal{H}_{(\mathbf{h},\mathbf{J})}(\mathbf{x}_{\text{in}}) - m_{\text{in}})]^2 + \mathbb{E}_{\mathbf{x}_{\text{out}} \sim \mathcal{D}_{\text{out}}}[\max(0, m_{\text{out}} - \mathcal{H}_{(\mathbf{h},\mathbf{J})}(\mathbf{x}_{\text{out}}))]^2 \tag{6}$$

where $\mathcal{H}_{(\mathbf{h},\mathbf{J})}(\mathbf{x})$ is the Hamiltonian (energy) of sequence $\mathbf{x}$ under our model, $\mathcal{D}_{\text{in}}$ is our training data, $\mathcal{D}_{\text{out}}$ is our out-of-distribution examples, and m_{in} and m_{in} are hyperparameters. This loss has been successfully used for energy-based out-of-distribution detection [14]. This training strategy can be thought of as maximizing the gap between likelihoods of in vs. out of distribution samples. Taking the log ratio between two likelihoods effectively cancels out the intractable normalizer.

Training Data

We will obtain homologous sequences of `6EQD`, a plastic-degrading enzyme called PETase, as training data [4]. One place to look is the dienelactone hydrolase Pfam family with 17213 sequences. These sequences form a "cluster" of likely homologous sequences using the HMM-based method described in [19]. Based on this sequence similarity, they are also predicted to have similar structures and thus similar functions. Training a generative model $P_{\boldsymbol{\theta}}(\mathbf{x})$ on this distribution of sequences models this particular protein family, which contains members with the ability to degrade plastics. Thus, the generated samples from this model distribution will also likely have plastic-degrading properties.

Reweighing

A generative model assumes the samples in the training data are *iid*. However, with protein evolution, this is not the case, see [7, 6]. Proteins not only evolve based on their ancestral sequences, existing sequences recombine, permute, and flip according to DNA mutation mechanisms. Moreover, we only observe sequences at the current point in evolution, not the evolutionary trajectory itself. Thus, sequences with high similarity to each other are less informative about the evolutionary process. The typical heuristic is to reweigh each sequence according to its sequence similarity to all other sequences in the training data using

$$w_b = \frac{1}{m_b} \tag{7}$$

$$m_b = |\{a \in \{1, \cdots, B\} : \text{sim}(\boldsymbol{\sigma}^{(a)}, \boldsymbol{\sigma}^{(b)}) \geq \gamma\}| \tag{8}$$

$$\gamma = 0.8 \tag{9}$$

where B is the total number of sequences, sim$(\boldsymbol{\sigma}^{(a)}, \boldsymbol{\sigma}^{(b)})$ is the percent sequence identity between two sequences $\boldsymbol{\sigma}^{(a)}$ and $\boldsymbol{\sigma}^{(b)}$, $b \in \{1, \cdots, B\}$ is the index of some sequence, and γ is a hyperparameter. Intuitively, this counts the number of sequences with over 80% sequence identity. The more similar sequences it has in the training data, the less the weight.

Evaluation Metrics

False positives

To evaluate model performance on out-of-distribution detection, we compute the following ratios

- In/In: $\log \frac{P(S_i)}{P(S_j)}$
- Out/Out: $\log \frac{P(T_i^{(1)})}{P(T_j^{(1)})}$
- In/Out1: $\log \frac{P(S_i)}{P(T_i^{(1)})}$
- In/Out2: $\log \frac{P(S_i)}{P(T_i^{(2)})}$

where S_i is the i^{th} sequence of the training data (in-distribution), $T_i^{(1)}$ is the i^{th} sequence of the training data but shuffled, and $T_i^{(2)}$ is the i^{th} sequence of the training data with two halves swapped, i.e. if $S_i = \texttt{MNFPRA}$, then $T_i^{(2)} = \texttt{PRAMNF}$. For In/Out1, the shuffled sequences preserve site-wise statistics, i.e. frequencies of residues at each position, but destroys pair-wise statistics. Experimental results show that destroying pair-wise statistics also destroy the protein fold and function [24, 23], thus the motivating the need to distinguish such out-of-distribution sequences. For In/Out2, swapping the two halves preserves local pair-wise statistics but destroys long-range pair-wise statistics. As proteins fold into 3-dimensional structures where residues near the end of the protein can interact with residues near the start, swapping the two ends generates an out-of-distribution sequence. Moreover, such sequences will fail to be classified as out-of-distribution using profile models such as HMMs which only capture local sequence patterns. Some model evaluation metrics such as using k-mer statistics will also fail to detect such sequences as out-of-distribution [18].

