Advanced statistical modeling and variable selection for protein sequences

Sorbonne Université

Kai Shimagaki

April 2021

Contents

APC Average Product Correction

BM Boltzmann Machine

CD Contrastive Divergence

DCA Direct Coupling Analysis

DNN Deep Neural Network

HMM Hidden Markov Model

HP Hopfield model

KL Kullback-Leibler

MaxEnt Maximum-Entropy

MC Monte Carlo

MCMC Markov Chain Monte Carlo

mfHP mean-field Hopfield model

MI Mutual Information

ML Maximum Likelihood

MLE Maximum Likelihood Estimation

MSA Multiple Sequence Alignment

PCD Persistent Contrastive Divergence

PDB Protein Data Bank

PPV Positive Predictive Value

RBM Restricted Boltzmann Machine

Abstract

Over the last few decades, protein sequencing techniques have been developed and continuous experiments have been done. Thanks to all of these efforts, nowadays, we have obtained more than $2 \cdot 10^8$ protein sequence data. In order to deal with such a huge amount of biological data, now, we need theories and technologies to extract information that we can understand and interprete. The key idea to resolve this problem is statistical physics and the state of the art of machine learning (ML). Statistical physics is a field of physics that can successfully describe many complex systems by extracting or reducing variables to be interpretable variables based on simple principles. ML, on the other hand, can represent data (such as reconstruction and classification) without assuming how the data was generated, i.e. physical phenomenon behind of data.

In this dissertation, we report studies of protein sequence generative modeling and protein-residue contact predictions using statistical physics-inspired modeling and ML-oriented methods. In the first part, we review the general background of biology and genomics. Then we discuss statistical modelings for protein sequence. In particular, we review Direct Coupling Analysis (DCA), which is the core technology of our research. Part 2 introduces two generative models based on the DCA method. The Chapter 3 model specifically considers the parameters corresponding to spatial contact, and the Chapter 4 model considers the effect of correlation between data derived from phylogeneies. Chapter 5 proposes variable selection methods that make use of the knowledge of these methods. Part 3 discusses the effects of higher-order statistics contained in protein sequences and introduces deep learning-based generative models as a model that can go beyond pairwise interaction.

Résumé

Au cours des dernières décennies, des techniques de séquençage de protéines ont été développées et des expériences continues ont été menées. Grâce à tous ces efforts, de nos jours, nous avons obtenu plus de $2 \cdot 10^8$ données relative à des séquences de protéines. Afin de traiter une telle quantité de données biologiques, nous avons maintenant besoin de théories et de technologies pour extraire des informations de ces données que nous pouvons comprendre et pour apporter des idées. L'idée clé pour résoudre ce problème est la physique statistique et l'état de l'art de le *machine learning* (ML). La physique statistique est un domaine de la physique qui peut décrire avec succès de nombreux systèmes complexes en extrayant ou en réduisant les variables pour en faire des variables interprétables basées sur des principes simples. ML, d'autre part, peut représenter des données (par example en les reconstruisant ou en les classifiant) sans comprendre comment les données ont été générées, c'est-à-dire le phénomène physique à l'origine de la création de ces données.

Dans cette thèse, nous rapportons des études de modélisation générative de séquences protéiques et de prédictions de contacts protéines-résidus à l'aide de la modélisation statistique inspirée de la physique et de méthodes orientées ML. Dans la première partie, nous passons en revue le contexte général de la biologie et de la génomique. Ensuite, nous discutons des modélisations statistiques pour la séquence des protéines. En particulier, nous passons en revue l'analyse de couplage direct (DCA), qui est la technologie de base de notre recherche. La partie 2 présente deux modèles génératifs basés sur la méthode DCA. Le modèle du chapitre 3 considère spécifiquement les paramètres correspondant au contact spatial, et le modèle du chapitre 4 considère l'effet de la corrélation entre les données dérivées des phylogénéies. Le chapitre 5 propose des méthodes de sélection de variables qui utilisent la connaissance de ces méthodes. La troisième partie traite des effets des statistiques d'ordre supérieur contenues dans les séquences de protéines et présente des modèles génératifs basés sur l'apprentissage profond en tant que modèle pouvant aller au-delà de l'interaction par paires.

Acknowledgements

First and foremost, I would like to express my special gratitude to my supervisor Martin Weigt for his thoughtful and invaluable advice and his generosity and patience during my Ph.D. There were many memorable fortunate circumstances during my doctoral course, but having had him as my supervisor is one of the luckiest events in my life.

I would like to express my deepest gratitude to Olivier Rivoire and Federico Ricci-Tersenghi for their acceptance as thesis reviewers. I am deeply grateful for their accepting to review of such a long thesis while having other important works. I would also like to express my sincere gratitude to Flora Jay, Silvio Franz, and Nataliya Sokolovska for accepting to be a member of the jury.

My deep gratitude goes to my collaborator Francesco Zamponi for his many excellent ideas rooted in deep physics insights. In addition, I would also like to thank Rémi Monasson and Elodie Laine for being my thesis monitoring committees.

My heartfelt thanks go to my colleagues. First of all, I would like to thank my colleagues who already left the laboratory: thanks Pierre, Giancarlo, Edwin, Anna, Carlos, and Edoardo. Exciting scientific stories, cultural stories including French and Italian languages, having dinner, traveling to French villages, and staying together in Havana, all of which are invaluable experiences. I also deeply thank my colleagues who are still in the group: Thanks Nika, Maureen, Juan, Jeanne, Matteo, Barthélémy, Sabrina, Francesco. Thank you for all of your dedicated supports: presentation practices for postdoc interviews at UCR and reviews for my thesis. I learned a lot from you, especially during the process of writing my thesis. Besides these life events, I enjoyed many events with you, discussing various topics, playing chess, disassembling the complex coffee machine, etc.

I would like to thank colleagues in LCQB for having talks, taking lunch/coffee, going out to bars, watching films, and watching many episodes of *Attack on Titan*. I would like to write all of your names, but I'll thank you directly when meeting you next time in Paris.

I would also like to thank Jérôme and Barbara for intriguing discussions about RBMs. They have helped deepen my understanding of the models.

I would express my great appreciation to the people at Politecnico di Torino for letting me stay a couple of weeks. Especially Andrea Pagnani and Carlo Lucibello for your hospitality and scientific discussions. My gratitude also goes to people at the Havana University for letting me participate in the summer school of the complex system. Staying in Cuba for one month was a special experience. Thank you for your heartfelt hospitality.

As for financial support whole three years of my doctoral program, I sincerely appreciate the Honjo International Scholarship Foundation. I am deeply impressed with their dedication to connecting and supporting young scientists around the world.

I would also like to thank people I met in Paris. Especially those I met in the Paris Young Physicists Association and the Japanese Researchers Association. Many interesting discussions and stories stimulated my scientific curiosity. I am also grateful for the experiences of having dinner and traveling together with wonderful people. I had various wonderful experiences at the Maison du Japon in the Cité Internationale Universitaire de Paris. I sincerely hope that such a wonderful facility for students/scientists/artists will be maintained eternally.

Finally, I would like to express my gratitude to the people who are close to me and care for me with special affection. I thank Émilie for being with me and doubling funs while halving sorrows. I sincerely thank my family for understanding and respecting my decisions and ideas. Thank you for your deep affection.

I

Introduction

Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.

—Maire Curie (1867 - 1934)

Chapter 1

Protein Structure and Sequence

In 1962, Sir John Cowdery Kendrew shared the Nobel Prize in Chemistry with Max Perutz because of their pioneering works to determine protein structures using X-ray crystallography. After ten years of technological advance, scientists were asking a question while examining structures of proteins and their mechanical and physiochemical properties: Why and how can an amino-acid sequence obtain its three-dimensional and quickly (in between 100 micro sec and 1 ms [1]), even if the number of the conformational possibilities is astronomically high? Nowadays, this question is known as Levinthal's paradox [2]. Later, an important hypothesis that helps to understand the question, was proposed by Christian B. Anfinsen [3]. Anfinsen's hypothesis or principle states that the structure of a protein is determined not by a biological process, but by the purely physicochemical properties of a particular amino-acid chain and its solvent environment. More precisely, the folding process can be interpreted as the process driven by a free energy landscape, which is uniquely characterized by the amino-acid sequence and the environment, and the natural state is the thermodynamically stable one. Anfinsen's hypothesis implies that it is possible to predict the three-dimensional structure of proteins from these amino-acid sequences.

Predicting a three-dimensional structure as well as understanding the functionality of a protein from a set of amino-acid sequences has been a grand challenge in computational biology for over 50 years since Anfinsen's hypothesis was proposed [4, 5].

1.1 Proteins

Proteins are biological macromolecular compounds that consist of at least one linear chain of amino acids linked by peptide bonds in between. They are the most essentials material in living systems: almost all life activities are directly associated with proteins. Most proteins fold into unique structures and have evolved to bind other proteins, DNA, RNA, or other molecules. Therefore they are required to form and maintain a precise spatial structure and biological functions. Well known function of proteins are:

- Structural proteins, which provide structural components.
- Contractile proteins, which associate with motion of muscles.
- Transport proteins, which carry other substances.
- Storage proteins, which store nutrients.
- Hormone, which regulate body metabolism and nervous system.
- Enzyme, which catalyze biochemical reactions.
- Protection, defend cells from foreign substances.

Each protein has a unique structure and functionality (except for intrinsically disordered proteins, which are not considered in this thesis). Hence if proteins are folded incorrectly or interact with other molecules inappropriately, it causes dysfunction of the system. In some diseases, protein misfolding is believed to be the primary cause, e.g., Alzheimer's disease, Parkinson's disease, Huntington's disease, and many other degenerative and neurodegenerative disorders [6].

1.1.1 Protein sequence

Proteins are made from some amino acids, ranges from about 30 to more than 10,000. The majority of them are 50 to 500 amino-acid residues in length. The amino-acid chains are tied due to covalent peptide bonds linking two consecutive amino acids. Therefore proteins are referred to as polypeptides, and protein chains are called polypeptide chains. The 20 naturally occurring amino acids have three-letter or one-letter abbreviations, and they are always written from N-terminus to C-terminus of a polypeptide chain.

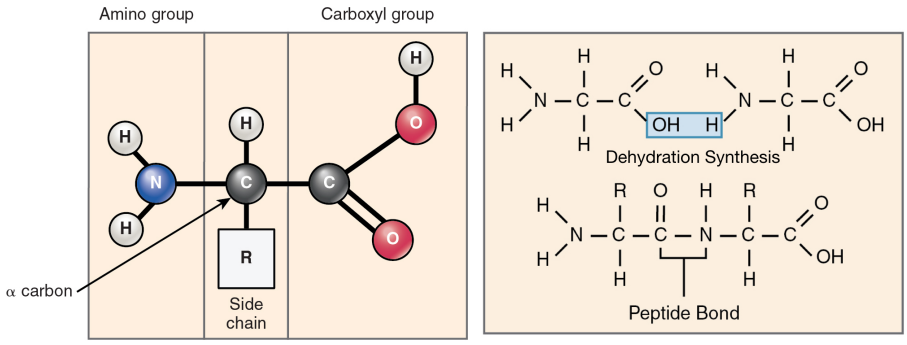

Figure 1.1: **Left**: An amino acid, a building block of proteins, contains both amine (base) and carboxylic acid. It is the R-group, a side chain that characterizes the 20 different natural amino acids. **Right**: The peptide bond, a joint between amino acids, emerges as a consequence of the dehydration synthesis. (Source [7])

Proteins are assembled from a set of 20 unique amino acids, which are organic compounds made of an α-carbon (C_α) with four substitutes, an amino (-NH2) group, a carboxyl (-COOH) group, a variable side chain (-R) and a hydrogen atom (-H) (Fig. 1.1 left).

Amino acids are commonly categorized into four groups, according to the charge and polarity of the side-chain [8]:

1. **Nonpolar** amino-acids group is hydrophobic, which means they are repelled from water since they have very small dipole moment due to the non-polarity. These amino acids have an aliphatic or aromatic side chain. These amino acids are generally located inside the protein core and participate in van der Waals interactions.

2. **Uncharged polar** amino acids are more water-soluble thus more hydrophilic than nonpolar amino acids. These types of amino acids are often found at the surface of proteins.

3. **Aromatic** amino acids. As the name suggested includes an aromatic ring and amphipathic, meaning that it has both hydrophilic and hydrophobic properties.

4. **Charged** amino acids, the side chains often include salt bridges that combine hydrogen bonding and ionic bonding. Amino acids classi-

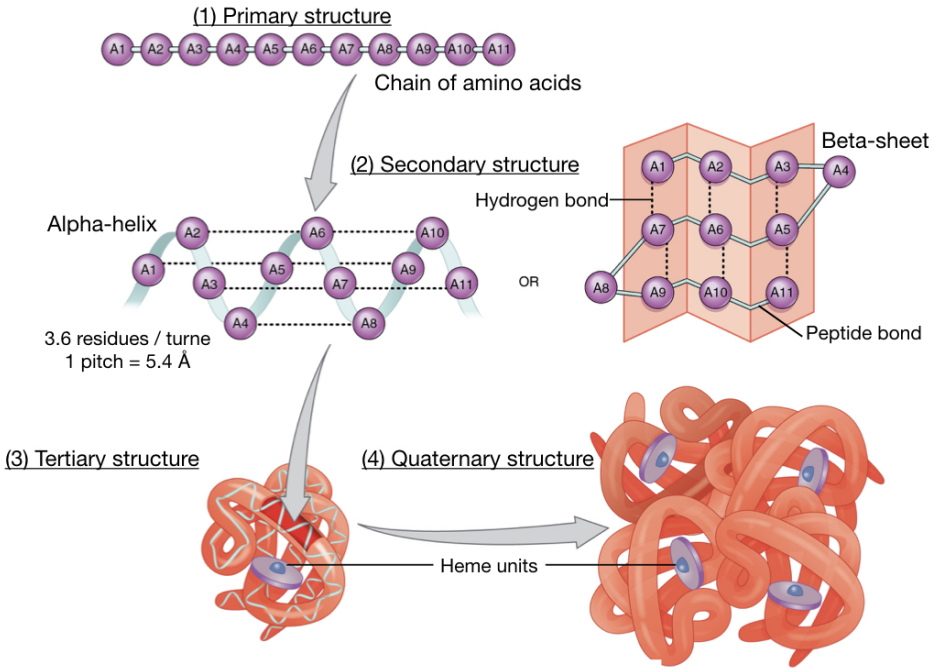

Figure 1.2: (1) The primary structure is given by an amino-acid chain, which is linked formed by peptide bonds. (2) The secondary structure can take a form of an alpha-helix or a beta-sheet. The hydrogen bonds, which are indicated as dotted lines, maintain the secondary structure. (3) The tertiary structure is a three-dimensional shape, which is formed by the further folding and binding of the secondary structure. (4) The quaternary structure occurs as a consequence of interactions between two or more tertiary structures. As an example, we show here hemoglobin, a protein transporting oxygen to body tissues. (Figure is adapted from [7]).

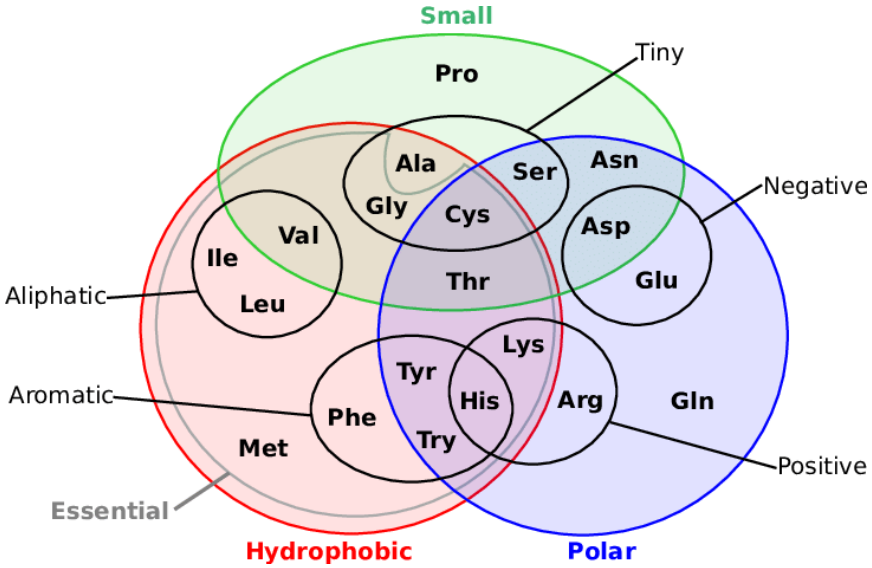

Figure 1.3: Venn's diagram of the 20 naturally occurring amino acids, classified according to some of these physicochemical properties. (Source [9])

fied in this group contribute to the stabilization of three-dimensional protein structure.

As show in Fig. 1.3 these groups are partially overlapped. As an example, Histidine (H) is a polar, charged, and aromatic amino acid.

The structure of proteins involves four different levels, cf. Fig. 1.2 :

1. **Primary structure**, amino-acids sequence, which is defined as the linear polypeptide chains.

2. **Secondary structure**, locally folded structure that forms within a polypeptide chain. It is formed by the interactions among atoms in the backbone chain. The typical secondary structures: ***α*-helices** that is a single peptide chain that forms a helical staircase-like structure. ***β*-sheets** that is a structure formed by segments of a polypeptide chain that line up next to each other and make a sheet-like structure.

3. **Tertiary structure**, a three-dimensional conformation that is formed by an entire polypeptide chain primarily due to interactions among the R-groups of amino acids.

4. **Quaternary structure**, a complex of multiple polypeptide chains.

These shapes of proteins can be divided into three distinct categories based on structure and functionality: globular proteins, fibrous proteins, and membrane proteins [10].
In this section, we explained some general properties of protein sequences. In the next section, we will see the basic properties of protein structures and their role in nature.

1.1.2 Protein structure

Determining the structure of proteins is vital for understanding how thousands of molecules can work together and keep our bodies healthy. For instance, understanding the behavior of small-drug molecules and target proteins in solution is fundamentally important for drug discovery. Thus precise information of structure is essential. The branch of molecular biology that aims to understand molecular structures of proteins and RNA is called structural biology and is known as one of the representative areas of biology.

In general, most protein structures are determined by experimental methods such as X-ray crystallography [1]. X-ray crystallography is the currently most favored technique for structure determination of protein in the sense of the accuracy of structure determination. Most of the structures included in PDB, a comprehensive protein structure database, are determined by this method [11] (there is a description of PDB in Appendix A.1). The structural information of the electron densities are encoded into the intensities of the diffraction patterns of crystallized proteins, which are known as *structure factor* [12]. Creating high-quality protein crystals is crucial for deducing the structural information of a protein. However, it is highly demanding to create relevant protein crystals in most cases, especially for relatively large proteins and certain types of protein. For example, most membrane proteins, which occupy 25% of all protein species and are essential as drug targets (as mentioned in the previous paragraphs), have their structures yet to be determined [13]. Therefore, determining protein structure using a computational approach is scientifically and medically important.

[1] Other commonly used methods are NMR spectroscopy, and with impressive recent progress, cryo-electron microscopy.

In the next section, we will see how to make protein sequences more amenable for applying statistical modeling. We will also discuss some statistical properties of protein sequences.

1.2 Multiple-sequence alignments and protein families

Beneficial proteins (or even if neutral) for adapting environments can spread among descendants and different species. Over the course of propagating proteins, i.e., evolution, associating genetic information can be changed while keeping its functions and a three-dimensional structure. Consequently, protein-sequence also involves mutation effects after the transcription (DNA $\rightarrow$ RNA) and translation (RNA $\rightarrow$ protein). A group that shares the same ancestor is called homologous, and characteristic similarity in such a group is called homology. A group of proteins in the relation of homologous is especially referred to as protein family (we will discuss the details in the next section). Homologous groups are important information resources that enable us to understand the genotype-phenotype relationship.

A codon, which is a combination of three consecutive nucleotides (A T G C) of DNA, can encode naturally occurring 20 different types of amino acids. There are three types of DNA point mutations that can affect protein sequences:

1. Substitution, replacing a nucleotide with another one [2].
2. Deletion, remove one or more nucleotides.
3. Insertion, adding one or more nucleotide.

As a result of the evolution of DNA sequences, the corresponding protein sequences often have different lengths. Typically, consecutive amino-acid insertions and deletions can occur in regions such as loops that do not

[2] In some cases, nucleotide substitutions do not cause any change in amino-acid encoding. For example, both AAA and AAG codons translated to the same amino-acid Lysine. It is called codon degeneracy, and this type of mutation is known as the silent mutation and does not influence the amino-acid sequence ensemble. On the other hand, when mutated to a codon that does not correspond to a naturally occurring amino acid (e.g., TAG, TAA, TGA), translation is stopped, and amino-acid sequence production is stopped at this codon position. This type of codon is called stop codon, and the mutation is called missense mutation, and there are some diseases due to the missense mutation.

contribute to structural stabilization, i.e., it's not involved in structure-preserving constraints. Therefore we cannot naively compare the same residue sites of different sequences due to the insertions and deletions. Hence, what is needed is alignment, which makes the length of the sequence ensemble constant by inserting an additional symbol "–" corresponding to insertion so that the similarity between sequences is maximized.

The basis of sequence alignment is a pairwise alignment algorithm, a method to align two sequences based on dynamic programming.

Probably Smith-Waterman algorithm is the most standard pairwise alignment algorithm [14]. This algorithm requires $O(L_1 L_2)$ computational cost (number of necessarily required operations) to align two sequences, where L_1 and L_2 are lengths of the two sequences to be aligned. It works well, but if we apply this method to M different sequences to align simultaneously, computational cost becomes $\min(L_1, \ldots, L_M)^M \leq L_1 \cdots L_M$, therefore it will be impractical with increasing the number of sequences. However, as we will see in this thesis, in many cases, we need to align thousands of sequences simultaneously. Therefore, alignment methods should be extended to multiple sequence alignment for applying statistical argument. The most commonly used heuristics for multiple alignments are:

1. **Progressive alignment** aligns pairs of the most similar sequences, and then progressively aligns the partially aligned sequences until all sequences are aligned. One of the shortcomings of this method is that the alignment errors mainly due to the initial pairing of sequences propagate and accumulate to the final results. Methods of Clustal series [15] are the most well-known progressive algorithms.

2. **Hidden Markov Model-based alignment** is probabilistic alignment methods that construct probabilities or profile Hidden Markov models (HMMs) of sequences. For the initial construction of HMMs, it needs a small number of well-curated and aligned sequences (around 100 sequences), referred to as seed alignments. Then apply local optimization of HMM alignments scoring system. These HMMs provide us with the ratio of the likelihood that a given sequence originates from a statistically identical distribution to the HMM (or equivalently the ensemble of sequence used for the training of HMM) and the likelihood that the sequence appears by chance (randomly).

Hereafter, we assume HMM-based alignment (cf. the section about pro-

file HMMs below for details). HMMs can also be used to search sequences using HMMs scoring systems. HMMer [16, 17], HH-suite [18] and HMM-HMM [19], commonly used bioinformatics softwares can search sequences, particularly homologous sequences from sequence databases such as UniProtKB [20] and SwissProt [21]. Our study frequently uses Pfam, a database of aligned protein families that contains MSAs and profile HMMs to retrieve protein domain sequences for statistical modelings (there is Appendix A.1 about the databases and explained, UniProtKB, SwissProt, and Pfam database).

1.2.1 Protein families

A protein family comprises evolutionarily related proteins that share a common ancestor protein while being structurally and functionally similar. Hereafter, we assume that MSA refers to be the MSA of the protein family.

MSAs are assembled based on sequence similarities, typically have more than 20% of sequence identity among a protein family. Despite such a small similarity of sequences, corresponding 3D structures and functionalities for each sequence in a protein family are similar or almost identical. Fig. 1.4 shows a part of a multiple sequence alignment for one family.

As we mentioned earlier, each sequence is quite diverse from the others, even if around 20% of sequence identity can be considered the same family [3]. However, some sites are highly conserved (such as sites involving a Cysteine, C, in Fig. 1.4). Some pairs change in correlated ways, meaning that if one site changes, the other site will be adjusted accordingly (correlations could be both negative and positive values). From now on, we represent an MSA as a $M \times L$ rectangular matrix, where M and L are assumed to be the number of sequences and the length of aligned sequences. Typically the number M can be $10^3 \sim 10^5$, and the number of amino acids L is $50 \sim 500$. We define a shorthand representation of an MSA matrix as $(A_i^m)_{i=1,\ldots,L}^{m=1,\ldots,M}$, where $A_i^m \in \mathcal{A}$ is one of the 20 naturally occurring amino-acid letters or the insertion and deletion symbol "gap", $\mathcal{A} = \{A, C, ..., Y, -\}$. Alternatively, we represent an MSA as $(\boldsymbol{A}^{1:M})^t = (\boldsymbol{A}^1, \ldots, \boldsymbol{A}^M)^t$, where $\boldsymbol{A} \in \mathcal{A}^L$ is one sequence, and $1{:}M = \{1, 2, \ldots, M\}$. To describe the states of amino acids simply, we assume the amino-acids letters and the gap to be mapped to numbers $1{:}q = \{1, 2, \ldots, q\}$, where q is the total number of states, i.e., $q = 21$. Note that the following discussions do not depend on the way of mapping

[3] Random sequences have $\sim 5\%$ sequence identity.

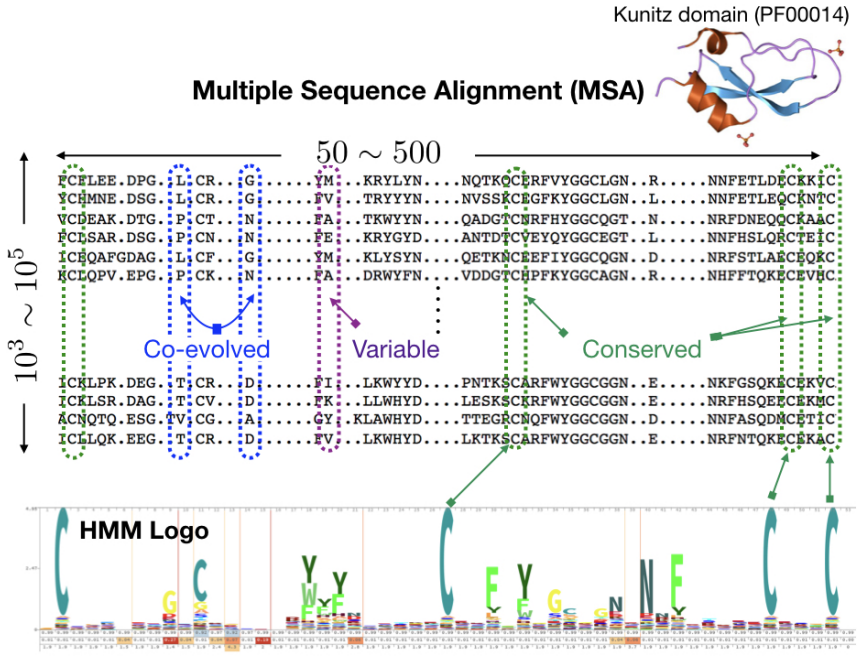

Figure 1.4: **Top**: An MSA is a rectangular matrix containing aligned (amino acid) sequences. Some sites (columns) show conservation, particularly sites involved in Cysteine (C) conserved clearly, in the case of the protein family PF00014 shown here Fig. 1.4. Notably, there are strong pairwise correlations between several sites, which presumably contain information of the co-evolution. **Bottom**: HMM logo, a sequence motif to characterize the protein family based on single-site frequencies and entropy measures. Typically conserved sites are more emphasized.

(this mapping causes no linear relation) [4].

As we can see in Fig. 1.4, the site-specific frequencies seem to characterize MSAs well. That is, the independent single-site variables are statistically important variables to be regarded. Formally, the single-site frequencies are define as

$$f_i(a) = \frac{1}{M} \sum_{m=1}^{M} \delta_{A_i^m, a} \ , \tag{1.1}$$

where i being a site index that runs $\forall i \in 1{:}L$ and a being as one of the alignment states that runs $\forall a \in 1{:}q$. The symbol, $\delta_{a,b}$ is the Kronecker delta that takes value 1 if $a = b$, and 0 otherwise. Indeed, these single-site frequencies permit us to understand and construct MSAs. For example, the single-site entropy H_i of $\{f_i(a)\}$, allows us to estimate the amount of the relative information R_i we can extract from knowing the id site $i \in 1{:}L$,

$$\begin{aligned} H_i &= -\sum_{a=1}^{q} f_i(a) \log_2 f_i(a) \\ R_i &= \log_2(q) - H_i \ . \end{aligned} \tag{1.2}$$

By definition, the information content gives a large value for conservation sites (small entropy). On the contrary, it gives a small value for variable sites (large entropy). Fig. 1.4 shows an example of the sequence logo (denoted as "HMM Logo" in the figure), which represents how each site is conserved using the information content R_i and the frequencies $f_i(a)$ for determining the symbol size.

1.2.2 Sequence motifs

Sequence motifs are site-specific nucleotide or amino-acid sequence patterns that are biologically significant and well characterize sequence ensemble. Typically these are closely associated with conserved and/or functionally relevant sites. Identifying functional regions or regularly appearing patterns in genetic data is a fundamental step to understand biological sequences. For example, to find DNA regions where transcription factors [5] can bind is known as a difficult and practically important problem [22, 23]. Sequence motifs can also be used to find corresponding sequences (methodologically,

[4] States are categorical as known q-state Potts spins

[5] It is a protein that regulates the rate of transcription of genetic information from DNA to mRNA

this task can be done by combining strings search algorithms such as the Rabin-Karp algorithm and Boyer-Moore algorithm [24]). It should be noted here that the profile HMM can also be treated as sequence motifs.

Last, but not least, phylogenetic motifs, sequence motifs that can discriminate the differences between evolutionarily related sequences, are also important motifs [25, 26].

Here, we show the basic methods to find sequence motifs,

1. Position frequency matrices (PFM). Count the number of occurrences of each state for each site in a given alignment. Formally, PFM at position $i \in 1{:}L$ as a state $a \in 1{:}q$ is $M_{ia} = \sum_{m=1}^{M} \delta_{A_i^m, a}$ for an MSA. Another variant, which is essentially equivalent to PFM is the position probability matrix (PPM) which is normalized simply, thus equivalent to the $\{f_i(a) = \frac{M_{ia}}{M}\}$ single-site frequency defined in Eq. 1.1.

2. Position specific scoring matrices (PSSM). Provide an information-theoretic weight that takes into account background frequencies. A PSSM is given as $M_{ia} = f_i(a) \log \frac{f_i(a)}{\bar{f}(a)}$ [6], where $\bar{f}(a) = \frac{1}{L} \sum_{i=1}^{L} f_i(a)$.

These sequence motif methods are based on a preexisting MSA and explicitly depend on site-specific frequencies.

In the study of the Hopfield-Potts models (see in Sec. 4.3.2), we show that the patterns of Hopfield-Potts model are closely related to sequence motifs and can characterize the protein-subfamilies.

1.2.3 Profile Hidden Markov Models

Profile HMMs are probabilistic models that can represent ensembles of biological sequences. They are constructed from curated sequences, which is referred to as the seed alignments [7]. HMMs can be used to align sequences and search sequences based on a query sequence.

Definition – The following explanation of the profile HMMs is based on [14, 27, 28]. A HMM is defined as the two types of variables, visible variables and hidden variables that are denoted as $\boldsymbol{v} = (v_1, v_2, \ldots, v_L)$ and $\boldsymbol{h} = (h_1, h_2, \ldots, h_L)$, respectively. Here, v_i is i-th observable taking one of the states in $\boldsymbol{o} = \{o_1, \ldots, o_q\}$ that correspond to the set of amino acids for

[6] Normally, the pseudocount method is required.

[7] Obtaining a profile HMM from an unaligned sequence is an open issue as it requires simultaneous optimization of the model and multiple alignments.

protein sequences (or nucleotide for DNA or RNA sequences), and h_j is j th state of hidden variable that takes a state from $\boldsymbol{s} = \{s_1, \ldots, s_p\}$, corresponding to match, insertion, and deletion state in protein sequences. As the name implies, the model assumes the Markov property in the hidden space. That is, the probability of hidden variables can be written as

$$p(h_{i+1} = s'|h_i = s, h_{i-1}, \ldots, h_1) = p(h_{i+1} = s'|h_i = s) = t(s'|s) \ , \quad (1.3)$$

where $s, s' \in \boldsymbol{s}$, and $t(s'|s)$ is called *transition probability*.

Similarly, the probability of the state v_i depends only on the state of h_i, formally it can be wwritten as

$$p(v_i = o|h_i = s, v_{i-1}, h_{i-1}, \ldots, h_1, v_1) = p(v_i = o|h_i = s) = e(o|s) \ . \quad (1.4)$$

where $o \in \boldsymbol{o}$, and $e(o|s)$ is called the *emission probability*.

By summarizing these emission and transition probabilities, we can represent the joint distribution of the visible and hidden variables,

$$\begin{aligned} p(\boldsymbol{v}, \boldsymbol{h}) &= p(\boldsymbol{v}|\boldsymbol{h})p(\boldsymbol{h}) \\ p(\boldsymbol{v}|\boldsymbol{h}) &= \prod_{i=1}^{L} e(v_i|h_i) \\ p(\boldsymbol{h}) &= p(h_1) \prod_{i=1}^{L-1} t(h_{i+1}|h_i) \end{aligned} \ . \quad (1.5)$$

Notably, the visible variables are conditionally independent when all hidden variables are given. Therefore, we can efficiently compute the observable sequences if hidden states are known.

Profile HMMs for protein sequences – For the basic HMMs, there are three states $\boldsymbol{s} = \{M, I, D\}$, where M, I and D are referred as *match*, *insertion* and *deletion* states for each site $i \in 1{:}L$. The emission probabilities for these cases are (Fig. 1.5):

1. $e(v_i|h_i = M)$: If a hidden variable is in the *match* state, the visible state emits one of the amino-acid variables according to the site-specific probability estimated by PPM $f_i(a) = \frac{M_{ia}}{M}$.

2. $e(v_i|h_i = I)$: If the hidden variable is in the *insertion* state, the visible state emits an amino-acid symbol according to the background

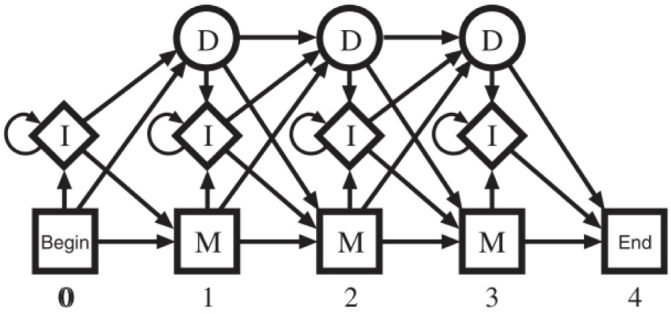

Figure 1.5: A typical profile HMM architecture. Shown squares, diamonds, and circles are denoted as *match*, *insertion*, and *deletion* states, respectively. An edge between the states is denoted as one of the transition probabilities $t(s'|s)$. (Source [14])

frequency $\bar{f}(a)$.

3. $e(v_i|h_i = D)$: If the hidden variable is in the *deletion* state, the visible state emits the gap symbol with probability one.

.

Fig. 1.5 shows a typical profile HMM architecture.

Application of profile HMMs – As we mentioned before, profile HMMs can provide a probability of an observed sequence. Particularly it can be used for finding an ensemble of sequences that are similar i.e., homologous to seed sequences of a given HMM, which corresponds to a high score. This task is known as the *scoring* problem. One way to compute the probability is marginalize the hidden variables $\boldsymbol{h}$,

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \ , \tag{1.6}$$

however, the exact naive computation of Eq. 1.6 is intractable because the complexity increases exponentially as increase the number of hidden states. A more sophisticated approach enables us to compute the score function $p(\boldsymbol{v})$ exactly by applying the forward-backward algorithm, a dynamic programming algorithm.

Second, another important application is to align a sequence to a given profile HMM. This problem is known as the *optimal alignment* problem. This problem needs to insert the gaps optimally. Therefore this problem is equivalent to find the optimal hidden sequence:

$$\boldsymbol{h}^* = \text{argmax}_{\boldsymbol{h}}\ p(\boldsymbol{h}|\boldsymbol{v})\ . \tag{1.7}$$

In order to find optimal hidden states, which equivalent to find an optimal path, typically the *Viterbi algorithm* [29], a dynamic programming algorithm, is employed.

1.3 Co-evolution

Residue-residue interactions are critically important for protein stabilization and acquisition of function. Suppose a mutation occurs at a site that alters the stability of a protein, residues that are influenced by the substituted residue are expected to be compensated over the evolutionary time scale. Therefore correlations between residues of MSAs contain coevolution information.

However, not all strong correlations are due to structural constraints since shared evolutionary history also induces background correlations or phylogenetic correlations.

The first attempts to predict the secondary and tertiary structure of the macromolecule from nucleic-acid sequences based on the insight of co-evolution mechanism was proposed in 1991 [30]. In this study, a mutual information (MI) based method was applied. The first attempt to predict residue-residue contact predict regarding the co-evolutionary information was done by the most naive way i.e., analyzing covariance matrix of residues in 1994 [31]. The first method for predicting MI-based residue-residue contact, waiting another year, was done in 1995 [32].

MI is one of the simplest information-theoretic quantities that can quantify how much random variables X and Y influence each other. The definition of MI between two variables X and Y can be written as follows,

$$MI_{XY} = \sum_{x,y} p_{XY}(x,y) \log \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}\ , \tag{1.8}$$

where, p_{xy} is the joint distribution of x and y, where as p_X and p_Y are

marginalized distribution by one another, $p_X(x) = \sum_y p_{XY}(x,y)$ and $p_Y(y) = \sum_x p_{XY}(x,y)$. The Eq. 1.8 is equivalent to the Kullback-Leiblear divergence (KLD) between the joint distribution p_{XY} and the factorized independent distribution $p_X p_Y$[8].

The MI function for the contact prediction is defined as following,

$$MI_{ij} := \sum_{a,b=1}^{q} f_{ij}(a,b) \log \frac{f_{ij}(a,b)}{f_i(a) f_j(b)} \ , \tag{1.10}$$

where i, j are site indices of protein-sequences running $\forall i, j \in 1{:}L$, and $\forall a, b \in 1{:}q$ are naturally occurring amino acid or gap state. hereafter if there is no specific explanation of i, j, k, we assume these are indices of residues in $1{:}L$, similarly a, b, c are amino acids/gap in $\mathcal{A}$ or $1{:}q$. The definition of $\{f_i(a)\}$ is the same given in Eq. 1.1, and $\{f_{ij}(a,b)\}$ are pairwise-frequencies, which given as the natural generalization of Eq. 1.1 :

$$f_{ij}(a,b) = \frac{1}{M} \sum_{m=1}^{M} \delta_{A_i^m, a} \delta_{A_j^m, b} \ . \tag{1.11}$$

If sites i and j are located in spatial contact, the occupation of amino acids at site i may receive influence from j (and vice versa), therefore MI_{ij} should assume a large value in this case.

Here, "contact" means the minimal distance of heavy atoms of amino acids is less than 8Å, and the correlations are used only for pairs that satisfy $|i - j| > 4$ in our study [9].

The accuracy of the residue contact prediction based on the MI method was drastically improved using phylogenetic correction methods, which is called average product correction method (APC) [33](there is a section about

[8]The Eq. 1.8 can be also represented as

$$MI_{XY} = H_X - H_{X|Y} \tag{1.9}$$

The last result Eq. 1.9 can be interpreted as the mutual information is the amount of information gained from the independent system X when adding the knowledge of the state of the variable Y.

[9]We exclude neighboring 4 sites since there are around 4 amino acids in one pitch of alpha-helix, such short-range predicted contact are therefore considered to trivial result and not very useful in protein structure prediction.

APC in the next section).

The MI-based method can give better predictions than the naive covariance-based residue contact predictions. However, as we will see in the next section, the MI-based approach also leads substantial amount of false positive predictions.

Many such false positives are likely due to spurious correlation. Even though there is no direct relationship between the stochastic variables A-B, the presence of another variable C causes a significant correlation between A-C, similarly B-C, so that an apparent correlation can emerge between A-B as a result. From now on, we refer to such a spurious correlation as a non-direct correlation and a true correlation as a direct correlation.

The following sections discuss statistical protein sequence modeling that can generate artificial sequences and predict direct correlations that can fairly disentangle the direct and non-direct correlations. We also show statistical modeling-based methods such as Direct Coupling Analysis (DCA) outperform the MI-based contact prediction.

Chapter 2

Statistical Sequence Modeling

In this section, we review the maximum entropy (MaxEnt) principle, which enables us to predict structural information of proteins from amino-acid sequence alignment. Then, we formulate the Direct Coupling Alignment (DCA) using the MaxEnt principle. Next, we show the features DCA can learn from MSA. We show that the problems of DCA involved and approximation methods to overcome these problems.

2.1 Maximum-Entropy Modeling

The maximum entropy principle is a method to derive an analytical form of probability distributions, which are initially introduced by E.T. Jaynes [34]. The MaxEnt principle is a framework that explicitly gives us a functional form of a probability distribution by maximizing the entropy under the constraint of reproducing the empirical averaged values of a set of *observables*.

Suppose that we choose a set of observables $\{\mathcal{O}^\mu(\boldsymbol{A})\}_{\mu\in 1:P}$, where $\mathcal{O}^\mu : \mathcal{A}^L \to \mathbb{R}$ is a real-valued function of an amino-acid sequence. Then, we aim at constructing a statistical model $p(\boldsymbol{A})$, which maximizes the entropy,

$$\mathcal{S}[p] = -\sum_{\boldsymbol{A}\in\mathcal{A}^L} p(\boldsymbol{A}) \log p(\boldsymbol{A}) \ , \tag{2.1}$$

under the following certain constraints, $\forall \mu \in 1{:}P$,

$$\sum_{\boldsymbol{A}\in\mathcal{A}^L} p(\boldsymbol{A})\mathcal{O}^\mu(\boldsymbol{A}) = \langle \mathcal{O}^\mu \rangle_{\boldsymbol{A}} \ . \tag{2.2}$$

Here we used a short hand representation $\langle \bullet \rangle_{\boldsymbol{A}} = \sum_{\boldsymbol{A}\in\mathcal{A}^L} \bullet f(\boldsymbol{A})$, where $f(\boldsymbol{A})$ is an empirical distribution.

The advantage of MaxEnt modeling is that it guarantees the desired properties while keeping the minimum necessary conditions for the formulation by the maximization of the entropy. Therefore, it is an objective modeling method.

As a standard approach for solving this type of optimization problem with equality constraints, we apply the method of Lagrange multipliers. We introduce multipliers, $\lambda_\mu,\ \mu \in 0{:}P$, which are the conjugate parameters for all constraints. Finally, a functional of the probability distribution is formulated as,

$$\begin{aligned}\mathcal{F}[p] = &- \sum_{\boldsymbol{A}\in\mathcal{A}^L} p(\boldsymbol{A}) \log p(\boldsymbol{A}) \\ &+ \sum_{\mu=1}^{P} \lambda_\mu \left(\sum_{\boldsymbol{A}\in\mathcal{A}^L} p(\boldsymbol{A})\mathcal{O}^\mu(\boldsymbol{A}) - \langle \mathcal{O}^\mu \rangle_{\boldsymbol{A}} \right) \\ &+ \lambda_0 \left(\sum_{\boldsymbol{A}\in\mathcal{A}^L} p(\boldsymbol{A}) - 1 \right) \ .\end{aligned}$$

Here, λ_0 imposes the normalization of $p(\boldsymbol{A})$ as a probability distribution. By maximizing $\mathcal{F}[p]$ with respect to p, we get the general form of the MaxEnt distribution,

$$p((\boldsymbol{A})|\{\lambda_\mu\}_\mu) = \frac{1}{Z(\{\lambda_\mu\})} \exp\Big(\sum_{\mu=1}^{P} \lambda_\mu \mathcal{O}^\mu(\boldsymbol{A}) \Big) \ . \tag{2.3}$$

We replace λ_0 explicitly by $e^{-(\lambda_0+1)} = Z(\{\lambda_\mu\}) = \sum_{\boldsymbol{A}\in\mathcal{A}^L} \exp\Big(\sum_{\mu=1}^{P} \lambda_\mu \mathcal{O}^\mu(\boldsymbol{A}) \Big)$ the normalization factor (also called partition function in the statistical physics literature). Note that the probability distributions that can be found using the MaxEnt modeling form are always a linear-exponential family, and

the energy function, which is the function defined within the exponential function, explicitly depends on this selected observables $\mathcal{O}^{\mu}(\boldsymbol{A})$.

In the next sections we will see the derivations of profile model and DCA based on the MaxEnt principle.

2.1.1 Profile models

Profile models or single-site models can reproduce the position-specific frequencies of multiple sequence alignments. As we saw in the introduction section, site-specific frequencies $\{f_i(a)\}$ are significant features to characterize MSA and carry the information about conservation. Therefore profile models can be placed as the simplest model to describe MSAs. Actually, in sequence bioinformatics, they are the most important and applied statistical model.

In terms of MaxEnt modeling, the observables of the profile model are each state of residues $\mathcal{O}^{ia}(\boldsymbol{A}) = \delta_{A_i,a}$ for each site $i \in 1{:}L$ and all amino-acid states $a \in 1{:}q$. Therefore, the constraints that have to be satisfied in our profile model are,

$$\sum_{\boldsymbol{A}\in\mathcal{A}^L} p(\boldsymbol{A})\mathcal{O}^{ia}(\boldsymbol{A}) = \sum_{A_i=1}^{q} p(A_i)\delta_{A_i,a} = f_i(a) \ , \tag{2.4}$$

where $f_i(a)$ are the single-site frequencies defined in Eq. 1.1. The Lagrange multipliers, which are the conjugate parameters of these observables are $\lambda_{ia} = h_i(a), \forall i, a$, and we can find the explicit form of profile models,

$$\begin{aligned} p((\boldsymbol{A})|\{h_i(a)\}_{ia}\}) &= \frac{1}{Z(\{h_i(a)\}_{ia})} \exp\Big(\sum_{i,a} h_i(a)\delta_{A_i,a}\Big) \\ &= \prod_{i=1}^{L} \frac{\exp\Big(h_i(A_i)\Big)}{\sum_{b=1}^{q}\exp\Big(h_i(b)\Big)} \ . \end{aligned} \tag{2.5}$$

An important consequence here is that the model parameters of the profile model can be easily obtained by using the relation Eq. 2.4, $h_i(a) = \log f_i(a) + \text{const}$. Note that the defined model factorizes over sites, therefore it is also called the independent site model.

2.2 Pairwise Potts model

We will see that the derivation of the pairwise Potts model (PPM) is a natural generalization of profile models within the MaxEnt modeling. PPM consider also pairwise interaction between amino acids, therefore the observables in MaxEnt are two-site (pairwise) quantities $\mathcal{O}^{iajb}(\boldsymbol{A}) = \delta_{A_i,a}\delta_{A_j,b}$, for all of locies $^{\forall}i, j \in 1{:}L$ and for all of the possible amono-acid letters $\forall a, b \in 1{:}L$. We also keep the single-site observables $\delta_i(a)$. The additional constraints for the MaxEnt modeling for the PPM are thus

$$\sum_{\boldsymbol{A}\in\mathcal{A}^L} p(\boldsymbol{A})\mathcal{O}^{iajb}(\boldsymbol{A}) = \sum_{A_i,A_j=1}^{q} p(A_i,A_j)\delta_{A_i,a}\delta_{A_j,b} = f_{ij}(a,b)\ . \tag{2.6}$$

Here, we used the empirical two point frequencies extracted from data, $f_{ij}(a,b)$. After a little calculation we can get the following probability distribution,

$$p((\boldsymbol{A})|\boldsymbol{h},J) = \frac{1}{Z(\boldsymbol{h},J)}\exp\Big(\sum_{i<j,a,b} J_{ij}(a,b)\delta_{A_i,a}\delta_{A_j,b} + \sum_{i,a} h_i(a)\delta_{A_i,a}\Big)\ , \tag{2.7}$$

with the $J_{ij}(a,b)$ being new Lagrange multipliers. Here $Z(\boldsymbol{h},J)$ is a normalization factor,

$$\begin{aligned} Z(\boldsymbol{h},J) &= \sum_{\boldsymbol{A}\in\mathcal{A}^L} \exp\left(-\mathcal{H}(\boldsymbol{A})\right) \\ -\mathcal{H}(\boldsymbol{A}) &= \sum_{i<j} J_{ij}(A_i,A_j) + \sum_i h_i(A_i)\ , \end{aligned} \tag{2.8}$$

where $\mathcal{A}(\boldsymbol{A})$ is known as the fully connected pairwise Potts energy function or Hamiltonian in physics. Thus, predicting the parameters of PPM means inferring the Hamiltonian [1]. In the case of the PPM, the normalization factor does not factorize due to the interactions between sites. Therefore analytically obtaining forms of partition functions is practically impossible. Moreover, the complexity for calculating it is q^L. Thus, it will soon be impractical to obtain it directly as L increase (typically, $L = 50-500$, hence too large to resolve directly).

The log-likelihood function of the probability distribution for a given

[1] In recent years, inference problems to predict effective Hamilton has attracted significant attention in material informatics [35, 36].

MSA is

$$l^{DCA}(\boldsymbol{h}, \boldsymbol{J}) = \sum_{a,b} f_{ij}(a,b) J_{ij}(a,b) + \sum_{a} f_i(a) h_i(a) - \log(Z(\boldsymbol{h}, \boldsymbol{J})) \ , \quad (2.9)$$

and can be used in maximum-likelihood (ML) inference as an on objective function.

The Lagrange multipliers $\{J_{ij}(a,b)\}$ contain spatial information and are supposed to exclude the spurious correlations successfully, i.e., correlations that are not associated with structural information as discussed in Sec. 1.3. These pairwise parameters are called couplings in physics, and they are also referred to as direct correlations in the context of residue contact prediction.

A framework to predict residues contact based on the model in Eq. 2.9 is known as direct coupling analysis (DCA), which was initially introduced using message passing algorithm [37]. DCA has made significant contributions in statistical genetics and structural biology, through modifying and improving the model itself.

Due to the numerically intractable partition function in Eq. 2.8, some approximation methods were developed. In the following section, we present representative approximation methods for DCA.

2.2.1 Boltzmann DCA

One of the first important contributions of DCA was done using the message passing algorithm. It showed astonishing results but still, it needed to accelerate more since there are convergency issues [2], and the limit of treatable protein sizes were around 80 [37]. In this section, we introduce Boltzmann machine DCA (bmDCA), the algorithmically simplest and versatile model.

The Boltzmann machine (BM) is a generative model for an arbitrary distribution, which was introduced by Geoffrey E. Hinton in 1985 [38]. BM is characterized by an energy function, the mathematical form of the distribution is an exponential family. The probability distribution is learned by MCMC method in general. Historically speaking, BM has been investigated in the field of information theory or machine learning. However, the model is equivalent to the Sherrington–Kirkpatrick model in statistical physics, which was introduced in 1975. Probably, to adapt Boltzmann ma-

[2] This problem may more severe if the underlying interaction networks of residues have many loops.

chine learning to DCA is the most naive and intuitive idea [39].

The equation for updating the model parameters for bmDCA [3] follows exactly the derivative of the likelihood function l^{DCA} introduced in Eq. 2.9 for the probability Eq. 2.8,

$$\begin{aligned}\frac{\partial l^{DCA}(\boldsymbol{h},\boldsymbol{J})}{\partial h_i(a)} &= \langle \delta_{A_i,a}\rangle_{\text{data}} - \langle \delta_{A_i,a}\rangle_{\mathcal{H}} = f_i(a) - p_i(a) \\ \frac{\partial l^{DCA}(\boldsymbol{h},\boldsymbol{J})}{\partial J_{ij}(a,b)} &= \langle \delta_{A_i,a}\delta_{A_j,b}\rangle_{\text{data}} - \langle \delta_{A_i,a}\delta_{A_j,b}\rangle_{\mathcal{H}} = f_{ij}(a,b) - p_{ij}(a,b)\end{aligned} \quad . \tag{2.10}$$

Parameters are updated iteratively using gradient ascent until the fixed point equation Eq. 2.10 goes to zero $^{\forall} i, j, a, b$,

$$\begin{aligned} h_i(a) &\leftarrow h_i(a) + \eta_h \frac{\partial l^{DCA}(\boldsymbol{h},\boldsymbol{J})}{\partial h_i(a)} \\ J_{ij}(a) &\leftarrow J_{ij}(a) + \eta_J \frac{\partial l^{DCA}(\boldsymbol{h},\boldsymbol{J})}{\partial J_{ij}(a)} \end{aligned} \quad . \tag{2.11}$$

The average, $\langle\cdot\rangle_{\mathcal{H}}$ over the statistical model is realized by using MC sampling from the distribution in Eq. 2.8. Note that MC sampling depends on the model parameters, we need to resample therefore at each time for computing the gradient of the log-likelihood with slightly updated model parameters $\{J_{ij}(a,b), h_i(a)\}$. Empirically, this learning algorithm suffers from the overlearning issue. We need to introduce regularization parameter for the optimization, the details of this regularization will be discussed in the following section.

One of the most significant features of bmDCA is that it constructs a generative model for protein sequence: It can reproduce one-point frequencies and two-point connected correlations for the empirical distribution. Moreover, bmDCA was reported to reproduce non-fitted quantities such as the three-point correlation, the principal component analysis (PCA) distribution, and the distribution of Hamming distances. Therefore bmDCA reaches incredibly high reproducibility of the empirical data. Furthermore, bmDCA

[3] The very beginning attempt of structure information which evolved in to DCA was [40], this method used MC to estimate the partition function but different method from bmDCA.

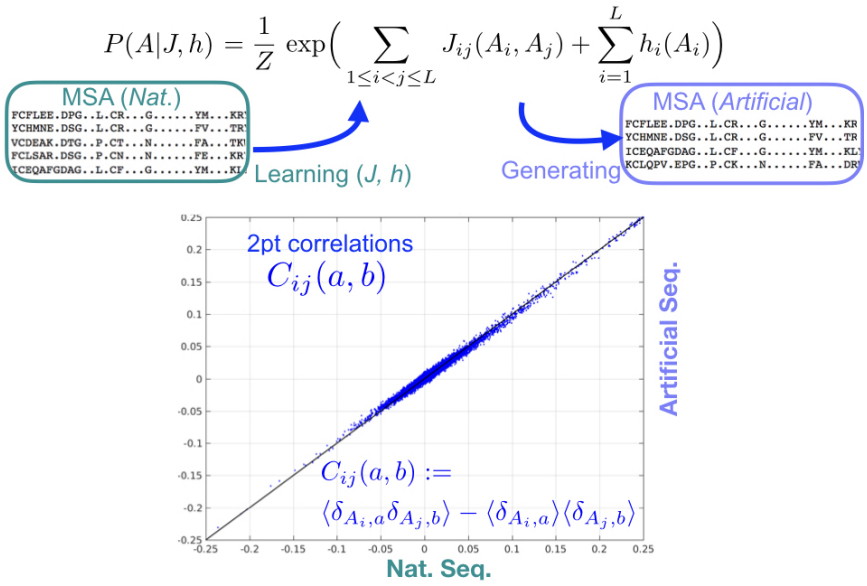

Figure 2.1: Two-point connected-correlations between the natural and artificially generated sequences for the Pfam protein family PF00072. The ensemble of the artificial sequences reproduces the statistics of natural sequences in terms of two-point correlations with high accuracy. The result is expected in BM learning, but technically it is difficult to reproduce the statistics.

can generate artificial functional protein sequences; details are explained in the following section 2.5 about the DCA applications.

2.2.2 bmDCA reproduces a wide range of statistics

MaxEnt modeling of bmDCA is constructed to reproduce single-site and pairwise frequencies, so also the connected correlations of the empirical distribution should be reproduced precisely. We show a scatter plot of two-point connected correlations between a natural MSA (the ensemble of natural sequences used as training data in bmDCA) and an artificial MSA (an ensemble of generated sequences using MC sampling) in Fig. 2.1. The fitting accuracy is significantly high.

Historically, one of the most important contributions of the DCA method is the contact prediction of contacts between residue in the three-dimensional

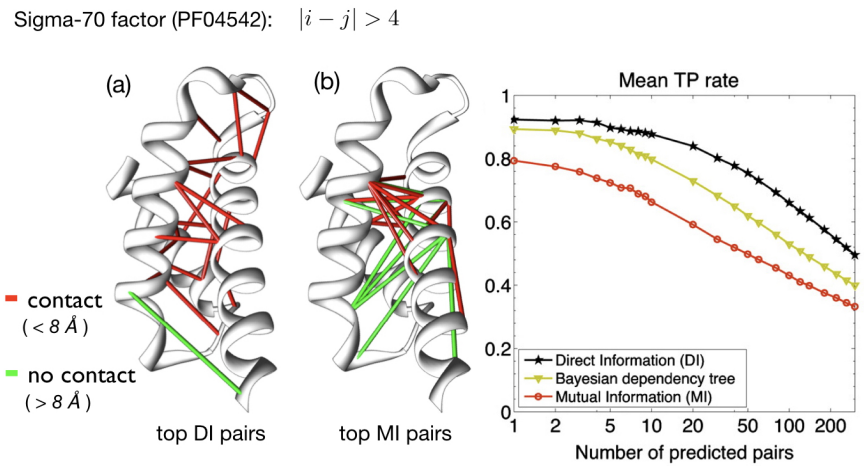

Figure 2.2: **Left**: Contact prediction for a protein family, sigma-70 factor as an example. (a) shows the top 20 DI predictions, which are essentially the DCA prediction, and (b) shows the top MI predictions. Each pair of residues less than the $8\mathring{A}$ are linked with the red edges, and pairs other than that linked with the green edges. **Right**: Mean true positive (TP) rate for 131 domain families, as a function of top-ranked contacts. The curve of the DCA (denoted as DI) outperform the other methods MI and Bayesian dependency tree [41]. Both of the figures are adapted from [42].

structure of the protein. The key idea is that strong couplings $\{J_{ij}(a,b)\}$ are typically associated with spatial interactions due to the co-evolution under the constraint of keeping the structure.

Fig. 2.2 shows the result of residue contact prediction for one exemplary protein family, we also compare the prediction with the MI method mentioned in the section 1.3.

As we mentioned several times, bmDCA reproduces non-fitted quantities, i.e., statistics that are not explicitly imposed to be reproduced. Most strikingly, none of the measurements can significantly distinguish natural sequences and sequences generated by bmDCA, which implies that the pairwise Potts distribution is effectively close to bee indistinguishable from the empirical distribution.

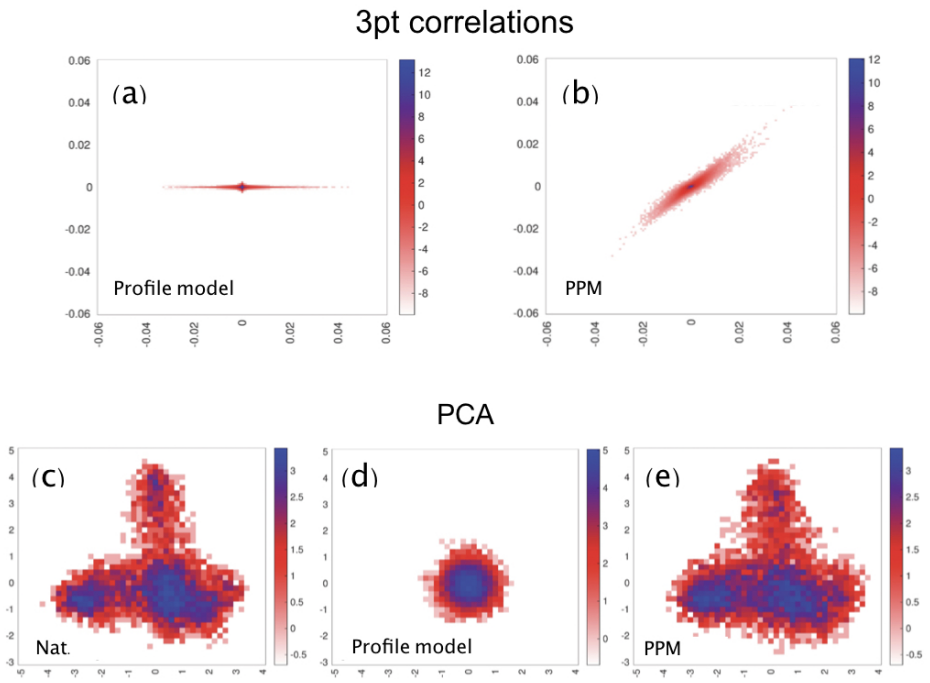

Figure 2.3: **Top**: The three-point correlations. (a) The three-point correlation distribution of the profile model (vertical axis) is almost zero, and there is no association with the three-point correlations of natural MSA (horizontal). However, the three-point correlation from sequences generated by a bmDCA can well reproduce the three-point distribution from the MSA. **Bottom**: PCA projection for (c) the MSA, (d) a profile model, and (e) a PPM. The profile model has a completely different sequence distribution in the PCA space compared with the result of the natural sequence. In contrast, the PPM almost perfectly reproduces the distribution of it. Figures were adapted from [43].

In Fig. 6.1, we show the three-point connected correlations $C_{ijk}(a,b,c)$,

$$\begin{aligned} C_{ijk}(a,b,c) = f_{ijk}(a,b,c) + 2f_i(a)f_j(b)f_k(c) \\ - f_{ij}(a,b)f_k(c) - f_{jk}(b,c)f_i(a) - f_{ki}(c,a)f_j(b) \ , \end{aligned} \tag{2.12}$$

and PCA diagram as examples of the non-fitted quantities for PF00072. As we reviewed in this section, bmDCA can markedly well reproduce statistical properties of natural sequences. However, there are computational and statistical problems:

1. **Computational complexity**: The normalization factor of the probability distribution, the partition function Z, is defined as a sum over a q^L sequence configuration space, hence it is intractable to calculate naively. Although bmDCA can realize the probability distribution without directly estimating the partition function Z, it takes substantial time to reach the equilibrium probability using MCMC, typically between five hours and three days. Therefore, it needs approximations to learn the model with a small computational cost.

2. **Non-uniqueness of estimated parameters**: Most of the estimated model parameters, especially those not involved in contacts, are close to zero but noisy. The gradient-based learning cannot achieve a global minimum with a finite number of MC samples for computing the gradient and with finite learning time. This issue becomes a bottleneck for accurately investigating, and interpreting coupling parameters.

3. **Over-learning**: The bmDCA model depends on too many model parameters. The effective number of available sequences is $10^3 - 10^5$ typically. However, the number of model parameters to be optimized is $10^5 - 10^7$. Therefore, DCA-based learning typically suffers from an over-learning problem.

4. **Selection of variables**: The choice of the observables in MaxEnt might be subjective. Three-body interactions and/or collective variables might be inherently significant variables for protein sequences, while many of the one- and two-site frequencies are not important.

In the next section, we show alternative models to resolve some of these problems. In particular, sections 2.3.1 and 2.3.2 are computationally fast and frequently used as residue contact prediction methods, section 2.3.3 is a computationally fast sequence generative method, and section 2.3.4 is contact prediction methods with a small number of model parameters and is closely associate with our works.

2.3 Approximations of DCA

In this section, we review some well-established approximation methods that we use typically in our study are selected.

First, we review the mean-field DCA, the fastest DCA-based algorithm, to predict the residue contacts. Then we discuss the pseudo-likelihood DCA, which is also the first algorithm having the asymptotic consistency. Third, we review the very recent autoregressive DCA, which is the currently fastest algorithm to generate artificial sequences as well as to predict contact prediction. Finally, we discuss Hopfield-Potts DCA, which can be used for the contact prediction problem and can reduce the number of model parameters.

2.3.1 Mean-field DCA

Mean-field DCA (mfDCA) was introduced in 2011 [42]. The computational cost of this method is proportional to $O(q^3L^3)$, which is relatively small, hence requiring computational time is substantially faster than bmDCA. The downside is that this model cannot be used as a generative model. Parameters of mfDCA are typically associated with a low-temperature regime $(1 < T)$ due to overestimated coupling parameters.

The algorithm of mfDCA is based on a small-coupling expansion which is equivalent to a high-temperature expansion in physics. The following derivation is based on the article [42].

First, we introduce a perturbation parameter into the Hamiltonian,

$$-\mathcal{H}(\boldsymbol{A}|\boldsymbol{h},\boldsymbol{J};\alpha) = \alpha\sum_{i<j} J_{ij}(A_i,A_j) + \sum_{i=1}^{q} h_i(A_i)\ . \tag{2.13}$$

If we put $\alpha = 0$, then the model will be an independent-site model or profile model. If $\alpha = 1$, the standard pairwise model appears. Furthermore, we introduce the Gibbs potential,

$$-\mathcal{G}(\boldsymbol{p}|\boldsymbol{J};\alpha) = \max_{\boldsymbol{h}}\left\{\log\left(\sum_{\boldsymbol{A}\in\mathcal{A}^L}\exp(-\mathcal{H}(\boldsymbol{A}|\boldsymbol{h},\boldsymbol{J};\alpha))\right) - \sum_{i=1}^{L}\sum_{a=1}^{q-1} h_i(a)p_i(a)\right\}\ , \tag{2.14}$$

, which is the Legendre transformation of the Helmholtz free energy $\mathcal{F} =$

$-\log Z$. Therefore short-hand representations of the Legendre transformation and its inverse are:

$$\mathcal{G}(\boldsymbol{p}|\boldsymbol{J};\alpha) = \mathcal{F}(\boldsymbol{h}|\boldsymbol{J};\alpha) - \boldsymbol{h}^t\boldsymbol{p}$$
$$\mathcal{F}(\boldsymbol{h}|\boldsymbol{J};\alpha) = \mathcal{G}(\boldsymbol{p}|\boldsymbol{J};\alpha) + \boldsymbol{h}^t\boldsymbol{p}$$

Here, $p_i(a)$ is a single-site frequency of $\mathcal{H}(\alpha = 1)$, and $\boldsymbol{p} \in \mathbb{R}^{qL}$ is a vector representation. Note that the sum over the amino acid takes only 1 to $q-1$ due to the lattice-gas gauge choice in order remove redundant degrees of freedom (we will discuss the gauge invariance in section 2.4.2). Since $h_i(a)$ and $p_i(a)$ are connected by the Legendre transformation,

$$h_i(a) = \frac{\partial \mathcal{G}(\boldsymbol{p})}{\partial p_i(a)}, \qquad p_i(a) = \frac{\mathcal{F}(\boldsymbol{h})}{\partial h_i(a)} \ , \tag{2.15}$$

we also have,

$$(C^{-1})_{ij}(a,b) = \frac{\partial h_i(a)}{\partial p_i(b)} = \frac{\partial^2 \mathcal{G}(\boldsymbol{p})}{\partial p_i(a) p_i(b)} \ . \tag{2.16}$$

Here, C is the connected correlation matrix, $C_{ij}(a,b) = f_{ij}(a,b) - f_i(a)f_j(b)$, which has $L(q-1)$ dimensions and indices $i,j \in 1{:}L$, $a,b \in 1{:}(q-1)$, this restriction is needed to make C be as invertible matrix (practically it needs pseudo-count, see the following section). For simplicity, we neglect the explicit dependence on $\boldsymbol{J}, \alpha$ in Eq. 2.15.

When we expanding the Gibbs potential up to the first order with respect to α, we get,

$$\begin{aligned} \mathcal{G}(\boldsymbol{p}|\boldsymbol{J};\alpha) &= \mathcal{G}(\boldsymbol{p}|\boldsymbol{J};0) + \frac{\partial \mathcal{G}(\boldsymbol{p}|\boldsymbol{J};\alpha)}{\partial \alpha}\Big|_{\alpha=0} \alpha + \mathcal{O}(\alpha^2) \\ &= \mathcal{G}(\boldsymbol{p}|\boldsymbol{J};0) + \Big\langle \sum_{i<j} J_{ij}(A_i, A_j) \Big\rangle_{\alpha=0} \alpha + \mathcal{O}(\alpha^2) \ . \end{aligned} \tag{2.17}$$

If we expand the Eq. 2.17 around $\alpha = 0$, i.e., the profile model, then put $\mathcal{G}(\boldsymbol{p}|\boldsymbol{J};\alpha = 1)$, we get,

$$\mathcal{G}(\boldsymbol{p}|\boldsymbol{J}) = \sum_{i=1}^{L}\sum_{a=1}^{q} p_i(a)\log(p_i(a)) - \sum_{i<j}\sum_{a,b=1}^{q} J_{ij}(a,b)p_i(a)p_j(b) \ . \tag{2.18}$$

Here, we put also $p_i(q) = 1 - \sum_{a=1}^{q-1} p_i(a)$. The aim is to find the self-consistent equation for the parameters $\{h_i(a)\}$. For this purpose, we perform

the first and second partial derivatives of this first-order expansion of the Gibbs potential Eq. 2.18 with respect to the single point frequency $p_i(a)$. We find self-consistent equations for the model parameter $h_i(a)$:

$$p_i(a) \propto \exp\left(h_i(a) + \sum_{j \in \partial i} \sum_{b=1}^{q} J_{ij}(a,b) p_j(b)\right) , \tag{2.19}$$

which take the characteristic form of mean-field equations. We further determine the inverse of the connected correlation matrix as,

$$(C^{-1})_{ij}(a,b) = \begin{cases} -J_{i,j}(a,b) & \text{for } i \neq j \\ \frac{\delta_{a,b}}{p_i(a)} + \frac{1}{p_i(q)} & \text{for } i = j \end{cases} . \tag{2.20}$$

Therefore, the inference problem for the couplings $J_{ij}(a,b)$ can be solved by just plugging the empirical connected correlation C into the Eq. 2.20, and by computing the inverse matrix. Surprisingly, the simple inverse of the covariance matrix gives accurate contact prediction. The inverse of the covariance matrix is probably the inherent quantity (there is a discussion why such a simple method can correctly deduce direct-interaction based on [44] in Chap. 5). Note that the covariance matrix is not a full rank matrix, in fact there are L zero eigenvalue modes [4]. The standard remedy for the rank deficient is a gauge transformation (see the technical section 2.4.2). However, even the reduced covariance matrix tends to be rank deficient, and it happens for sure if the number of the sequences is smaller than the $(q-1)L$, since

$$C = M^{-1} \sum_{m=1}^{M} (\boldsymbol{\delta}^m - \langle \boldsymbol{\delta} \rangle)(\boldsymbol{\delta}^m - \langle \boldsymbol{\delta} \rangle)^t , \tag{2.21}$$

where $\boldsymbol{\delta}$ is a vector representation of the single-site observables, $\boldsymbol{\delta}^m = (\delta_{A_1^m 1}, \ldots, \delta_{A_L^m q})^t$. The pseudo-count method, which adds a small positive value to empirical counts is normally used for avoiding the rank deficient issue [42] (the detail is discussed in the technical section 2.4.1).

[4] $\sum_{a=1}^{q} C_{ij}(a,b) = \sum_{b=1}^{q} C_{ij}(a,b) = 0, \forall i,j \in \{1,\ldots,L\}$, therefore we can easily construc L different zero modes, $\underbrace{(0\ldots,0,1,0,\ldots,0)^t}_{=L} \otimes \underbrace{(1,\ldots,1)^t}_{=q}$

2.3.2 Pseudo-likelihood maximization DCA

plmDCA can accurately predict contacts with a small computational time, thereby out-performing mfDCA [45].

