Contents

Appendix

1. Introduction

Proteins are biomolecules that are fundamental to nearly all biological processes. Their diverse roles include transporting nutrients, catalyzing chemical reactions, providing structural support, and more. The function of a protein is determined by the composition of its primary sequence created from amino acids. A single protein sequence can vary greatly in length and order of its amino acids, leading to a very large number of possible configurations.

The size of the protein sequence space makes exhaustive experimental exploration infeasible. However, the rapid growth of biological databases, driven by advances in sequencing technologies, has transformed biology into a data-rich discipline. Large repositories such as UniProt, Pfam, and the Protein Data Bank store millions of sequences and structures, providing crucial resources for computational approaches[1].

This wealth of data has enabled the development of advanced statistical and machine learning models capable of simulating protein sequence evolution, prediciting structural conformations, and generating novel sequences with desired properties. In addition, breakthroughs in protein structure prediction–most notably the Nobel Prize winning AlphaFold[2] –that computational methods can rival experimental accuracy, in a much cheaper manner.

In this work, we explore the implementation of an autoregressive network for protein sequence generation leveraging the Direct Coupling Analysis (DCA) method to model residue-residue dependencies. In Section 2, the biological background will be laid out. Section 3 will introduce the mathematical foundations. Section 4 will detail previous iterations of similar models, while Section 5 will explore our implementation (and improvements).

2. Biological Background

2.1. Proteins and their structure

Proteins are essential biological molecules responsible for a wide range of functions in living organisms. Despite their functional diversity, all proteins are polymers of the same set of standard building blocks: the 20 canonical amino acids arranged in different assortments.

Amino acids share a common core structure consisting of a central carbon atom bonded to a hydrogen atom, an amino group, a carboxyl group, and a variable side chain. This side chain is the defining feature of each amino acid, giving rise to differences in size, shape, chemical reactivity, and polarity. Proteins are formed by peptide bonds of distinct amino acids, where the term polypeptides comes from. In this process, certain atoms are lost, thus, within a protein sequence, individual amino acids are typically referred to as residues. Generally, protein sequences are made up of between 50 and 2000 amino acids. The ordering of the amino acids dictates how a protein folds into its three-dimensional structure, known as its conformation. Although each conformation is unique, two common folding patterns occur in many proteins: the α helix and the β sheet [3].

Protein structure is broken down into four levels of organization. The amino acid sequence is known as the primary structure. The regularly repeating local structures stabilized by chemical bonds, such as the previously mentioned helices and sheets, make up the secondary structure. The overall three-dimensional shape of a single protein molecule constitutes the tertiary structure. This tertiary structure is what defines a protein's function. Additionally, the structure of protein molecules formed as a complex of multiple polypeptide chains is known as the quaternary structure [3].

Each amino acid is chemically distinct and can occur at any position in a protein chain, giving rise to 20^n possible polypeptide sequences of length n. For a typical protein of 300

amino acids, the number of possible sequences is astronomically large. However, only a small fraction of these sequences are capable of folding into a stable three-dimensional conformation. Natural selection has enabled living organisms to explore sequence space, favoring those sequences that reliably fold into stable structures.

2.2. Protein families and evolutionary information

Proteins do not evolve in isolation; they often belong to protein families, groups of proteins that share a common evolutionary origin, therefore exhibit similar sequence features and functional properties [4]. Over evolutionary timescales, mutations accumulate in these families: some are beneficial, altering the protein activity in ways that give rise to new functions, while many others are neutral and have no effect on stability or activity. Harmful changes, by contrast, disrupt folding or function and are eliminated by natural selection. The result is a collection of homologous proteins that retain overall structural and functional characteristics, but also display sequence variability that encodes the evolutionary history of the family [3].

The study of evolutionary history and relationships among biological entities is referred to as phylogenetics. Protein families provide the domain for phylogenetic analysis, as examining families provides insight that cannot be obtained from a single sequence. Analyzing homologous proteins across diverse organisms enables the detection of important correlated mutations between amino acid positions, representing constraints enforced by evolution. These statistical patterns are exploited by computational approaches to predict three-dimensional structure and understand protein function.

2.3. Multiple sequence alignments

To extract important evolutionary clues from protein families, the homologous sequences need to be organized in a systematic way. This is done through a multiple sequence alignment (MSA), in which three or more sequences are arranged so that homologous sites are placed in the same column [5]. To maximise the positional

correspondence of sequences with varied length, alignment gaps are introduced when necessary.

Multiple sequence alignments reveal patterns of conservation and variation across the family. Conserved positions, those which are unchanged in multiple sequences, represent sites that are critical for maintaining structure or function, while variable positions indicate sites that can tolerate mutations without disruption to the protein. Beyond conservation, MSAs also capture covariation: pairs of positions that mutate in a correlated way across sequences. These covariation signals reflect couplings, where a mutation at one site requires compensation at another to maintain protein integrity [6].