A model is said to be an out-of-distribution detector if it can distinguish between the In/In versus the In/Out probabilities. We measure the quality of this distinction as the area under the intersection of In/In and In/Out2. A Potts model with randomly initialized weights cannot make such a distinction **(Figure S1)**.

First order correlation

First order statistics are the empirical frequencies of each amino acid at each residue position. Thus, for a sequence of length L, there are $L \times 21$ frequencies observed in the training data. We compute these frequencies for the training sequences and 10000 sampled sequences drawn from a trained Potts model using MCMC. The reported correlation is the Spearman's rho between the empirical frequencies in the training sequences vs. the sampled sequences.

Second order correlation

Second order statistics are the empirical frequencies of each pair of amino acids at each pair of residue positions. There are $L \times L \times 21 \times 21$ such frequencies for a sequence of length L. The reported correlation is the Spearman's rho between frequencies in the training vs. sampled sequences.

Contact prediction

The couplings matrix $\mathbf{J}$ of a Potts model has been shown to capture information about protein 3D structure [16]. We use Direct Coupling Analysis (DCA) to predict protein contacts from $\mathbf{J}$ and report the precision, i.e. the fraction of true contacts out of N predicted contacts, where $N = L/2$ and $L/5$, and L is the sequence length.

Sequence identity

We report the maximum sequence identity, i.e. the fraction of identical residues, of each sampled sequence to the sequences in the training set. The higher the sequence identity, the more likely the model has memorized the training data and has not generalized to homologous sequences.

Rosetta energies

The Rosetta Energy Function (REF) is a linear combination of 19 physics-based and heuristic energy terms [2]. The REF guides all Rosetta protocols and have been successfully applied towards *de novo* design of synthetic proteins of many diverse functions [27, 11]. We evaluate samples from the Potts model by mapping the mutations onto a wildtype structure and computing the REF. We compute the REF for 20 sequences, ten with highest energies and ten with lowest energies according to the Potts model. Ideally, the energies of the Potts model should be consistent with the REF.

Results

False positives, first-order and second-order correlations, precision at L/2 and L/5 for contact prediction, and training times are shown for each method.

We see that training a Potts model with contrastive divergence gives an excellent out-of-distribution detector, with GREMLIN following close behind. In practice, one can fix a known sequence $\mathbf{x}$ to be in-distribution, such as the

Method	False positives	First order	Second order	L/2	L/5	Training time
Pseudolikelihood	0.008	0.80	**0.47**	0.15	0.06	20sec
Boltzmann ML	0.46	**0.96**	**0.49**	**0.34**	**0.19**	12hr
Contrastive Divergence	**1e-06**	**0.95**	0.22	0.05	0.01	25min
Random	0.93	-0.01	0.00	0.11	0.05	0s

Table 1: Summary of results. False positives and training time: lower the better. All other metrics: higher the better.

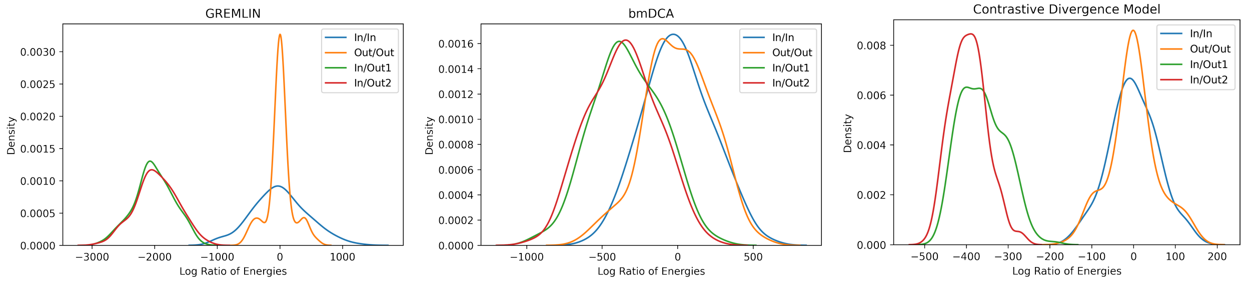

Figure 2: Plots of the In/In, Out/Out, In/Out1, and In/Out2 distributions for all three training methods.