The first use of the pseudo-likelihood method for Markov Random fields was in 2010 [46], but it used an Ising model. Then the idea was adapted to protein sequence modeling in 2011 [47]. The objective function of plmDCA is not the likelihood for the pairwise distribution. Instead, it maximizes the log-likelihood of conditional single-site distributions, which we refer to as pseudo-likelihood function,

$$p(A_i^m = a|\boldsymbol{A}_{\setminus i}^m) = \frac{\exp(h_i(a)) + \sum_{j\partial\in i} J_{ij}(a, A_j^m)}{\sum_{b=1}^q \exp(h_i(b)) + \sum_{j\partial\in i} J_{ij}(b, A_j^m)} \ , \tag{2.22}$$

with $\boldsymbol{A}_{\setminus i} = (A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_L)$. In consequence, the pseudo-likelihood function can be written as,

$$l^{plm}(\boldsymbol{h}, \boldsymbol{J}) := \sum_{i=1}^{L} l_i^{plm}(\boldsymbol{h}_i, \boldsymbol{J}_i) \ , \tag{2.23}$$

where the local pseudo-likelihood is given by

$$\begin{aligned} l_i^{plm}(\boldsymbol{h}_i, \boldsymbol{J}_i) &= M^{-1} \sum_{m=1}^{M} \log p(A_i^m|\boldsymbol{A}_{\setminus i}^m) \\ &= \sum_{a=1}^{q} f_i(a) h_i(a) + \sum_{j\partial\in i} \sum_{a,b=1}^{q} f_{ij}(a,b) J_{ij}(a,b) - z_i \ , \end{aligned} \tag{2.24}$$

where z_i is a position-specific normalization constant, depending on $\boldsymbol{A}_{\setminus i}^m$.

$$z_i := M^{-1} \sum_{m=1}^{M} \log \Big(\sum_{b=1}^{q} \exp(h_i(b)) + \sum_{j\partial\in i} J_{ij}(b, A_j^m) \Big) \ . \tag{2.25}$$

It is known that as the number of sequences increases, the estimators of plmDCA for the Gibbs distribution become closer to the estimator of the full likelihood [48] [5]. To find the optimal solution for the pseudo-likelihood, there are two ways of strategies. The first approach is to maximize the $l^{plm}(\boldsymbol{h}, \boldsymbol{J})$ directly, this method is called *symmetric* PLM [49]. A solution of

[5] This is a peculiarity of the pseudo-likelihood function, not plmDCA.

the pseudo-likelihood estimator is,

$$\{\boldsymbol{h}^*, \boldsymbol{J}^*\} = \underset{\boldsymbol{h},\boldsymbol{J}}{\operatorname{argmax}}\ l^{plm}(\boldsymbol{h}, \boldsymbol{J}) \ . \tag{2.26}$$

An alternative approach is to maximize each local likelihood $l_i^{plm}(\boldsymbol{h}_i, \boldsymbol{J}_i)$, separately $\forall i \in 1{:}L$. Note that this method gives asymmetric couplings since the local likelihoods, given: $l_i^{plm}(\boldsymbol{h}_i, \boldsymbol{J}_i) \rightarrow \boldsymbol{J}_{ij}^{*i}$ and $l_j^{plm}(\boldsymbol{h}_j, \boldsymbol{J}_j) \rightarrow \boldsymbol{J}_{ij}^{*j}$. These couplings results are generally different from each other. In practice, an arithmetic average of the two couplings is used for contact prediction. Therefore, the final solution should be written as

$$\boldsymbol{J}_{ij}^* = \frac{1}{2}(\boldsymbol{J}_{ij}^{*i} + \boldsymbol{J}_{ij}^{*j}) \ . \tag{2.27}$$

This approach is called *asymmetric* PLM [45]. Its computational cost is smaller than that of symmetric PLM.

Recently, another related method called auto-regressive DCA (arDCA) was introduced. Although arDCA and plmDCA are closely associated with each other, the plmDCA is not a generative model, while arDCA is a generative model. In the next section we will briefly review arDCA.

2.3.3 Autoregressive DCA

Auto-regressive DCA (arDCA) was introduce in 2021 [50]. The key idea of the arDCA is to factorize the joint probability into conditional probabilities based on Bayes decomposition, which means

$$p(\boldsymbol{A}|\boldsymbol{h}, \boldsymbol{J}) = p(A_1|\boldsymbol{h}_1) \prod_{i=1}^{L} p(A_i|\boldsymbol{A}_{1:(i-1)}; \boldsymbol{h}_i, \boldsymbol{J}) \ , \tag{2.28}$$

where $\boldsymbol{A}_{1:(i-1)} = (A_1, \ldots, A_{i-1})$. The mathematical form of each factor is defined as

$$p(A_i = a|\boldsymbol{A}_{1:(i-1)}; \boldsymbol{h}_i, \boldsymbol{J}) = \frac{\exp(\sum_{1\leq j<i} J_{ij}(a, A_j) + h_i(a))}{\sum_{b=1}^{q} \exp(\sum_{1\leq j<i} J_{ij}(b, A_j) + h_i(b))} \ , \tag{2.29}$$

and $p(A_1|\boldsymbol{h}_1)$ is a standard profile distribution. The arDCA model gives astonishing results; the contact prediction is almost as good as plmDCA, which is the currently best contact prediction method. Furthermore, arDCA can be used as a generative model, and the reproduction of the statistics is as good as bmDCA. One of the significant differences from bmDCA is that the

computational time is 100 to 1000 times faster due to the accessibility of the normalization rigorously and fastly. There were many situations in which the application of DCA was limited due to the time-consuming sampling using conventional generative models. Therefore, arDCA will be definitely useful to sample sequences in a short time and at a large scale.

In the next section, we introduce another mean-field approach, which has pairwise couplings but effectively reduces the number of model parameters by applying the low-rank representation of couplings.

2.3.4 Mean-field Hopfield-Potts DCA

The method of mean-field Hopfield-Potts DCA was introduced in 2013 [51]. This method can effectively reduce the number of model parameters by introducing a low-rank representation of the coupling matrix. Each non-zero mode of the coupling matrix is called patterns, and is associate with an eigenvector of Pearson correlation matrix in the mean-field approximation. These patterns are also closely related to *position-specific* scoring matrices, which are defined as characteristic sequence motifs as explained in Sec. 1.2.2.

While the dimensionality of the covariance matrix is $L(q-1)$, the number of essential dimensions is much smaller. These inherently significant directions are called *patterns* as the following $q \times L$ matrix $\boldsymbol{\xi} = \{\xi_i(a)\}$, with $i \in 1{:}L$ and $a \in 1{:}q$.

The *log-score* of a sequence

$$S(\boldsymbol{A}|\boldsymbol{\xi}) = \Big(\sum_{i=1}^{L} \xi_i(A_i) \Big)^2 . \tag{2.30}$$

Within the Hopfield-Potts model, the statistical modeling of protein sequences $\boldsymbol{A}$ is characterized by the combination of P log-scores, given by P patterns:

$$p(\boldsymbol{A}) = \frac{1}{Z} \exp\left(\frac{1}{2L} \sum_{\mu=1}^{P_+} S(\boldsymbol{A}|\boldsymbol{\xi}^{+,\mu}) - \frac{1}{2L} \sum_{\nu=1}^{P_-} S(\boldsymbol{A}|\boldsymbol{\xi}^{-,\nu}) \right) . \tag{2.31}$$

The patterns denoted as $\boldsymbol{\xi}^{+,\mu}$ with pattern index $\mu \in 1{:}P_+$ and $\boldsymbol{\xi}^{-,\nu}$ with pattern index $\nu \in 1{:}P_-(= P - P_+)$ are called *attractive*-pattern and *repulsive*-pattern, respectively. The log-likelihood function of the mean-field Hopfield-

Potts model leads [6]

$$
\begin{aligned}
&\mathcal{L}\Big[\{\boldsymbol{\xi}^{+,\mu},\boldsymbol{\xi}^{-,\nu}\}_{\mu,\nu}|\boldsymbol{A}^{1:M}\Big] \\
&= \sum_{ia} f_i(a)\log f_i(a) \\
&+ \frac{1}{2L}\sum_{\mu=1}^{P_+}\sum_{ia,jb} f_{ij}(a,b)\xi_i^{+,\mu}(a)\xi_j^{+,\mu}(b) - \frac{1}{2L}\sum_{\nu=1}^{P_-}\sum_{ia,jb} f_{ij}(a,b)\xi_i^{-,\nu}(a)\xi_j^{-,\nu}(b) \\
&+ \frac{1}{2}\sum_{\mu=1}^{P_+}\log\left(1-\frac{1}{L}\sum_{i,a} f_i(a)\xi^{+,\mu}(a)^2\right) + \frac{1}{2}\sum_{\nu=1}^{P_-}\log\left(1-\frac{1}{L}\sum_{i,a} f_i(a)\xi^{-,\nu}(a)^2\right)
\end{aligned}
\tag{2.32}
$$

The Eq. 2.31 corresponds to the specific case of PPM Eq. 2.8, with the couplings $J_{ij}(a,b)$ defined as

$$
J_{ij}(a,b) = \frac{1}{L}\sum_{\mu=1}^{P_+}\xi_i^{+,\mu}(a)\xi_j^{+,\mu}(b) - \frac{1}{L}\sum_{\nu=1}^{P_-}\xi_i^{-,\nu}(a)\xi_j^{-,\nu}(b)\ . \tag{2.33}
$$

As shown in the Eq. 2.33, the attractive pattern correspond to the P_+ largest eigenvalues ($\lambda_1^+ \geq \lambda_2^+ \geq \ldots \geq \lambda_{P_+}^+ > 1$) while the repulsive patterns correspond to the smallest P_- eigenvalues ($0 < \lambda_1^- \leq \ldots \leq \lambda_{P_-}^-$).

The likelihood function in Eq. 2.32 can also be expressed as a function of eigenvalues $\{\lambda_\mu\}$,

$$
\mathcal{L}[\boldsymbol{A}^{1:M}] = \sum_{i,a} f_i(a)\log f_i(a) + \frac{1}{2}\sum_{\mu=1}^{P_\pm}(\lambda_\mu^\pm - 1 - \log\lambda_\mu^\pm)\ . \tag{2.34}
$$

Significant findings in [51] are the following: First, the attractive patterns are associate with the largest eigenvectors. On the other hand, the repulsive patterns associate with the smallest eigenvectors. Second, the dominant contributions on the likelihood are the patterns associated with the largest

[6] Technical points to derive the Eq. 2.32 is the estimation of the partition function based on mean-field approximation. The outline of the derivation is shown as follows: First, applying Hubbard-Stratonovich transformation on the partition function. Second, apply the saddle point expansion of the joint distribution and approximate it as a multivariate Gaussian distribution considering the mean and covariance. Last, integrate the Gaussian distribution while imposing a particular gauge choice.

or smallest eigenvectors. Last but not least, the attractive patterns capture the sequence heterogeneity. On the other hand, the repulsive patterns tend to learn residue contact information.

2.4 Technical Points

In this section, we will show some practically necessary treatments for learning the DCA-based statistical models.

2.4.1 Regularization

Let us assume that there is no observation for certain frequencies $f_{ij}(a,b) = 0$ with $f_i(a)f_j(b) \neq 0$. Suppose we impose the frequentist point of view in such a situation. In that case, we need to introduce a negatively infinite conjugate parameter, $J_{ij}(a,b) \to -\infty$ into our statistical model, which causes unstable learning clearly. The standing point of Bayesian statistics, which is the opposite to the frequentist, tends to assume that the model parameter θ of a probability distribution $p(x|\theta)$ is also probabilistic and is governed by a prior distribution $p(\theta|\alpha)$, where α is known as a so-called hyper-parameter.

Here we recall Bayes' theorem,

$$p(\theta|x) \propto p(x|\theta)p(\theta). \tag{2.35}$$

In the context of the above mentioned problem, we can assume that the model parameters $\{J_{ij}(a,b)\}$ and $\{h_i(a)\}$ come from independent Gaussian distribution, $J_{ij}(a,b) \sim \mathcal{N}(0, \lambda_J^{-1})$ and $h_i(a) \sim \mathcal{N}(0, \lambda_h^{-1})$, hence we get another objective function reflecting the philosophy of Bayesian statistics,

$$l^{DCA}(\boldsymbol{h}, \boldsymbol{J}) \leftarrow l^{DCA}(\boldsymbol{h}, \boldsymbol{J}) + \lambda_J \|\boldsymbol{J}\|_2 + \lambda_h \|\boldsymbol{h}\|_2 \ , \tag{2.36}$$

where l^{DCA} is the log-likelihood function defined in Eq. 2.9, and $\|\mathrm{A}\|_2$ is a L2 norm, and λ_J, λ_h are hyper-parameters [7]. The estimator using the Eq. 2.36 is called Maximum a posteriori estimation (MAP). There is a simple criterion for the hyper-parameter of DCA learning that states that $\lambda_J \sim L/M$ and $\lambda_h \sim 1/M$, where M is the number of sequences in the training data [52]. In practice, we use several hyper-parameters for the couplings such as $\lambda_J \in \{10^{-4}, 2 \cdot 10^{-4}, \ldots, 10^{-3}\}$. Typically the value used

[7] $\sqrt{\lambda_J}$ and $\sqrt{\lambda_h}$ are the standard deviation values of the Gaussian distributions $\mathcal{N}(0, \lambda_J^{-1})$ and $\mathcal{N}(0, \lambda_h^{-1})$.

for the hyper-parameter depends strongly on the protein family, and careful tuning is needed. There exist different methods to regularize, such as L1 regularization, which corresponds to assuming the Laplace distribution as a prior distribution of the parameters. One of the advantage to use the L1 regularization is that it can sparsify the couplings [53]. However, the L1 norm makes small parameters to zero even if they are statistically significant, as a consequence, it tends thus to lead to a poor generative model.

Another regularization method we use frequently is *pseudo counts*,

$$\begin{aligned} f_i(a) &\leftarrow (1-\alpha) f_i(a) + \frac{\alpha}{q}, \\ f_{ij}(a,b) &\leftarrow (1-\alpha) f_{ij}(a,b) + \frac{\alpha}{q^2} \ . \end{aligned} \tag{2.37}$$

This method is typically used for mfDCA, and also we used it to investigate sparse Boltzmann machines (cf. Chap. 3). With a similar discussion of the relation between the L2 regularization and the Bayesian modeling, pseudo counts appear if we assume a Dirichlet distribution as a prior distribution of frequency counts. [14] .

2.4.2 Gauge Invariance and Gauge Transformation

In the case of DCA, the number of model parameters and the number of observables are exactly the same, $qL + q^2L(L-1)/2$. However, the observables are not independent: The single site frequency are normalized $\sum_{a=1}^{q} f_i(a) = 1$, and $\sum_{b=1}^{q} f_{ij}(a,b) = f_i(a)$, $\forall i,j \in 1{:}L$ and $\forall a,b \in 1{:}q$. Therefore, there are $L + qL$ redundant parameters. The number of independent conditions is $(q-1)L + (q-1)^2L(L-1)/2$. In order to avoid an over-parameterization, we can reduce the number of parameters keeping the energy function invariant. This fact is known as a *gauge invariance* in physics, and to reduce redundant model parameters is known as a *gauge choice* [42, 37]. Eq. 2.8 is invariant under the following transformations,

$$\begin{aligned} J_{ij}(a,b) &\leftarrow J_{ij}(a,b) + \mathcal{J}_{ij}(a) + \mathcal{K}_{ij}(b) + c_{ij} \\ h_i(a) &\leftarrow h_i(a) - \sum_{j(>i)} \mathcal{J}_{ij}(a) - \sum_{j(<i)} \mathcal{K}_{ji}(a) - H_i \ , \end{aligned} \tag{2.38}$$

where $\mathcal{J}_{ij}(a)$ and $\mathcal{K}_{ij}(b)$ are arbitrary functions.

In DCA, we typically use *lattice-gas gauge*:

$$J_{ij}(a,q) = J_{ij}(q,b) = h_i(q) = 0 \ , \tag{2.39}$$

for all $i, j \in 1{:}L$ and $a, b \in 1{:}(q-1)$ [42].

Another frequently used gauge choice is *zero-sum gauge* or *Ising gauge*:

$$\sum_{a=1}^{q} J_{ij}(a,b) = \sum_{b=1}^{q} J_{ij}(a,b) = \sum_{a=1}^{q} h_i(a) = 0, \tag{2.40}$$

for all i, j, a, b. This gauge can be achieved using the transformations,

$$\begin{aligned} J_{ij} &\leftarrow J_{ij}(a,b) - J_{ij}(a,\cdot) - J_{ij}(\cdot,b) + J_{ij}(\cdot,\cdot) \ , \\ h_i(a) &\leftarrow h_i(a) - h_i(\cdot) + \sum_{j(\neq i)} \left(J_{ij}(a,\cdot) - J_{ij}(\cdot,\cdot)\right) \ , \end{aligned} \tag{2.41}$$

where we denote $g(\cdot)$ as the average $q^{-1}\sum_{a=1}^{q} g(a)$. It is easy to check that the zero-sum gauge minimizes the Frobenius norm of couplings, a fact exploited in DCA-based residue contact prediction (cf. Sec. 2.5.1). Here is a sketch of proof: Substitute the general form of the gauge transformation into the definition of the Frobenius norm, then take the functional derivative with respect to $\{\mathcal{J}_{ij}(a), \mathcal{K}_{ji}(a), H_i\}$, note that the function of Frobenius is a convex function of these gauge variables. Therefore the set of variables that gives zero gradients can minimize the Frobenius. The zero-sum gauge satisfies this condition.

2.4.3 Phylogenetic Correction

Most inference methods assume that samples in the training data are independently and identically distributed (i.i.d.). However, this assumption is not true in general. In particular, ensembles of protein sequences or MSAs typically strongly depend on each other. Therefore they are not independent at least. This strong correlation among samples comes from the fact that sequences are evolutionarily related to each other, and this relation is known as the phylogenetic tree in evolutionary biology. Another reason is the selection bias of sequencing: some specific species such as *E. coli* are more frequently sequenced due to medical or academic interests.

Sequence reweighting – We use a simple reweighting method to reduce such biases [42]. First, we introduce a similarity threshold $0 < x < 1$: If the number of identical residues between two sequences is greater than xL, these sequences are assumed to be carrying almost the same information. On the

contrary, sequences that have smaller sequence similarities are considered to carry substantial information. Formally, the reweighting parameter for a sequence ensemble among the $1{:}M$ sequences can be written as

$$w^m = \Big(\sum_{n=1}^{M} \Theta(d_{mn} < 1 - x) \Big)^{-1} \ , \tag{2.42}$$

where d_{mn} is the normalized Hamming distance between $\boldsymbol{A}^m$ and $\boldsymbol{A}^n$, and $\Theta(c)$ is the Heaviside function that gives 1 if and only if the condition c is true, and 0 otherwise. Therefore, the more similar a sequence is to other sequences, the smaller its weight will be. Normally, we use $x = 0.8$, and the result of contact prediction is robust between $0.7 \leq x \leq 0.9$ [42].

The single and pairwise frequencies should be modified accordingly,

$$\begin{aligned} f_i(a) &= \frac{1}{M_{eff}} \sum_{m=1}^{M} w^m \delta_{A_i^m, a} \\ f_{ij}(a,b) &= \frac{1}{M_{eff}} \sum_{m=1}^{M} w^m \delta_{A_i^m, a} \delta_{A_j^m, b} \ , \end{aligned} \tag{2.43}$$

where $M_{eff} = \sum_{m=1}^{M} w^m$, denotes the effective number of sequences.

Averaged Product Correction – The APC is a method to remedy the bias due to background noise and phylogenetic effect on the Frobenius norms of couplings. Initially, this method was applied to MI [33], then it was applied to DCA-based contact predictions.

Here, we show a derivation of APC based on Burger *et al.* [41]. The APC is given by

$$F_{ij}^{APC} = F_{ij} - \frac{F_{i*} F_{*j}}{F_{**}} \ , \tag{2.44}$$

where $F_{i*} = \frac{1}{(L-1)} \sum_{j(\neq i)}^{L} F_{ij}$ and $F_{**} = \frac{2}{L(L-1)} \sum_{i<j}^{L} F_{ij}$. The above Eq. 2.44 can be derived by assuming that the noise B_{ij} that is in the observable pairwise interaction, factorizes into $B_i B_j$. Therefore the pairwise signal can be written in this way,

$$F_{ij} = F_{ij}^{true} + B_i B_j \ ,$$

where F_{ij}^{true} is Frobenius norm removing the background noise. By summing over indices, we get following relations

$$
\begin{aligned}
F_{i*} &= F_{i*}^{true} + B_i B_* \\
F_{**} &= F_{**}^{true} + B_*^2 \ .
\end{aligned}
\tag{2.45}
$$

Assuming that the true pairwise signal is smaller than noise, we find that $F_{i*} \sim B_i B_*$ and $F_{**} \sim B_*^2$. We can thus remove the noise from the true pairwise signal and get,

$$
F_{ij}^{true} = F_{ij} - \frac{F_{i*} F_{*j}}{F_{**}} \ . \tag{2.46}
$$

The above argument explains why APC works. There is another interesting method based on entropy correction [54], which leads to very similar results.

2.4.4 Learning by Contrastive Divergence and Persistent Contrastive Divergence learning

Contrastive Divergence – Contrastive divergence (CD) is a point estimation method that can be applied to a broad class of inference problems, including exponential families (bmDCA is in this class), especially efficient for learning restricted Boltzmann machines (RBM). Initially, the CD method was proposed to reduce the vast computational time of learning statistical models. The initial attempt was based on the products of experts (PoE) model by Geoffrey E. Hinton [55], where the PoE is a latent variable model closely related to the mixture model.

The key idea of the CD method is to replace the sample average in the gradient of the log-likelihood with an average overs CD samples, which is computationally easily attainable. Here, the CD sample is an instance after k steps of MCMC initialized in the training data. Therefore, the size of the CD ensemble is exactly one of the training data. Typically, k is a small number, and even $k = 1$ is quite frequently chosen. Note that if k is sufficiently large, the CD method is converged to the conventional MCMC based learning.

An intuition of the CD learning success is as follows: The empirical distribution and the model distribution might be close. Hence, a few MC steps from the data points are assumed to be reasonable approximations of the equilibrium distribution of the model probability (in other words, start from one of the training data means starting from the almost equilibrated state

in MCMC). Note that the principle of CD learning is different from ML learning. The policy of CD learning corresponds to the minimization of the transition probability flows escaping from the training data points (a similar idea can be found in Minimum Probability Flow learning [56, 57]).

PCD has an algorithmically similar idea to CD learning. However, the objective function that becomes effectively more similar to ML learning is persistent contrastive divergence (PCD) learning. Empirically, PCD learning gives more stable and accurate results than CD learning in our study. The first epoch is precisely the same as the CD learning. Each sample is initialized by training data and update a few times. The difference will be more noticeable after the second epoch: The last states at the previous epoch are used as the initial states of the next epoch. Such as initialize a sample state at the 2nd (3rd) epoch by an updated state at the 1st (2nd) epoch.

More detailed descriptions about CD-based learning are discussed in the Appendix E.

Batch learning – Some MSAs contain many sequences (about 10^5 sequences for PF00072), which cause a very long learning time for CD-based learning. In this case, we use mini-batch learning, a powerful learning method to reduce computational time to estimate gradients of the objective function: we divide the MSA (M sequences) to sub-ensembles that contain $B(< M)$ sequences used for CD learning. Using smaller B will make the calculations faster, but increases the statistical error of the estimated gradients due to the reduction of sample size. Hence there is a tradeoff between accuracy and computational time. It is also closely related to the optimal learning rate. If B is large, the gradient can be calculated accurately, so a large learning rate may also be used. On the contrary, for small B, one needs to use small learning rates.

2.4.5 Other learning techniques

Hyper-parameter optimization – Even simple DCA techniques rely heavily on hyperparameters such as learning rates and the strength of regularization (typically, $\eta_J = 0.001 - 0.1$ and $\lambda_J = 0.001 - 0.01$, respectively). Empirically, generative models require particularly careful fine-tuning. What we normally use is grid search, a simple but exhaustive hyperparameter searching algorithm. For example, find the best possible com-

bination among $\eta_J \in \{0.001, \ldots, 0.1\}, \lambda_J \in \{0.001, \ldots, 0.01\}$ in the above example.

Assessment of learning – It is crucial to evaluate the learning process properly, to avoid excessively long learning times. For this aim, one can use the Pearson correlation of two-point connected correlations, or the maximum difference of correlations between the training data and data sampled from the model. The evaluation of CD-based learning is particularly important, and the convergence test must be made in consideration of the statistical fluctuations involved in the gradient (see also Appendix E.3).

2.5 Applications

2.5.1 Predicting residue-residue contacts and protein

Initially, DCA was proposed to predict residue contacts in proteins using only sequence information [42, 58, 49, 59, 60].

Here, we typically use 8Å as a condition for two residues to be in "contact". If a relative distance between residues i and j is equal or less (greater) than this threshold, $\leq$ 8Å($>$ 8Å), we regard the pair as "contact" ("non-contact"). We represent it by a contact matrix, a binary matrix $D \in \{0,1\}^{L\times L}$, results $D_{ij} = D_{ji} = 1$ ("contact"), otherwise 0 ("non-contact").

Accurately predicting such a contact maps is extremely helpful for 3D-protein structure prediction, and it has been reported that the minimum number of necessary true contact predictions to reconstruct protein structure correctly is $8\% - 25\%$ [61, 62] in a contact map. Therefore, structure prediction methods aim to minimize the difference between the native contact map and a computationally predicted contact map. Due to the availability of many experimentally determined protein structures, "true" contact maps are known for many proteins and can be used for testing prediction methods.

For DCA-based contact prediction, coupling matrices $J_{ij} \in \mathbb{R}^{q\times q}$ are converted to a Frobenius norm $F_{ij} \in \mathbb{R}$, $\forall i, j \in 1{:}L$,

$$F_{ij} = \sqrt{\sum_{a,b}^{q} (J_{ij}(a,b))^2} \ . \tag{2.47}$$

As mentioned in Sec. 2.4.2, the zero-sum gauge is typically used for computing the Frobenius norm. It provides a real-scalar-valued score to a residue pair and can be used for contact prediction. In general, the APC score, the matrix after applying the APC correction in Eq. 2.44 to the Frobenius norm, is used.

To quantitative assess the accuracy of residue contact prediction, we commonly use the positive predictive value (PPV), which is defined as the number of true positives divided by the number of the total predictions [8]. To consider only non-trivial contacts, we exclude the effects of backbone interactions due to peptide bindings such as $|i-j| < 5$. Therefore, the number of total prediction is $n^* = L(L-5)/2)$.

In practice, we sort the (APC) score in descending order and represent the PPV as a function of the sorting rank, n. We represent the sorting of residue pairs as a function of the sorting rank as following,

$$\sigma(n) \in \{(i,j) \mid \forall i \in 1{:}(L-5),\ \forall j \in 6{:}L,\ i < j\} \ ,$$

with,

$$S_{\sigma(1)} \geq S_{\sigma(2)}, \geq, \ldots, \geq, S_{\sigma(n^*)} \ ,$$

where $S_{\sigma(n)}$ is a score ranked n-th. Note that the considering residue pairs are those satisfying $|i-j| \geq 5$. Thereby, the PPV curves can be written as a function of the rank $n \in 1{:}n^*$,

$$PPV(n) = \sum_{m=1}^{n} D_{\sigma(m)}/n \ , \tag{2.48}$$

The rationale for this constraint is that the alpha helix turns back to the same position after four consecutive amino acids. The result of the contact prediction was shown in Fig. 2.2.

2.5.2 Sequence scoring

Sequence scoring – This problem is one of the most classical bioinformatics questions. As we saw in section 1.2.3, by marginalizing the hidden state of a profile HMM, we can estimate the likelihood function and evalu-

[8] The sum of the number of true positives and the number of false positives corresponds to the total predicted number.

ate the probability of any sequence. It allows us to estimate the probability that a sequence belongs to a particular sequence group (such as a homology group). Similarly, DCA-based statistical models provide a probability. Therefore it must be possible to use it as a sequence scoring method. In particular, the DCA-based method can consider the pairwise couplings. Hence the scoring system with DCA can regard coevolution effects or epistasis effects. Technically, a log probability of the pairwise distribution, which is essentially the negative of generalized Potts energy function, is employed as the sequence score.

Indeed, it was found that scoring systems considering pairwise interactions are inherently important for decoding amino-acid sequence information. In [63], an experiment was conducted to verify the role of pairwise interactions for functional artificial sequences of WW-domains (Pfam ID PF00397). They revealed that the pairwise interactions between amino acids are essential for proteins to be folded into the native structures. More specifically, they generated artificial amino-acid sequences from models considering different statistical features such as one-point and two-point frequencies of an MSA of natural sequences. They examined whether the artificial sequences are able to fold into the native structure.

The models to consider are:

- Natural (N): Natural WW sequences, selected randomly from the MSA (we hereby name the natural sequences as Nat sequences).
- Random (R): Random sequences, where each amino-acid at each site is randomly drawn from a random distribution that keeps only the mean frequency of the MSA, $A_i \sim f(a)$.
- Independent conservation (IC): Site-independent conservation sequences, each amino acids at each site are drawn from the single-site distribution of the MSA, $A_i \sim f_i(a)$. Therefore, there is no statistical coupling between residues.
- Coupled conservation (CC): Sequences that keep both single-site and pairwise frequency, $(A_i, A_j) \sim f_{ij}(a, b)$. Sequences are generated by swapping an amino-acids in the same MSA while keeping pairwise frequencies. Specifically, this calculation was performed using MC sampling and a simulated annealing (SA) algorithm.

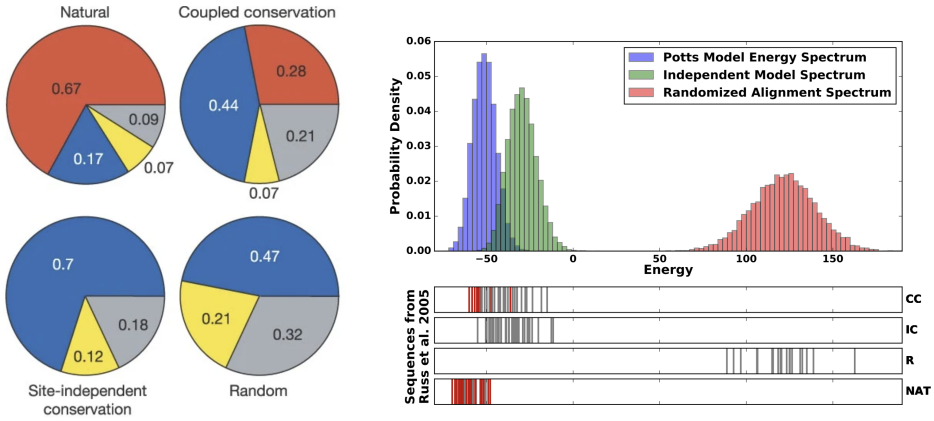

Figure 2.4: **Left**: Pie charts showing the results of folding experiments for N ($n = 42$), CC ($n = 43$), IC ($n = 43$), or random ($n = 19$) WW domain sequences. The experiments shown that none of the sequences folds into the native structure for both R sequences and IC sequences. On the contrary, 28% of the CC sequence realized the native structure. Considering that a 67% of the N sequences could be natively folded in the experimental conditions, the result of the CC sequences is astonishing. **Right**: The top panel shows the Potts-energy distribution of generated sequences for the WW domain (parameters were obtained by the ACE algorithm [64]). The histograms correspond to MC samples of the pairwise model (blue), which should be the best-fit distribution to the natural WW sequences of the single-site model (profile model) (green), and random model (red). (Source [65, 66])

Note that the CC sequences are equivalent to a sample from a PPM that reproduce the two-point frequencies observed in the MSA [63] [9].

The upper rights panel in Fig. 2.4 right shows energy distributions based on a PPM using the Adaptive Cluster Expansion (ACE) [64], a precise approximation of the pairwise-Potts model. It should be noted here that DCA relies on pairwise information, so it can provide an indicator that distinguishes between random, independent-site, and sequences considering

[9]The MC sampling with the same construction takes a very long convergence time. As the number of sequences and the sequence length increase, it becomes more difficult to converge the equilibrium distribution due to the funnel-like landscape of free energy. Therefore, it is better to treat pairwise statistics using a statistical model for technical reasons as well.

the pairwise interactions.

Interestingly, whether the sequences can be folded or not is strongly correlated with the DCA energy. The bottom in Fig. 2.4 shows the energy values for each N, R, IC, and CC sequences used in the experiment [65], the red and gray bands are assigned to the foldable and non-foldable sequences. The energy values for CC, IC, and R correspond to the energy histogram (the top panel) Potts energy, site-independent energy, and randomized energy, respectively (note, these sequences are generated differently as in [65]).

The remarkable thing is that sequences with lower energies are more likely to hold the native structure, with a significant increase in the percentage of sequences that fold below a certain energy threshold. It suggests that the pairwise-Potts energies can reasonably distinguish between foldable- and non-foldable- sequences. This experiment also implies the possibility of the PPM to generate sequences with high-folding probability, meaning that the model can be used not only for scoring but also for generating functional protein sequences.

2.5.3 Protein sequence design

A direct experiment to learn artificial functional protein sequences based on DCA has been conducted recently [67]. This experiment successfully designs artificial protein sequences of chorismate mutases (CMs), a standard model protein, to investigate principles of catalysis and enzyme design using bmDCA model. CM acts as a catalyst and may accelerate the rate of a reaction required for cell growth by more than 1 million times. It is essential for bacterial growth in a minimal glucose of medium. In order to monitor CMs activity in an *in vivo* complementation assay, this experiment employed CM-deficient *E. coli* cells, and adapted selective conditions (lacking Phe and Tyr, which are products of the reactions requiring CM catalysis) in the minimal medium. It is known that *E. coli* strains lacking CM are auxotrophic to Tyr and Phe, and both the degree of supplementation of these amino acids and the expression level of the reintroduced CM gene quantitatively determines the growth rate. In other words, the more the CM genes are expressed, the easier it is to grow even in a medium with low Tyr and Phe concentration. Hereafter, we denote *E. coli* that express CM genes as EcCM.

Around 1130 protein sequences were included in the MSA that was used for learning a bmDCA model that can precisely reproduce one-point frequen-

cies and two-point connected correlations in the MSA. Then more than 1900 sequences are generated under the several *temperatures* $T \in \{0.33, 0.66, 1.0\}$, where the temperature is a parameter rescaling model parameters globally and linearly. Therefore MC sampling is done using the following probability distribution,

$$p(\boldsymbol{A}) \sim \exp\left[1/T\Big(\sum_i h_i(A_i) + \sum_{i<j} J_{ij}(A_i, A_j)\Big)\right] .$$

To understand, which natural and artificial sequences are functional the authors introduce the relative enrichment (*r.e.*) defined as the difference of log ratio of frequencies of enrichment in the population before and after selection. A schematic explanation of the pipeline of the experiments is also shown in Fig. 2.5.a.

$$r.e. = \log \frac{f_{sel}(\boldsymbol{A})}{f_{inp}(\boldsymbol{A})} - \log \frac{f_{sel}(\boldsymbol{A}^{ref})}{f_{inp}(\boldsymbol{A}^{ref})} , \tag{2.49}$$

where $\boldsymbol{A}^{ref}$ is the sequence for EcCM, and the subscripts sel, inp are the set of selected and input samples, respectively. If the relative population size of the selected sequences is not changed, the *r.e.* becomes around zeros. On the contrary, the *r.e.* gives a positive value if the population of the selected sequence increases significantly. Fig. 2.5.c shows that the DCA energy (which is denoted as statistical energy in the original paper) is strongly correlated with the functionality of the *r.e.*. Sequences should have a low energy to realize functional sequences. There is a threshold in DCA energy above which only non-functional sequences exist.

As the results of the experiment, Fig. 2.6 shows that the sequences corresponding with low-energy drawn from $T \in \{0.33, 0.66\}$ typically recapitulate the qualities of natural sequences. On the contrary, sequences from $T = 1$ poorly performed to recapitulate natural sequences. Overall, among the 1618 total artificial sequences, 481 sequences ($\sim 30\%$, norm *r.e.* > 0.42) rescued growth in this experiment, although the top-hit sequences identities to any natural CM was between 42 and 92%. Even 46 sequences are functional with $< 65\%$ sequence identity to the natural sequences.

2.5.4 Fitness landscape

A fitness landscape can provide information about the genotype–phenotype evolutionary relation, including thermodynamic stability as well as the dy-

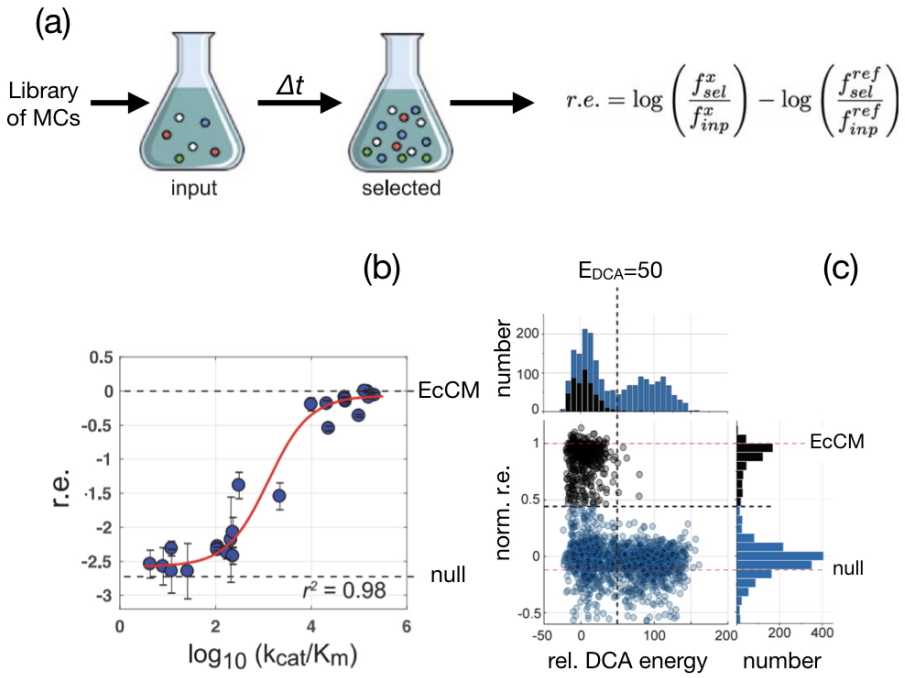

Figure 2.5: (a) Schematic figure of the workflow to characterize CM activities. (b) Relationship between $r.e.$ and $\log k_{\text{cat}}/K_m$. It shows as reaction proceed *E. coli* strains with defective CM decrease in the minimal medium relative to the wild type reference. (c) Relation between the DCA energy and $r.e.$ indicates that no functional sequence can exist above a certain DCA energy value. These figures were adapted from [67]

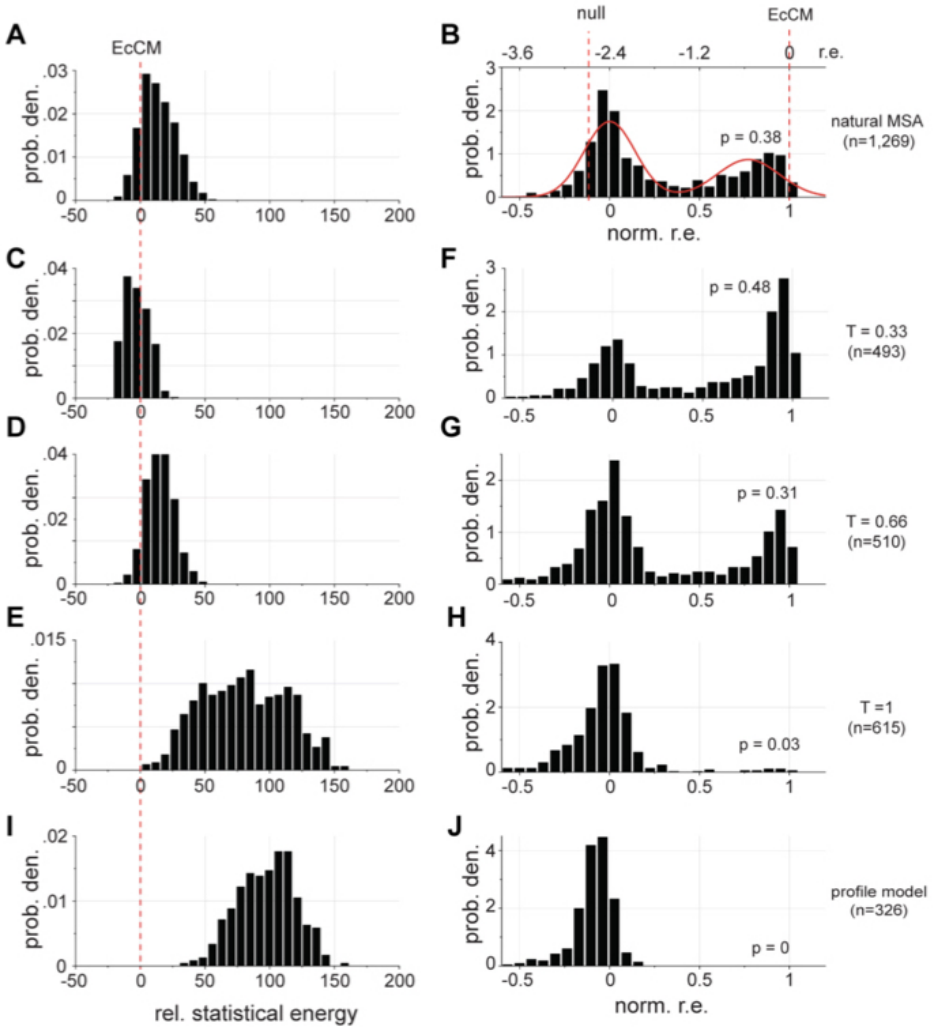

Figure 2.6: **The right** column shows frequencies of the statistical energy (measured relative to EcCM) for natural sequences, sequences from MC sampling ($T \in \{1.0, 0.33, 0.66\}$), and sequences from the profile model. Sequences with the negative statistical (DCA) energy are observed in the natural and $T = 0.66$ and $T = 0.33$ sequences. However, none of the sequences has a negative energy for $T = 1$ or profile sequences. **The left** column shows frequencies of the *r.e.* for these ensembles of sequences. It shows that the natural sequences and bmDCA sequences with $T = 0.66, 0.33$ are successfully adapted in the selective conditions since the normalized *r.e.* is shifted toward positive values and show bimodal distribution (the left peak in the histogram corresponds with null E. coli, and the right peak corresponds with EcCM). Notably, sequences from bmDCA with $T = 0.33$ give higher enrichment of EcCM compare to the natural sequences. On the other hand, it also shows that both bmDCA with $T = 1.0$ and the profile model generate rarely produced EcCM sequences. The figure was taken from [67].

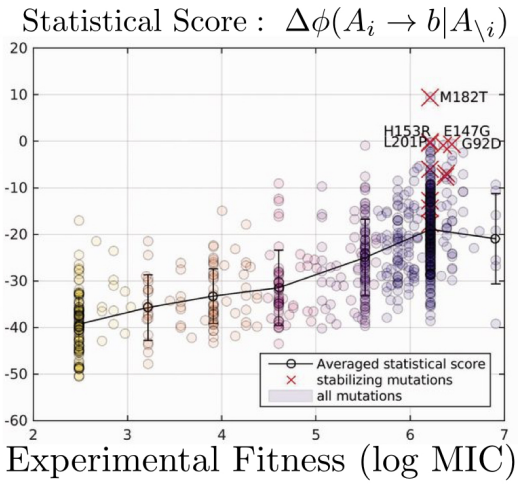

Figure 2.7: Comparison between the statistical scores ($\Delta\phi = \Delta E$) and the experimental fitness, the minimum inhibitory concentration (MIC) of the antibiotic [69]. The five highest-scoring mutations shown in the figure (M182T, H153R, I247V, T265L/Q, and N276D) are reported as stabilizing mutants (red crosses). (Source [70])

namic process of protein folding [3]. Prediction of the fitness landscape has attracted great attention in engineering and biomedical applications due to the possibility to design proteins that have acquired properties different from those of natural species while maintaining the desired function (e.g., enzymatic function, increased stability, etc).

Particularly prediction of mutation (e.g., stability, types of potentially prospering mutant), which can be assessed by comparing the fitness values between a wild type and its variants (mutant), has been accelerated for the last five years thanks to the accurate high-throughput mutation data or deep mutational scanning [68].

The DCA energy function takes into account the effects of conservation and co-evolution and can provide a protein sequence that folds into a naturally occurring structure with high probability, as we saw previously Sec. 2.5.3. From this consideration, the energy function of DCA is used as a sequence fitness landscape function, and the mutational landscape function

can be predicted in terms of the DCA energy function,

$$\Delta E(\boldsymbol{A}^{wt} \to \boldsymbol{A}^{mut}) = \mathcal{H}(\boldsymbol{A}^{mut}) - \mathcal{H}(\boldsymbol{A}^{wt}) \ , \tag{2.50}$$

where, $\boldsymbol{A}^{wt}$ and $\boldsymbol{A}^{mut}$ are sequences of wild type and mutant, respectively.

Fig. 2.7 shows one of the remarkable results in [70] using the large-scale mutagenesis data of beta-lactamase (TEM1), a model enzyme that provides resistance against beta-lactam antibiotics. In this case, mutations considered were only single-site mutations. For the quantitative comparison to the computational prediction, the antibiotic's minimum inhibitory concentration (MIC) was used [69].

This DCA-based fitness-landscape prediction has been applied, and its effectiveness has been recognized in many cases such as protein design and stability prediction [71], identification of drug resistance mutation in HIV[72], drug resistance-associated mutation in HIV-1 protease [73], virus escaping time and mutation-location prediction[74], antiviral drugs design [75], and vaccine design[76, 77].

2.5.5 Other applications

Protein-protein interaction networks – Proteins rarely function alone *in vivo* and work with other proteins to realize functions such as metabolisms, transport, making components, etc. Such relationships between proteins are known as protein-protein interaction (PPI) networks, consisting of proteins as nodes and interactions between proteins as edges. Understanding PPI networks is a fundamental question in systems biology.

PPI networks are also useful and give insight for understanding mechanisms of complex systems, such as identifying genes and proteins that are associated with disease using an assumption that neighbors of "disease genes" in PPI networks typically relate with similar diseases [78, 79, 80, 81], to understand metabolism [82], and to find proteins that can be influenced by certain chemical compounds [83].

Ref. [84] shows that DCA can be also used for PPI predictions and successfully infers underlying PPI networks. Training data is provided as N MSAs for each protein family. An MSA contains M_p sequences of length L_p in each alignment D_p for all of considering families p. The task of PPI

network prediction can be decomposed into two steps:

1. Matching procedures (concatenate sequences): For applying DCA-based interaction networks prediction, it is necessary to concatenate sequences in two putative co-evolutionary related families (p, p') in MSAs $(D_p, D_{p'})$. DCA learning is done on a new alignment $D_{(p,p')}$ of sequence length $L_p + L_{p'}$. Ref. [84] employed a matching algorithm based on the genomic distance between sequences.

2. Inference procedures (execute DCA learning): The DCA learning using two concatenated sequences $(\boldsymbol{A}, \boldsymbol{A}')$ can be formulated as the following joint distribution,

$$\begin{aligned} p(\boldsymbol{A}, \boldsymbol{A}') &= \frac{1}{Z} \exp\Big(- \mathcal{H}(\boldsymbol{A}) - \mathcal{H}'(\boldsymbol{A}') - \mathcal{H}^{int}(\boldsymbol{A}, \boldsymbol{A}')\Big) \\ \mathcal{H}^{int}(\boldsymbol{A}, \boldsymbol{A}') &= -\sum_{i=1}^{L_p} \sum_{j=1+L_p}^{L_{p'}} J_{ij}(A_i, A'_j) \ , \end{aligned} \tag{2.51}$$

   where $\mathcal{H}(\boldsymbol{A})$ and $\mathcal{H}'(\boldsymbol{A}')$ are the standard DCA energy function in Eq. 2.8. The function $\mathcal{H}^{int}(\boldsymbol{A}, \boldsymbol{A}')$ is the function that regards co-evolutionary couplings between the two families (p, p'). Ref. [84] used plmDCA method.

The DCA-based PPI prediction achieved 70% for small ribosomal subunit (SRU) and 90% for large ribosomal subunit (LRU) of the true positive rate among the top 10 predictions.

The authors discussed that these results can improve as the number of sequences available increases, which is likely to happen thanks to advances in next-generation sequencing technology.

Sequence pattern selection – Sequence motifs or position-specific scoring matrices (PSSM) are commonly used as representations of biological sequences to characterize ensembles of sequences such as functionality or phylogenetically related protein domains. Finding patterns of sequences to distinguish ensembles of data is a fundamental problem in bioinformatics.

It was found that the mean-field Hopfield-Potts model can construct patterns that are closely related to PCA vectors of the data covariance matrix in [51]. Although it is not a DCA-based method, there is another important method to find sequence motifs, statistical coupling analysis (SCA). SCA can

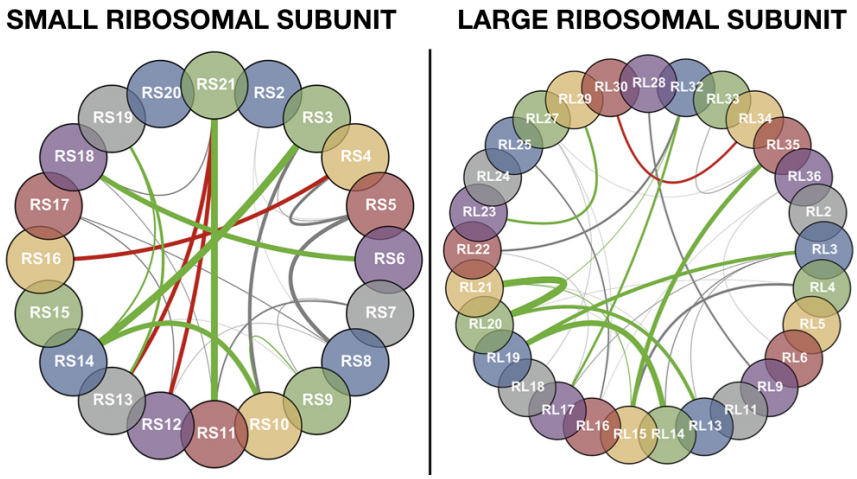

Figure 2.8: The left and right figures show the inferred interaction networks for the small ribosomal subunit (SRU) and large ribosomal subunit (LRU). The edge colors are assigned as follows: Green for true prediction. Red for false prediction (there is no such interaction in the experimental PPI networks). Gray for the interaction that is determined by experiment but not found in the prediction. (Source [84])

find other types of sequence motifs, which can distinguish groups of amino-acids that variate related to each other and can detect collectively evolving groups of positions, which are referred to as "protein sectors" [85, 86, 87]. A similar method can be found in [88]. It is based on a hierarchy of variables and aims to select statistically relevant residual variables.

Coupling parameters reveal biologically meaningful information – Indeed, today's best contact prediction methods rely heavily on deep neural networks, enabling more accurate three-dimensional structure prediction than other methods. The deep neural network (DNN) based methods may obtain impressive results, but a clear drawback of DNN is that it is hard to interpret millions of parameters and understand learning quantitatively. Concerning the interpretability problem of DNN, DCA-based algorithms provide clear relations among the statistical model, estimated parameters, and biophysical meaning while keeping almost the same performance with DNN.

Although the strong coupling parameter $J_{ij}(a, b)$ can be interpreted as biophysical interaction, strong couplings are extremely rare, and most of the remaining couplings are small, but some of them correspond with structural interaction [89]. It is desired to propose a method that allows the selection of statistically important but small couplings.

In the following section, we will report some statistical generative models of protein families. These models are assumed certain interaction architectures between residues to realize more biologically interpretable models, as shown in Fig. 2.9.

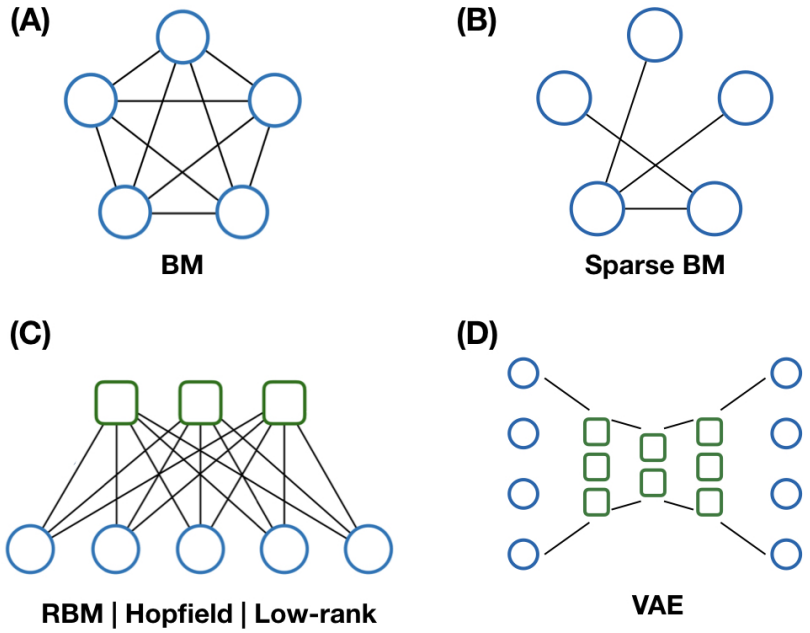

Figure 2.9: (A). Architecture Boltzmann machine (BM), edges, and nodes are denoted as coupling interactions $\{J_{ij}\}$ and amino-acid variables $\{A_i\}$. (B). Sparse BM, the architecture is represented by the same convention in Fig. A, redundant couplings interactions are diluted (see Chap. 3) (C). Restricted BM (RBM) or low-rank model, the squares in the green squares are represented as "hidden variables", which effectively induce interaction among amino-acid variables (see Chap. 4). (D). Variational Autoencoder (VAE), it is a generative model constructed by a multilayer neural network (see Chap. 6).

II

Variable selection for protein sequence modeling

> Everything should be made as simple as possible, but no simpler.
>
> —Albert Einstein (1879 - 1955)

Chapter 3

Specification of Sequences Statistics

3.1 Motivation

As shown in the previous chapter, DCA methods shed light on many biologically intriguing questions: realize probability distributions that can reproduce properties of biological sequence ensembles, predict residue-residue interactions, give an objective measure to score each sequence, reveal protein-protein interaction networks, and generate functional artificial protein sequences.

DCA-based methods have succeeded in a wide range of studies. However, the model depends on a considerable number of parameters, typically 10^2 to 10^3 times greater than the number of sequences available for learning. Clearly, the model would be sensitive to statistical noise in the training data. Moreover, most of the estimated coupling parameters are very noisy. Especially, those with small coupling parameters are particularly dependent on their initial conditions (Ref. [90] in section 6.1), which suggests that some of these small parameters are irrelevant. Though, other small couplings might come from properties of the protein family, such as physical contacts or the underlying phylogenies. Regarding the discrimination of relevant and irrelevant couplings parameters, we would like to address the following questions:

1. What is the minimum set of constraints required to reproduce the statistics of protein sequence ensembles within a generative model?

Can the minimally constrained models improve biological interpretability?

2. Is the minimally constrained model more robust than the standard fully connected pairwise Potts model (PPM)?

3. Can the minimally constrained model improve predictions of residues contacts and the effects of mutations that might be affected by statistical noise?

In order to address these questions, we propose a generative model that takes only into account the minimal set of constraints while minimizing the Kullback-Leibler divergence from the fully connected PPM. This method decimates redundant model parameters based on information-theoretic measures. This method can be applied to any problem that standard PPMs can handle.

Our work revealed that the remaining parameters after the decimations contain only statistically relevant parameters associate with 3D structures and/or phylogenies. As explained in the article (see the next section), the decimation protocol is different from both the color compression method in [52] and the pseudo-likelihood-based small coupling decimation in [91].

Our contributions to this study are:

1. We found that we can decimate (remove) around 90% coupling parameters $J_{ij}(a, b)$ without decreasing the accuracy of residues contact predictions and reproducibilities of statistics (two-point and three-point connected correlations).

2. Our methods achieve lower densities of finite couplings than the other regularization methods (e.g., L1 regularization, empirical frequencies-based decimation) without degrading the ability to reproduce statistics of the training data.

3. We found that by reducing the density of the coupling parameters, the model became more robust against small perturbations on the model parameters.

4. We also proposed a method for quantitatively estimating the difference between the contact and non-contact distribution. Using these measures, we found that the difference in contact and non-contact became more pronounced as the coupling densities decreased.

3.2 Article

Sparse generative modeling via parameter-reduction of Boltzmann machines: application to protein-sequence families

Pierre Barrat-Charlaix,[1] Anna Paola Muntoni,[2,3,4,5] Kai Shimagaki,[4] Martin Weigt,[4] and Francesco Zamponi[5]

[1] *Biozentrum, Universität Basel, Switzerland - Swiss Institute of Bioinformatics, Basel, Switzerland*
[2] *Department of Applied Science and Technology (DISAT), Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy*
[3] *Italian Institute for Genomic Medicine, IRCCS Candiolo, SP-142, I-10060 Candiolo (TO), Italy*
[4] *Sorbonne Université, CNRS, Institut de Biologie Paris Seine, Biologie Computationnelle et Quantitative – LCQB, 75005 Paris, France*
[5] *Laboratoire de Physique de l'Ecole Normale Supérieure, ENS, Université PSL, CNRS, Sorbonne Université, Université de Paris, F-75005 Paris, France*

Boltzmann machines (BM) are widely used as generative models. For example, pairwise Potts models (PM), which are instances of the BM class, provide accurate statistical models of families of evolutionarily related protein sequences. Their parameters are the local fields, which describe site-specific patterns of amino-acid conservation, and the two-site couplings, which mirror the coevolution between pairs of sites. This coevolution reflects structural and functional constraints acting on protein sequences during evolution. The most conservative choice to describe the coevolution signal is to include all possible two-site couplings into the PM. This choice, typical of what is known as Direct Coupling Analysis, has been successful for predicting residue contacts in the three-dimensional structure, mutational effects, and in generating new functional sequences. However, the resulting PM suffers from important over-fitting effects: many couplings are small, noisy and hardly interpretable; the PM is close to a critical point, meaning that it is highly sensitive to small parameter perturbations. In this work, we introduce a general parameter-reduction procedure for BMs, via a controlled iterative decimation of the less statistically significant couplings, identified by an information-based criterion that selects either weak or statistically unsupported couplings. For several protein families, our procedure allows one to remove more than 90% of the PM couplings, while preserving the predictive and generative properties of the original dense PM, and the resulting model is far away from criticality, hence more robust to noise.

I. INTRODUCTION

Many applications of generative modeling, especially in biological systems, are confronted to a limited amount of available data, from which a large number of parameters have to be inferred [1]. A particularly interesting example is that of proteins, which belong to the most interesting complex systems in nature and are essential in almost all biological processes. Most of them robustly fold into well-defined three-dimensional structures, which in turn form the basis of their functionality. This triangular sequence-structure-function relationship has, over several decades now, attracted substantial attention in biological physics [2, 3].

A fascinating approach to the generative modeling of biological sequences has emerged over the last years [4, 5]. In the course of evolution, biological sequences accumulate mutations and become more diverse. We can now easily observe the sequence variability across large families of so-called homologous proteins, *i.e.* proteins of common evolutionary ancestry and of close to equivalent function but in different species or biological pathways [6]. Such homologous proteins may differ by 70-80% of their amino acids without substantial changes in structure and function. However, their sequence variability is not fully random: a vast majority of mutations is deleterious, reducing protein stability or functionality. They are thus suppressed by natural selection. Only protein variants of similar or even better functionality are maintained. In this way, the protein's structure and function constrain the viable sequence space that can be explored by evolution. Inverting this argument, the empirically observed variability of homologous sequences contains information about such evolutionary constraints, albeit frequently well hidden. This idea is at the basis of the concept of data-driven "sequence landscapes", *i.e.* classes of models that describe the statistical properties of protein families, assigning high probabilities to functional amino-acid sequences and low probabilities to non-functional ones [5, 7]. The log-probability (or minus "energy") is thus interpreted as a measure of sequence fitness, hence the name of sequence (fitness) landscape [8]. Among the best known such models are Potts models (PM), parameterized by local fields and two-site interaction couplings (*cf.* below for details), and constructed via the Direct Couplings Analysis (DCA) method, which is now firmly established [5, 7]. The DCA parameters can be obtained via inference or learning procedures [9–12], and they can be used to extract useful information on molecular structure [13–16] and function [17, 18], on the effects of mutations [19, 20], and to generate new artificially-designed molecules with specific properties [21, 22].

A concrete implementation of DCA is the following [12]. Given training data in the form of a Multiple Sequence Alignment (MSA) of M homologous sequences of aligned length L, the PM parameters are learned by the

so-called Boltzmann machine learning (BML) algorithm [23]. By performing gradient ascent on the log-likelihood of the model given the data, BML determines values of couplings and fields such that the one- and two-site model frequencies match the empirical ones derived from the MSA. A standard pairwise q-state PM is thus specified by $q^2L(L-1)/2$ couplings and qL fields, where, for proteins, $q = 21$ corresponds to the 20 naturally occurring amino acids plus the gap symbol used for insertions or deletions.

Crucially, despite the fact that modern sequencing techniques are making available a huge amount of biological sequences, and in particular hundreds of millions of protein sequences [24], a serious over-fitting problem is present when PM are used as models of protein families. In fact, with typical sequence lengths $L \sim 50-500$, the parameters to be inferred are $\sim 10^6 - 10^8$, which in most cases substantially exceeds the available information from the MSA. The resulting over-fitting is manifested in several ways: (i) many couplings turn out to be rather small and noisy, (ii) the PM is close to a critical point, *i.e.* it can be very susceptible to small changes in its parameters, and (iii) different training procedures, *e.g.* with different initial conditions, can lead to significant changes in the sets of parameters without affecting the fitting accuracy, which severely limits the interpretability of the model.

These observations call for a parameter reduction procedure, which aims at identifying a minimal set of couplings needed to accurately describe the training data without overfitting. Hopfield-Potts models [25] and the more general Restricted Boltzmann Machines (RBM) [26] lead to a dimensional reduction of parameter space by learning collective "patterns" from sequence data, which in turn can be interpreted as extended sequence motifs and are activated via a limited number of hidden variables. The resulting coupling matrix is low-rank but still dense. A complementary approach aims at sparsifying the network of couplings: ℓ_1-norm regularization has been used in a number of approximate methods [27, 28], but cannot be easily used for generative modeling, because the regularization penalizes also non-zero couplings, which in turn assume too small values. Alternatively, a "color-compression" scheme [29] has been proposed, which groups together sequence symbols with low frequency in specific sites. However, frequent symbols may also be involved into statistically non-supported couplings. Another example is that proposed in [30] where a candidate sparse graph topology is sought by pruning the MSA columns associated with low values of the mutual information. Although this method has to be preferred when L is so large to prevent the standard DCA implementations, it completely loses some information on the target statistical model. Overall, a statistically principled and efficient approach to construct sparse PM for protein sequence modeling is still lacking.

In this work, we introduce an information-theory based "decimation" procedure, which allows for an iterative and controlled removal of irrelevant couplings. As a result, parameters are removed either if they have no statistical support (as in color compression), or if they have statistical support for being very small. We show that up to about 90% of the coupling parameters can be removed without observing any substantial change in the fitting accuracy and in the generative properties of the resulting Sparse Potts Model (SPM). Although greedy, our pruning scheme does not require to add extra terms in the energy function of the model, at variance with any treatable regularization, like ℓ_1 or ℓ_2, and therefore it preserves the generative properties of PM. Finally, we show that the resulting SPM are not close to criticality, at variance with the original PM learned using standard DCA. Our results thus demonstrate that the observed criticality of PMs inferred from protein sequence data is not an intrinsic feature of the biological systems themselves, *cf.* [31], but results from the over-fitting in the learning procedure.

II. AN INFORMATION-GUIDED DECIMATION PROCEDURE

With each sequence $S = (s_1, \cdots, s_L)$ of length L, in which s_i can take q possible values ($q = 21$ for proteins), a PM associates a statistical "energy" or Hamiltonian $H(S)$, written as a sum over single-site fields $h_i(s_i)$ and two-site couplings $J_{ij}(s_i, s_j)$:

$$H(S) = -\sum_{1\le i<j\le L} J_{ij}(s_i, s_j) - \sum_{1\le i\le L} h_i(s_i)\ . \quad (1)$$

The negative of the Hamiltonian can be interpreted as a "fitness score" for protein sequence S, with an associated Boltzmann probability $P(S) = \exp\{-H(S)\}/Z$, where $Z = \sum_S \exp\{-H(S)\}$ is the partition function guaranteeing correct normalization of P. Hence, the surface defined by $H(S)$ over the space of sequences can be interpreted as a "fitness landscape" or – using a more cautious term – "sequence landscape" for the protein family represented by the training MSA. We define the "model density" d as the number of non-zero couplings $J_{ij}(a, b) \neq 0$ divided by the total number of possible couplings $q^2L(L-1)/2$. Note that this definition is given element-wise, *i.e.* for each i, j, a, b, and not block-wise for entire $q \times q$ matrices J_{ij} coupling two sites i, j. Fields are not decimated and do not contribute to the model density: we consider them an essential ingredient of the model because they encode amino-acid conservation.

A fully connected model, *i.e.* with $d = 100\%$, can be trained to arbitrarily high accuracy using standard BML [12]. Let us define the empirical one-site frequency $f_i(a)$ of observing amino acid a in position i in the MSA, and two-site frequency $f_{ij}(a, b)$ of observing amino acid a in position i and b in position j in the same sequence of the MSA. BML performs a gradient ascent on the log-likelihood, which gives update equations for the couplings

and fields at each learning epoch:

$$\begin{aligned}\delta h_i(a) &= \eta_h[f_i(a) - p_i(a)] \ , \\ \delta J_{ij}(a,b) &= \eta_J[f_{ij}(a,b) - p_{ij}(a,b)] \ ,\end{aligned} \tag{2}$$

where the $p_i(a), p_{ij}(a,b)$ are the one- and two-site marginal probabilities of the PM, which are estimated at each iteration of the learning by sampling $P(S)$ via a standard Markov Chain Monte Carlo (MCMC) simulation, and η_h, η_J are the learning rates for fields and couplings. These equations are iterated until convergence to a fixed point, at which the model almost perfectly matches the empirical frequencies. Note that a PM trained in this way also corresponds to the maximum entropy or least constrained model that is compatible with the one- and two-site empirical frequencies [13, 32].

Our decimation procedure consists in choosing pairs of sites $i < j$ and amino acids a, b, and fixing the corresponding coupling permanently to $J_{ij}(a,b) = 0$. The coupling is removed from the set of adjustable parameters, and the corresponding two-site frequency $f_{ij}(a,b)$ is no longer explicitly fitted in the subsequent BML epochs. However, an important property of PM is the so-called "gauge" or reparameterization invariance: the transformation

$$\begin{aligned}J_{ij}(a,b) &\to J_{ij}(a,b) + \mathcal{J}_{ij}(a) + \mathcal{K}_{ij}(b) \ , \\ h_i(a) &\to h_i(a) - \mathcal{H}_i - \sum_{j(>i)} \mathcal{J}_{ij}(a) - \sum_{j(<i)} \mathcal{K}_{ji}(a) \ ,\end{aligned} \tag{3}$$

leaves the Hamiltonian in Eq. (1) and the associated Boltzmann distribution $P(S)$ invariant, for any choice of the $\mathcal{J}$, $\mathcal{K}$ and $\mathcal{H}$. Hence, a gauge transformation can transform a zero coupling into a non-zero one and vice versa. Because the decimation procedure fixes some couplings to zero, it breaks this invariance.

We thus begin our decimation procedure by a "gauge fixing" step, which sets to zero $2q-1$ out of all q^2 entries of each coupling matrix J_{ij}. To do so, we identify, independently for each pair of sites $1 \le i < j \le L$, the $2q-1$ amino-acid pairs (a,b) of smallest connected correlation $c_{ij}(a,b) = f_{ij}(a,b) - f_i(a)f_j(b)$, and fix the corresponding couplings $J_{ij}(a,b)$ to zero. Only the other $q^2 - 2q + 1 = (q-1)^2$ couplings are updated using the BML, Eq. (2). This procedure chooses a model of minimal density $d = [(q-1)/q]^2 = 90.7\%$ out of all equivalent PM related by the gauge transformation in Eq. (3). The parameters are initialized using a "profile model" fitting only the one-site frequencies $f_i(a)$. This initial model has zero couplings and fields $h_i^{(0)}(a) = \log f_i(a) + \mathcal{H}_i$, with the constant $\mathcal{H}_i$ being fixed by $\sum_a h_i^{(0)}(a) = 0$ (with a very small pseudo-count added to $f_i(a)$ to avoid infinite fields, see Appendix A 3). The fitting quality of the learned PM is tested by the Pearson correlation between the empirical $c_{ij}(a,b)$ and their counterparts in the model $P(S)$, the latter being estimated from a large independently and identically distributed MCMC sample. For all protein families considered in this work, this Pearson correlation exceeds 0.95, see Fig. 1 and Appendix B 1. Note that the results of our decimation procedure depend on the initialization and gauge fixing described above. We tried a different initialization, either fixing both couplings and fields to zero, or initializing both using pseudo-likelihood maximization (PLM) [11]. We found qualitatively similar results, but with slightly worse performance (Appendix C 3).

Any further decimation of couplings changes the model. To measure the impact of removing a given coupling $J_{ij}(a,b)$ from a PM, we determine the symmetric Kullback-Leibler (KL) divergence between the Boltzmann distributions with and without that coupling. We thus consider a Potts Model with Hamiltonian H, and another with Hamiltonian H' in which a given coupling is removed:

$$H'(S) = H(S) + J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j} \ . \tag{4}$$

We observe that averages over $P' = e^{-H'}/Z'$ can be expressed in terms of averages over $P = e^{-H}/Z$ as

$$\begin{aligned}\langle O(S)\rangle_{P'} &= \frac{\sum_S O(S) e^{-H'(S)}}{\sum_S e^{-H'(S)}} \\ &= \frac{\sum_S O(S) e^{-J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}} e^{-H(S)}}{\sum_S e^{-J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}} e^{-H(S)}} \\ &= \frac{\langle O(S) e^{-J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}}\rangle_P}{\langle e^{-J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}}\rangle_P} \ .\end{aligned} \tag{5}$$

Hence, the symmetric Kullback-Leibler divergence of P and P' is

$$\begin{aligned}D_{ij}^{ab} &= D_{\mathrm{KL}}(P||P') + D_{\mathrm{KL}}(P'||P) \\ &= -\sum_S [P(S) - P'(S)][\log P(S) - \log P'(S)] \\ &= \langle H' - H\rangle_P - \langle H' - H\rangle_{P'} \\ &= \langle J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}\rangle_P - \langle J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}\rangle_{P'} \\ &= \langle J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}\rangle_P \\ &\quad - \frac{\langle J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j} e^{-J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}}\rangle_P}{\langle e^{-J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}}\rangle_P} \\ &= J_{ij}(a,b)p_{ij}(a,b) - \frac{J_{ij}(a,b)p_{ij}(a,b)e^{-J_{ij}(a,b)}}{p_{ij}(a,b)e^{-J_{ij}(a,b)} + 1 - p_{ij}(a,b)} \ ,\end{aligned} \tag{6}$$

where $p_{ij}(a,b) = \langle \delta_{a,s_i}\delta_{b,s_j}\rangle_P$ is the marginal two-site probability of P, which coincides, at convergence of Eq. (2), with the empirical frequency $f_{ij}(a,b)$. Note that we could also have equivalently used the non-symmetrized KL divergence (Appendix A 1).