2.4. Protein contacts

One of the earliest challenges for researchers in computational biology–historically encouraged by the CASP (Critical Assessment of Structure Prediction) competition–has been to understand how the linear sequence of amino acids folds into its conformation. Researchers explored spatially close pairs of residues in a protein's folded three-dimensional structure, called contacts. As the structures can be represented in Cartesian coordinates (x, y, z), contacts are defined using distance thresholds. Two residues are generally considered to be in contact if the distance between the selected atoms is below a set threshold, usually 8 Ångströms (Å).

Residues that are far apart in the linear sequence, may be close together in the 3D structure. The contact distances are further categorized into short, medium, and long range predictions. Most computational approaches evaluate the long range contacts separately, as they are the most important for accurate predictions and unsurprisingly, the hardest to predict [6].

2.5. The problem of inference

The central inference problem in computational biology, particularly in the context of protein sequence analysis, is to disentangle the true structural and functional constraints

embedded in protein families from the noisy correlations introduced by evolutionary processes.

Correlations can arise indirectly, for example if A and B were correlated and B and C as well, A and C could appear to be correlated even with a direct interaction. Similarly, shared evolutionary history (phylogeny) and biases in sequence databases can create apparent patterns that obscure the true couplings that govern protein folding and function [7]. Distinguishing direct from indirect correlations is therefore a fundamental challenge in computational biology.

3. Mathematical Foundations

3.1. Proteins as statistical systems

Protein sequences can be thought of as random variables produced by a certain distribution. Each sequence of length L can be written as:

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_L), \quad \sigma_i \in \mathcal{A}, \tag{1}$$

where $\mathcal{A}$ is the alphabet of size $q = 21$ (20 amino acids and the alignment gap) and σ_i are the residue sites. We can organize these sequences into multiple sequence alignments, a table $\left\{\boldsymbol{\sigma}^{(m)}\right\}_{m=1}^{M}$ of M empirical samples. These samples are aligned to have a common length L. Each row in the MSA represents a protein, and each column a position in the sequence.

From the alignments, we can define single and pairwise frequency counts for the columns. The single-site frequency for MSA column i can be computed as:

$$f_i(k) = \frac{1}{M} \sum_{m=1}^{M} \delta\left(\sigma_i^{(m)}, A\right), \quad A \in \mathcal{A}, \quad \text{where } \delta \text{ is the Kronecker delta}^1 \tag{2}$$

while the pairwise frequency of MSA columns i, j is computed as:

[1] $\delta(x, y) := \begin{cases} 0 \text{ if } x \neq y \\ 1 \text{ if } x = y \end{cases}$

$$f_{ij}(A,B) = \frac{1}{M}\sum_{m=1}^{M} \delta\Big(\sigma_i^{(m)}, A\Big)\delta\Big(\sigma_j^{(m)}, B\Big) \tag{3}$$

The empirical frequencies will serve as constraints for the distribution we want to infer.

3.2. Maximum Entropy Principle

To find the probability distribution $P(\boldsymbol{\sigma})$ that will satisfy constraints, in other words reproduce the empirical marginals $f_i(A)$ and $f_{ij}(A,B)$, we can use the maximum entropy principle[8].

The first step to set up the MEP, is extracting information from the system. Usually, this information is given in the form of averages of functions $\langle f_k(x)\rangle$. For example, in a physical system, one could compute average magnetization or energy of the observed system. We also need to define a probability of occupancy of states, $p(x)$, which runs over all the possible states. This distribution has the usual property of mapping each state to a value within 0 and 1 and adding up to 1 when considering all states.

Our uncertainty on the system is expressed quantitatively through Shannon entropy S[9]:

$$S = -\sum_x p(x)\ln p(x) \tag{4}$$

Generally, the distribution which maximizes entropy is the uniform distribution. In this situation, constraints affect the probabilities of states, so the uniform is not suitable. The constraints on our system are:

$$\sum_x p(x) = 1, \quad \text{and} \quad \sum_x p(x) f_k(x) = \langle f_k\rangle \quad (k = 1, ..., m). \tag{5}$$

Selecting Shannon entropy as our measure of information, we can introduce Lagrange Multipliers $\lambda_0, \lambda_1, ...\lambda_m$ to maximize Eq. 4 subject to the constraints Eq. 5, yielding [10]:

$$p(x) = \exp\left(-\lambda_0 - \sum_{k=1}^{m} \lambda_k f_k(x)\right). \tag{6}$$

Define the partition function Z as:

$$Z(\lambda) = \sum_x \exp\left(-\sum_{k=1}^{m} \lambda_k f_k(x)\right). \tag{7}$$

With normalization $\lambda_0 = \ln Z$, we can write the moments as:

$$\langle f_k \rangle = -\frac{\partial}{\partial \lambda_k} \ln Z(\lambda) \tag{8}$$

The entropy of the distribution then reduces to[2]:

$$S_{\max} = \lambda_0 + \sum_{k=1}^{m} \lambda_k \langle f_k \rangle \tag{9}$$

The maximum-entropy (MaxEnt) distribution for this system is exactly the Potts model.