wildtype sequence. Then for any designed sequence $\hat{\mathbf{x}}$, we can compute $\log \frac{P(\curvearrowleft)}{\hat{\mathbf{x}}}$. If the designed sequence $\hat{\mathbf{x}}$ is in-distribution, then it should have a log ratio within the range of In/In. If the designed sequence is out-of-distribution, then its log ratio should fall within the range of In/Out1 or In/Out2. Thus, if our out-of-distribution detector returns FALSE for any log ratio within the range of In/In, the expected fraction of sequences that are not in-distribution is equal to the false positives score.

bmDCA is able to faithfully recover the first and second order correlations. Despite the high Spearman's rho for first order correlations, the samples from contrastive divergence do not match the scale of first order frequencies observed in the data. This is confirmed by visualizing the sequence distributions estimated by 10000 MCMC samples of each method in **Figure S3 S4 S5**, where only bmDCA produces a logo which resembles the sequence distribution of the Pfam HMM.

While GREMLIN may be used as an out-of-distribution detector, its samples have high sequence identity to the training data. The typical sequence identity for homologous sequences is in the 20% to 40% range [12]. When generating synthetic proteins where sequence identity is a *desideratum*, bmDCA and contrastive divergence appear more suitable.

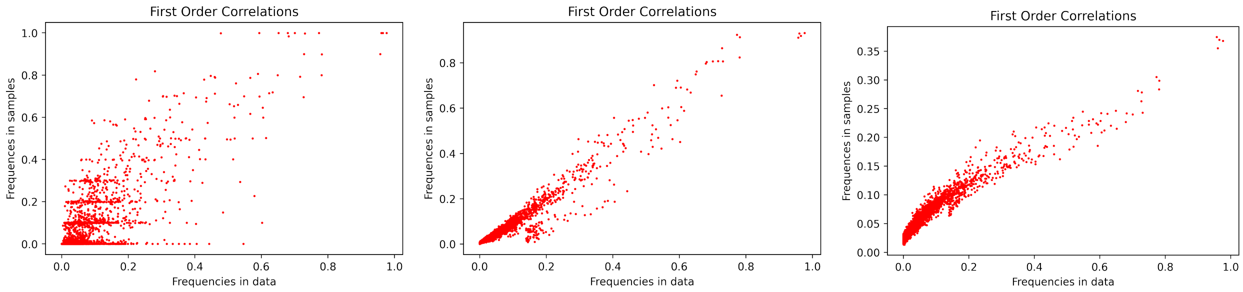

Figure 3: First order correlations of 10000 MCMC samples for GREMLIN (left), bmDCA (center), and contrastive divergence (right).

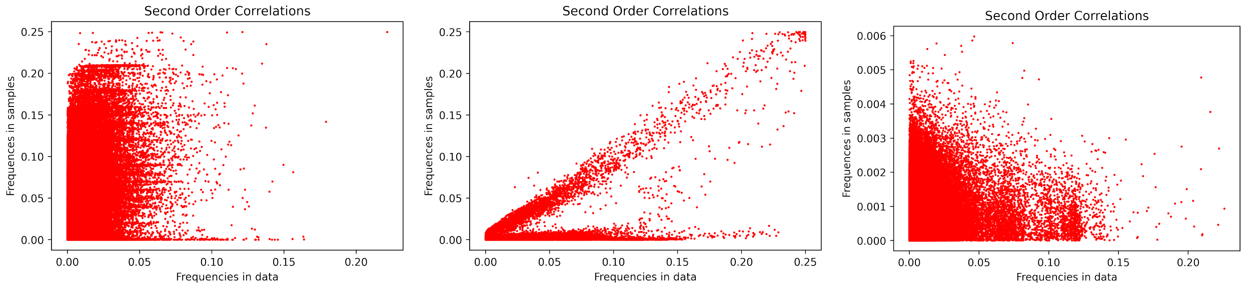

Figure 4: Second order correlations of 10000 MCMC samples for GREMLIN (left), bmDCA (center), and contrastive divergence (right).

