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, composed of 20 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–demonstrated that computational methods can rival experimental accuracy, in a much cheaper manner.

In this work, I 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 my 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. The distinct amino acids are bonded together through peptide bonds to form proteins, also known as polypeptides. In this process, certain atoms are lost, and what remains of each amino acid is called a residue. 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. A protein's conformation defines its function. Although each conformation is unique, two common folding patterns occur in many proteins: the α helix and the β sheet. [2]

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. Finally, if a protein molecule is formed as a complex of more than one polypeptide chain, the complete structure is known as the quaternary structure [2].

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 exhibiting related sequence features and functional properties [3]. 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 [2].

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 allows the detection of correlated mutations between amino acid positions, which in turn represent structural or functional 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 [4]. 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. [5]

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. One of the ways researchers did this was by exploring pairs of residues that are spatially close in a protein's folded three-dimensional structure, called contacts. As the structures can be represented in Cartesian coordinates (x, y, z), the contacts are defined using distance thresholds. Two residues are generally considered to be in contact if the distance between 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 [5].

Contacts are further broken down 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.

2.5. The problem of inference

The specific challenge in computational biology that will be explored in this paper is the prediction and generation of protein sequences given a multiple sequence alignment. As previously mentioned, the protein families contain correlations between residues and we want to build a model to take advantage of that structure. The covariance is confounded by phylogeny and sampling bias [6], but most importantly, the inherent challenge of this problem is to distinguish between the direct and indirect correlations.

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:

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

While the pairwise frequencies of MSA columns i, j are computed as:

$$f_{ij}(A, B) = \frac{1}{M} \sum_{m=1}^{M} \delta\left(s_i^{(m)}, A\right) \delta\left(s_j^{(m)}, B\right) \tag{3}$$

These empirical frequencies will serve as constraints for the model we want to infer.

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

3.2. Maximum Entropy Principle

To find the probability distribution $P(\boldsymbol{S})$ 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[7].

The first step in setting up the MEP, is to extract information from the system. Usually, this information is given in the form of averages of functions $\langle f(x) \rangle$. 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[8]:

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

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:

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

By selecting the Shannon entropy as our measure of information, it allows us to use Lagrange Multipliers for our maximization problem[9]. To maximize Eq. 4 subject to the constraints Eq. 5, we introduce Langrange multipliers $\lambda_0, \lambda_1, ...\lambda_m$ which yield:

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

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 MaxEnt distribution with single and pairwise constraints 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 defined as a way to explore ferromagnetism and phase transitions.

3.3.1. Ising Model

The Ising model 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

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

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

where J is the coupling constant that defines the strength between paired interactions, where the sum is calculated over nearest-neighbor pairs on the lattice (represented by $\langle i,j\rangle$). 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 generalizes the Ising model by allowing each spin to take on q possible states. The q-state Potts model spin can 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:

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

3.4. Direct Coupling Analysis

Naively, correlations in the alignment can be capture 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, which are propagated via other residues. Taking the 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 problem with this approach is that learning the interaction parameter J_{ij} from observations is a nontrivial inference problem. 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 was done through a message passing algorithm. This algorithm was computationally costly as it was based on a slowly converging iterative scheme[10]. 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), provides a computationally feasible approximation of the Potts model used to disentangle direct from indirect correlations in multiple sequence alignments. mfDCA's method includes reweighing the sequences, using maximum entropy formulation of the distribution, and a small-coupling expansion to reduce the inference problem to the inversion of the correlation matrix[11].

4.1.1. Method

The authors found the raw frequency counts to be suffering from sampling bias, so the weight of highly similar sequences was reduced. Each sequence A^a is assigned a weight

$$m^a = |\{b \in \{1, ..., M\} | \mathrm{seqid}(A^a, A^b) > 80\%\}|, \tag{19}$$

where M is the number of sequences in the MSA and seqid is their percentage identity. The effective weight of a sequence a is $1/m^a$, and the total effective number of sequences is

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

From these weights, the regularized empirical single-site frequencies are computed as

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

and the pairwise frequencies as

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

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

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

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

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

Since the number of free parameters in Eq. 23 exceeds the number of constraints, multiple equivalent solutions can be found for the fitting. Therefore, the couplings and fields are measured relative to the reference state, typically the last amino acid $A = q$:

$$\forall i, j : e_{ij}(A, q) = e_{ij}(q, A) = 0, \quad h_i(q) = 0 \tag{25}$$

The partition function Z cannot be computed exactly because it requires summing over q^L sequences. To make the problem tractable, 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

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

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:

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

with auxiliary fields $\tilde{h}_i, \tilde{h}_j$ chose 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 Direct Information (DI) between sites i,j is then defined as the mutual information of this two-site distribution:

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

Residue pairs are ranked by DI_{ij}, and the top-scoring pairs are predicted to be structural contacts.

4.1.2. Limitations

While mfDCA represented a breakthrough in usability of these models, it relies on approximations that impose limitations:

- 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 b, let

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

and define the effective sample size $B_{\text{eff}} = \sum_{b=1}^{B} w_b$.

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

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

The weighted sitewise negative log-pseudolikelihood is

$$g_r(h_r, J_r) = -\frac{1}{B_{\text{eff}}} \sum_{b=1}^{B} w_b \log P\Big(\sigma_r = \sigma_r^{(b)} | \sigma_{\backslash r} = \sigma_{\backslash r}^{(b)}\Big). \tag{32}$$

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,

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

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 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 “”·“” 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 B_{eff}, and sparse or biased MSAs degrade estimates. Phylogenetic 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 he 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 (with the asymmetric coupling penalty effectively halved relative to the symmetric setting), gauge handling is deferred to a single post-fit shift to the zero-sum gauge for scoring, and the APC-corrected Frobenius norm continues to provide a simple, 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

4.3.1. Method

Use Monte Carlo sampling from the Potts model to adjust the parameters until the model reproduces the observed single- and pairwise frequencies.

This is the most faithful approach to the original inference problem. Because of this, it is computationally expensive as each gradient step requires expensive sampling. The cost of this model makes it unsuitable for long protein sequences.

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:

$$P(\boldsymbol{S}) = \prod_{i=1}^{L} P(s_i | s_1, ..., s_{i-1}) \tag{41}$$

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] “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/

[3] 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/

[4] 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.

[5] 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.

[6] 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.

[7] J. Paul Penfield, “Lecture Notes for Information and Entropy.” 2003.

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

[9] 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.

[10] 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.

[11] 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.

A Full Langrage Multipliers Calculation for Maximum Entropy Principle

$$\begin{gathered} p_i = e^{-\lambda-\mu f(x_i)} \\ \sum_i p_i = 1 \\ \sum_i e^{-\lambda-\mu f(x_i)} = 1, \text{ factor out } \; e^{-\lambda} \\ e^{-\lambda} \sum_i e^{-\mu f(x_i)} = 1, \end{gathered} \tag{42}$$

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

$$\begin{gathered} Z(\mu) = \sum_i e^{-\mu f(x_i)}, \text{ thus} \\ e^{-\lambda} Z(\mu) = 1 \Rightarrow \lambda = \ln Z(\mu) \end{gathered} \tag{43}$$