3.3. Connection to statistical mechanics

The Potts model is defined in the context of statistical mechanics, where it was introduced as a generalization of the Ising spin model. Both models were created to explore ferromagnetism and phase transitions.

3.3.1. Ising Model

The Ising model [11] describes a system of spins σ_i arranged on a lattice. Each spin can take one of two possible values:

$$\sigma_i \in \{+1, -1\} \tag{10}$$

The Hamiltonian function, which represents the energy, of a spin configuration $\{\sigma_i\}$ is

[2]The full derivation can be found in Appendix A.

$$H_{\text{Ising}}(\{\sigma\}) = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - h \sum_i \sigma_i, \tag{11}$$

where J is the coupling constant calculated over nearest-neighbor pairs (represented by $\langle i, j \rangle$) that defines the strength between paired interactions. The h term is the external magnetic field acting on the spins.

At thermal equilibrium, the probability of a configuration is given by the Boltzmann distribution:

$$P(\{\sigma\}) = \frac{1}{Z} e^{-\beta H(\{\sigma\})}, \tag{12}$$

with $\beta = 1/(k_b T)$ and partition function

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})}. \tag{13}$$

3.3.2. Potts Model

The Potts model [12] generalizes the Ising model by allowing each spin to take on q possible states. The q-state Potts model states take values in $\{1, 2, ...q\}$. The Hamiltonian is written as

$$H_{\text{Potts}}(\{\sigma\}) = -J \sum_{\langle i,j \rangle} \delta(\sigma_i, \sigma_j). \tag{14}$$

J again represents the ferromagnetic coupling that encourages neighboring spins to align in the same state. The partition function becomes

$$Z = \sum_{\{\sigma\}} \exp\left(\beta J \sum_{\langle i,j \rangle} \delta(\sigma_i, \sigma_j) \right). \tag{15}$$

The Potts model simplifies to an Ising model when $q = 2$.

Putting this together, the probability distribution from the Potts model is written as:

$$P(\boldsymbol{\sigma}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{L} h_i(\sigma_i) + \sum_{1 \le i < j \le L} J_{ij}(\sigma_i, \sigma_j) \right). \tag{16}$$

It is important to note–anticipating further exploration–that the Potts model is over-parametrized: many different (h, J) sets define exactly the same distribution. Without a gauge choice, the parameters will not be uniquely identifiable and the norms of J can be misleading, hindering the process of optimization. Some common gauges, which will be explored in detail further, are reference states, zero-sum, and defining it implicitly via regularization.

3.4. Direct Coupling Analysis

Naively, correlations in the alignment can be captured by covariance, but this simple approach will not be able to separate the direct correlations, arising from structural or functional contacts, and the indirect correlations, propagated via other residues. With empirical frequencies as before, we can define the covariance matrix:

$$C_{ij}(A, B) = f_{ij}(A, B) - f_i(A) f_j(B). \tag{17}$$

A positive $C_{ij}(A, B)$ means that A at i and B at j co-occur more often than expected by chance. The reverse is true for a negative value, where the residues occur less often than expected. The covariance matrix is computed across the whole alignment, and it contains crucial information about pairwise correlations between residues.

From here, all classical DCA methods use the previously defined tools and assume that the rows of the MSAs are independent events drawn from a Potts-model distribution,

$$P(\boldsymbol{\sigma}) = \frac{1}{Z} \exp\left(\sum_{i} h_i(\sigma_i) + \sum_{i<j} J_{ij}(\sigma_i, \sigma_j) \right) \tag{18}$$

where the local field represents the single-column, and the couplings represent the column pairs.

The goal, fitting the Potts model, is defined, but the difficulty of this approach lies in the nontrivial inference problem of learning the interaction parameter J_{ij}. It requires an evaluation of the partition function Z, whose number of terms grows exponentially as the number of sequences grows (q^L). The distinct implementations of DCA introduce approximations to bypass this complex computation. The first implementation of DCA, a computationally costly message passing algorithm, was based on a slowly converging iterative scheme[13]. In this paper, we will explore some important innovations in models using the DCA framework following its conception.

4. Evolution of DCA Methods

4.1. Mean-field DCA (2011)

The mean-field Direct Coupling Analysis (mfDCA) algorithm, introduced by Morcos et al. (2011), provided the first computationally feasible approximation of the Potts model to disentangle direct from indirect correlations in MSAs. mfDCA uses a small-coupling expansion to reduce the inference problem to the inversion of the correlation matrix[14].