Rosetta Energies

We take out of each method's 10000 MCMC samples ten sequences with lowest energies and ten sequences with the highest energies. We compute the Rosetta Energy Function of the designed sequences after mapping it to the `6EQD` protein structure. We observe in **Figure 6** that for bmDCA and contrastive divergence, sequences with low energy under the Potts model also tend to have lower Rosetta energies. This is intuitively satisfying because this means the energies under a Potts model is correlated with an energy function based on protein chemistry and electrostatics. It suggests that the Potts model, using information from sequence alone, is able to capture biophysical aspects of protein structures beyond co-evolution and contact prediction.

Discussion

Here we showed that a simple contrastive divergence training strategy is able to serve as a robust out-of-distribution detector for protein design. We also showed that the Potts energies of sequences, when trained under bmDCA or contrastive divergence, correlates with a biophysical energy function of protein structure. Despite its long training time, of all training methods assessed, bmDCA can most faithfully capture first and second order correlations in protein sequence and generate sequences that are likely homologous.

To truly assess the performance of a generative model of protein sequences, we must perform wetlab experiments to synthesize the sampled sequences and test their activity. Such experiments can reveal whether the samples from contrastive divergence which have low sequence identity, falling into the twilight zone of protein homology, can indeed generate functioning synthetic sequences. bmDCA has already been shown to generate functioning synthetic sequences [22], where the energy of a sequence is inversely correlated with the likelihood of it being functional. Generative models can also be used as filters. Given a black-box design algorithm which proposes protein sequences, a generative model can filter sequences that have low likelihood under the model and thus only keep those that are likely to be functional. Ablation experiments that include or exclude a Potts model as a filter would be of interest.

If the goal is to generate functioning protein sequences, and not to predict protein structure, we can augment a generative model by incorporating information from structure, such as the likelihood of a sequence to fold into a given backbone [25]. Comparisons of experimental results of sequences generated by VAEs or GANs may be of interest [21, 20].

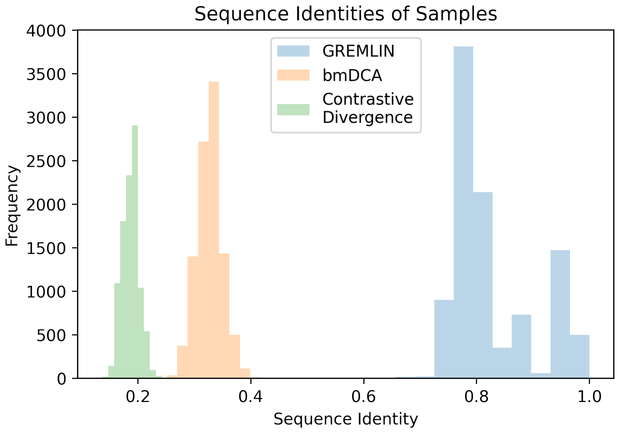

Figure 5: Frequency of sequence identities for each modelling method.