At each decimation step, we now remove the least significant couplings, *i.e.* those with the lowest D_{ij}^{ab}. For computational efficiency, this is done for a fixed fraction (in this work we choose 1%) of all remaining couplings. Note that $D_{ij}^{ab} = D(J, p \sim f)$ (dropping the indices for notational simplicity) goes to 0 either when $f \to 0$ or $f \to 1$ at fixed J, or when $J \to 0$ at fixed f. More

precisely, we have $D(J,f) \sim Jf(1-e^{-J})$ for $f \to 0$, $D(J,f) \sim J(f-1)$ for $f \to 1$, and $D(J,f) \sim J^2 f(1-f)$ for $J \to 0$. The first and second limits imply that finite couplings can be decimated if the corresponding frequency is close to zero or one, *i.e.* they have little statistical significance because the corresponding amino acids are almost never observed (as in color-compression [29]) or almost always observed. The third limit indicates that small couplings are decimated whatever f is (similar to the procedure proposed in [33] for the inverse Ising problem using PLM). Numerically, we observe that the percentage of pruned couplings corresponding to each category varies during decimation (Appendix C 2). After a decimation step, we perform additional BML iterations of Eq. (2) on all undecimated couplings and the fields, to reach convergence again. In this way, we progressively obtain PMs of reduced density, and we stop the decimation when $d = 1\%$.

Note that in order to accurately estimate D_{ij}^{ab}, it is important that the PM learning is well converged before each decimation step. We attempted an "online" decimation in which couplings are pruned either after a fixed number of iterations of Eq. (2) or for having reached convergence, and found that this provides no advantage (Appendix C 4), neither in terms of generative performance (*i.e.* the Pearson correlations at $d = 1\%$ do not improve), nor in computational efficiency (*i.e.* the computational time required to reach $d = 1\%$ is almost unchanged). Other decimation strategies based on $f_{ij}(a,b)$ only (removing statistically unsupported couplings), or on $J_{ij}(a,b)$ only (removing small couplings), or on applying ℓ_1-norm regularization to select relevant couplings, were found to perform substantially worse than the information-based procedure using Eq. (6) (Appendix C 1).

III. RESULTS AND DISCUSSION

We focus here on a representative protein family, the PF00076 family from the Pfam database [6], corresponding to a RNA recognition motif (RRM) of about 90 amino acids, known to bind single-stranded RNAs. The MSA provided by Pfam contains $M = 137605$ sequences of aligned length $L = 70$. Results obtained for other families (Appendix B) fully confirm the general conclusions drawn here for the RRM.

In Fig. 1 we show, for model densities down to 1%, the Pearson correlation coefficient between the empirical one-site frequencies $f_i(a)$ obtained from the original MSA, and the model one-site marginal probabilities $p_i(a)$, estimated by MCMC sampling. Similar curves are also shown for the two-site connected correlations $c_{ij}(a,b)$ and for a selected sub-set (specified in Appendix B 1) of three-site connected correlations, defined as

$$\begin{aligned} c_{ijk}(a,b,c) &= f_{ijk}(a,b,c) - f_{ij}(a,b)f_k(c) \\ -f_{jk}(b,c)f_i(a) &- f_{ki}(c,a)f_j(b) + 2f_i(a)f_j(b)f_k(c) \ , \end{aligned} \tag{7}$$

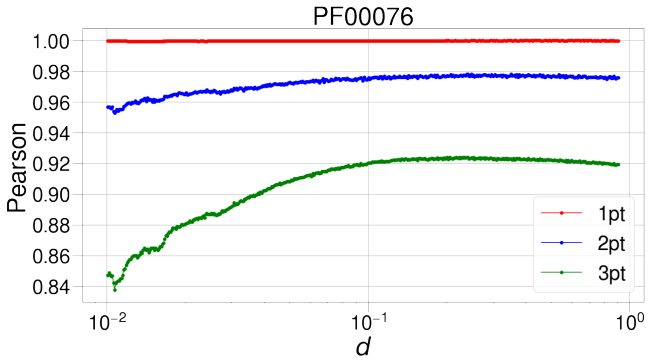

FIG. 1. **Fitting and generative quality** – Pearson correlation coefficient between model and data frequencies as a function of the model density. The one-site frequencies $f_i(a)$ are directly fitted. The two-site connected correlations $c_{ij}(a,b)$ are fully fitted by the densest model, while only a fraction of them are fitted for the sparse models at $d < 1$. The three-site connected correlations $c_{ijk}(a,b,c)$ are never fitted. The generative performance of the model is essentially unchanged down to a density of 10%, and slowly decays for even sparser couplings. However, even down to $d = 1\%$, the Pearson coefficients remain at remarkably high values above 95% for the two-site correlations, and above 84% for the three-site correlations.

where i,j,k are the indices of the columns of the MSA (which take value from 1 to L), and a,b,c run over the amino-acids and the gap symbol (practically, from 1 to q). The one-site frequencies are perfectly reproduced by the model, *i.e.* $f_i(a) = p_i(a)$, as a consequence of the fixed-point condition in Eq. (2), and the Pearson coefficient thus remains equal to one at all d (up to tiny deviations due to the finite MCMC samples used in BML and in estimating $p_i(a)$). For the maximal density $d_{\max} = [(q-1)/q]^2$ obtained after gauge fixing, the two-site correlations should also be perfectly reproduced because of Eq. (2). In practice we only reach a Pearson coefficient of ~ 0.975 due to sampling noise (Appendix A 4). On the contrary, for $d < d_{\max}$ only a fraction of all two-site frequencies is explicitly fitted by the model via sparse BML. Nevertheless the two-site Pearson coefficient is essentially independent of d, up to a slight reduction when $d < 10\%$. Finally, three-site correlations, that are never explicitly fitted by the model (the training process in Eq. (2) does not include three-site information), are nevertheless very accurately reproduced, with a Pearson coefficient around 0.94 for all $d > 10\%$. Note that the reproduction of unfitted observables is a highly non-trivial test for the generative properties of our models [12], *i.e.* of the capacity of the model to generate data being statistically close to indistinguishable from the natural sequence data used for model learning. Below density $d = 10\%$, the generative quality of the model for three-site correlations is slightly reduced, remaining nevertheless very high (above 84% down to $d = 1\%$). We discuss the generative property of the sparse models

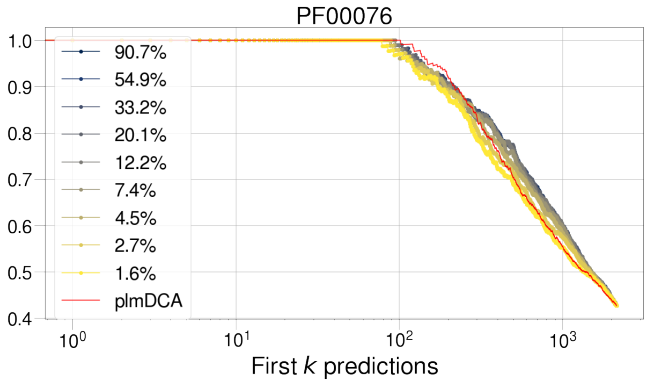

FIG. 2. **Contact prediction** – Positive predictive values (PPV) for several model densities, *i.e.* the fraction of true positives among the highest-ranking k pairs (i, j) of sites, when ordered by decreasing $F_{ij}^{\rm APC}$. Even the most sparse model, with only 1.6% of couplings, shows an excellent performance at contact prediction. The curve for plmDCA, a standard DCA approach for contact prediction, is shown for reference and gives comparable results.

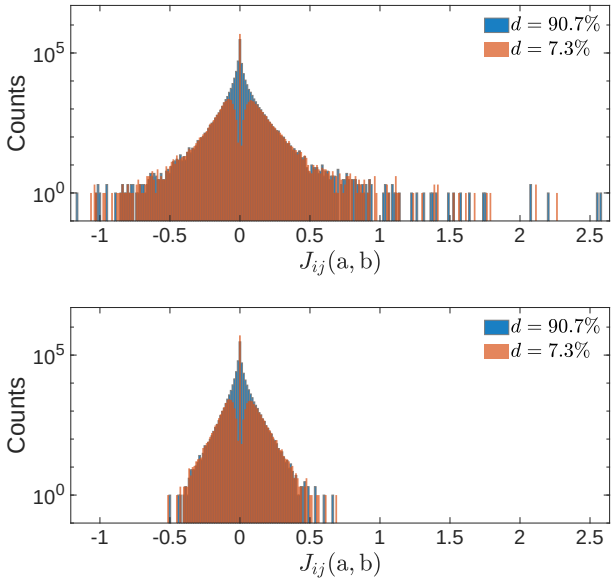

FIG. 3. **Coupling distributions** – Distribution of couplings corresponding to true contacts (top) and to non-contacts (bottom) in the three-dimensional protein fold, for the initial PM with density $d_{\rm max} = 91\%$ and a sparser model having density $d = 7\%$ associated with a reasonably accurate contact prediction.

introducing additional metrics in Appendix D.

A second test of model quality is the prediction of structural contacts, which constituted the major application of DCA in the last years. The idea is that pairs of strongly interacting sites in the PM should correspond to close-by residues in the three-dimensional structure, which display strong coevolution to maintain the proper protein fold and functionality. Using the standard convention for coevolutionary contact prediction, we consider a pair of residues to be in contact if the distance between them is at most 8 Å, and we exclude easy-to-predict short-range contacts by considering only pairs with $|i - j| \geq 4$ in our analysis. The reference (ground-truth) distance was obtained by the package [34] that takes the shortest distance between heavy atoms in all protein structures registered in the Protein Data Bank (PDB) [35] for the given Pfam family. We follow the standard procedure for contact prediction, which consists in computing the average-product corrected (APC) Frobenius norms of the coupling matrices (note that the coupling matrices are transformed into the zero-sum gauge and that the gap states $a, b = q$ are excluded from the sum [36]),

$$F_{ij} = \sqrt{\sum_{a,b=1}^{q-1} J_{ij}(a,b)^2} \; ; \; F_{ij}^{\rm APC} = F_{ij} - \frac{\sum_k F_{ik} \sum_k F_{kj}}{\sum_{kl} F_{kl}} \; . \tag{8}$$

In Fig. 2 we show the fraction of true contacts within the first k pairs of sites, ranked in decreasing order of $F_{ij}^{\rm APC}$. We observe that the performance of the model at inferring the structural contacts is only slightly deteriorated even in the sparsest case $d = 1.6\%$.

In Fig. 3 we show the probability distributions of couplings $J_{ij}(a, b)$, separately for pairs $i < j$ corresponding to contacts and all the others. We observe that, both for contacts and non-contacts, the decimation affects the shape of the distribution around $J \sim 0$ in a similar way, while the tails are essentially unaffected. Overall, these results explain why the performance of the PM for contact prediction using $F_{ij}^{\rm APC}$ is essentially independent of d (Fig. 2). Unfortunately, the large-J tail of the distributions of couplings on non-contacting sites does not change upon sparsifying the model, which suggest that our decimation procedure cannot help in devising better contact predictors.

In order to study the criticality of the models, we consider a simple perturbation of the statistical weight, *i.e.* we rescale the Hamiltonian $H(S)$ by a formal inverse temperature $\beta = 1/T$ and set $P(S) \sim e^{-\beta H(S)}$, in such a way that $T = 1$ corresponds to the original model trained on data, while measuring the variation of the model entropy S. In Fig. 4 we show the heat capacity $C = T{\rm d}S/{\rm d}T$ of the PM for several sparsities (see Appendix B 4 for details on the computation of C). Note that a large C indicates a large variation of the model entropy with T, or equivalently that the model statistics changes strongly after a slight change of the parameters. This is indeed the best definition of criticality in statistical physics, keeping in mind that our models have finite size L and we thus cannot perform a finite-size scaling analysis to determine if the observed peak in C corresponds to a phase transition in the thermodynamic limit. In Fig. 4 we observe that the denser models display a

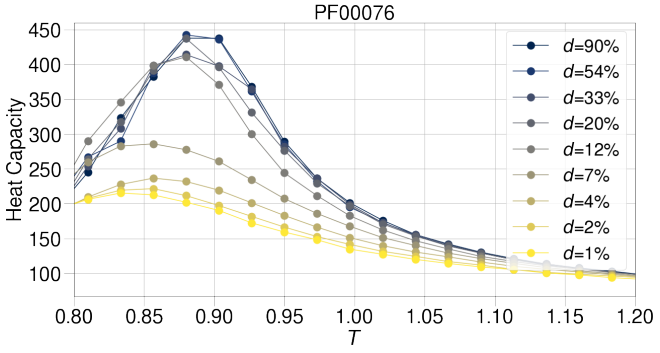

FIG. 4. **Criticality** – Heat capacity as a function of temperature for models with different density. The densest models show a strong peak of specific heat close to the reference scale $T = 1$, which is a signature of criticality: the model is extremely sensitive to a small change of couplings, due to over-fitting. On the contrary, sparse models display a much smaller peak, which is also shifted away from $T = 1$ towards lower temperatures, indicating a better robustness of the learning.

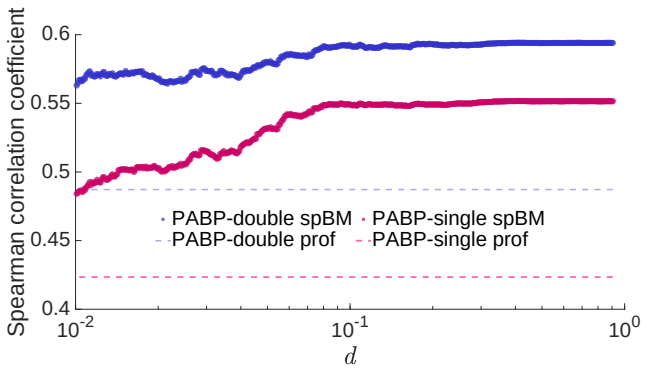

FIG. 5. **Single and double mutations** – Spearman correlation between the experimental fitnesses and the model predictions as a function of the model density, both for single and double mutants of the PABP, a member of the PF00076 family. The dashed lines show the same correlations for a profile model (*i.e.* $d = 0$) as a reference.

large peak in C close to $T = 1$, which indicates that the models are close to criticality. Upon sparsifying the model, the peak amplitude is strongly reduced and the peak is also shifted to lower temperatures, *i.e.* further away from the reference scale $T = 1$. These results suggest that the criticality of the dense models comes from over-fitting. Because the dense models have a huge number of parameters, they are able to fit all the details of the training data. As a consequence, the model becomes very sensitive to noise, and a little change of the parameters changes a lot the model statistics. On the contrary, sparse models have less parameters and are thus more robust to noise.

Ref. [37] provides Deep Mutational Scanning (DMS) data for a representative member of the PF00076 family, namely the RRM2 domain of the Poly(A)-Binding Protein (PABP) in the yeast species *Saccharomyces cerevisiae*. Using this domain as a reference, the authors generated a library of 110,745 protein variants, including 1,246 single amino-acid substitutions and 39,912 double amino-acid substitutions. Each of these variants was experimentally scored for function, by monitoring the growth of mutant yeast and finally, a "fitness score" was attributed to each mutant sequence in the experiment [37]. Within our models, the inferred Hamiltonian $H(S)$ in Eq. (1) is also interpreted as a sequence-fitness score. Hence, a good test of the generative property of our models is to check whether the energy differences $\Delta H = H(\text{mutant}) - H(\text{reference})$ between mutant sequences and the PABP reference correlate well with the experimental fitness variations. Because the mapping between experimental and model fitness may be non-linear, in Fig. 5 we show the Spearman's rank correlation between these two variables, both for single and double mutants. In the dense $d = d_{\max}$ case, we reproduce the reference values already given in Ref. [20]. We also observe that upon reducing density, once again the model quality is not degraded, down to $d \sim 10\%$. Even for $d = 1.6\%$ the model performs quite well, and much better than a profile model, which coincides with the limit $d \to 0$ of our decimation procedure.

IV. CONCLUSIONS

We introduced a general parameter reduction scheme for Boltzmann Machine Learning, and we applied it to Potts models for protein sequence data, *i.e.* for the learning of highly accurate and generative, but sparse DCA models. Our strategy makes use of a rigorous information-based criterion to select couplings that are iteratively pruned. Intuitively, removed couplings are either statistically unsupported, *i.e.* they correspond to pairs of amino acids that are almost never or almost always observed, similarly to the color-compression scheme [29], or they are small, *i.e.* they correspond to pairs that are only weakly correlated, or a combination of both. The statistical significance of a coupling is precisely quantified by the symmetric KL divergence between the Boltzmann measure of the Potts model with and without this coupling, which is exactly computable from the model or the empirical statistics.

While our method is fully general for learning Boltzmann machines from high-dimensional categorical data, here we focused on its application to model protein families via Potts models, in which strong couplings are usually associated with physical contacts in the three-dimensional protein fold. We stress that the aim of this work is not to provide a sparse graph topology underlying the true interaction network, and indeed the pruning is not performed block-wise but at the level of the individual coupling entries, but to provide a general framework

of parameters reduction strongly based on information-theoretic assumptions. We have shown that the model can be decimated down to less than 10% of the original couplings, while losing neither its generative quality, nor its accuracy in contact prediction. However, it has to be noted that many couplings not corresponding to structural contacts remain non-zero even in the lowest-density models. The interpretation of such couplings remains unclear. They may result from subtle effects due to the phylogenetic relations between the training sequences [38, 39], but also from extended functional constraints as those found by Restricted Boltzmann Machines or Hopfield-Potts models [25, 26]. As a result, further work is needed to make DCA-type modeling fully interpretable.

The sparse models resulting from our decimation procedure are also far away from criticality: they do not display the specific-heat peak close to the formal temperature $T = 1$ that characterizes the dense models. Hence, we attribute the criticality observed in dense models to over-fitting, and conclude that our decimation procedure makes model learning more robust to finite-sample noise. Finally, the model maintains its performance in predicting the fitness of mutations around a reference sequence, *i.e.* it is capable of predicting the *local* shape of the fitness landscape after having been trained on a *global* alignment of distantly diverged amino-acid sequences.

Our decimation procedure solves the first two problems mentioned in the introduction: we can eliminate small and noisy couplings, and the resulting model maintains its fitting and generative qualities, while being statistically more robust. Unfortunately, we were unable to solve the third problem, namely the strong dependence on the initial condition of the training: different initial conditions (zero couplings and fields, profile model, plmDCA) produce fully-connected PMs of equal fitting quality but with slightly different performances in predicting contacts and mutational effects. This difference does not disappear after decimation (Appendix C 3). In other words, our decimation procedure remains sensitive to the initial fully-connected model from which it is started.

The resulting sparse PMs attempt to fit the data by using the minimal number of two-site couplings, *i.e* using coupling matrices that are as sparse as possible. In the context of proteins (or RNA), it is natural to think that the sparse couplings identified by the model are the most relevant to describe the physical two-site correlations that arise from the need to maintain the three-dimensional folded structure. This strategy is complementary to collective-feature learning, *e.g.* via RBM or Hopfield-Potts models [25, 26], in which the number of parameters in the coupling matrix is reduced by assuming it to be of low-rank. The features learned by these machines are associated with global sequence motifs, related, *e.g.*, to protein function or its interactions, but the accuracy of contact prediction is reduced. An interesting and natural perspective would be to combine these two strategies into a general "sparse plus low-rank" scheme, *cf.* [40] for a related idea, which could lead to an accurate description of protein families in an easily interpretable way, with sparse two-site couplings describing physical constraints coming from structural contacts, and low-rank couplings describing biological features associated with protein function and its evolutionary history.

To conclude, we would like to stress once more that our information-based decimation strategy is not specific to the application of Potts models to protein sequence data. It can directly be used in other applications of inverse statistical physics and Boltzmann machine learning, as *e.g.* in modeling neural or socio-economic data [1], and may be adapted to more general network reconstruction schemes.

ACKNOWLEDGMENTS

We would like to thank Matteo Bisardi, Simona Cocco, Yaakov Kleeorin, Rémi Monasson, Rama Ranghanatan, Olivier Rivoire and Jeanne Trinquier for discussions related to this work. We acknowledge funding by the EU H2020 research and innovation programme MSCA-RISE-2016 (grant 734439 InferNet, to MW), by the Simons Foundation (#454955, to FZ) and by the Honjo International Scholarship Foundation (PhD grant to KS).

P.B.-C., A.P.M. and K.S. contributed equally to this work. Author contributions: P.B.-C. and M.W. designed the research; P.B.-C., A.P.M., K.S., and F.Z. performed the research; A.P.M. and K.S. analyzed the data; A.P.M., K.S., M.W., and F.Z. wrote the paper.

Appendix A: Methods

1. Alternative decimation score

Using the relation

$$\begin{aligned}\frac{Z'}{Z} &= \frac{1}{Z}\sum_S e^{-J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}} e^{-H(S)} \\ &= \langle e^{-J_{ij}(a,b)\delta_{a,s_i}\delta_{b,s_j}}\rangle_P \\ &= p_{ij}(a,b)e^{-J_{ij}(a,b)} + 1 - p_{ij}(a,b)\ ,\end{aligned} \tag{A1}$$

we obtain

$$\begin{aligned}\widehat{D}_{ij}^{ab} &= D_{\mathrm{KL}}(P||P') = \sum_S P(S)[\log P(S) - \log P'(S)] \\ &= J_{ij}(a,b)p_{ij}(a,b) \\ &+ \log[p_{ij}(a,b)e^{-J_{ij}(a,b)} + 1 - p_{ij}(a,b)]\ .\end{aligned} \tag{A2}$$

This second quantity also coincides with the variation of the likelihood of data under the change of model,

$$\begin{aligned}\Delta\mathcal{L} &= \frac{1}{M}\sum_{m=1}^{M}[\log P(S_m) - \log P'(S_m)] \\ &= \frac{1}{M}\sum_{m=1}^{M} J_{ij}(a,b)\delta_{a,s_i^m}\delta_{b,s_j^m} + \log\frac{Z'}{Z} \\ &= J_{ij}(a,b)f_{ij}(a,b) \\ &+ \log[p_{ij}(a,b)e^{-J_{ij}(a,b)} + 1 - p_{ij}(a,b)] \ ,\end{aligned} \tag{A3}$$

which coincides with (A2) when the model is well converged and $p_{ij}(a,b) = f_{ij}(a,b)$.

Note that the qualitative form of D_{ij}^{ab} and $\widehat{D}_{ij}^{ab}$ as a function of $f_{ij}(a,b)$ and $J_{ij}(a,b)$ is very similar, and D_{ij}^{ab} is a monotonous function of $\widehat{D}_{ij}^{ab}$. Using D_{ij}^{ab} or $\widehat{D}_{ij}^{ab}$ in the decimation procedure thus leads to fully equivalent results.

2. Data set

In the following, we report the details of the five protein families analyzed in our work, identified as PF00014, PF00072, PF00076, PF00595, and PF13354 in the Pfam database (https://pfam.xfam.org/) [6, 41]. For PF00014, PF00072, PF00076 and PF00595 we filter the full set of sequences downloaded from Pfam, keeping only those that have less than six consecutive gaps. Empirically, we have found that the presence of stretched gaps renders the training more difficult as the Markov Chain Monte Carlo (MCMC) used for sampling has difficulties in visiting both very gapped sequences and gap-free region of the sequence landscape in a proper way, i.e. proportionally to the correct Boltzmann weight. This leads to a systematic bias in the model statistics. For the Beta-lactamase2 family PF13354, we used a slightly different procedure: we downloaded the Pfam pHMM model for that family, and we scanned the NCBI database to obtain aligned sequences. We then filtered sequences according to two criteria: (i) 80% sequence coverage (i.e. less than 20% gaps) and (ii) redundancy reduction at 80% (so $M_{\rm eff} \approx M$ in this case). We also removed the sequence TEM-1 (which is used as reference in the deep mutational scanning, as discussed below), and all sequences very similar to it. Note that because there are overlapping Beta-lactamase families in Pfam, our procedure, based on a single pHMM, gives also sequences that would align better to some other family in Pfam, in particular to the Beta-lactamase family PF00144.

In Table I we show the name of the protein domain associated with each family, the length, i.e. the number of columns L of the multiple sequence alignment (MSA), the number of sequences M of the original MSA and $M_{\rm eff}$, the number of statistically relevant sequences after a standard re-weighting of close-by sequences [14].

3. Training protocol

We specify here the details of the Boltzmann learning used to train the dense Potts model and to refine the non-zero parameters within the decimation run.

First, we compute the data statistics from the input MSA as

$$f_i(a) = (1-\alpha)f_i^{\rm emp}(a) + \frac{\alpha}{q} \ , \tag{A4}$$

$$f_{ij}(a,b) = (1-\alpha)f_{ij}^{\rm emp}(a,b) + \frac{\alpha}{q^2} \ , \tag{A5}$$

with $f_i^{\rm emp}(a)$ and $f_{ij}^{\rm emp}(a,b)$ being the one- and two-site frequencies computed from the MSA (for all positions i,j and amino acids a,b), and with α being a pseudo-count [42] introduced to avoid divergent fields and couplings associated with poor statistics. Here we set $\alpha = 1/M_{\rm eff}$ except for the PF13354 family for which we set $\alpha = 10^{-50}$ (we observed that other values of the pseudo-count do not lead to a significant change of the trained models). Then, we start from a profile model, i.e. all couplings are set to zero and the fields are equal to $h_i(a) = \log[f_i(a)] + \mathcal{H}_i$ with $\mathcal{H}_i$ a constant ensuring $\sum_a h_i(a) = 0$. Subsequently we iteratively refine the parameters according to Eq. (2), using as learning rate $\eta_J = \eta_h = 5\cdot 10^{-2}$. We stop the algorithm when the convergence error ϵ, computed as the maximum error attained in the fitting of the two-site connected correlations,

$$\epsilon = \max_{i,j,a,b} |f_{ij}(a,b) - f_i(a)f_j(b) - p_{ij}(a,b) + p_i(a)p_j(b)| \ , \tag{A6}$$

reaches 10^{-2} (this value may slightly change depending on the family, up to $5\cdot 10^{-2}$ for the PF13354 family which is the most difficult to train). At each iteration, we use Metropolis-Hasting MCMC to compute the model statistics $p_i(a)$ and $p_{ij}(a,b)$. We run $N_{\rm chain}$ independent MC chains, with $N_{\rm chain} = 3000$ for PF00014 and PF00595, $N_{\rm chain} = 1000$ for the longer PF13354, and $N_{\rm chain} = 5000$ for the copious families PF00072 and PF00076. The chains are initialized at the first iteration from a uniform independent random distribution over all possible amino-acids, gap included, and are then persistent over iterations, i.e. at each new iteration the chain is initialized from the last configuration of the previous iteration. Each chain runs for $T_{\rm eq} = 20$ MC sweeps (one sweep corresponds here to L single-site Metropolis-Hastings MC steps) before starting to sample 10 configurations spaced by $T_{\rm wait} = 10$ MC sweeps. Hence, the total number of generated samples in a single iteration is $10 \times N_{\rm chains}$ (from 10^4 to $5\cdot 10^4$ depending on the family), and each chain is evolved by 110 MC sweeps in a single iteration.

4. Sampling protocol

Once the training is complete, for the final set of parameters of the Potts Model, we need to generate a new

Identifier	PF00014	PF00072	PF00076	PF00595	PF13354
Protein domain	Kunitz domain	Response regulator receiver domain	RNA recognition motif	PDZ domain	Beta-lactamase2
L	53	112	70	82	202
M	13600	823798	137605	36690	7515
M_{eff}	4364	229585	27785	3961	7454

TABLE I. We show here the Pfam identifier, the name of the protein domain, the length, the number of sequences and the effective number of sequences for the families analyzed in our work.

sample, from which we compute the model statistics to be compared with the MSA statistics. As in training, the MCMC method used for the sampling is the standard Metropolis-Hasting algorithm, using N_{chain} independent MC chains, initialized from a uniform independent random distribution over all possible amino-acids, gap included. Each chain is evolved for T_{eq} MC sweeps in order to achieve equilibration, before we start collecting samples, the waiting time between each sampled configurations being T_{wait} MC sweeps. We specify in Table II the values of N_{chain}, T_{eq} and T_{wait} and of the total number of collected samples, M_{MC}. Note that the conditions for the sampling are different from those used in the learning.

We also compute, for each model, the Hamming distance $d_H(t)$ between an equilibrium configuration and its time evolution under the MCMC dynamics after t MC sweeps (averaged over initial configurations and over the dynamics), see Fig. 6. Obviously, $d_H(0) = 0$ and for short times, $d_H(t)$ grows linearly, with a coefficient given by the acceptance rate of single-site mutations in the MCMC dynamics. At long times, $d_H(t \to \infty)$ saturates at the average distance between two independent samples from the Potts Model equilibrium distribution. This quantity can also be computed by measuring the Hamming distance between two independent MC chains, after equilibration, and is reported as a red horizontal line in Fig. 6. The time it takes for $d_H(t)$ to reach its asymptotic value gives an estimate of the decorrelation time of the MCMC dynamics, i.e. the time needed to generate a new independent equilibrium sample.

We observe that for PF00014, PF00072 and PF00076, the decorrelation time is of the order of 10^2 MC sweeps, and independent of sparsity, which suggest that the model is sampled in equilibrium during the learning process. In fact, we obtain exactly the same model statistics upon resampling the model in different conditions.

For PF00595 the decorrelation time is $\approx 10^3$ MC sweeps for the dense model. Because our training is done with persistent chains, and a small learning rate, we still believe that proper equilibrium sampling is achieved during learning. This is confirmed by the fact that we reproduce the same model statistics under resampling. Furthermore, we observe that the decorrelation time is reduced upon sparsifying the model, which suggests that the sparse models are less critical, as we discuss below.

For PF13354 the situation is radically different. In this case, the decorrelation time is huge (more than 10^4 MC sweeps for the dense model). This is likely due to the presence of multiple subfamilies, such that the MC chains take a lot of time to jump from one subfamily to another. With such a long decorrelation time, learning becomes extremely hard and we cannot guarantee that equilibration is achieved during it. In fact, we find that upon resampling the model starting from random initial states, the statistics is initially good (after $\approx 2 \times 10^4$ MC sweeps) but then is degraded, indicating that the model suffers from overfitting due to poor equilibration during learning. For the sparse models, the decorrelation time is substantially reduced (by almost a factor 100), and consistently we find that resampling is stable at all times.

Identifier	PF00014	PF00072	PF00076	PF00595	PF13354
M_{MC}	30000	30000	30000	30000	30000
T_{wait}	60	80	60	90	100
T_{eq}	10000	50000	30000	50000	50000
N_{chain}	100	100	100	100	300

TABLE II. We report here the details of the MC sampling performed to evaluate the model statistics.

Appendix B: Results for the other protein families

In this section we report the same type of results shown in the main text for PF00076, but for the four remaining families: PF00014, PF00072, PF00595 and PF13354.

1. Fitting quality

To evaluate the quality of the sparse models we compute, for each possible density, the Pearson correlation coefficients between a certain type of statistics computed from the empirical data (the MSA) and the model (via MCMC). More precisely, we focus on one-site frequencies and two- and three-site connected correlations, as defined in Eq. (7). To select the indices of the most significant three-site connected correlations we have first extensively scanned all possible triplets and computed the empirical frequencies for all possible color assignment. We then keep the elements $c_{ijk}(a, b, c)$ with empirical absolute values above 10^{-4}: only for those elements we compute the corresponding model correlations, in order to limit the computational cost. The model correlations are computed from a set of samples generated via the MCMC

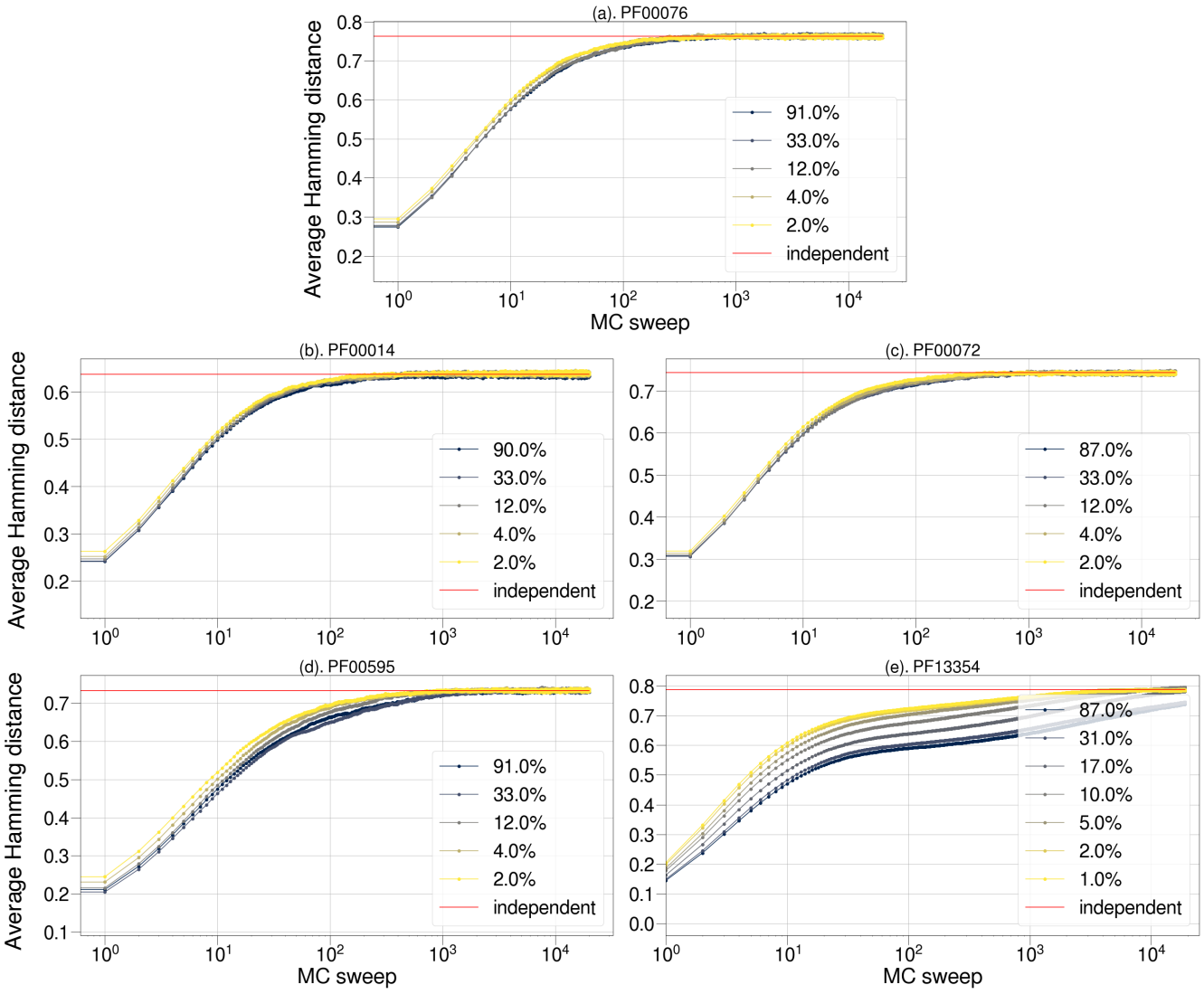

FIG. 6. Averaged Hamming distances between an equilibrium sequence at time $t = 0$ and the evolved sequence after t MC sweeps. The average is computed using 10^4 independent MC chains.

procedure described in Appendix A.

In Fig. 7, we show the Pearson correlation coefficients for the three metrics between the data and the models as a function of the model density, for the PF00014 (a), PF00072 (b), PF00595 (c) and PF13354 (d) families. For all families, the Pearson coefficient maintains almost the same value reached for the densest (fully connected) model up to a density of about 10%. When the density goes below 10%, the Pearson coefficient gradually decreases for all families; not surprisingly, the reduction associated with the three-site connected correlations is more pronounced, because this more-than-two-site correlation is not explicitly fitted by BM learning.

The case of PF13354 is special because, for the reasons discussed in Appendix A, the learning, which is done using rather short waiting times between samples, suffers from a very long decorrelation time in the dense case. Hence, the resampling degrades when MC chains are evolved for long times, which explains why the Pearson coefficients are poor for $d > 20\%$. For $d < 20\%$, the decorrelation time becomes much shorter, and the resampling is stable over time, but the Pearson coefficients get progressively degraded when d is reduced, as for the other families. The optimal compromise seems to be $d \approx 20\%$ for this atypical family.

2. Contact prediction

The APC-corrected Frobenius norms associated with the couplings can be used for scoring each pair of sites of the MSA (*cf.* main text). As already explored in literature, this score correlates well with the physical distances between pairs of residues in the three-dimensional struc-

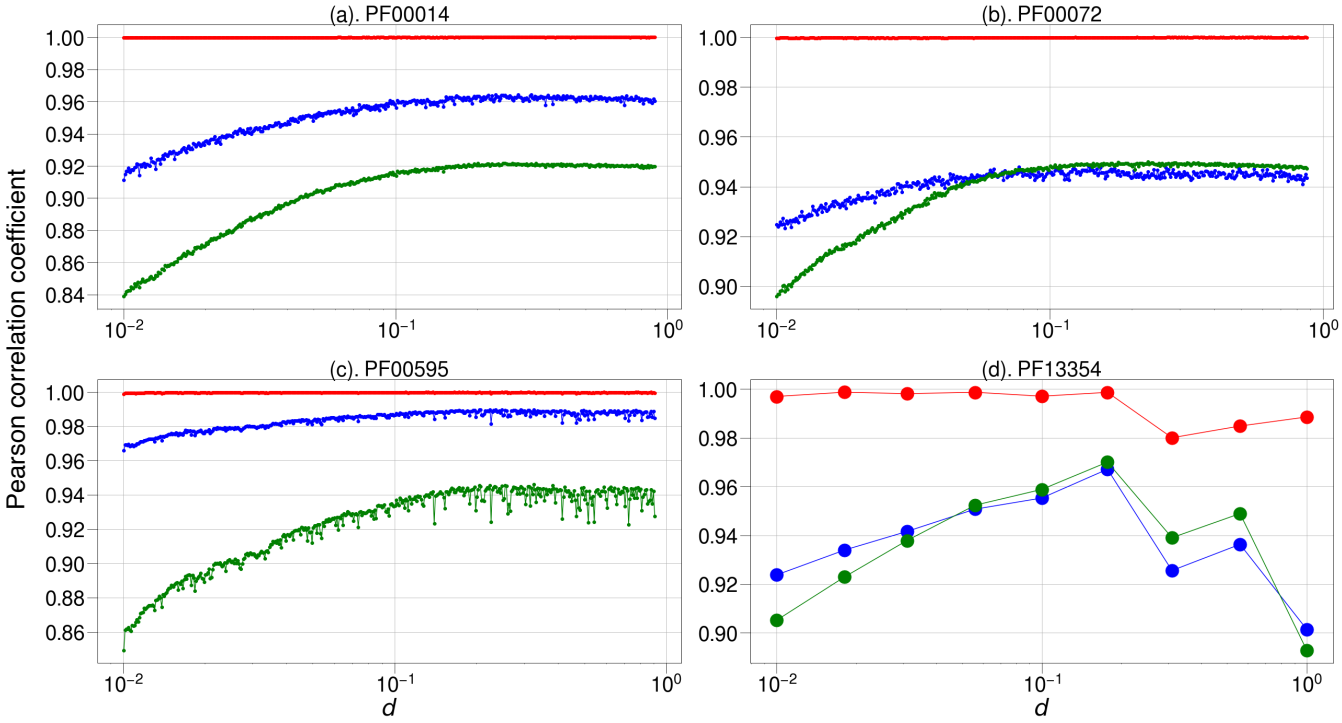

FIG. 7. Pearson correlation coefficients between the three chosen metrics (first moments, two-site and three-site connected correlations) of the data and the sparse models as a function of the density. Each panel (a), (b), (c) and (d) is associated with a different family, respectively, PF00014, PF00072, PF00595, PF13354.

ture of the protein domains. Larger Frobenius norms suggest larger probabilities of a physical interaction. As usual, we try to estimate the quality of the sparse models through a set of Positive Predictive Value (PPV) curves associated with the prediction of contacts. As reference structures we use those extracted from [34], a tool that outputs the shortest relative distance of pairs of residues over all known crystal structures registered in the Protein Data Bank (PDB) database [35]. In Fig. 8, we show the PPV curves for a sub-set of the sparse models (the density is mapped to a different color of the lines) together with the result of `plmDCA` [10] used here as comparison (red lines). Even keeping only 10% of the coupling parameters, i.e. when 90% of them are removed by the decimation procedure, the accuracy of the contact prediction remains stable, that is the performances are comparable to those of the fully connected models. The comparison to `plmDCA` is instead heterogeneous: as found in [12], the Boltzmann machine learning can have comparable performances to `plmDCA` as for PF13354 in panel (d), slightly worse as for PF00014 and PF00072 in panels (a), (b), or slightly better as for PF00595 in panel (c).

3. Coupling distribution

Because the couplings mirror a physical interaction among residues, one may guess that the more we decimate the model, the more we decimate the couplings not associated with residues in contact. Similarly, one may expect that the more a coupling is important in terms of three-dimensional structure, the larger will be its strength, hence it will be preserved by the decimation.

To check whether this is the case, we plot in Fig. 9, for PF00014 (a), PF00072 (b), PF00595 (c) and PF13354 (d), the distributions of the couplings linking residues in contact (panel 1) and not in contact (panel 2); the values of the corresponding densities are indicated in the legend. We note that as we reduce the density of the couplings, those corresponding to residues in contact are slightly enhanced (indeed, the original red histograms in Fig. 9 for the dense models are shifted to slightly larger values in the sparse case), but we do not observe a significant change in the tails of the distributions, as discussed in the main text.

4. Criticality

Dense Potts models are generally very sensitive to a perturbation of their model parameters: a slight change of the couplings or the fields leads to a dramatic transformation of the model statistics, which thus seems to be close to a phase transition, i.e. to be *critical*. A good measure of the criticality of statistical models is represented by the heat capacity, which is obtained by apply-

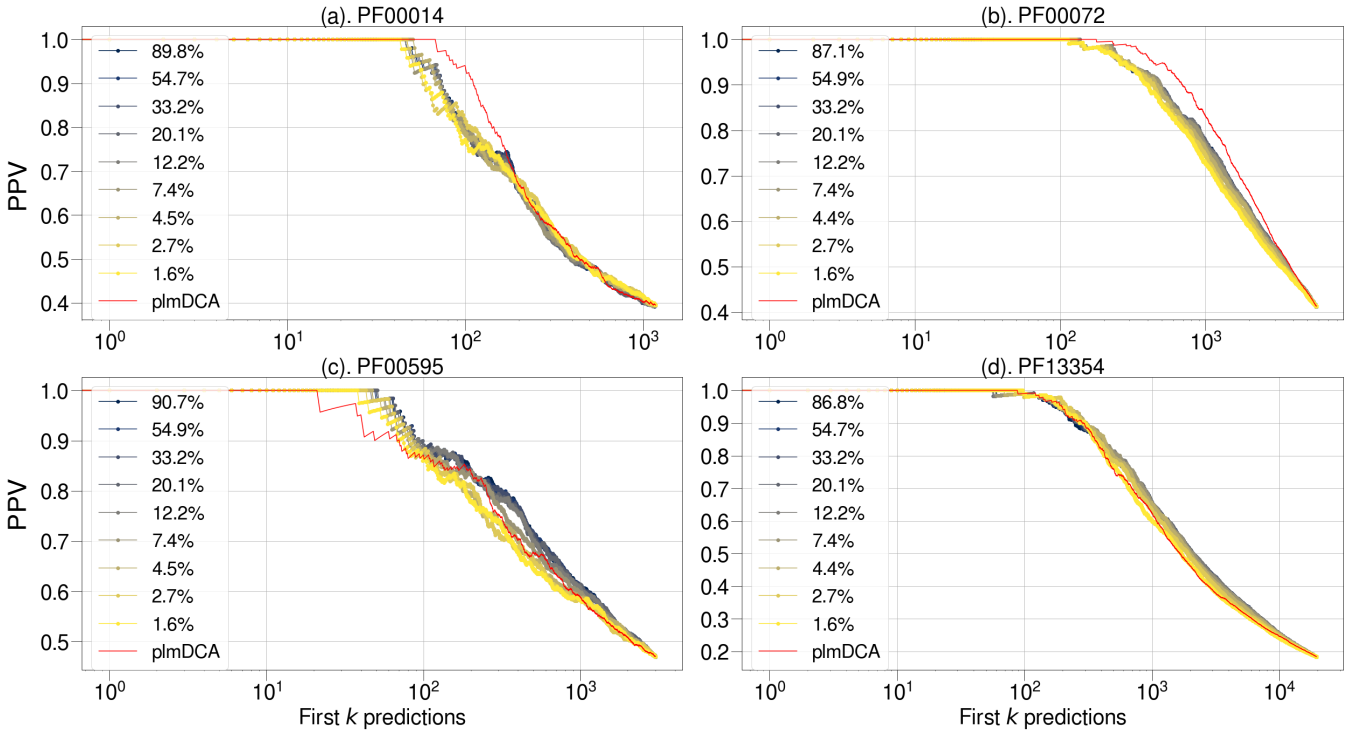

FIG. 8. Positive predictive value (PPV) curve for the other protein families associated with the contact prediction of several sparse models, from yellow to black lines. As a comparison we show the PPV curve (red line) obtained by the state-of-the-art method for this task, `plmDCA`.

ing a global variation to the parameters, $J \to J/T$, $h \to h/T$, and measuring the derivative of the average internal energy with respect to the temperature,

$$C(T) = \frac{\partial \langle H \rangle_T}{\partial T} = \frac{1}{T^2}\left(\langle H^2 \rangle_T - \langle H \rangle_T^2 \right) . \qquad \text{(B1)}$$

The averages in Eq. (B1), denoted as $\langle . \rangle_T$, are evaluated by sampling a system with Boltzmann weight $\exp\{-H/T\}$. Standard thermodynamic identities also show that $TC(T) = \partial S/\partial T$, where $S(T)$ is the entropy of the model. The model criticality is related to the magnitude of $C(T)$ in the vicinity of $T = 1$, which expresses how quickly the model entropy (or energy) varies under a small rescaling of all couplings.

Fig. 10 shows the behavior of the heat capacity $C(T)$ as a function of the temperature T for the models associated with the four families analyzed here: the color of the lines depend on the value of the density of the corresponding model, which spans the range (1, 90)%. We observe that for all families, upon sparsifying the model, (i) the heat capacity is reduced rendering the model less sensitive to changes in the model parameters and/or (ii) the peak slightly shifts towards a temperature smaller than $T = 1$, the natural temperature of the learning. In all cases, the value of $C(T = 1)$ decreases upon sparsifying the model. This observation suggests that a dense model learned by the empirical data is indeed close to a phase transition, but the criticality disappears (or decreases substantially) for the statistically equivalent sparser models. Hence, we conclude that the sensitivity of the dense model is related to over-fitting. Note that the suppression of criticality is also suggested by the reduction of the decorrelation time, as discussed in Appendix A.

5. Mutational landscape prediction

Similarly to the analysis we proposed in the main text for the PF00076 landscape and the experimentally determined single and double-mutants fitness, we show in Fig. 11 the Spearman correlation coefficient, as a function of the density, between the energy variation (computed according to our models) and the experimental fitness associated with single-residue mutations. Here we consider the libraries of single mutants for the Beta-lactamase2 domain of the TEM1 protein [43] (here the fitness is related to antibiotic resistance) and for the PDZ3 domain of the PSD95 protein [44] (here the fitness refers to the *CRIPT* ligand), which we assume to be described by the models for PF13354 and PF00595 families, respectively. As shown in Fig. 11, the correlation coefficient (*spBM* lines) between the experimental measures and the energy differences of our models are mostly constant as a function of the density; only a smooth increment (drop) is appreciated for densities smaller than 10^{-1} for PSD95 (TEM1). We remark that even in the sparsest case,

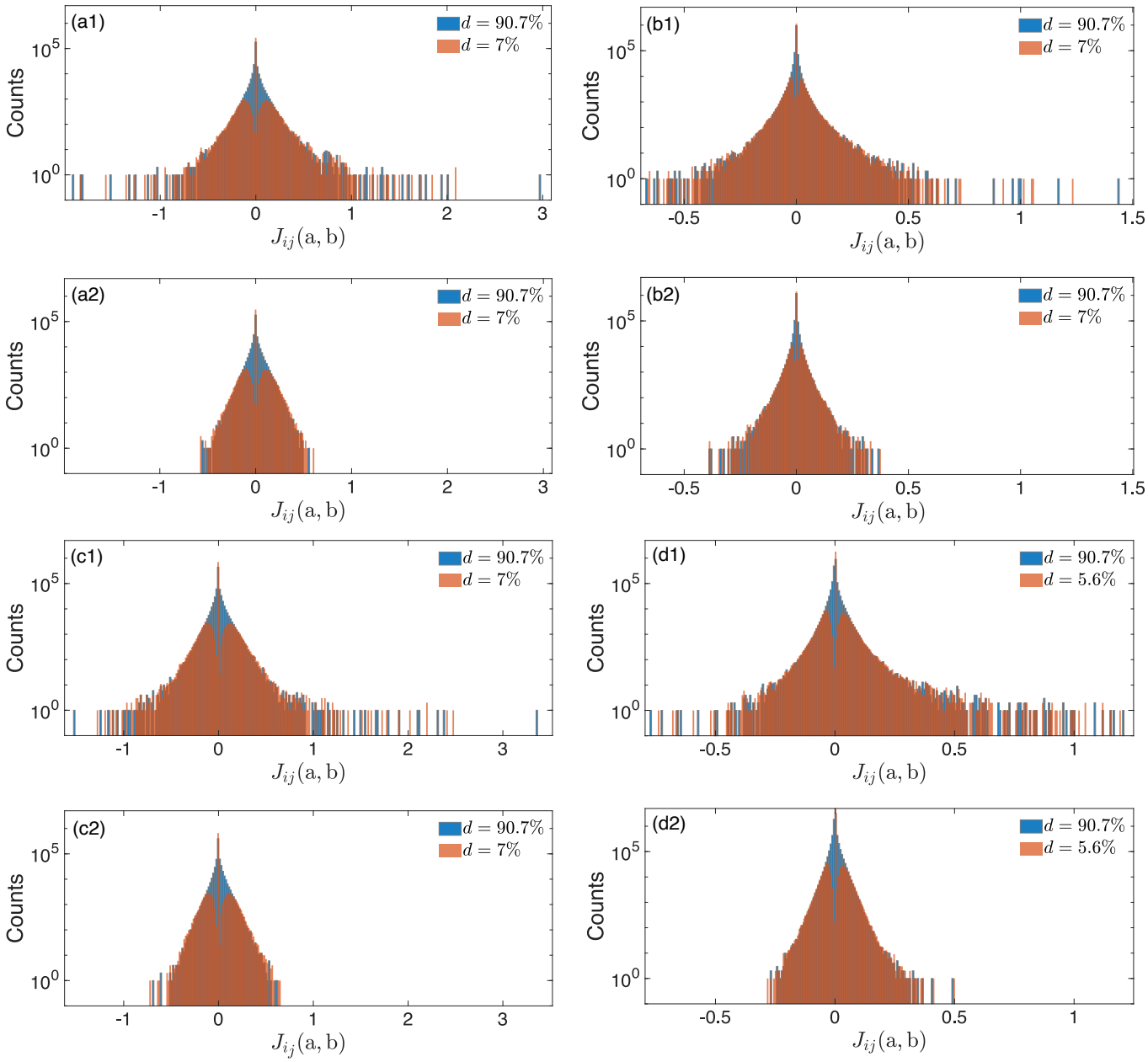

FIG. 9. Distribution of the couplings associated with residues physically in contact (labeled '1' histograms) and with residues not in contact (labeled as '2' histograms) for two different densities. Panels (a), (b), (c) and (d) refer to PF00014, PF00072, PF00595 and PF13354 respectively.

the Spearman correlation coefficient never crosses that obtained from a pure profile model (denoted as *prof*) suggesting that the remaining non-zero couplings of our sparse models are fundamental for the good description of the fitness landscape.

Appendix C: Additional results on PF00076

To complete the analysis described in the main text, we propose here a set of additional results for the PF00076 family. More precisely, we compare the learning and decimation strategy used in the main text and in Appendix B (initialize the parameters in the profile model, learn a dense model until convergence, then perform decimation) to several different initializations of the learning and to other decimation strategies based on different

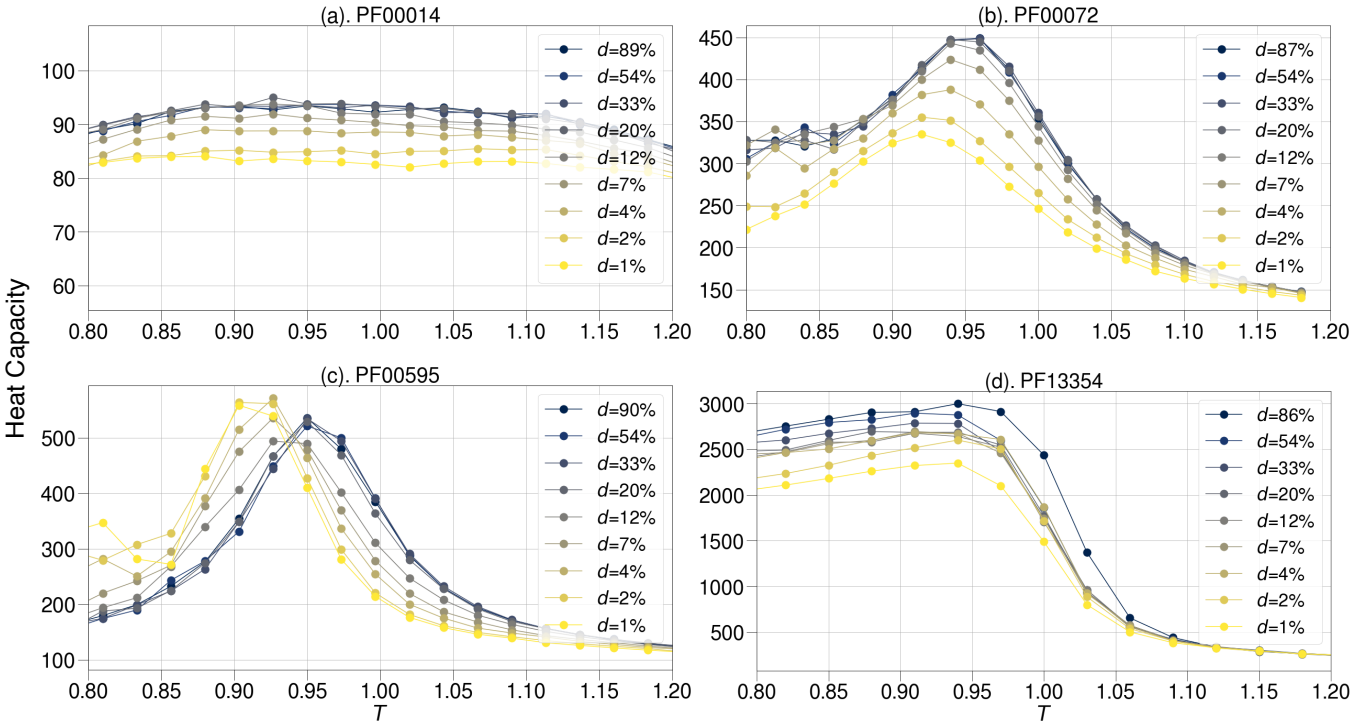

FIG. 10. Heat capacity as a function of the temperature T for the other protein families PF00014, PF00072, PF00595 and PF13354 in panels (a), (b), (c) and (d) respectively.

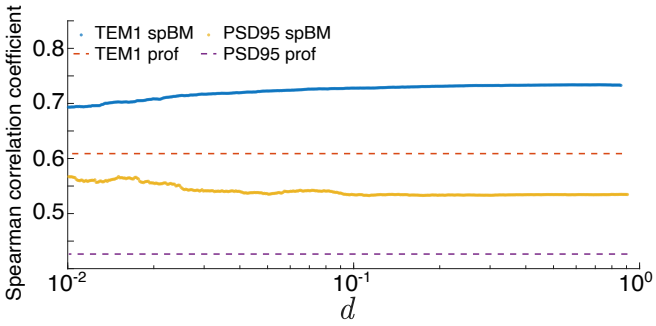

FIG. 11. Spearman correlation coefficient between the energy variations computed according to the sparse models (spBM lines) and the experimentally determined fitness variations of a set of single mutants, for the TEM1 (PF13354) and PSD95 (PF00595) proteins. The dashed lines show the results of the Spearman correlation coefficients when the energy variations are computed by the profile models of the corresponding families.

metrics. We also investigate the nature of the decimated couplings, via the statistics of the second moments associated with them, to stress the non-trivial nature of the symmetric Kullback-Leibler based decimation.

1. Decimation strategies

The method presented in the main text uses as criterion (or score) for the iterative decimation an information-theory based measure, the symmetric Kullback-Leibler divergence (symKLD) between the model with or without a certain coupling. As a result, the decimation score of each coupling takes into account both its statistical relevance (related to the second moments associated with it), and the strength of the coupling alone. We compare here the results presented in the main text to two simpler strategies where at each decimation step a) we remove 1% of the weakest couplings or b) we remove 1% of the couplings associated with the lowest, hence less statistically significant, two-site frequencies.

In Fig. 12, in the left panels, we compare the three possible decimation procedures using as comparison metric the fitting quality of the sparse models. We show the Pearson correlation coefficient of the empirical data and our sparse models predictions, as a function of the density, for the first moments (panel a), the two-site (panel b) and the most relevant three-site (panel c) connected correlations, respectively. Among the three procedures, that based on the two-site frequencies gives the poorest results, as it always provides the lowest Pearson up to density $\approx$3% where the algorithm fails to converge, meaning that it is no more able to fit the statistics associated with the non-zero parameters. The decimation based on the coupling strength outperforms the frequencies-based

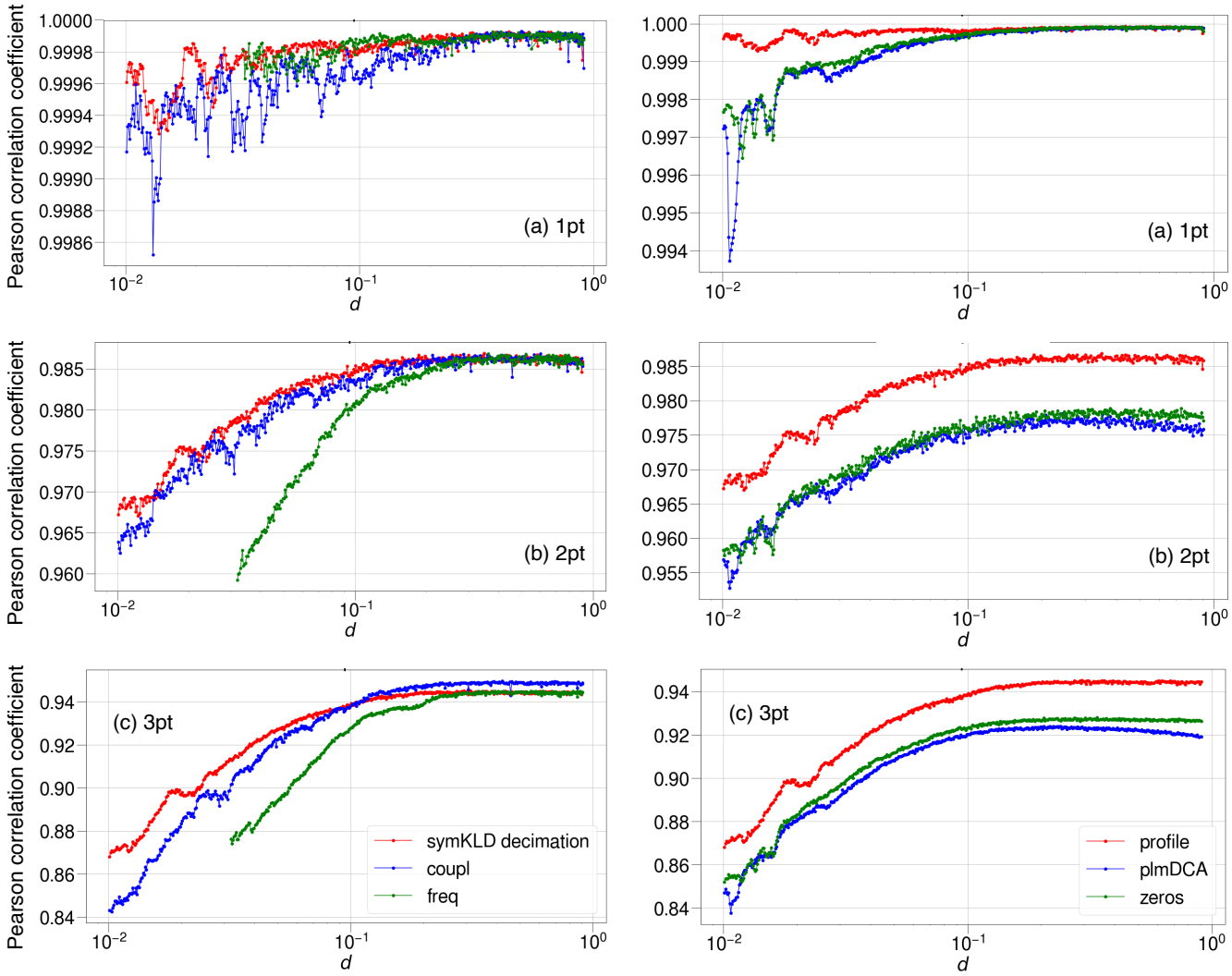

FIG. 12. Left panels: Pearson correlation coefficients between (a) the first moments, (b) the two- and (c) three-site connected correlations of each model (varying the density) compared to the empirical data. The lines are colored according to the metrics used within the decimation procedure: the symmetric Kullback-Leibler distance, the strength of the couplings or the two-site frequencies. Right panels: same plots, but varying the initial condition of the dense model learning. All data are for the PF00076 family.

one but the Pearson coefficients, for all comparison metrics, is systematically lower than that of the symKLD-based decimation.

In addition to the fitting quality, we compare the three methods looking at the contact prediction PPV curves, shown in the top panel of Fig. 13, varying the model density. It is worth noting that all procedures, for all densities (except 3.2% using a frequency-based measure) perform equally well.

We also considered a standard network selection strategy, in which we first learn a series of dense models with a ℓ_1-norm regularization at different strength γ, i.e. Eq. (2) for the couplings is modified to

$$\delta J_{ij}(a,b) = \eta_J[f_{ij}(a,b) - p_{ij}(a,b)] - \gamma \mathrm{sgn}(J_{ij}(a,b)) \ . \tag{C1}$$

At convergence, all couplings such that $|f_{ij}(a,b) - p_{ij}(a,b)| < \gamma$ thus have zero gradient and are considered as decimated. In this way one can obtain PMs of different density d by tuning γ. After selection, the sparse PMs is trained again keeping the decimated couplings to zero, but without the ℓ_1-norm regularization for the non-decimated couplings, until convergence. The results for this procedure are reported in Fig. 14, and are outperformed by the symDKL-based procedure.

2. Decimated couplings

As mentioned in the previous section, the couplings that are decimated at each iteration are either associated

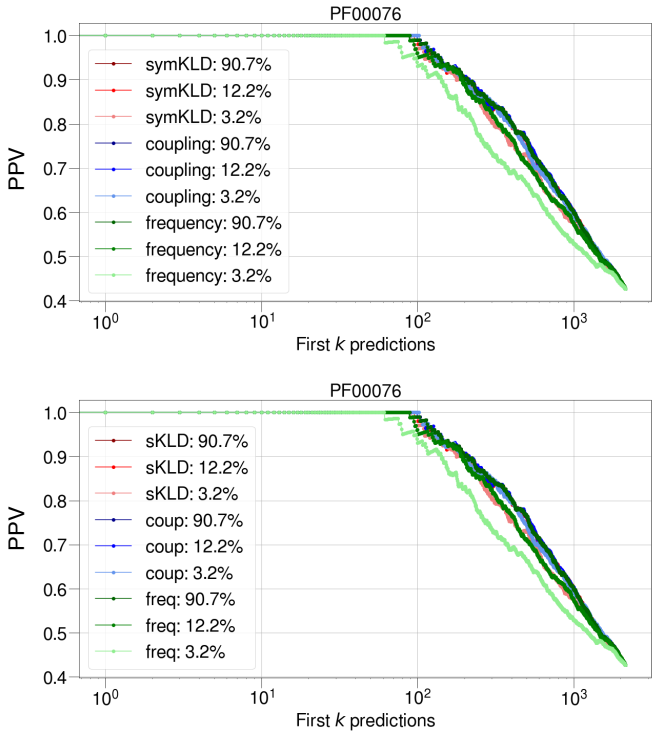

FIG. 13. Positive predictive value for each decimation procedure (top) and for each initial condition (bottom) for the PF00076 family.

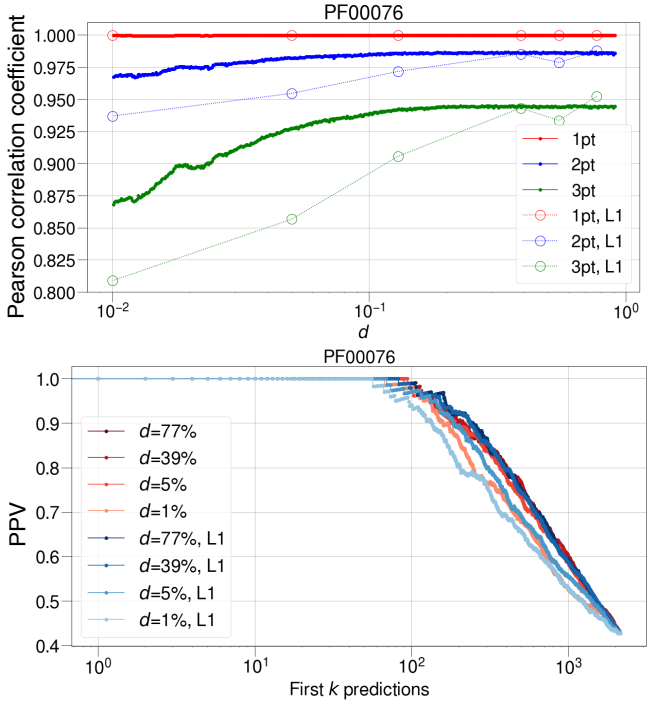

FIG. 14. Pearson correlations (top) and positive predictive value (bottom) for the decimation via ℓ_1-norm regularization, for the PF00076 family, compared with those reported in the main text.

with poor statistics, i.e. pairs of residues that are rarely or very frequently observed in two specific positions, or their strength is very small rendering their contribution in the Boltzmann weight negligible. It is interesting to quantify how many decimated couplings fall into the first or second class, as a function of the density. To this purpose we plot in Fig. 15 the empirical cumulative density function of the (logarithms of) the two-site empirical frequencies associated with the decimated couplings. We report in the same plot several curves depending on the density of the considered model: more specifically we observe the cases $d \in \{90.7, 69.9, 49.7, 12.2, 3.2\}\%$. The values of $\log_{10}[f_{ij}(a,b)]$ in the range $[-5, -4.3]$ empirically correspond to pairs of residues (a,b) appearing one time in position (i,j). Note that, although these frequencies are associated with a single occurrence, they span an interval, i.e. they are not always equal to the same value, because their computation takes into account the re-weighting protocol described in Ref. [14], in which each sequence may have a statistical weight smaller than one. Therefore, the value of the cumulative density function in $\log_{10}[f_{ij}(a,b)] = -5$ gives the fraction of decimated couplings associated with the pairs (a,b) that are never observed in sites (i,j). We see that this quantity changes as a function of the density: when the model is quite dense (for values of $d = \{90.7, 69.9, 49.7\}\%$) about 70 % of the decimated couplings corresponds to never observed statistics and thus only 30 % are associated with negligible couplings. As the model becomes sparser and sparser the fraction reduces and reaches about 30 % for the sparsest models: here about 70 % of the decimated couplings are associated with a rich statistics but nonetheless their contribution to the Boltzmann weight is negligible.

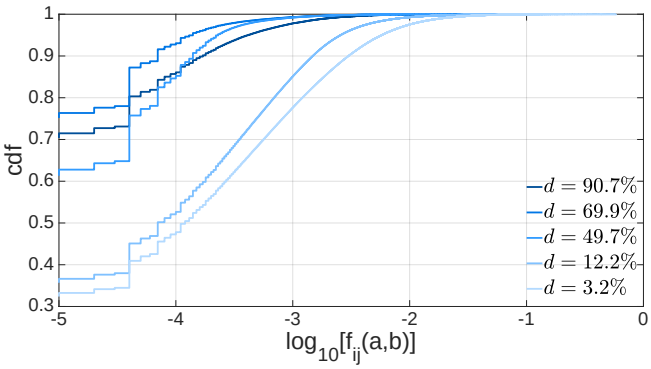

FIG. 15. Cumulative density function of the logarithms of the two-site frequencies associated with the decimated couplings for the sparse models having densities $d = 90.7\%$, $d = 69.9\%$, $d = 49.7\%$, $d = 12.2\%$ and $d = 3.2\%$. The data refers to the PF00076 and the decimation is performed according to the standard protocol described in the main text.

3. Initialization of the learning for the dense Boltzmann machine

An intrinsic difficulty arises when comparing statistical models for protein sequences: the set of parameters that are able to reproduce the empirical statistics well and also give a good contact prediction is not unique. Therefore, giving a clear interpretation of the fields and the couplings of the inferred Potts model, i.e. to detect which variables are sufficient to characterize the target ensemble of protein sequences, is a challenging task. When the sufficient set of observables is not known, and one attempts to fit all possible pairwise couplings and single-site statistics through the Boltzmann machine learning, it is common to encounter 'flat' directions of the log-likelihood landscape, where the learning usually converges (as any attempt at modifying the parameters does not lead to any significant improvement). The parameters found at convergence thus strongly depend on the initial conditions.

Here we evaluate how the results of the decimation procedure are affected by the dense model used as starting point, which in turn depends on the initial conditions of the parameters. For this comparison, we consider three distinct initial conditions for the initial learning of the dense model: (a) the profile model ($h = h^{\rm profile}, J = 0$, used for the results presented in the main text), (b) the parameters from pseudo-likelihood maximization ($h = h^{\rm plmDCA}, J = J^{\rm plmDCA}$), as implemented in `plmDCA` [10], and (c) a null initial condition for all model parameters ($h = 0, J = 0$). We then let the Boltzmann machine learning converge, and we use the converged Potts model as the starting model of the decimation run described in the main text.

In the right panels of Fig. 12 we show the Pearson correlation coefficients between the empirical frequencies $f_i(a)$ and the model frequencies $p_i(a)$ (in panel (a)), and the two-site and three-site connected correlations of the empirical data and of the sparse models, for panels (b) and (c) respectively. When all parameters are initialized to *profile* we reach the larger Pearson correlation coefficients, for all the three measures and for all densities. The *plmDCA* and *zeros* initializations have comparable results, and they reach Pearson correlation coefficients equal to those of the *profile* initialization only for the first moment in the high density regime.

In addition to the fitting quality, we can compare the three different initializations through the contact map prediction. We observe in Fig. 13 that all the three strategies, independently of the density, provide very similar contact prediction as the associated PPV curves completely overlap.

4. Online learning

In our decimation protocol, we proceed with a new decimation step only when the learning has reached convergence. Starting from a well converged dense Potts model, and decimating only 1% of the couplings at the time, allows us to modify smoothly the remaining parameters during the decimation. Indeed, we empirically observe that most of the times two consecutive decimations are separated by just a few learning steps. However, the entire protocol requires to learn a dense model first, which can be time-consuming.

We thus explored an alternative strategy in which the decimation is performed on-line, i.e. within a unique learning run. Here the decimation step is applied either because the learning has performed K steps or because it has reached the tolerance required for convergence. In these experiments, we start from a set of parameters corresponding to the profile model (as in the protocol illustrated in the main text) for PF00076 and we proceed with the decimation step every $K = 10, 20, 40$ steps.