4.1.1. Method

To begin, the raw frequency counts suffer from sampling bias, so the weight of highly similar sequences is reduced. Each sequence A^a is assigned a weight

$$m^a = |\{b \in \{1, ..., M\} | \mathrm{sim}(A^a, A^b) > x\}|, \quad x \approx 0.8 \tag{19}$$

where M is the number of sequences in the MSA and sim is their similarity[3]. The effective weight of a sequence a is $1/m^a$, and the total effective number of sequences is

[3]The original paper used "seqid" to represent percentage identity. We chose "sim" as the future methods used this notation.

$$M_{\text{eff}} = \sum_{a=1}^{M} 1/m^a. \tag{20}$$

From these weights, the new regularized empirical single-site frequency is computed as

$$f_{i(A)} = \frac{1}{M_{\text{eff}} + \lambda} \left(\frac{\lambda}{q} + \sum_{a=1}^{M} \frac{1}{m^a} \delta(A, A_i^a) \right) \tag{21}$$

and the pairwise frequency as

$$f_{ij}(A, B) = \frac{1}{M_{\text{eff}} + \lambda} \left(\frac{\lambda}{q^2} + \sum_{a=1}^{M} \frac{1}{m^a} \delta(A, A_i^a) \delta\left(B, A_j^a\right) \right) \tag{22}$$

where $q = 21$ is the amino acid alphabet with the gap symbol, and λ is a pseudocount parameter used for regularization.

The maximum entropy principle is applied to reproduce the empirical single- and pairwise frequencies. This yields the previously derived Potts model distribution

$$P(A_1, ...A_L) = \frac{1}{Z} \exp\left(\sum_{1 \leq i < j \leq L} J_{ij}\left(A_i, A_j\right) + \sum_{i=1} h_i(A_i) \right) \tag{23}$$

with local fields $h_{i(A)}$ and pairwise couplings $J_{ij}(A, B)$, and partition function

$$Z = \sum_{A_1, ..., A_L} \exp\left(\sum_{1 \leq i < j \leq L} \left(A_i, A_j\right) + \sum_{i=1} h_i(A_i) \right) \tag{24}$$

The gauge choice employed in mfDCA is defining a reference state. Typically, it is set as the last amino acid $A = q$. This gives us

$$\forall i, j : J_{ij}(a, q) = J_{ij}(q, a) = h_i(q) = 0 \tag{25}$$

To circumvent the intractable Z computation, mfDCA assumes weak correlations between sites, expanding the exponential in Eq. 23 by the Taylor series to first order. This results in the relation found in Eq. 17 between the couplings and the connected correlation matrix.

The couplings are then approximated as

$$e_{ij}(A,B) = -\left(C^{-1}\right)_{ij}(A,B), \tag{26}$$

where C is treated as a $((q-1)L)\times((q-1)L)$ matrix, and the pair (i,A) is regarded as a single index.

In practice, the correlation matrix is often singular without regularization. mfDCA opts to use a strong pseudocount ($\lambda \approx M_{\text{eff}}$) to stabilize the inversion and prevent spurious large couplings.

Once the couplings are inferred, the next step is to rank residue pairs by their likelihood of physical contact. For each pair (i,j), a two-site model is constructed:

$$P_{ij}^{(\text{dir})}(A,B) = \frac{1}{Z_{ij}} \exp\big(e_{ij}(A,B) + \tilde{h}_i(A) + \tilde{h}_j(B)\big), \tag{27}$$

with auxiliary fields $\tilde{h}_i, \tilde{h}_j$ chosen such that

$$\sum_B P_{ij}^{(\text{dir})}(A,B) = f_{i(A)}, \quad \sum_A P_{ij}^{(\text{dir})}(A,B) = f_{j(B)}. \tag{28}$$

The metric used to rank residue pairs is Direct Information (DI). It is defined as the mutual information of this two-site distribution:

$$\text{DI}_{ij} = \sum_{AB} P_{ij}^{(\text{dir})}(A,B) \ln \frac{P_{ij}^{(\text{dir})}(A,B)}{f_i(A) f_j(B)} \tag{29}$$

The top-scoring pairs are predicted to be structural contacts.

4.1.2. Limitations

While mfDCA represented a breakthrough in usability of DCA, the simplifying approximations impose limitations on the model:

- Weak coupling assumptions: the small-coupling expasion assumes nearly linear correlations, which can underestimate strong epistatic effects in proteins. {add in-text}
- Computational scaling: the inversion of the correlation matrix scales as $\mathcal{O}(((q-1)L)^3)$, which is costly for very large MSAs.
- Pseudocount dependence: due to the algorithm's vast parameter size (around $400N^2$) strong regularization is required. This makes the choice of the pseudocount λ significantly affect performance.

4.2. Pseudo-likelihood Maximization DCA (2013)

The pseudolikelihood maximization DCA (plmDCA) algorithm replaces the intractable full-likelihood fit of the Potts model (which earlier methods approximated) with a tractable product of conditional likelihoods. Concretely, the inverse Potts problem on an alignment of length L becomes L coupled multinomial logistic regressions rather than a single optimization over the global partition function.