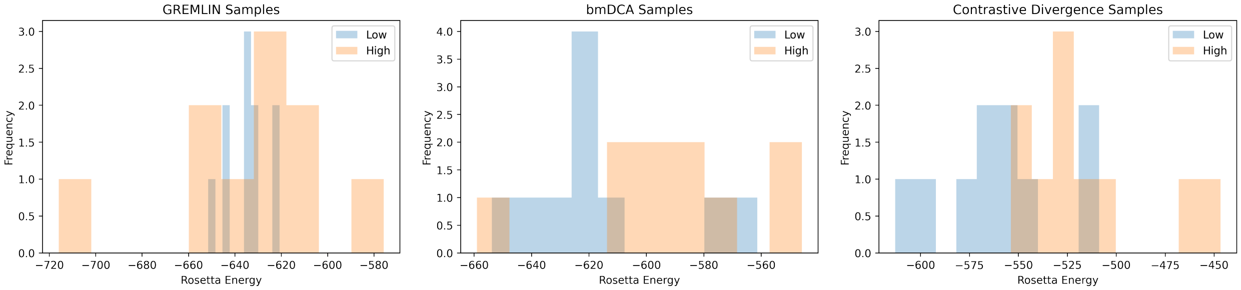

Figure 6: Rosetta energies computed from mutating the wildtype structure `6EQD` to the amino acids specified in the sampled sequences.

Code Availability

The code to reproduce all results is available at https://bit.ly/Potts-Proteins.

References

[1] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. “A learning algorithm for Boltzmann machines”. In: *Cognitive science* 9.1 (1985), pp. 147–169.

[2] Rebecca F Alford et al. “The Rosetta all-atom energy function for macromolecular modeling and design”. In: *Journal of chemical theory and computation* 13.6 (2017), pp. 3031–3048.

[3] Christian B Anfinsen. “Principles that govern the folding of protein chains”. In: *Science* 181.4096 (1973), pp. 223–230.

[4] Harry P Austin et al. “Characterization and engineering of a plastic-degrading aromatic polyesterase”. In: *Proceedings of the National Academy of Sciences* 115.19 (2018), E4350–E4357.

[5] Sivaraman Balakrishnan et al. “Learning generative models for protein fold families”. In: *Proteins: Structure, Function, and Bioinformatics* 79.4 (2011), pp. 1061–1078.

[6] Matthew Bashton and Cyrus Chothia. “The generation of new protein functions by the combination of domains”. en. In: *Structure* 15.1 (Jan. 2007), pp. 85–99.

[7] Jamie T Bridgham, Eric A Ortlund, and Joseph W Thornton. “An epistatic ratchet constrains the direction of glucocorticoid receptor evolution”. In: *Nature* 461.7263 (2009), pp. 515–519.

[8] David Brookes, Hahnbeom Park, and Jennifer Listgarten. “Conditioning by adaptive sampling for robust design”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 773–782.

[9] Simona Cocco et al. “Inverse statistical physics of protein sequences: a key issues review”. In: *Reports on Progress in Physics* 81.3 (2018), p. 032601.

[10] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Jiayi Dou* et al. “De novo design of a fluorescence-activating beta-barrel”. In: *Nature* (Sept. 12, 2018). ISSN: 1476-4687. DOI: `10.1038/s41586-018-0509-0`. URL: `https://www.nature.com/articles/s41586-018-0509-0%20https://www.bakerlab.org/wp-content/uploads/2018/09/s41586-018-0509-0.pdf`.

[12] Magnus Ekeberg et al. “Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models”. In: *Physical Review E* 87.1 (2013), p. 012707.

[13] Thomas A Hopf et al. “Mutation effects predicted from sequence co-variation”. In: *Nature biotechnology* 35.2 (2017), pp. 128–135.

[14] Weitang Liu et al. “Energy-based Out-of-distribution Detection”. In: *arXiv preprint arXiv:2010.03759* (2020).

[15] Fábio Madeira et al. “The EMBL-EBI search and sequence analysis tools APIs in 2019”. In: *Nucleic acids research* 47.W1 (2019), W636–W641.

[16] Debora S Marks et al. “Protein 3D structure computed from evolutionary sequence variation”. In: *PloS one* 6.12 (2011), e28766.

[17] Dylan Marshall et al. “The structure-fitness landscape of pairwise relations in generative sequence models”. In: *bioRxiv* (2020).

[18] Francisco McGee et al. “Generative Capacity of Probabilistic Protein Sequence Models”. In: *arXiv preprint arXiv:2012.02296* (2020).

[19] Jaina Mistry et al. “Pfam: The protein families database in 2021”. In: *Nucleic Acids Research* 49.D1 (2021), pp. D412–D419.

[20] Donatas Repecka et al. “Expanding functional protein sequence spaces using generative adversarial networks”. In: *Nature Machine Intelligence* (2021), pp. 1–10.