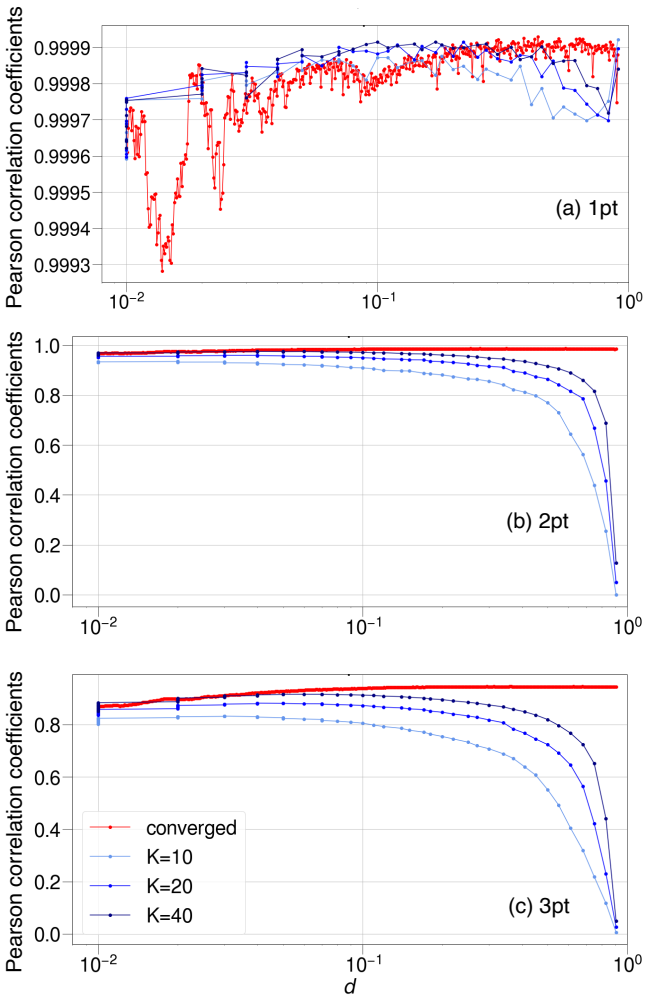

FIG. 16. Pearson correlation coefficients for the *online* learning (blue line), for $K = 10, 20, 40$, compared to the *converged* run (red line) as a function of the density, for the one-site frequencies (a), two-site (b) and three-site (c) connected correlations.

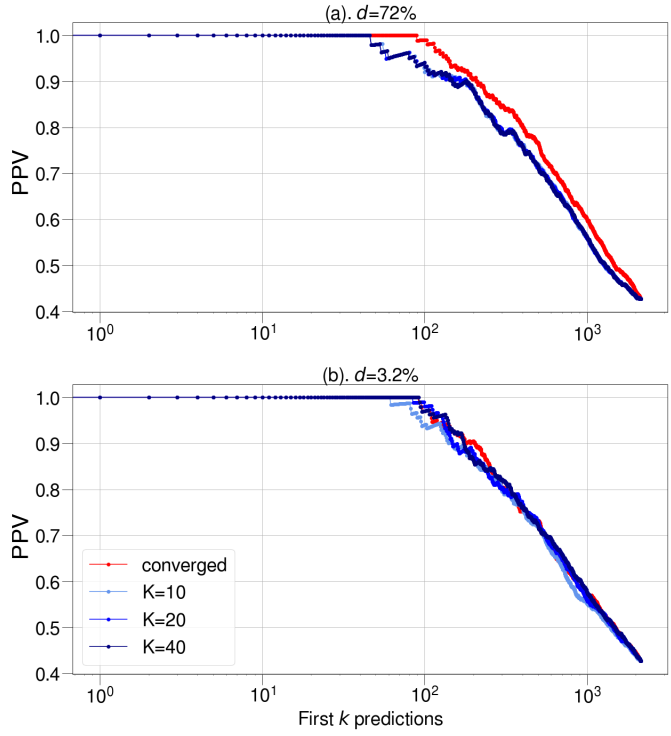

FIG. 17. Comparison between the Positive Predictive Value (PPV) curve obtained for the *online* (for $K = 10, 20, 40$) and *converged* runs at two different densities, 72% and 3.2%. All data are for the PF00076 family.

In Fig. 16 we compare the Pearson correlation coefficient between the one-site frequencies (panel a), two-site (b) and three-site (c) connected correlations, of the data and the models obtained by the two different strategies: we refer to the conventional method as *converged* (corresponding to $K \to \infty$) while the on-line learning method is characterized by the number K of steps. It is worth noting that, at convergence, both strategies, and independently of K, reach the same fitting quality even in the three-site connected correlations. For completeness, we show in Fig. 17 the contact prediction performance of the *converged* and on-line runs for densities equal to 72% (panel a) and 3.2% (panel b). In the denser case (when we consider 72% of non-zero couplings) the converged run outperforms the on-line learning for any K. This can be explained by the poor fitting quality reached by the on-line runs at the initial steps of the algorithm, that is when the model is still inaccurate in fitting the two-site frequencies. It is worth noting that, in the sparse regime, i.e. for density equals to 3.2%, all the strategies show comparable results, qualitatively similar to the performance of the dense case (Fig. 17a).

Although the results of the on-line run resemble those of the converged run for the very sparse models, the on-line procedure is not always advantageous from the point of view of the running time. We notice that, depending on the family, a unique learning-decimation run may have problems fitting the statistics, i.e. to converge, because the decimation affects and 'deviates' the learning of the machine, for small K. To cure this issue, one may think of increasing the number of steps between each decimation. However, this results in a very slow procedure, because we remove 1% of the couplings every (large) K steps. Instead, if the model is well converged first, then convergence is achieved quite fast after each decimation, resulting in a faster procedure overall.

Appendix D: Sequences similarity

The defining feature of generative models is the ability to generate configurations that are statistically equivalent to those used within the training process, but substantially different in the residue composition, i.e. a good generative model should not just reproduce the sequences of the training set. Hence, it is important to quantify the distances between generated samples and the training data. For this purpose, we employed the following metrics, introduced in [45, 46]:

$$D_Y(x) = \min_{y \in Y} D(x,y) \ , \qquad D_{XY} = \frac{1}{N_X} \sum_{n=1}^{N_X} D_Y(x_n) \ . \tag{D1}$$

where X and Y are ensembles of the generic statistical variables x and y, $D(x,y)$ is a certain distance defined for the sequences x and y. The metric $D_Y(x)$ computes the minimum distance of the sequence x reached when compared to each of the possible sequences in the ensemble Y; the quantity D_{XY} is instead the average value of $D_Y(x)$ over the ensemble of X. In our problem, we choose $D(x,y)$ as the Hamming distance between sequence x and sequence y and the ensembles X and Y are respectively t (the training set) and s, the synthetic sequences generated from the sparse Boltzmann machines. A proper generative model would produce comparable $D_{\rm st}$ and $D_{\rm ss}$ and, concurrently, the two measures must be sufficiently large (practically 20% of sequence similarity is required for good training sets). This corresponds to a scenario where generated sequences are variable (large $D_{\rm ss}$), and similarly distant to natural or the other generated sequences ($D_{\rm ss} \simeq D_{\rm st}$). This corresponds to a scenario where the average distance between each pair of generated sequences is comparable to that obtained between the two ensembles t and s: therefore, the generated synthetic sequences are indistinguishable from the natural sequences using distance based methods (like nearest-neighbor classification, distance based clustering). A similar argument can be applied to $D_{\rm ts}$. In Fig. 18, we show the average distances $D_{\rm ts}$, $D_{\rm st}$ and $D_{\rm ss}$ for each protein family; we do not show the $D_{\rm tt}$ measure which is obviously constant for all densities, and takes values $D_{\rm tt}(\text{PF00076}) = 0.308$, $D_{\rm tt}(\text{PF00014}) = 0.0917$, $D_{\rm tt}(\text{PF00072}) = 0.421$, $D_{\rm tt}(\text{PF00595}) = 0.295$, and $D_{\rm tt}(\text{PF00076}) = 0.445$. Because of the phylogenetic relationship among sequences, the training set is composed

of similar (correlated) sequences and, as a consequence, the $D_{\rm tt}$ is significantly smaller than the other distance metrics. Regarding $D_{\rm ts}$, $D_{\rm st}$ and D_{ss} we notice that, as the density of the couplings decreases, the distances remain unchanged up to a density in the range 10% - 20%, depending on the family. Then the minimum average distance significantly increases which suggests that the synthetic sequences are distributed more broadly in the sequence space as the number of model parameters decreases. Besides, the difference between $D_{\rm ts}$, $D_{\rm st}$, and $D_{\rm ss}$ decreases for most of the protein families, in the sparse regime, suggesting that the synthetic sequence ensembles and the set of the natural sequences become more and more statistically similar for increasing sparsity. We can conclude that, according to these metrics, the decimation improves the generative properties of the model.

[1] H. C. Nguyen, R. Zecchina, and J. Berg, Advances in Physics **66**, 197 (2017).
[2] J. D. Bryngelson, J. N. Onuchic, N. D. Socci, and P. G. Wolynes, Proteins: Structure, Function, and Bioinformatics **21**, 167 (1995).
[3] K. A. Dill and J. L. MacCallum, Science **338**, 1042 (2012).
[4] D. De Juan, F. Pazos, and A. Valencia, Nature Reviews Genetics **14**, 249 (2013).
[5] S. Cocco, C. Feinauer, M. Figliuzzi, R. Monasson, and M. Weigt, Reports on Progress in Physics **81**, 032601 (2018).
[6] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart, *et al.*, Nucleic Acids Research **47**, D427 (2019).
[7] R. M. Levy, A. Haldane, and W. F. Flynn, Current Opinion in Structural Biology **43**, 55 (2017).
[8] J. A. G. De Visser and J. Krug, Nature Reviews Genetics **15**, 480 (2014).
[9] S. Balakrishnan, H. Kamisetty, J. G. Carbonell, S.-I. Lee, and C. J. Langmead, Proteins: Structure, Function, and Bioinformatics **79**, 1061 (2011).
[10] M. Ekeberg, T. Hartonen, and E. Aurell, Journal of Computational Physics **276**, 341 (2014).
[11] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell, Phys. Rev. E **87**, 012707 (2013).
[12] M. Figliuzzi, P. Barrat-Charlaix, and M. Weigt, Molecular Biology and Evolution **35**, 1018 (2018).
[13] M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, and T. Hwa, Proceedings of the National Academy of Sciences **106**, 67 (2009).
[14] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, Proceedings of the National Academy of Sciences **108**, E1293 (2011).
[15] D. S. Marks, T. A. Hopf, and C. Sander, Nature Biotechnology **30**, 1072 (2012).
[16] S. Ovchinnikov, H. Park, N. Varghese, P.-S. Huang, G. A. Pavlopoulos, D. E. Kim, H. Kamisetty, N. C. Kyrpides, and D. Baker, Science **355**, 294 (2017).
[17] F. Morcos, N. P. Schafer, R. R. Cheng, J. N. Onuchic, and P. G. Wolynes, Proceedings of the National Academy of Sciences **111**, 12408 (2014).
[18] H. Szurmant and M. Weigt, Current Opinion in Structural Biology **50**, 26 (2018).
[19] M. Figliuzzi, H. Jacquier, A. Schug, O. Tenaillon, and M. Weigt, Molecular Biology and Evolution **33**, 268 (2015).
[20] T. A. Hopf, J. B. Ingraham, F. J. Poelwijk, C. P. Schärfe, M. Springer, C. Sander, and D. S. Marks, Nature Biotechnology **35**, 128 (2017).
[21] P. Tian, J. M. Louis, J. L. Baber, A. Aniana, and R. B. Best, Angewandte Chemie International Edition **57**, 5674 (2018).
[22] W. P. Russ, M. Figliuzzi, C. Stocker, P. Barrat-Charlaix, M. Socolich, P. Kast, D. Hilvert, R. Monasson, S. Cocco, M. Weigt, and R. Ranganathan, Science **369**, 440 (2020).
[23] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, Cognitive Science **9**, 147 (1985).
[24] The UniProt Consortium, Nucleic Acids Research **47**, D506 (2018).
[25] K. Shimagaki and M. Weigt, Physical Review E **100**, 032128 (2019).
[26] J. Tubiana, S. Cocco, and R. Monasson, Elife **8**, e39397 (2019).
[27] D. T. Jones, D. W. Buchan, D. Cozzetto, and M. Pontil, Bioinformatics **28**, 184 (2012).
[28] H. Kamisetty, S. Ovchinnikov, and D. Baker, Proceedings of the National Academy of Sciences **110**, 15674 (2013).
[29] F. Rizzato, A. Coucke, E. de Leonardis, J. P. Barton, J. Tubiana, R. Monasson, and S. Cocco, Physical Review E **101**, 012309 (2020).
[30] C.-Y. Gao, H.-J. Zhou, and E. Aurell, Phys. Rev. E **98**, 032407 (2018).
[31] T. Mora and W. Bialek, Journal of Statistical Physics **144**, 268 (2011).
[32] A. S. Lapedes, B. G. Giraud, L. Liu, and G. D. Stormo, Lecture Notes-Monograph Series , 236 (1999).
[33] A. Decelle and F. Ricci-Tersenghi, Physical Review Letters **112**, 070603 (2014).
[34] E. Sarti and A. Pagnani, "infernet-h2020/pfam_interactions: Initial release," (2020).
[35] H. Berman, K. Henrick, H. Nakamura, and J. L. Markley, Nucleic acids research **35**, D301 (2007).
[36] C. Feinauer, M. J. Skwark, A. Pagnani, and E. Aurell, PLoS Comput Biol **10**, e1003847 (2014).
[37] D. Melamed, D. L. Young, C. E. Gamble, C. R. Miller, and S. Fields, RNA **19**, 1537 (2013).
[38] C. Qin and L. J. Colwell, Proceedings of the National Academy of Sciences **115**, 690 (2018).
[39] E. Rodriguez Horta and M. Weigt, bioRxiv (2020).
[40] H. Zhang, Y. Gao, M. Deng, C. Wang, J. Zhu, S. C. Li, W.-M. Zheng, and D. Bu, Biochemical and Biophysical Research Communications **472**, 217 (2016).
[41] A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, S. Griffiths-Jones, K. L. Howe, M. Marshall, and E. L. Sonnhammer, Nucleic acids research **30**, 276 (2002).
[42] J. P. Barton, S. Cocco, E. De Leonardis, and R. Monasson, Phys. Rev. E **90**, 012132 (2014).

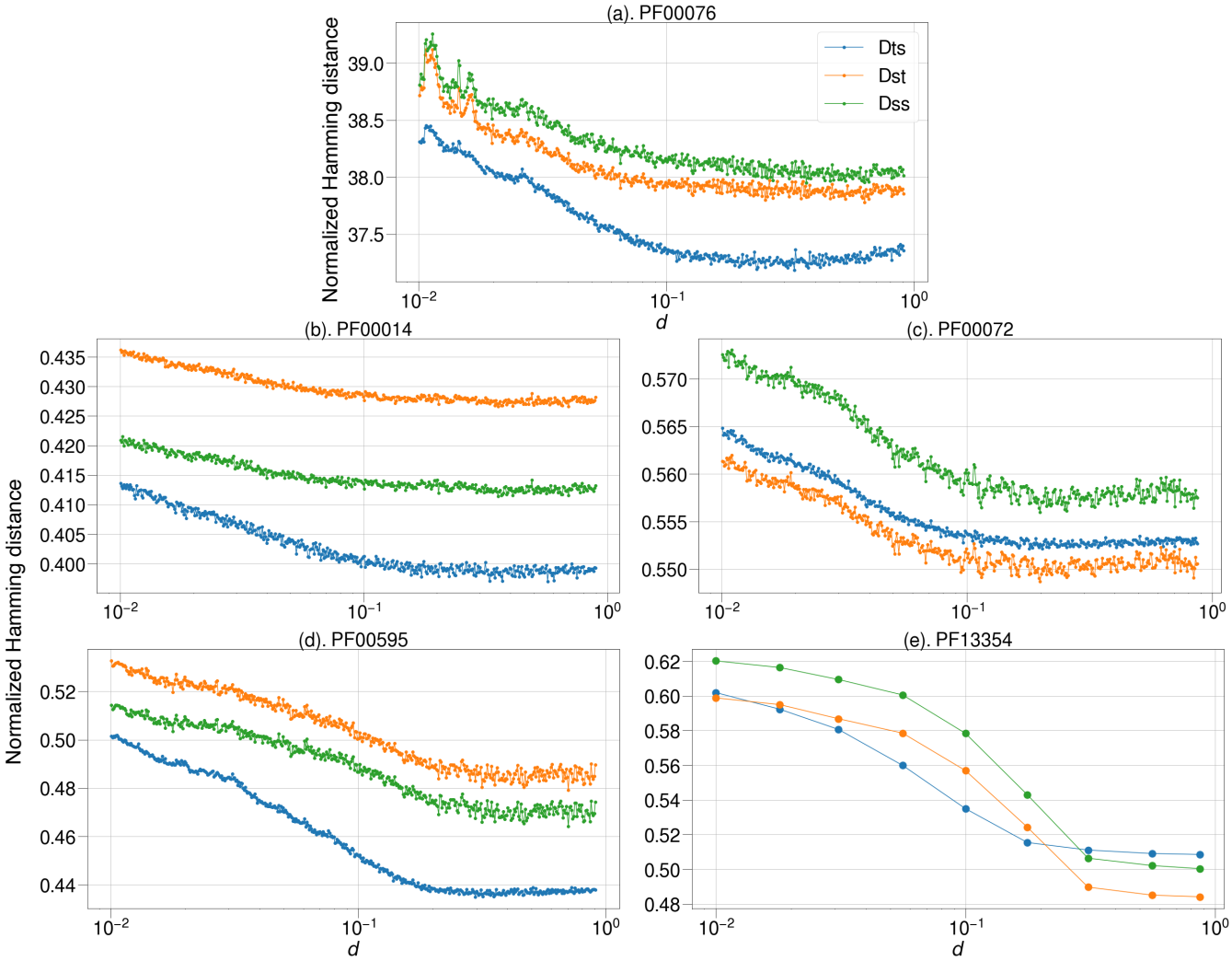

FIG. 18. Sequence variability for all the protein families considered in this work. We plot $D_{\rm ts}$, $D_{\rm st}$ and $D_{\rm ss}$ as a function of the density using blue, orange and green lines, respectively, for PF00076 (panel a), PF00014 (b), PF00072 (c), PF00595 (d), PF13354 (e).

[43] E. Firnberg, J. W. Labonte, J. J. Gray, and M. Ostermeier, Molecular biology and evolution **31**, 1581 (2014).
[44] R. N. McLaughlin Jr, F. J. Poelwijk, A. Raman, W. S. Gosal, and R. Ranganathan, Nature **491**, 138 (2012).
[45] A. Yale, S. Dash, R. Dutta, I. Guyon, A. Pavao, and K. Bennett, "Privacy preserving synthetic health data," (2019).
[46] B. Yelmen, A. Decelle, L. Ongaro, D. Marnetto, C. Tallec, F. Montinaro, C. Furtlehner, L. Pagani, and F. Jay, "Creating artificial human genomes using generative models," (2019).

3.3 Interpretation of couplings

Here are some additional results related to the three questions we asked at the beginning of this section.

3.3.1 Discrepancy of contacts and non-contacts

Since the accuracy of the contact predictions is maintained even in the case of the very low coupling densities ($d \sim 5\%$), the non-decimated couplings may tend to correspond with contacts. Intuitively, the difference between the probability of couplings for contacts and non-contacts is expected to becomes more noticeable as the density d decreases. Here, probabilities of couplings for contacts, $p_c(J)$, and non-contacts, $p_n(J)$, are defined as the probability densities of interactions $J_{ij}(a, b)$ for distances between residue-pairs $i < j$ smaller than 8Å and greater than 8Å , respectively. We introduce the Jensen–Shannon divergence (JSD), a distance between two probability distributions [1]

$$\mathrm{JSD}_{\mathrm{tot}} = \frac{1}{2} D_{KL}(p_c \| q) + \frac{1}{2} D_{KL}(p_n \| q) = \int_{-\infty}^{+\infty} \mathrm{d}J f(J) \ , \tag{3.1}$$

where $q = (p_c + p_n)/2$ and $f = p_c \log(p_c/q) + p_n \log(p_n/q)$. We define the relative cumulative contribution of couplings smaller than J to JSD as

$$\mathrm{JSD}(J) = \frac{\int_{-\infty}^{J} \mathrm{d}J' f(J')}{\mathrm{JSD}_{\mathrm{tot}}} \ . \tag{3.2}$$

Fig. 3.1.a and Fig. 3.1.b show histograms of the coupling for both contact and non-contact couplings for a dense and a sparse model. Both probability densities, p_c and p_n change around $J \sim 0$ while diluting the couplings. Interestingly, contact density p_c becomes slightly more pronounced by the coupling decimation at both tails ($0 \ll |J|$).

We can observe that JSD(J) changes more abruptly around $J \sim 0$ for the dense model. In contrast, this change is smoother and in a broader region for the sparse mode. In particular, there are significant contributions to JSD(J) in the tails of the distribution.

[1] The JSD is definable for all domains $\mathbb{R}$ even if the support of two distributions namely p^A and p^B are different, therefore it meets our study.

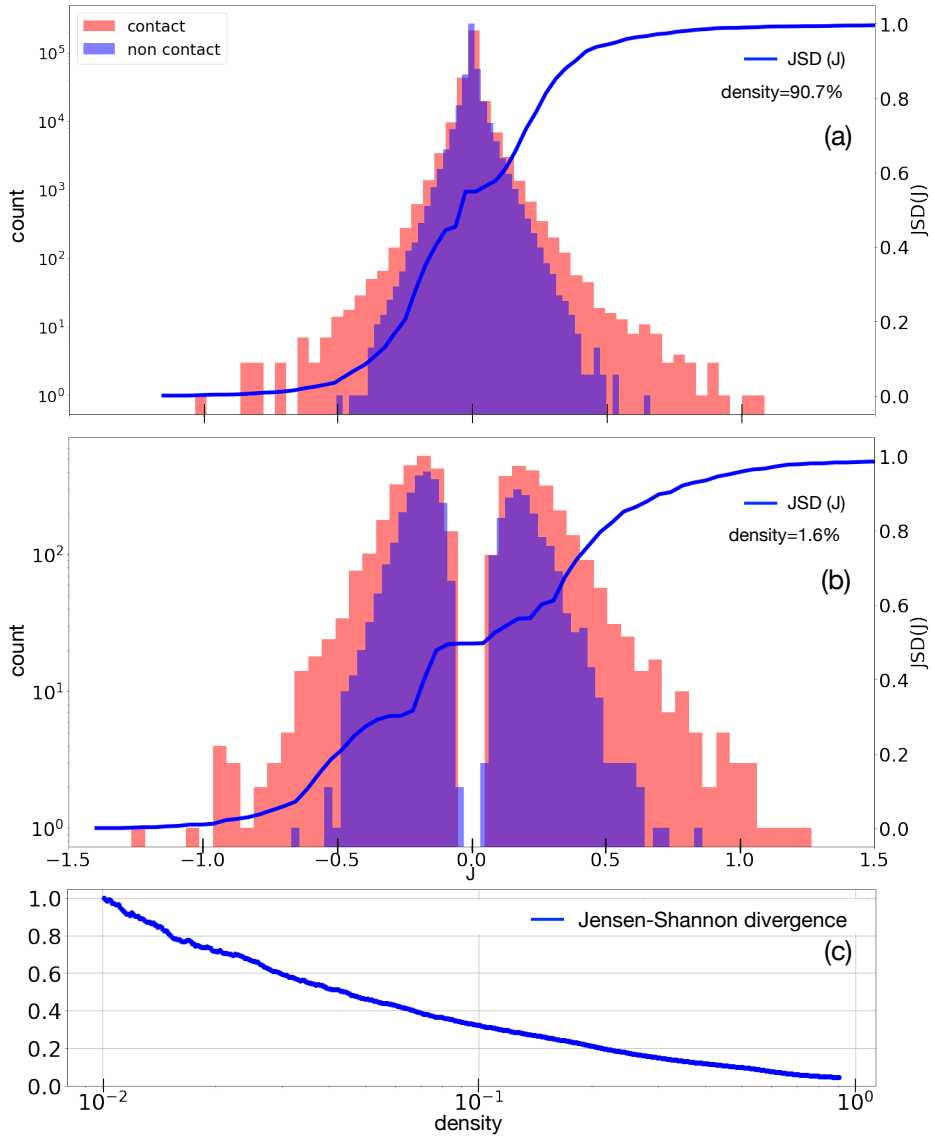

Figure 3.1: (a) Distributions of couplings for contact (red) and non-contact (blue) for a fully connected PPM with density $d = 90.7\%$ (some couplings are zeros due to the gauge choice as mentioned Sec. 2.4.2). The blue line shows cumulative JSD(J). (b) Same for a sparse model with $d = 1.6\%$. (c) It shows the JSD(J) as a function of the density d. We used the data set of PF00076, which is also used in the article [92].

Apparently, both contact and non-contact coupling parameters could increase as decreasing the coupling density d (cf. Fig. 9 in [92]). However, the rate of increase in the contact coupling parameters is much more pronounced than that of non-contact coupling parameters.

Empirically, residue contact predictions by the standard $\{F_{ij}^{APC}\}$ are not substantially changed under the decimations, whereas coupling distributions between contacts and non-contacts become more distinguishable. It would be interesting to exploit this discrimination in further studies.

By removing redundant couplings, the remaining couplings tend to be more enhanced to compensate for the removed couplings. The decimated weak correlations can probably be reproduced by contact couplings, whereas remaining non-contact couplings cannot significantly affect decimated correlations. Therefore, contact couplings are enhanced while non-contact couplings remain small.

3.3.2 Non-decimated and non-structural couplings

As shown before and in the article, large couplings $J_{ij}(a,b)$ that do not correspond to contact pairs remain in the set of non-decimated couplings (even at $d = 1.6\%$, there are non-contact couplings $|J| > 0.5$, cf. Fig. 3.1). Such non-decimated and non-contact couplings are also considered essential parameters in KLD-based parameter selection. Sometimes, non-decimated couplings correspond to too spatially distant residue pairs, even for long-range Coulomb interactions. Several explanations are possible for these non-decimated couplings. They could be associated with the underlying phylogeny, distant allosteric interactions and/or the molecular dynamics of the protein.

Fig. 3.2 shows some coupling matrices J_{ij} containing particularly strong couplings $J_{ij}(a,b)$ in the sparse model ($d = 1.6\%$). The coupling matrices of the sparse model show that the number of strong couplings pairs in contact is significantly greater than for pairs that are not in contact [2]. By construction, the coupling matrices in the sparse model contain a small number of finite couplings. Therefore, it is likely that those couplings that are not decimated are statistically important. However, the interpretation for the

[2] It is suggestive that using the L1 norm as the Frobenius norm may give better contact predictions than in the case of the L2 norm.

non-contact and non-decimated large couplings remains unclear.

Notably, the numbers of large entries in sparse non-contact coupling matrices are significantly fewer than the case in sparse contact coupling matrices. Therefore, instead of taking the L2 norm for converting the coupling matrices to Frobenius norm, it could be interesting to apply L1 or L0 norm to improve residue-contact predictions in further studies.

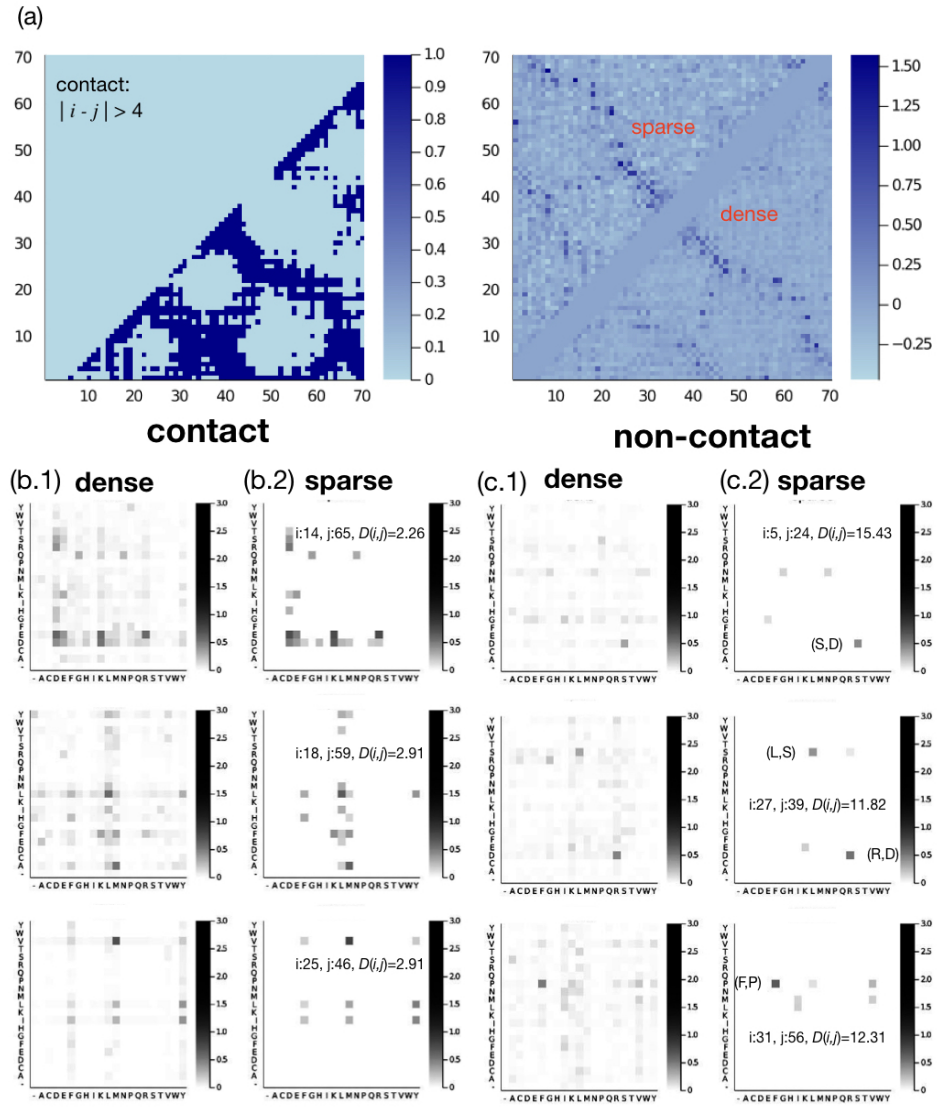

Figure 3.2: (a) Contact map (left) and the APC score matrix (right) for protein doamin, RNA biding domain (Pfam ID, PF00076). In the right panel, the upper left triangle and the lower right triangle are results for sparse model ($d = 1.6$) and dense model ($d = 90.7$), respectively. (b) Heat maps of coupling matrices J_{ij} for pairs that are close (< 3 Å) and have strong couplings ($J_{ij}(a, b) > 0.5$) in the sparse model. (c) The same type of heat maps but for non-contact pairs (> 11 Å). There are a few strong coupling entries in non-contact coupling matrices. However, the number of strong coupling is much smaller than the contact coupling matrices, and the difference is more pronounced in the sparse model.

As a last remark, the coupling matrices $J_{ij} \in \mathbb{R}^{L\times L}$ tends to have non-zero coupling parameters only between certain amino-acid states as shown in Fig. 3.2. Therefore, J_{ij} are essentially low-rank matrices and assuming the low-rank structure for coupling matrices, such as $J_{ij}(a,b) \sim \sum_\mu v_{ij}^\mu(a) v_{ij}^\mu(b)$, or equivalently assume a statistical model $p(\boldsymbol{A}) \propto \exp\left(\sum_\mu \sum_{i<j} v_{ij}^\mu(A_i) v_{ij}^\mu(A_j)\right)$ [3] might reflect well features of the estimated couplings parameters for protein sequences. The following section will introduce a statistical generative model that considers a low-rank structure for the coupling matrices.

3.4 Conclusion

In this chapter, we proposed pairwise-Potts models or Boltzmann machines for protein sequences that depend only on statistically important parameters. Using this method, we can decimate coupling parameters by about 90% while maintaining the reproducibilities of statistics and diversities of the generated samples, which are the expected properties as generative models. Note that a pairwise Potts model that reduces the number of coupling parameters by introducing a strong L1 regularization does not reproduce statistics as accurately as the proposed model (cf., Sec. 3.2).

By reducing coupling parameters, the model obtains the following advantages:

1. The original models are sensitive to the global perturbations of the coupling parameters. That is, these are in a state of over-learning, but the the proposed parameter-reduced models became more robust (cf. Fig. 4 in Sec. 3.2).

2. The increasing rate of parameters involved in physical contact is significantly greater than the case of non-contact couplings (cf. Fig. 3.1).

As Fig. 3.2 shows, the distribution of a finite number of elements in each coupling can be a useful measure to distinguish whether the pair is in contact or is not in contact. Further analysis should be done based on these insights.

[3] The fixed point equation for a paramter $v_{ij}^\mu(a)$ becomes a simple modification of the case of PPM,

$$\sum_b f_{ij}(a,b) v_{ij}^\mu(b) - \sum_b p_{ij}(a,b) v_{ij}^\mu(b) = 0 \ ,$$

where $p_{ij}(a,b)$ is a two-point frequency of the assumed model.

The accuracy of residue-residue contacts may be further improved by evaluating whether the estimated coupling parameters are significantly different from the background noize by statistical tests [93, 94].

Chapter 4

Selection of variables and low-rank model

4.1 Motivation

Despite their great success, DCA-based methods have drawbacks. In most situations, they depend on too many model parameters that need to be learned with a limited number of protein sequences. Thus over-learning problems are likely to happen. Moreover, the number of DCA parameters is necessarily high in order for these models to be generative, but a majority of these parameters are notably small and do not correspond to spatial interactions. Lastly, the choice of statistical variables or observables in MaxEnt modeling is still subjective. It might obscure the underlying simpler model for protein sequences.

As mentioned in Sec. 3.3.4, the coupling matrices can be low-rank matrices (cf. Fig. 3.2). This result is consistent biologically. Amino acids located on the protein surface are typically hydrophilic, whereas, those located in the protein core are normally hydrophobic. Therefore each coupling matrix shows specific amino-acid preferences. As a consequence, coupling matrices can exhibit a low-rank structure. Regarding these problems and the nature of the coupling matrix, the naturally emerging idea is to introduce a low-rank structure into the PPM. Moreover, by assuming the low-rank coupling matrices, effective number of parameter can be reduced.

Our contributions to this study are:

1. We show that even with a number of hidden variables (e.g., $P =$

$20 - 40$), which is sufficiently smaller than the coupling matrix size qL, RBMs can reproduce protein sequence statistics (Pearson values of two-point correlations are ~ 0.9).

2. We have proposed a framework that selects patterns based on the likelihood contributions. Such selected patterns can classify protein subfamilies successfully.

3. We show RBM with Gaussian hidden variables learns patterns similar to the attractive mfHP patterns (Sec. 4.3.1).

4. We show the rotation invariance in the space of the pattern indices can be fixed by using attractive mfHP patterns as initial conditions of RBM patterns (Sec. 4.3.2).

5. We show the fewer patterns in RBMs, the more robust RBM statistics will be to the global perturbations of the parameters (Sec. 4.3.3).

4.2 Article

Selection of sequence motifs and generative Hopfield-Potts models for protein families

Kai Shimagaki and Martin Weigt
Sorbonne Université, CNRS, Institut de Biologie Paris-Seine, Laboratoire de Biologie Computationnelle et Quantitative–LCQB, Paris, France

(Received 28 May 2019; published 19 September 2019)

Statistical models for families of evolutionary related proteins have recently gained interest: In particular, pairwise Potts models as those inferred by the direct-coupling analysis have been able to extract information about the three-dimensional structure of folded proteins and about the effect of amino acid substitutions in proteins. These models are typically requested to reproduce the one- and two-point statistics of the amino acid usage in a protein family, i.e., to capture the so-called residue conservation and covariation statistics of proteins of common evolutionary origin. Pairwise Potts models are the maximum-entropy models achieving this. Although being successful, these models depend on huge numbers of *ad hoc* introduced parameters, which have to be estimated from finite amounts of data and whose biophysical interpretation remains unclear. Here, we propose an approach to parameter reduction, which is based on selecting collective sequence motifs. It naturally leads to the formulation of statistical sequence models in terms of Hopfield-Potts models. These models can be accurately inferred using a mapping to restricted Boltzmann machines and persistent contrastive divergence. We show that, when applied to protein data, even 20–40 patterns are sufficient to obtain statistically close-to-generative models. The Hopfield patterns form interpretable sequence motifs and may be used to clusterize amino acid sequences into functional subfamilies. However, the distributed collective nature of these motifs intrinsically limits the ability of Hopfield-Potts models in predicting contact maps, showing the necessity of developing models going beyond the Hopfield-Potts models discussed here.

DOI: 10.1103/PhysRevE.100.032128

I. INTRODUCTION

Thanks to important technological advances, exemplified, in particular, by next-generation sequencing, biology is currently undergoing a deep transformation towards a data-rich science. As an example, the number of available protein sequences deposited in the UNIPROT database was about 10^6 in 2004, crossed 10×10^6 in 2010, and 100×10^6 in 2018, despite an important reorganization of the database in 2015 to reduce redundancies and, thus, limit the database size [1]. On the contrary, proteins with detailed experimental knowledge are contained in the manually annotated SWISSPROT subdatabase of UNIPROT. Although their number remained almost constant and close to 500 000 over the past decade, the knowledge about these selected proteins has been continuously extended and updated.

This rapidly growing wealth of data is presenting both a challenge and an opportunity for data-driven modeling approaches. It is a challenge because for less than 0.5% of all known protein sequences, at least, some knowledge going beyond sequence is available. Applicability of standard supervised machine-learning approaches is, thus, frequently limited. However, more importantly, it is an opportunity since protein-sequence databases, such as UNIPROT, are not large sets of unrelated random sequences but contain structured functional proteins resulting from natural evolution.

In particular, protein sequences can be classified into so-called *homologous protein families* [2]. Each family contains protein sequences, which are believed to share common ancestry in evolution. Such homologous sequences typically show very similar three-dimensional folded structures and closely related biological functions. Put simply, they can be seen as equivalent proteins in different species or in different pathways of the same species. Despite this high level of structural and functional conservation, homologous proteins may differ in more than 70–80% of their amino acids. Detecting homology between a currently uncharacterized protein and a well-studied one [3,4] is, therefore, the most important means for computational sequence annotation, including protein-structure prediction by homology modeling [5,6].

To go beyond such knowledge transfer, we can explore the observable sequence variability between homologous proteins since it contains its own important information about the evolutionary constraints acting on proteins to conserve their structure and function [7]. Typically, very few random mutations do actually destabilize proteins or interrupt their function. Some positions need to be highly conserved, whereas others are permissive for multiple mutations. Observing sequence variability across entire homologous protein families, and relating them to protein structure, function, and evolution, is, therefore, an important task [8].

Over the past years, *inverse statistical physics* [9] has played an increasing role in solving this task. Methods, such as direct-coupling analysis (DCA) [10,11] or related approaches [12,13] allow for predicting protein structure [14,15], mutational effects [16–18], and protein-protein interactions [19]. However, many of these methods depend on huge numbers of typically *ad hoc* introduced parameters, making these methods data hungry and susceptible to overfitting effects.

In this paper, we describe an attempt to substantially reduce the amount of parameters and to select them systematically

using sequence data. Despite this parameter reduction, we aim at so-called *generative* statistical models: Samples drawn from these models should be statistically similar to the real data, even if similarity is evaluated using statistical measures, which were not used to infer the model from data.

To this aim, we first review in Sec. II some important points about protein-sequence data, maximum-entropy (MaxEnt) models of these data, in general, profile, and DCA models, in particular. In Sec. III, we introduce a way for rational selection of so-called sequence motifs, which generalizes maximum-entropy modeling. The resulting Hopfield-Potts models are mapped to restricted Boltzmann machines (RBMs) (recently introduced independently for proteins in Ref. [20]) in Sec. IV to enable efficient model inference and interpretation of the model parameters. Section V is dedicated to the application of this scheme to some exemplary protein families. The conclusion and outlook in Sec. VI are followed by some technical Appendices.

II. A SHORT SUMMARY: SEQUENCE FAMILIES, MAXENT MODELS, AND DCA

To put our work into the right context, we need to review shortly some published results about the statistical models of protein families. After introducing the data format, we summarize the maximum-entropy approach typically used to justify the use of Boltzmann distributions for protein families together with some important shortcomings of this approach. Next, we give a concise overview over two different types of maximum-entropy models—profile models and direct-coupling analysis—which are currently used for protein sequences. For all cases, we discuss the strengths and limitations, which have motivated our current paper.

A. Sequence data

Before discussing modeling strategies, we need to properly define what type of data is used. Sequences of homologous proteins are used in the form of *multiple-sequence alignments* (MSAs), i.e., rectangular matrices $(A_i^m)_{i=1,\dots,L}^{m=1,\dots,M}$. Each of the rows $m = 1, \dots, M$ of this matrix contains one aligned protein sequence $\underline{A}^m = (A_1^m, \dots, A_L^m)$ of length L. In the context of MSA, L is also called the alignment width, and M is called its depth. Entries in the matrix come from the alphabet $\mathcal{A} = \{-, A, C, \dots, Y\}$ containing the 20 natural amino acids and the alignment gap "−." Throughout this paper, the size of the alphabet will be denoted by $q = 21$. In practice, we will use a numerical version of the alphabet, denoted by $\{1, \dots, q\}$, but we have to keep in mind that variables are categorical variables, i.e., there is no linear order associated with these numerical values.

The PFAM database [2] currently (release 32.0) lists almost 18 000 protein families. Statistical modeling is most successful for large families, which contain between 10^3 and 10^6 sequences. Typical lengths span the range of $L = 30$–500.

B. Maximum-entropy modeling

The aim of statistical modeling is to represent each protein family by a function $P(\underline{A})$, which assigns a probability to each sequence $\underline{A} \in \mathcal{A}^L$, i.e., to each sequence formed by L letters from the amino acid alphabet $\mathcal{A}$. Obviously, the number of sequences, even in the largest MSA, is much smaller than the number $q^L - 1$ of *a priori* independent parameters characterizing P. So we have to use clever parametrizations for these models.

A commonly used strategy is based on the MaxEnt approach [21]. It starts from any number p of observables,

$$O^\mu : \mathcal{A}^L \to \mathbb{R}, \quad \mu = 1, \dots, p, \tag{1}$$

which assign real numbers to each sequence. Only the values of these observables for the sequences in the MSA $(\underline{A}^m)$ go into the MaxEnt models. More precisely, we require the model to reproduce the empirical mean of each observable over the data,

$$\forall\, \mu = 1, \dots p: \sum_{\underline{A} \in \mathcal{A}^L} P(\underline{A}) O^\mu(\underline{A}) = \frac{1}{M} \sum_{m=1}^{M} O^\mu(\underline{A}^m). \tag{2}$$

In a more compact notation, we write $\langle O^\mu \rangle_P = \langle O^\mu \rangle_{\mathrm{MSA}}$. Besides this consistency with the data, the model should be as unconstrained as possible. Its entropy has, therefore, to be maximized

$$-\sum_{\underline{A} \in \mathcal{A}^L} P(\underline{A}) \ln P(\underline{A}) \longrightarrow \max . \tag{3}$$

Imposing the constraints in Eq. (2) via Lagrange multipliers λ_μ, $\mu = 1, \dots, p$, we immediately find that $P(\underline{A})$ assumes a Boltzmann-like exponential form

$$P(\underline{A}) = \frac{1}{Z} \exp\left\{ \sum_{\mu=1}^{p} \lambda_\mu O^\mu(\underline{A}) \right\} . \tag{4}$$

Model inference consists in fitting the Lagrange multipliers such that Eqs. (2) are satisfied. The partition function Z guarantees normalization of P.

MaxEnt relates observables and the analytical form of the probability distribution, but it does not provide any rule on how to select observables. Frequently, prior knowledge is used to decide which observables are important and which are not. More systematic approaches, therefore, have to address, at least, the following two questions:

(1) Are the selected observables *sufficient*? In the best case, model P becomes *generative*, i.e., sequences $\underline{A}$ sampled from P are statistically indistinguishable from the natural sequences in the MSA $(\underline{A}^m)$ used for model learning. Although this is hard to test in full generality, we can select observables *not* used in the construction of the model and check if their averages in the model and over the input data coincide.

(2) Are the selected observables *necessary*? Would it be possible to construct a parameter-reduced, thus, more parsimonious, model of same quality? This question is very important due to, at least, two reasons: (a) The most parsimonious model would allow for identifying a minimal set of evolutionary constraints acting on proteins and, thus, offer deep insight into protein evolution; and (b) a reduced number of parameters would allow to reduce overfitting effects, which result from the limited availability of data ($M \ll q^L$).

Although there has been promising progress in the first question, cf. the next two subsections, our paper attempts

to approach both questions simultaneously thereby going beyond standard MaxEnt modeling.

To facilitate the further discussion, two important technical points have to be mentioned. First, MaxEnt leads to a family of so-called exponential models where the exponent in Eq. (4) is *linear* in the Lagrange multipliers λ_μ, which parametrize the family. Second, MaxEnt is intimately related to maximum likelihood. When we postulate Eq. (4) for the mathematical form of model $P(\underline{A})$, and when we maximize the logarithmic likelihood,

$$\mathcal{L}\big[\{\lambda_\mu\}|\big(A_i^m\big)\big] = \sum_{m=1}^{M} \ln\, P(\underline{A}^m), \tag{5}$$

with respect to the parameters λ_μ, $\mu = 1, \ldots, P$, we rediscover Eqs. (2) as the stationarity condition. The particular form of $P(\underline{A})$ guarantees that the likelihood is convex, having only a unique maximum.

C. Profile models

The most successful approaches in statistical modeling of biological sequences are probably *profile models* [22], which consider each MSA column (i.e., each position in the sequence) independently. The corresponding observables are simply $O^{ia}(\underline{A}) = \delta_{A_i,a}$ for all positions $i = 1, \ldots, L$ and all amino acid letters $a \in \mathcal{A}$ with δ being the standard Kronecker symbol. These observables, thus, just ask, if in a sequence $\underline{A}$, amino acid a is present in position i. Their statistics in the MSA is, thus, characterized by the fraction,

$$f_i(a) = \frac{1}{M} \sum_{m=1}^{M} \delta_{A_i^m,a}, \tag{6}$$

of sequences having amino acid a in position i. Consistency of model and data requires marginal single-site distributions of P to coincide with f_i,

$$\forall\, i = 1, \ldots, L, \quad \forall A_i \in \mathcal{A}\colon \sum_{\{A_j|j\neq i\}} P(\underline{A}) = f_i(A_i). \tag{7}$$

The MaxEnt model results as $P(\underline{A}) = \prod_{i=1}^{L} f_i(A_i)$, which can be written as a factorized Boltzmann distribution,

$$P(\underline{A}) = \frac{1}{Z} \exp\left\{\sum_i h_i(A_i)\right\}, \tag{8}$$

where the local fields equal $h_i(a) = \ln\, f_i(a)$. Pseudocounts or regularization can be used to avoid infinite negative parameters for amino acids, which are not observed in some MSA column.

Profile models reproduce the so-called *conservation* statistics of a MSA, i.e., the heterogenous usage of amino acids in the different positions of the sequence. Conservation of a single or few amino acids in a column of the MSA is typically an indication of an important functional or structural role of that position. Profile models, frequently in their generalization to profile hidden Markov models [3,4,23], are used for detecting homology of new sequences to protein families, for aligning multiple sequences, and—using the conserved structural and functional characteristics of protein families—indirectly for the computational annotation of experimentally uncharacterized amino acid sequences. They are, in fact, at the methodological basis of the generation of the MSA used here.

Despite their importance in biological sequence analysis, profile models are not generative. Biological sequences show significant correlation in the usage of amino acids in different positions, which are said to *coevolve* [7]. Due to their factorized nature, profile models are not able to reproduce these correlations, and larger sets of observables have to be used to obtain potentially generative sequence models.

D. Direct-coupling analysis

The DCA [10,11], therefore, includes also pairwise correlations into the modeling. The statistical model $P(\underline{A})$ is not only required to reproduce the amino acid usage of single MSA columns, but also required to reproduce the fraction $f_{ij}(a, b)$ of sequences having, simultaneously, amino acid a in position i and amino acid b in position j for all $a, b \in \mathcal{A}$ and all $1 \leqslant i < j \leqslant L$,

$$\begin{aligned} f_{ij}(a,b) &= \frac{1}{M} \sum_{m=1}^{M} \delta_{A_i^m,a} \delta_{A_j^m,b} \\ &= \sum_{\underline{A}\in\mathcal{A}^L} P(\underline{A}) \delta_{A_i,a} \delta_{A_j,b}. \end{aligned} \tag{9}$$

The corresponding observables $\delta_{A_i,a}\delta_{A_j,b}$ are, thus, products of pairs of observables used in profile models.

According to the general MaxEnt scheme described before, DCA leads to a generalized q-state Potts model,

$$P(\underline{A}) = \frac{1}{Z} \exp\left\{\sum_{i<j} J_{ij}(A_i, A_j) + \sum_i h_i(A_i)\right\}, \tag{10}$$

with heterogeneous pairwise couplings $J_{ij}(a, b)$ and local fields $h_i(a)$. The inference of parameters becomes computationally hard since the computation of the marginal distributions in Eq. (9) requires to sum over $O(q^L)$ sequences. Many approximation schemes have been proposed, including message-passing [10], mean-field [11], Gaussian [13,24], and pseudolikelihood maximization [12,25] approximations. DCA and related global inference techniques have found widespread applications in the prediction of protein structures, of protein-protein interactions, and of mutational effects, demonstrating that amino acid covariation as captured by f_{ij} contains biologically valuable information.

Although these approximate inference schemes do not lead to generative models—not even f_i and f_{ij} are accurately reproduced—recently, very precise but time-extensive inference schemes based on Boltzmann-machine learning have been proposed [26–29]. Astonishingly, these models do not only reproduce the fitted one- and two-column statistics of the input MSA, but also reproduce nonfitted characteristics, such as the three-point statistics $f_{ijk}(a, b, c)$ or the clustered organization of sequences in sequence space. These observations strongly suggest that pairwise Potts models as inferred via DCA are generative models, i.e., that the observables used in DCA—amino acid occurrence in single positions and in position pairs—are actually defining a (close to) sufficient statistics. In a seminal experimental work [30], the importance of respecting pairwise correlations in amino acid usage in

generating small artificial but folding protein sequences was shown.

However, DCA uses an enormous amount of parameters. There are independent couplings for each pair of positions and amino acids. In the case of a protein of limited length $L = 200$, the total number of parameters is close to 10^8. Very few of these parameters are interpretable in terms of, e.g., contacts between positions in the three-dimensional protein fold. We would, therefore, expect that not all of these observables are really important to statistically model protein sequences. On the contrary, given the limited size ($M = 10^3$–10^4) of most input MSAs, the large number of parameters makes overfitting likely, and quite strong regularization is needed. It would, therefore, be important to devise parameter-reduced models as proposed in Ref. [31] but without giving up on the generative character of the inferred statistical models.

III. FROM SEQUENCE MOTIFS TO THE HOPFIELD-POTTS MODEL

Seeing the importance of amino acid conservation in proteins and of profile models in computational sequence analysis, we keep Eqs. (7), which link the single-site marginals of $P(\underline{A})$ directly to the amino acid frequencies $f_i(a)$ in single MSA columns. Furthermore, we assume that the important observables for our protein-sequence ensemble can be represented as so-called *sequence motifs*,

$$O^\mu(\underline{A}) = \sum_i \omega_i^\mu(A_i), \quad \mu = 1, \ldots, p, \tag{11}$$

which are linear additive combinations of single-site terms. In sequence bioinformatics, such sequence motifs are widely used, also under alternative names, such as *position-specific scoring and weight matrices*, cf. Refs. [32,33]. Note that, in difference to the observables introduced before for profile or DCA models, motifs constitute *collective observables* potentially depending on the entire amino acid sequence.

Let us assume for a moment that these motifs, or more specifically the corresponding ω matrices, are known. We will address their selection later. For any model P reproducing the sequence profile, i.e., for any model fulfilling Eqs. (7), also the ensemble average of O^μ is given

$$\sum_{\underline{A}} P(\underline{A}) O^\mu(\underline{A}) = \sum_{i,a} \omega_i^\mu(a) f_i(a). \tag{12}$$

The empirical mean of these observables, therefore, does not contain any further information about the MSA statistics beyond the profile itself. The key step is to consider also the variance, or the second moment,

$$\frac{1}{M} \sum_m [O^\mu(\underline{A}^m)]^2 = \sum_{i,j,a,b} \omega_i^\mu(a) \omega_j^\mu(b) f_{ij}(a,b), \tag{13}$$

as a distinct feature characterizing the sequence variability in the MSA, which has to be reproduced by the statistical model $P(\underline{A})$. This second moment actually depends on combinations of f_{ij}, which were introduced in DCA to account for the correlated amino acid usage in pairs of positions.

The importance of fixing this second moment becomes clear in a very simple example: Consider only two positions $\{1, 2\}$ and two possible letters $\{A, B\}$, which are allowed in these two positions. Let us assume further that these two letters are equiprobable in these two positions, i.e., $f_1(A) = f_1(B) = f_2(A) = f_2(B) = 1/2$. Assume further a single motif to be given by $\omega_1(A) = \omega_2(A) = 1/2$, $\omega_1(B) = \omega_2(B) = -1/2$. In this case, the mean of O equals zero. We further consider two cases:

(1) *Uncorrelated positions:* In this case, all words AA , AB, BA, and BB are equiprobable. The second moment of O, thus, equals $1/2$.

(2) *Correlated positions:* As a strongly correlated example, only the two words AA and BB are allowed. The second moment of O, thus, equals 1.

We conclude that an increased second moment (or variance) of these additive observables with respect to the uncorrelated case corresponds to the preference of combinations of letters or entire words; this is also the reason why they are the called sequence motifs.

Including, therefore, these second moments as conditions into the MaxEnt modeling, our statistical model takes the shape,

$$P\left[\underline{A} \middle| \left\{\lambda_\mu, h_i(a), \omega_i^\mu(a)\right\}\right] = \frac{1}{Z} \exp\left\{ \sum_{\mu=1}^{p} \lambda_\mu \sum_{i,j=1}^{L} \omega_i^\mu(A_i) \omega_j^\mu(A_j) + \sum_{i=1}^{L} h_i(A_i) \right\}, \tag{14}$$

with Lagrange multipliers λ_μ, $\mu = 1, \ldots, p$, imposing means (13) to be reproduced by the model, and $h_i(a)$, $i = 1, \ldots, L$, $a \in \mathcal{A}$, to impose Eqs. (7).

The Hopfield-Potts model: from MaxEnt to sequence-motif selection

As mentioned before, an important limitation of MaxEnt models is that they assume certain observables to be reproduced, but they do not offer any strategy on how these observables have to be selected. In the case of Eq. (14), this accounts, in particular, to optimizing the values of the Lagrange parameters λ_μ to match the ensemble averages over $P(\underline{A})$ with the sample averages Eq. (13) over the input MSA. As mentioned before, this corresponds also to *maximizing the logarithmic likelihood* of these parameters given MSA and the ω matrices describing the motifs,

$$\mathcal{L}\left(\{\lambda_\mu, h_i(a)\} \middle| \left(A_i^m\right), \left\{\omega_i^\mu(a)\right\}\right) = \sum_{m=1}^{M} \ln P\left[\underline{A}^m \middle| \left\{\lambda_\mu, h_i(a), \omega_i^\mu(a)\right\}\right]. \tag{15}$$

The important, even if quite straightforward, *step from MaxEnt modeling to motif selection* is to optimize the likelihood also over the choice of all possible ω matrices as parameters. To remove degeneracies, we absorb the Lagrange multipliers λ_μ into the motif matrix ω^μ, and introduce

$$\xi_i^\mu(a) = \sqrt{\lambda_\mu}\, \omega_i^\mu(a), \quad i = 1, \ldots, L, \quad \mu = 1, \ldots, p, \quad a \in \mathcal{A}\,. \tag{16}$$

The model in Eq. (14), thus, slightly simplifies into

$$P\big(\underline{A}|\{h_i(a),\xi_i^\mu(a)\}\big) = \frac{1}{Z}\exp\left\{\sum_{\mu=1}^{p}\sum_{i,j=1}^{L}\xi_i^\mu(A_i)\xi_j^\mu(A_j)+\sum_{i=1}^{L}h_i(A_i)\right\}, \quad (17)$$

with parameters, which have to be estimated by maximum likelihood,

$$\{\hat{h}_i(a),\hat{\xi}_i^\mu(a)\} = \underset{\{h_i(a),\xi_i^\mu(a)\}}{\operatorname{argmax}}\sum_{m=1}^{M}\ln P\big[\underline{A}^m|\{h_i(a),\xi_i^\mu(a)\}\big]. \quad (18)$$

Our model becomes, therefore, the standard *Hopfield-Potts model*, which has been introduced in Ref. [31] in a mean-field treatment, and the sequence motifs equal, up to the rescaling in Eq. (16), the patterns in the Hopfield-Potts model.

The mean-field treatment of Ref. [31] has both advantages and disadvantages with respect to our present paper: On one hand, the largely analytical mean-field solution allows to relate the Hopfield-Potts patterns ξ^μ to the eigenvectors of the Pearson-correlation matrix of the MSA, and their likelihood contributions to a function of the corresponding eigenvalues. This is, in particular, interesting since not only the eigenvectors corresponding to large eigenvalues were found to contribute—as one might expect from the apparent similarity to principal-component analysis (PCA)—but also the smallest eigenvalues lead to large likelihood contributions. However, the mean-field treatment leads to a nongenerative model, which does not even reproduce precisely the single-position frequencies $f_i(a)$. The aim of this paper is to reestablish the generative character of the Hopfield-Potts model by more accurate interference schemes without losing too much of the interpretability of the mean-field approximation.

The model in Eq. (17) contains now an exponent, which is nonlinear in the parameters ξ^μ. As a consequence, the likelihood is not convex anymore, and possibly many local likelihood maxima exist. This is also reflected by the fact that any p-dimensional orthogonal transformation of ξ^μ leaves the probability distribution $P(\underline{A})$ invariant, thus, leading to an equivalent model.

IV. INFERENCE AND INTERPRETATION OF HOPFIELD-POTTS MODELS

A. The Hopfield-Potts model as a restricted Boltzmann machine

The question how many and which patterns are needed for generative modeling, therefore, cannot be answered properly within the mean-field approach. We, therefore, propose a more accurate inference scheme based on RBM learning [34,35], exploiting an equivalence between Hopfield models and RBM originally shown in Ref. [36]. To this aim, we first perform p Hubbard-Stratonovich transformations to linearize the exponential in ξ^μ,

$$P(\underline{A}) = \frac{1}{\tilde{Z}}\int_{\mathbb{R}^p}\prod_{\mu=1}^{p}dx^\mu\exp\left\{\sum_{i,\mu}x^\mu\xi_i^\mu(A_i)+\sum_i h_i(A_i)-\frac{1}{2}\sum_\mu(x^\mu)^2\right\}, \quad (19)$$

with $\tilde{Z}$ containing the normalizations both of the Gaussian integrals over the new variables x^μ and the partition function of Eq. (14). The distribution $P(\underline{A})$ can, thus, be understood as a marginal distribution of

$$P(\underline{A},\underline{x}) = \frac{1}{\tilde{Z}}\exp\left\{\sum_{i,\mu}x^\mu\xi_i^\mu(A_i)+\sum_i h_i(A_i)-\frac{1}{2}\sum_\mu(x^\mu)^2\right\}, \quad (20)$$

which depends on the so-called *visible variables* $\underline{A}=(A_1,\ldots,A_L)$ and the *hidden (or latent) variables* $\underline{x}=(x^1,\ldots,x^p)$. It takes the form or a particular RBM with a quadratic confining potential for x^μ: The important point is that couplings in the RBM form a bipartite graph between visible and hidden variables, cf. Fig. 1. RBM may have more general potentials $u_\mu(x^\mu)$ confining the values of the new random variables x^μ. This fact has been exploited in Ref. [20] to cope with the limited number of sequences in the training MSA. However, in our paper, we stick to quadratic potentials in order to keep the equivalence to Hopfield-Potts models, and thus, the interpretability of patterns in terms of pairwise residue-residue couplings via Eq. (17).

B. Parameter learning by persistent contrastive divergence

Maximizing the likelihood with respect to the parameters leads, for our RBM model, to the stationarity equations,

$$\frac{1}{M}\sum_m\delta_{A_i^m,a} = \big\langle\delta_{A_i,a}\big\rangle_{P(\underline{A},\underline{x})},$$
$$\frac{1}{M}\sum_m\delta_{A_i^m,a}\langle x^\mu\rangle_{P(\underline{x}|\underline{A}^m)} = \big\langle\delta_{A_i,a}x^\mu\big\rangle_{P(\underline{A},\underline{x})} \quad (21)$$

for all i, a, and μ; the difference of both sides equals the gradient of the likelihood in the direction of the corresponding parameter. Although the first line matches the standard MaxEnt form—sample and ensemble average of an observable have to coincide, the second line contains a mixed sample-ensemble average on its left-hand side. Since the variables x^μ are latent and, thus, not contained in the MSA, an average over their probability $P(\underline{x}|\underline{A}^m)$ conditioned to the sequences $\underline{A}^m$ in the MSA has to be taken. Having a P dependence on both sides of Eqs. (21) is yet another expression of the nonconvexity of the likelihood function.

Model parameters $h_i(a)$ and $\xi_i^\mu(a)$ have to be fitted to satisfy the stationarity conditions Eq. (21). This can be performed iteratively: Starting from arbitrarily initialized model parameters, we determine the difference between the left- and the right-hand sides of this equation and use this difference to update parameters (i.e., we perform gradient ascent of the likelihood); each of these update steps is called an *epoch* of learning. A major problem is that the exact calculation of averages over the $(L+p)$-dimensional probability distribution P is computationally infeasible. It is possible to estimate these averages by Markov chain Monte Carlo (MCMC) sampling, but efficient implementations are needed since accurate parameter learning requires, in practice, thousands of epochs. To this aim, we exploit the bipartite structure of RBM: Both conditional probabilities $P(\underline{A}|\underline{x})$ and $P(\underline{x}|\underline{A})$ are factorized.

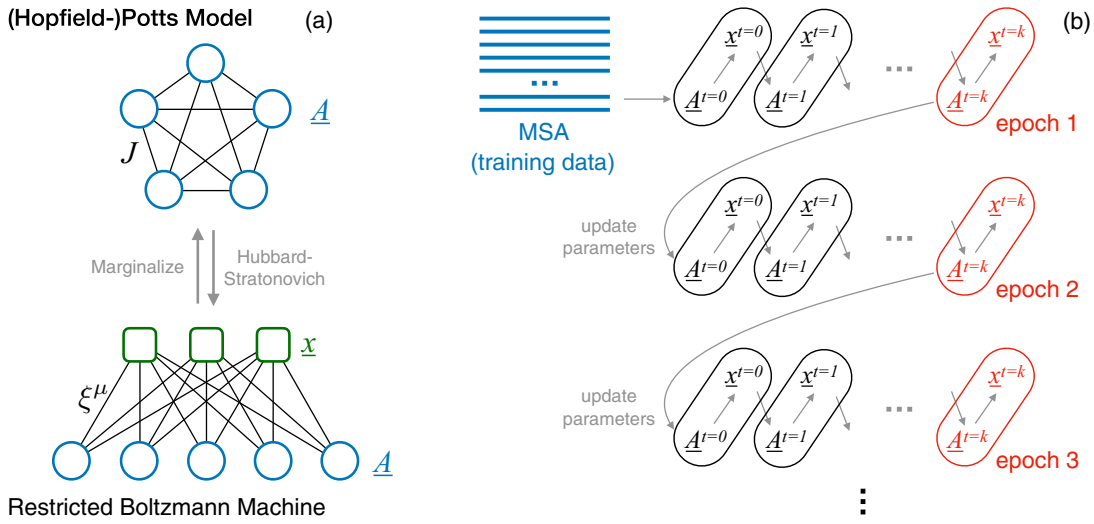

FIG. 1. Panel (a) represents the (Hopfield-)Potts model as a statistical model for sequences $\underline{A} \in \mathcal{A}^L$, typically characterized by a fully connected coupling matrix J and local fields h (not represented). The model can be transformed into a RBM by introducing Gaussian hidden variables $\underline{x} \in \mathbb{R}^p$ with p being the rank of J. Note the bipartite graphical structure of RBM, which causes the conditional probabilities $P(\underline{A}|\underline{x})$ and $P(\underline{x}|\underline{A})$ to factorize. Panel (b) shows a schematic of persistent contrastive divergence (PCD). Initially, the sample is initialized in the training data (the MSA of natural sequences), and then, k alternating steps of sampling from $P(\underline{A}|\underline{x})$, respectively, $P(\underline{x}|\underline{A})$'s are performed. Parameters are updated after these k sampling steps, and sampling is continued using the updated parameters.

This allows us to initialize MCMC runs in natural sequences from the MSA and to sample $\underline{x}$ and $\underline{A}$ in alternating fashion. As a second simplification, we use PCD [37]. Only in the first epoch the visible variables are initialized in the MSA sequences, and each epoch performs only a finite number of sampling steps (k for PCD k), cf. Fig. 1(b). Trajectories are continued in a new epoch after parameter updates. If the resulting parameter changes become small enough, PCD will thereby generate close-to-equilibrium sequences, which form an (almost) independent and identically distributed (i.i.d.) sample of $P(\underline{A}, \underline{x})$ uncorrelated from the training set used for initialization.

Details of the algorithm and comparison to the simpler contrastive divergence are given in Appendix B. Further technical details, such as regularization, are also delegated to Appendix B.

C. Determining the likelihood contribution of single Hopfield-Potts patterns

It is obvious that the total likelihood grows monotonously when increasing the number p of patterns ξ^μ. It is, therefore, important to develop criteria, which tell us if patterns are more or less important for modeling the protein family. To this aim, we estimate the contribution of single patterns to the likelihood by comparing the full model with a model where a single pattern ξ^μ has been removed, whereas the other $p-1$ patterns and the local fields have been retained. The corresponding normalized change in logarithmic likelihood reads

$$\Delta\ell_\mu = \frac{1}{M}\sum_{m=1}^{M}[\ln P(\underline{A}^m) - \ln P_{-\mu}(\underline{A}^m)], \tag{22}$$

where $P_{-\mu}$ has the same form as given in Eq. (14) for P but with pattern $\xi^\mu = \{\xi_i^\mu(a);\ i = 1, \ldots, L,\ a \in \mathcal{A}\}$ removed. Plugging Eq. (14) into Eq. (22), we find

$$\Delta\ell_\mu = \frac{1}{M}\sum_{m=1}^{M}\left[\sum_{i=1}^{L}\xi_i^\mu\left(A_i^m\right)\right]^2 + \ln\frac{Z_{-\mu}}{Z}\,. \tag{23}$$

The likelihood difference depends, thus, on the ratio of the two partition functions Z and $Z_{-\mu}$. Although each of them is individually intractable due to the exponential sum over q^L sequences, the ratio can be estimated efficiently using importance sampling. We write

$$\begin{aligned}\frac{Z_{-\mu}}{Z} &= \frac{1}{Z}\sum_{\underline{A}\in\mathcal{A}^L}\exp\left\{\sum_{\nu\neq\mu}\sum_{i,j=1}^{L}\xi_i^\nu(A_i)\xi_j^\nu(A_j) + \sum_{i=1}^{L}h_i(A_i)\right\}\\ &= \sum_{\underline{A}\in\mathcal{A}^L}P(\underline{A})\exp\left\{-\sum_{i,j=1}^{L}\xi_i^\mu(A_i)\xi_j^\mu(A_j)\right\}.\end{aligned} \tag{24}$$

The last expression contains the average of an exponential quantity over $P(\underline{A})$, so estimating the average by MCMC sampling of P might appear a risky idea. However, since P and $P_{-\mu}$ differ only in one of the p patterns, the distributions are expected to overlap strongly, and sufficiently large samples drawn from $P(\underline{A})$ can be used to estimate $Z_{-\mu}/Z$. Note that sampling is performed from P, so the likelihood contributions of all patterns can be estimated in parallel using a single large sample of the full model.

Once these likelihood contributions are estimated, we can sort them, and identify and interpret the patterns of largest importance in our Hopfield-Potts model.

V. HOPFIELD-POTTS MODELS OF PROTEIN FAMILIES

To understand the performance of Hopfield-Potts models in the case of protein families, we have analyzed three protein families extracted from the PFAM database [2]: the Kunitz-bovine pancreatic trypsin inhibitor domain (PF00014), the response regulator receiver domain (PF00072), and the RNA recognition motif (PF00076). They have been selected since they have been used in DCA studies before; in our case, RBM results will be compared to the ones of BMDCA, i.e., the generative version of DCA based on Boltzmann machine learning [29]. MSAs are downloaded from the PFAM database [2], and sequences with more than five consecutive gaps are removed; cf. Appendix B for a discussion of the convergence problems of PCD-based inference in the case of extended gap stretches. The resulting MSA dimensions for the three families are, in the order given before, $L = 52/112/70$ and $M = 10\,657/15\,000/10\,000$. As can be noted, the last two MSAs have been subsampled randomly since they were very large, and the running time of the PCD algorithm is linear in the sample size. The MSA for PF00072 was chosen to be slightly larger because of the longer sequences in this family.

In the following sections, results are described in detail for the PF00072 response regulator family. The results for the other protein families are coherent with the discussion; they are moved to Appendix B to seek the conciseness of our presentation.

A. Generative properties of Hopfield-Potts models

PCD is able, for all values of the pattern number p, to reach parameter values satisfying the stationarity conditions Eqs. (21). This is not only true when these are evaluated using the PCD sample propagated via learning from epoch to epoch, but also when the inferred model is resampled using MCMC, i.e., when the right-hand side of Eqs. (21) is evaluated using an i.i.d. sample of the RBM.

In the leftmost column of Fig. 2 [panels (a.1)–(g.1)], this is shown for the single-site frequencies, i.e., for the first of Eqs. (21). The horizontal axis shows the statistics extracted from the original data collected in the MSA, whereas the vertical axis measures the same quantity in an i.i.d. sample extracted from the inferred model $P(\underline{A}, \underline{x})$. The fitting quality is comparable to the one obtained by BMDCA as can be seen by comparison with the last panel in the first column of Fig. 2.

The other two columns of the figure concern the *generative* properties of RBM: connected two-point correlations [panels (a.2)–(g.2) in Fig. 2] and three-point correlations [panels (a.3)–(g.3) in Fig. 2],

$$\begin{aligned} c_{ij}(a,b) &= f_{ij}(a,b) - f_i(a) f_j(b), \\ c_{ijk}(a,b,c) &= f_{ijk}(a,b,c) - f_{ij}(a,b) f_k(c) - f_{ik}(a,c) f_j(b) \\ &\quad - f_{jk}(b,c) f_i(a) + 2 f_i(a) f_j(b) f_k(c), \end{aligned} \tag{25}$$

with the three-point frequencies $f_{ijk}(a,b,c)$ defined in analogy to Eqs. (6) and (9). Note that, in difference to DCA, already the two-point correlations are not fitted directly by the RBM but only the second moments related to the Hopfield-Potts patterns. This becomes immediately obvious for the case of $p = 0$ where RBM reduces to simple profile models of statistically independent sites but remains true for all values of $p < (q-1)L$. Note also that connected correlations are used since the frequencies f_{ij} and f_{ijk} contain information about the fitted f_i and, therefore, show stronger agreement between data and model.

The performance of RBM is found to be, up to statistical fluctuations, monotonous in the pattern number p. As in the mean-field approximation [31], no evident overfitting effects are observed. Even if not fitted explicitly, as few as $p = 20$–40 patterns are sufficient to faithfully reproduce even the nonfitted two- and three-point correlations. This is very astonishing since only about 1.7–3.5% of the parameters of the full DCA model are used: The p patterns are given by $p(q-1)L$ parameters, whereas DCA has $(q-1)^2\binom{L}{2}$ independently inferred couplings. The times needed for accurate inference decrease accordingly: In some cases, a slight decrease in accuracy of BMDCA is observed as compared to RBM with the largest p; this could be overcome by iterating the inference procedure for further epochs.