4.2.1. Method

We retain the Potts parametrization of fields and couplings introduced in Eq. 16. To mitigate redundacy, sequences are reweighted using an identity threshold $x \in [0.8, 0.9]$: for sequence m, let

$$w_m = \frac{1}{m_m}, \quad \text{with } m_m = |\{a : \mathrm{sim}(\sigma^{(a)}, \sigma^{(b)}) \geq x\}|, \tag{30}$$

and define the effective sample size $M_{\text{eff}} = \sum_{m=1}^{M} w_m$.

Method Setup

The pivotal idea is to optimize pseudolikelihoods. For site r, the conditional distribution given all other sites $\sigma_{\setminus r}$ is

$$P\Big(\sigma_r = l \mid \sigma_{\setminus r}\Big) = \frac{\exp\Big(h_r(l) + \sum_{i\neq r} J_{ri}(l,\sigma_i)\Big)}{\sum_{k=1}^{q}\exp\Big(h_r(k) + \sum_{i\neq r} J_{ri}(k,\sigma_i)\Big)}). \tag{31}$$

The weighted sitewise negative log-pseudolikelihood is

$$g_r(h_r, J_r) = -\frac{1}{M_{\text{eff}}}\sum_{m=1}^{M} w_m \log P\Big(\sigma_r = \sigma_r^{(m)} | \sigma_{\setminus r} = \sigma_{\setminus r}^{(m)}\Big). \tag{32}$$

and the global objective aggregates these points:

$$\mathcal{L}_{\text{pseudo}(h,J)} = \sum_{r=1}^{L} g_{r(h_r,J_r)}. \tag{33}$$

To curtail overfitting, we add convex l_2 penalties,

$$R_{\ell_2} = \lambda_h \sum_{r=1}^{N}\|h_r\|_2^2 + \lambda_J \sum_{1\leq i<j\leq N}\|J_{ij}\|_2^2 \tag{34}$$

and minimize

$$\{h^{\text{PLM}}, J^{\text{PLM}}\} = \arg\min_{h,J}\Big\{\ell_{\text{pseudo}}(h,J) + R_{\ell_2}(h,J)\Big\} \tag{35}$$

The ℓ_2 penalty fixes the gauge implicitly by selecting a unique representative among gauge-equivalent parameters. For scoring with the Frobenius norm, we convert couplings to the zero-sum gauge,

$$J'_{ij}(k,l) = J_{ij}(k,l) - J_{ij}(\cdot,l) - J_{ij}(k,\cdot) + J_{ij}(\cdot,\cdot), \tag{36}$$

where "$\cdot$" denotes an average over the alphabet.

Each g_r is precisely the loss of a multinomial logistic regression (softmax): classes are the q states of σ_r; features are one-hot encodings of $\{\sigma_i\}_{i\neq r}$ with $(L-1)(q-1)$ degrees of freedom after dropping a reference state. Hence plmDCA is implemented as L independent, weighted softmax problems with $\mathcal{L}_2$ penalty (e.g. solved with L-BFGS or mini-batch SGD). Two variants are used in practice: asymmetric PLM, which fits each independently and then symmetrizes averaging:

$$\hat{J}_{ij} \leftarrow \frac{1}{2}\Big(J_{ij}^{(i)} + J_{ij}^{(j)}\Big) \tag{37}$$

and symmetric (joint) PLM, which minimizes $\mathcal{L}_{\text{pseudo}}$ over all parameters at once.

For pair scoring, plmDCA avoids DI as it would introduce a third regularization for the pseudocounts. Instead it, (i) converts to zero-sum gauge J'_{ij}; (ii) computes the Frobenius norm

$$S_{ij}^{\text{FN}} = \left(\sum_{k,l=1}^{q} \big(J'_{ij}(k,l)\big)^2\right)^{\frac{1}{2}}; \tag{38}$$

(iii) applies Average Product Correction (APC) to reduce background/phylogenetic effects,

$$S_{ij}^{\text{CN}} = S_{ij}^{\text{FN}} - \frac{S_{i\cdot}^{\text{FN}} S_{\cdot j}^{\text{FN}}}{S_{\cdot\cdot}^{\text{FN}}}, \tag{39}$$

where $S_{i\cdot}^{\text{FN}}$ and $S_{\cdot j}^{\text{FN}}$ are row/column means and $S_{\cdot\cdot}^{\text{FN}}$ is the grand mean. Residue pairs (i,j) are ranked by S_{ij}^{CN} to predict structural contacts.

4.2.2. Limitations

Despite its practical impact, plmDCA remains sensitive to sampling: accurate contact recovery still requires large M_{eff}, and sparse or biased MSAs degrade estimates. Phylo-

genetic and positional bias persist (reweighting and APC help but do not eliminate them), which can inflate false positives.