[21] Adam J Riesselman, John B Ingraham, and Debora S Marks. “Deep generative models of genetic variation capture the effects of mutations”. In: *Nature methods* 15.10 (2018), pp. 816–822.

[22] William P Russ et al. “An evolution-based model for designing chorismate mutase enzymes”. In: *Science* 369.6502 (2020), pp. 440–445.

[23] William P Russ et al. “Natural-like function in artificial WW domains”. In: *Nature* 437.7058 (2005), pp. 579–583.

[24] Michael Socolich et al. “Evolutionary information for specifying a protein fold”. In: *Nature* 437.7058 (2005), pp. 512–518.

[25] Alexey Strokach et al. “Fast and flexible protein design using deep graph neural networks”. In: *Cell Systems* 11.4 (2020), pp. 402–411.

[26] “UniProt: The universal protein knowledgebase in 2021”. In: *Nucleic Acids Research* 49.D1 (2021), pp. D480–D489.

[27] Chunfu Xu et al. “Computational design of transmembrane pores”. In: *Nature* 585 (Aug. 26, 2020), pp. 129–134. DOI: `10.1038/s41586-020-2646-5`.

Supplementary

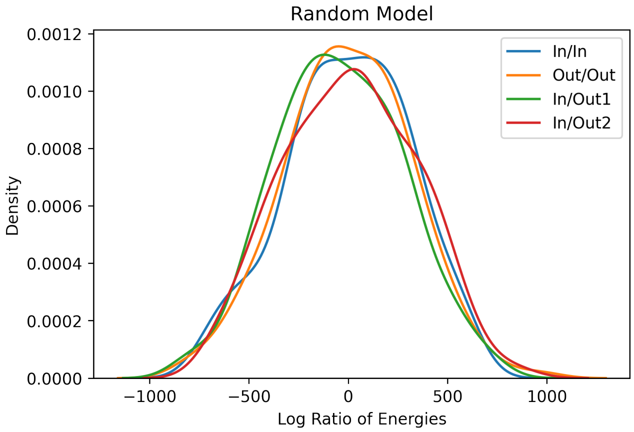

Figure S1: Log ratios of Potts model likelihoods for a Potts model with randomly initialized weights.

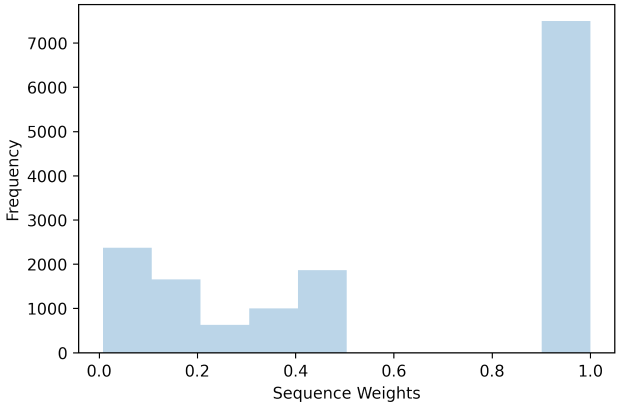

Figure S2: Sequence weights used when training each model.