B. Strong couplings and contact prediction

One of the main applications of DCA is the prediction of contacts between residues in the three-dimensional protein fold, based only on the statistics of homologous sequences. To this aim, we follow Ref. [25] and translate $q \times q$ coupling matrices $J_{ij}(a,b) = \sum_{\mu=1}^{p} \xi_i^\mu(a) \xi_j^\mu(b)$ for individual site pairs (i, j) into scalar numbers by first calculating their Frobenius norm,

$$F_{ij} = F_{ji} = \sum_{a,b \in \mathcal{A}} J_{ij}(a,b)^2, \tag{26}$$

followed by the empirical average-product correction (APC),

$$F_{ij}^{\mathrm{APC}} = F_{ij} - \frac{F_{i\cdot} F_{\cdot j}}{F_{\cdot\cdot}}, \tag{27}$$

where the $\cdot$ denotes an average over the corresponding index,

$$\begin{aligned} F_{i\cdot} &= \frac{1}{L-1} \sum_k F_{ik}, \\ F_{\cdot j} &= \frac{1}{L-1} \sum_k F_{kj}, \\ F_{\cdot\cdot} &= \frac{2}{L(L-1)} \sum_{k<l} F_{kl}. \end{aligned} \tag{28}$$

The APC is intended to remove systematic nonfunctional bias due to conservation and phylogeny. These quantities are sorted, and the largest ones are expected to be contacts.

The results for several values of p and for BMDCA are depicted in Fig. 3(a): The PPV is the fraction of true positives (TPs) among the first n predictions as a function of n. TPs are defined as native contacts in a reference protein structure (PDB ID 3ilh [38] for PF00072) with a distance cutoff of 8 Å between the closest pair of heavy atoms forming each residue. Pairs in the vicinity along the peptide chain are not considered in this prediction since they are trivially in contact: In coherence with the literature standard, Fig. 3 only considers predictions with $|i-j| \geqslant 5$.

Despite the fact that, even for as few as $p = 20$–40 patterns, the model appears to be generative, i.e., nonfitted statistical

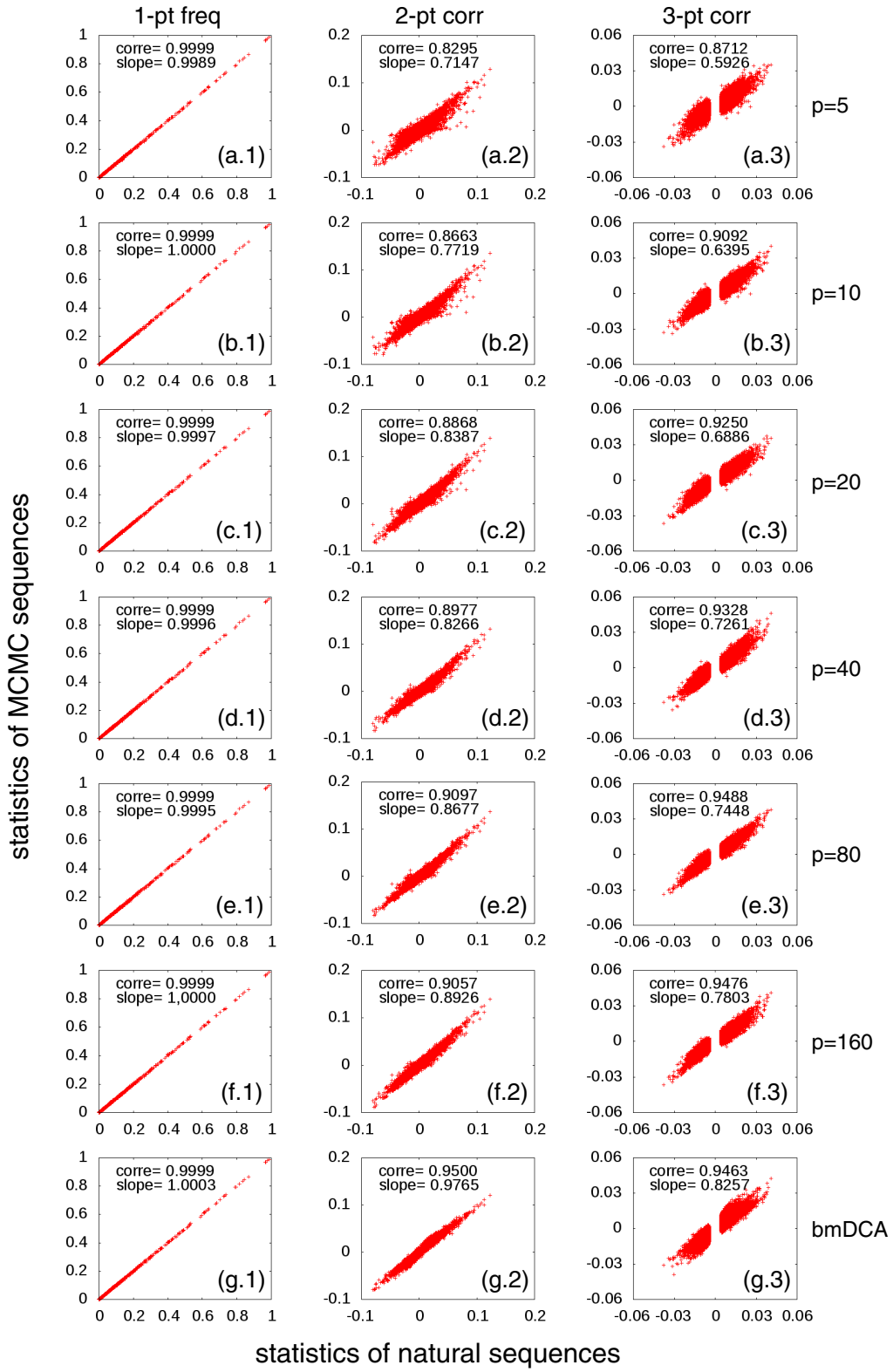

FIG. 2. Statistics of natural sequences (PF00072, horizontal axes) vs MCMC samples (vertical axes) of Hopfield-Potts models for values of $p \in \{5, 10, 20, 40, 80, 160\}$ and for a full-rank Potts model inferred using BMDCA. The first column [panels (a.1)–(g.1)] shows the one-point frequencies $f_i(a)$ for all pairs (i, a) of sites and amino acids; the other two columns show the connected two- and three-point functions $c_{ij}(a, b)$ [panels (a.2)–(g.2)] and $c_{ijk}(a, b, c)$ [panels (a.3)–(g.3)]. Due to the huge number of combinations for the three-point correlations, only the 100 000 largest values (evaluated in the training MSA) are shown. The Pearson correlations and the slope of the best linear fit are inserted in each of the panels.

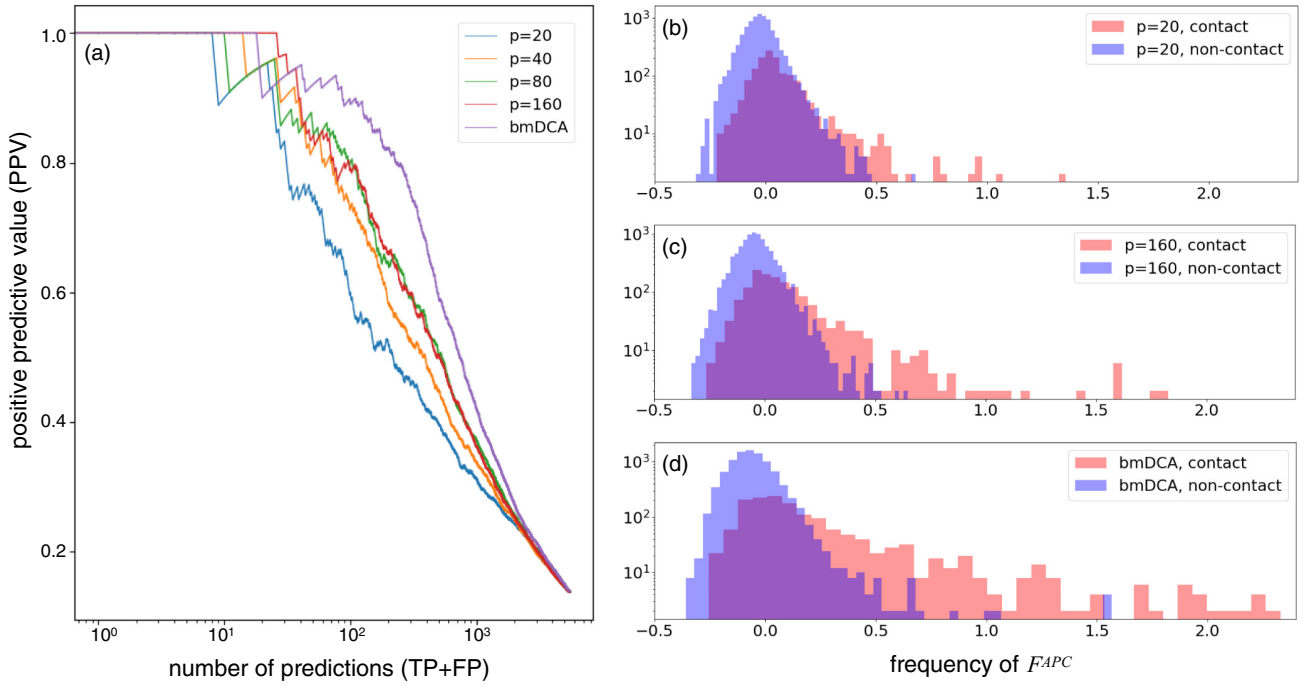

FIG. 3. Panel (a) shows the positive predictive value (PPV) for contact prediction as a function of the number of predictions for various values of the pattern number p and for BMDCA. Panels (b)–(d) show, for $p = 20$, 160, and BMDCA, the distribution of coupling scores F_{ij}^{APC}. All residue pairs are grouped into contacts (red) and noncontacts (blue). The best contact predictions correspond to the positive tail of the red histogram, which becomes more pronounced when increasing p or even going to BMDCA.

observables are reproduced with good accuracy, the PPV curves depend strongly on the pattern number p. Up to statistically probably insignificant exceptions, we observe a monotonous dependence on p, and none of the RBM-related curves reach the performance of the full-rank J_{ij} matrices of BMDCA. Even large values of p where RBM have more than 30% of the parameters of the full Potts model show a drop in performance in contact prediction.

Can we understand this apparent contradiction: similarly accurate reproduction of the statistics but reduced performance in contact prediction? To this end, we consider, in Figs. 3(b)–3(d), the histograms of coupling strengths F_{ij}^{APC} divided into two subpopulations: Values for sites i, j in contact are represented by red, and values for distant sites are represented by blue histograms. It becomes evident that the rather compact histogram of noncontacts remains almost invariable with p (even if individual coupling values do change), but the histogram of contacts changes systematically: The tail of large F_{ij}^{APC} going beyond the upper edge of the blue histogram is less pronounced for small p. However, in the procedure described before, these F_{ij}^{APC} values provide the first contact predictions.

The reduced capacity to detect contacts for small p is related to the properties of the Hopfield-Potts model in itself. Although the residue-residue contacts form a sparse graph, the Hopfield-Potts model is explicitly constructed to have a low-rank coupling matrix $[J_{ij}(a, b)]$. It is, however, hard to represent a generic sparse matrix by a limited number of possibly distributed patterns. Hopfield-Potts models are more likely to detect distributed sequence signals than localized sparse ones. However, for larger pattern numbers p, we are able to detect more and more localized signals thereby improving the contact prediction until BM and Hopfield-Potts models become equivalent for $p = (q - 1)L$.

This observation establishes an important limitation to the generative character of Hopfield-Potts models with limited pattern numbers: The applicability of DCA for residue-residue contact prediction has demonstrated that physical contacts in the three-dimensional structure of proteins introduce important constraints on sequence evolution. A perfectly generative model should respect these constraints and, thus, lead to a contact prediction being, at least, as good as the one obtained by full DCA, cf. also the discussion in the outlook of this article.

C. Likelihood contribution and interpretation of selected sequence motifs

So what do the patterns represent? In Sec. IV C, we have discussed how to estimate the likelihood contribution of patterns thereby being able to select the most important patterns in our model. Figure 4 displays the ordered contributions for different values of p. We observe that, for small p, the distribution becomes more peaked with few patterns having very large likelihood contributions. For larger p, the contributions are more distributed over many patterns, which collectively represent the statistical features of the data set.

Figure 5 represents the first five patterns for $p = 20$. Panels (a.1)–(e.1) of Fig. 5 represent the pattern $\xi_i^{\mu}(a)$ as a sequence logo, a standard representation in sequence bioinformatics. Each site i corresponds to one position, the possible amino acids are shown by their one-letter codes, the size of the

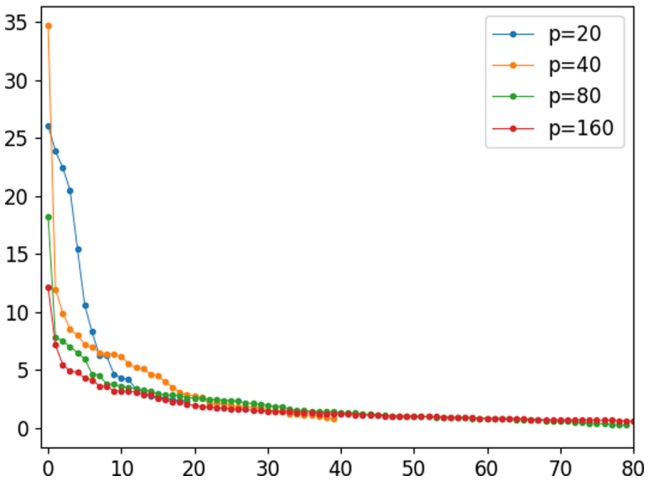

FIG. 4. Likelihood contribution of the individual patterns for pattern numbers $p = 20,\ 40,\ 80$, and 160.

letter being proportional to $|\xi_i^\mu(a)|$, according to the sign of $\xi_i^\mu(a)$, letters are represented above or below the zero line. The alignment gap is represented as a minus sign in an oval shape, which allows to represent its size in the current pattern.

Patterns are very distributed, both in terms of the sites and in the amino acids with relatively large entries $\xi_i(a)$. This makes a direct interpretation of patterns without prior knowledge rather complicated. The distributed nature of patterns explains also why they are not optimal in defining localized contact predictions. Rather than identifying contacting residue pairs, the patterns define larger groups of sites, which are connected via a dense network of comparable couplings. However, as we will see in the next section, the sites of large entries in a pattern define functional regions of proteins, which are important in subensembles of proteins of strong (positive or negative) activity values along the pattern under consideration. In particular, we will show that the largest entries may have an interpretation connecting structure and function to sequence in protein subfamilies.

The middle column [panels (a.2)–(e.2)] of Fig. 5 shows a histogram of pattern-specific activities of single sequences, i.e., of

$$x^\mu(\underline{A}) = \sum_{i=1}^{N} \xi_i^\mu(A_i). \tag{29}$$

Note that, up to the rescaling in Eq. (16), these numbers coincide with the sequence motifs, introduced in Eq. (11) at the beginning of this article. They also equal the average value of the latent variable x^μ given sequence $\underline{A}$. The blue histograms result from the natural sequences collected in the training MSA. They coincide well with the red histograms, which are calculated from an i.i.d. MCMC sample of our Hopfield-Potts model, including the bimodal structure of several histograms. This is quite remarkable: The Hopfield-Potts model was derived, in the beginning of this paper, as the maximum-entropy model reproducing the first two moments of the activities $\{x^\mu(\underline{A}^m)\}_{m=1\cdots M}$. Finding higher-order features, such as bimodality, is again an expression of the generative power of Hopfield-Potts models.

Figures (5.a.3)–(5.e.3) prove the importance of individual patterns for the inferred model. The panels show the two-point correlations $c_{ij}(a,b)$ of the natural data (horizontal axis) vs the one of samples drawn from the distributions $P_{-\mu}(\underline{A})$, introduced in Eq. (22) as Hopfield-Potts models of $p-1$ patterns with pattern ξ^μ removed (vertical axis). The coherence of the correlations is strongly reduced when compared to the full model, which was shown in Fig. 2: Removal even of a single pattern has a strong global impact on the model statistics.

D. Sequence clustering

As already mentioned, some patterns show a clear bimodal activity distribution, i.e., they identify two statistically distinct subgroups of sequences. The number of subgroups can be augmented by using more than one pattern, i.e., combinations of patterns can be used to cluster sequences.

To this aim, we have selected three patterns (numbers 6, 13, and 14) with a pronounced bimodal structure from the model with $p = 20$ patterns. In terms of likelihood contribution, they have ranks 8, 4, and 1 in the contributions to the logarithmic likelihood, cf. Fig. 5.

The clustered organization of response-regulator sequences becomes even more evident in the two-dimensional plots characterizing, simultaneously, two activity distributions. The results for all pairs of the three patterns are displayed in Fig. 6, panels (a.1–(a.3). As a first observation, we see that the main modes of the activity patterns give rise to one dominant cluster. Smaller clusters deviate from the dominant one in a single pattern but show compatible activities in the other patterns—the two-dimensional plots, therefore, show typically an L-shaped sequence distribution and three clusters instead of the theoretically possible four combinations of activity models. It appears that single patterns identify the particularities of single subdominant sequence clusters.

We have chosen the response-regulator protein-domain family in this paper also due to the fact that it constitutes a functionally well studied and diversified family. Response regulators are predominantly used in bacterial signaling systems:

(1) In *chemotaxis*, they appear as single-domain proteins named CheY, which transmit the signal from kinase proteins (activated by signal reception) to flagellar motor proteins, which trigger the movement of the bacteria. CheY proteins can be identified in our MSA as those coming from single-domain proteins, i.e., with lengths compatible to the PF00072-MSA width $L = 112$. We have selected a sub-MSA consisting of all proteins with total sequence lengths between 110 and 140 amino acids.

(2) In *two-component signal transduction* (TCS), response regulators are typically transcription factors, which are activated by signal-receiving histidine sensor kinases. The corresponding proteins contain two or three domains, in particular, a DNA-binding domain, which is actually responsible for the transcription-factor activity of the activated response-regulator protein. According to the present DNA-binding domain, these TCS proteins can be subdivided into different classes, the dominant ones are the OmpR, the GerE, and the Sigma54-HTH8 classes, we identified three sub-MSA corresponding to these classes by co-occurrence of the

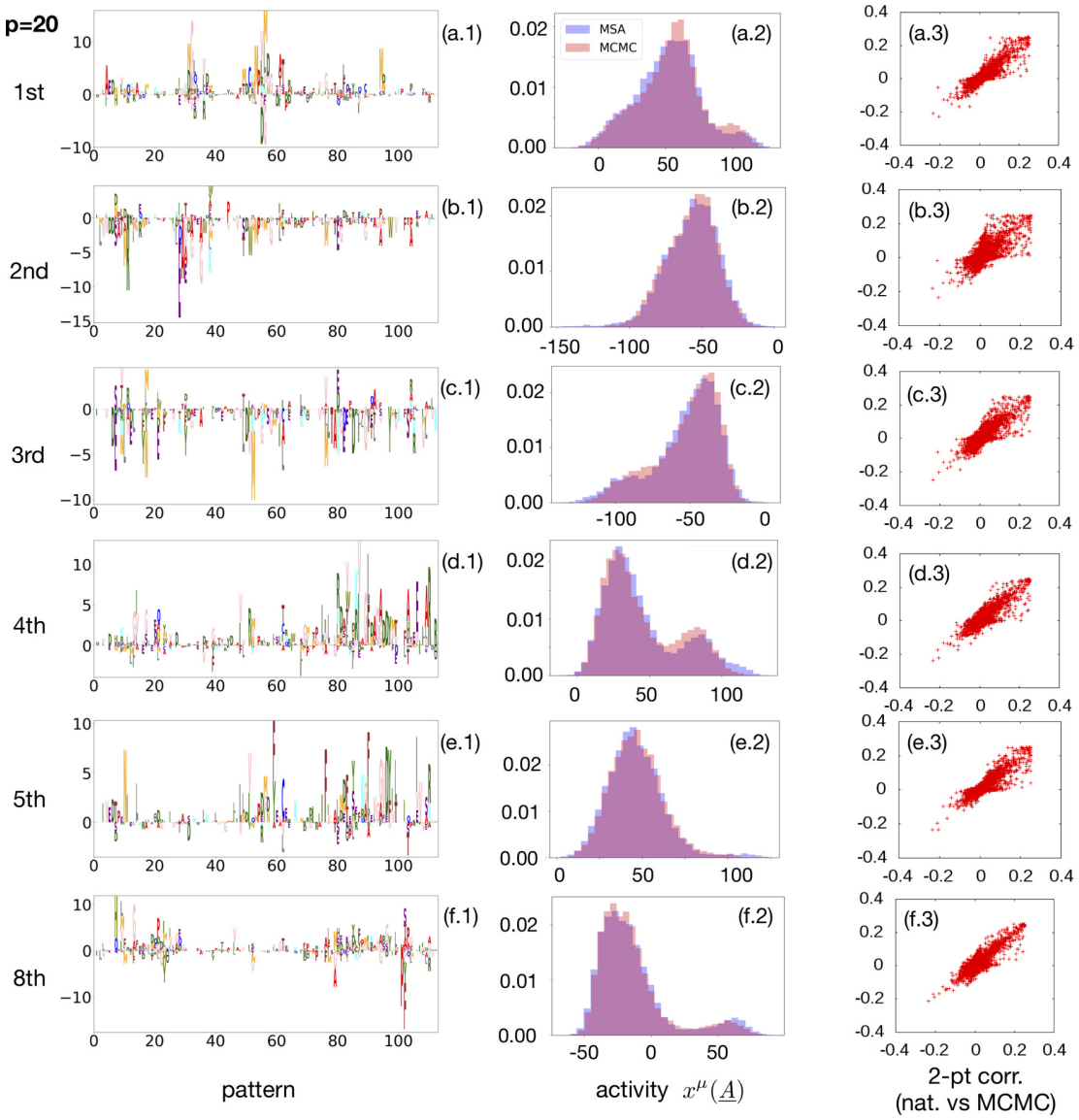

FIG. 5. The five patterns of highest likelihood contribution for $p = 20$, the eighth ranking is added since used later in the text. The left panels (a.1)–(f.1) show the patterns in logo representation, the letter size is given by the corresponding element $\xi_i(a)$. The middle panels (a.1)–(f.2) show the distribution of the activities, i.e., the projections of sequences onto the patterns. The blue histogram contains the natural sequences from the training MSA, and the red histogram contains sequences sampled by MCMC from the Hopfield-Potts model. The right-hand side [panels (a.3)–(f.3)] shows the connected two-point correlations of the natural data (horizontal axis) vs data sampled from $P_{-\mu}(\underline{A})$, i.e., a Hopfield-Potts model with one pattern removed. Strong deviations from the diagonal are evident.

DNA-binding domains with the response-regulator domain in the same protein. The different DNA-binding domains are indicative for distinct homodimer structures assumed by the active transcription factors; DCA run on the sub-MS identifies their specific subfamily interfaces [39].

(3) *Phosphorelays* are similar to TCS but consist of more complex multicomponent signaling pathways. In these systems, found in bacteria and plants, response-regulator domains are typically fused to the histidine-sensor kinases. They do not act as transcription factors but transduct a signal to a phosphotransferase, which finally activates a down-stream transcription factor of the same architecture mentioned in the last paragraph. We identified a class of response-regulator domains, which are fused to a histidine kinase domain. In terms of domain architecture and protein length, this subfamily is extremely heterogenous.

Panel columns (a)–(f) in Fig. 6 show the activities of these five subfamilies. It is evident that distinct sub-MSAs fall actually into distinct clusters according to these three patterns:

(1) The CheY-like single domain proteins [panels (b.1)–(b.3)] of Fig. 6 fall, according to all three patterns, into the dominant mode.

(2) The OmpR-class transcription factors [panels (c.1)–(c.3)] show a distinct distribution of higher activities for the second of the patterns (which actually has the most pronounced bimodal structure, probably due to the fact that the OmpR class forms the largest sub-MSA). As can be seen

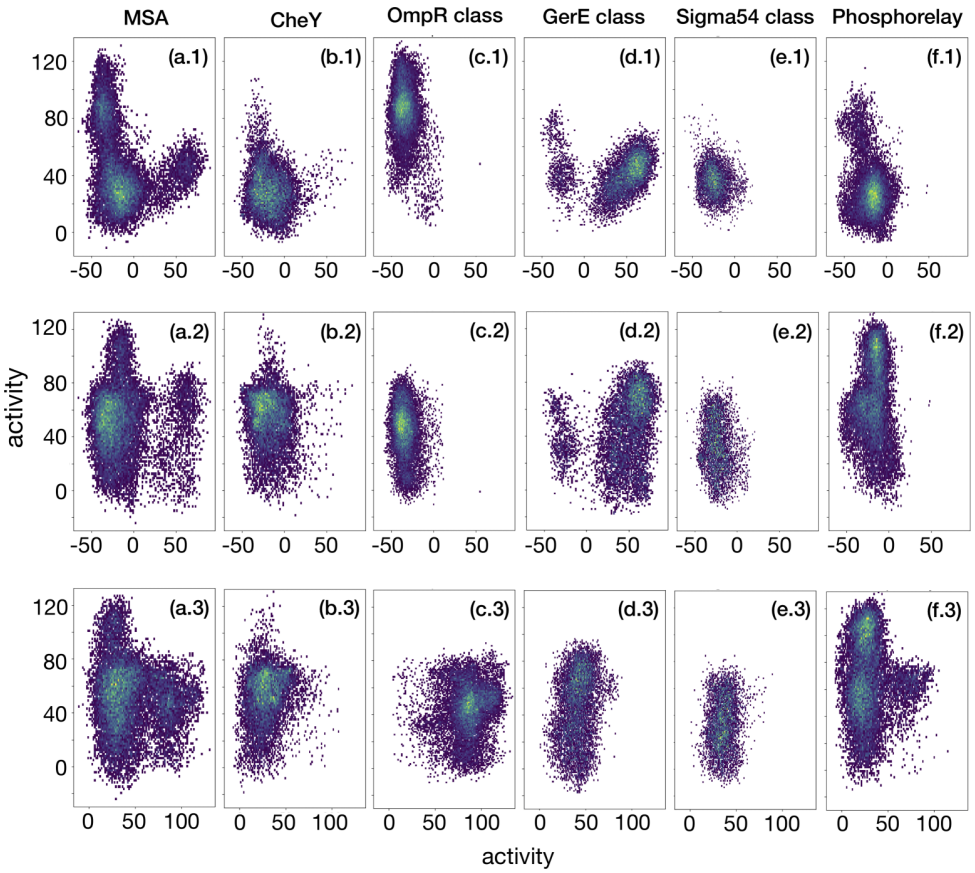

FIG. 6. Patterns with multimodal activity distributions for the set of all MSA sequences can be used to cluster sequences. The rows show combinations of patterns 6–13 [panels (a.1)–(f.1)], 6–14 [panels (a.2)–(f.2)], and 13 to 14 [panels (a.3)–(f.3)]. Each sequence corresponds to a density-colored dot. A strongly clustered structure is clearly visible. When dividing the full MSA into functional subclasses, we can relate clusters to subclasses and, thus, patterns to biological function.

in Fig. 6, this pattern has the largest positive entries in the region of positions 80–90 and 100–110. Interestingly, these regions define the interface of OmpR-class transcription-factor homodimerization, cf. Ref. [39]. In accordance with this structural interpretation, we also find a periodic structure of period 3 to 4 of the large entries in the pattern, which reflects the fact that the interface is formed by two helices, which lead to a periodic exposure of amino acids in the protein surface.

(3) The GerE class [panels (d.1)–(d.3)] of Fig. 6 differs in activities in the direction of the first pattern, only GerE-class proteins have positive, and all others have negative activities. Dominant positive entries are found in regions 5–15 and 100–105, again identifying the homodimerization interface, cf. Ref. [39].

(4) The Sigma54 class [panels (e.1)–(e.3)] does not show a distinct distribution of activities according to the three selected patterns. It is located together with the CheY-type sequences. However, when examining all patterns, we find that pattern number 5 (ranked sixth according to the likelihood contribution) is almost perfectly discriminating the two.

(5) Last but not least, the response regulators fused to histidine kinases in phosphorelay systems [panels (f.1)–(f.3)] of Fig. 6 show a distinct activity distribution according to the third pattern, mixing a part of activities compatible with the main cluster, and others being substantially larger (this mixing results presumably from the previously mentioned heterogenous structure of this sub-MSA). Structurally known complexes between response regulators and histidine phosphotransferases (PDB ID 4euk [40], 1bdj [41]) show the interface located in residues 5–15, 30–32, and 50–55, regions being important in the corresponding pattern. It appears that the pattern selects the particular amino acid composition of this interface, which is specific to the phosphorelay sub-MSA.

These observations do not only show that the patterns allow for clustering sequences into sub-MSA, but also show that the discriminating positions in the patterns have a clear biological interpretation. This is very interesting since the analysis in Ref. [39] required a prior clustering of the initial MSA into sub-MSA, and the application of DCA to the individual sub-MSA. Here, we have inferred only one Hopfield-Potts model describing the full MSA, and the patterns automatically identify biologically reasonable subfamilies together with the sequence patterns characterizing them. The prior knowledge needed in Ref. [39] is not needed here; we use it only for the posterior interpretation of the patterns.

It is also important to remember that sequence clustering can be obtained by a technically simpler PCA. PCA is based

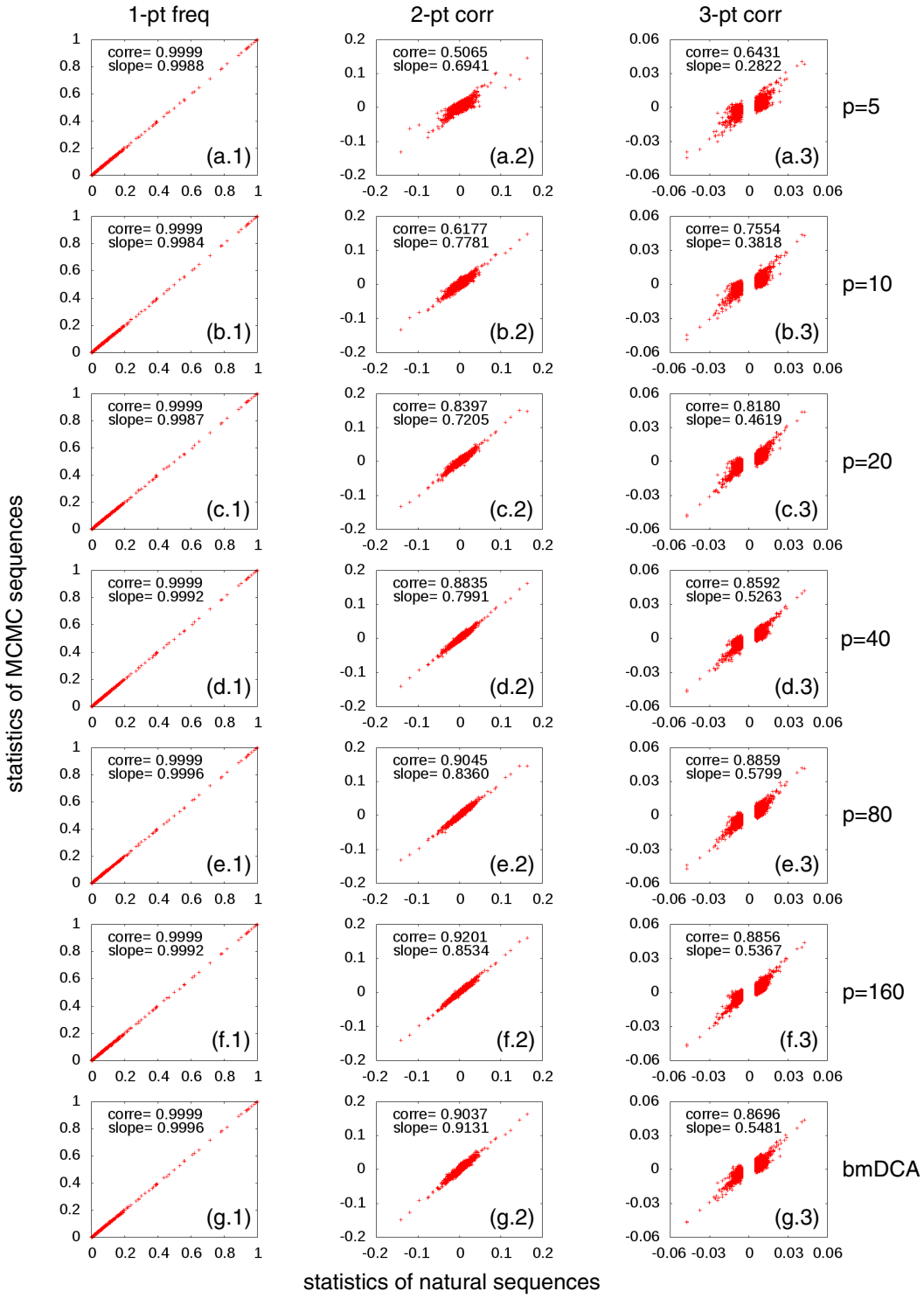

FIG. 7. The same as Fig. 2 but for the protein family PF00014.

on the leading eigenvectors of the data-covariance matrix, i.e., exclusively on the largest eigenvalues. The potential differences were already discussed in Ref. [31] in the context of the mean-field approximation of Hopfield-Potts models. It was shown that not only the eigenvectors with large eigenvalues lead to important contributions in likelihood, but also those corresponding to the smallest eigenvalues. Both tails of the spectrum are, thus, important for the statistical description of protein-sequence ensembles. A second drawback of PCA as compared to our approach is the nongenerative character of PCA. No explicit statistical model is learned, but the data covariance matrix is simply approximated by a low-rank matrix.

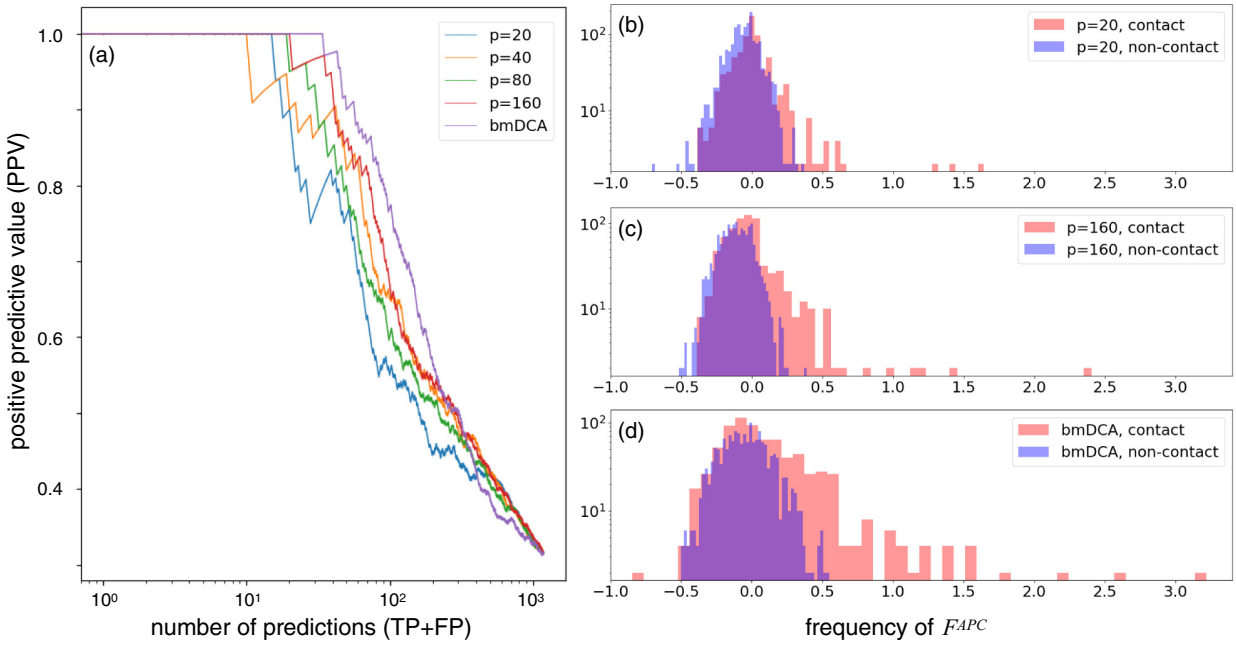

FIG. 8. The same as Fig. 3 but for the protein family PF00014.

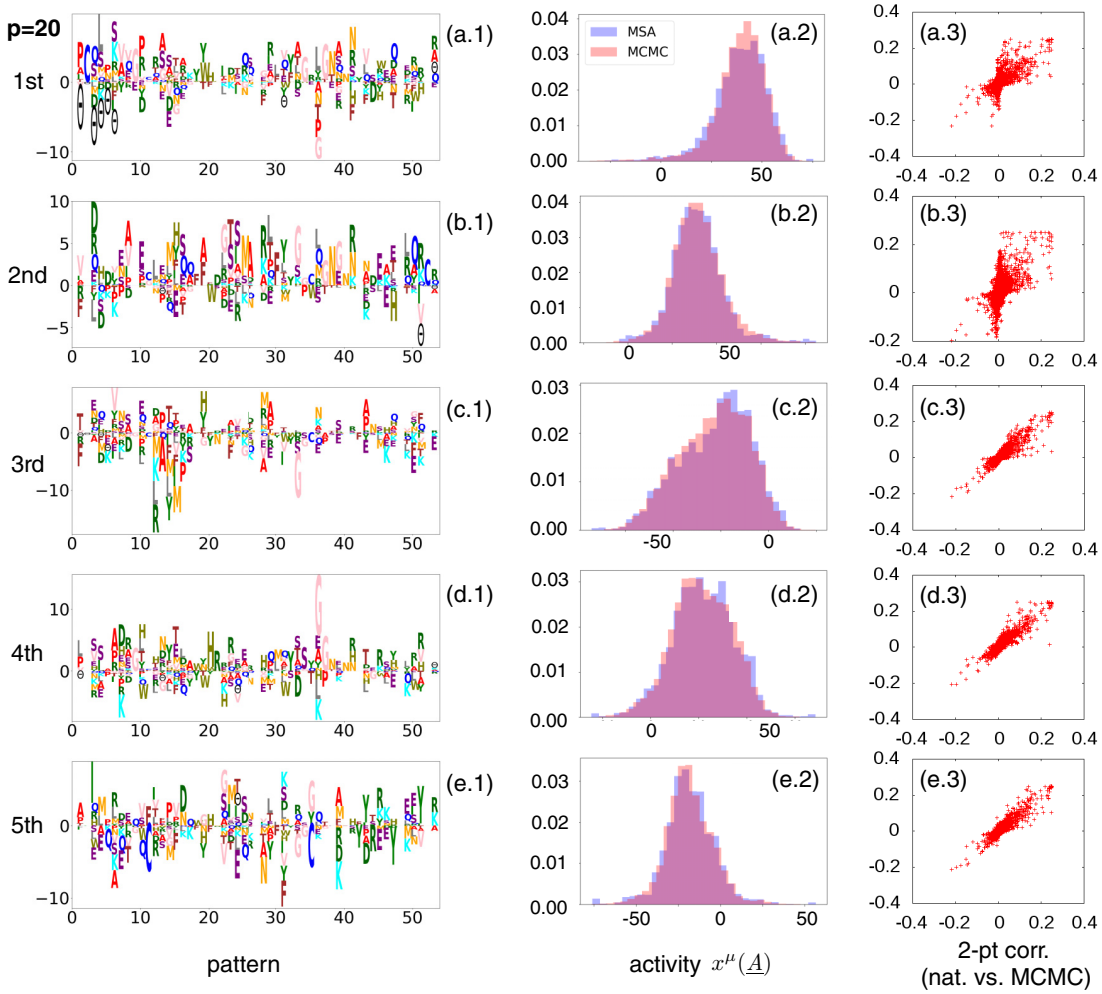

FIG. 9. The same as Fig. 5 but for the protein family PF00014.

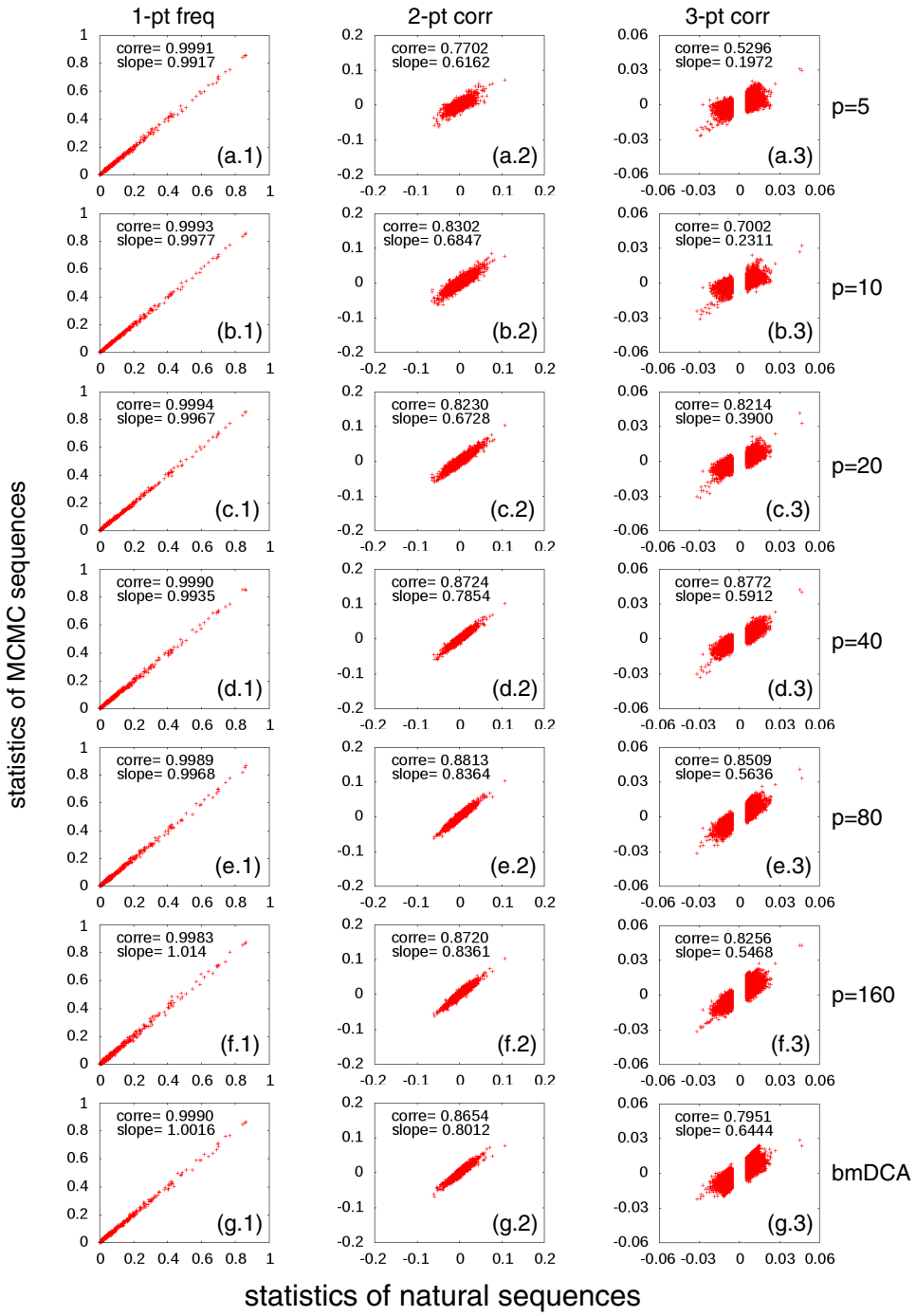

FIG. 10. The same as Fig. 2 but for the protein family PF00076.

VI. CONCLUSION AND OUTLOOK

In this paper, we have rederived Hopfield-Potts models as statistical models for protein sequences by selection of additive sequence motifs. Statistical sequence models are required to reproduce the first and second moments of the empirical motif distributions (i.e., over the MSA of natural sequences). Within a maximum-entropy approach, these motifs are found to be (up to a scaling factor) the Hopfield-Potts patterns defining a network of residue-residue couplings. In addition to the maximum-entropy framework, which is built upon known observables, the Hopfield-Potts model adds a step of variable selection: The probability of the sequence data is maximized over all possible selections of sequence motifs.

The quadratic coupling terms can be linearized using a Hubbard-Stratonovich transformation. When the Gaussian variables introduced in this transformation are interpreted as latent random variables, the Hopfield-Potts model takes the form of a restricted Boltzmann machine. This interpretation, originally introduced in Ref. [36], allows for the application of

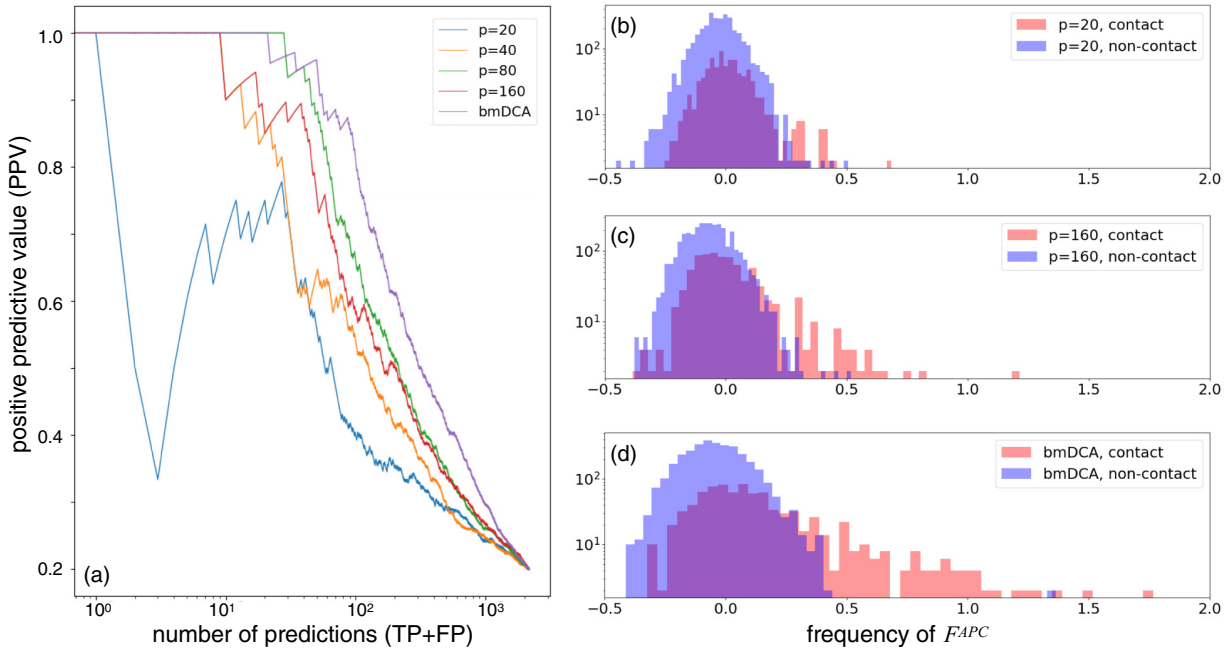

FIG. 11. The same as Fig. 3 but for the protein family PF00076.

efficient inference techniques, such as persistent contrastive divergence and, therefore, for the accurate inference of the Hopfield-Potts patterns for any given MSA of a homologous protein family.

We find that Hopfield-Potts models acquire interesting generative properties even for a relatively small number of parameters ($p = 20$–40). They are able to reproduce nonfitted properties, such as higher-order covariation of residues. Also, the bimodality observed in the empirical activity distributions (i.e., the projection of the natural sequences onto individual Hopfield-Potts patterns) is not automatically guaranteed when using only the first two moments for model learning, but it is recovered with high accuracy in the activity distributions of artificial sequences sampled from the model. This observation is not only interesting in the context of generative-model learning, but also forms the basis of sequence clustering according to interpretable sequence motifs in the main text.

The Hopfield-Potts patterns, or sequence motifs, are typically found to be distributed over many residues thereby representing global features of sequences. This observation explains why Hopfield-Potts models tend to lose accuracy in residue-residue contact prediction as compared to the full-rank Potts models normally used in direct coupling analysis: The sparsity of the residue-residue contact network cannot be represented easily via few distributed sequence motifs, which describe more global patterns of sequence variability. Despite the fact that Hopfield-Potts models reproduce also nonfitted statistical observables, the loss of accuracy in contact prediction demonstrates that these models are not fully generative and alternative concepts for parameter reduction should be explored.

Individual sequences from the input MSA can be projected onto the Hopfield-Potts patterns, resulting in sequence-specific activity values. Some patterns show a monomodal histogram for the protein family. They introduce a dense network of relatively small couplings between positions with sufficiently large entries in the pattern without dividing the family into subfamilies. These patterns have great similarity to the concept of protein sectors, which was introduced in Refs. [42,43] to detect distributed modes of sequence coevolution. However, the conservation-based reweighting used in determining sectors is not present in the Hopfield-Potts model, and the precise relationship between both ideas remains to be elaborated. Other Hopfield-Potts patterns show bimodal activity distributions, leading to the detection of functional subfamilies. Since these are defined by, e.g., the positive vs the negative entries of the pattern, the entries of large absolute value in the patterns identify residues, which play a role similar to so-called specificity determining residues [44,45], i.e., residues, which are conserved inside specific subfamilies, but vary between subfamilies. Both concepts—sectors and specificity-determining residues—emerge naturally in the context of Hopfield-Potts families, even if their precise mathematical definitions differ and, thus, also their precise biological interpretations.

These observations open up new ways of parameter reduction in statistical models of protein sequences: The sparsity of contacts, which are expected to be responsible for a large part of localized residue covariation in protein evolution, has to be combined with the low-rank structure of Hopfield-Potts models, which detect distributed functional sequence motifs. However, distributed patterns may also be related to phylogenetic correlations, which are present in the data, cf. Ref. [46]. As has been shown recently in a heuristic way [47], the decomposition of sequence-data covariance matrices or coupling matrices into a sum of a sparse and a low-rank matrix can substantially improve contact prediction if only the sparse matrix is used.

Combining this idea with the idea of generative modeling seems a promising road towards parsimonious sequence

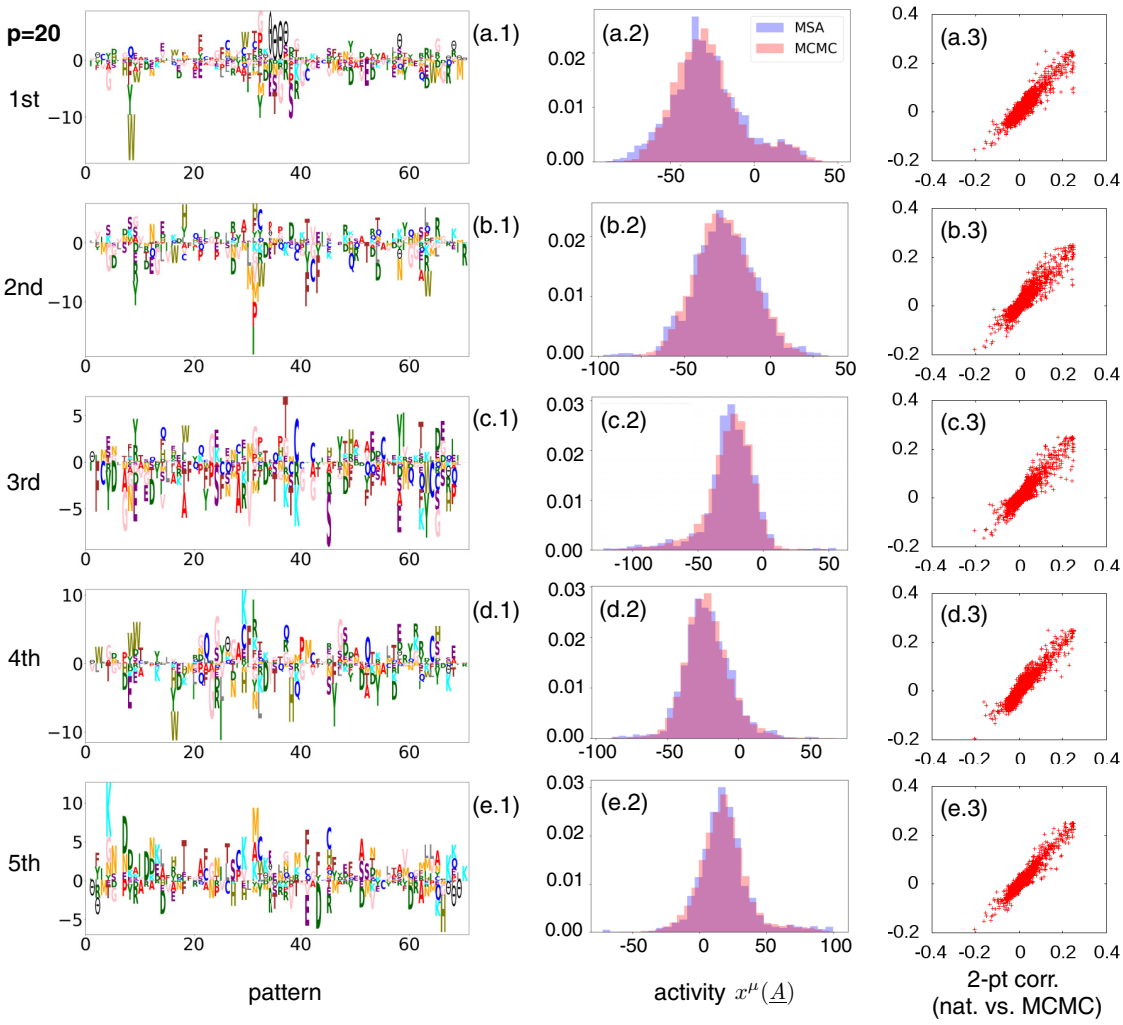

FIG. 12. The same as Fig. 5 but for the protein family PF00076.

models, which, in turn, would improve parameter interpretability and reduce overfitting effects, both limiting factors of current versions of DCA. In this context, it will also be interesting to explore more general regularization strategies which favor more localized sequence motifs or Hopfield-Potts patterns thereby unifying sparse and low-rank inference in a single framework of parameter-reduced statistical models for biological sequence ensembles.

ACKNOWLEDGMENTS

We are particularly grateful to P. Barrat-Charlaix, G. Croce, C. Lucibello, A.-P. Muntoni, A. Pagnani, E. Sarti, J. Tubiana, and F. Zamponi for numerous discussions and assistance with data. We acknowledge funding by the EU H2020 Research and Innovation Programme MSCA-RISE-2016 under Grant Agreement No. 734439 InferNet. K.S. acknowledges an Erasmus Mundus TEAM (TEAM—Technologies for information and communication, Europe—East Asia Mobilities) scholarship of the European Commission in 2017 and 2018 and a doctoral scholarship of the Honjo International Scholarship Foundation since 2018.

APPENDIX A: RESULTS FOR OTHER PROTEIN FAMILIES

The first Appendix is dedicated to other protein families. As discussed in the main text, we have analyzed three distinct families and discussed only one in full detail in the main text. Here, we present the major results—generative properties, contact prediction, and selected collective variables (patterns)—for two more families. These results show the general applicability of our approach beyond the specific response-regulator family used in the main text.

1. Kunitz-bovine pancreatic trypsin inhibitor domain PF00014

Figures 7–9 display the major results for the PF00014 protein family. PPV curves are calculated using PDB ID 5pti [48].

2. RNA recognition motif PF00076

Figures 10–12 display the major results for the PF00076 protein family. PPV curves are calculated using PDB ID 2x1a [49].

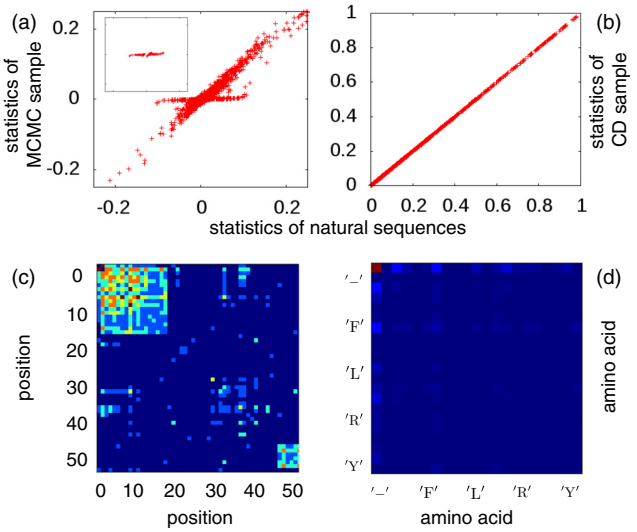

FIG. 13. The upper two panels show the statistics (two-point connected correlations) of the training data (PF00014) against an i.i.d. MCMC sample extracted from the inferred model and the CD sample used for inference. The perfect coincidence of the two in the CD case demonstrates that the CD algorithm is converged and contrast is lost despite the fact that the correlations extracted from an i.i.d. sample do not match the empirical ones. To understand this, we have selected those (i, j, a, b)'s with substantial deviations, cf. the inset in the first panel and analyzed their location in the protein (first panel) and their amino acid composition (second panel, amino acids in alphabetical order of one-letter code $[-, A, C, \ldots, Y]$), densities are represented via heat map plots. Location at the extremities in the sequences and in gap-gap correlations emerge clearly.

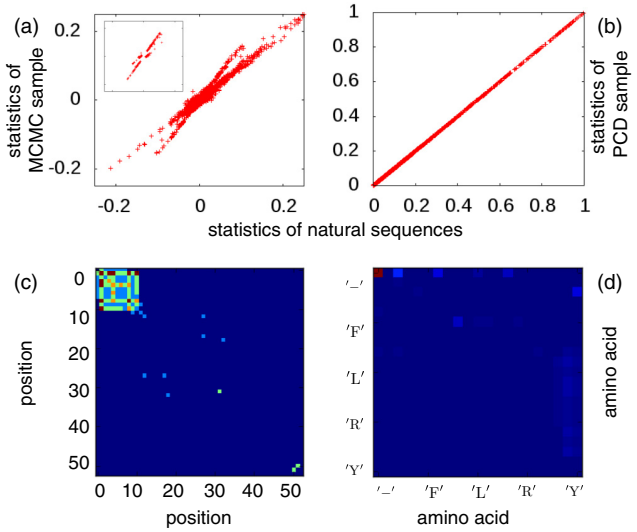

FIG. 14. The upper two panels show the statistics (two-point connected correlations) of the training data (PF00014) against an i.i.d. MCMC sample extracted from the inferred model and the PCD sample used for inference. As in the CD case, the MCMC sample shows larger deviations from the empirical observations than the PCD sample, but correlations appear overestimated and contrast in the PCD plot is sufficient to drive further evolution of parameters. To understand these observations, we have selected those (i, j, a, b)'s with substantial deviations, cf. the inset in the first panel and analyzed their location in the protein (first panel) and their amino acid composition (second panel, amino acids in alphabetical order of one-letter code $[-, A, C, \ldots, Y]$), densities are represented via heat map plots. Locations at the extremities in the sequences and in gap-gap correlations emerge clearly.

APPENDIX B: NOTES AND DETAILS ON INFERENCE METHODS

1. Regularization

In the case of limited data but many parameters, i.e., the case (Hopfield-)Potts models for protein families are in, the direct likelihood maximization in Eq. (18) can lead to overfitting effects, causing problems in sampling and parameter interpretation. To give a simple example, a rare and, therefore, unobserved event would be assigned zero probability, corresponding to (negative) infinite parameter values.

To cope with this problem, regularization is used. Regularization, in general, penalizes large (respectively, nonzero) parameter values and can be justified in Bayesian inference as a prior distribution acting on the parameter values. In this paper and following Ref. [20], we use a block regularization of the form

$$R(\xi, h) = \eta_0 \sum_{\mu=1}^{p} \left(\sum_{i,a} \left| \xi_i^\mu(a) \right| \right)^2 + q\eta_0 \sum_{i,a} h_i(a)^2 \,, \quad \text{(B1)}$$

with η_0 being a hyperparameter determining the strength of regularization. This regularization weakly favors sparsity of the patterns.

We use $\eta_0 = \alpha_0 L/qM$ with $\alpha_0 = 0.0525$ as default values throughout this paper. In the last section of this Appendix, we show that Hopfield-Potts inference is robust with respect to this choice.

2. Contrastive divergence vs persistent contrastive divergence

a. Contrastive divergence does not reproduce the two-point statistics

CD is a method for training restricted Boltzmann machines similar to persistent contrastive divergence. Initialized in the original data, i.e., the MSA of natural amino acid sequences, a few sampling steps are performed in analogy to Fig. 1, and the kth step is used in the parameter update to approach a solution of Eq. (21). However, rather than continuing the MCMC sampling from this sample, the sample is re-initialized in the original data after each epoch. This has, *a priori*, advantages and disadvantages: The sample remains close to a good sample of the model in CD but far from a sample of the intermediate model with not yet converged parameters.

As can be seen in Fig. 13, after a sufficient number of epochs the statistics of the CD sample and the training data are perfectly coherent, the model appears to be converged. However, the connected two-point correlations are not well reproduced when resampling the inferred model with standard MCMC. Part of the empirically nonzero correlations are not reproduced and mistakenly assigned very small values in the inferred model.

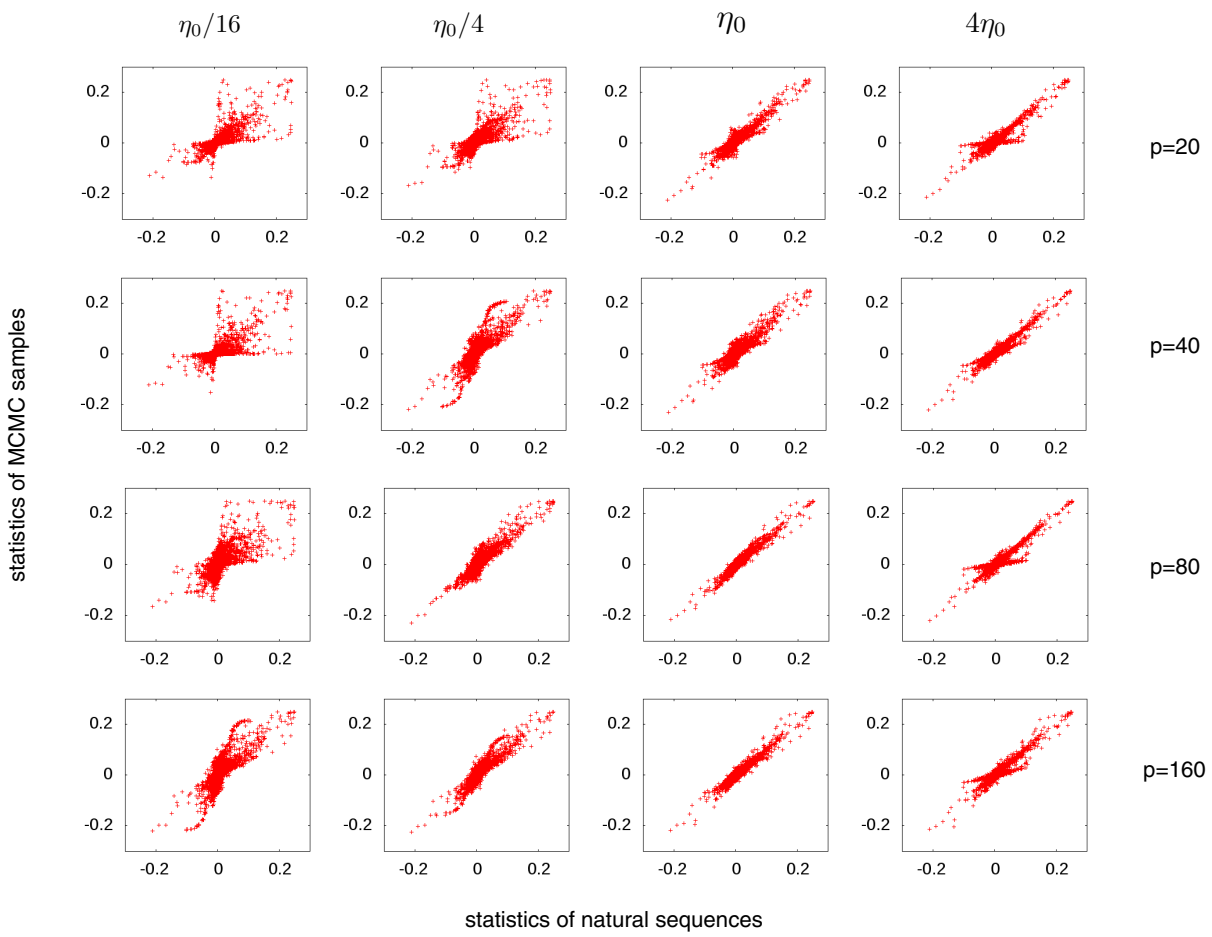

FIG. 15. Regularization dependence for CD inference, empirical two-point connected correlations (PF00014) are plotted against those estimated from the model using an i.i.d. MCMC sample. The regularization strength is varied over almost two orders of magnitude with $\eta_0 = \alpha_0 L/qM$ and $\alpha_0 = 0.0525$, going from a zone of overfitting to one of over-regularization. Results are shown for various values of p, illustrating a strong p dependence of the optimal coupling strength.

To understand this observation, we have selected the elements of the second panel, which show discrepancies between empirical and model statistics, cf. the inset in the figure. The corresponding values of (i, j, a, b) are strongly localized in the beginning and the end of the protein chain and correspond to the gap-gap statistics $c_{ij}(-,-)$. This gives a strong hint towards the origin of the problems in CD-based model inference: gap stretches, which exist in MSA of natural sequences, in particular, at the beginning and the end of proteins due to the local nature of the alignment algorithm used in PFAM. Those located at the beginning of the sequence start in position 1 and continue with only gaps until they are terminated by an amino acid symbol. They never start later than in position 1 or include individual internal amino acid symbols (analogous for the gap stretches at the end, which go up to the last position $i = L$).

In CD, only a few sampling steps are performed, so stretched gaps in the initialization tend to be preserved even if the associated gap-gap couplings are very weak. Basically, to remove a gap stretch, an internal position cannot be switched to an amino acid, but the gap has to be removed iteratively from one of its end points, namely, the one inside the sequence (i.e., not positions 1 or L). So, in CD, even small couplings are, thus, sufficient to reproduce the gap-gap statistics.

If resampling the same model with MCMC, parameters have to be such that gap stretches emerge spontaneously during sampling. This requires quite large couplings, actually, in BMDCA, gap-gap couplings between neighboring sites are the largest couplings of the entire Potts model. Using now the small couplings inferred by CD, these gap stretches do not emerge at sufficient frequency, and correspondingly the positions at the extremities of the sampled sequences appear less correlated.

b. Persistent contrastive divergence and transient oscillations in the two-point statistics

Persistent contrastive divergence overcomes this sampling issue by not reinitializing the sample after each epoch but by continuing the MCMC exploration in the next epoch with updated parameters.

As shown in the main text, PCD can actually be used to infer parameters, which lead to accurately reproduced two-point correlations when i.i.d. samples are generated from the inferred model. However, during inference, we have

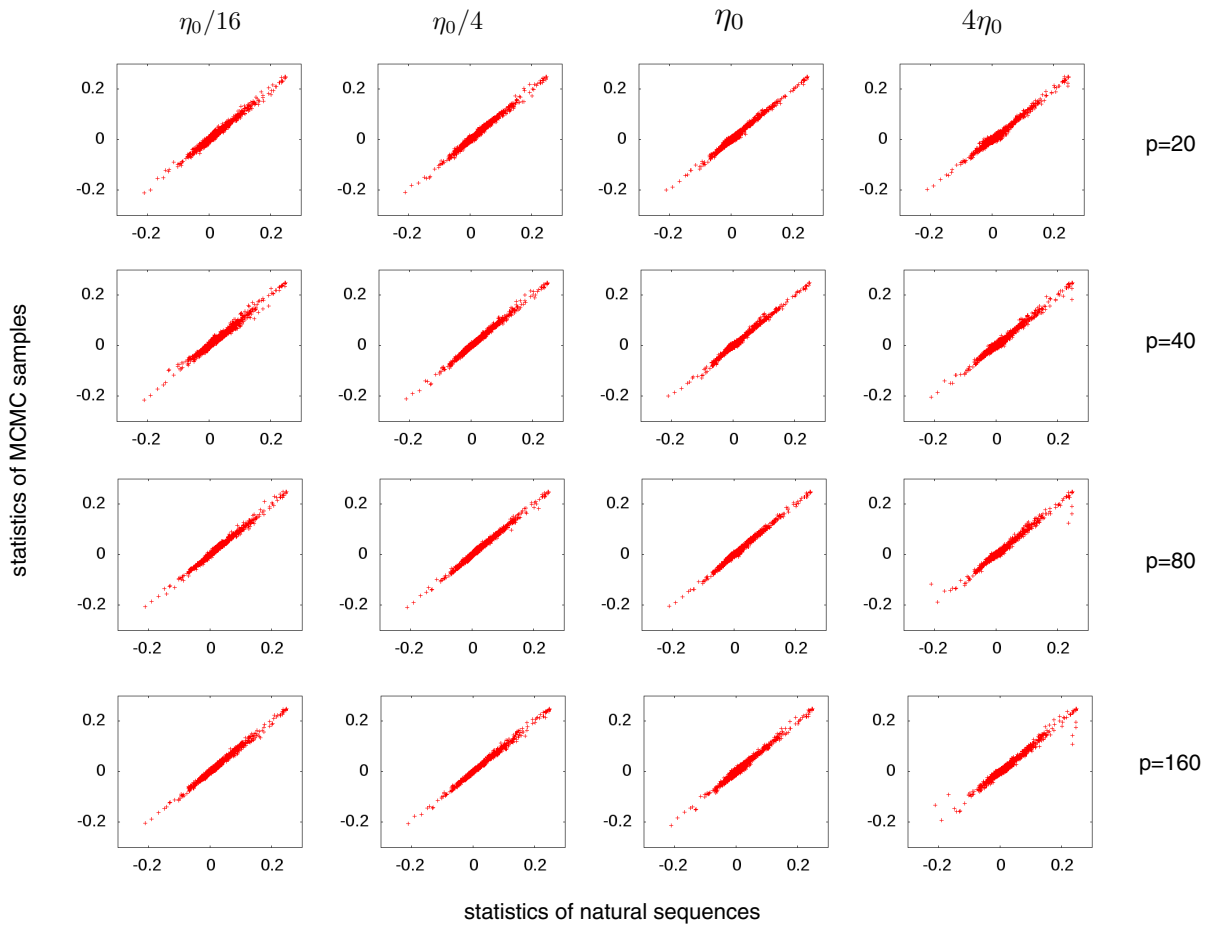

FIG. 16. Regularization dependence for PCD inference, empirical two-point connected correlations (PF00014) are plotted against those estimated from the model using an i.i.d. MCMC sample. The regularization strength is varied over almost two orders of magnitude with $\eta_0 = \alpha_0 L/qM$ and $\alpha_0 = 0.0525$; results are shown for various values of p. We find a strong robustness of results with respect to regularization.

observed transient oscillations, cf. Fig. 14 for an epoch where correlations in a subset of positions and amino acids are overestimated. An analysis of the positions and the spins involved in these deviations shows that again gap stretches are responsible.