4.2.3. Fast plmDCA (2014)

To make plmDCA deployable at scale, two of the original authors and a third collaborator, revisited the optimization and engineering choices of the paper. In this second version, the joint problem is decomposed into L independent, weighted softmax regressions, one for each site, and the final couplings are made symmetric by averaging

$$J_{ij} = \frac{1}{2}\Big(J_{ij}^{(i)} + J_{ij}^{(j)}\Big). \tag{40}$$

This reduces per-solve dimensionality and, crucially, enables trivial parallelization across CPU cores or nodes, which is the primary source of time reduction. Furthermore, regularization remains ℓ_2 and the APC-corrected Frobenius norm still provides a simple and robust ranking criterion. The result is comparable contact accuracy at a fraction of the runtime, making large families and long sequences tractable in practice.

4.3. Boltzmann Machine DCA (2021)

In mfDCA accuracy was traded for speed via the small-coupling inversion; in plmDCA, a product of sitewise conditionals was optimized, avoiding the global partition function. adabmDCA instead proposes a solution closer to the statistical ideal: fitting the full Potts model by maximizing the true likelihood, using Monte Carlo Markov Chains to estimate intractable expectations and an adaptive procedure to keep sampling reliable and efficient. The result is a generative model that (by construction) matches the empirical one- and two-site statistics of the reweighted MSA.

4.3.1. Method

Let the MSA have length L, and M sequences $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_L)$. The maximum-entropy distribution that reproduces chosen moments is the Potts model defined in Eq. 16.

To reduce phylogenetic redundancy, assign each sequence a weight

$$w_\mu = \frac{1}{|\{a : \mathrm{sim}(s^{(a)}, s^{(\mu)}) \geq x\}|}, \quad x \approx 0.8, \quad M_{\mathrm{eff}} = \sum_{\mu=1}^{M} w_\mu \tag{41}$$

Empirical frequencies are then reweighted and smoothed with pseudocount λ as

$$f_i(a) = (1-\lambda) f_i^{\mathrm{data}(a)} + \frac{\lambda}{q}, \quad f_{ij}(a,b) = (1-\lambda) f_{ij}^{\mathrm{data}(a,b)} + \frac{\lambda}{q^2} \tag{42}$$

$$f_i^{\mathrm{data}}(a) = \frac{1}{M_{\mathrm{eff}}} \sum_{\mu} w_\mu \delta\Big(\sigma_i^{(\mu)}, a\Big), \quad f_{ij}^{\mathrm{data}}(a,b) = \frac{1}{M_{\mathrm{eff}}} \sum_{\mu} w_\mu \delta\Big(\sigma_i^{(\mu)}, a\Big)\Big(\sigma_j^{(\mu)}, b\Big) \tag{43}$$

A practical starting point for the system is the profile model $J \equiv 0$, and $h_i^{\mathrm{prof}(a)} = \log f_i(a) + \mathrm{const}$, which already matches the one-site frequencies. In addition, zero or user-provided initial parameters can also work.

Likelihood and moment matching

The average log-likelihood of the MSA under $P(\cdot\,|J,h)$ is

$$\mathcal{L}(J,h) = \frac{1}{M} \sum_{\mu=1}^{M} \left[\sum_i h_i\Big(\sigma_i^{(\mu)}\Big) + \sum_{i<j} J_{ij}\Big(\sigma_i^{(\mu)}, \sigma_j^{(\mu)}\Big) \right] - \log Z(J,h). \tag{44}$$

As an exponential-family model, $\mathcal{L}$ is concave in the natural parameters, so gradient ascent converges to the unique optimum. The gradients are moment gaps:

$$\frac{\partial \mathcal{L}}{\partial h_i(a)} = f_i(a) - p_i(a), \quad \frac{\partial \mathcal{L}}{\partial J_{ij}(a,b)} = f_{ij}(a,b) - p_{ij}(a,b), \tag{45}$$

where p_i and p_{ij} are model marginals under current (J,h). Hence the update

$$\begin{aligned} h_i^{t+1}(a) &= h_i^t(a) + \eta_h \Big[f_i(a) - p_i^{(t)}(a) \Big], \\ J_{ij}^{t+1}(a,b) &= J_{ij}^t(a,b) + \eta_J \Big[f_{ij}(a,b) - p_{ij}^{(t)}(a,b) \Big]. \end{aligned} \tag{46}$$

drives the model toward exact moment matching f = p. The obstacle is that p_i and p_{ij} are not analytically accessible at scale; adabmDCA estimates them by MCMC at each epoch.

Estimating model expectations via adaptive MCMC

At training epoch t, run N_s independent Markov chains, using Metropolis-Hastings [15] or Gibbs [16], each producing N_c samples after an equilibration period T_{eq} and with an inter-sample waiting time T_{wait}. The Monte Carlo estimators are

$$\begin{aligned} p_i^{(t)}(a) &= \frac{1}{N_s N_c} \sum_{\nu=1}^{N_s N_c} \delta\Big(\sigma_i^{(\nu)}(t), a\Big), \\ p_{ij}^{(t)}(a,b) &= \frac{1}{N_s N_c} \sum_{\nu=1}^{N_s N_c} \delta\Big(\sigma_i^{(\nu)}(t), a\Big)\delta\Big(\sigma_j^{(\nu)}(t), b\Big). \end{aligned} \tag{47}$$

Chains may be transient, reinitialized every epoch, or persistent, warm-started from the previous epoch. Equilibration is often sped up through persistence of chains.