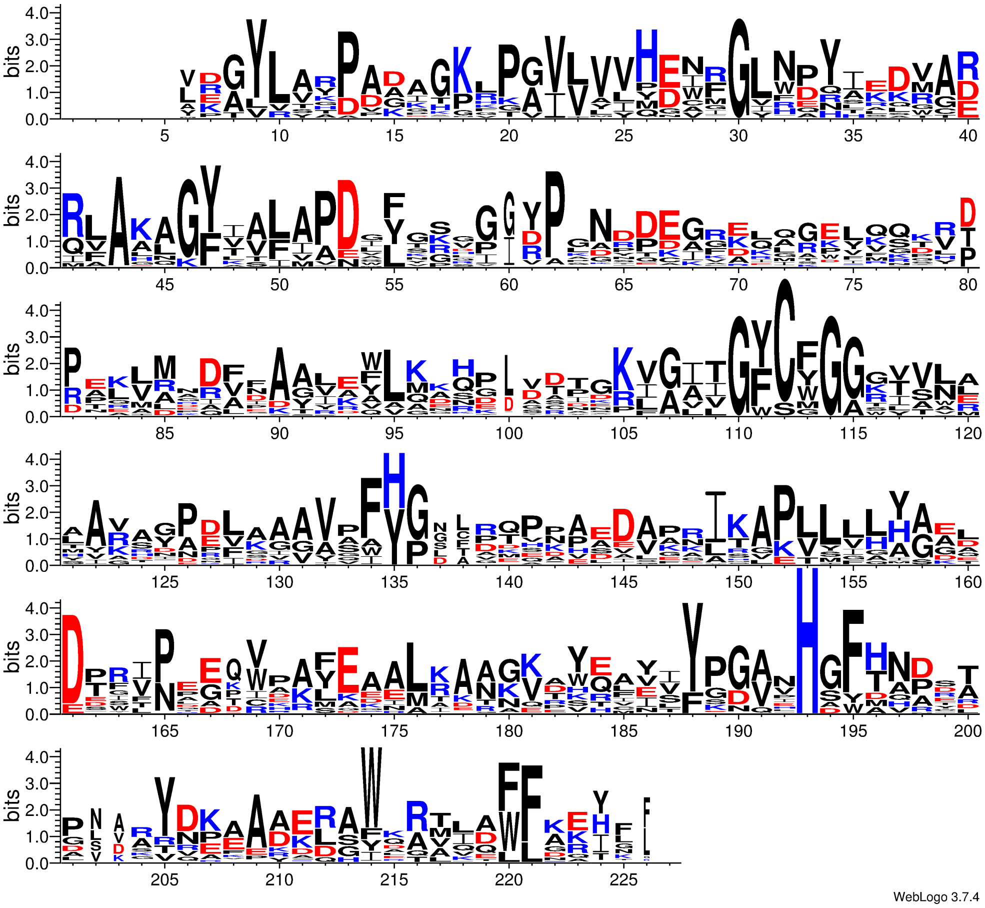

Figure S3: Sequence logo of 10000 MCMC sampled sequences from GREMLIN.