The reason can be understood easily. Initially, PCD is not very different from CD. Gap stretches are present due to the correlation with the training sample, and only small gap-gap interactions are learned. However, after some epochs, the sample will lose the correlation with the training sample. Due to the currently small gap-gap correlations, gap stretches are lost in the PCD sample. According to our update rules, the corresponding gap-gap couplings will fastly increase. However, due to the few sampling steps performed in each PCD epoch, this growth will go on even when parameters would be large enough to generate gap stretches in an i.i.d. sample. Also, in the PCD sample, gap stretches will now emerge, but due to the overestimation of parameters, they will be more frequent than in the training sample, i.e., parameters start to decrease again. An oscillation of gap-gap couplings is induced.

The strength of these oscillations can be strongly reduced by removing samples with large gap stretches from the training data and train only on data with limited gaps. If the initial training set was large enough, the resulting models are even expected to be more precise since gap stretches do contain no or little information about the amino acid sequences under study. However, if samples are too small, the suggested pruning procedure may reduce the sample to an insufficient size for accurate inference. Care has, thus, to be taken when removing sequences.

3. Robustness of the results

As discussed before, we need to include regularization to avoid overfitting due to limited data. In Figs. 15 and 16, we show the dependence of the inference results due to changes in the regularization strength over roughly two orders of magnitude. The first of the two figures shows the results for CD: Empirical connected two-point correlations are compared with i.i.d. samples of the corresponding models. We note that the results depend strongly on the regularization strength. For low regularization, the correspondence between model and MSA is low due to overfitting. At strong regularization, only part of the correlations is reproduced, we over-regularize and, thus, underfit the data. For each protein family and each number p of patterns, the regularization strength would have to be tuned.

For PCD, the situation is fortunately much better; results are found to be very robust with respect to regularization, cf. Fig. 16. This allows us to choose one regularization strength across protein families and pattern numbers.

[1] UniProt Consortium, Nucleic Acids Res. **47**, D506 (2018).
[2] R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas *et al.*, Nucleic Acids Res. **44**, D279 (2016).
[3] S. R. Eddy, *Genome Informatics 2009: Genome Informatics Series Vol. 23* (World Scientific, Singapore, 2009), pp. 205–211.
[4] M. Remmert, A. Biegert, A. Hauser, and J. Söding, Nat. Methods **9**, 173 (2012).
[5] T. Schwede, J. Kopp, N. Guex, and M. C. Peitsch, Nucleic Acids Res. **31**, 3381 (2003).
[6] B. Webb and A. Sali, Curr. Protoc. Bioinf. **47**, 5 (2014).
[7] D. De Juan, F. Pazos, and A. Valencia, Nat. Rev. Genet. **14**, 249 (2013).
[8] S. Cocco, C. Feinauer, M. Figliuzzi, R. Monasson, and M. Weigt, Rep. Prog. Phys. **81**, 032601 (2018).
[9] H. C. Nguyen, R. Zecchina, and J. Berg, Adv. Phys. **66**, 197 (2017).
[10] M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, and T. Hwa, Proc. Natl. Acad. Sci. USA **106**, 67 (2009).
[11] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, Proc. Natl. Acad. Sci. USA **108**, E1293 (2011).
[12] S. Balakrishnan, H. Kamisetty, J. G. Carbonell, S.-I. Lee, and C. J. Langmead, Proteins: Struct., Funct., Bioinf. **79**, 1061 (2011).
[13] D. T. Jones, D. W. Buchan, D. Cozzetto, and M. Pontil, Bioinformatics **28**, 184 (2012).
[14] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander, PLoS One **6**, e28766 (2011).
[15] S. Ovchinnikov, H. Park, N. Varghese, P.-S. Huang, G. A. Pavlopoulos, D. E. Kim, H. Kamisetty, N. C. Kyrpides, and D. Baker, Science **355**, 294 (2017).
[16] J. K. Mann, J. P. Barton, A. L. Ferguson, S. Omarjee, B. D. Walker, A. Chakraborty, and T. Ndung'u, PLoS Comput. Biol. **10**, e1003776 (2014).
[17] F. Morcos, N. P. Schafer, R. R. Cheng, J. N. Onuchic, and P. G. Wolynes, Proc. Natl. Acad. Sci. USA **111**, 12408 (2014).
[18] M. Figliuzzi, H. Jacquier, A. Schug, O. Tenaillon, and M. Weigt, Mol. Biol. Evol. **33**, 268 (2016).
[19] H. Szurmant and M. Weigt, Curr. Opin. Struct. Biol. **50**, 26 (2018).
[20] J. Tubiana, S. Cocco, and R. Monasson, eLife **8**, e39397 (2019).
[21] E. T. Jaynes, Phys. Rev. **106**, 620 (1957).
[22] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids* (Cambridge University Press, Cambridge, UK, 1998).
[23] S. R. Eddy, Bioinformatics **14**, 755 (1998).
[24] C. Baldassi, M. Zamparo, C. Feinauer, A. Procaccini, R. Zecchina, M. Weigt, and A. Pagnani, PloS One **9**, e92721 (2014).
[25] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell, Phys. Rev. E **87**, 012707 (2013).
[26] L. Sutto, S. Marsili, A. Valencia, and F. L. Gervasio, Proc. Natl. Acad. Sci. USA **112**, 13567 (2015).
[27] A. Haldane, W. F. Flynn, P. He, R. Vijayan, and R. M. Levy, Protein Sci. **25**, 1378 (2016).
[28] J. P. Barton, E. De Leonardis, A. Coucke, and S. Cocco, Bioinformatics **32**, 3089 (2016).
[29] M. Figliuzzi, P. Barrat-Charlaix, and M. Weigt, Mol. Biol. Evol. **35**, 1018 (2018).
[30] M. Socolich, S. W. Lockless, W. P. Russ, H. Lee, K. H. Gardner, and R. Ranganathan, Nature (London) **437**, 512 (2005).
[31] S. Cocco, R. Monasson, and M. Weigt, PLoS Comput. Biol. **9**, e1003176 (2013).
[32] G. D. Stormo, T. D. Schneider, L. Gold, and A. Ehrenfeucht, Nucleic Acids Res. **10**, 2997 (1982).
[33] E. van Nimwegen, BMC Bioinf. **8**, S4 (2007).
[34] P. Smolensky, Tech. Rep. No. CU-CS-321-86, Colorado University, Colorado, 1986.
[35] G. E. Hinton and R. R. Salakhutdinov, Science **313**, 504 (2006).
[36] A. Barra, A. Bernacchia, E. Santucci, and P. Contucci, Neural Networks **34**, 1 (2012).
[37] G. E. Hinton, *Neural Networks: Tricks of the Trade*, edited by G. Montavon, G. B. Orr, and K. R. Müller, Lecture Notes in Computer Science, Vol. 7700 (Springer, Berlin, Heidelberg, 2012), pp. 599–619.
[38] B. Syed Ibrahim, S. K. Burley, and S. Swaminathan (2009), released on PDB by: New York SGX Research Center for Structural Genomics (NYSGXRC).
[39] G. Uguzzoni, S. J. Lovis, F. Oteri, A. Schug, H. Szurmant, and M. Weigt, Proc. Natl. Acad. Sci. USA **114**, E2662 (2017).
[40] J. Bauer, K. Reiss, M. Veerabagu, M. Heunemann, K. Harter, and T. Stehle, Mol. Plant **6**, 959 (2013).
[41] M. Kato, T. Shimizu, T. Mizuno, and T. Hakoshima, Acta Crystallogr., Sect. D: Biol. Crystallogr. **55**, 1257 (1999).
[42] N. Halabi, O. Rivoire, S. Leibler, and R. Ranganathan, Cell **138**, 774 (2009).
[43] O. Rivoire, K. A. Reynolds, and R. Ranganathan, PLoS Comput. Biol. **12**, e1004817 (2016).
[44] G. Casari, C. Sander, and A. Valencia, Nat. Struct. Mol. Biol. **2**, 171 (1995).
[45] L. A. Mirny and M. S. Gelfand, J. Mol. Biol. **321**, 7 (2002).
[46] C. Qin and L. J. Colwell, Proc. Natl. Acad. Sci. USA **115**, 690 (2018).
[47] H. Zhang, Y. Gao, M. Deng, C. Wang, J. Zhu, S. C. Li, W.-M. Zheng, and D. Bu, Biochem. Biophys. Res. Commun. **472**, 217 (2016).
[48] A. Wlodawer, J. Walter, R. Huber, and L. Sjölin, J. Mol. Biol. **180**, 301 (1984).
[49] C. Pancevac, D. C. Goldstone, A. Ramos, and I. A. Taylor, Nucleic Acids Res. **38**, 3119 (2010).

4.3 Mean-field Potts model for learning RBM

It is known that the RBM with Gaussian hidden variables is equivalent to the Hopfield-Potts (HP) model [95, 96]. As shown in [97], a HP model can be represented as the marginalization of the joint probability distribution of $p(\boldsymbol{A}, \boldsymbol{x})$, which corresponds to RBM with Gaussian hidden variables,

$$p(\boldsymbol{A}, \boldsymbol{x}) = \frac{1}{\tilde{Z}} \exp\left(\sum_{i,\mu} x^\mu \xi_i^\mu(A_i) + \sum_i^L h_i(A_i) - \frac{L}{2} \sum_\mu^P (x^\mu)^2 \right) , \tag{4.1}$$

where $\tilde{Z}$ is the partition function. The probability of a sequence $p(\boldsymbol{A})$ can be understood as the consequence of the marginalized distribution of Eq. 4.1,

$$\begin{aligned} p(\boldsymbol{A}) &= \int_{\mathbb{R}^P} p(\boldsymbol{A}, \boldsymbol{x}) d\boldsymbol{x} \\ &= \frac{1}{Z} \exp\left(\sum_{i,j} \frac{1}{L} \sum_\mu^P \xi_i^\mu(A_i) \xi_j^\mu(A_j) + \sum_i h_i(A_i) \right) . \end{aligned} \tag{4.2}$$

Although RBMs can reproduce statistics of training data, mean-field HP (mfHP) models cannot serve reasonable generative models. Note that mean-field DCA overestimates coupling parameters, thous cannot also be used as a generative model. On the other hand, mfHP models are theoretically and numerically well-investigated, and mfHP patterns directly associate with data covariance matrices [98, 95, 51].

This section aims to understand RBM patterns by comparing mfHP patterns. Our findings are as follows:

- RBM tends to preferentially learn patterns that are similar to the attractive mfHP patterns with significant likelihood contributions in the HP model (Sec. 4.3.1).
- Attractive mfHP patterns can be close to one of the optimal patterns for RBM. In other words, when the attractive mfHP patterns are used as the initial states of RBM, they keep patterns close to the initial states (Sec. 4.3.1).
- Introducing HP patterns into RBM helps to improve learning efficiency

and control each pattern (Sec. 4.3.2).

In the entire section, we assumed a Gaussian prior distribution for the hidden variables. The following experiments were performed for the response regulator domain (Pfam ID, PF00072), but qualitatively similar results are obtained for other domain families.

4.3.1 Comparison between mean-field Hopfield-Potts model and RBM with Gaussian hidden variables

Overlaps between mfHP and RBM patterns – In the case of the mfHP model [95], there are two types of patterns: Those corresponding to large eigenvalues of the Pearson correlation matrix ($\lambda_\mu > 1$) and the others associated with small eigenvalues ($0 < \lambda_\mu < 1$). These patterns are called attractive and repulsive [98, 95]. As established in Ref. [95], distributions of entries of attractive patterns are typically dense. On the contrary, entries of repulsive patterns are typically sparse and strongly localized in some specific positions i and amino-acids a. Such localized entries are commonly associated with structural interactions, hence necessary for accurate contact predictions [98].

Fig. 4.1.a shows typical attractive and repulsive patterns. These are the second most significant attractive pattern (top) and the most significant repulsive pattern (bottom).

Ref. [95] shows that the patterns, whose eigenvalues are significantly larger than 1 or markedly smaller than 1, contribute to the log-likelihood function of the mfHP model, cf. Eq. 2.34. Fig. 4.1.b shows the log-likelihood contributions as a function of eigenvalues.

It is important to examine how far the estimated patterns of the mean-field approximation are from the patterns of the exact HP model. Note that the learning dynamics are quite different between the HP model and RBM. However, as long as RBM uses Gaussian hidden variables, these models are equivalent.

In order to investigate how similar mfHP patterns and RBM patterns

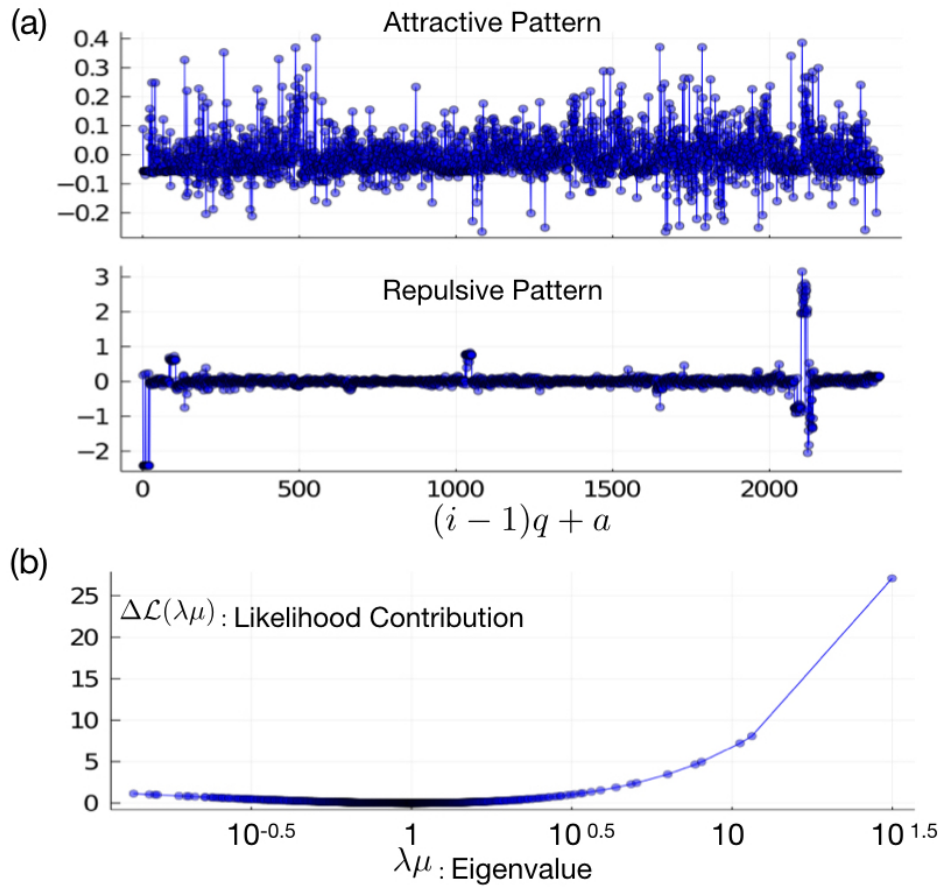

Figure 4.1: (a) An attractive pattern (top) and a repulsive pattern (bottom). Distributions of entries of attractive patterns are usually dense across the site and residue types. In contrast, the distributions of repulsive patterns are typically localized at specific residues and amino acids, which are generally associated with spatial interaction. (b) Likelihood contribution as a function of eigenvalues of the Pearson covariance matrix. Patterns with eigenvalues close to 1 have a small contribution to the log-likelihood, whereas those eigenvalues that are greater than / less than 1 make a significant contribution. Note that the largest eigenvalue gives the largest influence on the likelihood, but that corresponds with conservation terms, hence associating with the single site frequencies (see the following discussion and Fig. 4.2).c.

are, we introduce here the overlap, i.e., inner products of these patterns,

$$\begin{aligned} \langle \boldsymbol{\xi}^{\mu}, \boldsymbol{v}^{\nu} \rangle &= (\boldsymbol{\xi}^{\mu})^{t} \boldsymbol{v}^{\nu} \\ \|\boldsymbol{\xi}^{\mu}\| &= \|\boldsymbol{v}^{\nu}\| = 1 \ , \end{aligned} \tag{4.3}$$

where $\{\boldsymbol{\xi}^{\mu}\}$ are RBM patterns and $\{\boldsymbol{v}^{\nu}\}$ are mfHP patterns.

Fig. 4.2.a shows the overlaps between the RBM and mfHP attractive patterns $|\langle \boldsymbol{\xi}^{\mu}, \boldsymbol{v}^{\nu} \rangle|$. In this test, we used $P = 2$ hidden variables for the RBM. For attractive mfHP patterns, $\mu = 1, 2, \ldots, 400$ patterns are selected from the top in descending order of eigenvalues, that is, in descending order of contribution to the log-likelihood.

The top three attractive mfHP patterns show significant overlap with RBM patterns, which is greater than the overlaps between the two different RBM patterns (red lines) (an estimated overlap using the same qL dimension 1000 Gaussian random vectors is around 0.0165). Notably, the mfHP patterns corresponding with the second and third largest eigenvalues show a strong correlation with each RBM pattern (we hereafter call a HP pattern corresponding with the n-th largest eigenvalue as n-th mfHP pattern). The first mfHP pattern does not show a strong correlation with any RBM patterns. Instead, we found a large overlap with the local field, which associates with single-site frequency Fig. 4.2.c .

In Fig. 4.2.b, the overlaps between the repulsive patterns and RBM patterns are significantly smaller than the case of the attractive patterns and the overlaps between the two RBM patterns (red lines). These results clearly illustrate the absence of the repulsive modes in RBM (with Gaussian hidden variables) patterns.

We will also investigate the comparison between a mfHP model and RBM with $P = 16$ hidden variable case. Note that learning RBM is a non-convex optimization problem, and the set of RBM patterns is not unique and depends on the initial conditions. Therefore, as the number of hidden variables increases, so does the number of solutions (the optimal set of patterns). Consequently, it is nontrivial how the similarity between RBM patterns and mfHP patterns can be changed as an increase in the number of patterns. Regarding this fact, we use mfHP patterns as the initial states of RBM patterns and investigated how well RBM patterns can preserve mfHP patterns after learning.

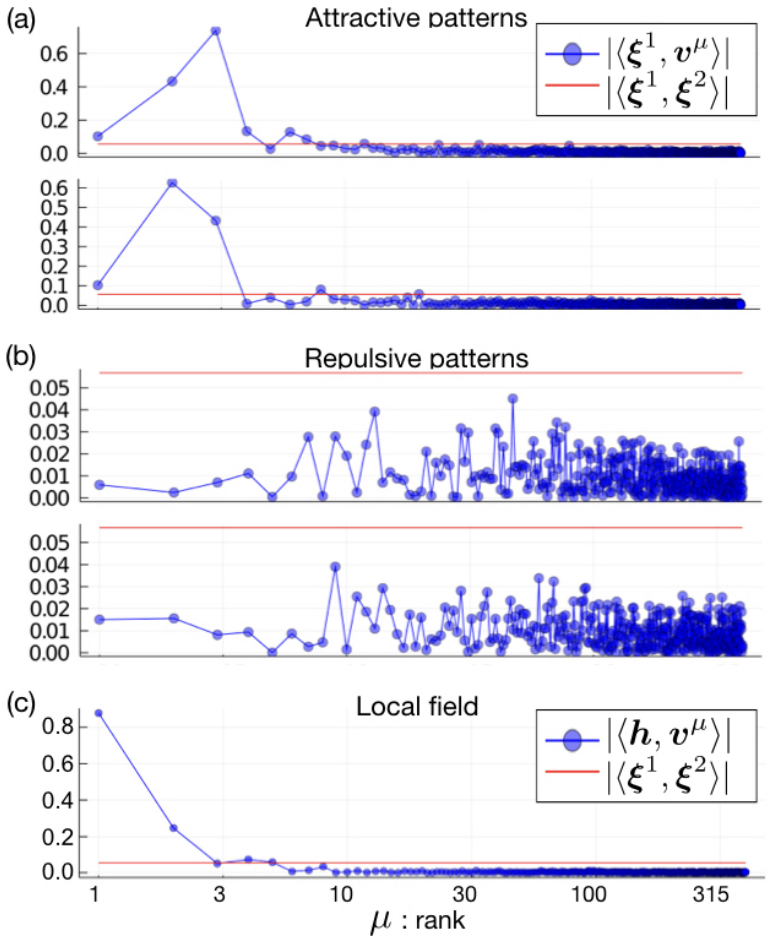

Figure 4.2: (a) Overlap between RBM patterns ($P = 2$) and attractive HP patterns ($P = 400$), $|\langle \boldsymbol{\xi}^\mu, \boldsymbol{v}^\nu \rangle|$. mfHP patterns are sorted in descending order according to the eigenvalues (the likelihood contribution decrease in the same manner). The red lines in each panel are an overlap value between the RBM patterns, $|\langle \boldsymbol{\xi}^1, \boldsymbol{\xi}^2 \rangle|$. The second and third mfHP patterns show significant correlations with RBM patterns. (b) The same types of figures as (a) but for repulsive patterns. None of the repulsive patterns show similarity with the RBM patterns. (c) Overlap between between local field $\boldsymbol{h}$ and the attractive mfHP patterns. The first attractive mfHP pattern shows substantial similarity with the local field, which also associate with the single site frequency $f_i(a)$ (See Appendix B.2).

As for learning the RBM, we selected only attractive mfHP patterns as initial conditions of RBM patterns. These mfHP patterns are ranked second to 17th in the likelihood contribution. The first mfHP pattern was excluded because of its similarity with the single-site frequency. The other learning conditions are same as in the case of the random initializations (cf. Fig. 4.3)

Fig. 4.3 shows the overlaps between RBM patterns with $P = 16$ (each panel corresponds with one of these patterns) and attractive mfHP patterns. We cannot find RBM patterns that are particularly similar to one of the mfHP patterns. On the other hand, as Fig. 4.4 shows that when a RBM pattern was initialized by one of mfHP patterns, it is specifically similar to the one used as the initial condition even after the learning. That is, the set of mfHP patterns is close to one of the stable optimal solutions for RBM (using Gaussian hidden variables).

Interestingly the overlap values are small (< 0.2) even if mfHP patterns are corresponding the top ten largest eigenvalues, except those that were used as the initialization of the learning. Therefore, these RBM patterns after training show strong specificity for the initialized mfHP pattern. This tendency becomes more pronounced as the contribution to log-likelihood increases (the left on the top panel shows a greater maximum overlap than the right on the bottom panel). It indicates that one of the optimal RBM pattern sets (there are multiple optimal solutions) is close to the mfHP pattern set.

Moreover, although the RBM patterns show specificity to the initialized mfHP patterns, they differ to some extent, and their overlap values are not very close to 1. Therefore, the difference between a mfHP pattern and the RBM pattern that uses it as the initial condition makes a difference whether it is a generative model or not.

In conclusion, using mfHP patterns as initial states of the RBM pattern, it is possible to effectively "label" each RBM pattern (hidden variable) and encourage the patterns to become independent. In addition, it is also possible to accelerate the learning time by using mfHP pattern as the initial condition of RBM.

The similarities between RBM patterns are shown in Appendix D. None of the RBM pattern pairs show an overlap greater than 0.2. Even for pairs

of 0.1 or more, the pair is only about 15% Fig. D.1 and Fig. D.2.

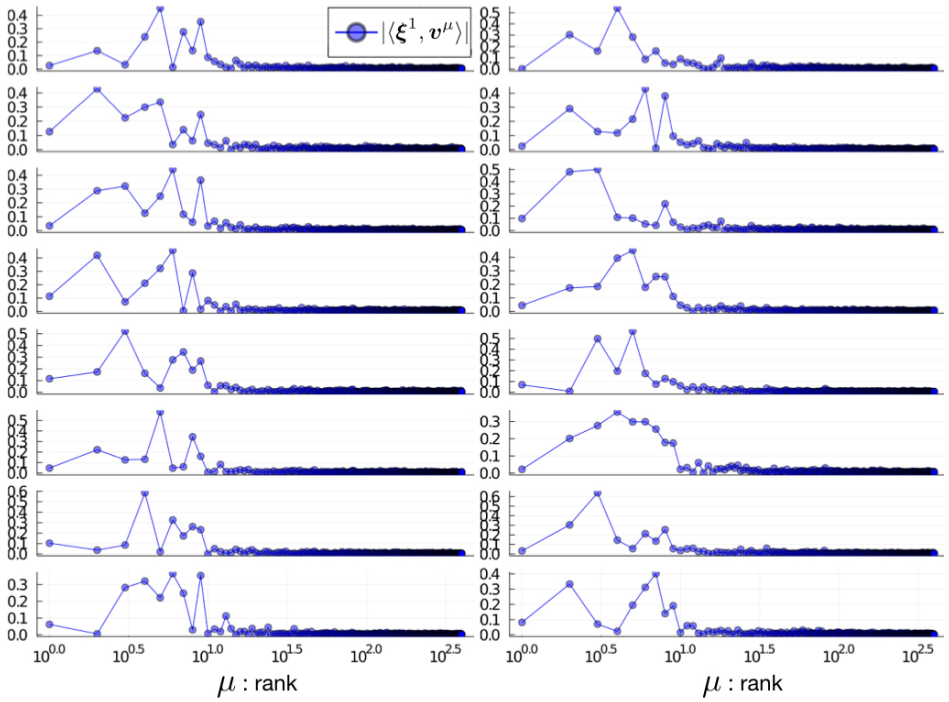

Figure 4.3: In contrast to the case of RBM with $P = 2$, each RBM pattern overlaps with multiple HP patterns. The overlap greater than around 0.1 (below of which corresponds to estimation noise) is across the top around ten patterns. The RBM patterns with $P = 16$ are expressed by superimposing multiple mfHP patterns.

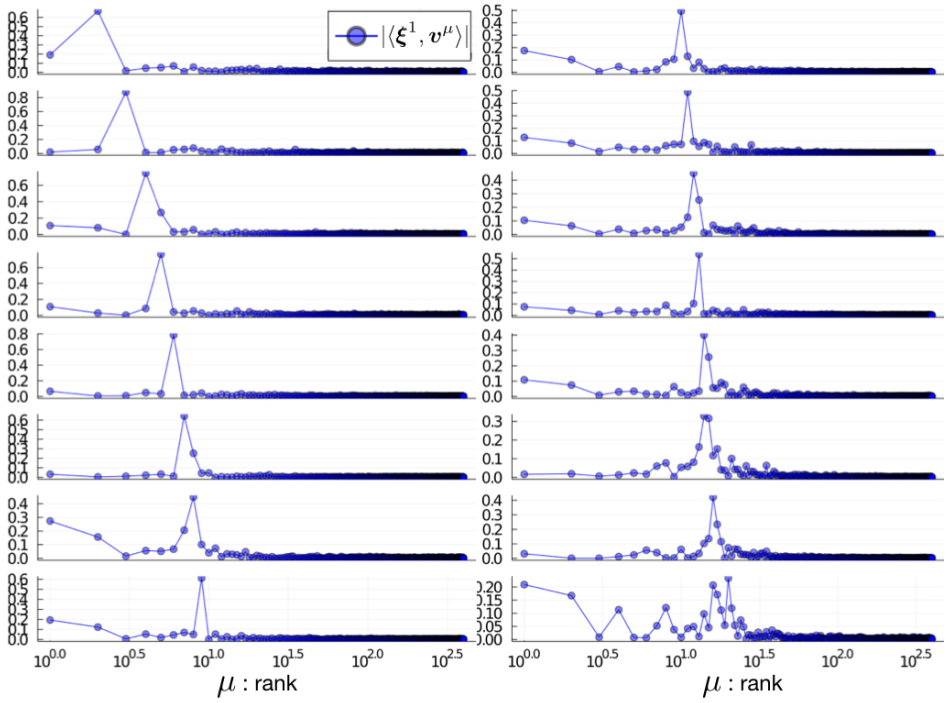

Figure 4.4: Overlap values between the attractive mfHP patterns and RBM patterns using the mfHP patterns as initial states. In the order of the upper left to lower right panel, it corresponds to the attractive mfHP patterns with the rank according to the likelihood contribution. The left, on the top to the bottom: 1st to 8th likelihood contribution. The right, on the top to the bottom: 9th to 16th.

4.3.2 Applications

Mean-field Hopfield-Potts pattern for a generative model

The HP Energy function in Eq. 4.2 is a rotational invariant with respect to HP patterns. That is, the energy function does not change under the following transformation,

$$\xi_i^\mu(a) \to \sum_\nu \mathcal{O}^{\mu\nu}\xi_i^\nu(a) \ . \tag{4.4}$$

It is called a gauge invariance, and such a degree of freedom is referred to as the gauge freedom in physics (see also Sec. 2.4.2). Although RBM patterns and mfHP patterns show significant overlap, these gauge choices are different. Thus those inner product depends on the gauge choice.

In this section, we will compare RBM patterns and mfHP patterns in terms of gauge-invariant quantities. Specifically, we will investigate statistics that are generated from RBM using mfHP patterns.

To build a generative model using RBM with mfHP patterns, we choose as free parameters the number of pattern P and the scale parameters of the mfHP pattern, a parameter that multiplies the mfHP pattern in the energy function of RBMs, $\boldsymbol{v}^\mu \to \boldsymbol{v}^\mu/T^* =: \boldsymbol{\xi}^\mu$, where $\boldsymbol{v}^\mu$ is a mfHP pattern and $\boldsymbol{\xi}^\mu$ is a RBM pattern. As we saw in the last section, repulsive mfHP patterns show very low similarity with RBM patterns, hence we consider only attractive patterns.

Finding the optimal scale parameters T^* can be useful for learning RBMs efficiently and well-controlled manner:

1. The first advantage is the acceleration of the learning processes, by choosing patterns that are close to the final configuration of the pattern reduces the computational cost.

2. In addition, as we mentioned in the last Sec. 4.3.1, the RBM learning is a non-convex problem and usually depends on the initial conditions of the model parameters. Therefore by introducing the mfHP patterns as the initial condition, we can label for each hidden variable (using the likelihood-contribution of the mfHP patterns).

As we understand from Sec. 4.3.1, the pattern with the largest contribution to the likelihood function tends to associate with the single-site

frequency. Therefore, we select patterns from the second to the $(P+1)$ th patterns. In this experiment we used $P = 2$ and $P = 16$.

Fig. 4.5.a shows the values of Pearson coefficients for two-point connected correlations as a function of $1/T$. The correlations are computed from an ensemble of sequences generated from the RBM with $P = 2$ mfHP attractive patterns using a scale parameter $1/T$, and an MSA. We selected protein families and the MSA that we used in our article [92]. The maximum Pearson coefficients using the attractive mfHP patterns are $0.3 - 0.7$, which are always smaller than the Pearson coefficients using the RBM patterns (> 0.6) as expected.

Fig. 4.5.b shows the same types of figure with Fig. 4.5.a, but with many mfHP attractive patterns ($P = 16$). Surprisingly, the Pearson coefficients are much smaller values than the cases of $P = 2$. One of the possible reasons is that the optimal scale parameter might be different for each pattern. It may depend on the eigenvalues. According to the likelihood contributions Fig. 4.1.b, the second and third eigenvalues are close. Therefore, scaling with a single parameter can achieve relatively good statistics when $P = 2$. However, the range of the eigenvalues up to the top 17th change substantially ($\lambda_\mu = 0.3 - 12$). Therefore it may need to introduce scale parameters for each pattern when $P = 16$.

Low-dimensional analysis using $P = 2$ model

In the previous section, we saw that RBM patterns show substantial similarity with attractive mfHP that are significantly contributing to the likelihood. Notably, RBM patterns of $P = 2$ show strong specificity for a few attractive mfHP patterns (Fig. 4.2), whereas RBM patterns of $P = 16$ show more broad similarities among many attractive mfHP patterns (Fig. 4.3). Therefore, the RBM patterns with $P = 2$ would be also similar to the largest eigenmodes of the covariance matrix (hereafter we call as "principal modes"), which have a clear relation with the attractive mfHP patterns (Eq. C.5).

Moreover, RBM with $P = 2$ patterns can still reproduce statistical properties; the values of Pearson correlation are typically greater than 0.6. It can be even 0.8 in some cases (see the red lines in Fig. 4.5). Therefore, it is a generative model that shows the similarity to the principle modes of the

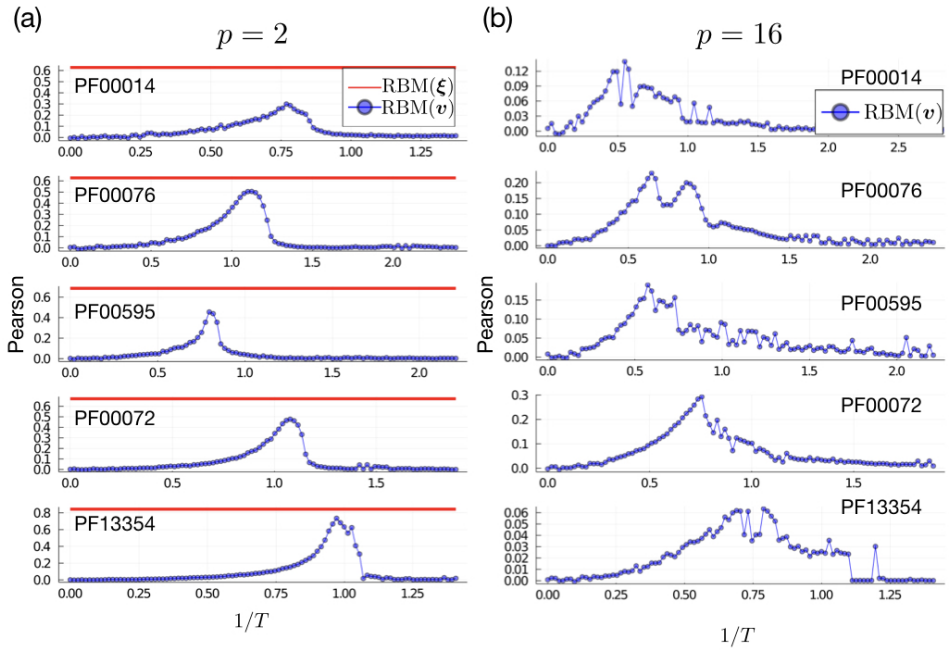

Figure 4.5: (a) Pearson coefficients of two-point connected correlation $C_{ij}(a, b)$, computed from the natural MSA and samples from the RBM with mfHP attractive patterns for as a function of a parameter $1/T$ (represented as blue circles and labeled as RBM(ξ^{HP})). As comparisons, we also included the Pearson values using sequences generated from the RBM with $P = 2$ RBM patterns, which are represented as red lines (denoted as RBM($\boldsymbol{\xi}$)). (b) the same types of plot with the (a), but Pearson values using sequences generated from the RBM with the $P = 16$ mfHP patterns (Pearson values using RBM with $P = 16$ RBM patterns are typically greater than 0.9).

covariance matrix.

Another advantage of using a smaller number of patterns is that they can be used for low-dimensional analysis by projecting sequences to the pattern space. Since RBM can be used as a generative model, projected variables in the low-dimensional space corresponding to the hidden variables are thought to capture the important features that characterize the training data.

Note that these hidden variable activities in the low-dimensional spaces can be understood in terms of the statistical modelings, which making it significantly different from commonly used low-dimensional analysis. For example, the marginalized HP distribution in Eq. 4.2 can be written as a function of the mean of hidden variables, $x^\mu(\boldsymbol{A}) = \frac{1}{L}\sum_i \xi_i^\mu(A_i)$. That is,

$$p(\boldsymbol{A}) \propto \exp\left(\frac{L}{2}\sum_\mu^P (x^\mu(\boldsymbol{A}))^2 + \text{const.}\right) \ .$$

It shows the probability distribution of a sequence $\boldsymbol{A}$ for each pattern $\mu \in 1{:}P$, and the larger $(\sum_i \xi_i^\mu(A_i))^2$, the greater the dependence on this pattern μ.

We show here the low-dimensional spaces of protein sequences. As the low-dimensional spaces, we used following patterns:

(a) Principal component vectors of the data covariance matrix (we will explain the data set bellow).

(b) $P = 2$ RBM patterns used random initialization for learning. These patterns are obtained using RBM's standard learning method PCD, and the training data is the same as in (a).

(c) mfHP attractive patterns, correspond with the second and third most contributing patterns in the likelihood function (cf. Eq. 2.34). The dataset used is the same as in (a).

(d) $P = 2$ RBM patterns used the mfHP patterns used in (c) for the initial condition of learning. Other conditions are the same as in (a).

We used the data set of the response regulator domain family (Pfam ID, PF00072) for this experiment. More precisely, within protein sequences of the PF00072 domain, we constructed subgroups of sequences according to

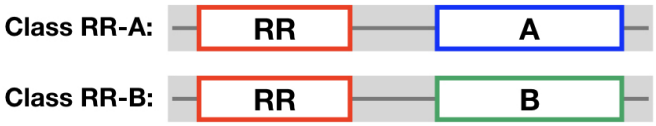

Figure 4.6: Schematic of two protein subfamilies of RR protein domain, those are distinguished according to the adjoining protein domains. Class RR-A protein family, which adjacents to domain A. Class RR-B protein family adjacents to domain B.

other specific protein domains adjacent to the PF00072 domain. For example, sequences belong to the other protein subfamily adjacent to a protein domain A, but sequences belong to the other protein subfamily adjacent to a different protein domain B (see a schematic of adjacent protein domains, Fig. 4.6). In each protein subfamily, slight changes can occur in the three-dimensional structure and function according to other adjacent protein domains. Consequently, the difference of sequences between different protein subfamilies is pronounced. In Table 1, we summarized the considering protein subfamilies of PF00072 protein domain.

Fig. 4.7.a shows projections of PF00072 protein sequences using the low-dimensional space mentioned in (a), where the projection means mapping to a sequence to two-dimensional space such that $\{1{:}q\}^L \rightarrow \mathbb{R}^2$. The color of each point corresponds to each protein subfamily (the mapping between the colors and protein subfamilies is summarized in Table 1).

Similarly, Fig. 2, Fig. 3, and Fig. 4 correspond to the projections of the protein sequences to the low-dimensional spaces (b) the RBM patterns with the random initialization, (c) attractive mfHP patterns, and (d) the RBM patterns with the mfHP patterns initialization mentioned earlier, respectively. For example, variables in the two-dimensional space projected using the RBM pattern can be written as: $\sum_i \xi_i^\mu(A_i)$, $\mu \in \{1, 2\}$.

For protein subfamilies named classes PF00072-PF00512, PF00072-PF00512 (after 500 aa), and PF00072-PF00512 (after 1000 aa) (purple, red, and orange colors are assigned, respectively). These protein families can be very similar in nature, as the only difference is how far the domains PF00072 and PF00512 are from each other on the protein sequence. In fact, they are projected on similar areas, and non of the low-dimensional space can

Pfam ID	Color	M
PF00158	Green	5,046
PF00196	Cyan	8,219
PF00486	Blue	15,234
PF00512	Purple	14,727
PF00512 (after 500 aa)	Red	9,231
PF00512 (after 1000 aa)	Orange	3,400
PF00990	Pink	1,698
PF01339	Brown	943
PF04397	Gray	3,176
PF12833	Darkblue	2,474
No pair	Yellow	9,415

Table 1: Summary of protein subfamilies for PF00072. The left column shows Pfam IDs of the protein subfamily (discussed below). The center and the right columns explain the colors used in Fig. 4.7 and the number of sequences included in the protein subfamily data, respectively. The protein subfamily of each PF00072 is adjacent to the protein domain shown in the leftmost column of the table. PF00512 (after 500 aa) means that there are 500 amino acids between the PF00072 domain and PF00512. Similarly, after 1000 aa, means there are 1000 amino acids between them. No pair means that the PF00072 domain does not adjacent to any protein domain. The middle column shows the color types used in Fig. 4.7. M states the number of protein sequences in each ensemble of the protein subfamilies.

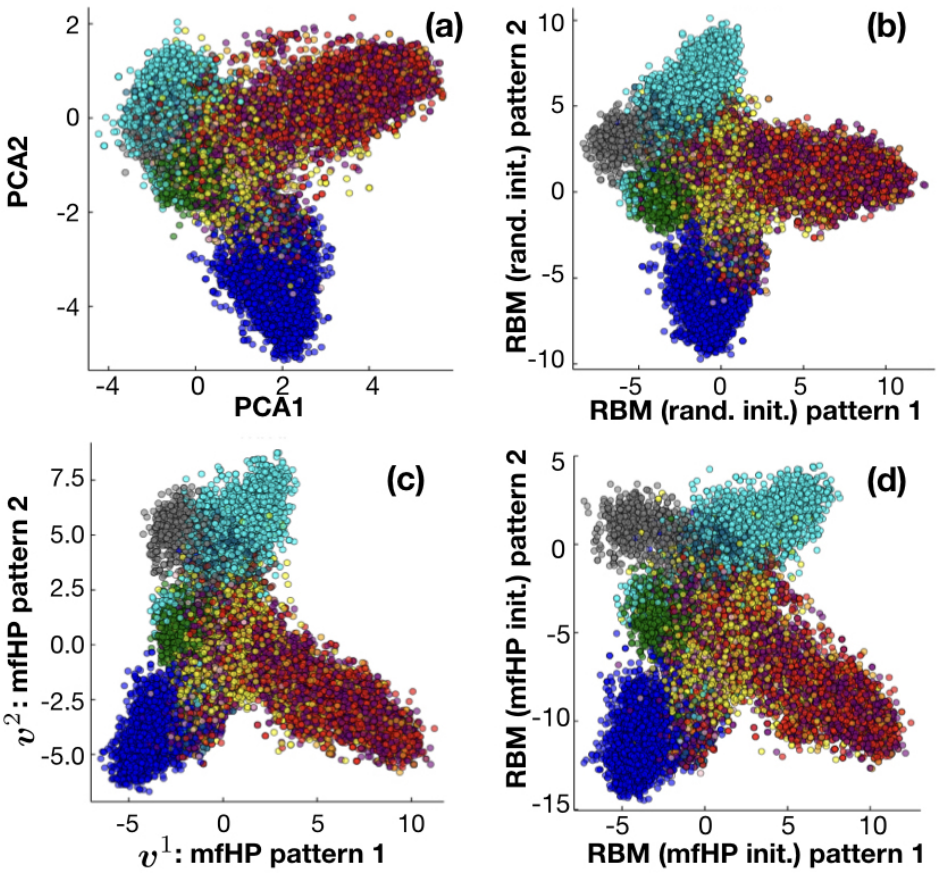

Figure 4.7: (a) Projection of the protein sequences to the PCA space. Each color represents different protein subfamilies (cf. Table 1). we can see three distinct regions in the PCA space. (b) Projection using the RBM patterns (initial states of learning were random patterns). (c) Projection using the attractive mfHP patterns. (d) Projection using the RBM patterns (initial states were the mfHP patterns used in (c)).

distinguish them.

Sequences of protein domains that are not adjacent to any protein domain (yellow color is assigned) are projected over a large area. This tendency can be observed in all low-rank spaces. Probably because there is no adjacent protein family, the sequences belonging to this protein family are more likely to be changed by mutational drift rather than specific mutations, and the sequence variability among this protein family is high.

As explained previously in Sec. 4.3.1, RBM patterns with a small P shows significant similarity to attractive mfHP patterns. Therefore, the projection using the RBM patterns (initially random patterns) Fig. 4.7.b and the projection using the mfHP patterns Fig. 4.7.c are similar. However, the shape of the projected distribution is not exactly the same direction in Fig. 4.7.b and Fig. 4.7.c , but it is projected in rotation (Fig. 4.7.b is the shape of "+", but Fig. 4.7.c is rather the shape of "x"). We presume that this is the direct result of the rotational invariance between the patterns described in Sec. 4.3.1 (Eq. 4.4). Note that (d) is "gauge-fixed" by using the attractive mfHP patterns as the initial state of RBM patterns, and shows the same shape of distribution as (c) (both are "x" shapes).

In the case of the low-dimensional spaces except for the PCA space, the protein subfamilies, including large populations (more than 3,000 sequences), are mostly projected into regions that differ from other protein subfamilies except for the protein families we mentioned before. In the PCA projections, the subfamilies classes PF00072-PF00158 (Green) and PF00072-PF04397 (Gray) overlap with other prominent subfamily areas and are indistinguishable.

Interestingly, the low-dimensional spaces based on RBM patterns (both (b) and (c)) tend to project these subfamilies PF00072-PF00158 and PF00072-PF04397 to regions that are different from other subfamilies. Such differences may characterize the differences between generative and non-generative models.

4.4 Criticality of RBMs

This section shows that the number of hidden variables P required to reproduce the statistical properties of protein sequences is much smaller than

the number qL of the pairwise Potts model. We may think that the reduction of model parameters helps the low-rank model evade criticality, i.e., the sensitivity of the statistics against slight changes in model parameters (see Chap.3 and Ref. [92]).

We exploit the heat capacity, which is equivalent to a variance of an energy function, to assess the robustness of the statistics against the global change of the model parameters. Similar argument can be found in our study [92].

For the HP model, the heat capacity function $C^{HP}(T)$ is defined as:

$$
\begin{aligned}
C^{HP}(T) &= \frac{\partial \langle E^{HP} \rangle_T}{\partial T} = \frac{1}{T^2}\left(\langle (E^{HP})^2 \rangle_T - \langle E^{HP} \rangle_T^2 \right) , \\
E^{HP}(\boldsymbol{A}) &= -\sum_{i<j} J_{ij}^{HP}(A_i, A_j) - \sum_{i=1}^{L} h_i^{HP}(A_i) , \\
J_{ij}^{HP}(a,b) &= \frac{1}{L}\sum_{\mu=1}^{P} \xi_i^\mu(a)\xi_j^\mu(b) , \qquad h_i^{HP}(a) = h_i(a) + \frac{1}{L}\sum_{\mu=1}^{P}(\xi_i^\mu(a))^2 ,
\end{aligned}
\tag{4.5}
$$

where the average in Eq. 4.5, denoted as $\langle \bullet \rangle_T$ is evaluated as a sample average from a RBM with Gaussian hidden variables. Therefore, although we use the energy function of the HP model, the effect on the heat capacity is purely due to the statistical properties of the RBM. In order to be equivalent to a HP model $p^{HP}(\boldsymbol{A}|T) \propto \exp(-E^{HP}(\boldsymbol{A}/T))$, we scaled the energy function of RBM by $h_i(a) \to h_i(a)/T$ and $\xi_i^\mu(a) \to \xi_i^\mu(a)/\sqrt{T}$ [1]

Fig. 4.8.a shows the heat capacity for different numbers of hidden variables P in the RBMs. As the data set, we used the RNA recognition motif domain (Pfam ID PF00076). The figure clearly demonstrates that as we increase the number of hidden variables, the peaks of the heat capacity become rapidly steeper. As P increases further ($P > 16$), it becomes difficult to

[1] Introducing the temperature parameter to the RBM energy function is not unique, we can obtain equivalent result using different scaling transformation:

$$
-E^{RBM}(\boldsymbol{A}, \boldsymbol{x}) = \sum_{i,\mu} x^\mu \xi_i^\mu(A_i) + \frac{1}{T}\sum_i h_i(A_i) - T\sum_\mu \frac{L}{2}(x^\mu)^2 ,
$$

after we integrating out the hidden variables we get the same result as $\exp(-E^{HP}(\boldsymbol{A}/T))$.

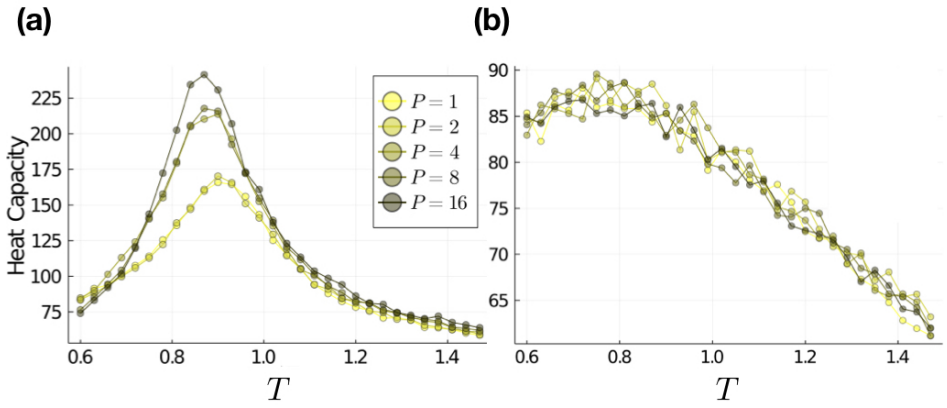

Figure 4.8: (a) Heat capacity $C(T)$ of the RBMs using a protein family data, the RNA recognition motif domain (Pfam ID PF00076) as the training data. We used $P \in \{1, 2, 4, 8, 16\}$ patterns cases. There are the peaks below $T = 1$ and as P increase the peak becomes more pronounced that means the statistics of the RBMs become more sensitive against a slight perturbation on the parameters. (b) Heat capacity of the RBMs with randomized patterns ($\boldsymbol{\xi}^{rand}$). The means μ^{rand} were estimated as $(-0.234, -0.152, -0.0291, 0.00391, -0.00509)$ for each case of $P \to (1, 2, 4, 8, 16)$. Similarly, the standard deviations σ^{rand} were estimated as $(0.423, 0.438, 0.4331, 0.406, 0.369)$.

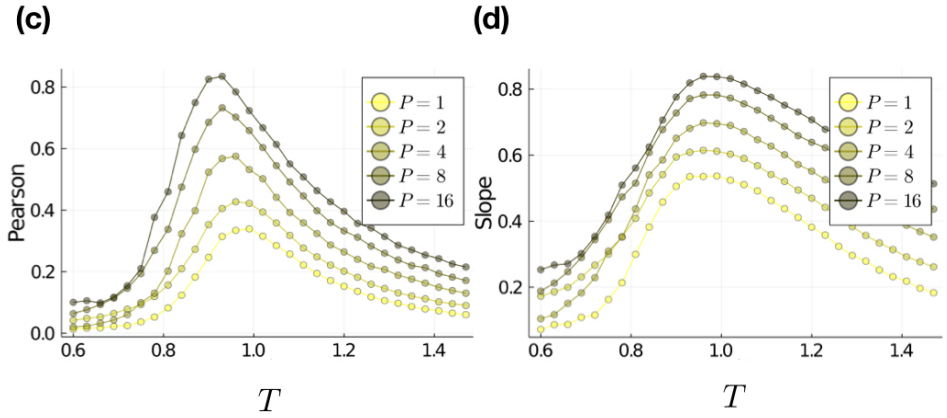

Figure 4.9: (a) Pearson coefficient of two-point connected correlations comparing the protein sequences (Pfam ID PF00076) and sequences generated from the RBMs. As P increases, the Pearson value increases in all temperature regions T and the P dependence becomes more pronounced around $T \sim 0.95$. (b) Similarly, it shows the slope of the comparison of the two-point correlations between natural sequences and sequences generated from RBMs. Qualitatively similar to the result of Pearson values in Fig. 4.9.c, the optimal temperature is slightly below $T = 1.0$.

perform sampling from the equilibrium distribution in the low-temperature regime ($T < 0.9$). This result indicates that the influence is not because of architecture but rather the effects of the underlying features of the training data. Fig. 4.8.b shows the heat capacity using random patterns $\boldsymbol{\xi}^{rand}$. These random patterns are generated from Gaussian distribution with the same mean and variance values as the learned RBM patterns. Formally, we can represent ξ^{rand}

$$\begin{aligned} &\mathcal{N}(\mu^{rand}, \sigma^{rand}) \sim \xi_i^{\mu,\ rand}(a) \in \mathbb{R}^1 \ , \forall i, a, \mu, \\ &\mu^{rand} = \mathrm{Mean}(\boldsymbol{\xi}^{1:P}) \in \mathbb{R}^1 \ , \\ &(\sigma^{rand})^2 = \mathrm{Var}(\boldsymbol{\xi}^{1:P}) \in \mathbb{R}^1 \ , \end{aligned} \tag{4.6}$$

where μ^{rand} and $(\sigma^{rand})^2$ are a mean and variance of the Gaussian distribution (normal distribution), and these values are estimated by the RBM patterns $\boldsymbol{\xi}^{1:P}$ using the functions Mean($\bullet$) which returns arithmetic average of the arguments $\bullet$, similarity Var($\bullet$) is a function that returns a variance.

Fig. 4.9 shows comparison of two-point correlation $C_{ij}(a, b)$ of training data (PF00076, the same data used in Fig. 4.8) and sequences generated from the model using Pearson (a) values and slope (b).

Both Pearson value and the slope increase steadily as P increases. In the case of $P = 16$, it already achieved the Pearson values 0.8 (it improves further as increase the P).

We found that the temperature that achieves the highest Pearson value is below $T < 1.0$ (between $0.9 < T < 1.0$). Considering the argument of the heat capacity, this result, therefore, indicates the presence of optimal P and T in the sense that it achieves reasonably good statistics but is less critical (low-temperature is feasible).

4.5 Conclusion

This Chapter investigated low-rank models, also named Hopfield-Potts (HP) models, for proteins sequence generative models. The HP model can be converted to a restricted Boltzmann machine (RBM) with Gaussian hidden variables using the Hubbard-Stratonovich transformation, both of which are equivalent statistical models.

Sec. 4.2 (cf., [97]) showed that RBMs with a number of hidden variables (~ 40), which is sufficiently smaller than qL can reproduce two-point cor-

relations of protein sequences (Pearson values of two-point correlations are ~ 0.9). Note that two-point correlations, unlike in the case of bmDCA, are quantities that are not guaranteed to be reproduced by the model.

Moreover, hidden variables of RBM show specificities for each protein subfamily. Therefore, protein subfamilies can be classified by the activities of hidden variables, which are equivalent to projections of protein sequences using RBM patterns.

Sec. 4.3.1 compared RBM patterns, which are equivalent to HP patterns and mean-field HP (mfHP) patterns. RBM (with Gaussian hidden variables) patterns show similarity with attractive mfHP patterns that especially contribute to the log-likelihood (cf. Fig. 4.2).

RBM contains a set of patterns similar to mfHP patterns as the optimal solutions. When mfHP patterns are used as the initial states of the RBM, they show notable similarities with the mfHP patterns used in each initial state after learning (cf. Fig. 4.3 and 4.4).

Sec. 4.3.2 shows the application of observations in Sec. 4.3.1. First, we investigated how accurately the statistics could be reproduced when the mfHP patterns were used as the RBM patterns.

In the case of $P = 2$, some RBMs that use mfHP patterns can generate statistics as well as standard RBMs by scaling the mfHP patterns with temperature parameters (cf. Fig. 4.5.a. in Sec. Astonishingly, it achieves Pearson values of two-point correlations that are bout 0.8 in the case of PF13345). As we discussed in Sec. 4.3.1, mfHP patterns are close to the ones of the RBM's solutions. Therefore the case of P=16 RBM with mfHP patterns could also generate statistics well if these patterns were correctly scaled (cf. Fig. 4.5.b). To understand appropriate scaling for each pattern, that is to understand the dependency of eigenvalues on the optimal scaling is the objective of a future work.

As another application, we performed low-dimensional analyses of protein subfamilies using these RBM and mfHP patterns. As a further comparison, we performed the same analysis based on RBM patterns that are initialized by mfHP patterns (cf. Fig. 4.7).

We constructed the intermediate space (d) between the RBM pattern space (b) and the mfHP pattern space (c) by using the RBM patterns that are used mfHP patterns as the initial states (these labels (b), (c) and (d) correspond to the labels in Fig. 4.7).

In Fig. 4.7, the comparison between (c) and (d) demonstrates the effect of the mean-field approximation. Some protein sequences of protein subfamilies (e.g. Class PF00072-PF00158 and PF00072-PF04397) degenerate in the mfHP pattern space. In other words, the activities of hidden variables must be specific for each of the different major subgroups in order to be a generative model. Similarly, the comparison between (b) and (d) shows the effect of the rotational invariance of the HP patterns.

In Sec. 4.4, we discuss the criticality of RBMs using heat capacities [92]. Increasing the number of hidden variables P has a more pronounced effect on the statistics of RBMs by perturbing the model parameters. It is interesting to understand how there is a difference between how the heat capacity changes with respect to change in P in RBMs and how the heat capacity changes with respect to the change in the coupling densities d in sparse BM (cf., Sec. 3.2 and [92]).

There are many questions we couldn't address in our studies. Some of them are written as follows.

As we saw in [97], large P improves residue contact accuracy, and also the repulsive patterns tend to associate with residue contacts [98, 51]. A question considering these facts is: how increasing P influences the presence of repulsive patterns? Do the increase in the number of P induce repulsive patterns?

Other related questions concerning the emergence of repulsive patterns and the independence [2] of patterns are: How does the selection of the hidden variable prior distributions affect the patterns? How does the choice of regularization affects the patterns?

[2]For this purpose, we need to look at the collective activity of hidden variables, $\langle x^\mu x^\nu \rangle$, where $\langle \bullet \rangle$ means the average of hidden space gives the training data.

Chapter 5

Models combining sparse and low-rank couplings

In the case of sparse bmDCA, we successfully reduced the number of couplings. In general, contact prediction performance remains stable even below coupling densities $d \leq 5\%$ (cf., Fig. 2 in Chap. 3.2). Therefore, the remaining strong couplings after the decimation are mostly couplings associated with contact pairs. However, statistics from sparse bmDCAs are significantly degraded when we go below coupling densities $d \leq 10\%$ (cf., Fig. 1 in Chap. 3.2).

On the other hand, low-rank or Hopfield-Potts (HP) models are excellent for generating statistics using a very limited number of parameters. Moreover, the HP patterns can distinguish protein subfamilies, especially those of high likelihood contributions. However, the accuracy of contact prediction of the low-rank model is not as good as plmDCA or bmDCA. The main reason is the absence of the repulsion pattern in the models [98, 95]. Therefore, importantly, both sparse BM and low-rank models learn different features from the same data.

This chapter explores the possibilities of statistical models that combine both sparse couplings and low-rank models (Fig. 5.1). In particular, the chapter is organized as follows:

- Section 5.1 introduces some backgrounds and motivations of this study: Residue-residue contact predictions when phylogenetic correlations are present in training data.

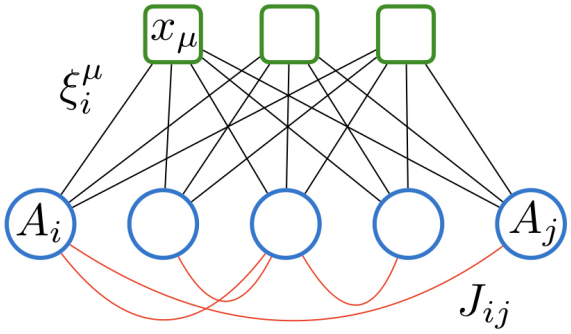

Figure 5.1: A graphical representation of the model we propose in this section: Combining sparse couplings and low-rank couplings. Blue circles and green squares denote amino acids and hidden variables respectively. Interactions between these variables are represented as edges, where black edges are interactions defined in RBM and red edges are interactions of the sparse bmDCA.

- Section 5.2 shows how to combine sparse couplings and low-rank couplings. We propose methods for selecting sparse couplings that are complementary parameters to low-rank couplings. Particularly, we discuss element-wise couplings (Sec. 5.2.1) and block-wise couplings (Sec. 5.2.2) individually.
- Section 5.3 shows some results of the proposed method. In Sec. 5.3.1, we discuss the residue-residue contact predictions.

5.1 Motivation

It was reported [44] that the eigenvalue distributions of the covariance matrices of biological sequences show power-law distributions. This phenomenon is due to the effects of phylogenies, which are included in protein sequence ensemble typically. Ref. [44] shows that removing the largest eigenmodes from the covariance matrices can reduce phylogenetic effects and improve residue-residue contact prediction accuracy. (Fig. 5.2 shows the eigenvalue distribution and the contact prediction's accuracy). Here we emphasize that while RBM is a generative model of protein sequences, as explained in the previous section (see Sec. 4.3.1), the RBM patterns are strongly associated

with the eigenmodes of the covariance matrix. Especially when RBMs depend on a small number of hidden variables such as $P = 2$, this relation becomes more pronounced. On the other hand, the RBM with Gaussian hidden variables cannot treat the repulsive patterns that correspond with residue-residue contacts [98, 51].

Ref. [44] also provides a simple yet very suggestive insight to the question "Why does DCA work well with non-i.i.d. samples?". Here we sketch this idea demonstrated in this reference article. Suppose the covariance matrix C can be decomposed into eigenmodes, $C = \lambda_1 \boldsymbol{v}_1 \boldsymbol{v}_1^t + \cdots + \lambda_P \boldsymbol{v}_P \boldsymbol{v}_P^t$. Where, $\boldsymbol{v}_\mu$ and λ_μ are a non-zero-mode eigenvector and eigenvalue, respectively. Here, P is the rank of the covariance matrix. In the case of mfDCA, the coupling matrix is nothing but its inverse. Hence, it can be represented as $-J = C^{-1} = \lambda_1^{-1} \boldsymbol{v}_1 \boldsymbol{v}_1^t + \cdots + \lambda_P^{-1} \boldsymbol{v}_P \boldsymbol{v}_P^t$, therefore the coupling matrices effectively down-weights large eigenvalue modes. Combining this simple argument and the main claim in the reference article, that is the that large eigenvalues are the consequences of phylogeny, we can understand why DCA can accurately predict residue contacts even if protein sequence data is non-i.i.d.

Although the largest eigenmodes of the covariance matrix may show signs of phylogenies, still, we cannot bring clear answers to the following questions:

- How to remove more effectively the large eigenmodes within the framework of statistical modelings?
- How to distinguish phylogenetically or structurally induced correlations?

Here we propose a tentative answer to these questions from a generative modeling framework.

5.2 Coupling activation

As we noted, RBMs tend to learn the patterns that correspond to the largest eigenvalue modes (hereafter, we referred to them as "principal modes") of the covariance matrix. Moreover, such principal modes presumably associate with phylogenetic effects [44]. By considering these factors, the following hypothesis can be stand: Distributions obtained by subtracting the RBMs probability distributions from an empirical distribution suppresses

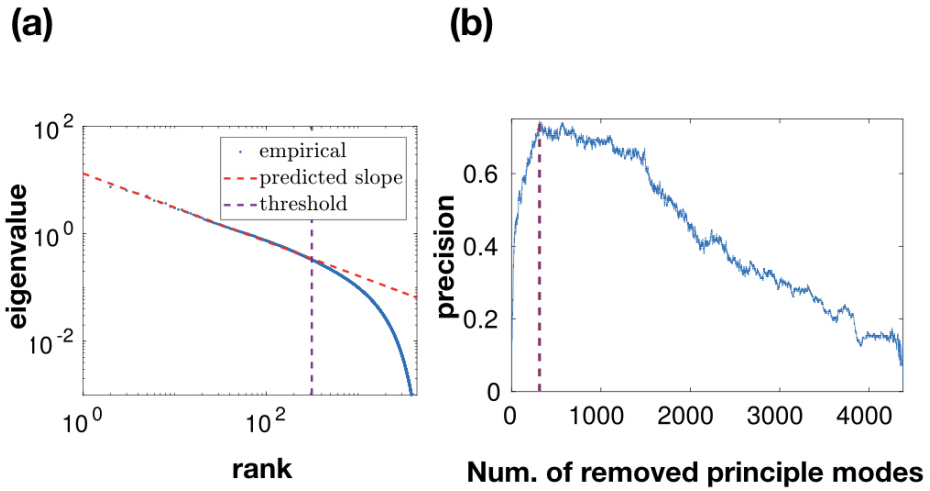

Figure 5.2: (a) An eigenvalue distribution of protein sequences covariance matrix, which shows power-law. (b) The precision of the residue contact predictions as a function of the number of removing the principle modes of data covariance matrix. These eigenmodes are sorted in descending order of the eigenvalues. It shows that removed some largest principle modes improves contact predictions. These figures are adapted from [44].

the influence of the principal modes, and can enhance correlations caused by spatial interactions.

To compare an empirical distribution and the RBM distribution more quantitatively, KL divergence between them is a reasonable measure. It can be formally written as

$$D_{KL}(p^*\|p^{RBM}) = \sum_{\boldsymbol{A}\in\mathcal{A}^L} p^*(\boldsymbol{A})\Big(\log p^*(\boldsymbol{A}) - \log p^{RBM}(\boldsymbol{A})\Big)\ , \tag{5.1}$$

where p^* is assumed that to be an empirical distribution, and it can be represented as a pairwise Potts distribution with true model parameters $\{J^*_{ij}(a,b)\}$, and p^{RBM} is a probability distribution depending on a small number of hidden variables ($P = O(1)$). Here, we assumed the empirical distribution could be fully described by a pairwise Potts distribution with the set of coupling parameters. If p^{RBM} can adequately learn the correlations related to the principle modes, $D_{KL}(p^*\|p^{RBM})$ in Eq. 5.1 would be a function of the couplings $\{J^*_{ij}(a,b)\}$ that are more likely associated with spatial contacts.

Although Eq. 5.1 is intractable, we may directly assess how KL divergence (KLD) in Eq. 5.1 changes by adding an individual coupling to an RBM distribution. Here we slightly generalize the problem so that the assumed model is not only RBM but also an arbitrary statistical model $p(\boldsymbol{A})$. The difference in the KLD can be written as

$$\begin{aligned}\Delta l &= D_{KL}(f\|p) - D_{KL}(f\|p') \\ &= \sum_{\boldsymbol{A}} f(\boldsymbol{A})\Big(\log p'(\boldsymbol{A}) - \log p(\boldsymbol{A})\Big)\ ,\end{aligned} \tag{5.2}$$

where $f(\boldsymbol{A})$ is an empirical distribution, and $p'(\boldsymbol{A})$ is a model that is added an individual coupling parameter (e.g., $J_{ij}(a,b)$). Hereafter, we refer to Δl in Eq. 5.2 as *likelihood variation*. As we will discuss in the following section, Δl can be obtained analytically as a function of the empirical and the model frequencies (e.g., $f_{ij}(a,b)$ and $p_{ij}(a,b)$).

Note that a larger Δl value means that the added parameter can be more influential to the model $p(\boldsymbol{A})$. Therefore, it is possible to select sparse parameters to add to the assumed model based on Δl.

In the next section, we will formulate the ideas presented here in a more concrete way.

5.2.1 Element-wise coupling activation

The key idea is to estimate relevant couplings based on a likelihood variation Eq. 5.2. In order to formulate it, we first define a modified Hamiltonian,

$$H'(\boldsymbol{A}) = H(\boldsymbol{A}) - J_{ij}(a,b)\delta_{A_i,a}\delta_{A_j,b} \ . \tag{5.3}$$

The old Hamiltonian $H(\boldsymbol{A})$ is an arbitrary function of protein sequences. If $H(\boldsymbol{A})$ is not a pairwise Potts energy function (e.g., a profile model or RBM), the operation in Eq. 5.3 means adding a single coupling parameter $J_{ij}(a,b)\delta_{A_i,a}\delta_{A_j,b}$, where i and a indicate the residue position being $i \in 1{:}L$ and amino acid being $a \in 1{:}q$, respectively. On the other hand, if $H(\boldsymbol{A})$ is a pairwise-Potts energy function, Eq. 5.3 means changing the existing coupling.

Using these $H(\boldsymbol{A})$ and $H'(\boldsymbol{A})$, we can estimate the likelihood variation as following,

$$\begin{aligned}
\Delta l &= \frac{1}{M}\sum_{m=1}^{M}\Big(\log p'(\boldsymbol{A}^m) - \log p(\boldsymbol{A}^m)\Big) \\
&= \frac{1}{M}\sum_{m=1}^{M}\Big(H'(\boldsymbol{A}^m) - H(\boldsymbol{A}^m)\Big) - \log\frac{Z'}{Z} \\
&= J_{ij}(a,b)f_{ij}(a,b) - \log\frac{Z'}{Z} \ ,
\end{aligned} \tag{5.4}$$

where $p(\boldsymbol{A})$ and $p'(\boldsymbol{A})$ are Gibbs-Boltzmann distributions characterized by the Hamiltonians, e.g., $p(\boldsymbol{A}) = \exp(-H(\boldsymbol{A}))/Z$. Similarly, Z and Z' are the partition functions of the Boltzmann measure, e.g., $Z = \sum_{\boldsymbol{A}}\exp(-H(\boldsymbol{A}))$. The two-point frequency $f_{ij}(a,b)$ is defined as a partial marginalization of the empirical frequency, $f_{ij}(a,b) = \sum_{\boldsymbol{A}\in\{1:q\}^L} f(\boldsymbol{A})\delta_{A_i,a}\delta_{A_j,b}$. Note that the ratio of the partition function is tractable and can be easily computed:

$$\begin{aligned}
\frac{Z'}{Z} &= \frac{1}{Z}\sum_{\boldsymbol{A}} e^{-H(\boldsymbol{A})+J_{ij}(a,b)\delta_{A_i,a}\delta_{A_j,b}} = \langle e^{J_{ij}(a,b)\delta_{A_i,a}\delta_{A_j,b}}\rangle_H \\
&= e^{J_{ij}(a,b)}p_{ij}(a,b) + 1 - p_{ij}(a,b) \ ,
\end{aligned}$$

where $p_{ij}(a,b)$ is the marginal two-point probability distribution for the assumed model $p(\boldsymbol{A})$. By plugging the analytical formula of Z'/Z, we get the exact formula of the likelihood variation,

$$\Delta l = J_{ij}(a,b) f_{ij}(a,b) - \log\left(e^{J_{ij}(a,b)} p_{ij}(a,b) + 1 - p_{ij}(a,b)\right) . \tag{5.5}$$

Note that Δl is derived without knowing the function $H(\boldsymbol{A})$. The dependency on $H(\boldsymbol{A})$ comes via the probability $p_{ij}(a,b)$ which we can estimate using MC sampling. All we have assumed is that the probability distribution is only Gibbs-Boltzmann, $p(\boldsymbol{A}) \propto \exp(-H(\boldsymbol{A}))$.

Optimize of the element-wise likelihood variation – Here, we discuss more basic properties of the likelihood variation as a function of the coupling $J_{ij}(a,b)$. For the sake of simplicity, we neglect the indices $(ijab)$:

$$\Delta l(J) = Jf - \log(e^J p + 1 - p) . \tag{5.6}$$

Eq. 5.6 has a simple shape,

$$\Delta l(J) = \begin{cases} -J(1-f) & \text{if } J \to +\infty \\ J(f-p) & \text{if } J \sim 0 \\ -|J| f & \text{if } J \to -\infty \end{cases} . \tag{5.7}$$

This function goes to minus infinity as $|J| \to \infty$ in a linear manner and approaches zero as $J \to 0$ also linearly, hence this function has a strictly positive maximum:

$$J^* = \log\left[\frac{f(1-p)}{p(1-f)}\right] , \tag{5.8}$$

with,

$$\Delta l^* = f \log\frac{f}{p} + (1-f)\log\frac{1-f}{1-p} \geq 0 . \tag{5.9}$$

We refer to J^* as *activating couplings*, which has also been suggested in [99], but we can calculate the optimal value instead of a perturbative calculation.

Note that J^* goes to infinity if p or f is close to zero or one, however that situation can easily be avoided by introducing a pseudo-count for frequencies [1]. We also show another derivation of $\Delta l(J)$ by evaluating the likelihood of

[1] One easily finds that this function becomes $\Delta l(J) \to 0$ as $f \to 0, 1$ for arbitrary J (suppose $f \sim p$).

p given an observation of f (Appendix F.1).

Note that the above arguments can be held whenever an adding parameter is introduced individually. The same result can obtain by introducing other types of parameters such as external field $h_i(a)$ and many-body interactions parameters (e.g., $J_{ijk}(a,b,c)$).