Convergence and quality control

A practical stopping criterion is the maximum covariance discrepancy

$$\varepsilon_c = \max_{i,j,a,b} |c_{ij}^{\text{model}(a,b)} - c_{ij}^{\text{emp}(a,b)}|, \quad c_{ij}^{\text{model}} = p_{ij} - p_i p_j, \quad c_{ij}^{\text{emp}} = f_{ij} - f_i f_j \tag{48}$$

with a target $\varepsilon_c \approx 10^{-}2$ In addition to this, some other commonly used diagnostics is the Pearson correlation between c^{model} and c^{emp}, one- and two-site fitting errors, and optionally, a third connected correlation on a subset of triples is used to assess generative fidelity beyond pairwise constraints.

Priors and sparsity

A fully connected Potts model has $\sim \frac{L(L-1)}{2}q^2 + Lq$, meaning a number in the order of 10^7-10^9 parameters for a realistic L [17]. Due to the finite sample size of MSAs, they rarely contain enough independent information to estimate all of them robustly. Without controlling this uncertainty, the consequences could be overfitting, high variance or instability, and bad conditioning in the model. To address these issues, adabmDCA employs two complementary strategies: First, we can place a prior on $P(J, h)$ and maximize the posterior, equivalent to adding penalties to the objective. The two standard choices are the ℓ_1 and ℓ_2 priors:

$$\begin{aligned} R_{\ell_1}(J, h) &= \theta_{1,h} \sum_i \|h_i\|_1 + \theta_{1,J} \sum_{i<j} \|J_{ij}\|_1 \\ R_{\ell_2}(J, h) &= \theta_{2,h} \sum_i \|h_i\|_2^2 + \theta_{2,J} \sum_{i<j} \|J_{ij}\|_2^2 \end{aligned} \tag{49}$$

Under ℓ_2, the gradients are shrunk toward zero:

$$\begin{aligned} \frac{\partial}{\partial h_i(a)} &: f_i(a) - p_i(a) - \theta_{2,h} h_i(a), \\ \frac{\partial}{\partial J_{ij}(a,b)} &: f_{ij}(a,b) - p_{ij}(a,b) - \theta_{2,h} J_{ij}(a,b). \end{aligned} \tag{50}$$

Under ℓ_1, they include subgradient terms that promote exact zeros.

$$\begin{aligned} \frac{\partial}{\partial h_i(a)} &: f_i(a) - p_i(a) - \theta_{1,h}\ \mathrm{sign}(h_i(a)), \\ \frac{\partial}{\partial J_{ij}(a,b)} &: f_{ij}(a,b) - p_{ij}(a,b) - \theta_{1,h}\ \mathrm{sign}\big(J_{ij}(a,b)\big). \end{aligned} \tag{51}$$

The result is that ℓ_2 reduces variance and improves conditioning by smoothly shrinking all parameters. It also selects a unique gauge and tends to preserve the relative ordering of strong couplings. On the other hand, ℓ_1 induces sparsity by zeroing weak parameters, also reducing overfitting, though at a cost of biasing small effects downward. Generally,

in stochastic settings, an elastic-net mix is used (a combination of both parameters), that stabilizes training near zero. In practice a separate parameter is used for fields and couplings, typically regularizing J more strongly than h.

The second method is introducing sparsity via pruning or decimation. The reason for this is that true contact maps in nature are indeed sparse; most residue pairs are not in direct physical contact. Encoding this structural prior can reduces variance and speed up learning. There are two approaches:

1. A priori topology. Reduce the number of parameters by starting from a restricted edge set, for example pairs with high MI, and learn only those J_{ij}, omitting the rest.
2. Information-based decimation. In this approach, start dense and iteratively remove the least informative couplings until target sparsity. This can be done by comparing the KL divergence for a candidate element. This directly controls overfitting by only keeping the paramets that actually affect the model's predictions.

In short, pruning and decimation prevent overfitting by removing parameters that don't materially alter the model, and have the added benefit of aligning with the biological prior of contact sparsity.

4.4. Autoregressive Network DCA

ArDCA was built to explore the ability of generative models for protein design coming from sequence-data. ArDCA emerges as a capable model for the extraction of structural and functional protein information which encoded in rapidly growing protein databases.

4.4.1. Method

In arDCA, the exponential-family MaxEnt distribution is replaced with a conditional probability model, where each residue is predicted from the previous ones. This arises from the chain rule decomposition of the join probability distributions:

$$P(\boldsymbol{S}) = \prod_{i=1}^{L} P(s_i|s_1, ..., s_{i-1}) \tag{52}$$

The parameters are learned by predicting each residue given the previous ones, a similar approach to that taken in NLP methods.