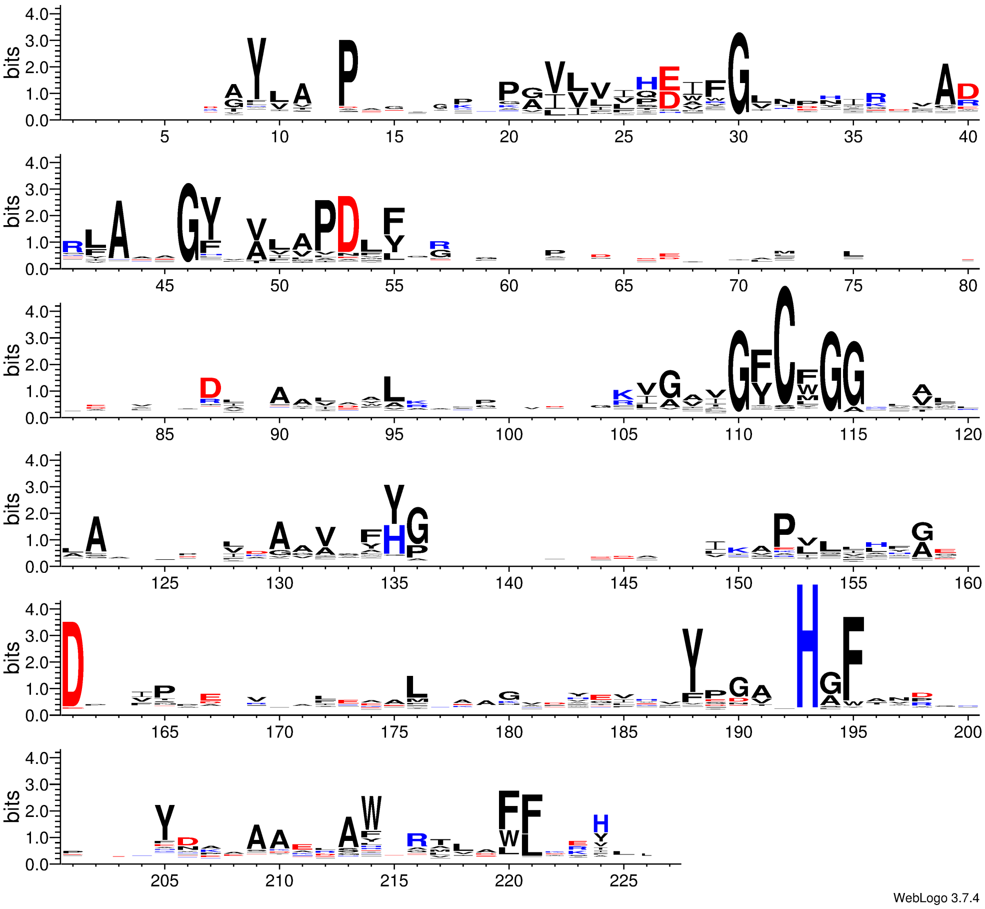

Figure S4: Sequence logo of 10000 MCMC sampled sequences from bmDCA.

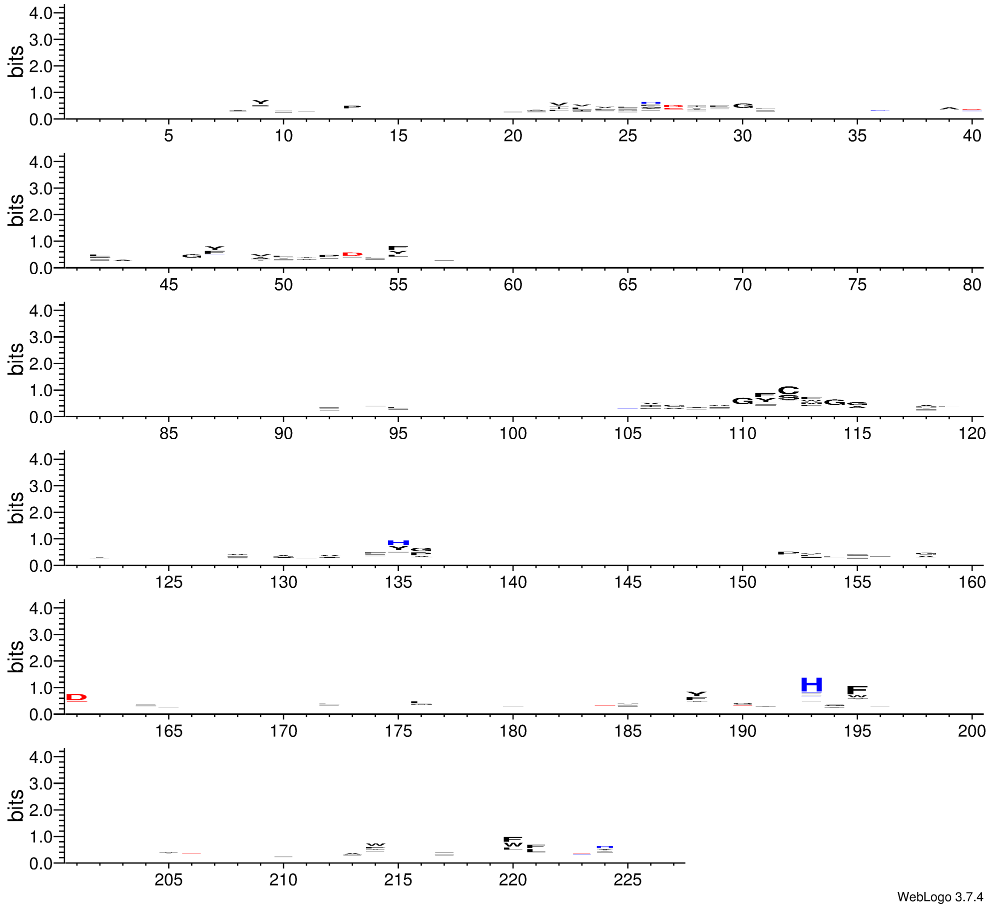

Figure S5: Sequence logo of 10000 MCMC sampled sequences from contrastive divergence.