5.2.2 Block-wise couplings activation

Similar to the previous discussion, we can estimate the likelihood change due to adding an entire coupling matrix $J_{ij} \in \mathbb{R}^{q\times q}$.

$$H'(\boldsymbol{A}) = H(\boldsymbol{A}) - J_{ij}(A_i, A_j) \ , \forall \boldsymbol{A} = (A_1, \ldots, A_L) \in \{1{:}q\}^L \ . \tag{5.10}$$

Note that Eq. 5.10 is not an elementwise coupling addition on Eq. 5.3 but blockwise. Accordingly, we get the following relation for the likelihood variation:

$$\begin{aligned} \Delta l &= \sum_{a,b} J_{ij}(a,b) f_{ij}(a,b) - \log \frac{Z'}{Z} \ , \\ \frac{Z'}{Z} &= \langle e^{J_{ij}(A_i,A_j)} \rangle_H = \sum_{a,b} e^{J_{ij}(a,b)} p_{ij}(a,b) \ . \end{aligned}$$

Optimize the block-wise likelihood variation – Here we determine the properties of the likelihood variation in the case of block-wise likelihood variations. In the following analysis we neglect site indices and represent states $(ijab)$ as μ, $J_{ij}(a,b) = J_\mu$.

$$\Delta l = \sum_\mu J_\mu f_\mu - \log \sum_\mu e^{J_\mu} p_\mu \ . \tag{5.11}$$

Here the Hessian matrix of the likelihood variation is,

$$\mathrm{H}_{\mu\nu} := \frac{\partial^2 \Delta l(J)}{\partial J_\mu \partial J_\nu} = \begin{cases} r_\mu r_\nu & \text{if } \nu \neq \mu \\ -r_\mu(1-r_\mu) & \text{if } \nu = \mu \end{cases} \ , \tag{5.12}$$

where $r_\mu = \frac{\exp(J_\mu)p_\mu}{\sum_\nu \exp(J_\nu)p_\nu}$, hence $\sum_\nu r_\nu = 1$ [2].

[2] Thus r_ν can be regarded as a probability measure.

Note that the inner-product for an arbitrary vector $\boldsymbol{a}$ and $\mathrm{H}\boldsymbol{a}$ is

$$\begin{aligned}\boldsymbol{a}^t \mathrm{H}\boldsymbol{a} &= \sum_{\mu,\nu} a_\mu r_\mu a_\nu r_\nu - \sum_\mu a_\mu^2 r_\mu \\ &= \langle a_* \rangle_{\boldsymbol{r}} \langle a_* \rangle_{\boldsymbol{r}} - \langle a_*^2 \rangle_{\boldsymbol{r}} = -\left\langle \left(a_* - \langle a_* \rangle_{\boldsymbol{r}} \right)^2 \right\rangle_{\boldsymbol{r}} \leq 0 \ ,\end{aligned} \tag{5.13}$$

where $\langle a_* \rangle_{\boldsymbol{r}} = \sum_\mu a_\mu r_\mu$. Thus, the block-wise Δl is a convex function of $J \in \mathbb{R}^{q \times q}$ and there is only one maximum.

Therefore, we can find the coupling matrix that realizes the maximum, by solving the saddle point equation,

$$\frac{\partial \Delta l(J)}{\partial J_{ij}(a,b)} = f_{ij}(a,b) - \frac{e^{J^*_{ij}(a,b)} p_{ij}(a,b)}{\sum_{c,d} e^{J^*_{ij}(c,d)} p_{ij}(a,b)} = 0 \ . \tag{5.14}$$

Eq. 5.14 can be easily solved and we get a solution,

$$J^*_{ij}(a,b) = \log \frac{f_{ij}(a,b)}{p_{ij}(a,b)} \tag{5.15}$$

with,

$$\Delta l(J^*_{ij}) = \sum_{a,b} f_{ij}(a,b) \log \frac{f_{ij}(a,b)}{p_{ij}(a,b)} = D_{KL}(f_{ij} \| p_{ij}) \tag{5.16}$$

Note that when the frequency of model $p_{ij}(a,b)$ is given by the profile model i.e., $p_{ij}(a,b) = f_i(a) f_j(b)$, the right most side of Eq. 5.16 becomes the mutual information.

5.3 Applications

This section discusses residue-residue contact predictions using the likelihood-variation methods for both element-wise and block-wise. Assume multiple RBMs with different hidden variables P as generative models to be added coupling parameters.
The RBM patterns are closely associated with the correlations associated with the principal mode ([98, 51] and Sec. 4.3.1), which is significantly influenced by phylogenies [44]. Therefore, it is expected that the residue-residue contact prediction can be improved when correlations of RBM sequences efficiently reproduce non-contact correlations.

Here, increasing the number of hidden variables P is associated with increasing the number of principal modes to be removed from a covariance matrix for residue contact predictions (see Fig. 5.2). The effect of removing the patterns is nontrivial.

The learning protocol is the following. First, we initialize the RBM parameters based on the mfHP patterns as we discussed in Sec. 4.3.1. Second, we learn the RBMs using the standard learning algorithm (PCD-based learning algorithm with Gaussian hidden variables, see Appendix E.2). Lastly, we apply the likelihood-variation methods and obtain the coupling activation values $J^*_{ij}(a,b), J^*_{ij}$ and the maximized likelihood variations $\Delta l(J^*_{ij}(a,b)), \Delta l(J^*_{ij})$ for both elementwise and blockwise cases.

For obtaining the scoring function for contact predictions, we apply different strategies for both element/block-wise and coupling/likelihood-variation methods:

- $J^*_{ij}(a,b) \in \mathbb{R}^1$: For element-wise coupling activation, we simply use Frobenius norm, $F^{\text{E.C.}}_{ij} := \sqrt{\sum_{a,b}(J^*_{ij})^2}$.
- $\Delta l^*_{ij}(a,b) \in \mathbb{R}^1$: For element-wise optimal likelihood variation, we use the sum for the overall state, $F^{\text{E.L.}}_{ij} := \sum_{a,b} \Delta l^*_{ij}(a,b)$ [3]
- $J^*_{ij} \in \mathbb{R}^{q\times q}$: For block-wise coupling activation, we take its Frobenius norm, $F^{\text{B.C.}}_{ij} := \|J^*_{ij}\|$.
- $\Delta l^*_{ij} \in \mathbb{R}^1$: For Block wise optimal likelihood variation, we use that value directly, $F^{\text{B.L.}}_{ij} := \Delta l^*_{ij}$ [4].

Fig. 5.3 shows the contact predictions i.e., positive predictive value (PPV) curves using the coupling activation values for both element- and block-wise cases. For RBMs, we tested multiple different number of hidden variables, $P \in \{0, 1, 2, 16, 32\}$. Note that a $P = 0$ RBM is equivalent to the profile model. For comparison, we also included results using MI and plmDCA. As for the MSA to predict residue contacts, we use the data set of PF00072 employed in Ref. [92].

[3] The validity of this formulation could be that, as in the derivation of the likelihood function, adding two couplings such as $J_{ij}(a,b)$ and $J_{ij}(c,d)$ gives an arithmetic sum if the entropic term is negligible in Eq. 5.5.

[4] By definition, they are non-negative values, $\forall i, j$.

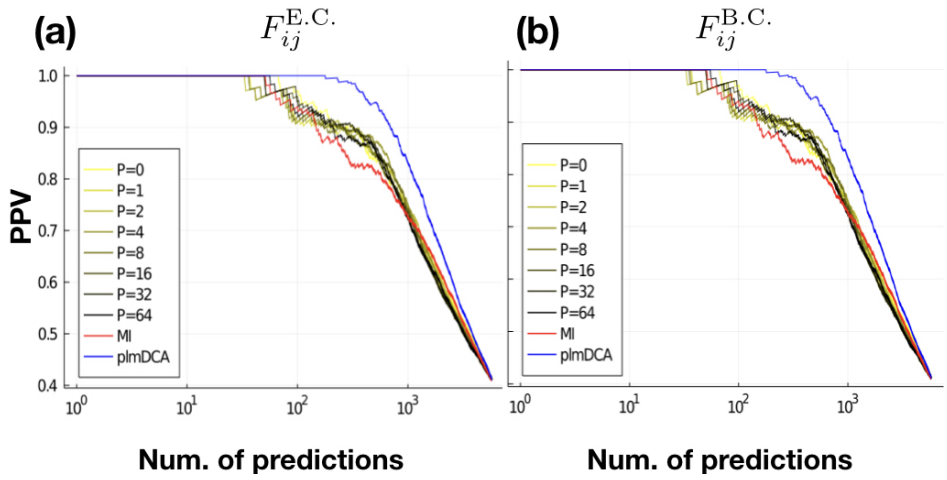

Figure 5.3: (a) PPV curves based on element-wise coupling activation $F^{\mathrm{E.C.}}$. As a model to add coupling parameters, we used RBM with several different P, the number of hidden variables. There is not much difference in the results between different P. the accuracies of these results are between the PPV curve of MI (red line) and PPV curve of plmDCA (blue line), which is the best residue contact prediction from only MSAs. (b) Same as (a) but these predictions are based on block-wise coupling activation $F^{\mathrm{B.C.}}$. These PPV curves are very similar to the results in Fig. 5.3.a.

As shown in Fig. 5.3, residue contact prediction with coupling activation values for element-wise $F_{ij}^{\mathrm{E.C.}}$ and block-wise $F_{ij}^{\mathrm{B.C.}}$ gave almost the same results. In fact, the coupling activation values for each element-wise and block-wise were very similar. By the definition of the coupling activations in Eq. 5.8 and Eq. 5.15, it can be seen that when the two-point empirical distribution $p_{ij}(a, b)$ is close to one or zero (that is, it corresponds to a situation sites where sites i and j are likely to be contact), both values are close to each other.

Fig. 5.4 shows PPV curves using the likelihood variation values for both element-wise $F_{ij}^{\mathrm{E.L.}}$ and block-wise $F_{ij}^{\mathrm{B.L.}}$ cases. They show that increasing the number of hidden variables tends to improve contact prediction. Surprisingly, these relatively simple methods can achieve almost the same accuracy as plmDCA up to the top 300 predictions.

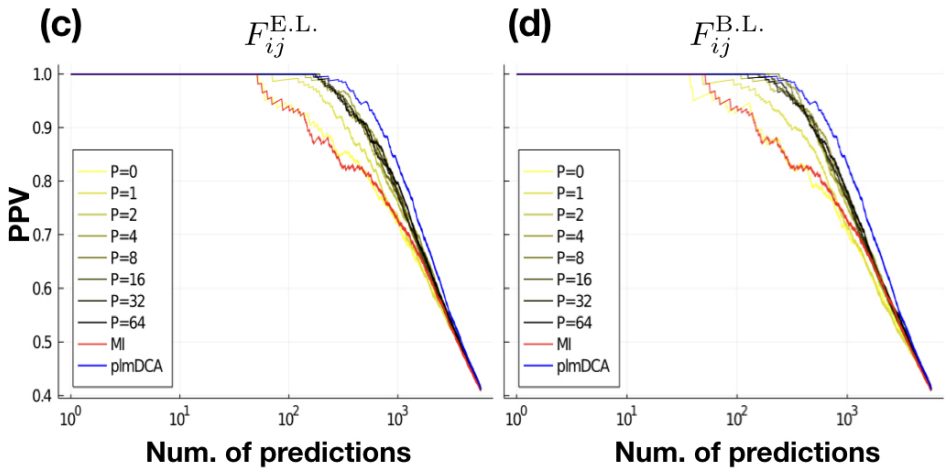

Figure 5.4: (c) PPV curves based on element-wise likelihood variation $F^{\mathrm{E.L.}}$. These residue-contact accuracies are systematically better than the result based on MI (shown as a red line). If P is greater than some values ($P > 4$), predictions based on $F_{ij}^{\mathrm{E.C.}}$ are almost the same with plmDCA results (blue line) until around 300. (d) Same as (c) but for the block-wise likelihood variation $F^{\mathrm{B.L.}}$. These show almost the same curves as the element-wise cases, but the block-wise predictions are slightly more accurate (the point the PPV curve based on plmDCA drops is almost the same point as some of the cases for $F^{\mathrm{B.L.}}$).

Interestingly, adding a small number of hidden variables ($P = 4$) could improve the residue contact predictions, and adding more hidden variables ($P > 4$) did not show any significant improvement in accuracy. The improvement of the residue contact predictions is thought to be because RBMs have specifically learned the pairwise correlation due to the phylogenetic effects of protein sequences. Furthermore, they did not learn the correlation caused by the three-dimensional structure so strongly (Sec. 4.3.1). In Appendix F.2, we show the same analyses for other protein families. Phylogenetic effects are highly dependent on the protein family, and the efficiencies of residue contact predictions based on the likelihood variation also tend to be dependent on the protein families.

5.4 Conclusion

In this chapter, we introduced statistical models that concern both a small number of strong pairwise couplings in sparse BMs and the low-rank couplings in RBMs. These two types of parameters are complementary: the sparse couplings are typically associated with pairs of residues in spatial contact. On the other hand, low-rank couplings tend to learn features of each subfamily and phylogenies.

Sec. 5.1 reviewed Ref. [44], which is about phylogenetic effects on the principal modes of data covariance matrices. Principle modes of covariance matrices tend to carry phylogenetic effects and subfamily specific features, and those eigenvalues are typically following a power-law distribution. Therefore removing the principal modes from the covariance matrices can improve residue-contact predictions (cf., Fig. 5.2).

Sec. 5.2 introduced methods that select variables that are presumably significant variables to describe the training data. These methods allow us to select variables to add to an assumed statistical model by assessing the impact of virtually introducing an individual model parameter (e.g., pairwise coupling parameters $J_{ij}(a, b)$, higher-order interaction parameters $J_{ijk}(a, b, c), ...$, etc.) into the likelihood function of the assumed model. Note that this is an opposite principle to the sparse BM, as we add the couplings instead of decimating them. More specifically, we investigated in detail how to add element-wise and block-wise coupling parameters that increase the likelihood. We can also find another approach to add coupling to a pairwise

model based on pseudo-likelihood. We can also find another interesting research to add coupling parameters iteratively to a pairwise model based on pseudo-likelihood [99].

From the results of Ref. [44] and Sec. 4.3.1, it is expected that RBMs can efficiently learn the correlations that affect phylogenies. Therefore, assuming RBMs as statistical models to which parameters are added, the coupling parameters that increase the likelihood function correspond to pairs in spatial contact.

In Sec. 5.3, we predicted residue-residue contact based on the proposed method using RBMs. Assuming a model with a small number of hidden variables ($P > 2$), residue contact prediction was significantly improved. Further studies are needed to apply these methods to different protein families depending on the strength of the phylogenetic effects.

III

Higher-order statistical modeling

Nature uses only the longest threads to weave her patterns, so that each small piece of her fabric reveals the organization of the entire tapestry.

—Richard Feynman (1918 - 1988)

Chapter 6

Variational Autoencoders for Protein Sequences

Up to this chapter, we have discussed parameter decimation and dimensionality reduction. However, the objective of this chapter is to investigate other possibilities of variables that explain training data more comprehensively than pairwise interactions in stochastic modeling for protein sequences. Therefore, we will explore the possibility of other variables besides pairwise interactions.

Particularly, we aim at investigating the presence of higher-order interactions in natural protein sequences. From now on, the higher-order interactions are defined as interactions between amino acids that include three or more sites in underlying probability distributions for protein sequences. Especially, three-body interactions are considered in Sec. 6.1. Note that we need to distinguish interactions and correlations.

On the other hand, there has been extensive research on protein sequence generative models and protein structure predictions using the state of the art of deep neural networks over the last few years (DNNs) [100, 101, 102].

Notably, DNN-based generative models do not require to decide which variables should be considered in a priori and can take into account higher-order interactions. Variational Auto-Encoder (VAE) is a generative statistical model that can handle DNNs well within the framework of statistical modeling. VAEs have been applied as a protein sequence generative model in recent years [103, 104], and further applications are expected.

The structure of this section is as follows: First, we show a relatively straightforward experiment that studies causes of strong third-order correlations. Second, we investigate the presence of higher-order and non-linear effects by exploiting VAEs.

6.1 Decomposition of strong three-point correlations in two-point correlations

It has been said that there are higher-order interactions in protein-sequence ensembles, and taking them into account is essential for improving properties as a generative model[105, 106, 103]. Besides the computational challenge to include higher-order interactions explicitly in a model, there are also statistical challenges. That is, searching the evidence of the higher-order interaction is like hearing a faint sound of insects at a construction site. Most of the signals from relevant higher-order combinations are still much smaller than the majority of lower-order statistics (single site frequencies and pairwise frequencies).

Instead of directly investigating higher-order interactions, we first explore the nature of higher-order correlations. More specifically, we search three-point correlations not explainable by two-point correlations [1]. Therefore, we focus on strong three-point correlations, especially when the two-point correlations involved in the three-point correlation are not very strong. Such three-point correlations can be defined as follows:

$$\begin{aligned} &|C_{\xi\eta\zeta}| \geq \theta_3^* \quad \text{and,} \\ &|C_{\xi\eta}|, |C_{\eta\zeta}|, |C_{\zeta\xi}| \leq \theta_2^*, \end{aligned} \tag{6.2}$$

[1] More formally, three-point correlations that are explainable by two-point correlations satisfy the following properties:

$$f_{\xi\eta\zeta} = f_{\xi\eta} f_{\zeta} \ . \tag{6.1}$$

We can easily check that the three-point correlation can be written in terms of means and two-point correlations as follows:

$$\begin{aligned} C_{\xi\eta\zeta} &= f_{\xi\eta\zeta} - (f_\xi f_{\eta\zeta} + f_\eta f_{\zeta\xi} + f_\zeta f_{\xi\eta}) + 2 f_\xi f_\eta f_\zeta \\ &= +2 f_\xi f_\eta f_\zeta - (f_\xi f_{\eta\zeta} + f_\eta f_{\zeta\xi}) \\ &= -f_\xi C_{\eta\zeta} - f_\eta C_{\zeta\xi} \ . \end{aligned}$$

Therefore, three-point correlations that satisfies Eq. 6.1 can be written with the two-point correlations and means. Thus, a strong three-point correlation that involves two or more weak two-point correlations must be a different form.

where each index in (ξ, η, ζ) is supposed to specify a position and an amino acid, and $C_{\xi\eta\zeta}$ is the three-point (connected) correlation defined in Eq. 2.12. The θ_3^*, θ_2^* are preselected thresholds for third- and second-order correlations that distinguish significance.

The question raised here is whether there are strong three-point correlations, even if the associated tow-point correlations are small. Concerning this question, we compare the following two types of three-point correlation ensembles:

- ensemble-1: an ensemble for all of the possible three-point correlations.
- ensemble-2: an ensemble of the three-point correlations, where two or more of the three two-point correlations are large (e.g., if $|C_{\xi\eta}|, |C_{\eta\zeta}| > \theta_2^*$, then we include $C_{\xi\eta\zeta}$ in to ensemble-2).

For both ensembles, we treat only three-point correlations where indices are pairwise different ($\xi \neq \eta, \eta \neq \zeta, \zeta \neq \xi$). If there are three-point correlations not explainable by the involved two-point correlations, we cannot find them in ensemble-2.

The two-point correlations that are induced by the background noise can be estimated numerically by taking sequences from a profile model. In fact, the two-point correlations due to the background noise are quite small, 0.0057, 0.0046, and 0.0049, for protein families, PF00014, PF00072, and PF00076, respectively. Hence, we selected two-point correlations whose values are greater than $\theta_2^* = 0.01$ as significant two-point correlations, which correspond to less than 0.3% of all two-point correlations. Based on the selected strong two-point correlations, we determined a set of pairs of indices and states $\Omega_2 := \{(ijab)\}$, then construct all of the possible three-point correlations $\Omega_3 := \{(ijkabc)\}$, which should include at least two out of these three pairs $(ijab), (jkbc)$ or $(kica) \in \Omega_2$.

In Fig. 6.1, we show the distributions of the three-point correlations for the three protein families (the MSAs are those in Ref. [92]). For comparisons, we selected the largest 10^5 correlations as particularly strong three-point correlations (the red histogram, which corresponds to ensemble-1), which is typically less than 0.05% of the total, and three-point correlations whose associated two-point correlations are significantly strong (the blue and green histograms, which correspond with ensemble-2 with different threshold $\theta_2^* = 0.01$ and $\theta_2^* = 0.015$ respectively). It clearly suggests that the

strong three-point correlations occur as a consequence of strong two-point correlations.

6.2 Variational Autoencoders as generative models and non-linear low-dimensional analysis for proteinn sequences

Variational Autoencoders (VAEs) are alternative generative models getting recently considerable attention. VAEs were proposed by Kingma and Welling in 2013 [107]. VAE can be used as a generative model, but also as low-dimensional analysis such as PCA, but in a non-linear manner. Because of the usefulness and flexibility of VAE, they have been applied for designing protein sequences and extract a lot of biologically useful information from sequence alignments [103, 108, 109]. The application of VAEs to a protein sequence generation model has attracted great attention [103, 104].

It has been known that liner systems such as PCA and linear-autoassociate model are not appropriate for certain types of distributions such as multimodal and non-linear distributions, whereas non-linear systems can classify such distributions correctly [110]. Formally, both methods can be understood as reconstruction spaces, one being a linear projection, the other a non-linear projection. In fact, the difference between the two methods becomes clear in terms of reconstruction error. For instance, if there is a non-linear relation in a given data set (suppose the variables are continuous), a linear transformation cannot reconstruct original data [110]. Note that we can find similar situations when dealing with categorical variables, but the non-linear effect in the above example corresponds to multilinear effects, i.e., higher-order interactions. In the case of protein-sequence modeling, therefore, autoencoder-based lower-dimensional analysis might be useful if the distributions are multi-modal or have multilinear effects, not captured by pairwise interactions.

As we mentioned earlier, a VAE is a generative model. Moreover, it can provide a space where the information of raw data (e.g., protein sequences) is compressed or encoded. This space is called a hidden space or feature space. The mapping of a raw data $\boldsymbol{x}$ to hidden variables $\boldsymbol{z}$ is called *encoder*, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, the and reconstruction of the raw data from the encoded hidden variables is called a *decoder*, $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, cf. Fig. 6.2.a (we show more details

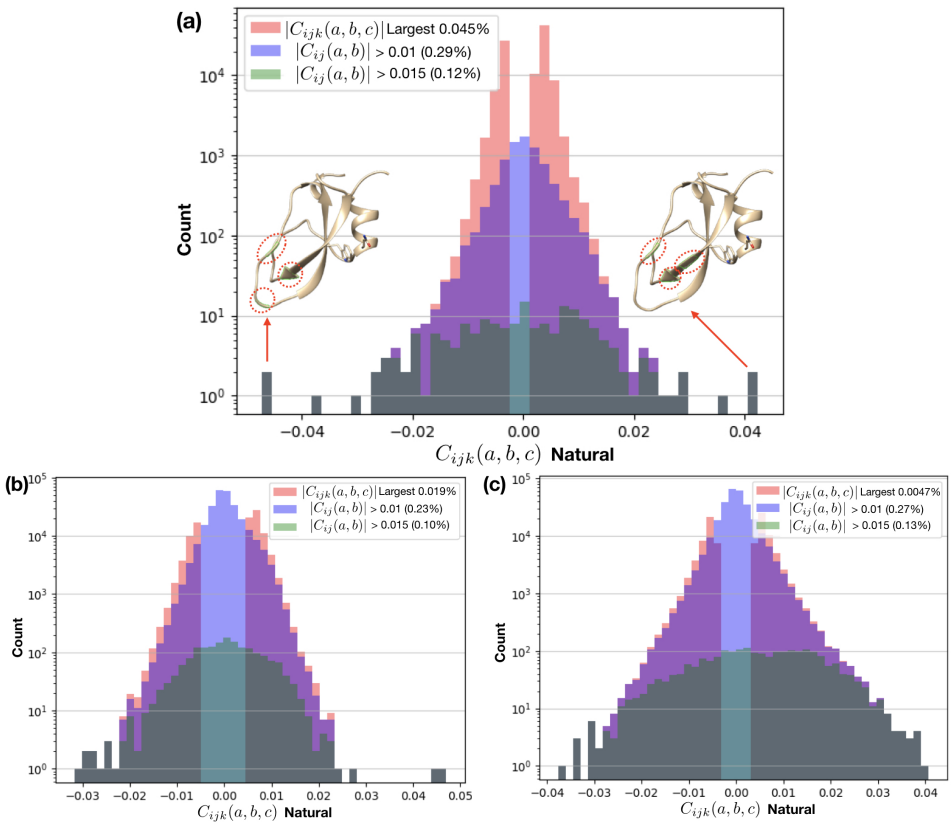

Figure 6.1: (a) Distribution of the three-point correlations for PF00014. We show the largest 10^5 $|C_{ijk}(a,b,c)|$ (red histogram), which corresponds to 0.045% of all three-point correlations (ensemble-1). We also show three-point correlations that are associated with two-point correlations (ensemble-2) whose values are greater than $\theta_2^* = 0.01$ (blue) and $\theta_2^* = 0.015$ (green). The three amino-acid positions corresponding to the largest and the second-largest correlations are highlighted on the 3D structure. The largest positive three-point correlations emerge at $(i,j,k) = (10,32,34)$ with states, (a,b,c) = (G,G,C), (C,G,C). Similarly, the largest negative three-point correlations correspond with $(i,j,k) = (10,14,34)$ with (a,b,c) = (G,L,C),(C,L,C). (b) The same type of figure but for PF00076. (c) The same type of figure but for PF00072.

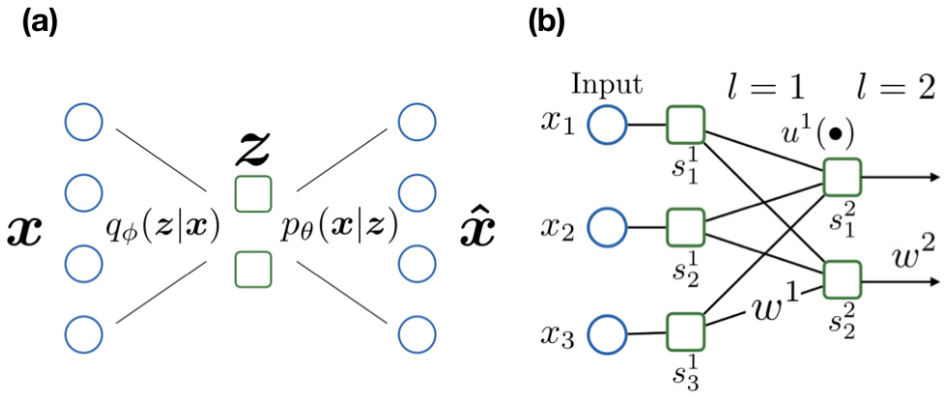

Figure 6.2: (a) Schematic representation of VAE. An encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ transforms an input sample $\boldsymbol{x}$ to a hidden variable $\boldsymbol{z}$. A decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ transforms the hidden variable $\boldsymbol{z}$ to a reconstructed variable $\hat{\boldsymbol{x}}$.

in Sec. 6.3.1). If the dimension of the hidden space is smaller than the dimension of the input data, it acts as a dimensionality reduction or data compression method.

Encoders and decoders are usually built by deep neural networks, as described in the following section (Sec. 6.3.2), but $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ can be treated as probability distributions. Hence, VAEs can be formulated in terms of statistical modeling and also explicitly represent the hidden space as a probability distribution.

The main advantages of VAE for protein-sequence modeling can be summarized as follows: First, there is no assumption such as the pairwise interactions among the amino-acid variables, so possible interactions can be multi-variable interactions. Second, as we mentioned earlier, the distribution in the hidden space is flexibly adaptable considering the characteristics of data.

The organization of this section is as follows:

- In Sec. 6.2.1, we define the objective function of VAE, and discuss how encoder and decoder depend on it.
- In Sec. 6.2.2, we review (deep) neural network algorithms regarding the application to VAEs.

- In Sec. 6.2.3, we apply a VAE as protein-sequence generative models.
- In Sec. 6.2.4, we investigate the statistical properties of the hidden spaces of protein-sequence VAEs.
- In Sec. 6.2.5, we classify protein subfamilies by exploiting the hidden space. Here we also propose another VAE-based algorithm using a generalized prior distribution for hidden variables.

6.2.1 Introduction

As mentioned before, VAEs are defined in terms of statistical models. Indeed, behind VAEs, there is the idea of a (log-) likelihood function $\mathcal{L}(\theta) = \log p_\theta(\boldsymbol{x})$.

One of the most important ideas of VAEs is that, instead of dealing with $p_\theta(\boldsymbol{x})$ directly, it assumes that there is another statistical variable, a hidden variable (we will formally define it in the context of VAEs later) and its marginalized distribution is defined by the Bayesian theorem:

$$p_\theta(\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} \ . \tag{6.3}$$

The benefit of adding an additional stochastic variable is the gain in the complexity of probability distribution. If $p_\theta(\boldsymbol{x})$ is rather a simple model but the empirical distribution aimed to be learned is more complex, learning of parameters θ will be difficult. Simple $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and $p(\boldsymbol{z})$ may lead to complex models, e.g., Gaussian mixtures instead of a simple Gaussian.

The objective function of VAE can be derived as follows:

$$\begin{aligned}
&\log p_\theta(\boldsymbol{x}) \\
&= \left\langle \log p_\theta(\boldsymbol{x}) \right\rangle_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \\
&= \left\langle \log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} \right\rangle_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} = \left\langle \log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})q_\phi(\boldsymbol{z}|\boldsymbol{x})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})p_\theta(\boldsymbol{z}|\boldsymbol{x})} \right\rangle_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \\
&= \left\langle \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \right\rangle_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z})) + D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z}|\boldsymbol{x})) \\
&\geq \left\langle \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \right\rangle_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z})) \ ,
\end{aligned} \tag{6.4}$$

where $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is defined as another conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{x}$. Note that $p_\theta(\boldsymbol{z})$ is an assumed prior distribution, thus given. For the second to the third equation, we used the Bayesian theorem in Eq. 6.3. For the transformation to the final inequality in Eq. 6.4, we used the non-negativity of KLD. A necessary and sufficient condition for establishing equality in the Eq. 6.4 is $D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z}|\boldsymbol{x})) = 0$.

The final result in Eq. 6.4 is a lower bound of the log-likelihood function. It is as objective function of the VAE and called Evidence Lower Bound (ELBO):

$$\mathcal{L}^{\mathrm{ELBO}}(\theta,\phi) = \Big\langle \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \Big\rangle_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\mathbf{z})) \ . \tag{6.5}$$

It has to be maximized with respect to the model parameters θ (for decoder) and ϕ (for encoder). These probabilities can be interpreted in relation to each function of VAE as follows: $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ reproduces a data point $\boldsymbol{x}$ from $\boldsymbol{z}$, hence it stands as a (probabilistic) decoder. On the contrary, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ embeds a data point $\boldsymbol{x}$ to $\boldsymbol{z}$, therefore it works as a (probabilistic) encoder.

ELBO learning consists of two distinctive elements. First, accurate data reconstruction (the first term in Eq. 6.5). Second, minimization of KLD between the encoder and an assumed hidden prior distribution (the second term in Eq. 6.5).

To accurately reconstruct training data, the conditional log-likelihood $\log p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is maximized with respect to θ, after marginalizing the hidden variables. Note that the marginalization is not done by $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, but $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. Therefore, $\mathcal{L}^{\mathrm{ELBO}}$ is a lower bound of the log-likelihood function, and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is its variational function.

As the second element: KLD between the parameterized encoder distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and assumed prior distribution $p_\theta(\boldsymbol{z})$ is minimized by changing the parameter ϕ. In general, factorized Gaussian distribution is chosen for the encoder distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, and standard Gaussian distribution is used for the hidden prior distribution $p_\theta(\boldsymbol{z})$.

Therefore, the idea of ELBO is that a method alternately optimizes between the variational function $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and the lower bound of the log-

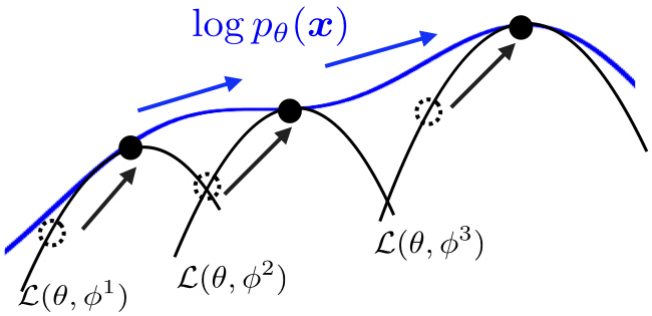

Figure 6.3: Schematic of ELBO learning. The log-likelihood function is bounded by the variational lower bound. Each time optimize variational function via ϕ, then optimize lower bound function via θ.

likelihood (optimization of $\log p_\theta(\boldsymbol{x}|\boldsymbol{z})$ [2] Fig. 6.3.b.

6.2.2 Deep neural networks for VAE

In this section, we will review deep neural networks (DNNs) for VAE briefly and introduce some quantities of VAEs in terms of DNNs.

A neural network (NN) is a machine (algorithm) that has a set of variables called perceptrons or neurons as components and can learn arbitrary functional forms for a certain input data $\boldsymbol{x}$ (we assume $\boldsymbol{x} \in [0,1]^d$). Here, we denote a n-th perceptron at l-th layer as s_n^l, where n being $n \in 1{:}d_l$, (as we defined later, we set $l \in 1{:}l^E$ for the encoder, and $l \in l^E{:}l_{tot}(= l^E + l^D)$ for the decoder). A perceptron s_n^l is multiplied by a weight parameter w_{mn}^l and is summed over all perceptrons in the same layer l, then becomes an argument of another perceptron s_m^{l+1} in the next layer using $u^l(\bullet)$ a non-linear function or activation function, cf., Fig. 6.2.b. As a nonlinear function, we use tanh or Rectified Linear Unit (ReLU) function [111]. So, m-th perceptron located at $(l+1)$-th layer can be written as

$$s_m^{l+1} = u^l\left(\sum_n^{d_l} w_{mn}^l s_n^l\right) \ , \quad \forall l \in 1{:}l^E \ . \tag{6.6}$$

Here, the perceptrons in the first layer are the input values themselves, therefore $d_1 = d$. In the case of protein sequence data, we define each se-

[2] We can find a similar idea in the Expectation-Maximization algorithm.

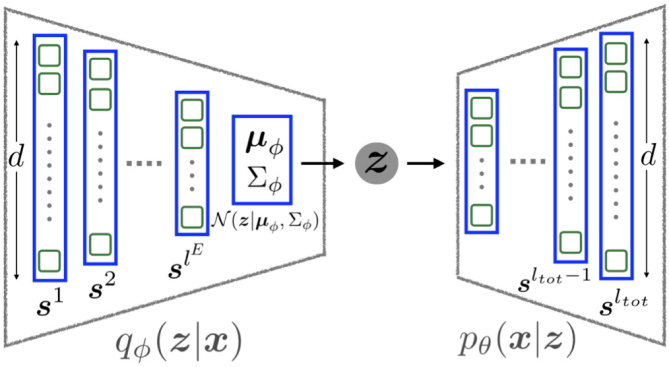

Figure 6.4: An expression of VAE with a factorized Gaussian distribution and DNNs. The left trapezoid represents an encoder, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, which is a DNN parmetrizing a factorized Gaussian distribution. The green squares represent perceptrons. Each blue rectangular is a layer of perceptrons. The perceptrons in the last layer are treated as model parameters of Gaussian distributions, $\mathcal{N}(\boldsymbol{\mu}_\phi, \Sigma_\phi)$. Similarly, the right trapezoid is a decoder. Hidden variables $\boldsymbol{z}$ generated from the $\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\phi, \Sigma_\phi)$ are used as an input variable of the decoder. $\boldsymbol{z}$ is transformed to an output variable through multiple layers of perceptrons. The output variable of the decoder and the input variable of the encoder should have the same dimension.

quence $\boldsymbol{A}$ as a *one-hot* sequence such that

$$x_{(i-1)q+A_i} = \delta_{A_i,a} \ . \tag{6.7}$$

Therefore, a perceptron in the first layer is defined using Eq. 6.7,

$$s^1_n = x_n \ , \quad \forall n \in 1{:}d(= qL) \ . \tag{6.8}$$

From now on, we will discuss DNNs take into account VAE implementation i.e., formally define the decoder and encoder using the DNN language (we will consider a model as shown in Fig. 6.4).

One of the key factors in determining the performance of a VAE is how and what probability distribution is constructed using DNNs. In most cases, the factorized Gaussian distribution is chosen for this purpose because it can well approximate continuous probability distributions to some extent and is theoretically well-grounded (we can analytically obtain the functional form)

[3].

Hence, we define the probability distribution for the encoder as follows:

$$\begin{aligned} q_\phi(\boldsymbol{z}|\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_\phi(\boldsymbol{x}), \Sigma_\phi(\boldsymbol{x})) \\ &= \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-(\boldsymbol{z}-\boldsymbol{\mu}_\phi(\boldsymbol{x}))^t \Sigma(\boldsymbol{x})^{-1} (\boldsymbol{z}-\boldsymbol{\mu}_\phi(\boldsymbol{x}))\right) \ , \\ \Sigma_\phi(\boldsymbol{x}) &= \mathrm{diag}(\sigma_1(\boldsymbol{x}), \ldots, \sigma_P(\boldsymbol{x})) \ , \end{aligned} \tag{6.9}$$

where P is the dimension of the hidden space. The mean values and covariance matrices are defined by perceptrons of the last layer in the decoder DNN,

$$\begin{aligned} \mu_{\phi,n}(\boldsymbol{x}) &= s_n^{l^E}(\boldsymbol{x}) \\ \log \sigma_{\phi,n}(\boldsymbol{x}) &= s_{P+1}^{l^E}(\boldsymbol{x}) \ , \end{aligned} \tag{6.10}$$

where $\forall n \in 1{:}P$. Note that, although we use uniform covariances $\log \sigma_{\phi,n}(\boldsymbol{x}) = \text{const.}, \forall n \in 1{:}P$ (therefore the number of perceptrons in the l^E th layer is $P+1$), we can also assign different perceptrons for each covariance (in that case, the number of the perceptron would be $2P$) .

A hidden variable $\boldsymbol{z}$ is the input variable to the decoder. It is converted to an output variable through the repetition of multiple nonlinear transformations involved in the decoder,

$$\begin{aligned} s_m^{l+1} &= u^l\left(\sum_n w_{mn}^l s_n^l\right) \ , \quad \forall l \in (l^E+1){:}l_{tot} \\ s_n^{l^E+1} &= z_n \ , \quad \forall n \in 1{:}P \ . \end{aligned} \tag{6.11}$$

Note that, in order to have an output variable compatible to an one-hot sequence, we use a Sigmoid function for the nonlinear function at the last layer, $u^{l_{tot}}(\bullet) = \sigma(\bullet) = (1-\exp(-\bullet))^{-1}$. Since these elements are continuous valuables, it is not yet a one-hot sequence. To be an one-hot sequence, the maximum argument for each residue is normally used.

For the learning, we apply the backpropagation algorithm. Other technical information is summarized in Appendix G.1.

Finally, when using it as a generative model, the model parameters are

[3] As we saw the derivation of the ELBO function, however, there are no restrictions on the functional form of $p_\phi(\boldsymbol{z}|\boldsymbol{x})$ (we will discuss this point later in Sec. 6.3.4).

fixed, and only the decoder is used. Since the decoder is assumed to be a conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{z}$, it can generate sequences. For a DNN, it can be done by feeding a random variable (hidden variable) to the decoder neural network. Random variables were sampled from the P-dimensional standard normal distribution $\mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \mathrm{I}_P)$ and used as input values for the decoder. Note that assuming $\mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \mathrm{I}_P)$ for $\boldsymbol{z}$ means, the prior distribution $p_\theta(\boldsymbol{z})$ is a standard Gaussian distribution in Eq. 6.5. In this case, there is no dependency on the model parameters θ, so we will ignore the subscript parameter to the prior distribution.

6.2.3 Protein sequence design using VAE

The first thing that has to be assessed is the capacity of VAE as a generative model, to confirm it as a model that can properly capture protein data. To this purpose, we report here the standard single-site frequencies and two-point connected correlations from VAE. Note that these are not fitted quantities unlike in pairwise-Potts models. In the following experiments, we focus on PF00072 domain data (the same data as in [92]) because of the abundant sequences and available information about distinct protein subfamilies (cf., Sec. 4.3.2).

In this experiment, the architecture we used here is summarized in Table 2. We employed relatively simple architecture for DNNs because as the number of layers increases, required learning processes increase, and learning control also becomes difficult. As a heuristic, we used more layers in the encoder to avoid the *posterior collapse*, which is a well-known problem for ELBO learning and is commonly occurred [112, 113]. Posterior collapse is that the decoder ignores the encoded variables, which corresponds with neglecting the second term in Eq. 6.5. This problem can happen if the encoder is not enough complex and/or encoded signals are less informative.

Fig. 6.5 shows the comparison of correlations with natural sequences (training data) and generated sequences from the VAE and bmDCA. VAE sequences were generated as follows: First, we learned the decoder and encoder based on the ELBO function and backpropagation algorithm. After that, we confirmed that the reconstructed sequences are similar enough to the input sequences (property as an autoencoder). Then, by using a random variable taken from a standard Gaussian distribution $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \mathrm{I}_P)$ as an input variable to the decoder, the decoder can generate artificial sequences. Note that we assumed standard Gaussian distribution as a prior distribution

Encoder

l	d_l	$u^l(\bullet)$
1	$d(=qL)$	$u^l(\bullet) = \mathrm{ReLU}(\bullet)$
2	256	$u^l(\bullet) = \mathrm{ReLU}(\bullet)$
3	256	$u^l(\bullet) = \tanh(\bullet)$
Hidden	128+1	

Decoder

l	d_l	$u^l(\bullet)$
4+1	128	$u^l(\bullet) = \mathrm{ReLU}(\bullet)$
4+2	256	$u^l(\bullet) = \sigma(\bullet)$
Output	d	

Table 2: Table shows the parameters of the architecture of the encoder and the decoder. l indicates the ID of layer. d_l indicates the number of perceptrons contained in the layer. $u^l(\bullet)$ is a nonlinear function (activation function) used in the layer. ReLU($\bullet$) is defined as Rectified linear units function. $\tanh(\bullet)$, is the hyperbolic tangent function. $\sigma(\bullet)$ is the Sigmoid function.

$p(\boldsymbol{z})$ in ELBO learning in Eq. 6.5 .

These two-point correlations are obtained by matrix-matrix multiplication operations using $q \times L$ matrices and did not use the one-hot sequence filter. It shows that statistics come from VAE is almost the same quality with bmDCA in terms of single-site and two-point frequencies.

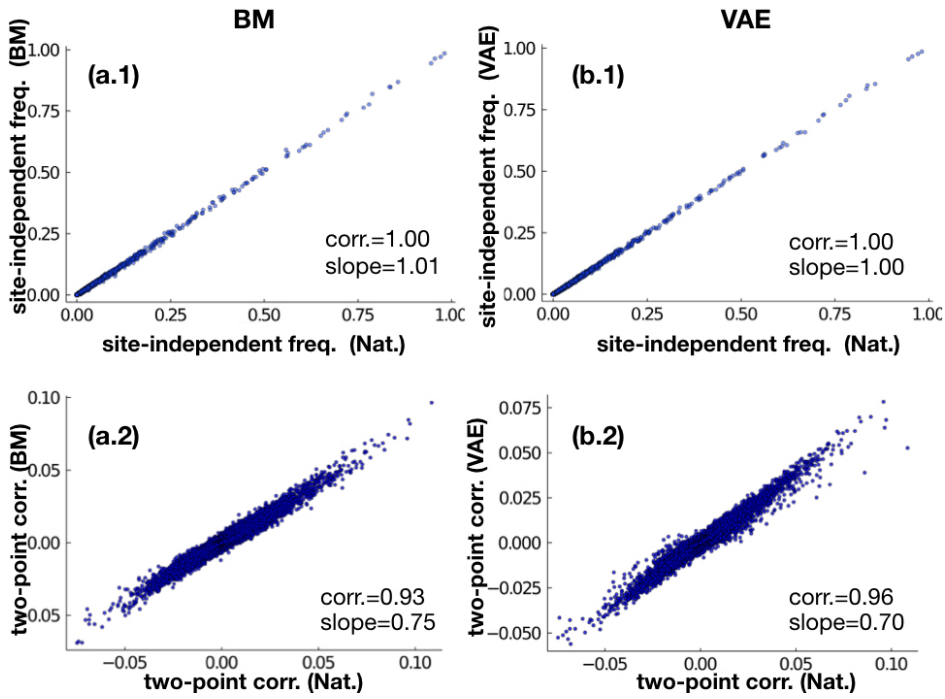

Figure 6.5: (a.1) Comparison of single-site frequencies for natural sequences (horizontal) and natural sequences from bmDCA (vertical). At the bottom right corner, we show Pearson correlation values (denoted as "corr.") and the slope of a linear fitting. (a.2) Comparison of two-point connected correlations for natural sequences (horizontal) and natural sequences from bmDCA (vertical). (b.1) and (b.2) are the sames plot as (a.1) and (a.2) respectively, but using the sequences from VAE (vertical).

6.2.4 Latent space analysis

As mentioned earlier, PCA analysis cannot well capture the underlying data features in cases where these include non-linearities and thus higher-order effects. On the other hand, autoencoder-based low-dimensional analysis can consider non-linear and higher-order effects, and all information of the data point is encoded into the internal hidden space.

The latent space of VAE can encode the underlying statistical features in low dimensions. Hence, the embedded variables are statistically as informative as raw-data space when VAEs are generative.

Moreover, both encoder and decoder are probability distributions. Not only it is theoretically easy to handle, but it also has the following practical advantages: suppose two input data $\boldsymbol{x}^A$ and $\boldsymbol{x}^B$ are given and we want to understand this similarity. A naive idea is a Hamming distance between $\boldsymbol{x}^A$ and $\boldsymbol{x}^B$. Similarly, we can compute the distance between the corresponding hidden variables for both data points, $\boldsymbol{z}^A$ and $\boldsymbol{z}^B$, in the feature space used by VAE. Furthermore, VAE can also enable us to estimate the likelihood of an encoded sequence via $q_\phi(\boldsymbol{z}^A|\boldsymbol{x}^B)/q_\phi(\boldsymbol{z}^B|\boldsymbol{x}^B)$, which can assess how likely a data point $\boldsymbol{x}^A$ belongs to the same ensemble as $\boldsymbol{x}^B$.

As we saw in the last section, the sequence-ensemble of VAE can reproduce the connected correlations of the natural sequences when the learning of VAE is correctly done. Hence we can assume that VAE provide a statistically supported higher-order hidden space. In this section, we exploit the hidden space of VAE. More specifically, in order to investigate statistical properties of protein-sequences and some generative models, we investigate single-point mean and two-point connected correlations in hidden space,

$$\begin{aligned} \psi_u &= \frac{1}{M}\sum_{m=1}^{M} \mu_u(\boldsymbol{x}^m) \\ \psi_{uw} &= \frac{1}{M}\sum_{m=1}^{M} \mu_u(\boldsymbol{x}^m)\mu_w(\boldsymbol{x}^m) - \psi_u\psi_w \ , \quad u \neq w \ , \end{aligned} \tag{6.12}$$

here we omit the symbol of the model parameters ϕ for simplicity. As introduced in Sec. 6.2.2, $\mu_u(\boldsymbol{x})$ is the mean value of u-th hidden variable. Since, $\mu_u(\boldsymbol{x})$ itself is a typical hidden value given $\boldsymbol{x}$, ψ_u represent an ensemble average of hidden values. Similarly, ψ_{uw} is a covariance of typical hidden variables. Since the hidden variables are obtained by repeating the non-linear transformation of the linear combination of perceptrons variables (including amino acids), $\{\psi_u\}$ contains also multilinear effects. Moreover, since the decoder can successfully reproduce the training data, the encoded variable $\mu_u(\boldsymbol{x})$ and $\{\psi_u\}$ should be statistically relevant.

Fig. 6.6 shows comparisons of the natural sequences used for learning the VAE in Sec. 6.2.3, and (a) profile sequences, (b) bmDCA sequences, and (c) VAE sequences (generated using the same condition of Sec. 6.2.3) in hidden space.

In Fig. 6.6.a, a profile model shows strong correlations between naturals sequences in the hidden space. Notably, the hidden-covariancies can show

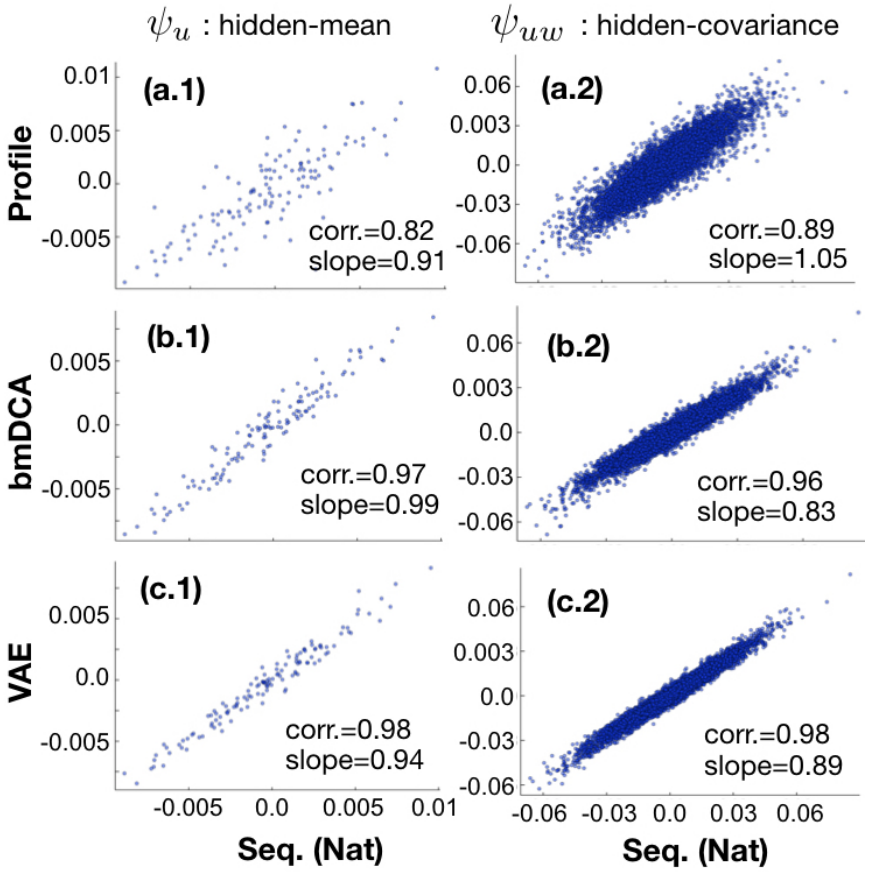

Figure 6.6: (a.1) Comparison of $\{\psi_u\}$, hidden-means between the natural sequences (horizontal) and sequences generated from the profile model (vertical). At the bottom right corner, we also show the Pearson correlation and slope of correlation. (a.2) Comparison of $\{\psi_{uw}\}$, hidden-covariance distribution for both the natural and profile sequences. (b.1, c.1) Same type of plots as (a.1) but comparing bmDCA sequences and VAE sequences (vertical) respectively. (b.2, c.2) Same type of plots as (a.2) but comparing bmDCA sequences and VAE sequences (vertical) respectively.

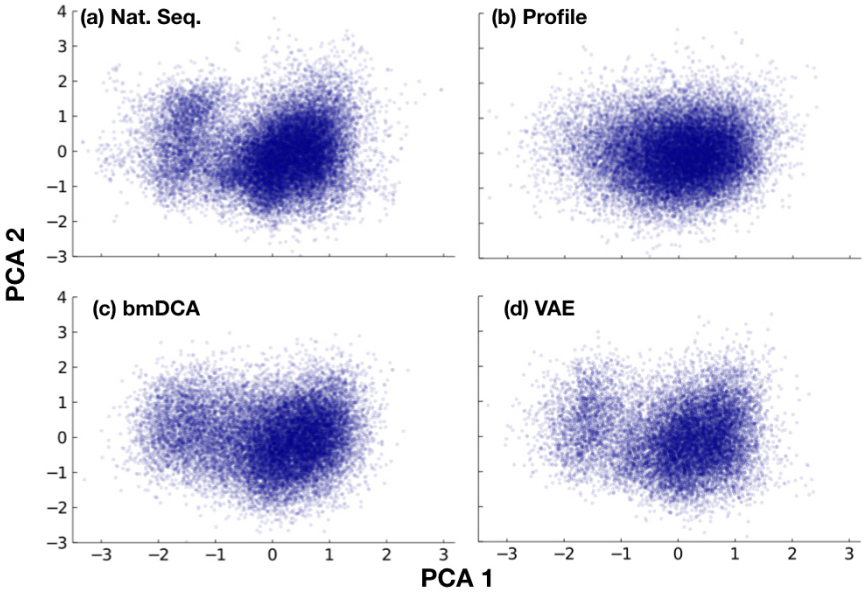

Figure 6.7: Projections of hidden variables in a PCA space of hidden variables of the natural sequence (sequences that were used for learning the VAE). The ensemble sequences projected to the hidden PCA space are: (a) Natural sequences (used in PCA space and learning the VAE). (b) Sequences from a profile model. (c) Sequence form a bmDCA. (d) Sequences from the VAE, which also used for providing the hidden space.

strong correlation even if profile models cannot generate strong correlations at all in terms of two-point connected correlations in sequence space. This result is not surprising, because there are strong influences from the single-site frequencies $f_i(a)$ in ψ_{uw} due to the multiple non-linear transformations in the DNN.

In Fig. 6.6.b and Fig. 6.6.c, we show the comparison between natural sequences, bmDCA and VAE. It shows that the reproducabilites of statistics using the VAE is better than the bmDCA in the hidden space, it is clear by comparing the bmDCA and VAE in terms of ψ_{uw}. However, it is not very astonishing because the model generates sequences, and the model provides the hidden space is the same VAE model, therefore the features in hidden space are characteristic variables for the VAE, which might be different from the important variables for bmDCA (single site frequencies and pairwise fre-

quencies).

The more important observation here is that bmDCA produces statistics almost as good as the VAE in hidden space. As supplementary quantities to assess the statistics, we also examine the distributions of hidden variables using PCA. The PCA space is constructed by the ensemble of hidden variables for the the natural sequences that are used for learning.

In Fig. 6.7, we show the projections of hidden variables to PCA space. The distribution of the profile model (b), does not have a cluster structure and is different from the case of the natural sequences (a). On the contrary, the distributions of bmDCA (c) and VAE (d) can reproduce the similar distribution of the natural sequence in the hidden space.

6.2.5 Protein sequence classification

In general, VAEs use a factorized Gaussian distribution for $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and assume a prior distribution $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathrm{I}_P)$ (hence we omit the parameter symbol θ in the prior distribution). Therefore the decoder can generate samples when fed with statistical variables from the standard Gaussian distribution. However the choice of the standard Gaussian distribution as a prior distribution is not a fundamental constraint of VAE. Relaxing the conditions of the standard Gaussian distribution as a prior distribution does not conflict with the derivation of the ELBO function.

It is a rather reasonable choice to assume a factorized Gaussian distribution for $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ to describe latent space, which contains continuous variables. Due to the conditioning by $\boldsymbol{x}$, assuming the factorized Gaussian distribution result effectively in a mixture of factorized Gaussian distributions. However, when assuming a standard Gaussian distribution as the prior distribution, the distribution of $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ would be simple and the mean values tend to be zero because of the minimization of the KLD between $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z})$ in Eq. 6.5.

A problem that occurs frequently in VAE-based classification problems, is that the hidden variables are projected into approximately the same area and that their distribution does not show a clear structure. In fact, it is an anticipated result since when we assume the standard Gaussian distribution as a prior distribution $p_\theta(\boldsymbol{z})$. This problem could be more pronounced for some data structures, such as one-hot encoded sequences. This tends to make VAEs spend a lot of effort learning features of specific data-structure (e.g., one-hot sequence). Differences between sub-ensembles will be more ambiguous when compared to differences due to data structures.

VAE with structured prior distribution –

Here, it is assumed that an encoder and a decoder are learned initially using standard VAE learning. We introduce additional hidden variables, $c \in 1{:}K$ (K is any positive integer), in order to make the hidden prior be more complex distribution. To do this, we assume the following hidden prior distribution,

$$p_\theta(\boldsymbol{z}, c) = p_\theta(\boldsymbol{z}|c)p_\theta(c) \ , \tag{6.13}$$

where $p_\theta(c)$ is a categorical distribution over $1{:}K$ and $p_\theta(\boldsymbol{z}|c)$ stands for a non-zero mean factorized Gaussian distribution,

$$p_\theta(\boldsymbol{z}|c) = \mathcal{N}(\boldsymbol{z}|\tilde{\boldsymbol{\mu}}_c, \mathrm{I}_P) \ . \tag{6.14}$$

Here, $\tilde{\boldsymbol{\mu}}_c$ is a parameter vector defined in the hidden space, which we will discuss it later in detail. In this manner, we modified the Decoder. Similarly, the encoder changes in the following fashion:

$$q_\phi(\boldsymbol{z}, c|\boldsymbol{x}) = q_\phi(c|\boldsymbol{z}) q_\phi(\boldsymbol{z}|\boldsymbol{x}) \ . \tag{6.15}$$

The factorized distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is the same as the standard encoder distribution defined in Eq. 6.5. The other distribution, $q_\phi(c|\boldsymbol{z})$ is a probability distribution, which generates a value for c given $\boldsymbol{z}$.

From here, we will discuss how to obtain the probability distributions within the framework of VAE we introduced earlier.

Let us make a first point about $p_\theta(\boldsymbol{z}|c)$, which generates $\boldsymbol{z}$ given c from a Gaussian distribution with a mean $\tilde{\boldsymbol{\mu}}_c$, as shown in Eq. 6.14. We define $\tilde{\boldsymbol{\mu}}_c$ using the information of the distribution in hidden space at the previous learning step:

$$\tilde{\boldsymbol{\mu}}_c^{t+1} = \left\langle \int \boldsymbol{z} p_{\theta^t}(\boldsymbol{z}|c) \ d\boldsymbol{z} \right\rangle_{Data} \ , \tag{6.16}$$

were the superscript t indicates a learning epoch. Note that the data dependency result from c and $\boldsymbol{z}$. As the initial states of the mean vectors $\tilde{\boldsymbol{\mu}}_c^{t=0}, c \in 1{:}K$, we assign random vectors drawn from a standard Gaussian distribution, $\tilde{\boldsymbol{\mu}}_c^{t=0} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}|0, \mathrm{I}_P)$.

Second, we define one of the factorized distribution, $q_\phi(c|\boldsymbol{z})$. As in the previous discussion, we take an iterative strategy using the previous state of model parameters,

$$c^{t+1} = \underset{c' \in 1:K}{\operatorname{argmax}} \ q_{\phi^t}(c'|\boldsymbol{z}) \ , \tag{6.17}$$

i.e., each $\boldsymbol{z}$ is assigned to the most probable cluster c . As for the specific mathematical form of $q_\phi(c|\boldsymbol{z})$, we use a Gaussian distribution with the mean vector,

$$q_\phi(c|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\tilde{\boldsymbol{\mu}}_c, \mathrm{I}_P) \ , \tag{6.18}$$

As an initial state of $c^{t=0}$, it is chosen from a Gaussian distribution $\mathcal{N}(\boldsymbol{z}|\tilde{\boldsymbol{\mu}}_c^{t=0}, \mathrm{I}_P)$, hence the categorical variables are assigned uniformly.

Note also that this iterative optimization of $\{\tilde{\boldsymbol{\mu}}_c, c\}$ defined in Eq. 6.16, Eq. 6.17 and Eq. 6.18, is equivalent to the K-means clustering algorithm, an supervised classification algorithm.

Last, in order to assess and control effects of the assumed generalized prior distribution for hidden variables, we introduce a control parameter $\beta \in [0, 1]$ that changes the importance of the conditional log-likelihood term and the KLD term in Eq. 6.5. Note, a VAE including such a parameter β is known as β-VAE [114]. Finally, we define the VAE objective functions with the generalized prior distribution regarding all the quantities we defined here,

$$\begin{aligned}\mathcal{L}_{\beta,K}(\theta, \phi) = (1-\beta)\Big\langle \log\ p_\theta(\boldsymbol{x}|\boldsymbol{z}, c)\Big\rangle_{q_\phi(\boldsymbol{z},c|\boldsymbol{x})} \\ - D_{KL}\left(q_\phi(\boldsymbol{z}, c|\boldsymbol{x}) \| p_\theta(\boldsymbol{z}, c)\right)\ .\end{aligned} \tag{6.19}$$

By definition, as we increase β, we can enhance the effect of the non-zero mean Gaussian distribution via the KLD.

This approach is closely related with the conditional VAE (CVAE), which is a semi-supervised learning algorithm using label information (cf., Appendix G.1.3). The major difference is, although this method can use labeled variables, it is not necessary; we perform unsupervised learning, whereas CVAE is a semi-supervised learning algorithm.

Moreover, the cluster information of c has in effect via the prior distribution of hidden variables $p_\theta(\boldsymbol{z}, c)$. However, CVAE introduce additional perceptrons to the encoder and decoder to take into account the label information, and the dependency of the label is absorbed in the DNNs. Hence, how c depends analytically on the probability distribution is not clear.

In the next section, we will see the results of the classification of protein sequence subfamilies based on this method, and compare it to an existing functional protein classification based on domain architecture, as already used in [97].

Results

Here we apply this method to the classification of protein domain subfamilies, response regulator domain (Pfam ID, PF00072). The data set is the

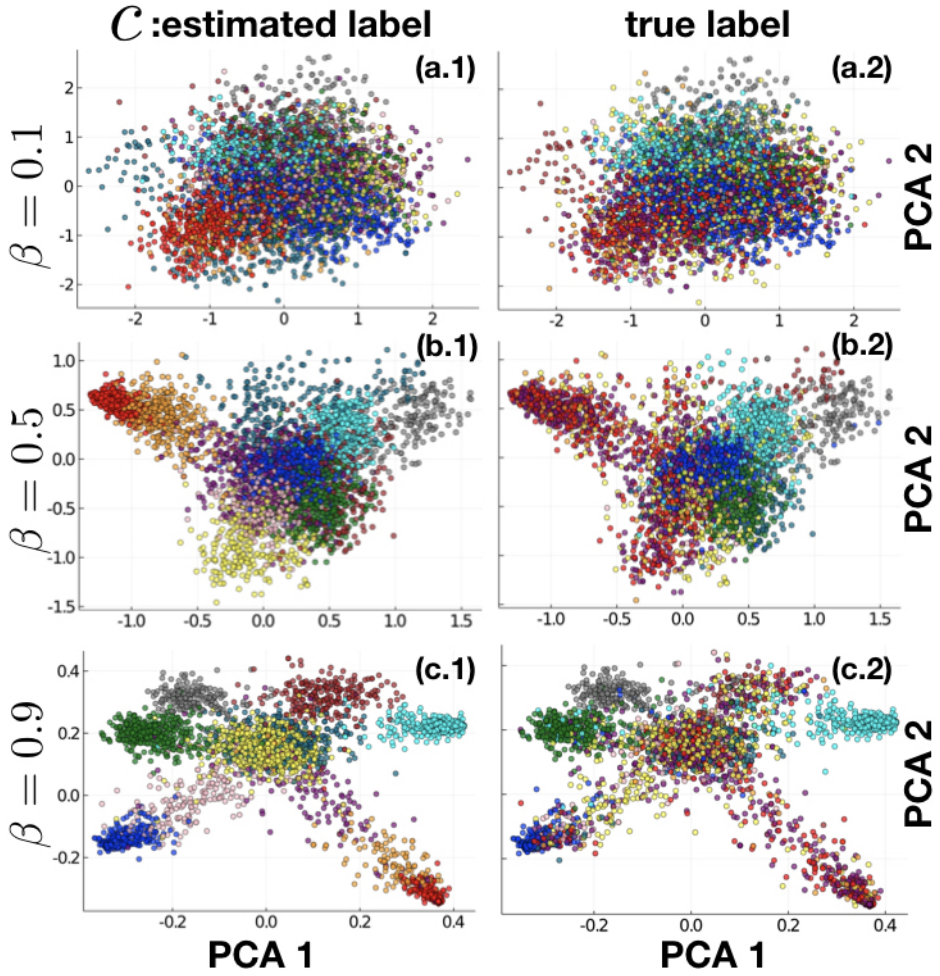

Figure 6.8: Projection of training sequences to the hidden PCA space. using the non-mean Gaussian hidden prior. Show several conditions of β, $\beta = 0.1$ (a), $\beta = 0.5$ (b), and $\beta = 0.9$ (c). As beta increases, the constraint of the KLD increases, thus a stronger effect of the non-mean Gaussian hidden prior distribution. In the left column, colors are assigned using the hidden variable c, which is obtained during VAE learning (a.1, b.1, and c.1). In the right column, on the other hand, colors are assigned using the true labels, therefore classifying based on the information of protein subfamilies (a.2, b.2, and c.2).

same as in Sec. 4.3.2 (see also Table 1 in Sec. 4.3.2).

In this experiment, the architecture of the VAE is the same as Table 2 in Sec. 6.2.3. As mentioned earlier, we assigned initial states of mean vectors $\tilde{\boldsymbol{\mu}}_c^{t=0}, c \in 1{:}K$, from a standard Gaussian distribution. For the initial class label, we assigned the true labels (correspond with the protein subfamilies) to make it easier to compare with the true labels. However, the classification properties, such as the cluster structure of hidden variables and assignment of the label did not change much by the initial state of the labels.

Fig. 6.8 shows results of sequence projections in to hidden PCA spaces, which are constructed form the hidden variables of training data (including all of the protein subfamilies). To demonstrate the effect of the constraint on the hidden prior, we examined several, $\beta \in \{0.9, 0.5, 0.1\}$. Sequences are colored according to their VAE-estimated label (left) or the true label (right). Note that the sequences of the PF00072 domains themselves were not used for the labeling but for the other adjacent domains in the same protein.

As we increase β, i.e., we impose stronger constraints on the conditional posterior for hidden variables, the hidden spaces show more and more complex structures, and clusters become more distinctive. Some subfamilies are correctly classified in any β, but some subfamilies always seem to be difficult to classify. Probably these protein subfamilies are inherently very similar.

When $\beta = 0.1$ (a.1 and a.2), the constraint on the hidden space is very weak by this construction. Therefore the distribution of hidden variables is similar to that pre-learned with standard VAEs, cf., Fig. 6.7. As β increases, the heterogeneity of the hidden-variable distribution increases.

On the other hand, when the hidden space is imposed greater constraint, i.e., $\beta = 0.9$ (c.1 and c.2), distribution of the hidden variable projections becomes more inhomogeneous and demonstrates characteristic structures. In the case of the labels that are estimated by the VAE (c.1), label (color) assignments differ from projected area to area; it shows the one-to-one correspondence between label types and regions. However, in the case of the true labels (c. 2), some regions involved hidden variables that are assigned different labels, i.e., those labels are indistinguishable in the hidden space.

Interestingly, the subfamilies projected onto the characteristic regions are the same as the low dimensional analysis based on RPM and mfHP,

cf., Fig. 4.7. That is, subfamilies that are indistinguishable in RBM-based low-dimensional space are also indistinguishable in VAE hidden space. For example, Class PF00072-PF00512 (purple), PF00072-PF00512(after 500 aa) (red) and PF00072-PF00512(after 1000 aa) (orange) are projected into same areas.

6.3 Conclusion

In this chapter, we investigated the effects of higher-order statistics/interactions in protein sequence data. We also discussed the Variational Auto-Encoder (VAE) as a generative model considering the influence of higher-order interactions.

Sec. 6.1 discussed the presence of significant three-point correlations that are not explainable by the two-point correlations as a model-independent analysis. Our experiments found that all significant three-point correlations always involve more than two significantly large two-point correlations. Although further numerical experimental evaluation is required, it would be interesting to directly compare large three-point correlations with a combination of two-point correlations and means as a supplementary study.

In Sec. 6.2, we examined VAEs as protein-sequence generative models that can consider higher-order interactions. Sec. 6.2.1 reviewed the principle of VAEs which are formulated by the objective function, Evidence Lower Bound (ELBO). We also confirmed the relations between the main components of VAE: encoder, decoder, and hidden space. In Sec. 6.2.2, we defined deep neural networks (DNNs) formally and explained how to construct probability distributions based on DNNs. In Sec. 6.2.3, we demonstrated an experiment of generative model for protein sequences based on a VAE (cf. Table 2). VAE can reproduce statistics of a protein family (PF00072), both single-site frequencies and two-point connected correlations are as good as the bmDCA (cf. Fig. 6.5).

In Sec. 6.2.4, we investigated statistical properties in the hidden variable space, defined by variables including higher-order effects. For this investigation, we exploited the hidden space provided by the experiment in Sec. 6.2.3. Note, since the decoder of the VAE can reproduce statistics accurately from encoded variables, the hidden space provides statistically significant variables. Notably, generated sequences from bmDCA can reproduce hidden

statistics (hidden-means and hidden-covariance) accurately (cf. Fig. 6.6). In other words, no significant difference can be found between the natural protein sequences and the sequence generated from the pairwise coupling model, even in the space where non-linear and higher-order interaction effects exist.

Sec. 6.2.5 proposed a VAE method based on structured prior distributions, based on the insight that distributions of hidden variables tend to be unimodal Gaussian distributions, which may prevent maximizing efficiencies of VAEs encoding and decoding. Here, the structured prior distribution is defined as a Gaussian mixture distribution with different means that are expected to reflect underlying data clusters.

As expected from the construction of the structured prior distributions, characteristic structures of hidden variable distributions were enhanced (cf. Fig. 6.8). Protein subfamilies that were distinctive in the results of low-dimensional analysis using the Hopfield-Potts model (cf. Fig. 4.7 in Sec. 4.3.2) have emphasized their cluster structures more in this VAE hidden space (c.2). However, some protein subfamilies still share areas, that is, such protein families are intrinsically indistinguishable. The same type of experiment should be conducted on more protein families to ensure the reproductibility.

IV

CONCLUDING REMARKS

Thanks to the development of next-generation sequencing machines, technology for extracting meaningful information from vast amounts of genomic data has become increasingly important over the last years. Consequently, machine learning technologies that can extract useful information concerning data without understanding the physical mechanisms of the data become more and more necessary. Such technologies as Deep neural networks, machine learning have come to astonishing results in structural biology in the last years [100].

While the developing machine learning technologies are crucially important, the traditional modeling approaches that aware of understanding the physical phenomena behind the data are just as crucial for further evolutions of biology and genomics.

In this dissertation, we discussed protein sequence generation models based on statistical modelings and machine learning methods. Especially, we focused our studies on constructing biophysically understandable statistical models and selecting variables that are inherently important for describing protein sequences. Chapters 3, 4, and 5 are based on DCA [42, 45].

Chapter 3 aimed to construct minimally constraint pairwise Potts models so that the remaining coupling parameters correspond to structural contacts by decimating coupling parameters. This sparse pairwise Potts models can remove more than 90% of coupling parameters without degrading the statistical properties expected as generative models. Moreover, even though coupling parameters are removed by more than 95%, the accuracy of residue-residue contact is maintained [92].

Chapter 4 investigated the HP models, a model that imposes a low-rank structure on pairwise coupling parameters. These HP models can reproduce the statistics reasonably well, even with about 95% reduction in model parameters. Furthermore, the HP model can provide low-dimensional spaces that are able to capture the characteristics of protein subfamilies [51, 115, 97].

Chapter 5 proposed methods for selecting statistically significant variables that increase the likelihood function for any generative model. Using these methods, we specified pairwise coupling parameters to enhance the likelihood of HP models, thus combining sparse couplings and low-rank couplings. Note that the sparse couplings tend to associate with spatial contacts, whereas low-rank couplings typically learn global correlations that are supposed to be due to phylogenies. Therefore, this chapter served as

further analysis to distinguish correlations that are due to spatial contacts or phylogenies in protein sequences [44].

Lastly, chapter 6 explored the presence of higher-order statistics. Section 6.1 investigated significantly large three-point correlations that are not explainable by the two-point correlations. It revealed that all large three-point correlations are consequences of the large two-point correlations in our experiments. In section 6.2, we employed variational autoencoders (VAEs) [107, 116] as protein sequence generative models to consider nonlinear and higher-order effects. We proposed a framework to investigate higher-order effects in protein sequences by exploiting hidden spaces of VAEs. According to these methods, the protein sequences generated from the pairwise Potts model were not statistically significantly different from the natural protein sequences. In other words, up to second-order interactions are sufficient as a generative model of protein sequences.