This method has the advantage of being tractable, able to generate new sequences from the given learned parameters, and be scalable.

4.4.2. Key Contributions

This lightweight approach performs at a similar accuracy of previous iterations, but at a substantially lower computational cost (a factor between 10^2 and 10^3) [1]. It presents an important innovation also due to its mathematical advantages, which will be explored further, leading to improved applicability in sequence generation and evaluation.

4.5. Attention DCA - (MAYBE)

- Attention-Potts Model: factored self-attention -> potts model

5. Implementation and Extension

5.1. Implementation details

- Julia -> Python re-implementation (vectorization, frameworks, missing functions)

5.2. Computational challenges

- Benchmarks

5.3. (MAYBE) Improvements:

- Allowing arbitrary sequence length (GPT-style transformers)
- Incorporating attention mechanism (Potts with attention)

Evaluation with Boltz-2 / AlphaFold3

6. Results and Discussion

Bibliography

[1] M. Weigt, "Coevolutionary Analysis of Protein-Protein Interactions." [Online]. Available: https://www.youtube.com/watch?v=IYA8WEsUcG0

[2] J. Jumper *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021, doi: 10.1038/s41586-021-03819-2.

[3] "The Shape and Structure of Proteins," in *Molecular Biology of the Cell*, 4th ed., New York: Garland Science, 2002. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK26830/

[4] European Bioinformatics Institute (EMBL-EBI), "What are protein families?." [Online]. Available: https://www.ebi.ac.uk/training/online/courses/protein-classification-intro-ebi-resources/protein-classification/what-are-protein-families/

[5] M. Wiltgen, "Algorithms for Structure Comparison and Analysis: Homology Modelling of Proteins," *Encyclopedia of Bioinformatics and Computational Biology*. Academic Press, pp. 38–61, 2019. doi: https://doi.org/10.1016/B978-0-12-809633-8.20484-6.

[6] B. Adhikari and J. Cheng, "Protein Residue Contacts and Prediction Methods," *Methods in Molecular Biology*, vol. 1415, pp. 463–476, 2016, doi: 10.1007/978-1-4939-3572-7_27.

[7] N. Dietler, U. Lupo, and A.-F. Bitbol, "Impact of phylogeny on structural contact inference from protein sequence data," *Journal of the Royal Society, Interface*, vol. 20, no. 199, 2023, doi: 10.1098/rsif.2022.0707.

[8] J. Paul Penfield, "Lecture Notes for Information and Entropy." 2003.

[9] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423623–656, 1948.

[10] E. T. Jaynes, "Information Theory and Statistical Mechanics," *Physical Review*, vol. 106, no. 4, pp. 620–630, May 1957, doi: 10.1103/PhysRev.106.620.

[11] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Zeitschrift für Physik*, vol. 31, no. 1, pp. 253–258, 1925, doi: 10.1007/BF02980577.

[12] R. B. Potts, "Some generalized order-disorder transformations," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, no. 1, pp. 106–109, 1952, doi: 10.1017/S0305004100027419.

[13] S. H. H. J. H. T. Weigt M White RA, "Identification of direct residue contacts in protein–protein interaction by message passing," *Proc Natl Acad Sci USA*, vol. 106, pp. 67–72, 2009, doi: 10.1073/pnas.0805923106.

[14] F. Morcos *et al.*, "Direct-coupling analysis of residue coevolution captures native contacts across many protein families," *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. E1293–E1301, 2011, doi: 10.1073/pnas.1111471108.

[15] W. K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970, [Online]. Available: http://www.jstor.org/stable/2334940

[16] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984, doi: 10.1109/TPAMI.1984.4767596.

[17] A. P. Muntoni, A. Pagnani, M. Weigt, and F. Zamponi, "adabmDCA: adaptive Boltzmann machine learning for biological sequences," *BMC Bioinformatics*, vol. 22, no. 528, 2021, doi: 10.1186/s12859-021-04441-9.

A Full Langrage Multipliers Calculation for Maximum Entropy Principle

$$
\begin{aligned}
p_i &= e^{-\lambda-\mu f(x_i)} \\
\sum_i p_i &= 1 \\
\sum_i e^{-\lambda-\mu f(x_i)} &= 1, \text{ factor out } \ e^{-\lambda} \\
e^{-\lambda} \sum_i e^{-\mu f(x_i)} &= 1,
\end{aligned}
\tag{53}
$$

Define the partition function $Z(\mu)$:

$$
\begin{aligned}
Z(\mu) &= \sum_i e^{-\mu f(x_i)}, \text{ thus} \\
e^{-\lambda} Z(\mu) &= 1 \Rightarrow \lambda = \ln Z(\mu)
\end{aligned}
\tag{54}
$$

B Appendix 2