The last two chapters need further researches in the future. For chapter 5, we should examine generative models that are re-trained after learning sparse spatial couplings and low-rank couplings. As a result of the relearning, these generative models could improve residue contact prediction further. Chapter 6 proposed the idea to exploit VAE hidden spaces for verification of statistical reproducibility of generative models. This direction has just been opened, many fundamental questions to be addressed are leaving. Some of the intriguing questions are: Can we construct a situation in which the hidden variables of VAE can have physical meanings? Can we understand VAE hidden spaces by comparing them with other hidden variable models, such as the HP models? There many things to be done.

Appendix A

Databases and Data Format

A.1 Database

Pfam – Pfam contains multiple sequence alignments (MSAs) of protein domain families and their hidden Markov models (HMMs). E-value (*expect value*) is also calculated according to an MSA construction [117]. Pfam consists of the following six types of sequences: family, domain, repeat, motif, coiled-coil, and disordered.

To make a large MSA, it needs a curated small number of sequences pre-aligned (the number of sequences is around 100). Based on an MSA that contains a small number of alignments, construct a HMM to search reference protein sequences from external databases such as UniProt. The full alignments can improve the HMMs further by reselecting the seed alignments accordingly, then repeating the same process using the new HMMs. This iterative construction of HMMs and seed alignments are repeated until there is no new entry of sequences from the database.

UniProt – Universal Protein resource (UniProt) [20] is a comprehensive resource of protein sequences and annotated data such as types of proteins and taxonomy. It has been managed by Swiss Institute of Bioinformatics (SIB), European Bioinformatics Institute (EBI), and Protein Information Resource (PIR).
UniProt contains the following databases;

1. UniProtKB: a central hub for collections of functional protein information. These entries come from coding sequences (CDS) that are

also submitted from the EMBL-Bank/GenBank/DDBJ nucleotide sequence resources (International Nucleotide Sequence Database Collaboration (INSDC)).

2. UniProtKB/Swiss-Prot: a high-quality manually annotated [1] and non-redundant protein sequence database. All registered sequences are reviewed. It can also contain experimental conditions, functional information, and mechanical and thermodynamic features determined *in silico*.

3. UniProtKB/TrEMBL: a protein sequence database, which contains automatically annotated information, and these are unreviewed information. It also contains automatically generated annotations and functional characterization.

PDB – Protein Data Bank (PDB) is a repository of information regarding three-dimensional biological molecules such as proteins and nucleic acids. It can provide information about 3D structures of molecules, and experimental conditions that are used to determine the coordinates of atoms in molecules [118]. Protein structure information is obtained by X-ray crystallography, NMR spectroscopy, or cryo-electron microscopy. Structural/ functional information for proteins is maintained by a web-based data server (e.g., PDBe [11], PDBj [119], and BMRB [120]).

The majority of protein structures are determined by X-Ray crystallography (around 146,000 entries) [118]. The second and third most commonly used protein structure determination methods are NMR (around 13,000 entries) and electron microscopy (around 5,000 entries). Only 500 entries are determined by other methods.

X-ray crystallography is almost inapplicable for some types of proteins due to technical and experimental limitations. The main factor is the difficulty in creating protein crystals of sufficient quality for analysis. For example, large proteins and membrane proteins are extremely difficult to make protein crystals in general. Therefore, some classes of protein structure information are significantly insufficient.

[1] Here, manual annotation means reviews of experimentally proven facts or computationally predicted features.

A.2 Data Format

In general, protein sequence files we use are in FASTA format. In the FASTA format, sequences are represented as a series of lines, each of which is no longer than 80 characters. The first line of each protein sequence in a FASTA file, which is also called the header, starts as a later ">", then follows a unique name and/or a unique identifier of the sequence, and there may be additional information. NCBI provides identifiers for protein sequences. The after first lines are the series consist characters that correspond with one of 20 amino acids or gap "–" . Note that the character " Z " means glutamic acids, "X" means any amino acids, and "*" means translation stop.

Appendix B

Other modeling techniques

B.1 Sequence gap filtering

Amino acid sequences tend to have consecutive gap symbols at the beginning and the end of sequences typically. These gaps cause some learning problems. Especially quantities that directly depend on gaps cause slow learning convergences and unstable learnings [97]. A method to remedy this problem is to exclude sequences that contain too many consecutive gaps. We typically keep sequences that have less than six consecutive gaps.

B.2 Initialization of model parameters

The reasonable choice of initial states for the local-field parameters $\{h_i(a)\}$ in Eq. 2.8 are solutions of the profile model :

$$f_i(a) = p_i^{profile}(a|h_i^*) \propto \exp(h_i^*(a)) \ . \tag{B.1}$$

Thus, we can assume the local-field parameters as following

$$h_i(a) = \log(f_i(a)) + \text{const.} \ , \tag{B.2}$$

Empirically, solutions of $\{h_i(a)\}$ in bmDCA in Eq. 2.8 do not change much after the learning when these parameters are initialized based on Eq. B.2.

Especially, efficient initializations for coupling parameters is needed because learning of $\{h_i(a)\}$ are much faster than $\{J_{ij}(a,b)\}$. Furthermore, solutions of bmDCA depend heavily on initializations in practice [90, 92], even if optimizations of pairwise Potts models, including bmDCA are convex prob-

lems. Indeed, bmDCA initialized by small couplings and another bmDCA initialized by the plmDCA couplings converge to different couplings, especially small couplings after the learning are substantially different.

One of the reasons is that the objective functions of pairwise Potts models involve an almost flat direction in the space of coupling parameters. Therefore, they tend to need long learning epochs to satisfy the fixed point equations Eq. 2.10 using gradient-based optimization methods. The situation becomes more severe when the number of model parameters is significantly greater than the number of training data, which is the typical case of our problems.

We typically use small random parameters as initial conditions of coupling parameters because small parameters correspond to a high-temperature regime and thus the mixing time is generally fast and the ergodicity breaking does not occur.

One could use optimized coupling parameters using CD-based learning as initial conditions of the coupling parameters (the similar idea to optimize bmDCA can be found in [57]).

Appendix C

Hopfield-Potts model and Restricted Boltzmann machines

Here, we summarize the solutions of mean-field Hopfield-Potts (HP) patterns based on Ref. [95].

The key steps for obtaining the mfHP solutions are as follows:
(1) Apply the Harvard Stratonovich transformation by introducing continuous variables to unwind interactions between categorical variables for the partition function. (2) Execute Gaussian integration, including the effects of second-order fluctuations, i.e., variance. (3) Fix the gauge to remove redundant gauge freedom due to rotational invariance in the space of the pattern (cf., Eq. 4.4),

$$\sum_{i,a} f_i(a)\xi_i^\mu(a)\xi_i^\nu(a) = 0 \ , \ \mu \neq \nu \quad . \tag{C.1}$$

These arguments lead to the following equation of log-likelihood function

of mfHP model,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\xi}^{1:P}|\boldsymbol{A}^{1:M}) = &\sum_{i}^{L}\sum_{a}^{q} f_i(a)\log f_i(a) \\ &+ \frac{1}{2L}\sum_{i,j,a,b,\mu} \xi_i^\mu(a) C_{ij}(a,b)\xi_j^\mu(b) \\ &+ \frac{1}{2}\log\left(1 - \frac{1}{L}\sum_{i,a,b,\mu}\xi_i^\mu(a)C_{ii}(a,b)\xi_i^\mu(b)\right) ,\end{aligned} \tag{C.2}$$

where C is covariance matrix $C_{ij}(a,b) = f_{ij}(a,b) - f_i(a)f_j(b)$.

Since Eq. C.2 is a convex function of the pattern from the equation, the maximum likelihood estimation solution can be obtained easily. There are two types of solutions, depending on the eigenvalues of the Pearson covariance matrix. Here, Pearson covariance matrix is defined as follows,

$$\Gamma_{ij}(a,b) = \frac{C_{ij}(a,b)}{\sqrt{f_i(a)f_j(b)}} . \tag{C.3}$$

Suppose we can easily get eigenvalues and eigenvectors as follows,

$$\Gamma\boldsymbol{v}^\mu = \lambda_\mu \boldsymbol{v}^\mu . \tag{C.4}$$

It can be written as follows using the eigenvalues and eigenvectors of the Pearson covariance matrix. When the eigenvalue is greater than 1, it is called attractive patterns $\boldsymbol{\xi}^{+,\mu}$, and when the eigenvalue is less than 1, it is called repulsive patterns $\boldsymbol{\xi}^{-,\nu}$.

P_+ attractive mfHP patterns are:

$$\boldsymbol{\xi}^{+,\mu} = \left(1 - \frac{1}{\lambda_\mu}\right)^{1/2}\tilde{\boldsymbol{v}}^\mu , \quad \mu \in 1{:}P_+ , \tag{C.5}$$

where attractive eingenvalues are $\lambda_1 \geq \lambda_2, \geq, \ldots, \geq, \lambda_{P_+} > 1$.

Similarly, $P(= P_+ + P_-)$ repulsive mfHP patterns are:

$$\boldsymbol{\xi}^{-,\nu} = \left(\frac{1}{\lambda_\nu} - 1\right)^{1/2}\tilde{\boldsymbol{v}}^\nu , \quad \nu \in (P_+ + 1){:}P , \tag{C.6}$$

where repulsive eigenvalues are $0 < \lambda_P \leq \lambda_{P-1}, \ldots, \leq \lambda_{P_++1} < 1$. Note

that these mfHP patterns consider only non-zero eigenmodes therefore $P \leq L(q-1)$.

Here we denote $\tilde{\boldsymbol{v}}^{\mu}$ as the scaled eigenvectors:

$$\begin{aligned} \tilde{\boldsymbol{v}}^{\mu} &= D^{-1/2}\boldsymbol{v}^{\mu} \ , \quad \mu \in 1{:}P \\ D &= \mathrm{diag}(\{\sqrt{f_i(a)}\}) \end{aligned} \tag{C.7}$$

Appendix D

RBM pattern orthogonality and regularization effect

This section shows the overlap or inner products of RBM patterns with the number of hidden variables $P = 16$. The learning protocols are the same as in Sect. 4.3. As the data set, we used the response regulator domain (Pfam ID, PF00076).

Fig. D.1.a shows heat maps of absolute values of inner-products among the $P = 16$ patterns. This model used L2 regularization for the learning. Similarly, Fig. D.1.b shows the same type of plot but using L1 norm. Except for the type of regularization (L1 or L2) both models are identical conditions including the other hyper-parameters.

Fig. D.2 shows absolute values of inner products values as a function of the sorted ranks in descending order. For comparison, we included the overlap value between RBM patterns using the $P = 2$ RBM model. Contributions from the self overlaps are excluded.

In the case of the L2 regularization, the overlap values of all pairs are less than 0.2 . Around 15% of pairs of the patterns show overlap values above 1.0 . The maximum overlap value for L1 regularization is below 0.15, and only a few pairs show overlap values greater than 1.0 .

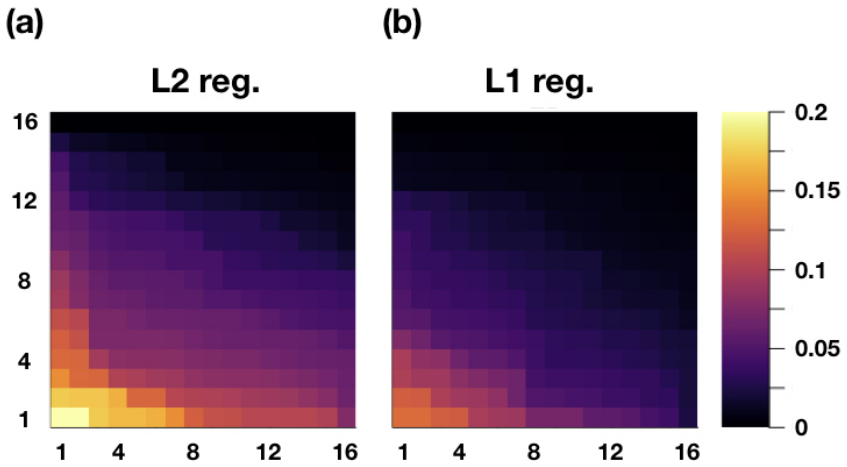

Figure D.1: Absolute value of inner products among RBM patterns, sorted but their value, using L2 (a) or L1 (b) regularization. The self overlaps are excluded and assumed to be zero.

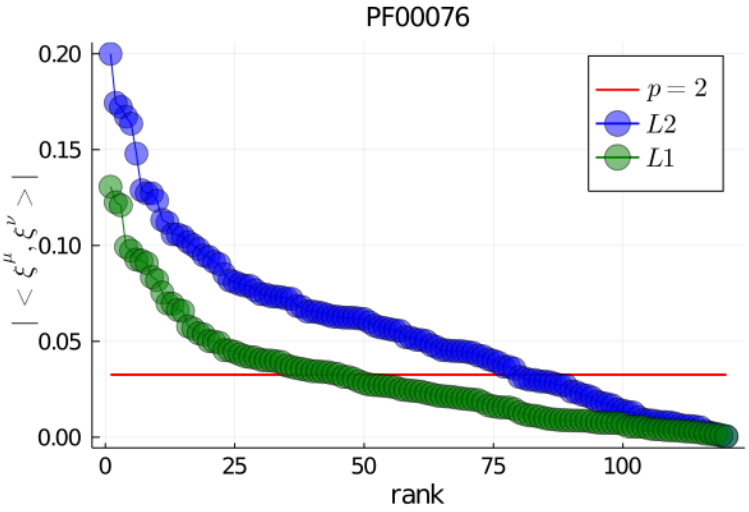

Figure D.2: Same data as in Fig. D.1, but shown as a function of sorted ranks. The blue and green markers correspond to the absolute overlap values for L2 and L1 regularization, respectively. The red line is the overlap values of RBM patterns for the $P = 2$ model.

Appendix E

Contrastive Divergence

E.1 Contrastive Divergence based methods

MCMC is a flexible and relatively fast method to realize probability distribution. However, most of the probability distributions we want to infer require iterative optimization of model parameters and MCMC execution each time. Repeating such a process thousands of times is computationally too demanding, even if running MCMC on a set of model parameters is relatively light.

CD methods are stochastic learning methods for exponential families that can remedy such a computational burden due to the repetitive executions of MCMC. CD-based methods are widely used as common methods for learning exponential families (e.g, Boltzmann machines, Restricted Boltzmann machines). However, there is not yet a sufficient theoretical understanding of the basic properties of learning based on CD methods. The followings are examples:

- Are solutions using CD methods the same as the MLE solutions? (it is generally accepted that CD is a different estimation method from MLE methods [121]).
- Guarantee of convergence [122], speed for convergence, the dependence of estimation bias Etc. have not been understood.

E.1.1 Contrastive Divergence learning

Suppose there is an ensemble of statistics $\boldsymbol{x}^{1:N} = (\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^N) \in \mathcal{X}^N$ that we use as a set of training data, and our primary objective is to describe

the data in terms of statistical modeling. That means to construct a certain probability density $p(\boldsymbol{x}|\boldsymbol{\theta})$, which carries the same statistical properties of the training data (as following we will denote vectors without bold for simplicity such as $\boldsymbol{\theta} \to \theta$) . Here, θ is a set of model parameters that we want to optimize so that the probability distribution can reproduce statistics of the training data. Up to here, the assumed setup is the same as the MLE-based learning.

The significant difference with the maximum likelihood (ML) method is that ML based methods assume equilibrium distribution for estimating gradients of the objective functions, i.e., log-likelihood functions, whereas CD-based methods use non-equilibrium distribution to obtain gradients.

Suppose that the probability distribution of interest belongs to an exponential family, $p(x|\theta) = \exp(-E(x|\theta))/Z$, where Z is the partition function. Accordingly, the log-likelihood function can be written as

$$l(\theta) = \langle E(x|\theta) \rangle_{p_{\text{data}}} - \log Z(\theta) \ , \tag{E.1}$$

where, p_{data} denotes an empirical distribution. Similarly we denote $p_{\text{data}} = p(x|\theta)$ hereafter. Suppose we optimize the model parameters θ using the standard ML approach. Therefore, we perform the differential of the log-likelihood function $l(\theta)$ with respect to the model parameters θ,

$$\frac{\partial l(x^{1:N})}{\partial \theta} = \Big\langle \frac{-\partial E(x|\theta)}{\partial \theta} \Big\rangle_{p_{\text{data}}} - \Big\langle \frac{-\partial E(x|\theta)}{\partial \theta} \Big\rangle_{p_{\text{model}}} \ . \tag{E.2}$$

Where, $\langle \bullet \rangle_{p_{\text{data}}}$ and $\langle \bullet \rangle_{p_{\text{model}}}$ are expected values with the probability distribution of the data p_{data} and the model p_{model}. Eq. E.2 is the fixed point equation of MLE learning to be solved. The corresponding fixed point equation using CD-based learning can be written similarly as follows,

$$\frac{\partial l(x^{1:N})}{\partial \theta} = \Big\langle \frac{-\partial E(x|\theta)}{\partial \theta} \Big\rangle_{p_{\text{data}}} - \Big\langle \frac{-\partial E(x|\theta)}{\partial \theta} \Big\rangle_{q^{(k)}(x|\theta;x^{1:N})} \ . \tag{E.3}$$

Where, $q^{(k)}(x|\theta; x^{1:N})$ is a probability distribution, after k step transitions with an appropriate transition kernel $p(x|x';\theta)$ from an empirical distribution p_{data} . We call Eq. E.3 as the update equation for CD learning. From now on, we call this probability distribution as the distribution after

k step transitions. Formally, we can define $q^{(k)}(x|\theta; x^{1:N})$ as following,

$$q^{(k)}(x|\theta; x^{1:N}) = \left(\prod_{t=1}^{k} \sum_{x^{(t)} \in \mathcal{X}} p(x^{(t)}|x^{(t-1)}; \theta) \right) p_{\text{data}}(x^{(0)}|x^{1:N}) \ , \tag{E.4}$$

where $x^{(k)} = x$, and $x^{(0)} \sim p_{\text{data}}(x|x^{1:N})$. Ref. [123] shows that this fixed point equation can be derived as a consequence of an approximation of the log-likelihood function by expanding it. Historically, the CD method was introduced to minimize a contrastive divergence [124, 125], therefore its objective function is assumed to be different from the log-likelihood function. The contrastive of the divergence is defined as

$$D_{KL}(p_{\text{data}} \| p_{\text{model}}) - D_{KL}(q^{(k)} \| p_{\text{model}}) \ , \tag{E.5}$$

If $1 \ll k$, it leads to $q^{(k)} \sim p_{\text{model}}$, therefore the objective function becomes the same as the one of the MLE.

The differentiation of Eq. E.5 with respect to the parameter θ cannot be exactly the same as Eq. E.2, but assuming the parameter θ dependency on $q^{(k)}$ is not so large i.e., $|\partial q^{(k)}/\partial\theta| \ll 1$. This assumption is usually satisfied unless the set of parameters is at a critical point.

E.1.2 Contrastive Divergence for latent variable models

CD methods are used for Restricted Boltzmann Machines (RBMs) as the standard learning algorithm.

Suppose a probability distribution $p(v, h)$ is characterized by two types of variables, namely visible variables v and hidden or latent variables h. Here we assume there are corresponding observables in an assumed data set for the visible variables, and the probability we can obtain is only a probability after marginalizing the hidden variables, $p(v) = \sum_h p(v, h)$.

The CD learning algorithm for hidden variable models is as follows:

1. Initialize the visible variables using one sample in the training data at the first MC transition step $t = 0$, $v^{(t=0)} \leftarrow x \in x^{1:N}$.

2. Then, the hidden variables are obtained from the conditional probability distribution, $h^{(t=0)} \sim p(h|v^{(t=0)}; \theta)$. Note that the conditional distribution in the case of RBMs becomes independent among each

hidden variable h because each hidden variable has interacted only via the visible variables (there is no interaction between hidden variables).

3. The visible variables are updated by taking them from the conditional distribution, $v^{(t=1)} \sim p(v|h^{(t=0)};\theta)$.

4. Repeat process 1. and 2. $t = k$ times.

If we assume RBM as an example of a latent variable models,

$$\begin{aligned} p(v,h|w,a,b) &\propto e^{-E(v,h|w,a,b))} \\ -E(v,h) &= \sum_{ij} w_{ij} v_i h_j + \sum_i a_i v_i + \sum_j b_j h_j \end{aligned} \tag{E.6}$$

For simplicity, we assume the visible variables are binary variables $v_i \in \{0,1\}$. Where, $\{w_{ij}\}$ are model parameters for interactions between visible and hidden variables. $\{a_i\}$ and $\{b_j\}$ are other model parameters for visible and hidden variables, respectively.

These model parameters are optimized using a gradient-based learning algorithm. Eq. E.5 and Eq. E.6 lead to the following equations for updating the model parameters.

$$\begin{aligned} \delta w_{ij} &= \left\langle v_i h_j \right\rangle_{p(h|v;w,a,b)p_{\text{data}}(v)} - \left\langle v_i h_j \right\rangle_{p(h|v;w,a,b)q^k(v|w,a,b)} \\ \delta a_i &= \left\langle v_i \right\rangle_{p_{\text{data}}(v)} - \left\langle v_i \right\rangle_{q^k(v|w,a,b)} \\ \delta b_j &= \left\langle h_j \right\rangle_{p(h|v;w,a,b)p_{\text{data}}(v)} - \left\langle h_j \right\rangle_{p(h|v;w,a,b)q^k(v|w,a,b)} \end{aligned} \tag{E.7}$$

Learn the RBM so that $\delta w_{ij}, \delta a_i, \delta b_j, \forall i,j$ will be small. A more appropriate learning evaluation method is explained in the following section, E.3.

E.2 Persistent Contrastive Divergence

CD methods can remarkably reduce the computational processes. However, CD samples that are generated from $q^{(k)}(x|\theta;x^{1:N})$ tend to depend strongly on the training data. As a result, the energy function (probability distribution) using the CD learning assigns a significantly lower value (high probability) to state points that are included in the training data. On the contrary, state points that are not included in the training data, tend to

have high energy values (low probabilities).

Besides, the probability distribution constructed using MCMC with the model parameters estimated by the CD method tends not to reproduce the training data correctly. Therefore the data statistics are not accurately reproduced (cf. [97, 126]).

One of the reasons for this phenomenon is that all data points are used as initial states of the probabilistic transitions $p(x^{(t)}|x^{(t-1)};\theta)$ at each learning epoch, and the samples used for estimating the update equations Eq. E.3 are constructed without sufficiently exploring the state space. Thus, the CD learning minimizes the flow of the transition probability that escapes from the data points. This observation is closely related to Minimum Probability Flow learning [56, 127] .

Considering the drawbacks of the CD learning due to the lack of sufficient state space search, T. Tieleman Ref. [128] proposed Persistent Contrastive Divergence (PCD) learning. CD methods and PCD methods are equivalent in terms of computational complexity, but the PCD methods can estimate the update equations with samples that are similar to those generated from the model. Thus, it is closer to the ML learning. When using protein family data, empirically, the probability distributions learned by the PCD methods reproduce the training data well.

The significant difference between the PCD method and the CD method is the initialization of the visible variables: Instead of initializing the visible variables by samples in the training data at the starting point of the transition of each learning epoch, reuse the final states of the previous CD chains.

More precisely, PCD learning consists in the following steps:

1. Initialize the visible variables using one of the training data (same as CD)
2. Apply MCMC k times, then keep samples that are used to estimate the update equations.
3. Use the samples generated and stored during the previous learning epoch as the initial states for the current learning epoch.
4. Repeat processes 2. and 3.

Note that if a learning ratio is sufficiently small (slight changes in model parameters between epochs), the PCD samples would be equivalent to the samples from the equilibrium distribution i.e., the PCD learning gives an equivalent solution to the MLE.

E.3 Convergence criterion for Contrastive Divergence based learning

The update equations are used to evaluate the learning processes in general. However, these quantities cannot be used as appropriate measures to assess the CD-based learning as showed in [129]. Note that the update equation for the CD learning Eq. E.3 is not the "reconstruction error", i.e., the difference between statistics based on the training data and samples from the model. Hence the update equation Eq. E.3 cannot be used to assess the learning processes (e.g., assessing adequate learning epoch). Particularly learning hidden variable models require cautious treatments [130, 129].

To evaluate CD-based learning processes, Annealed Importance Sampling (AIS) [131, 130, 132] can be used. The AIS is defined as a ratio between two partition functions named $Z(\theta^{(t)}), Z(\theta^{(t+1)})$. It can be estimated statistically rigorously:

$$\begin{aligned}\frac{Z(\theta^{(t+1)})}{Z(\theta^{(t)})} &= \frac{\int p^*(x|\theta^{(t+1)})dx}{Z(\theta^{(t)})} = \left\langle \frac{p^*(x|\theta^{(t+1)})}{p^*(x|\theta^{(t)})} \right\rangle_{p(x|\theta^{(t)})} \\ &\sim \frac{1}{M}\sum_{m=1}^{M} \frac{p^*(x^m|\theta^{(t+1)})}{p^*(x^m|\theta^{(t)})} \ ,\end{aligned} \tag{E.8}$$

where $p^*(x|\theta)$ denotes the unnormalized probability distribution of the model, and $\langle \bullet \rangle_{p(x|\theta^{(t)})}$ is defined as an expected value of the probability distribution $p(x|\theta) = e^{-E(x|\theta)}/Z(\theta)$. In the last equation we assume that the expected value $\langle \bullet \rangle_{p(x|\theta^{(t)})}$ can be approximated as an ensemble average and that those samples come from the probability $x^m \sim p(x|\theta^{(t)})$.

If $p(x|\theta^{(t+1)})$ and $p(x|\theta^{(t)})$ are quite different, the ratio cannot be estimated accurately. In such a situation, we can introduce intermediate probability distributions between these two probability distributions and estimate

the ratio as follows:

$$
\begin{aligned}
p_\beta(x|\theta^{(t)}, \theta^{(t+1)}) &= p^\beta(x|\theta^{(t+1)}) p^{1-\beta}(x|\theta^{(t)}) \\
\frac{Z(\theta^{(t+1)})}{Z(\theta^{(t)})} &= \prod_{k=0}^{K-1} \frac{Z_{\beta_{k+1}}(\theta^{(t+1)}, \theta^{(t)})}{Z_{\beta_k}(\theta^{(t+1)}, \theta^{(t)})} \ ,
\end{aligned}
\tag{E.9}
$$

where $\beta_k = k/K, k = 1, 2, .., K$, and $Z_\beta(\theta^{(t+1)}, \theta^{(t)})$ is a normalization factor of the probability distribution $p_\beta(x|\theta^{(t)}, \theta^{(t+1)})$.

It is reported that AIS values are good measurements to evaluate learning processes and to determine the optimal learning stopping epochs [130] and it also works for protein sequence modeling [115].

Appendix F

Other remarks of likelihood-variation method

F.1 Another derivation element-wise likelihood variation

Here, we show another derivation of the optimal likelihood variation $\Delta l(J^*_{ij}(a,b))$ in Eq. 5.9.

Suppose that $M \times f_{ij}(a,b)$ out of M sequences in a given MSA are drawn from a probability distribution, $p_{ij}(a,b) = \sum_{\boldsymbol{A}\in\mathcal{A}} p(A)\delta_{A_i,a}\delta_{A_j,b}$, where $p(\boldsymbol{A}) \propto e^{-H(\boldsymbol{A})}$.

Therefore, the binomial distribution to consider can be written as:

$$p_M[f,p] := \binom{M}{Mf} p^{Mf}(1-p)^{M(1-f)} \ . \tag{F.1}$$

For simplicity, we ignored site and amino-acid indexes.

Eq. F.1 can be simplified to an exponential form with a large M limit,

$$\begin{aligned} p_M[f,p] &\propto \exp\Big(- ML[f,p]\Big) \\ L[f,p] &= f\log\frac{f}{p} + (1-f)\log\frac{1-f}{1-p} \ . \end{aligned} \tag{F.2}$$

Here, we used Stirling formula $N! \sim e^{N\log N - N}$ with large N. Eq. F.2

is exactly the same form as we saw in Eq. 5.9.

If there is a significant discrepancy between the empirical distribution $f_{ij}(a,b)$ and the assumed model $p_{ij}(a,b)$, in this case $p_M[f_{ij}(a,b), p_{ij}(a,b)]$ becomes particularly small. That is exactly the case in which we need to make corrections by introducing the conjugate parameters $J_{ij}(a,b)$.

The same argument can hold not only for two-point frequencies, but also for single-site frequencies, and even three- or more-point frequencies.

F.2 Residue contact for additional protein families

As we showed in Sec. 5.3.1, the likelihood-variation-based residue-residue contacts prediction can improve contact as the number of hidden variables P increases in the RBMs. In this section, we will report residue contact prediction results using the same methods for other protein families, PF00072 and PF13354. We used identical data sets that are used in Ref. [92].

Fig. F.1 shows PPV curves for PF13354. Similarly, Fig. F.2 shows PPV curves for PF00076. Likelihood-variation-based methods $F^{\text{E.L.}}$ and $F^{\text{B.L.}}$ show accurate results for both families. Especially in Fig. F.1.c and Fig. F.1.d, the PPV values are higher than the result of plmDCA above around 200 contact predictions.

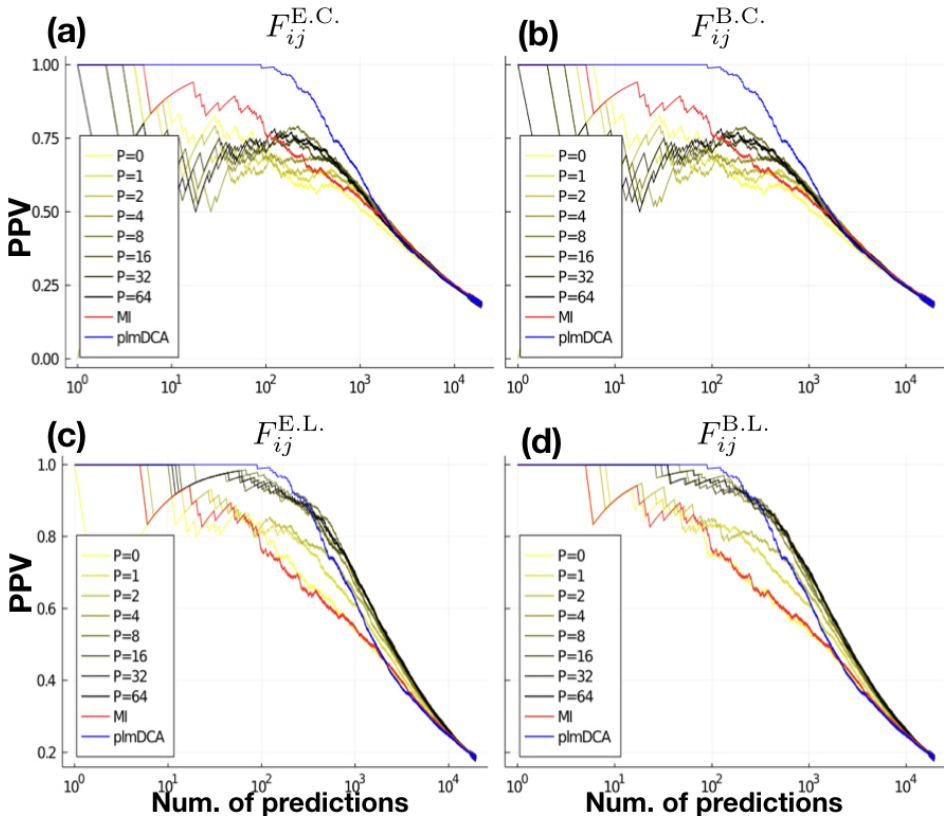

Figure F.1: (a) PPV curves based on the element-wise coupling activation $F^{\mathrm{E.C.}}$. (b) The same types of plots as (a) but for the block-wise coupling activation $F^{\mathrm{B.C.}}$. (c) PPV curves based on the element-wise likelihood variation $F^{\mathrm{E.L.}}$. Fig. F.1.c and Fig. F.1.d show a clear dependency on the number of hidden variables in RBM P. Increasing P improves these PPV curves, with almost the same accuracy as contact predictions based on plmDCA. Especially, when $P \geq 8$, above a certain number of predictions, they achieve better accuracy than plmDCA accuracy. (d) The same types of plots as (d) but for the block-wise likelihood variation $F^{\mathrm{B.L.}}$.

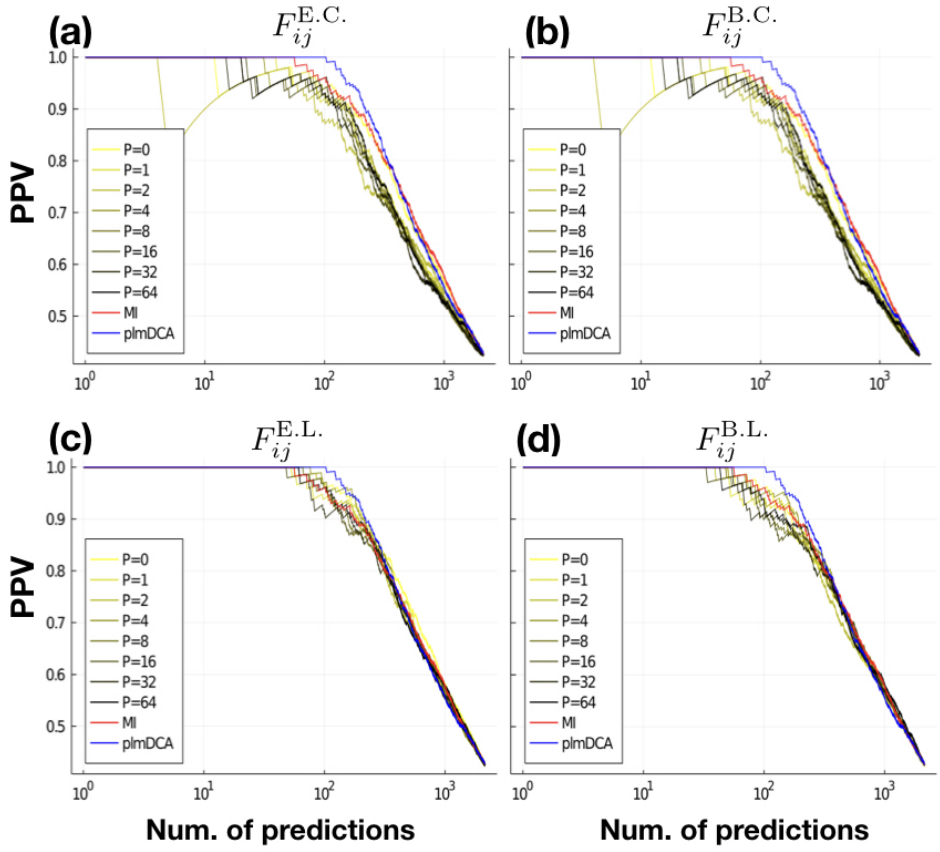

Figure F.2: PPV curves for $F^{\mathrm{E.C.}}$ (a), $F^{\mathrm{B.C.}}$ (b), $F^{\mathrm{E.L.}}$ (c), $F^{\mathrm{B.L.}}$ (d). Since the MI result (corresponding to $P = 0$ in the case of $F^{\mathrm{B.L.}}$) itself has already been predicted as accurately as plmDCA, the effect of the phylogenetic effect on this protein family is probably not so substantial.

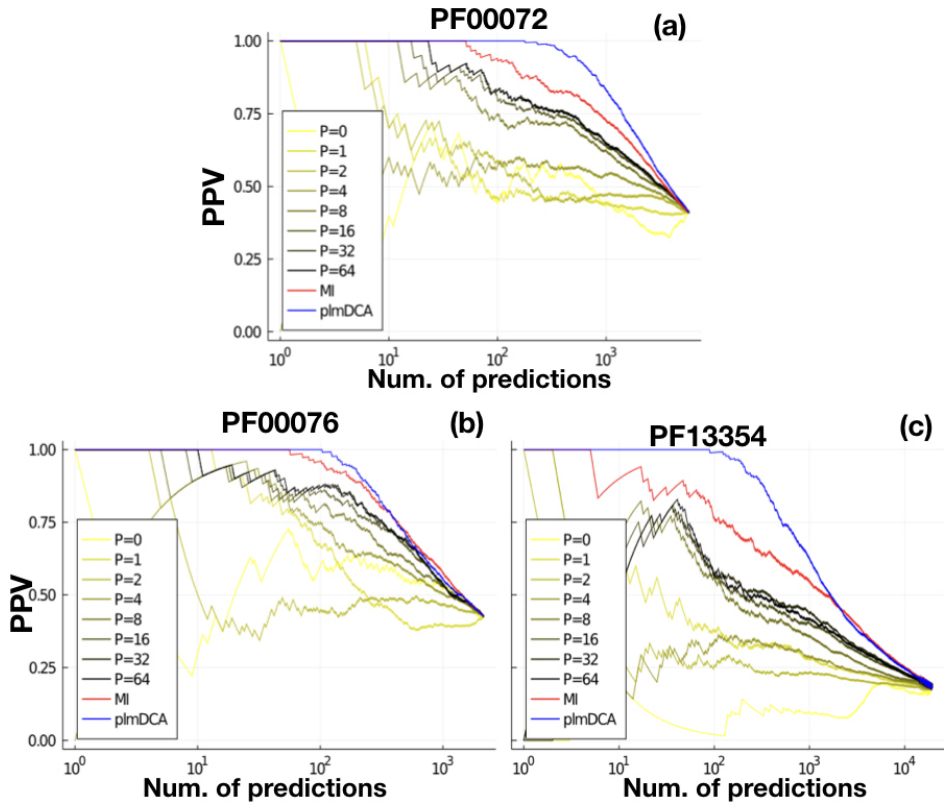

Figure F.3: (a) PPV curves based on the coupling matrix constructed by mfHP patterns (the same definition of couplings shown in Eq. 4.5). The accuracy of contact prediction increase as increases P, which is expected result and is coherent with Ref. [97]. The same types of plots as (a) are shown for PF00076 (b) and for PF13354 (c). For all of these families, the accuracy of contact predictions improves as increase P.

Appendix G

Variational Autoencoder

G.1 Other technical points of VAE

G.1.1 Reparametrization

There are numerous works of literature on optimization methods for deep neural networks such as Momentum, Adam, Adadelta, Adagrad, etc. Most methods are based on the backpropagation (BP, backprop) algorithm, which is an essential gradient-based optimization, particularly for neural networks.

The encoder and decoders are represented as the neural network architectures, therefore, the objective function, ELBO is differentiable. However, problems arise when parameters are stochastics: the dependency of ϕ on the expectation $\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\cdot]$ comes via the hidden variable, which has a stochastic quantity, so it is not differentiable.

An important idea to solve this problem is to make all variables including hidden variables differentiable by separating the deterministic parts and stochastic parts [116]. Note that variables $\boldsymbol{z}$ are assumed as functions of random variables, and dependency of $\boldsymbol{x}$ and ϕ are absorbed in the function. Therefore, hidden variables can be written as

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{z}(\boldsymbol{\xi}, \boldsymbol{x}; \phi) = \boldsymbol{\mu}_\phi(\boldsymbol{x}) + \boldsymbol{\Sigma}_\phi(\boldsymbol{x})\boldsymbol{\xi} \\ \boldsymbol{\xi} &\sim \mathcal{N}(\boldsymbol{\xi}; \mathbf{0}, \mathrm{I}) \ . \end{aligned} \tag{G.1}$$

This method to separate stochastic factors and deterministic factors is called *reparameterization trick* [107]. Accordingly, the objective function

ELBO should also be slightly modified (we don't rewrite the objective function but it is just plug Eq. G.1).

It is easy to check that the derivative with respect to parameters of the expectation of this stochastic objective function is equivalent to the derivative of the original ELBO objective function, $\nabla_{\theta,\phi}\mathbb{E}_{\mathcal{N}(\boldsymbol{\xi};\mathbf{0},\mathrm{I})}[\tilde{\mathcal{L}}_{\theta,\phi}(\boldsymbol{x},\boldsymbol{\xi})] = \nabla_{\theta,\phi}\mathcal{L}(\boldsymbol{x})$.

Regarding the reparametrization, we also need to take into account Jacobian,

$$q_\phi(\boldsymbol{z}|\boldsymbol{x})\left|\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\xi}}\right| = \mathcal{N}(\boldsymbol{\xi};\boldsymbol{\mu},\boldsymbol{\Sigma}) \ , \tag{G.2}$$

where $\left|\mathrm{A}\right|$ is determinant of a matrix A, and $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\xi}}$ is the Jacobian,

$$\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\xi}} = \frac{\partial(z_1, z_2, \ldots, z_P)}{\partial(\xi_1, \xi_2, \ldots, \xi_P)} = \left(\frac{\partial z_i}{\partial \xi_j}\right)_{i,j\in 1:P} . \tag{G.3}$$

In the case of the above example, we supposed that the hidden variable comes from Gaussian distribution,

$$\frac{\partial z_i}{\partial \xi_j} = \frac{\partial}{\partial \xi_j}(\boldsymbol{\mu} + \Sigma\boldsymbol{\xi})_i = \Sigma_{ij} \ . \tag{G.4}$$

The conditional probability distribution of hidden valiables given visible variables can be represented as following,

$$\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \log \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}(\boldsymbol{x}))) = \sum_u^P \Big(\log \mathcal{N}(z_u; \mu_u(\boldsymbol{x}), \sigma_u^2(\boldsymbol{x})) - \log \sigma_u(\boldsymbol{x}) \Big) .$$

Put everything together, the objective function we used is represented as follows,

$$\begin{aligned} &\tilde{L}_{\theta,\phi}(\boldsymbol{x},\boldsymbol{\xi}) \\ &= \log p_\theta(\boldsymbol{x}|\boldsymbol{z}))p_\theta(\boldsymbol{z}) - \log q_\phi(\boldsymbol{z}|\boldsymbol{x}) \\ &= \log p_\theta(\boldsymbol{x}|\boldsymbol{z})) + \sum_u^P \Big(\log \mathcal{N}(z_i; 0, 1) - \log \mathcal{N}(z_u; \mu_u(\boldsymbol{x}), \sigma_u^2(\boldsymbol{x})) - \log \sigma_u(\boldsymbol{x}) \Big) . \end{aligned} \tag{G.5}$$

Here, we assumed $\boldsymbol{z} = \boldsymbol{z}(\boldsymbol{\xi}, \boldsymbol{x}; \phi)$. Note that we have not explicitly stated the index of the data points. To be the objective function, need to sum Eq.

G.5 for all data points [1].

G.1.2 Other types of VAEs

Here we show some variants of VAE that are closely related to our study.

β-VAE – β-VAE is a simple powerful effective method that aims to resolve the frequently occurred problem, *posterior collapse* [133], where the decoder learns to ignore the hidden variables.

The β-VAE can be realized by introducing a hyper-parameter β to ELBO objective function [2].

$$\begin{aligned}\mathcal{L}^{\beta\text{VAE}}(\theta,\phi) = (1-\beta)\Big\langle \log p_\theta(\boldsymbol{x}|\boldsymbol{z})\Big\rangle_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \\ - \beta KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\mathbf{z}))\ , \quad \beta\in(0,1)\ ,\end{aligned} \tag{G.6}$$

if $\beta = 0.5$, β-VAE recovers original VAE. The small β imposes less constraints to the latent space. On the other hand, large β induces strong constraints so that the hidden variables are independent when it is assumed standard Gaussian $p_\theta(\boldsymbol{z}) \sim \prod_\mu p_\theta(z_\mu)$.

In order to avoid the posterior collapse, we can push VAE to learn KLD terms more by increasing β. That is, the posterior of the encoder becomes similar with the distribution of hidden prior [135]. Moreover, by controlling the β, we can enforce features or hidden variables [134].

Conditional VAE – The standard VAE learning is unsupervised, which means there is no additional label to train VAEs. The conditional VAE (CVAE) [136, 137] is a semi-supervised VAE that can generate a certain class of variables according to a label variable. For example, it can specifically generate images of "7" when the model learns images of handwritten numbers (0-9) with the labels (also 0-9) (cf. MNIST data set).

Formally, the objective function of CVAE is defined as the variational lower bound of the conditional probability distribution of data points given

[1] Note also that need to marginalize the hidden variables to be the objective function in practice.

[2] The original β-VAE in [134] has β dependency only on the second term in eq.(G.6) and the $\beta > 0$, but essentially the same.

class variables. Thereby, the hidden prior distribution also depends on the class variables.

$$\mathcal{L}^{\mathrm{CVAE}} = \left\langle \log p_\theta(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c}) \right\rangle_{q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})} - KL(q_\phi(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})\|p_\theta(\boldsymbol{z},\boldsymbol{c})) \ , \tag{G.7}$$

where, $\boldsymbol{c}$ is the conditional variable. The class variables are assumed as additional input variables, so they are assigned to additional perceptrons. Therefore, CVAE can be learned by standard ELBO learning.

G.2 Conditions of the experiments

Batch learning – We also used batch-learning and batch-normalization methods. Batch-learning is a method to optimize model parameters using a sub-ensemble of training data for each learning epoch. The gradients of the objective function are obtained using sub-ensembles that contain B sequences.

Our study constructed a batch ensemble by resampling using the reweighting parameters (cf. Sec. 2.4.3) from all sequences in the training data for each learning epoch. We didn't find any significant difference by changing the batch size B between 300 and 1000.

Advantages of batch learning are:

1. Reducing requiring computational processes.
2. Various combinations of samples can be provided to DNN for each learning epoch.
3. Reducing the sample size enhances the stochastic effect and helps avoid local minima. Note that DNN-based problems usually involve many local minima solutions.

The second reason can be useful for VAE learning, but particularly for Generative Adversarial Network (GAN) algorithm [138, 139, 140], to overcome so-called *mode collapse* [141, 142, 143, 144].

Batch normalization is a method for removing gradient bias, which is equivalent to using normalized training data as input. By using this method, learning is expected to be stable and fast [145].

Optimization of DNNs – For optimization methods, we employed a stochastic gradient-based optimization method *Adam* [116] (adaptive moment estimation), which is a commonly used optimization method for DNN-based algorithms. To apply Adam, only the first and second moments of gradients are needed (these are obtained as a by-product while optimizing the DNN) and are computationally efficient.

We have selected the weight decay parameters for the first and second moments, which are Adam's hyper-parameters, to 0.99 and 0.9999, respectively.

One-hot sequence – To learn protein sequence data using a neural-network-based algorithm, we map categorical variables representing amino acids to binary variables, $\{1{:}q\}^L \to \{1,0\}^{qL}$. We refer to this binarized sequences as one-hot sequences.

In general, initial conditions of input neurons are symmetric; all input neurons connect to the next layer's all neurons. Asymmetricity of the neural networks due to the one-hot data structure emerges as a consequence of learning.

We often represent a one-dimensional vector containing qL elements as a $q \times L$ matrix. Here, it corresponds to each residue site in the column, and in the case of one-hot sequence, only one of the q elements is 1 and the others are 0. Fig. G.1.a shows a typical PF00072 protein sequence using one-hot representation (cf. Sec. 6.2). These white and black elements in the matrix correspond to one and zero, respectively.

Output sequences from the VAEs are continuous variables $\mathbb{R}^{qL}$ because the activation functions of the output layer are the Sigmoid function. Therefore, it is necessary to convert it into a one-hot sequence by some filter. One naive way is to take the argument with the largest output value of all the q-state amino acids.

Fig. G.1.b shows an output sequence from VAE (cf. Sec. 6.2.3) using the filter. Fig. G.1.c shows the same output sequence without the filter.

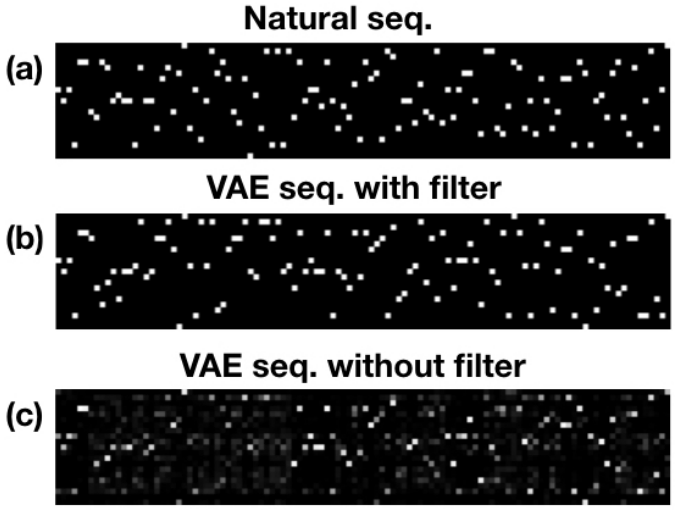

Figure G.1: (a) One-hot encoded matrix ($q \times L$, q=21, L=112) of a protein sequence (PF00072). White elements and black elements are assigned as 1 and 0 respectively. (b) One-hot encoded matrix generated from the VAE using the argument-max filter. (c) One-hot encoded matrix generated from the VAE.

Bibliography

[1] Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[2] Cyrus Levinthal. How to fold graciously. *Mossbauer spectroscopy in biological systems*, 67:22–24, 1969.

[3] Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[4] CA Floudas, HK Fung, SR McAllister, M Mönnigmann, and R Rajgaria. Advances in protein structure prediction and de novo protein design: A review. *Chemical Engineering Science*, 61(3):966–988, 2006.

[5] Brian Kuhlman and Philip Bradley. Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, 20(11):681–697, 2019.

[6] Tapan K Chaudhuri and Subhankar Paul. Protein-misfolding diseases and chaperone-based therapeutic approaches. *The FEBS journal*, 273(7):1331–1349, 2006.

[7] Frederic Martini et al. *Anatomy and Physiology'2007 Ed.* Rex Bookstore, Inc., 2006.

[8] Bruce Alberts, Dennis Bray, Karen Hopkin, Alexander D Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Essential cell biology*. Garland Science, 2015.

[9] Lamy Jean-Baptiste, Hélène Berthelot, and Madeleine Favre. Rainbow boxes: a technique for visualizing overlapping sets and an application to the comparison of drugs properties. In *2016 20th International Conference Information Visualisation (IV)*, pages 253–260. IEEE, 2016.

[10] Anders Liljas, Lars Liljas, Goran Lindblom, Poul Nissen, Morten Kjeldgaard, and Miriam-rose Ash. *Textbook of structural biology*, volume 8. World Scientific, 2016.

[11] Sameer Velankar, Younes Alhroub, Anaëlle Alili, Christoph Best, Harry C Boutselakis, Ségolène Caboche, Matthew J Conroy, Jose M Dana, Glen van Ginkel, Adel Golovin, et al. Pdbe: protein data bank in europe. *Nucleic acids research*, 39(suppl_1):D402–D410, 2010.

[12] MS Smyth and JHJ Martin. x ray crystallography. *Molecular Pathology*, 53(1):8, 2000.

[13] Elisabeth P Carpenter, Konstantinos Beis, Alexander D Cameron, and So Iwata. Overcoming the challenges of membrane protein crystallography. *Current opinion in structural biology*, 18(5):581–586, 2008.

[14] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[15] Julie D Thompson, Desmond G Higgins, and Toby J Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.

[16] Sean Eddy. Hmmer user's guide. *Department of Genetics, Washington University School of Medicine*, 2(1):13, 1992.

[17] Robert D Finn, Jody Clements, and Sean R Eddy. Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, 39(suppl_2):W29–W37, 2011.

[18] Johannes Söding, Andreas Biegert, and Andrei N Lupas. The hhpred interactive server for protein homology detection and structure prediction. *Nucleic acids research*, 33(suppl_2):W244–W248, 2005.

[19] Johannes Söding. Protein homology detection by hmm–hmm comparison. *Bioinformatics*, 21(7):951–960, 2005.

[20] UniProt Consortium. Uniprot: a hub for protein information. *Nucleic acids research*, 43(D1):D204–D212, 2015.

[21] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine

Michoud, Claire O'Donovan, Isabelle Phan, et al. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research*, 31(1):365–370, 2003.

[22] Martin Tompa, Nan Li, Timothy L Bailey, George M Church, Bart De Moor, Eleazar Eskin, Alexander V Favorov, Martin C Frith, Yutao Fu, W James Kent, et al. Assessing computational tools for the discovery of transcription factor binding sites. *Nature biotechnology*, 23(1):137–144, 2005.

[23] Anthony Mathelier and Wyeth W Wasserman. The next generation of transcription factor binding site prediction. *PLoS Comput Biol*, 9(9):e1003214, 2013.

[24] Yue Jiang, Bojan Cukic, Donald A Adjeroh, Heath D Skinner, Jie Lin, Qingxi J Shen, and Bing-Hua Jiang. An algorithm for identifying novel targets of transcription factor families: application to hypoxia-inducible factor 1 targets. *Cancer informatics*, 7:CIN–S1054, 2009.

[25] Xiaoman Li and Wing H Wong. Sampling motifs on phylogenetic trees. *Proceedings of the National Academy of Sciences*, 102(27):9481–9486, 2005.

[26] Rahul Siddharthan, Eric D Siggia, and Erik Van Nimwegen. Phylogibbs: a gibbs sampling motif finder that incorporates phylogeny. *PLoS Comput Biol*, 1(7):e67, 2005.

[27] SL Salzberg, DB Searls, and S Kasif. models for biological sequences. *Computational methods in molecular biology*, page 45, 1998.

[28] Byung-Jun Yoon. Hidden markov models and their applications in biological sequence analysis. *Current genomics*, 10(6):402–415, 2009.

[29] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

[30] David KY Chiu and Ted Kolodziejczak. Inferring consensus structure from nucleic acid sequences. *Bioinformatics*, 7(3):347–352, 1991.

[31] Ulrike Göbel, Chris Sander, Reinhard Schneider, and Alfonso Valencia. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics*, 18(4):309–317, 1994.

[32] Neil D Clarke. Covariation of residues in the homeodomain sequence family. *Protein Science*, 4(11):2269–2278, 1995.

[33] Stanley D Dunn, Lindi M Wahl, and Gregory B Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, 2008.

[34] Edwin T Jaynes. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70(9):939–952, 1982.

[35] Ryo Tamura and Koji Hukushima. Bayesian optimization for computationally extensive probability distributions. *Plos one*, 13(3):e0193785, 2018.

[36] Ryo Tamura, Koji Hukushima, Akira Matsuo, Koichi Kindo, and Masashi Hase. Data-driven determination of the spin hamiltonian parameters and their uncertainties: The case of the zigzag-chain compound kcu 4 p 3 o 12. *Physical Review B*, 101(22):224435, 2020.

[37] Martin Weigt, Robert A White, Hendrik Szurmant, James A Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009.

[38] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[39] Ludovico Sutto, Simone Marsili, Alfonso Valencia, and Francesco Luigi Gervasio. From residue coevolution to protein conformational ensembles and functional dynamics. *Proceedings of the National Academy of Sciences*, 112(44):13567–13572, 2015.

[40] Alan S Lapedes, Bertrand G Giraud, LonChang Liu, and Gary D Stormo. Correlated mutations in models of protein sequences: phylogenetic and structural effects. *Lecture Notes-Monograph Series*, pages 236–256, 1999.

[41] Lukas Burger and Erik Van Nimwegen. Disentangling direct from indirect co-evolution of residues in protein alignments. *PLoS computational biology*, 6(1):e1000633, 2010.

[42] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[43] Matteo Figliuzzi, Pierre Barrat-Charlaix, and Martin Weigt. How pairwise coevolutionary models capture the collective residue variability in proteins? *Molecular Biology and Evolution*, 35(4):1018–1027, 2018.

[44] Chongli Qin and Lucy J Colwell. Power law tails in phylogenetic systems. *Proceedings of the National Academy of Sciences*, 115(4):690–695, 2018.

[45] Magnus Ekeberg, Tuomo Hartonen, and Erik Aurell. Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *Journal of Computational Physics*, 276:341–356, 2014.

[46] Pradeep Ravikumar, Martin J Wainwright, John D Lafferty, et al. High-dimensional ising model selection using 1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

[47] Sivaraman Balakrishnan, Hetunandan Kamisetty, Jaime G Carbonell, Su-In Lee, and Christopher James Langmead. Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078, 2011.

[48] Basilis Gidas. Consistency of maximum likelihood and pseudo-likelihood estimators for gibbs distributions. In *Stochastic differential systems, stochastic control theory and applications*, pages 129–145. Springer, 1988.

[49] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E*, 87(1):012707, 2013.

[50] Jeanne Trinquier, Guido Uguzzoni, Andrea Pagnani, Francesco Zamponi, and Martin Weigt. Efficient generative modeling of pro-

tein sequences using simple autoregressive models. *arXiv preprint arXiv:2103.03292*, 2021.

[51] Simona Cocco, Remi Monasson, and Martin Weigt. From principal component to direct coupling analysis of coevolution in proteins: Low-eigenvalue modes are needed for structure prediction. *PLoS Comput Biol*, 9(8):e1003176, 2013.

[52] Francesca Rizzato, Alice Coucke, Eleonora de Leonardis, John P Barton, Jérôme Tubiana, Remi Monasson, and Simona Cocco. Inference of compressed potts graphical models. *Physical Review E*, 101(1):012309, 2020.

[53] David T Jones, Daniel WA Buchan, Domenico Cozzetto, and Massimiliano Pontil. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

[54] Susann Vorberg, Stefan Seemayer, and Johannes Söding. Synthetic protein alignments by ccmgen quantify noise in residue-residue contact prediction. *PLoS computational biology*, 14(11):e1006526, 2018.

[55] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[56] Jascha Sohl-Dickstein, Peter Battaglino, and Michael R DeWeese. Minimum probability flow learning. *arXiv preprint arXiv:0906.4779*, 2009.

[57] Ahmed A Quadeer, Matthew R McKay, John P Barton, and Raymond HY Louie. Mpf–bml: a standalone gui-based package for maximum entropy model inference. *Bioinformatics*, 36(7):2278–2279, 2020.

[58] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d structure computed from evolutionary sequence variation. *PloS one*, 6(12):e28766, 2011.

[59] Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. *Proceedings of the National Academy of Sciences*, 110(39):15674–15679, 2013.

[60] Caleb Weinreb, Adam J Riesselman, John B Ingraham, Torsten Gross, Chris Sander, and Debora S Marks. 3d rna and functional interactions from evolutionary couplings. *Cell*, 165(4):963–975, 2016.

[61] Marco Vassura, Pietro Di Lena, Luciano Margara, Maria Mirto, Giovanni Aloisio, Piero Fariselli, and Rita Casadio. Blurring contact maps of thousands of proteins: what we can learn by reconstructing 3d structure. *BioData mining*, 4(1):1, 2011.

[62] Sebastian Bittrich, Michael Schroeder, and Dirk Labudde. Structure-distiller: Structural relevance scoring identifies the most informative entries of a contact map. *Scientific reports*, 9(1):1–15, 2019.

[63] Michael Socolich, Steve W Lockless, William P Russ, Heather Lee, Kevin H Gardner, and Rama Ranganathan. Evolutionary information for specifying a protein fold. *Nature*, 437(7058):512–518, 2005.

[64] John P Barton, Eleonora De Leonardis, Alice Coucke, and Simona Cocco. Ace: adaptive cluster expansion for maximum entropy graphical model inference. *Bioinformatics*, 32(20):3089–3097, 2016.

[65] William P Russ, Drew M Lowery, Prashant Mishra, Michael B Yaffe, and Rama Ranganathan. Natural-like function in artificial ww domains. *Nature*, 437(7058):579–583, 2005.

[66] Simona Cocco, Christoph Feinauer, Matteo Figliuzzi, Rémi Monasson, and Martin Weigt. Inverse statistical physics of protein sequences: a key issues review. *Reports on Progress in Physics*, 81(3):032601, 2018.

[67] William P Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, et al. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, 2020.

[68] Douglas M Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nature methods*, 11(8):801–807, 2014.

[69] Helen C Davison, Mark EJ Woolhouse, and J Chris Low. What is antibiotic resistance and how can we measure it? *Trends in microbiology*, 8(12):554–559, 2000.

[70] Matteo Figliuzzi, Hervé Jacquier, Alexander Schug, Oliver Tenaillon, and Martin Weigt. Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase tem-1. *Molecular biology and evolution*, 33(1):268–280, 2016.

[71] Pengfei Tian, John M Louis, James L Baber, Annie Aniana, and Robert B Best. Co-evolutionary fitness landscapes for sequence design. *Angewandte Chemie International Edition*, 57(20):5674–5678, 2018.

[72] Thomas C Butler, John P Barton, Mehran Kardar, and Arup K Chakraborty. Identification of drug resistance mutations in hiv from constraints on natural evolution. *Physical Review E*, 93(2):022412, 2016.

[73] Tian-hao Zhang, Lei Dai, John P Barton, Yushen Du, Yuxiang Tan, Wenwen Pang, Arup K Chakraborty, James O Lloyd-Smith, and Ren Sun. Predominance of positive epistasis among drug resistance-associated mutations in hiv-1 protease. *PLoS genetics*, 16(10):e1009009, 2020.

[74] John P Barton, Nilu Goonetilleke, Thomas C Butler, Bruce D Walker, Andrew J McMichael, and Arup K Chakraborty. Relative rate and location of intra-host hiv evolution to evade cellular immunity are predictable. *Nature communications*, 7(1):1–10, 2016.

[75] Barbara Bravi, Riccardo Ravasio, Carolina Brito, and Matthieu Wyart. Direct coupling analysis of epistasis in allosteric materials. *PLoS computational biology*, 16(3):e1007630, 2020.

[76] Jaclyn K Mann, John P Barton, Andrew L Ferguson, Saleha Omarjee, Bruce D Walker, Arup Chakraborty, and Thumbi Ndung'u. The fitness landscape of hiv-1 gag: advanced modeling approaches and validation of model predictions by in vitro testing. *PLoS Comput Biol*, 10(8):e1003776, 2014.

[77] Kevin J Kaczorowski. *Data-driven strategies for vaccine design*. PhD thesis, Massachusetts Institute of Technology, 2017.

[78] Tuba Sevimoglu and Kazim Yalcin Arga. The role of protein interaction networks in systems biomedicine. *Computational and structural biotechnology journal*, 11(18):22–27, 2014.

[79] Thanh-Phuong Nguyen and Tu-Bao Ho. Detecting disease genes based on semi-supervised learning and protein–protein interaction networks. *Artificial intelligence in medicine*, 54(1):63–71, 2012.

[80] Kwang-Il Goh, Michael E Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

[81] Christof Winter, Glen Kristiansen, Stephan Kersting, Janine Roy, Daniela Aust, Thomas Knösel, Petra Rümmele, Beatrix Jahnke, Vera Hentrich, Felix Rückert, et al. Google goes cancer: improving outcome prediction for cancer patients by network-based ranking of marker genes. *PLoS Comput Biol*, 8(5):e1002511, 2012.

[82] Dániel Bánky, Gábor Iván, and Vince Grolmusz. Equal opportunity for low-degree network nodes: a pagerank-based method for protein target identification in metabolic graphs. *PLoS One*, 8(1):e54204, 2013.

[83] Muhammed AY, Kwang-Il Goh, Michael E Cusick, Albert-Laszlo Barabasi, Marc Vidal, et al. Drug–target network. *Nature biotechnology*, 25(10):1119–1127, 2007.

[84] Christoph Feinauer, Hendrik Szurmant, Martin Weigt, and Andrea Pagnani. Inter-protein sequence co-evolution predicts known physical interactions in bacterial ribosomes and the trp operon. *PloS one*, 11(2):e0149166, 2016.

[85] Najeeb Halabi, Olivier Rivoire, Stanislas Leibler, and Rama Ranganathan. Protein sectors: evolutionary units of three-dimensional structure. *Cell*, 138(4):774–786, 2009.

[86] Rama Ranganathan and Olivier Rivoire. Note 109: A summary of sca calculations, 2012.

[87] Olivier Rivoire, Kimberly A Reynolds, and Rama Ranganathan. Evolution-based functional decomposition of proteins. *PLoS computational biology*, 12(6):e1004817, 2016.

[88] Silvia Grigolon, Silvio Franz, and Matteo Marsili. Identifying relevant positions in proteins by critical variable selection. *Molecular BioSystems*, 12(7):2147–2158, 2016.

[89] Alice Coucke, Guido Uguzzoni, Francesco Oteri, Simona Cocco, Remi Monasson, and Martin Weigt. Direct coevolutionary couplings reflect biophysical residue interactions in proteins. *The Journal of chemical physics*, 145(17):174102, 2016.

[90] Pierre Barrat-Charlaix. *Understanding and improving statistical models of protein sequences*. PhD thesis, Sorbonne Université, 2018.

[91] Aurélien Decelle and Federico Ricci-Tersenghi. Pseudolikelihood decimation algorithm improving the inference of the interaction network in a general class of ising models. *Physical review letters*, 112(7):070603, 2014.

[92] Pierre Barrat-Charlaix, Anna Paola Muntoni, Kai Shimagaki, Martin Weigt, and Francesco Zamponi. Sparse generative modeling of protein-sequence families. *arXiv preprint arXiv:2011.11259*, 2020.

[93] Yingying Xu, Erik Aurell, Jukka Corander, and Yoshiyuki Kabashima. Statistical properties of interaction parameter estimates in direct coupling analysis. *arXiv preprint arXiv:1704.01459*, 2017.

[94] Yingying Xu, Santeri Puranen, Jukka Corander, and Yoshiyuki Kabashima. Inverse finite-size scaling for high-dimensional significance analysis. *Physical Review E*, 97(6):062112, 2018.

[95] Simona Cocco, Rémi Monasson, and Martin Weigt. Inference of hopfield-potts patterns from covariation in protein families: calculation and statistical error bars. In *Journal of Physics: Conference Series*, volume 473, page 012010. IOP Publishing, 2013.

[96] Adriano Barra, Alberto Bernacchia, Enrica Santucci, and Pierluigi Contucci. On the equivalence of hopfield networks and boltzmann machines. *Neural Networks*, 34:1–9, 2012.

[97] Kai Shimagaki and Martin Weigt. Selection of sequence motifs and generative hopfield-potts models for protein families. *Physical Review E*, 100(3):032128, 2019.

[98] Simona Cocco, Remi Monasson, and Vitor Sessak. High-dimensional inference with the generalized hopfield model: Principal component analysis and corrections. *Physical Review E*, 83(5):051123, 2011.

[99] Silvio Franz, Federico Ricci-Tersenghi, and Jacopo Rocchi. A fast and accurate algorithm for inferring sparse ising models via parameters activation to maximize the pseudo-likelihood. *arXiv preprint arXiv:1901.11325*, 2019.

[100] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[101] Yuko Tsuchiya and Kentaro Tomii. Neural networks for protein structure and function prediction and dynamic analysis. *Biophysical reviews*, pages 1–5, 2020.

[102] Burak Yelmen, Aurélien Decelle, Linda Ongaro, Davide Marnetto, Corentin Tallec, Francesco Montinaro, Cyril Furtlehner, Luca Pagani, and Flora Jay. Creating artificial human genomes using generative neural networks. *PLoS genetics*, 17(2):e1009303, 2021.

[103] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

[104] Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. Generating functional protein variants with variational autoencoders. *PLoS computational biology*, 17(2):e1008736, 2021.

[105] Michael Schmidt and Kay Hamacher. Three-body interactions improve contact prediction within direct-coupling analysis. *Physical review E*, 96(5):052405, 2017.

[106] Michael Schmidt and Kay Hamacher. hodca: higher order direct-coupling analysis. *BMC bioinformatics*, 19(1):1–5, 2018.

[107] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[108] Sam Sinai, Eric Kelsic, George M Church, and Martin A Nowak. Variational auto-encoding of protein sequences. *arXiv preprint arXiv:1712.03346*, 2017.

[109] Kristian Davidsen, Branden J Olson, William S DeWitt III, Jean Feng, Elias Harkins, Philip Bradley, and Frederick A Matsen IV. Deep generative models for t cell receptor protein sequences. *Elife*, 8:e46935, 2019.

[110] Nathalie Japkowicz, Stephen Jose Hanson, and Mark A Gluck. Nonlinear autoassociation is not equivalent to pca. *Neural computation*, 12(3):531–545, 2000.

[111] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[112] Thomas Lucas, Corentin Tallec, Yann Ollivier, and Jakob Verbeek. Mixed batches and symmetric discriminators for gan training. In *International Conference on Machine Learning*, pages 2844–2853, 2018.

[113] James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. *arXiv preprint arXiv:1911.02469*, 2019.

[114] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27:3581–3589, 2014.

[115] Jérôme Tubiana, Simona Cocco, and Rémi Monasson. Learning protein constitutive motifs from sequence data. *Elife*, 8:e39397, 2019.

[116] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[117] Alex Bateman, Ewan Birney, Lorenzo Cerruti, Richard Durbin, Laurence Etwiller, Sean R Eddy, Sam Griffiths-Jones, Kevin L Howe, Mhairi Marshall, and Erik LL Sonnhammer. The pfam protein families database. *Nucleic acids research*, 30(1):276–280, 2002.

[118] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[119] Akira R Kinjo, Hirofumi Suzuki, Reiko Yamashita, Yasuyo Ikegawa, Takahiro Kudou, Reiko Igarashi, Yumiko Kengaku, Hasumi Cho, Daron M Standley, Atsushi Nakagawa, et al. Protein data bank japan

(pdbj): maintaining a structural data archive and resource description framework format. *Nucleic Acids Research*, 40(D1):D453–D460, 2012.

[120] BMRB Mallesham, BM Rajesh, P Rajamohan Reddy, D Srinivas, and Sanjay Trehan. Highly efficient cui-catalyzed coupling of aryl bromides with oxazolidinones using buchwald's protocol: a short route to linezolid and toloxatone. *Organic Letters*, 5(7):963–965, 2003.

[121] Miguel A Carreira-Perpinan and Geoffrey E Hinton. On contrastive divergence learning. In *Aistats*, volume 10, pages 33–40. Citeseer, 2005.

[122] Ilya Sutskever and Tijmen Tieleman. On the convergence properties of contrastive divergence. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 789–795, 2010.

[123] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, 2014.

[124] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[125] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

[126] Mathias Berglund and Tapani Raiko. Stochastic gradient estimate variance in contrastive divergence and persistent contrastive divergence. *arXiv preprint arXiv:1312.6002*, 2013.

[127] Jascha Sohl-Dickstein, Peter B Battaglino, and Michael R DeWeese. New method for parameter estimation in probabilistic models: minimum probability flow. *Physical review letters*, 107(22):220601, 2011.

[128] Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071, 2008.

[129] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[130] Asja Fischer and Christian Igel. Empirical analysis of the divergence of gibbs sampling based learning algorithms for restricted boltzmann machines. In *International conference on artificial neural networks*, pages 208–217. Springer, 2010.

[131] Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

[132] Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pages 872–879, 2008.

[133] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019.

[134] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[135] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in \beta $-vae. arXivpreprintarXiv$: 1804.03599, 2018.

[136] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

[137] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[138] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[139] Donatas Repecka, Vykintas Jauniskis, Laurynas Karpus, Elzbieta Rembeza, Jan Zrimec, Simona Poviloniene, Irmantas Rokaitis, Audrius Laurynenas, Wissam Abuajwa, Otto Savolainen, et al. Expanding functional protein sequence space using generative adversarial networks. *bioRxiv*, page 789719, 2019.

[140] Burak Yelmen, Aurélien Decelle, Linda Ongaro, Davide Marnetto, Corentin Tallec, Francesco Montinaro, Cyril Furtlehner, Luca Pagani, and Flora Jay. Creating artificial human genomes using generative models. *bioRxiv*, page 769091, 2019.

[141] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[142] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical image analysis*, 58:101552, 2019.

[143] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[144] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[145] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *arXiv preprint arXiv:1805.11604*, 2018.